

darktable 5.8 user manual

Table of Contents

1. Overview 6
   1.1. user interface 6
      1.1.1. views 6
      1.1.2. screen layout 6
      1.1.3. filmstrip 8
      1.1.4. top panel 8
      1.1.5. bottom panel 9
      1.1.6. keyboard shortcuts 9
   1.2. supported file formats 9
   1.3. sidecar files & non-destructive editing 10
      1.3.1. sidecar files 10
      1.3.2. importing sidecar files generated by other applications 12
      1.3.3. local copies 13
   1.4. an introduction to darktable's workflow 13
      1.4.1. introduction 13
      1.4.2. import & review 13
      1.4.3. process 14
      1.4.4. tone mapping 21
      1.4.5. export 21
2. Lighttable 22
   2.1. overview 22
   2.2. lighttable view layout 22
   2.3. undo/redo 24
   2.4. lighttable modes 24
      2.4.1. filemanager 24
      2.4.2. zoomable lighttable 24
      2.4.3. culling 25
      2.4.4. full preview 26
   2.5. digital asset management 26
      2.5.1. collections & film rolls 26
      2.5.2. image sequence 27
      2.5.3. thumbnails 27
      2.5.4. star ratings & color labels 29
      2.5.5. image grouping 30
      2.5.6. metadata and tagging 30
3. Darkroom 31
   3.1. overview 31
   3.2. darkroom view layout 31
   3.3. the pixelpipe 33
      3.3.1. the anatomy of a processing module 33
      3.3.2. the pixelpipe & module order 34
      3.3.3. the history stack 37
      3.3.4. undo and redo 37
   3.4. processing modules 37
      3.4.1. module header 37
      3.4.2. multiple instances 38
      3.4.3. presets 39
      3.4.4. module controls 41
      3.4.5. curves 43
      3.4.6. wavelets 45
      3.4.7. deprecated modules 51
   3.5. masking & blending 52
      3.5.1. overview 52
      3.5.2. blend modes 53
      3.5.3. masks 56
         3.5.3.1. overview 56
         3.5.3.2. drawn masks 56
         3.5.3.3. parametric masks 60
         3.5.3.4. combining drawn & parametric masks 63
         3.5.3.5. mask refinement & additional controls 64
         3.5.3.6. AI masking 65
         3.5.3.7. raster masks 66
   3.6. organization 66
      3.6.1. overview 66
      3.6.2. module groups 67
      3.6.3. quick access panel 68
      3.6.4. manage module layouts 70
4. Tethering 72
   4.1. overview 72
   4.2. tethering view layout 72
   4.3. examples 73
   4.4. troubleshooting 74
5. Map 75
   5.1. overview 75
   5.2. map view layout 76
6. Slideshow 77
   6.1. overview 77
   6.2. usage 77
7. Print 78
   7.1. overview 78
   7.2. print view layout 78
8. Module Reference 79
   8.1. overview 79
   8.2. processing modules 80
      8.2.1. AgX 80
      8.2.2. astrophoto denoise 88
      8.2.3. base curve 89
      8.2.4. bloom 90
      8.2.5. blurs 90
      8.2.6. censorize 92
      8.2.7. chromatic aberrations 93
      8.2.8. color balance 94
      8.2.9. color balance rgb 97
      8.2.10. color calibration 103
      8.2.11. color contrast 116
      8.2.12. color correction 116
      8.2.13. color equalizer 117
      8.2.14. color harmonizer 119
      8.2.15. color look up table 122

8.2.16. color mapping 123
8.2.17. color reconstruction 124
8.2.18. color zones 125
8.2.19. colorize 129
8.2.20. composite 129
8.2.21. contrast & texture 130
8.2.22. contrast equalizer 132
8.2.23. crop 135
8.2.24. demosaic 137
8.2.25. denoise (profiled) 141
8.2.26. diffuse or sharpen 145
8.2.27. dither or posterize 150
8.2.28. enlarge canvas 151
8.2.29. exposure 152
8.2.30. external raster mask 154
8.2.31. filmic rgb 155
8.2.32. framing 163
8.2.33. graduated density 164
8.2.34. grain 165
8.2.35. haze removal 165
8.2.36. highlight reconstruction 166
8.2.37. highpass 169
8.2.38. hot pixels 170
8.2.39. input color profile 170
8.2.40. lens correction 172
8.2.41. liquify 174
8.2.42. local contrast 178
8.2.43. lowlight vision 180
8.2.44. lowpass 180
8.2.45. LUT 3D 181
8.2.46. monochrome 182
8.2.47. negadoctor 183
8.2.48. orientation 186
8.2.49. output color profile 187
8.2.50. raw black/white point 187
8.2.51. raw chromatic aberrations 188
8.2.52. raw denoise 189
8.2.53. retouch 189
8.2.54. rgb curve 194
8.2.55. rgb levels 195
8.2.56. rgb primaries 196
8.2.57. rotate and perspective 197
8.2.58. rotate pixels 201
8.2.59. saturation curve 201
8.2.60. scale pixels 206
8.2.61. shadows and highlights 206
8.2.62. sharpen 207
8.2.63. sigmoid 208
8.2.64. soften 210
8.2.65. spektrafilm 211
8.2.66. split-toning 219
8.2.67. surface blur 220
8.2.68. tone curve 220
8.2.69. tone equalizer 222
8.2.70. unbreak input profile 225
8.2.71. velvia 226
8.2.72. vignetting 226
8.2.73. watermark 227
8.2.74. white balance 229
8.2.75. (deprecated) basic adjustments 231
8.2.76. (deprecated) channel mixer 232
8.2.77. (deprecated) contrast brightness saturation 234
8.2.78. (deprecated) crop and rotate 235
8.2.79. (deprecated) defringe 238
8.2.80. (deprecated) fill light 239
8.2.81. (deprecated) global tonemap 239
8.2.82. (deprecated) invert 239
8.2.83. (deprecated) levels 240
8.2.84. (deprecated) spot removal 241
8.2.85. (deprecated) tone mapping 242
8.2.86. (deprecated) vibrance 242
8.2.87. (deprecated) zone system 243
8.3. utility modules 244
8.3.1. darkroom 244
8.3.1.1. clipping warning 244
8.3.1.2. color assessment 245
8.3.1.3. duplicate manager 247
8.3.1.4. gamut check 247
8.3.1.5. global color picker 247
8.3.1.6. guides & overlays 248
8.3.1.7. high quality processing 249
8.3.1.8. history stack 249
8.3.1.9. image information line 250
8.3.1.10. mask manager 250
8.3.1.11. module order 254
8.3.1.12. navigation 254
8.3.1.13. raw overexposed warning 254
8.3.1.14. snapshots 255
8.3.1.15. soft proof 256
8.3.2. lighttable 256
8.3.2.1. (deprecated) lua scripts installer 256
8.3.2.2. actions on selection 256
8.3.2.3. history stack 258
8.3.2.4. import 258
8.3.2.5. scripts 262
8.3.2.6. selection 263
8.3.2.7. timeline 263
8.3.3. map 263
8.3.3.1. find location 263
8.3.3.2. locations 264
8.3.3.3. map settings 265
8.3.4. print 265
8.3.4.1. print settings 265
8.3.5. shared 267
8.3.5.1. collection filters 267
8.3.5.2. collections 272
8.3.5.3. copy and paste 275
8.3.5.4. export 276
8.3.5.5. filmstrip 280

8.3.5.6. focus peaking 282
8.3.5.7. geotagging 282
8.3.5.8. image information 284
8.3.5.9. metadata editor 284
8.3.5.10. neural restore 285
8.3.5.11. recently used collections 288
8.3.5.12. scopes 288
8.3.5.13. styles 294
8.3.5.14. tagging 295
8.3.6. tethering 301
8.3.6.1. camera settings 301
8.3.6.2. live view 301
8.3.6.3. session 301
9. Preferences & Settings 302
9.1. overview 302
9.2. backup settings & edits 302
9.3. general 302
9.4. import 304
9.5. lighttable 305
9.6. darkroom 306
9.7. processing 308
9.8. security 311
9.9. storage 312
9.10. miscellaneous 313
9.11. shortcuts 315
9.12. presets 322
9.13. AI 322
9.14. lua options 325
9.15. configuration directory 325
10. Scripting with Lua 327
10.1. overview 327
10.2. darktable's scripts 327
10.3. basic principles 329
10.4. a simple lua example 330
10.5. printing labeled images 330
10.6. adding a simple shortcut 332
10.7. exporting images with lua 333
10.8. building user interface elements 334
10.9. sharing scripts 335
10.10. calling lua from dbus 337
10.11. using darktable from a lua script 337
10.12. lua API 337
11. Guides & Tutorials 338
11.1. developing monochrome images 338
11.2. batch-editing images 339
11.3. other resources 342
12. Special Topics 343
12.1. color management 343
12.1.1. overview 343
12.1.2. display profile 343
12.1.3. rendering method 343
12.1.4. rendering intent 343
12.1.5. darktable's color spaces 344
12.1.6. unbounded colors 346
12.1.7. possible color artifacts 346
12.1.8. darktable's color dimensions 346
12.2. OpenCL 353
12.2.1. the background 353
12.2.2. how OpenCL works 353
12.2.3. activating OpenCL in darktable 354
12.2.4. setting up OpenCL 354
12.2.5. possible problems & solutions 356
12.2.6. scheduling profile 356
12.2.7. multiple devices 358
12.2.8. OpenCL still does not run for me 359
12.3. AI features 359
12.3.1. overview 359
12.3.2. how AI features work 361
12.3.3. GPU acceleration 362
12.3.4. performance & troubleshooting 365
12.4. memory & performance tuning 366
12.5. using darktable-chart 370
12.5.1. overview 370
12.5.2. usage 371
12.5.3. source image 371
12.5.4. reference values 372
12.5.5. process 373
12.5.6. making input images for darktable-chart 374
12.6. program invocation 374
12.6.1. invocation per OS 374
12.6.2. darktable 375
12.6.3. darktable-cli 377
12.6.4. darktable-generate-cache 381
12.6.5. darktable-chart 382
12.6.6. darktable-cltest 383
12.6.7. darktable-cmstest 383
12.6.8. darktable-mcp 383
12.6.9. purge_non_existing_images.sh 384
12.6.10. gimp integration 385
12.7. variables 385
12.8. darktable's color pipeline 389
12.9. midi device support 391
12.10. using a test environment 394
12.11. using multiple computers 396
13. Contributing to the Docs 399
13.1. overview 399
13.2. development & deployment 399
13.3. workflow 400
13.4. styles & conventions 402
13.5. translations setup 408
13.6. translators guide 411

darktable is an open source photography workflow application and raw developer — a virtual lighttable and darkroom for photographers. It manages your digital negatives in a database, lets you view them through a zoomable lighttable and enables you to develop and enhance your raw images.

The source repository for this documentation can be found at Github . Any feedback relating to this documentation can be provided by creating a ticket or a pull request against this repository.

This documentation is released under the GPL 3.0 license.

1. Overview

1.1. user interface

1.1.1. views

The functionality in darktable is separated into six different views:

lighttable
Manage images and collections.

darkroom
Develop a single image.

map
Show geo-tagged images on a map and manually geo-tag new images.

print
Send images to a printer.

slideshow
Display images as a slideshow, processing them on-the-fly.

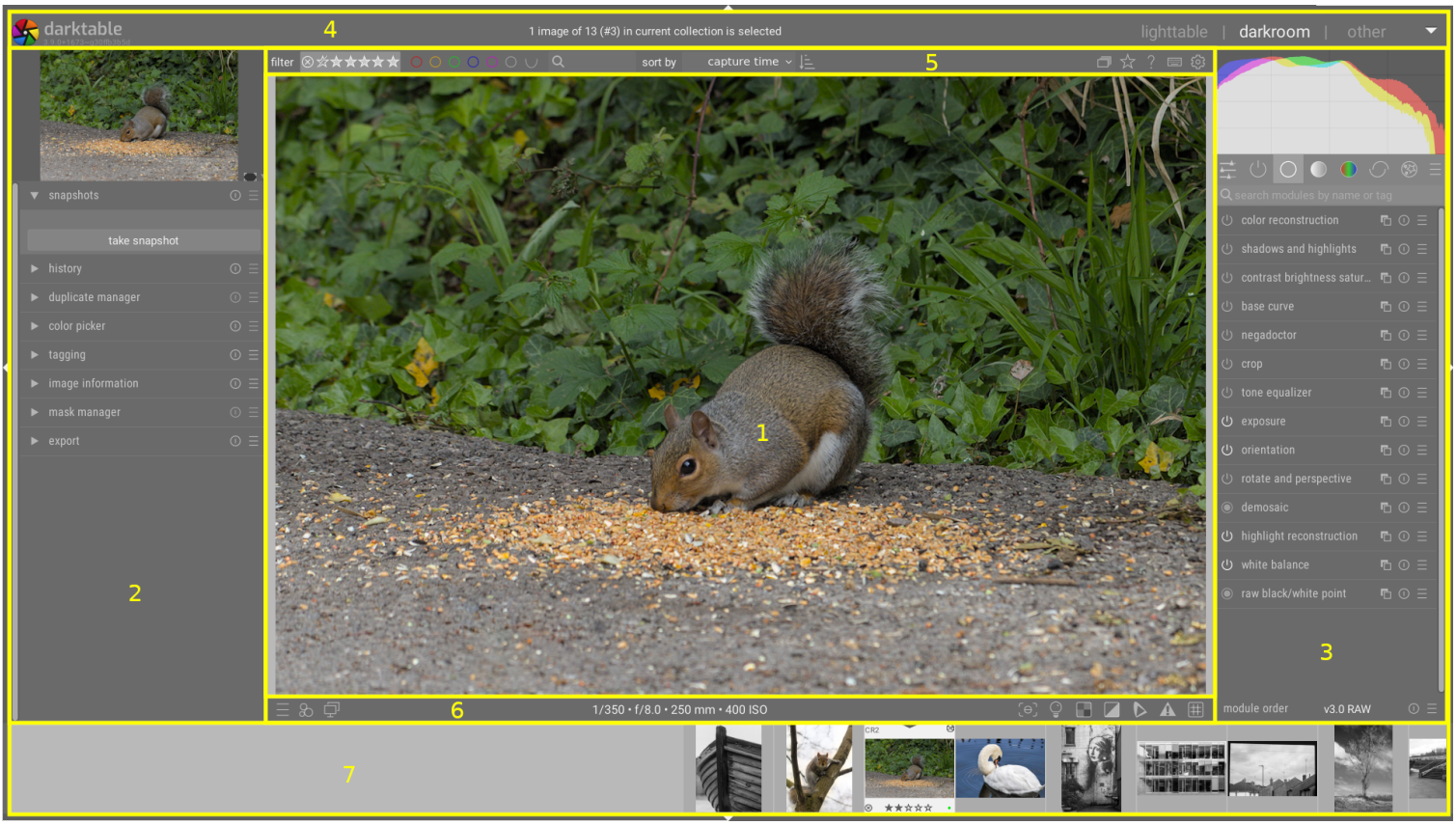
tethering
Remotely capture and save images taken with a connected camera.

You can switch between views by clicking the view name at the top of the right-hand panel (the currently active view is highlighted) or by using one of the following keyboard shortcuts:

- L switches to lighttable
- D switches to darkroom
- M switches to map
- P switches to print
- S switches to slideshow
- T switches to tethering

1.1.2. screen layout

The layout of all darktable views is similar and consists of a center area with panels at the edges:



1. center area : Contains information and functionality specific to the current view.
2. left panel : Contains modules primarily used to provide information.
3. right panel : Contains modules primarily used for image processing.
4. top banner : Contains information about the current darktable version and allows you to switch between views. Also used by some modules to show additional hints and messages.
5. top panel : Provides access to global settings and shortcuts.
6. bottom panel : Provides access to mostly view-specific settings and shortcuts.
7. filmstrip / timeline panel : An optional panel that can be enabled at the bottom of the screen to display a timeline (in the lighttable view) or a filmstrip (in other views) of images in the current collection.

See screen layouts specific to the lighttable view or darkroom view .

panel size and visibility

The left, right and filmstrip/timeline panels can be resized by dragging their inner borders.

Each of the panels can be expanded or collapsed by pressing the triangle located at the outside edge of the panels. Panel visibility can also be adjusted using keyboard shortcuts, as follows:

- TAB temporarily expands the centre view to fill the whole window. Press again to return to the previous view.
- F11 toggles fullscreen mode
- `Shift+Ctrl+t` toggles the top panel (between the image and the top banner)
- `Shift+Ctrl+b` toggles the bottom panel (between the image and the filmstrip/timeline, if shown)
- `Shift+Ctrl+l` toggles the left panel
- `Shift+Ctrl+r` toggles the right panel
- `Ctrl+f` toggles the filmstrip/timeline
- `Ctrl+h` toggles the top banner
- b toggles all borders and panel-collapse controls

Note: The size and visibility of the panels are stored independently for each view.

1.1.3. filmstrip

The filmstrip, when enabled, is shown at the bottom of the screen (except in the lighttable view, where it is replaced with the timeline module) and displays the images from the current collection (set in the lighttable view). You can navigate the filmstrip by scrolling with the mouse wheel.

The filmstrip allows you to interact with images while you are not in the lighttable view. For example, while developing an image in darkroom mode, you can switch to another image by double-clicking its thumbnail in the filmstrip. You can also rate and color-classify the images as you do in lighttable, as well as copy & paste the history stack using keyboard shortcuts. As as in the lighttable view, you can drag and drop thumbnail images to change images' sequence when using a custom sort order.

See the filmstrip module documentation for more information.

1.1.4. top panel

The top panel appears in all darktable views and provides access to common functions.



On the left-hand-side

filter

Choose which of a collection's images are displayed, making them available to different views for selection and action . Criteria can be altered in the collection filters module or by clicking the icon.

sort by

Choose the property that orders the displayed image sequence .

sort order

Switch the sort order (ascending / descending).

On the right-hand-side

grouping

Expand or collapse grouped images .

thumbnail overlays

Define what information is overlaid on to thumbnails in the lighttable/filmstrip.

You can define different settings depending on the thumbnail size. See preferences > lighttable for details of how size delimiters are set.

context-sensitive help

Click this icon and then click on a control element to be directed to the appropriate online help page.

shortcut mapping

Click this icon to enter the visual shortcut mapping mode to create keyboard/mouse shortcuts.

workspace

Only shown when preferences > storage > allow for multiple workspaces is enabled. Click to open a menu to switch workspaces or create a new one (blank by default, with an optional copy of settings from an existing workspace). Switching or creating relaunches darktable into the chosen workspace.

preferences

View and amend darktable's preferences & settings .

1.1.5. bottom panel

The bottom panel appears in all darktable views. Common controls are described here. View-specific controls are described in the appropriate pages:

- lighttable view
- darkroom view
- map view
- tethering view

common controls



view log history

Display a scrollable popover showing stored log messages with timestamps. The log history viewer provides a persistent record of application log messages (e.g., export/import progress, library operations, warnings, and errors). The key features are:

- Messages are stored in a circular buffer; when the buffer is full, older entries are removed as new ones arrive
- Consecutive identical messages are automatically deduplicated (e.g., repeated “exporting image” updates)
- Entries are displayed in monospace format with HH:MM:SS timestamps
- Individual entries are selectable and can be copied to the clipboard with Ctrl+C
- The viewer is closed by clicking anywhere outside the popover

1.1.6. keyboard shortcuts

Much of the functionality in darktable can be controlled via keyboard shortcuts, which can be customised in preferences > shortcuts .

Press the H key (for help) in any darktable view to show a list of all shortcuts that are applicable to the current view.

1.2. supported file formats

darktable supports a huge number of file formats from various camera manufacturers. In addition darktable can read a wide range of low- and high-dynamic-range images – mainly for data exchange between darktable and other software.

In order for darktable to consider a file for import, it must have one of the following extensions (case independent): `3FR, ARI, ARW, BAY, BMQ, CAP, CINE, CR2, CR3, CRW, CS1, DC2, DCR, DNG, GPR, ERF, FFF, EXR, IA, IIQ, JPEG, JPG, JXL, K25, KC2, KDC, MDC, MEF, MOS, MRW, NEF, NRW, ORF, PEF, PFM, PNG, PXN, QTK, RAF, RAW, RDC, RW1, RW2, SR2, SRF, SRW, STI, TIF, TIFF, X3F`

If darktable was compiled with JPEG2000 support, the following extensions are also recognized: `J2C, J2K, JP2, JPC, JPF, JPX`.

If darktable was compiled with GraphicsMagick support, the following extensions are also recognized: `BMP, CIN, DCM, DPX, GIF, JNG, JPC, JP2, JPF, JPX, MIFF, MNG, PBM, PGM, PNM, PPM, WEBP`.

camera raw files

darktable reads raw files using the open source library RawSpeed , originally developed by Klaus Post and now maintained as part of the darktable project. If RawSpeed cannot load the file, darktable will attempt to load it with the LibRaw library. The number of supported cameras and file formats is constantly increasing. Most modern camera models are supported, and new ones tend to get added very quickly. It is beyond the scope of this manual to give an exhaustive list.

With the exception of Fujifilm X-Trans cameras, darktable does not decode images from cameras with non-Bayer sensors (e.g. Sigma cameras with the Foveon X3 sensor).

other image files

darktable natively reads “ordinary” images in JPEG, 8-bit/16-bit PNG and 8-bit/16-bit TIFF format, as well as 16-bit/32-bit floating point TIFF formats.

darktable also reads high dynamic range images in OpenEXR, RGBE and PFM formats.

1.3. sidecar files & non-destructive editing

1.3.1. sidecar files

darktable is a non-destructive image editor and opens all images in read-only mode. Most data created within darktable (metadata, tags, and image processing steps) is stored in separate XMP and DTDATA *sidecar* files. These files are stored alongside the original Raw files and allow darktable to store information about the images as well as the full editing history without touching the original raw files. When you import an image into darktable for the first time, an XMP file is automatically generated. The generation of sidecar files can be disabled in preferences > storage > XMP sidecar files but this is not recommended in normal use.

When configured to use XMP sidecar files, darktable may also store some data in DTDATA sidecar files .

Note: custom sort sequence numbers are not stored in sidecar files. This could be destructive if imported into another darktable image library – including their original library – after import, duplication, or image resequencing.

For a given source image, multiple editing versions, called *duplicates*, can co-exist, sharing the same input image data but each having their own metadata, tags and processing steps. Each duplicate of a given image (named `<basename>.<extension>`) is represented by separate sidecar files (with a filename constructed in the form `<basename>_<number>.<extension>.<xmp or dtdata>`, where `<number>` represents the version number of that edit). Information for the initial edit – the “duplicate” with version number zero – is stored in the sidecar file named `<basename>.<extension>.xmp`. The version number of each duplicate is displayed in the image information module in each of darktable’s views.

Your work is automatically synchronised to the sidecar files without the need to press a “save” button. When backing up your data, make sure that you also retain copies of the XMP files, as these are required to fully reconstruct your work in case of a disaster.

In addition to the sidecar files, darktable keeps all image-related data (apart from the DTDATA information) in its library database for fast access. An image can only be viewed and edited from within darktable if its data has first been loaded into the library database. This happens automatically when you first import an image. If an image is subsequently re-imported, the database will be updated from the contents of its XMP file.

Once an image has been imported into darktable, the database entries take precedence over the XMP file. Subsequent changes to the XMP file by any other software are not visible to darktable – such changes will be overwritten the next time darktable synchronizes the file. On request, darktable can be configured to search for updated XMP files at startup, offering a choice to update the database or overwrite the XMP file where changes are identified. This configuration can be changed in preferences > storage > XMP sidecar files .

To import XMP sidecar files created by other applications, see importing sidecar files generated by other applications .

To load an XMP sidecar file (e.g. one received from another darktable user), see load sidecar file .

drag and drop XMP sidecar files

You can drag and drop XMP sidecar files to the current image in the darkroom. The message *drop XMP sidecar files here to apply edits* will be displayed. On releasing the mouse button, a dialog will be displayed with options, which depend on the number of files being dragged:

single sidecar file

cancel
cancels without making any changes.
create duplicate
create a new duplicate of the current image using the history stack in the dragged file.
replace history
replaces the history stack of the current image with the history stack in the dragged file.

multiple sidecar files

cancel
cancels without making any changes.
create
creates new duplicates of the current image using the history stack in the dragged files, one per sidecar file.

DTDATA sidecar files

If the creation of sidecar files is enabled in preferences > storage > XMP sidecar files then, in addition to the XMP sidecar files, darktable will store additional data in DTDATA sidecar files.

The DTDATA sidecar holds per-pixel edit data such as raster masks. It is only created for images that use such data, and only when sidecar writing is enabled; most images never get one. It follows the image when it is moved, duplicated or deleted using darktable.

1.3.2. importing sidecar files generated by other applications

When importing an image, darktable automatically checks if it is accompanied by an XMP sidecar file. As well as looking for files named `<basename>.<extension>.xmp` and `<basename>_<number>.<extension>.xmp` (darktable's XMP file naming formats) darktable also checks for the presence of a file in the form `<basename>.xmp` (the naming format for Lightroom's XMP sidecar files). Files with the latter naming format will be read by darktable but will not be written to. Once the image has been imported, darktable will generate an additional XMP file using its own naming convention.

At present, darktable is able to load the following metadata from Lightroom-generated XMP sidecar files during the import process:

- tags (including hierarchical tags)
- color labels
- ratings
- GPS information

In addition, darktable has been designed to help migrate some image operations from other specific applications. The aim here is not to make darktable a drop-in replacement for any other software, but rather to help you to recover part of the work you have already invested into your image. It is important to understand that the import process will never give identical results to other software. The underlying processing engines are very different from application to application, and depend a lot on the individual image. In some cases, the results may be similar but often, further adjustment will be required in darktable.

This migration happens automatically when entering the darkroom view, provided that a corresponding XMP sidecar is found.

At present, darktable is able to handle the following development steps from Lightroom-generated XMP files (with the corresponding darktable module in parentheses):

- crop (*crop*)
- straightening angle (*rotate and perspective*)
- black level (*exposure*)
- exposure (*exposure*)
- vignette (*vignetting*)
- clarity (*local contrast*)
- tone curve (*tone curve*)
- HSL (*color zones*)
- split-toning (*split-toning*)
- grain (*grain*)
- spot removal (*retouch*)

Lightroom's exposure and black level adjustments are added to darktable's default exposure rather than replacing it. The working profile in the *input color profile* module is set to linear ProPhoto RGB, the color space used by Lightroom. Tone curves that would not change the image are not imported.

History items created by this process are labeled "lr import" in the *history stack* .

1.3.3. local copies

Many users have huge image collections stored on extra hard drives in their desktop computer, or on an external storage medium (RAID NAS, external hard drives etc.).

It is a common requirement to develop a number of images while travelling using a laptop and then later synchronize them back to the original storage medium. However, copying images manually from the main storage to the laptop and back is cumbersome and prone to errors. The “local copies” feature of darktable has been designed to directly support these use cases.

You can create local copies of selected images from within the lighttable. Local copies are always used when present, giving continued access to images even if the external storage is no longer connected. At a later point, when your primary storage medium has been reconnected, you can synchronize the XMP sidecar files back to this storage, deleting any local copies. These operations can be found in the actions on selection module in the lighttable.

For safety reasons, if local copies exist and the external storage is available, the local XMP sidecars are automatically synchronized at start up.

Local copies are stored within the cache directory and named `img-<SIGNATURE>.<EXT>` (where `SIGNATURE` is a hash signature (MD5) of the full pathname, and `EXT` is the original filename extension).

Local copies can be identified in the lighttable view by a white marker on the top right of the thumbnail. In addition, all local copies carry the `darktable|local-copy` tag to allow them to be easily selected.

1.4. an introduction to darktable's workflow

1.4.1. introduction

This section is intended to provide you with a flavor of what darktable can do and offer a suggested end-to-end processing workflow. New users should start by working through this introduction and following the links provided for further details.

Please note that this is not an exhaustive guide and many useful topics are not covered. You are advised to read through the reference sections of this manual once you are familiar with the basics.

Further information on most of the subjects in this guide are also covered in the other resources section.

1.4.2. import & review

Before you can do anything in darktable you must first add some images files to the library using the import module in the lighttable view . This will create entries for your images in darktable’s library database so that it can keep track of the changes you make. There are three ways to import images, each accessible through buttons in the import module:

add to library
: This option adds images to the library without copying or moving – your original files will stay in their current location and will not be altered. On import, darktable will read the metadata from the image files and any accompanying XMP sidecar file . If an image has already been added to the database, any updates you have made to the sidecar file will be loaded.

copy & import
: Copies the images to the storage location (following the file naming pattern defined in preferences > import), then adds the *copied* images to the library – the original images are not changed or moved. Only the images are copied and if an existing XMP sidecar file is available for the image, it will *not* be read, copied or used.

copy & import from camera

Connect the camera to your system with a USB cable (if your camera is auto-mounted by your system, you will need to un-mount it before it can be accessed by darktable). If you don't see your camera listed in the import module, press the "scan for devices" button. Once your camera is detected the import module should offer the ability to *copy & import* images from the camera. Clicking the "copy & import" button physically copies the selected images from the camera into a specified directory (following the file naming pattern defined in preferences > import) and then adds the copied images to the library.

Once images have been imported, their thumbnails are displayed in the lighttable view, within which you can organize and catalog your imported images – please refer to the digital asset management section for more information.

The primary use for the lighttable view is to review your images and decide which you would like to edit further and which to discard. The following is a possible culling process to choose which images to edit/delete:

1. Set the lighttable view to only show images with a rating of exactly 1 star using the *view* setting on the top panel . Since darktable assigns a 1 star rating to newly-imported images by default, this will show all of the images you have just imported. You can use the collections module to further refine your selection if needed.
2. Perform a quick first-level screening of your images: If any images are badly out-of-focus or otherwise unwanted, *reject* them or give them a 0-star rating (by pressing the "R" or "0" keys, respectively, while hovering over the image with your mouse). If you would like an image to pass to the next review phase, press 2 to give it a 2 star rating. Any images that no longer have a 1 star rating will automatically disappear from view. Continue in this manner until no more images remain visible in the lighttable view.
3. Alter the lighttable view to only show images with 2 stars. Go through these images more carefully, and decide whether to promote them to a 3 star rating, or put them back down to a lower rating. Again, proceed until no more images remain in the lighttable view.
4. Alter the lighttable view to only show images with 3 stars. Perform quick edits and experiments on those images in the darkroom view to decide if they are worthy of further effort. If you are happy with the results of those edits, promote that image to a 4 star rating for final editing.
5. Go through your 4 star images, perform any final edits on them and then export to see the final result. Increase the rating a final time to 5 stars to indicate that processing is complete.

If space is at a premium you might consider permanently deleting your rejected or 0-star images. Select these images in the lighttable view and use the 'trash/delete' option in the actions on selection module. You should only do this for images you are certain you will never need again.

1.4.3. process

This section is intended to get you comfortable processing images in the darkroom view using a *scene-referred* workflow. You are advised to follow the guidelines provided below, up to the end of the *image processing in 3 modules* section and then choose other areas to learn as-and-when you need to use those techniques in your images.

getting started

take a well-exposed photograph

Good image processing techniques start in the camera – a well-exposed image (without blown highlights or heavily crushed blacks) will always make post-processing much more straightforward. Under- or over-exposure can be "fixed" by darktable to some extent but no software can recover information that is not present in the Raw image (clipped highlights). Where possible, you are advised to use exposure to-the-right (ETTR) techniques to maximize the amount of data available for processing while avoiding clipping. As a general rule of thumb, in cases where the scene-dynamic-range exceeds that of your camera, it is safe to underexpose all images by 0.5 to 1 EV (by reducing the ISO sensitivity if possible) even if the in-camera preview looks darker than expected (the preview is not the raw data). If the scene dynamic range is lower than that of your camera, you may wish to dial in some over-exposure (decrease shutter speed or increase aperture) to capture more light and reduce noise.

scene-referred workflow: a new approach

If you have used other Raw software in the past (or darktable prior to version 3.0) you may notice some significant differences from what you are used to – darktable now uses a *scene-referred* approach for most of its processing modules. This approach is used extensively in cinematography and is known to be much more robust than the traditional *display-referred* approach.

In *display-referred processing* the data from your Raw file is initially compressed into a range that represents pure black as 0 and pure white as 1, fixing mid-gray at 0.5. A tone curve is automatically (and irreversibly) applied to this data to make the image look “good” on your display and subsequent edits are carried out on top of this already highly-modified image data. The cost of *display-referred* is an early loss of the relationship between pixel luminosity and saturation (usually also involving hue shifts), which is responsible for the infamous “HDR look” when the dynamic range increases.

In the real world, “pure black” does not really exist (there is always *some* light) and there is no limit to how bright things can be (so no “pure white” either). *Scene-referred processing* attempts to retain the physical properties of the scene for as long as possible by placing the Raw data on an unbounded linear scale and only compressing the data to the dynamic range of your display after image processing is complete.

In a scene-referred workflow many common tools (tone curves and levels, for example) are no longer useful ways to manipulate the image, since they rely on now-invalid definitions of black, white and gray. Experienced users may need to learn new techniques and discard old ones, but will be rewarded with much more robust and predictable outputs.

The scene-referred workflow in darktable enables the filmic rgb and exposure modules by default when you open new images in the darkroom view.

white balance and color calibration

Most processing software uses a traditional temperature/tint model for adjusting the white balance of an image. In darktable, the color calibration module provides a much more robust and flexible approach, allowing you to explicitly define the color of the light source. This is particularly useful for scenes illuminated by artificial lighting.

Please note that the white balance module is still enabled in this approach, but its settings normally should not be altered.

Set preferences > processing > auto-apply pixel workflow defaults to “scene-referred (filmic)” now.

Enter preferences by clicking on the gear icon in the top panel .

edit in a controlled environment

Image processing should be performed in a controlled environment, lit by a white light source against a background approximating mid-gray, and on a monitor that has been properly calibrated.

While this may not be practical in many home editing environments you *can* control the background colors on your monitor’s display. You should set the darktable color scheme to use one of the “grey” themes and use the color assessment mode when altering tones and colors in your image. Dark themes might look good but, unless you are processing images to be viewed on a cinema screen in a darkened room, they should not be used for photo processing.

Set preferences > general > theme to “darktable-elegant-grey” or “darktable-icons-grey” now.

---

enter the darkroom

Choose an image to edit from the lighttable view and double-click to load that image into the darkroom view. For now try to choose an image that is well exposed – we will discuss some techniques to recover badly-exposed images later.

In the darkroom view, you will see a list of processing modules to the right of your image. Each module performs its own processing on the image, in the order shown in the module list, starting at the bottom of the list and moving up to the top. You can think of this like a stack of building blocks where each block builds on the processing performed by the modules below it.

On the left hand side is the history stack (you may need to expand the module), which shows the order in which adjustments were made to the controls of the various modules. This allows you to undo changes by reverting to an earlier step in the history stack. You will see that a number of modules are applied automatically – these are needed in order to generate a legible image from the Raw data.

It is important to understand the distinction between the order of the modules on the right-hand-side of the screen (which represents the order in which modules are *executed*) and the order of the modules in the history stack (which represents the order in which modules were *modified*).

To the top right is the scopes module, which shows the spread of tones/colors in your image.

If you have previously viewed or edited the image in the darkroom view, start by *discarding history* (click the *reset* button in the *history stack*). This will reapply defaults using your new settings and provide a clean starting point for editing.

---

why doesn’t the raw image look like the JPEG?

...because you haven’t processed it yet

One of the first things people notice when switching from lighttable to darkroom view is that the image looks different – often flatter and less saturated than that shown in the lighttable view. This is because the darkroom view displays the (mostly unprocessed) Raw image, but the lighttable view initially displays the (in-camera) JPEG preview. Now that you have opened the image in the darkroom view, the lighttable view will update to show the edited version.

Most Raw software goes to great lengths to reproduce the look of standard camera JPEGs out of the box. While this can be useful (if you only want to make very minor adjustments to the camera’s rendition of an image) we assume that you are using a Raw editor to make the image your own, and that the camera does not know how to do this. Certainly, if you are using the ETTR techniques mentioned above, the camera JPEG will rarely be close to how you want the final image to look.

The default settings in darktable are therefore intended to provide you with a neutral *starting point* for further editing and nothing more. *We do not intend to change this*.

module groups

Below the scopes module, at the top right of the screen, is a set of tabs into which darktable’s modules are grouped . If you cannot find a module in one of the tabs you can use the search feature to locate it.

For the purposes of this guide, click on the hamburger icon (to the right of the tabs) and select the “workflow: scene-referred” preset now.

---

image processing in 3 modules

The following basic adjustments are fundamental to scene-referred editing and will be required, to some extent, on the majority of images. You can usually produce a good-looking image with these steps alone.

As you will be adjusting the tones and colors of the image, start by enabling color assessment mode (press Ctrl+B) and perform the following edits on the zoomed-out image while in this mode.

1. **Set overall image brightness**: First, set the overall (average) brightness of the image (the mid-gray point) by adjusting the *exposure* slider in the *exposure* module. This is a purely artistic setting and should be defined based on your intent – for example, for a high-key image you will set the average brightness to be lighter than for a low-key image. The color assessment mode provides you with two reference points to assist with this by surrounding the image with a white frame against a middle-gray background.

   At this point, don't worry if the brightest parts of your image lose detail – this can be recovered in the next step.

   Note: *The lens correction module can also affect the image brightness so you may want to consider enabling it before adjusting exposure.*

   ---

2. **Set white and black points**: The next two steps use the *filmic rgb* module to define how the tones in your image will be mapped to the dynamic range of your display. Start by setting the *white* and *black* relative exposure sliders in the *scene tab* . These are purely technical settings, defining white and black relative to the mid-gray point you set in the previous step. If your image contains tones you want to treat as pure white or pure black you can use the pickers beside the sliders to set these values (using the maximum and minimum brightness of the image). Otherwise set the values manually using the color assessment frames as a reference.

3. **Adjust the contrast**: Now move to the *look tab* in *filmic rgb* (for now we will skip the *reconstruct tab*). Enable the *look only* view at the top of the module to see a representation of the filmic tone curve, which consists of a straight section in the middle (used to set the contrast of the mid-tones) and curved sections at the top and bottom (where the shadows and highlights are compressed to fit the dynamic range of the display).

   The *contrast* slider changes the slope of the straight section (the mid-tone image contrast), the *linear region* slider changes its length and the *shadows/highlights balance* slider changes its position. There is a lot of give-and-take involved here – if you want to increase the contrast of the mid-tones, you must sacrifice contrast in the shadows/highlights and vice versa. The default settings of this module are tuned to work for the majority of images but you should experiment with these sliders to understand how they affect the image.

   Note: *The highlight compression in the filmic rgb module can cause detail to be lost in the highlights. You can mitigate this to some extent by reducing the white relative exposure, adjusting the shadows/highlights balance or changing the contrast in highlights setting in the options tab . The tone equalizer module can also be used to reduce the relative brightness of the sky.*

   ---

4. **Color preservation**: The tone mapping in the filmic rgb module attempts to redistribute the tones in your image without affecting color reproduction. While the default color preservation algorithm works for most images, you can experiment with the *highlights saturation mix* setting in the *look tab* if you do not like how the colors appear.

5. **Saturation**: Your image will probably not look very colorful at this point. You can adjust the global saturation of the image using the color balance rgb module. The "basic colorfulness" preset should provide you with generally-reasonable defaults, but you are encouraged to experiment further with these settings as required.

   Note: *This guide assumes that the white balance of the image has been correctly captured by your camera. If this is not the case, you may need to make some corrections in the color calibration module first.*

You can now switch off *color assessment mode* by pressing Ctrl+B again.

other processing techniques

With practice the above workflow can quickly provide you with a reasonable-looking image, though most will need some additional work before they are ready for export. The following sections are intended to provide a brief outline of some more image processing techniques in darktable, with links to the relevant reference sections for more information.

As a general rule, you should begin with the basic steps outlined in the previous section, then perform *corrections* and finish with *creative adjustments* .

corrections

color calibration

Traditional white balance correction attempts to ensure that whites and grays are really neutral (R = G = B) and doesn't really try to manage the impact on other colors. The *CAT tab of the color calibration module* extends this treatment to handle the remainder of the color range and works in a color space designed specifically for chromatic (color) adaptation. As with traditional white balance controls you can select a patch of neutral gray in your image to calculate the white balance, or use a selection of other automatic and manual methods. The default settings use the white balance from the image's Exif data and are usually sufficient.

If you need to make adjustments in the *color calibration* module, you may want to also revisit any saturation corrections you made earlier in the *color balance rgb* module.

correct lens distortions

All lenses introduce some artifacts (distortion, chromatic aberrations, vignetting) to the image. The *lens correction* module can correct many of these issues for a wide variety of lenses. The *chromatic aberrations* and *raw chromatic aberrations* modules can also be used to handle chromatic aberrations for lenses that are not (or only partially) supported by *lens correction*. In most cases simply enabling the *lens correction* module will auto-detect your lens and auto-apply all available corrections.

If you decide to use the *lens correction* module, it should be enabled at the start of your edit, before you adjust exposure, since vignetting correction can alter the overall brightness of your image.

reduce noise / retain detail

At a pixel level there is a trade-off to be made between the retention of fine details and the reduction/removal of sensor noise. In most situations, a small amount of noise is perfectly acceptable and will not be noticeable except when you zoom in to 100%. At this scale you are not viewing the image at a realistic size – even when represented on a large monitor or print, noise that is obvious at high zoom factors will be virtually invisible on the final image. However, some modules further along the image pipeline (especially those that increase local contrast) may end up exaggerating any noise that is present so, again, there are trade-offs to be made.

The first module you can use to manage this is *demosaic* , which controls how the single-color (R, G or B) pixels in your Raw file are converted to image pixels combining all three colors. Each demosaic algorithm has its own trade-offs between retaining fine detail and reducing noise. The default demosaic algorithm (RCD) usually provides a reasonable compromise.

Demosaic algorithms can only do so much to manage noise in your image. The *denoise (profiled)* module is individually tuned for a number of common camera sensors and can be used to reduce or remove pixel noise. As with demosaic you should alter the settings until you are happy with the balance of denoising vs fine-detail reproduction. The default settings are usually sufficient.

sharpness and local contrast

A number of modules can be used to adjust the local contrast and sharpness of your image. Most of these modules aim to enhance the apparent contrast of edges and do not add “real” sharpness (they are not the same as lens deconvolution). You should take care when using these modules as most of them can introduce artifacts (such as halos) when settings are pushed too far:

- the *contrast equalizer* module allows you to adjust contrast, limiting the effect to certain feature sizes. For example you can use it to increase the contrast of fine details without impacting larger-scale objects, or vice versa
- the *diffuse or sharpen* module offers a number of presets for sharpening, lens deblurring and the addition of local contrast
- the *local contrast* module provides a simpler interface for quickly adding local contrast to your images – just enabling the module or selecting one of the presets is often all that is required
- the *sharpen* module is intended to re-introduce sharpness that was removed by your camera’s anti-alias filter (if present) and can be enabled by default in preferences > processing . The methods listed above are usually preferred to this legacy module.

As with the modules mentioned in the previous section, you should take care when adding contrast to small-scale objects – an image viewed at 100% is not a realistic representation of your final edit and local contrast adjustments are usually better judged when zoomed out.

reconstruct blown highlights

While a well-exposed image will make post-processing much easier, darktable does provide a few tools to handle blown highlights.

The *highlight reconstruction* module attempts to reconstruct blown highlights (colors and structure) using adjacent pixels. A number of different approaches are provided, some of which may be better on certain images, however, the default algorithm produces good results in most cases.

Even well-reconstructed highlights can show color and edge artifacts, some of which may be exacerbated by subsequent modules in the pipe. In this case the *reconstruct tab on the filmic rgb module* provides additional methods to further smooth/correct highlights at the end of the processing pipeline.

adjust angle and perspective

The *rotate and perspective* module can be used to adjust the angle of the image or to simulate the functionality of a tilt/shift lens by altering the perspective, making converging horizontal and/or vertical lines parallel (keystone correction). This latter technique is most commonly used for architectural photography. If you just want to correct the angle of the horizon you can do this by right-clicking and dragging along the horizon line.

remove spots and unwanted objects

Use the *retouch* module to remove unwanted objects by replacing pixels with detail from elsewhere in the image. This module also offers powerful techniques for removing large-scale objects (such as spots or blemishes) while leaving fine-scale details (like hairs and follicles) intact. The most common use for this module is to remove dust spots from images or blemishes from skin.

remove atmospheric haze

There are two methods for removing atmospheric haze in darktable. The haze removal module provides a much simpler interface, but the “dehaze” preset in the *diffuse or sharpen* module can provide more flexibility when needed.

creative adjustments

crop and frame

Use the *crop* module to crop your image and the *framing* module to surround your image with a colored frame. Both modules can be set to use a predefined or custom aspect ratio – for example, you could place a square-cropped image into a 3:2 frame.

dodge and burn

Dodging and burning is a traditional darkroom technique to add and remove brightness from an image. There are two recommended ways to achieve this

- If you want to selectively dodge or burn only certain objects you can apply a new instance of the *exposure* module using a drawn mask to isolate the effect to the required area of the image (see also the mask refinement section for more information). Move the exposure slider to alter the brightness of the masked area.
- If you wish to dodge or burn areas with a similar brightness (for example, to brighten the shadows or darken the highlights) you should use the *tone equalizer* module.

convert to monochrome

darktable provides a number of ways to remove the color from your image. The most flexible method is to use the *gray tab of the color calibration module* . A number of film emulation presets are available in this module to provide you with a starting point.

See the developing monochrome images section for details of other techniques.

color grading

The *color balance rgb* module is your one-stop-shop for controlling the colors in your image. Adjustments can be isolated to the shadows, highlights and mid-tones, or applied at a global level.

other important topics

reuse common module settings

If you find yourself using the same module parameters repeatedly, you can create presets containing your favorite settings. If you use the same settings on every image, you can also make presets apply automatically to new images. For example, you may find yourself adding the same exposure settings to every image taken by a certain camera – in this case you can create a preset that automatically applies those corrections only to images from that camera.

You may also have groups of module settings that you commonly apply only to certain types of image. You can use styles to apply multiple module settings at once to a selection of images.

perform adjustments locally

Most darktable modules can either be applied to the whole image, or restricted to parts of the image using drawn and parametric masks .

control darktable with other input methods

You don’t need to use the darktable UI to make adjustments to your images. Most of the functionality in darktable can also be controlled using shortcuts defined with your keyboard/mouse and even with other input devices such as game controllers and MIDI devices. See the shortcuts section for details.

1.4.4. tone mapping

Camera sensors are capable of capturing a wider dynamic range than can be displayed on a monitor or print. So at some point during processing the cameras data has to be compressed or clipping will occur. This is called *tone mapping* or sometimes a *display transform*. You may be familiar with the term *tone mapping* from HDR-processing but it is not limited to that field: every JPEG produced by your camera and every raw processing application has to tone-map at some point. In many other raw-processors, the tone mapping is done early in the processing pipeline. In darktable's scene referred approach it is usually one of the last modules in the module order.

Typically exactly one tone mapper is enabled for an image, but darktable allows you to be creative and have zero, one, or more tone mappers enabled. For instance, scenes with a low dynamic range can work well without any tone mapper.

darktable has four scene-referred tone mappers:

- *AgX*
- *filmic rgb*
- *sigmoid*
- *spektrafilm*

In the display-referred workflow, which darktable offered as a default in the past, *base curve* is used early in the pipeline.

You can enable a default tone mapper in preferences > processing > auto-apply pixel workflow defaults .

1.4.5. export

darktable is a non-destructive editor, which means that all changes are recorded in the library database (with a backup stored in an XMP sidecar file), and the original Raw file is left untouched. You therefore need to export images in order to bake your edits into an output file that can be distributed outside of darktable.

1. Choose an export scenario.

   The export module offers many options, but by far the most common use is to "save a developed raw image as a JPEG". You can either export the currently-edited image directly from the darkroom view or select one or more images from the lighttable view and export them all at once.

2. Select which images to export (if you are in the lighttable view), open the export module, set *target storage* to "file on disk" and select a location to save your images – by default, they will be exported to a "darktable_exported" directory within the directory that contains your Raw file(s). Choose a "file format" of JPEG and keep the default settings.

3. If the exported collection's image sequence is important (e.g. a slideshow or to merge photos into a document) and not reflected by file names, select the appropriate "sort by" criteria and include `$(SEQUENCE)` in the export file name template.

4. Click the "export" button to save your processed images in the selected location.

Note: While JPEG is useful for most purposes, if you wish to perform further edits in a raster editor like GIMP or Krita, it is normally better to export in TIFF format.

---

2. Lighttable

2.1. overview

The lighttable view allows you to view and manage your image collection.

The centre view contains thumbnails of your images – how they are displayed depends on which mode you are working in.

While the mouse is over an image thumbnail or images are selected, there are a number of actions you can perform:

keyboard shortcuts

The following keyboard shortcuts are enabled by default in the lighttable view:

- `F1, F2, F3, F4, F5` adds or removes a color label (red, yellow, green, blue, purple, respectively). A color label will be added if *any* selected image does not currently have the label; otherwise the label will be removed
- `0, 1, 2, 3, 4, 5` sets the star rating
- `R` rejects the image(s)
- `Ctrl+D` duplicates the image(s)
- `Ctrl+C` copies the full history stack
- `Ctrl+V` pastes all of the copied history stack
- `Ctrl+Shift+C` selectively copies the history stack
- `Ctrl+Shift+V` selectively pastes from the copied history stack
- `D` opens the image in the darkroom view for developing
- `W` fully zooms into the current image while the key is pressed
- `Ctrl+W` fully zooms into the current image and show areas in focus

drag-and-drop

You may change the sequence of images displayed, processed, or exported when *sort by* is set to custom sort by dragging and dropping their thumbnails.

touchpad

In lighttable culling layouts, touchpad zooming and panning gestures can be used:

- pinch with two fingers to zoom
- scroll with two fingers to pan

2.2. lighttable view layout

left panel

From top to bottom:

import
Import images from the filesystem or a connected camera.
collections
Filter the images displayed in the lighttable center panel – also used to control the images displayed in the filmstrip and timeline modules.
recently used collections
View recently used collections of images.
image information
Display image information.
lua scripts installer **(optional)**
Install lua scripts.

right panel

From top to bottom:

selection
Select images in the lighttable using simple criteria.
actions on selection
Perform actions on selected images.
history stack
Manipulate the history stack of selected images.
styles
Store an image's history stack as a named style and apply it to other images.
metadata editor
Edit metadata for selected images.
tagging
Tag selected images.
geotagging
Import and apply GPX track data to selected images.
export
Export selected images to local files or external services.
neural restore
Apply AI-powered raw denoise, RGB denoise, or upscaling to selected images.

bottom panel



From left to right:

star ratings
Apply star ratings to images.
color labels
Apply color labels to images. Right-click to assign a descriptive name to any color label. The description is displayed only in the tooltip that appears when hovering over the color label (and not in collection filters or any other UI element).
mode selector
Choose a lighttable mode.
zoom
Adjust the number of thumbnails displayed in each row of the file manager view.
enable focus-peaking mode
Highlight the parts of the image that are in focus.



set display profile

Set the display profile of your monitor(s).

2.3. undo/redo

Most changes made within the lighttable are recorded and can be reverted to a previous state. This includes modifications to color labels, ratings, geo-localization, tags, metadata, orientation, copy/paste of history, image duplication, or application of a style. Note that the facility to undo/redo actions is unlimited in the number of steps while in the lighttable view, but it is reset each time you switch to a different view.

Press Ctrl+Z to undo the last modification and Ctrl+Y to redo the last undone modification (if any).

Note: “Undo” cannot reverse changes made to the custom sort order by dragging-and-dropping thumbnails in lighttable filemanager and the filmstrip.

2.4. lighttable modes

2.4.1. filemanager

In the default mode images are displayed in a grid with an adjustable number of images per row.

controls

zoom

The number of images in each row can be altered using the numeric input in the bottom panel, or by holding Ctrl while scrolling over the center view with your mouse.

navigate

You can navigate through the images using the arrow keys (←/→/↑/↓) or by scrolling with your mouse. Press the Home key to scroll to the top of the collection, the End key to scroll to the bottom, and PageUp/PageDown to scroll up/down by a page.

select

You can select the image under the pointer by clicking on its thumbnail or by pressing Enter. To select a range of images, click on the first image and then Shift+click on the last one. To add or remove images in a selection, Ctrl+click on a thumbnail or hover over a thumbnail and press Spacebar.

hover

When the pointer hovers over any image’s thumbnail, it is highlighted and its image implicitly selected for keyboard shortcuts. If that thumbnail is part of a currently-selected set, certain shortcuts will be applied to *all* images of the set (for example, a numeric digit star rating).

drag

You can drag a thumbnail (**along with any selected set** to which it may belong) to set its custom sort position just before or after the image it’s dropped upon.

2.4.2. zoomable lighttable

The *zoomable lighttable* mode provides an alternative way to navigate large collections of images, but with some similarities to the filemanager mode (described below).

controls

zoom

Scroll with the mouse wheel to zoom in and out of the lighttable (compared to Ctrl+scroll in filemanager mode).

Zooming the thumbnails does not change the number of thumbnails per row, so the lighttable can exceed the visible area on all sides.

navigate

Hold down the left mouse button and drag to move the lighttable around and navigate through your collection.

select
As with the filemanager mode, you can select the image under the pointer by clicking on its thumbnail or by pressing Enter. To select a range of images, click on the first image and then Shift+click on the last one. To add or remove images in a selection, Ctrl+click on a thumbnail or hover over a thumbnail and press Spacebar.

hover
Hovering over a thumbnail highlights and impicitly selects it for keyboard shortcuts as in filemanager mode. However, dragging pans the entire thumbnail display, so you cannot change images' custom sort order in this mode.

Hint: you may find that image thumbnails are slow to load when zooming quickly through a large collection. One way to speed up the navigation is to generate a cache containing all the thumbnails using the `darktable-generate-cache` command.

2.4.3. culling

Culling mode allows you to display images side by side for easy comparison.

Thumbnail overlays can be controlled by selecting the star icon in the top panel and changing the "culling overlays" setting. If "no overlays" is selected, then the overlay can be shown on-demand by pressing the "I" key.

controls

zoom
In culling mode, you can zoom into images (up to 100%) by holding Ctrl while scrolling with the mouse wheel.

Pan within zoomed images with click+drag.

By default, zooming and panning are synchronized between all visible images. If you want to zoom or pan only a specific image, add the Shift modifier to the above actions.

navigate
Use the mouse wheel or arrow keys (←/→) to scroll through your collection.

modes

There are two different culling modes, which define how many images are shown at the same time: "fixed" and "dynamic". Switch between these while in culling mode by pressing the "<" key.

fixed mode
The number of images displayed is always the same, independent of the selection length. This number can be set with a slider on the bottom panel.

In this mode, you will navigate through all selected images. If no selection is set (or if only one image is selected), you will navigate through all images.

The default keyboard shortcut to enter culling in fixed mode is "X".

dynamic mode
All of the selected images are shown. If no selection is set (or if only one image is selected) the last value from fixed mode is used.

The default keyboard shortcut to enter culling in dynamic mode is Ctrl+X.

Hint: To enhance performance when loading zoomed images, you can enable (preferences > lighttable > thumbnails > enable disk backend for full preview cache). Bear in mind that this can occupy a lot of disk space.

2.4.4. full preview

From any of the lighttable modes, you can display a fully-zoomed preview of the image that is currently under the mouse pointer by pressing and holding down W. This is useful to more closely inspect an image while rating and selecting images.

Pressing and holding Ctrl+W fully zooms into the image and also identifies any regions of sharpness in the image that may indicate image focus. For this tool to work the input image needs to hold an embedded JPEG thumbnail, which is the case for most raw files.

Regions in the image with a high level of sharpness are indicated with red borders. If no such regions are found, any regions of moderate sharpness are identified with a blue border. Note that this is not the same as the focus peaking indicator, which is another way to identifying areas of sharpness within an image.

Sometimes pressing W or Ctrl+W may not appear to have any effect – in such cases, click on the image thumbnail and press the corresponding key again.

If you want the full preview to stay in place without having to hold the W key, you can enable sticky preview mode by pressing F. In sticky preview mode, you can zoom and pan within the image in a similar way to the culling mode. Press F or ESC to return to the original view.

2.5. digital asset management

2.5.1. collections & film rolls

A collection is a group of images matching a given set of selection criteria.

The most basic kind of collection is a film roll, which contains all of the images that have been imported from a specific folder on disk – whenever you import images from the filesystem, those images are organized in a film roll whose name is derived from that of their parent folder.

You can easily construct other kinds of collection in the collections module based on one or more image attributes (e.g. Exif data, filename, tags). The collection filters module allows you to refine a collection by setting additional selection and exclusion criteria.

A list of the most recently used collections is retained for quick access. These can be accessed from the recent collections module.

2.5.2. image sequence

A collection of images is always presented in a specific *sort by* sequence that determines the order of their display, processing, and export. It is often convenient to be able to control this sequence (e.g. for a slideshow or document).

You may choose to sort by an image attribute (filename, capture time, rating, title, etc.) using the *sort by* drop-down menu at the left-hand-side of darktable's top panel or combine multiple sort criteria by clicking on *new sort* in the left panel at the bottom of the collection filters module.

Whatever *sort by* criterion is chosen, a collection may contain some images you need to present in a different order. In such cases, darktable's custom sort allows you to alter image sequence directly by dragging and dropping thumbnails in lighttable's filemangager mode or the filmstrip panel.

There are two distinct "custom sort" sequencing regimes:

- One *global sequence* over every image in a darktable library.
- Multiple *tag-based* sequences, one for each tag associated with one or more images.

When an image is imported into your darktable library, it is automatically positioned in the global sequence after all previously-imported images. A duplicated image is initially positioned in the global sequence immediately after its original, regardless of intervening image imports.

Tagged images are sequenced within each attached tag in the order that the tag is attached. Selected sets of images are tagged in order of the *sort by* setting in effect at the moment the tag is attached. Newly-tagged images are initially positioned after all images previously associated with that tag. A duplicated image inherits all of its original's tags and is positioned within each as a newly-tagged image.

A collection's selection criteria determines which sequence drag-and-drop alters while working in lighttable filemanager or the filmstrip. Dragging-and-dropping within a tag-based collection alters a designated tag's sequence. Otherwise, drag-and-drop alters the global sequence of all images in the library.

See the custom sort section for information on how to efficiently and safely use drag-and-drop image resequencing.

2.5.3. thumbnails

Each image in the current collection is represented by a thumbnail in the lighttable view and filmstrip module. A cache of the most recently used thumbnails is stored in a file on disk and loaded into memory at startup.

thumbnail creation

A thumbnail is created when an image is imported into darktable for the first time, after an image has been modified in the darkroom, or when revisiting an image whose thumbnail is no longer available.

When an image is imported for the first time darktable can either try to extract an embedded thumbnail from the input image (most raw files contain these, usually in JPEG format) or process the raw image itself using default settings. You can define how darktable obtains its thumbnails in preferences > lighttable > thumbnails .

Extracting an embedded thumbnail from the input image is usually very fast. However, these thumbnails have been generated by the raw converter of the camera and do not represent darktable's "view" of that image. You will notice the difference as soon as you open the image in the darkroom mode, at which point darktable replaces the thumbnail with its own internally processed version.

After import darktable automatically generates thumbnails for new images as they are needed. When importing a large set of new images, thumbnail generation can slow down navigation in the lighttable view. Alternatively you may terminate darktable and generate the thumbnail cache separately by running `darktable-generate-cache` . This program will generate all missing thumbnails in one go.

As the thumbnail cache has a pre-defined maximum size it will eventually get filled up. If new thumbnails are subsequently added, old thumbnails are dropped from the cache. However, darktable will keep all thumbnails on disk if the corresponding disk backend option is activated in preferences > lighttable > thumbnails . Access to the thumbnails in this secondary cache is slower than the primary cache, but still much faster than reprocessing thumbnails from scratch. The size of the secondary cache is limited only by the available disk space.

Thumbnails are never removed from the secondary cache. You can manually clean the secondary cache by recursively deleting all images in the `$CACHE/mipmaps-xyz.d` folder (where `$CACHE` represents the darktable cache directory and `xyz` represents an alphanumeric identifier of the cache). After clearing the secondary cache you can simply allow darktable to re-generate thumbnails as needed, or you can generate all thumbnails in one go with `darktable-generate-cache` .

If you choose not to activate the disk backend and select too small a cache size, darktable may become unresponsive, you may experience continuous regeneration of thumbnails when you navigate your collection or flickering of thumbnail images. A good choice of cache size is 512MB or higher (see memory & performance tuning for more information).

All thumbnails are fully color managed. Colors are rendered accurately on screen as long as your system is properly set up to hand over the right monitor profile to darktable. For more information see the color management section.

skulls, question marks, and warning triangles

If for some reason darktable is unable to generate a thumbnail, it displays one of three placeholder images to indicate the type of error preventing it from doing so.

There are three main reasons this could happen:

- *Missing image file*: If the file could not be found at the location recorded in the database, a skull  will be displayed in place of the image. You are advised to remove images from the database using the actions on selection module before physically removing them from disk. Alternatively you may occasionally run the purge script from darktable's toolset to clean-up your database.
- *Invalid image format*: While darktable will attempt to import all supported file extensions, the extension is not a guarantee that darktable will be able to interpret the file's contents. If the file format (or an option within that format, such as compressed mode) is unsupported, darktable will display a question mark  in place of the image. If the file appears to be corrupted, darktable will display a warning triangle  in place of the image.
- *Low memory*: In the rare event that darktable runs out of memory while generating a thumbnail, it will warn you and display a skull. This can happen if darktable is run with sub-optimal settings, especially on a 32-bit system. See memory & performance tuning for more information.

thumbnail icons

The following icons may be displayed over the top of thumbnails to provide additional information:

- Images that are members of a group are denoted by a light  or dark  group icon, where the darker icon identifies the lead image in that group. See image grouping for details
- A  icon identifies that tags have been added to the image. Hover over the icon to see a list of the applied tags.
- A  icon identifies that the image has been processed. Hover over the icon to see a list of the processing modules that have been applied.

2.5.4. star ratings & color labels

Star ratings and color labels help you to sort and rank images according to your own criteria. An image's star rating and color labels can be displayed over thumbnails in the lighttable view and filmstrip module.

star ratings

You can give an image a rating from zero to five stars. Whenever you import images, each image receives a default rating which you can define in the import module. You can also mark an image as "rejected".

There are several ways to change a rating. While hovering the cursor over an image thumbnail, you can press a number key 0 – 5 to define the number of stars, or press R to "reject" an image. This is probably the fastest way to rate your images on first inspection of a film roll.

You can also directly click on the star icons that are overlaid on the thumbnails or in the bottom panel. Click the x to reject.

You can undo the rejection by clicking x or pressing R again.

Similarly, you can click the first star for a second time to reset the image rating to unranked, or zero stars. This behavior can be changed in preferences > lighttable .

To rate multiple images at once, select those images in the lighttable or filmstrip and then press the appropriate shortcut key, or click the desired star rating in the bottom panel of the lighttable view. To reset the star rating of the selected images, press the shortcut key 0 or click the corresponding icon in the bottom panel.

You can filter images by star rating in the top panel .

color labels

Color labels are another way to classify images, and can be used as an alternative to star ratings or to work alongside them. Each image can carry any combination of one or more color labels (red, yellow, green, blue, or purple).

You can set the color labels for a single image by hovering your cursor over the thumbnail and pressing the function keys F1 – F5, which correspond with the labels in the order given above.

To set the color labels of one or more images, select the desired images in the lighttable or filmstrip and then press the appropriate shortcut key or click the corresponding color button in the bottom panel. A color label will be added to all selected images if **any** of them do not currently have the label; otherwise the label will be removed from all selected images. To remove all labels (of any color) from the selected images, press the gray button.

You can filter images by color label in the collections module.

2.5.5. image grouping

Grouping images helps to improve the structure and clarity of your image collection when displayed in the lighttable view.

You can combine images into a group by selecting them and clicking the “group” button in the actions on selection module, or by pressing Ctrl+G. Likewise, you can remove selected images from a group by clicking the “ungroup” button, or pressing Ctrl+Shift+G.

Duplicated images are automatically grouped together. Similarly, if you import multiple images from the same directory, having the same base name, but different extensions (eg. `IMG_1234.CR2` and `IMG_1234.JPG`), those images automatically form a group.

Images that are members of a group are denoted by a group icon  in their thumbnails. Note that this icon is only shown when “overlays” are displayed on image thumbnails. Thumbnail overlays can be enabled by selecting the star icon in the top panel .

This icon also appears as a button, in the top panel of the lighttable view, that can be used to toggle grouping on and off. If grouping is off, all images are displayed as individual thumbnails. If grouping is on, the images in a group are represented by a single thumbnail image (the group leader). If you press the group icon in the group leader’s thumbnail, that group is expanded (click a second time to collapse). If you then expand another group, the first group collapses.

An expanded group in the filemanager mode of lighttable view is indicated by an orange frame that appears as soon as your mouse pointer hovers over one of the images. This frame surrounds all images in the group.

You can define which image is considered to be the group leader by clicking on the group icon of the desired image while that group is expanded. The group icon is shown only if grouping mode is enabled, so to change the group leader, you need to first enable grouping, expand the appropriate group and finally click the group icon of the desired “group leader” image. The current group leader is shown in a tooltip when you hover over the group icon of an image.

If you collapse an image group and then enter darkroom mode (e.g. by double-clicking on the thumbnail), the group leader image will be opened for developing.

Image groups are also a convenient way to protect an existing history stack against unintentional changes. If you have just finalized an image and want to protect its current version, simply select the image, click “duplicate” in the actions on selection panel, and make sure that grouping is switched on and that the group is collapsed. Now, whenever you open the image group again in the darkroom, only the group leader will be altered. The underlying duplicate will remain unchanged.

Note: “duplicating images” only generates a copy of an image’s history stack, stored in another small XMP file. There is still only one raw file.

2.5.6. metadata and tagging

darktable allows you to store additional information about your images to allow them to be more easily searched and grouped. This information is stored in darktable’s database and XMP sidecar files and can also be included within exported images.

metadata

Metadata (e.g. title, description) is free-format text that usually differs for each image. You can add metadata to images in the metadata editor module.

tagging

Tags are usually shared between multiple images and are used to categorise, group, and/or sequence them. You can add tags to images in the tagging module.

3. Darkroom

3.1. overview

The darkroom view is where you develop your images. There are multiple ways to start developing an image in the darkroom:

1. double-click an image in the lighttable view .
2. use the keyboard shortcut (which by default is a D) from any view.
3. click *darkroom* in the top banner from any view.

In the latter two methods, the image used will use the first selected image in the current lighttable view, or the first image in the current collection if no image is selected.

If the *prioritize the hovered image over the selected images* preference is selected then an image under the cursor in the lighttable, filmstrip, or map will be the one opened in the darkroom, regardless of other selections.

The center panel contains the image currently being edited.

zoom

Middle-click on the center panel cycle between “fit to screen”, 1:1 and 2:1 zoom.

Alternatively you can zoom by scrolling with your mouse or touchpad. When the constrain darkroom zoom between screen fit and 100% preference is enabled, scroll-wheel and pinch-to-zoom is constrained to the range between screen fit (zoom out) and 100% (zoom in). Hold *Ctrl* while zooming to temporarily lift the limit and zoom beyond 100%. When the preference is disabled, zoom is never constrained and the additional *Ctrl* key press is not needed to zoom beyond 100% .

touchpad

If the enable touchpad gestures in darkroom preference is enabled, then touchpad scroll gestures can be used:

- pinch with two fingers to zoom
- scroll with two fingers to pan

3.2. darkroom view layout

left panel

From top to bottom:

navigation
Navigate and zoom the center view.

snapshots
Take and view snapshots for comparison with the current edit.

duplicate manager
View and manage duplicates.

global color picker
Select and display color information taken from parts of the image.

tagging
Manage tags.

image information
Display information about the current image.

mask manager
View and edit drawn shapes.
export
Export selected images to local files or external services.
neural restore
Apply AI-powered raw denoise, RGB denoise, or upscaling to selected images.

right panel

From top to bottom:

scopes
A graphical depiction of the image's light levels and colors. This module can be moved to the left-hand panel if desired (see preferences > miscellaneous > position of the scopes module).
module groups
Select module groups (if enabled).
search module
Search for a module (if enabled).
processing modules
The modules used to process an image.
module order
Choose the order in which processing modules are executed in the pixelpipe.

bottom panel



From left to right:

presets
Quick access menu for module presets. You can manage the contents of this menu by selecting "manage quick presets list...".
styles
Quick access menu for styles. Hover over a style name with your mouse to show a preview of the current darkroom image with the selected style applied.
second darkroom window
Allows the user to display a second image in a separate window. By default, the second window shows the same contents as the main view. When an image is "pinned" in the second window, it will remain unchanged when making module changes, navigating the history stack or opening a different image in the main view. This is especially useful on large screens and multi-monitor setups.
focus peaking
Toggle focus peaking mode.
color assessment
Toggle the ISO12646 color assessment view.
high quality processing
Toggle high quality processing mode.
raw overexposed warning
Toggle raw overexposed indicators (right-click for options).
clipping warning
Toggle clipping warnings (right-click for options).

soft proof

Toggle softproofing overlays (right-click for options).

gamut check

Toggle gamut checking (right-click for options).

guides & overlays

Left-click to switch global guide overlays on/off and right-click to change the guide settings, including the color of all on-image drawing (masks, crop guides etc.)

filmstrip



The filmstrip module at the bottom of the screen allows you to select and interact with the currently selected collection.

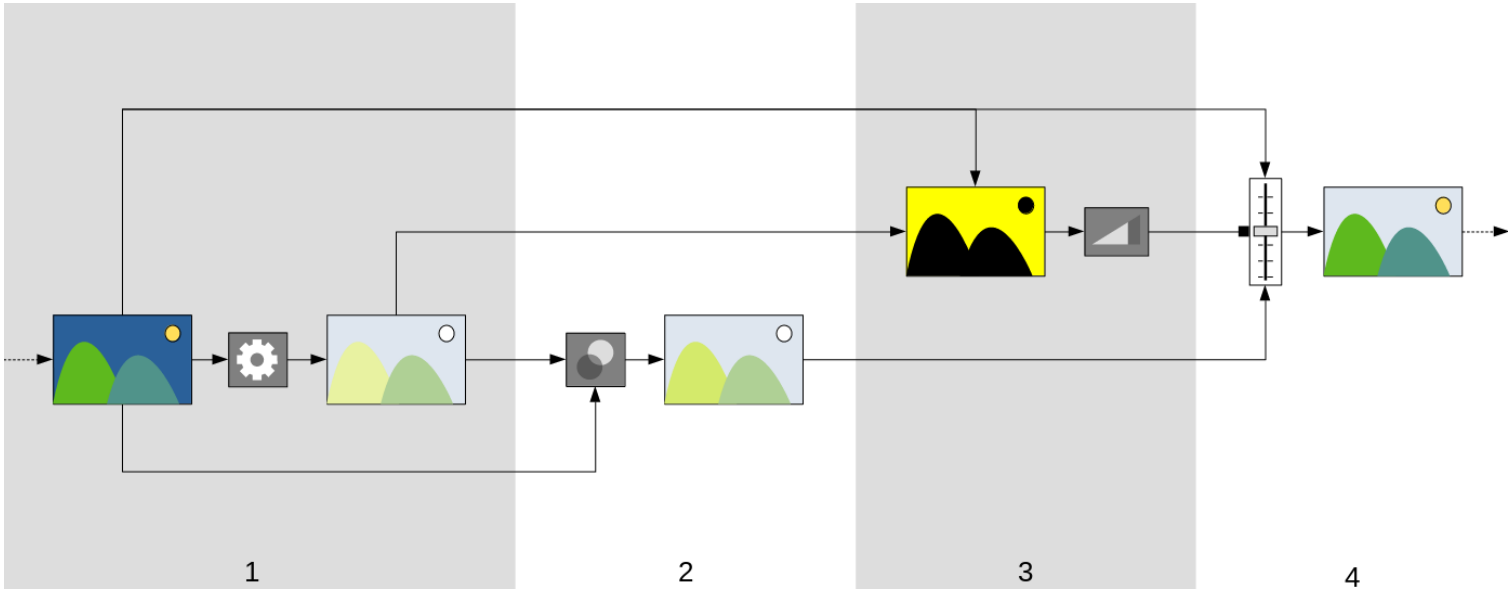
3.3. the pixelpipe

3.3.1. the anatomy of a processing module

The basic element of image processing in darktable is the processing module . In order to process a raw image a number of such modules act on the input image in sequence, each performing a different *operation* on the image data. For those familiar with Adobe Photoshop, the concept of a *processing module* in darktable is analogous to that of an *adjustment layer* in that both make an incremental adjustment to the image, building on top of the adjustments that came before.

Utility modules are also provided by darktable, however these are not directly involved in image processing, instead providing a GUI that allows you to manage your images, tag them, export them etc.

Every processing module acts independently of the others, but all modules perform their processing in a similar manner:



1. Receive the *module input* from the last executed module and perform an *operation* on it to produce the *processed output*. This *operation* is different for every processing module .

2. Combine the *module input* and *processed output* using a blending operator to produce the *blended output*. If no blending is performed, the output of this step is the same as the *processed output*.

3. Generate a *mask*, which defines an *opacity* for each pixel in the image. The *opacity* is later used to control how strongly the module's operation is applied to each part of the image.

   You may define your own mask by drawing shapes over the image or by using pixel properties from the *module input* or *processed output* (see masks for details). This mask may be further modified with a global opacity setting, which affects every pixel equally.

   If no drawn/parametric mask is used, the output of this step is a mask where every pixel has the same opacity (governed by the global opacity setting). If no opacity is defined (no blending is performed) a global opacity of 1.0 (or 100%) is assumed.

4. Combine the *module input* and *blended output* pixel-by-pixel using the *mask* as a mixing operator, to produce the *final output*. Where the mask opacity is 100%, the *final output* is the *blended output* for that pixel. Where the mask opacity is 0 the final output is the *module input* for that pixel. An intermediate opacity combines the *blended output* and *module input* proportionally. The *final output* is passed to the next module for further processing.

Steps 2 and 3 are optional and not supported by all modules. For example, the demosaic module must be applied to the entire raw file in order to produce a legible image so it does not make sense to mask or blend its output.

Each of the above steps is defined in more detail in subsequent sections.

3.3.2. the pixelpipe & module order

The ordered sequence of processing modules operating on an input file to generate an output image is known as the "pixelpipe".

The order of the pixelpipe is represented graphically by the order in which modules are presented in the user interface – the pixelpipe starts with a RAW image at the bottom of the module list, and applies the processing modules one by one, piling up layer upon layer of processing from the bottom up, until it reaches the top of the list, where it outputs the fully processed image.

Note: The order in which processing modules are executed exactly matches the order in which the modules appear in darktable's user interface. **Changing the order of the modules in the user interface changes how your image is processed.**

---

types of pixelpipe

Processing images can be very resource-intensive. For this reason, darktable includes various types of pixelpipe, optimised for different parts of the system. For example:

- The export pixelpipe processes the full-sized image at full quality. This is the slowest type of pixelpipe, but provides the maximum available image quality.
- The thumbnail pixelpipe is optimised for speed, since it needs to process multiple small images at the same time without impacting lighttable or filmstrip performance. These optimisations impact image quality, but this is not usually a problem for the small images used in thumbnail generation.
- The standard darkroom pixelpipe attempts to produce as accurate an image as possible, while also maintaining responsiveness when modifying module parameters. It does this by only processing the pixels that are visible on-screen – known as the “region of interest” (ROI). However, this can mean that the image doesn’t accurately reflect the exported file, especially when using modules that rely on the properties of neighboring pixels. This is particularly noticeable in modules that impact local contrast (e.g. *diffuse or sharpen*, *denoise (profiled)* and *local contrast*) and can mean that the darkroom view appears over-sharpened compared to a full-sized export.
- A cut-down darkroom pixelpipe is used while interacting with some darkroom modules that display the full image with overlays (like *retouch*, *crop* and *liquify*). This pixelpipe excludes some long-running modules (like *diffuse or sharpen*) in order to improve responsiveness, but can temporarily make the image look under-processed (blurred). This limitation is removed as soon as you move focus to a different module.
- In order to overcome the above limitations within the standard darkroom pixelpipe, you can enable high quality processing mode in the darkroom view. This mode processes the entire image using the export pixelpipe and only downscales for display at the end of the pipe. This means that the darkroom image will very closely match the exported image, but will significantly degrade responsiveness when interacting with processing modules. You are advised to only use this mode after you have already done most of your processing. Its performance will be significantly improved by using an OpenCL-capable GPU.

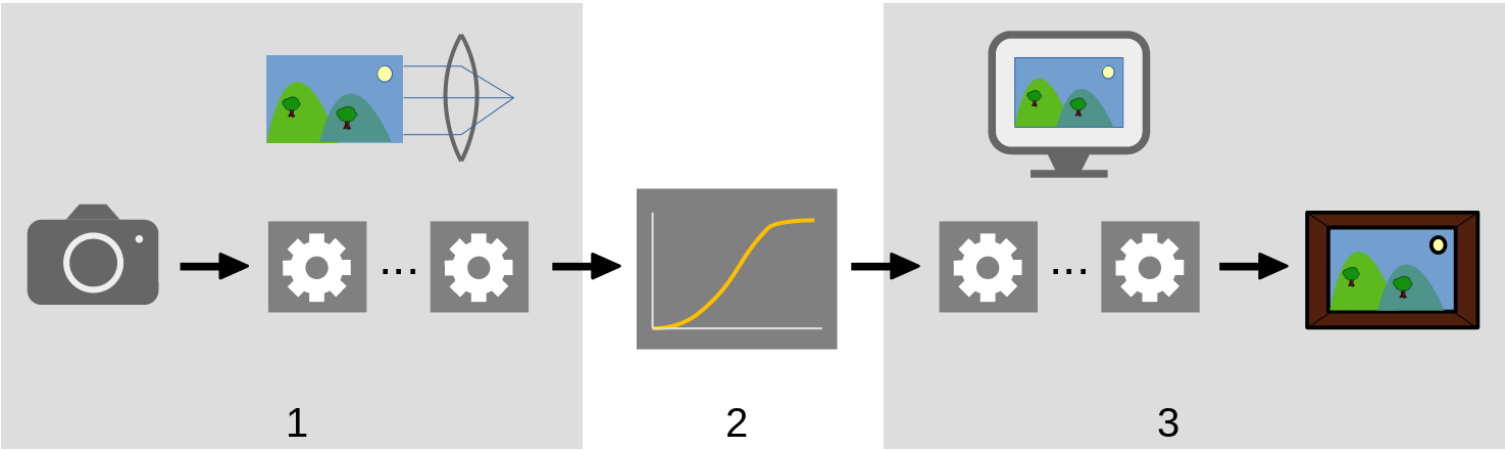
module order and workflows

The order in which modules are executed within the pixelpipe has been carefully chosen to give the best output quality. In previous versions of darktable it was not possible to change the module order. However, there are a number of very specific use cases where the movement of some modules within the pixelpipe is advised.

One of the main reasons to change the module order came about with darktable version 3.0, which introduced the new *scene-referred* way of working. Version 3.2 formalised this by introducing the *display-referred* and *scene-referred* workflows, which are controlled by the preferences > processing > auto-apply pixel workflow defaults setting. Starting with version 3.6, *scene-referred* workflow is now the official recommended (and default) way to use darktable.

The *scene-referred* workflow attempts to perform as many operations as possible in a linear RGB color space, only compressing the tones to fit the output medium (with a non-linear tone mapping) at the end of the pixelpipe. This has the advantage of being a more physically-realistic space to do transformations than the traditional *display-referred* workflow, which attempts to perform operations in a non-linear *perceptual* color space. Honoring the physical realism (rather than the perceptual realism) makes it much easier to produce predictable processing algorithms with a minimum of artifacts.

The following diagram should help you to understand the difference between these workflows:



1. *Scene-referred* modules process linear data that is proportional to the amount of light collected by the camera at the scene. The dynamic range of an image in the scene-referred section of the pixelpipe is often larger than that of the display medium.

2. At some point in the pixelpipe, these pixel values are compressed by a non-linear tone mapping operation into a smaller dynamic range more suitable for display on a monitor or a print.

3. The remaining modules operate in the non-linear *display-referred* section of the pixelpipe to produce the final output image.

display-referred workflow

Prior to version 3.0 darktable's workflow was *display-referred* (auto-apply pixel workflow defaults = "display-referred") and this option is still provided as a legacy mode. In this workflow, the *base curve* or *filmic rgb* module performs tone mapping early in the pixelpipe and most other darktable modules operate on image data in the compressed *display-referred* space.

Selecting the display-referred workflow enables the legacy (pre-darktable-3.0) module order and automatically switches on the *base curve* module for new images.

Pixel data within the *display-referred* space is non-linear and is not a physically realistic representation of the original scene. This can lead to various artifacts with some modules, hence the creation of the (now default) *scene-referred* workflow.

scene-referred workflow

Scene-referred workflow (auto-apply pixel workflow defaults = "scene-referred") was introduced as part of darktable 3.0. The module order was entirely rearranged to place the *filmic rgb* and *base curve* tone mapping modules much later in the pixelpipe. This means that most modules now operate in *linear rgb* space with only a few modules remaining within the non-linear *display-referred* space. Within this workflow it is now recommended that the majority of image processing takes place using the modules up to and including *filmic rgb* . Operations in this section of the pixelpipe, being truly linear, are much more physically realistic and produce fewer artifacts.

Selecting the scene-referred workflow enables the *v5.0* module order and automatically enables the *exposure* and *filmic rgb* modules with some presets designed to act as a reasonable starting point for scene-referred editing.

changing module order

It remains highly recommended that users not change the order within the pixelpipe for a number of reasons:

- The sequence of modules has been selected with great care in order to give highest output quality. Changes to the sequence often worsen the result rather than improving it.
- Some processing modules simply don't make sense if they are shifted in the pixelpipe. For example, *highlight reconstruction* needs to be performed on raw data before *demosaic* , which itself needs to be performed before any *input color profile* can be applied. For this reason it is still not possible to move some of the modules that are placed early in the pixelpipe.
- Most processing modules are designed to work within a specific color space (see the color management section for more details). Full flexibility would require modules to support different parallel algorithms depending on the color space they are working in, which would drastically increase complexity.

Despite the general recommendation to leave the pixelpipe order alone, it is possible to move modules within the pixelpipe by holding Ctrl+Shift and dragging and dropping the desired module to a new location. This should only be done by experienced users who understand the impact this will have on the image.

The module order can be manually changed back to either the *v5.0*, *v3.0* or *legacy* versions using the module order module, which can also be used to define your own custom module order presets.

3.3.3. the history stack

The *history stack* stores the entire editing history for a given image, in the order in which those edits were applied. It is saved to darktable's library database and the image's XMP sidecar file and persists between editing sessions.

Each time a processing module is enabled, disabled, moved or amended a new entry is added to the top of the *history stack*.

The history stack can be queried and modified within the history stack module in the darkroom.

Note: The history stack is not a representation of the order in which the modules are **executed** but the order in which they were **amended**. The execution order is represented by the order of the modules in the right-hand panel.

3.3.4. undo and redo

While you are editing your image, darktable records all of the modifications you make to that image. This means that it is possible to undo and redo changes to recover a previous editing state. Note that the undo/redo facility is unlimited in the number of steps while editing an image, but is reset each time the darkroom is switched to a new image.

Press Ctrl+Z to undo the last modification and Ctrl+Y to redo the last undone modification (if any).

3.4. processing modules

3.4.1. module header

At the top of each processing module is the *module header*.



Click on the module name to expand the module and display the parameters that control its operation.

By default darktable will only allow one processing module to be expanded at a time – if you click the header of another module, the previously-opened module's controls are collapsed. If you want to expand more than one module, you may expand further modules by Shift+clicking on the header and all previously expanded modules will remain open. This behavior can be reversed via a setting in preferences > darkroom .

Note: Expanding a module does not cause it to be activated. See below for how to activate modules.

The module header contains the following controls in order from left to right:

on/off button

Click to toggle the module on or off. Some modules are essential for image processing and cannot be disabled (though their parameters may be amended). Similarly, some modules are not applicable for certain types of image and cannot be enabled.

Ctrl+click on the on/off button to toggle whether the module has focus. The focus state is usually used to activate any overlays that a module places over the image to control its functionality. For example, the *crop* module only shows the composition and crop guide lines on the image if it has focus. Modules are automatically given focus when expanded.

module name

The module name consists of a description of the module's operation (which cannot be changed) followed by the module's instance name (which can). By default the first instance of a module has an empty instance name. If you create additional instances, the name of each new instance is initiated with a unique integer. For example, the second created instance of an exposure module will be automatically named `exposure 1`.

Ctrl+click on a module's name to manually amend its instance name.

mask toggle
This icon will appear in the header whenever a mask is active on a module. Hover over the icon to see what type of mask is enabled. Click it to display the current mask as a yellow overlay over a black-and-white version of the image. Solid yellow indicates an opacity of 100%; a fully visible gray background image (without yellow overlay) indicates an opacity of 0%. This toggle button can be disabled in preferences > darkroom > show mask indicator in module headers .

multiple instance menu
This menu allows you to create, delete, move and rename module instances. Right-click on this icon to directly create a new instance of the module. See the multiple instances section for more information.

reset
Click to reset all module controls to their default values. Ctrl+click to reapply any automatic presets for the module – if no automatic presets are applicable for this module, Ctrl+click will simply reset to default values (same as click).

presets menu
This menu allows you to apply, create and edit module presets. See the presets section for more information.

The visibility of the four icons to the right of the module name can be controlled in preferences > darkroom > show right-side buttons in processing module headers .

3.4.2. multiple instances

Many of darktable's modules can be applied more than once in the pixelpipe. Each instance of a module behaves independently, taking its input from the module below it in the pixelpipe and delivering its output to the next module.

As with the base instance of a module, all instances can be moved independently within the pixelpipe either by holding Ctrl+Shift while dragging & dropping or by choosing "move up" or "move down" in the *multiple instances* drop-down menu.

Instances can be renamed by Ctrl+clicking on the module name.

typical use cases

There are many occasions where it makes sense to have a module apply more than once in the pixelpipe. Here are some typical use cases.

- The *exposure* module can be used in combination with masks to lighten or darken parts of an image. A separate instance may be created to modify each masked region of the image.
- You may wish to handle luma and chroma noise independently. This can be accomplished by generating two instances of your chosen denoising module and using the first one only on luma (by selecting blend mode "lightness") and the second one only on chroma (by selecting blend mode "color").

Note: Each instance also adds to the workload of your pixelpipe. Generating too many instances – especially of the more demanding modules – will cause noticeable slow-down.

---

managing multiple instances

Click on the *multiple instance menu* in the module header to display a menu with the following options. Right-click on the menu icon to create a new instance directly (same action as clicking on the "new instance" option of the menu).

new instance
Create a new instance of the current module with all of its parameters reset to their default values. The 'instance name' is automatically set to a unique integer so that it can be distinguished from its parent.

duplicate instance
Create a new instance of the current module with all of its parameters inherited from the current instance. As with 'new instance' the 'instance name' is automatically set to a unique integer.

move up/down
Move the instance up or down in the pixelpipe

delete
Remove the current instance. This option is not available if only one instance is present.

rename
Rename the current instance. See the history stack section for more details on how the instance name impacts copying and pasting history stacks.

3.4.3. presets

Presets allow you to store commonly-used module settings for future use. Some modules already come with pre-defined (internal) presets and you may also define your own (user-defined). Both internal and user-defined presets can be shown by clicking the *presets menu* in the module header .

Most of the functionality described here applies to processing modules only. However, presets can also be used with some utility modules. When used with utility modules, the functionality to auto-apply or auto-show presets based on image Exif data is not available.

Please note that, for processing modules, the saved preset also includes the active state of the module. You can use this to create your own default settings, which you can activate on-demand. Simply set your desired defaults, disable the module, and save the preset.

In addition to the individual modules preset menu presets can be managed at preferences > presets .

the presets menu

The presets menu will contain one or more of the following entries depending on the presets that are defined or selected for the current module:

preset list
: A list of the presets available for the current module. The currently selected preset (if any) is shown in **bold** and with a small check mark beside it. In addition, if the selected preset displays on a submenu, it parent entry is also highlighted in **bold**.

edit this preset
: If a preset has been selected, edit the selected preset (see below).

delete this preset
: If a preset has been selected, delete the selected preset.

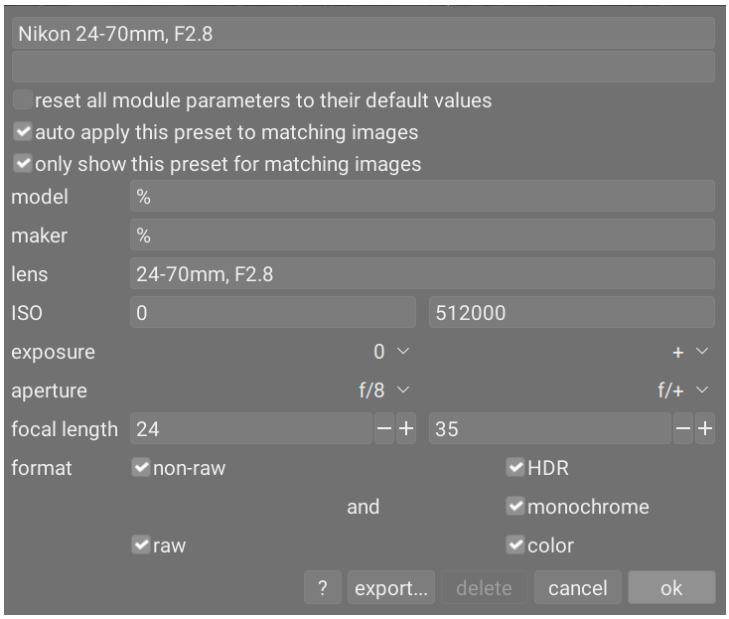
update preset [name]
: Update the named preset to match the module's current parameters.

store new preset
: Create a new preset using the module's current parameters.

Click on a preset name to apply the preset to the current instance of the module. Long-left-click on a preset name to apply the preset but keep the preset list open (so that you can experiment with multiple presets without needing to re-open the menu). Right-click on a preset name to create a new instance of the module and apply the selected preset to it. You can also apply a preset at any time while you are in the darkroom using a keyboard shortcut – if you have assigned one (see preferences > shortcuts).

creating and editing presets

When creating or editing presets, the following dialog is shown:



controls

name
The name of the preset. You can create a hierarchy of presets with shared categories by inserting vertical bar characters ("|") in the name. For example, “Nikon|24-70mm, F2.8” would be displayed as the category “Nikon” with a submenu containing “24-70mm, F2.8” as well as any other presets whose names start with “Nikon|”.

description
A searchable description for the preset (optional)

reset all module parameters to their default values
Selecting this option will cause all module parameters to be reset to their default values (as if you had simply enabled the module on a new image or clicked the module reset button). This can be used to automatically set certain modules (color calibration, exposure, filmic) based on the Exif properties of the current image rather than setting to hard-coded parameters. Note that this option is not available when editing presets from the preferences > presets tab.

auto apply this preset to matching images *(processing modules only)*
Check this box to automatically apply this preset to matching images when they are opened in the darkroom for the first time (you can reapply such automatic presets by Ctrl+clicking on the *reset* button in the module header). Additional controls will appear to allow you to define which images the preset will be applied to based on image Exif data (see below).

For example, if you want a preset to be applied to all images from a specific camera leave all fields at default values except for “model”. Leave all fields unchanged to auto-apply a preset to all images.

The example dialog above sets up following rules: if the lens name matches, the aperture is greater than or equal to f/8 and the focal length is between 24 and 35mm the preset will be automatically applied.

The image information module displays the camera model and lens name for each image. Use this to ensure you have the correct spelling.

If there is more than one preset with matching parameters for a given image, darktable creates a separate module instance for each matching preset, in the order they were defined – the last preset that was created will be placed after all other instances of the module in the pipeline.

only show this preset for matching images *(processing modules only)*
Check this box to automatically show the preset in the preset menu, using the same set of filters.

filter criteria

The following criteria can be used to auto-apply or auto-show presets for processing modules:

model
A pattern to be matched against the Exif field that describes your camera model; use % as wildcard.
maker
A pattern to be matched against the Exif field that describes the maker of your camera; use % as wildcard.
lens
A pattern to be matched against the Exif field that describes your lens; use % as wildcard.
ISO
Only apply the preset if the ISO value of your image lies within the given range (shows ∞ if the upper range is unlimited).
exposure
Only apply the preset if the exposure time of your image lies within the given range; set + as the upper value to match arbitrarily long exposures.
aperture
Only apply the preset if the aperture of your image lies within the given range; set f/0 as the lower value to match arbitrarily open apertures; set f/+ as the upper value to match arbitrarily closed apertures.
focal length
Only apply the preset if the focal length of your image lies within the given range (from 0 to 1000).
format
Only apply the preset to certain types of image. Check boxes to include files matching these criteria; uncheck boxes to exclude those files. Choose from “raw”, “non-raw”, “HDR”, “monochrome” and “color”.

managing presets

Both user-defined and internal presets can be viewed and managed from within preferences > presets .

Note: If you create a user-defined preset with the same name as a built-in preset, your preset will override the built-in version, which will no longer be accessible.

If you delete a preset that has the same name as one of the built-in presets, then your user preset will be deleted, and that preset name will no longer appear in the preset menu at all. The next time you start darktable, the corresponding built-in preset will once again become visible.

---

module naming in the darkroom view

By default, if the current parameters of a processing module match those of a saved preset, darktable will attempt to automatically set the name (label) of the module in question, as follows:

- If the user has manually amended the name of the module in the current image, the module name will be left unchanged,
- If the module instance from which the preset was created had a manually-set name, any subsequent module that matches this preset will automatically be given the same name. Note that if this was unintentional, the only way to revert will be to drop and recreate the preset, since the automatic name is a hidden field in the database,
- If the module instance from which the preset was created did not have a manually-set name, the name of any subsequent module that matches this preset will be set to the same as the preset name.

As soon as the module parameters are changed such that it no longer matches a preset, the module’s name will be reset.

This functionality can be disabled in preferences > darkroom > automatically update module name .

3.4.4. module controls

This section describes how to interact with processing module controls.

sliders

Sliders offer five different methods of interaction, depending on the level of control you require.

left-click (set)

Click anywhere in the slider area to set the value. You can also click and drag to change it. You don't have to aim for the triangle or even the line – you can click anywhere in the entire height of the slider, including the label.

mouse wheel (adjust)

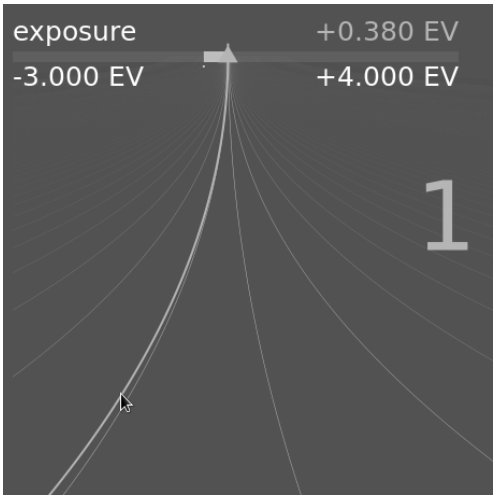
Hover over the slider with your mouse, then use your mouse wheel to adjust the value. You can alter the default speed at which the mouse scroll adjusts a slider by scrolling over that slider while in visual shortcut mapping mode .

keyboard arrow keys (adjust)

When the slider has focus you can hover over the slider with your mouse, then use your keyboard's arrow keys (←/↓ and →/↑) to adjust the value. In order to give focus to the widget without changing the current value you can right-click, then right-click again.

right-click (pop-up edit)

When your mouse is over a slider right-clicking enables a multi-functional pop-up below the slider for fine control with your mouse or numerical entry using the keyboard.



A bent line extending from the triangular marker moves with your mouse. The closer your mouse pointer is to the triangular marker the coarser the control you have over the value; the further away from the triangular marker the finer your control. Left-click with your mouse to accept the new value and hide the pop-up.

Alternatively you can type in a new value using your keyboard and commit by hitting the Enter key. You may even supply the new value in the form of an arithmetic expression which darktable will calculate for you – the previous value is referenced as "x".

For most sliders, the minimum and maximum values displayed are known as "soft limits" – they do not represent the minimum/maximum values that you may enter, merely a suggested range of "normal" values that most users will not need to exceed. As well as these soft limits, each slider also has "hard limits" which may not be exceeded.

You may right-click and type any value up to the hard limits for a given slider. For example, in the *rotate and perspective* module, the soft limits for angle are -10 to +10 degrees while the hard limits are -180 to +180 degrees; In the *exposure* module, the soft limits for the exposure slider are -3 to +4 EV while the hard limits are -18 to +18 EV. If you try to enter a value beyond the hard limit, it will be automatically be adjusted to the minimum/maximum allowable value.

double-click (reset)

Double-click on a slider or its label to reset back to the default value. Ctrl+double-click to reset its value back to any auto-applied preset (if one applies for the current image). If controls are grouped together within a tabbed interface, you can double-click on the tab's header label to reset all settings in that tab.

In addition, the speed of mouse-wheel, arrow-key and click+drag adjustments can be altered:

- hold down the Shift key while adjusting to *increase* the step size by a factor of 10.
- hold down the Ctrl key while adjusting to *decrease* the step size by a factor of 10.

Both of these multipliers can be amended in preferences > shortcuts by altering the speed of the fallbacks/value actions.

comboboxes

Click on a combobox to show a list of available options which you can click to select. Occasionally the selection list will open close to the bottom or top of the screen meaning that only some of the items are visible – simply scroll with your mouse wheel to bring up the full list. Alternatively, you can also use the mouse wheel and keyboard arrow keys to select an option, or start typing to filter the combobox entries.

As with sliders, you can double-click the combobox or its label to reset back to the default value, or Ctrl+double-click to reset back to any auto-applied preset.

pickers

Many modules allow parameters to be set using pickers (identified by the  icon). Pickers allow you to select sample areas of the image from which to calculate values for module parameters. You can usually choose to select either a “point” (a single pixel) or an “area” (a rectangular selection of pixels) from the image, although some modules only allow one mode to be used.

Selecting pixel(s) will cause the appropriate module value(s) to be updated, and may also cause additional visual feedback (for example, overlaying the range of selected pixels on a tone curve).

Activate point mode by left-clicking the picker icon and then left-click anywhere on the image to select the pixel from which to calculate values. The selected point will be shown on-screen (with a cross icon) which you can adjust by left-clicking another pixel. Deactivate the picker by left-clicking the picker icon again.

Activate area mode by either right-clicking or Ctrl+clicking the picker icon and then left-click and drag to select a rectangular area of the image from which to calculate values. The selected area will be shown on-screen as an overlaid rectangle which can be moved by either drawing another rectangle elsewhere or by left-clicking and dragging the corners of the drawn rectangle. Reset the rectangle (to select the whole image) by right-clicking anywhere on the image. Deactivate the picker by right-clicking (or Ctrl+clicking) the picker icon again.

In modules where only one mode is available, left-clicking the icon will usually toggle that mode on and off.

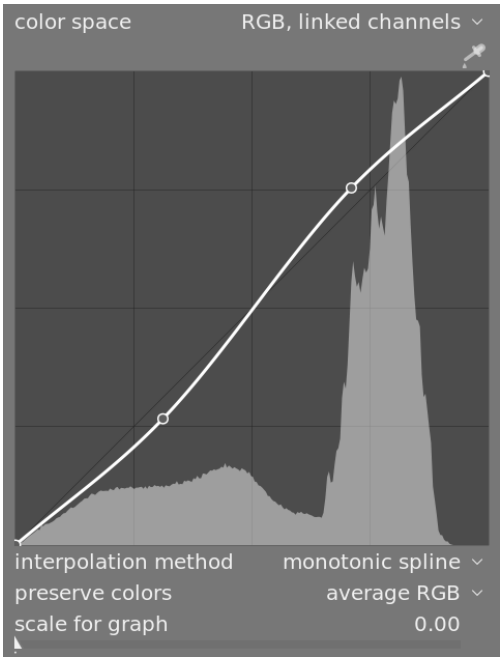
In modules where both modes are available, Ctrl+clicking anywhere on the image will switch modes.

keyboard shortcuts

Module parameters can also be adjusted using keyboard shortcuts. See preferences > shortcuts for more information.

3.4.5. curves

The base curve , tone curve and rgb curve modules use curves to control the tones in the image. These modules have some common features that warrant separate discussion.



nodes

In their default state, curves are straight lines, defined by two anchor nodes at the top-right and bottom-left of the graph. You can move the nodes to modify the curve or generate new nodes by clicking on the curve. Ctrl+click to generate a new node at the x-location of the mouse pointer and the corresponding y-location of the current curve – this adds a node without the risk of accidentally modifying the curve. Up to 20 nodes can be defined per curve. To remove a node, click on it and drag it out of the widget area.

curve controls

The following controls are common to two or more of the above processing modules and are therefore discussed separately here. Please see the individual module documentation for discussion of any additional controls.

interpolation method

tone curve and rgb curve only

Interpolation is the process by which a continuous curve is derived from a few nodes. As this process is never perfect, several methods are offered that can alleviate some of the issues you may encounter.

- *cubic spline* is, arguably, the most visually pleasing. Since it gives smooth curves, the contrast in the image is better enhanced. However, this method is very sensitive to the nodes' position, and can produce cusps and oscillations when the nodes are too close to each other, or when there are too many of them. This method works best when there are only 4 to 5 nodes, evenly spaced.
- *centripetal spline* is designed specifically to avoid cusps and oscillations but, as a drawback, it will follow the nodes more loosely. It is very robust, no matter the number of nodes and their spacing, but will produce a more faded and dull contrast.
- *monotonic spline* is designed specifically to give a monotonic interpolation, meaning that there will be none of the oscillations the cubic spline may produce. This method is most suitable when you are trying to build an analytical function from a node interpolation (for example: exponential, logarithm, power, etc.). Such functions are provided as presets. It is a good trade-off between the two aforementioned methods.

preserve colors

If a non-linear tone curve is applied to each of the RGB channels individually, then the amount of tone adjustment applied to each color channel may be different, and this can cause hue shifts. The *preserve colors* combobox provides different methods of calculating the “luminance level” of a pixel in order to minimise these shifts. The amount of tone adjustment is calculated based on this luminance value, and then this same adjustment is applied to all three of the RGB channels. Different luminance estimators can affect the contrast in different parts of the image, depending on the characteristics of that image. The user can therefore choose the estimator that provides the best results for the given image. Some of these methods are discussed in detail in the **preserve chrominance** control in the *filmic rgb* module. The following options are available:

- *none*
- *luminance*
- *max RGB*
- *average RGB*
- *sum RGB*
- *norm RGB*
- *basic power*

scale for graph

tone curve and base curve only

The scale allows you to distort the graph display so that certain graphical properties emerge to help you draw more useful curves. Note that the scaling option only affects the curve display, not the actual parameters stored by the module.

By default, a “linear” scale is used (scale factor 0), which uses evenly spaced horizontal and vertical axes. Positive values give the graph a logarithmic scale, compressing high values and dilating low values on both the horizontal and vertical axes, so that nodes in lowlights get more space on the graph and can be controlled more precisely.

Increasing the ‘scale for graph’ slider sets the base of the logarithm used to scale the axes. This allows you to control the amount of compression/dilation operated by the scale. If you draw purely exponential or logarithmic functions from identity lines, setting this value defines the base of such functions.

3.4.6. wavelets

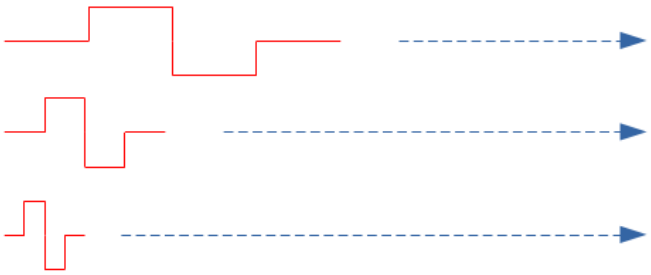
Wavelets are used to separate (or *decompose*) an image into a number of distinct *layers*, each containing a different level of detail. After decomposing an image in this way, a module can limit its processing to one or more of these detail layers, and then piece the layers back together again at the end to form its output. This allows us to be surgical about which features in the image we wish to impact when working with a module.

Some of the operations that darktable can perform in this way are:

- noise removal (in the *denoise (profiled)* , *raw denoise* and *contrast equalizer* modules)
- contrast adjustment (in the *contrast equalizer* module)
- blurring or removal of unwanted detail (in the *retouch* module)

theory

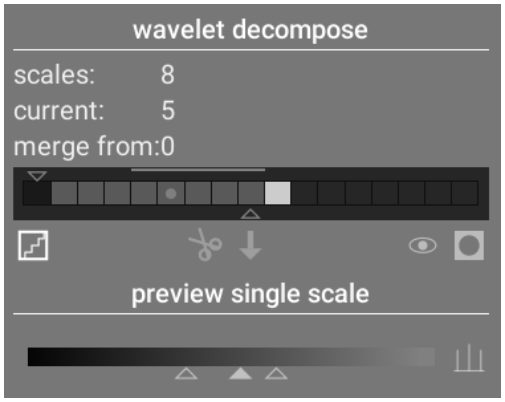
A wavelet is an oscillating mathematical function that starts and ends at zero. The following diagram shows some simple wavelets of differing sizes.





These wavelet functions are used to scan and decompose an image using a mathematical operation called *convolution*. This picks out details from the image that are on a similar scale to the size of a given wavelet, and builds a number of these detail layers, each corresponding to a different wavelet scale.

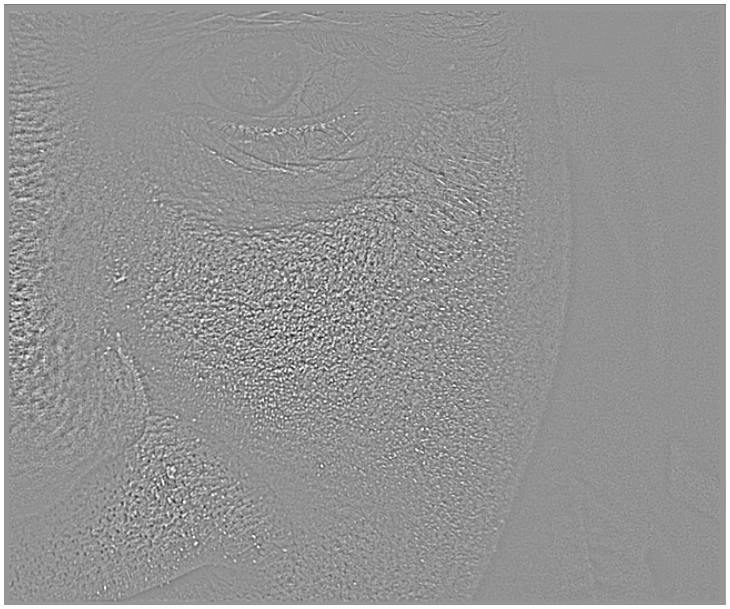
The following is an example where detail layers have been extracted from the above image. In this case, the images were produced using the *retouch* module, splitting the image into 8 different layers, and using the module's "display wavelet scale" button to visualise each of the detail layers:



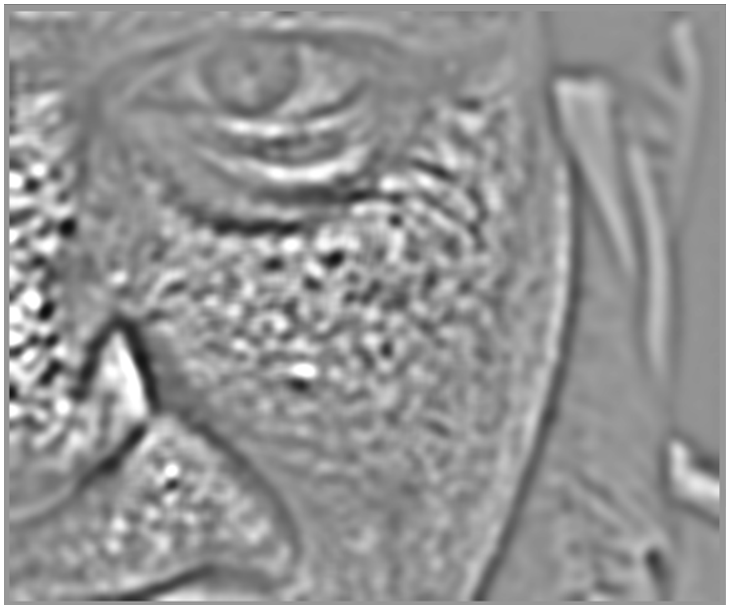
The bars in the *wavelet decompose* section indicate the layers that have been extracted at different wavelet scales. The darkest rectangle at the left represents the entire image (before decomposition) and the gray boxes each represent one of the decomposed layers. Clicking on the staircase icon below the bar graph enables the layer visualization overlay so that you can see what the currently selected layer looks like.

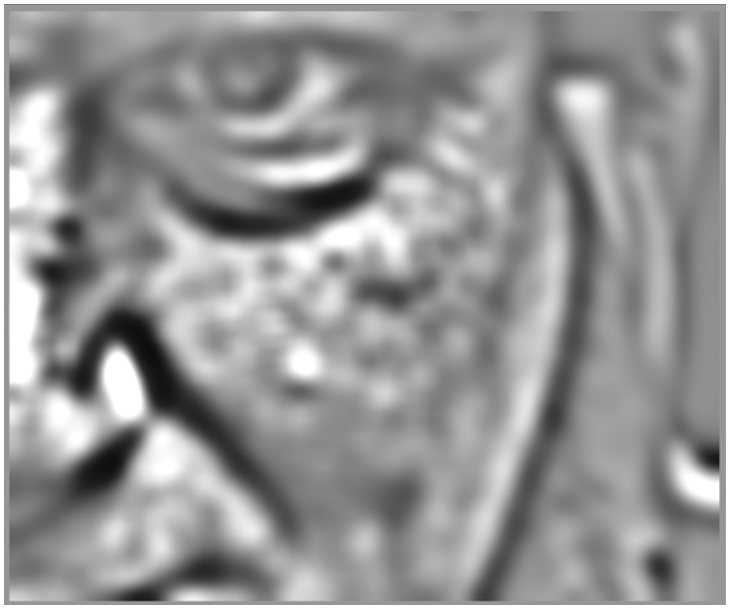
Let's take a look at some of the layers generated for the above image.

At scale #2, the image contains only very fine details, including the model's eyebrows, eye lashes and the pores of his skin. It doesn't include the coarser details of the image, because those details have been extracted to other layers:

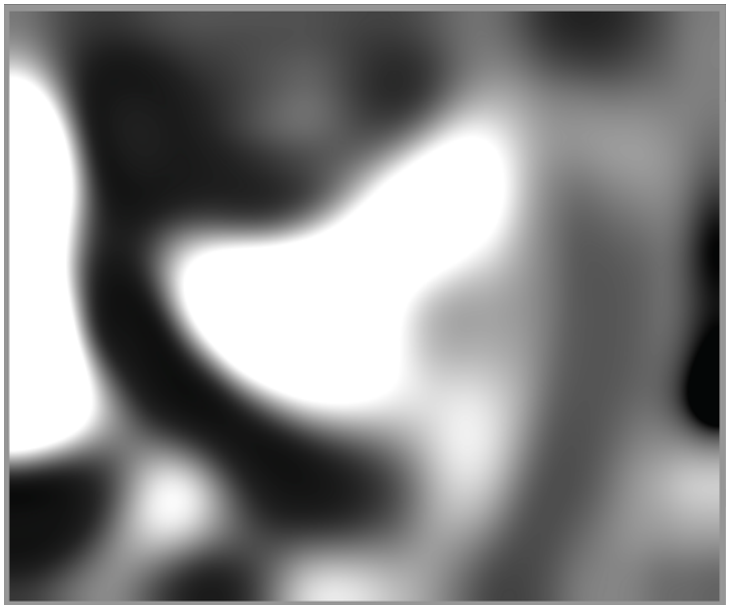


At scales #5 and #6 we begin to see larger and larger features:





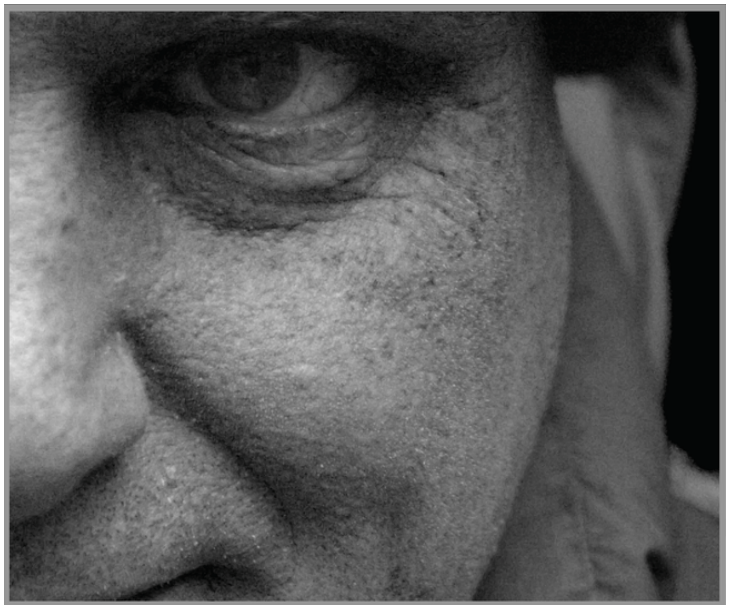
By scale #8 we only see very high-level features such as the overall shape of the model’s nose, eye and cheek:

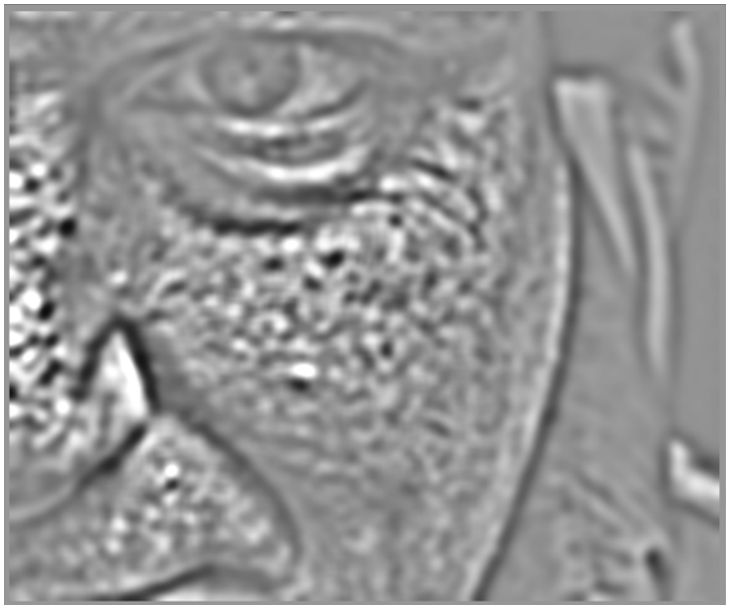


why use wavelets?

Suppose, in the above example, that we wanted to smooth out some of the blotchiness in the model’s skin, without losing any of the underlying small-scale texture. Wavelet decomposition makes this a trivial operation - we can simply use the retouch module to apply a Gaussian blur to only the ‘blotchy’ detail layer(s), leaving all other layers untouched. Once the adjustment is complete, the retouch module simply recombines the adjusted layer with the remaining untouched layers to produce the final image.

The sequence of images below shows (1) The original image; (2) The layer (scale 5) that we wish to blur; and (3) The final image after the scale 5 layer has been blurred and the layers recombined:







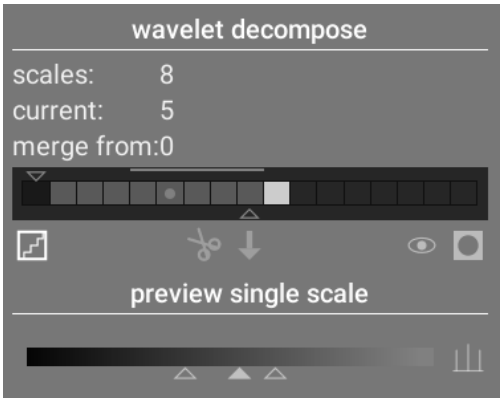
As you can see, the large-scale skin blotches have been removed, but the small-scale details remain untouched.

interacting with wavelet scales

There are two methods by which processing modules allow you to modify their operation using wavelet scales.

wavelet decomposition

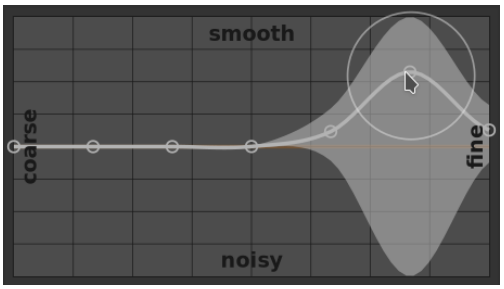
As discussed above, the *retouch* module allows you to choose how many detail levels to split your image into. It decomposes the image into separate layers and allows you to perform operations selectively on each individual layer or on the image as a whole:



See the *retouch* module documentation for more details.

spline controls

The *denoise (profiled)* , *raw denoise* and *contrast equalizer* modules allow their effects to be applied more or less to each wavelet scale using *splines*.



Here, each node in the graph represents a different level of detail in the image, from coarse detail on the left to fine detail on the right. You can raise or lower each of these nodes with your mouse to increase or decrease the module's effect, respectively, on that wavelet scale.

To modify the curve, click slightly above or below the line near to a node and drag up or down. You can change the width of your modification by scrolling with your mouse wheel, which increases or reduces the size of the circle displayed under your mouse pointer. A small circle indicates that the effect of dragging the curve up or down will be isolated mostly to the node being adjusted. A larger circle indicates that the effect will be more broad and will increasingly impact adjacent nodes. When you hover your mouse over the graph, shaded areas indicate the parts of the spline that will be impacted when you attempt to modify the curve.

3.4.7. deprecated modules

The darktable team regularly reviews old modules and updates their implementation where issues are found or updated science means that they can be improved. Most of the time we try to update existing modules with new functionality but occasionally that becomes problematic, often due to the overhead of having to maintain multiple versions of the module. In such circumstances a new replacement module is created and the old module becomes *deprecated*.

Module deprecation follows a fairly standard process: In the release that deprecates a module, that module can no longer be searched or added to custom module groups. It is moved into the read-only *deprecated modules* group and a message added to the module to warn of its removal and direct the user to a suitable alternative.

After a year, the module is removed from the *deprecated modules* group and is then only available for use in old edits (where that module was already active against a given image). Deprecated modules are never fully removed from darktable (in order to retain old edits) but development ceases and you are advised to use supported modules instead if you encounter an issue.

For example, modules deprecated in darktable 3.4 will be removed from the *deprecated modules* group in darktable 3.8 (assuming two major releases each year).

3.5. masking & blending

3.5.1. overview

Each processing module takes its input from the preceding module in the pixelpipe, performs its operation on the image data, and then hands the output to the next module in the pixelpipe.

A module's output data can optionally be reprocessed (combined) with its input data before being handed to the next module. This additional processing step is called *blending* – input and output data is reprocessed using algorithms called blending operators or blend modes .

Each blend mode is further controlled by the *opacity* parameter (having a value between 0% and 100%) which defines how much the input and output images contribute to the final result. Typically an opacity of 0% outputs an image which is identical to the input image (the module has no effect) whereas an opacity of 100% delivers the maximum effect of the module.

This opacity can be the same for every pixel (using the global opacity slider), in which case blending acts uniformly over the entire image. Alternatively the opacity values can vary depending on the properties or location of each pixel. This local modification of opacity is called a *mask*. Masks provide the user with fine control over which parts of an image are affected by a module and to what extent. The user may choose to activate a drawn mask , a parametric mask or a combination .

Blending and masking functionality is controlled via a group of icons located at the bottom of each applicable module.



These icons enable the following masking and blending options, from left to right:

off
: Module output is passed to the next module in the pixelpipe without additional reprocessing. No further controls are displayed.

uniformly
: Input and output images are reprocessed uniformly with the chosen blend mode, where the amount of blending is controlled by a single opacity slider. Additional controls are displayed to allow the blend mode and opacity to be selected. The default is a blend mode of "normal" with an opacity of 100%.

drawn mask
: Reprocessing takes place with the chosen blend mode and an opacity based on pixel location as defined by one or more drawn shapes. Additional controls are displayed to allow mask elements to be drawn. If no mask elements are drawn then all pixels have the same opacity, as defined by the opacity slider.

parametric mask
: Reprocessing takes place with the chosen blend mode and an opacity based on the properties of individual pixels. Additional controls are displayed to allow the opacity to be adjusted on a per-pixel basis, determined by pixel values.

drawn & parametric mask
: Reprocessing takes place with the chosen blend mode and an opacity based on a combination of a drawn and parametric mask.

raster mask
: Reprocessing takes place with the chosen blend mode and an opacity based on a mask that was generated by an active module earlier in the pixelpipe.

blending options
: Choose which color space to use when calculating the blending mask, and specify whether or not to allow a mask to be generated based on the module's output channels (normally a parametric mask is generated based on the input channels coming into the module). The following options are available:

- *reset to default blend colorspace*: Use the default color space for the module to specify the parametric mask.
- *Lab*: Use the Lab color space (where available) to specify the parametric mask.
- *RGB (display)*: Use the display-referred RGB/HSL color space to specify the parametric mask.
- *RGB (scene)*: Use the scene-referred RGB/$J_zC_zh_z$ color space to specify the parametric mask.
- *show output channels*: Show the parametric mask output channel controls, so that the parametric mask can be defined in terms of the module's output channels.

Note: Not all of these blending options are available for every module.

3.5.2. blend modes

Blend modes define how the input and output of a module are combined (blended) together before the module's final output is passed to the next module in the pixelpipe.

Classic blending modes, designed for display-referred RGB (constrained to 0-100%), implicitly define a fulcrum at 50% (gray) or 100% (white) in their algorithms, depending on the blend mode. Because scene-referred is not subject to these restrictions, this fulcrum needs to be explicitly defined by the user when performing blending operations in the "RGB (scene)" color space. The additional *blend fulcrum* parameter will be presented to the user when using one of these blend modes in this color space. The effect depends on the operator used. For example, values above the fulcrum might be brightened and values below darkened, or vice versa.

The final output of a module is computed 'per-pixel' as follows:

```
final_output = (1.0 - opacity) * module_input + opacity * blended_output
```

where the `blended_output` is a combination of the input and output images, depending on the blend mode (below), and the `opacity` is defined 'per-pixel' by a combination of the mask and global opacity parameter. An opacity of 0% outputs an image that is identical to the input image of the module.

The "reverse" button  effectively reverses the roles of the input and output images in the 'per-pixel' computation:

```
final_output = (1.0 - opacity) * module_output + opacity * blended_input
```

where the `blended_input` is a combination of the output and input images, depending on the blend mode below where *output* and *input* image references are reversed. In "reversed" blend modes, an opacity of 0% outputs an image that is identical to the *output* image of the module.

normal modes

normal

The most commonly used blend mode, "normal" simply mixes input and output to an extent determined by the opacity parameter. This mode is commonly used to reduce the strength of a module's effect by reducing the opacity. This is also usually the blend mode of choice when applying a module's effect selectively with masks. This mode is also known as the "over" Porter-Duff alpha blending operator (see alpha compositing for more details).

normal bounded

not available in the "RGB (scene)" color space

This blend mode is the same as "normal", except that the input and output data are clamped to a particular min/max value range. Out-of-range values are effectively blocked and are not passed to subsequent modules. Sometimes this helps to prevent artifacts. However, in most cases (e.g. highly color-saturated extreme highlights) it is better to let unbound values travel through the pixelpipe to be properly handled later. The "normal" blend mode is therefore usually preferred.

arithmetic modes

addition

Add together the pixel values of the input and output images, lightening the output. When blending in the "RGB (scene)" color space, the pixel values of the output image are multiplied by a value proportional to the "blend fulcrum".

subtract

Subtract the pixel value of the *output* from the *input*. When blending in the "RGB (scene)" color space, the pixel values of the output image are multiplied by a value proportional to the "blend fulcrum". Pixel values less than 0 are set to 0.

multiply

Multiply the pixel values of the input and output together. When blending in display-referred color spaces, pixel values are between 0 and 1.0, the final output will be clamped and will always be darker. When blending in the “RGB (scene)” color space, this value is further multiplied by a value proportional to the “blend fulcrum”. In this case, values may be greater than 1.0 and therefore brighten the base image. This may have other side-effects, such as updating the white point in the filmic module.

Multiply blending simulates an optical variable density filter, where the density is defined by the output of the module. It has many applications, from blooming and local contrast enhancements (when used with a blur or low-pass filter) to dodging/burning and global contrast enhancements (when used with exposure). The fulcrum sets the output intensity threshold between darkening and brightening (any RGB value below fulcrum will darken).

divide

Divide the pixel values of the input by the output. When blending in the “RGB (scene)” color space, the pixel values of the output image are multiplied by a value proportional to the “blend fulcrum”.

Since this is the inverse of the multiply mode, it will darken where multiply brightens and vice versa. Everything else works in essentially the same way.

screen

not available in the “RGB (scene)” color space

Invert the input and output pixel values, multiply those values together and invert the result. This yields approximately the opposite effect to “multiply” mode – the resulting image is usually brighter, and sometimes “washed out” in appearance.

average

Return the arithmetic mean of the input and output pixel values.

difference

Return the absolute difference between the input and output pixel values.

geometric mean

Return the square root of the product of the input and output pixel values.

harmonic mean

Return the product of the input and output pixel values, multiplied by 2 and divided by their sum.

contrast enhancing modes

The following modes are not available in the “RGB (scene)” blending color space as they rely on an assumption of “50% mid gray” which only applies to display-referred and non-linear color spaces. In addition, they clamp pixel values of *both* input and output images as the underlying math is not valid outside the range 0..1, and can thus cause visible changes *anywhere* in the image. To avoid these, you will need to ensure that the module’s input does not exceed the display-referred range.

overlay

This mode combines the “multiply” and “screen” blend modes: The parts of the input where the output is brighter, become brighter; The parts of the image where the output is darker, become darker; Mid-gray is unaffected. This mode is often combined with the *lowpass* and *highpass* modules.

softlight

This mode is similar to “overlay”, except the results are softer and less bright. As with overlay, it is often combined with the *lowpass* and *highpass* modules.

hardlight

This mode is not related to “softlight” in anything but name. Like overlay mode it is a combination of “multiply” and “screen” modes and has a different effect above and below mid-gray. The results with hardlight blend mode tend to be quite intense and usually need to be combined with a reduced opacity.

vividlight

This mode is an extreme version of overlay/softlight. Values darker than mid-gray are darkened; Values brighter than mid-gray are brightened. You will probably need to tone down its effect by reducing the opacity

linearlight

This mode is similar to the effect of “vividlight”.

pinlight

This mode performs a darken and lighten blending simultaneously, removing mid-tones. It can result in artifacts such as patches and blotches.

color channel modes

Lab channels

The following are available for blending in the Lab color space only

Lab lightness
Mix the lightness from the input and output images, while taking the color channels (a and b) unaltered from the input image. In contrast to “lightness” this blend mode does not involve any color space conversion and does not clamp any data. In some cases this blend mode is less prone to artifacts than “lightness”.

Lab a-channel
Mix the Lab “a” color channel from the input and output images, while taking the other channels unaltered from the input image.

Lab b-channel
Mix the Lab “b” color channel from the input and output images, while taking the other channels unaltered from the input image.

Lab color
Mix the Lab color channels (a and b) from the input and output images, while taking the lightness unaltered from the input image. In contrast to “color” this blend mode does not involve any color space conversion and does not clamp any data. In some cases this blend mode is less prone to artifacts than “color”.

RGB channels

The following are available when blending in RGB color spaces only.

RGB red channel
Mix the “red” channel from the input and output images, while taking the other channels unaltered from the input image. When blending in the “RGB (scene)” color space, the “red” channel from the output image is multiplied by a value proportional to the “blend fulcrum”.

RGB green channel
Mix the “green” channel from the input and output images, while taking the other channels unaltered from the input image. When blending in the “RGB (scene)” color space, the “green” channel from the output image is multiplied by a value proportional to the “blend fulcrum”.

RGB blue channel
Mix the “blue” channel from the input and output images, while taking the other channels unaltered from the input image. When blending in the “RGB (scene)” color space, the “blue” channel from the output image is multiplied by a value proportional to the “blend fulcrum”.

HSV channels

The following are available when blending in the “RGB (display)” color space only.

HSV value
Mix the lightness from the input and output images, while taking color unaltered from the input image. In contrast to “lightness” this blend mode does not involve clamping.

HSV color
Mix the color from the input and output images, while taking lightness unaltered from the input image. In contrast to “color” this blend mode does not involve clamping.

others

lightness
Mix lightness from the input and output images, while taking color (chromaticity and hue) unaltered from the input image.

chromaticity
Mix chromaticity from the input and output images, while taking lightness and hue unaltered from the input image. This blend mode uses RGB ratios, divided by a Euclidean norm.

lighten
not available in the “RGB (scene)” color space

Compare the pixel values of the input and output images, and output the lighter value.

darken

not available in the “RGB (scene)” color space

Compare the pixel values of the input and output images, and output the darker value.

hue

not available in the “RGB (scene)” color space

Mix hue (color tint) from the input and output images, while taking lightness and chroma unaltered from the input image.

color

not available in the “RGB (scene)” color space

Mix color (chroma and hue) from the input and output images while taking lightness unaltered from the input image.

Caution: When modules drastically modify hue (e.g. when generating complementary colors) this blend mode can result in strong color noise.

coloradjustment

not available in the “RGB (scene)” color space

Some modules act predominantly on the tonal values of an image but also perform some color saturation adjustments (e.g. the *levels* and *tone curve* modules). This blend mode takes the lightness from the module’s output and mixes colors from input and output, enabling control over the module’s color adjustments.

3.5.3. masks

3.5.3.1. overview

Masks allow you to limit the effect of a module so that it only applies to certain parts of the image.

A mask can be regarded as a grayscale image where each pixel has a value between 0 and 1.0 (or between 0% and 100%). This value is the *opacity* and is used to determine how much a module affects each pixel.

The following sections explain how to construct masks in darktable.

3.5.3.2. drawn masks

With the drawn mask feature you can construct a mask by drawing shapes directly onto the image canvas. Shapes can be used alone or in combination. Once a shape has been drawn on an image it can be adjusted, removed, or reused in other modules.

Shapes are stored internally as vectors and are rendered with the required resolution during pixelpipe processing. Shapes are expressed in the coordinate system of the original image and are transformed along with the rest of the image by any active distorting modules in the pipe (*lens correction* , *rotate and perspective* for example). This means that a shape will always work on the same image area regardless of any modifications that may be subsequently applied.

The controls required to create and alter drawn masks may be enabled by selecting either the “drawn mask” or “drawn & parametric mask” icon at the bottom of a module. You can also create and edit shapes using the mask manager module.

creating shapes

Choose a shape by clicking on the appropriate shape icon (from left to right: brush, circle, ellipse, path, gradient, and AI object – the latter if AI features are enabled and a model for the *mask* task is active).



This will take you into the creation mode for that shape. Once you have finished drawing your shape you will automatically be taken into edit mode.

Ctrl+click on the shape icon to continuously draw multiple shapes of the same type – each time a shape is completed, you will re-enter creation mode for a new instance of that shape. While in continuous creation mode, right-click on the image to stop drawing shapes and enter edit mode.

For all drawn shapes, you can hold Shift while scrolling with the mouse wheel to change the extent of the shape's feathering (the blur at the edge of the shape) and use Ctrl+scroll to change the shape's opacity (how transparent it is). The feathering shape can also be changed by dragging a node on the feathering outline. If the feathering node coincides with a path node, hold Shift when selecting the node to limit selection to feathering nodes. These operations are available in both creation and edit modes (as long as your mouse is over the shape in question).

By default, scrolling your mouse up increases the value of the relevant shape parameters. This behavior can be changed in preferences > darkroom > scroll down to increase mask parameters .

Note: When used in shape creation mode, the preceding scroll operations will also cause the *default* feathering or opacity to be changed. The new default values will be used the next time you create a new shape.

---

editing shapes

Click on the 'show and edit mask elements' icon  to enter shape edit mode. This will show all drawn masks in use by the current module and allow you to edit those shapes. It will also expand the associated group in the mask manager . Ctrl+click on the same icon to enter restricted edit mode, where certain actions (e.g. dragging a complete shape or changing its size) are blocked. This is particularly useful to avoid costly mistakes when editing path and brush shapes.

Click and drag a shape to move it around the image canvas. Clicking on a shape will also select that shape in the mask manager.

Ctrl+click and drag within the shape to rotate it.

removing shapes

While in edit mode right-click on a shape to remove it.

reusing shapes

You can reuse shapes that you have drawn in other modules. Click on the shapes drop-down (next to the 'show and edit mask elements' button) to choose previously-drawn shapes individually or to use the same group of shapes as used by another module. The following options are available for selection:

add existing shape
: Choose either an individual shape or a group of shapes that you've drawn previously (either within the mask manager or from within the drawn mask of another module). If that shape or group is used elsewhere, any changes you make will be reflected everywhere the shape or group is used.

use same shapes as
: Add a list of shapes used in another module to the current module's mask. This differs from the previous option in that it creates a new group of shapes, allowing shapes to be added to or removed from the group independently of the module from which they were copied. All shapes that are common to both groups remain linked.

combining and managing shapes

The mask manager module can be used to manage your drawn shapes. This module also allows you to group and combine drawn masks using set operators (union, intersection, difference, exclusion).

shape distortions

In order to ensure a consistent co-ordinate system, when you place a shape on the image, it is actually drawn on the original RAW file. This shape then passes up through the pixelpipe before finally being used by the module and drawn on the screen. This means that, if you have enabled any distorting modules (such as lens correction), drawn shapes may appear distorted on the screen and in the final image. This can lead, for example, to circles being rendered as ellipses and gradient lines becoming curved. If you need to create a more accurate shape (to overcome these distortions) it is recommended that you avoid using the simple shapes (circles / ellipses) in favor of the path shape (which can be drawn using more points, reducing distortions). You can adjust the curve on gradient lines to overcome the simple distortions introduced by lens correction.

available shapes



brush

Start drawing a brush stroke by left-clicking on the image canvas and moving the mouse while keeping the button pressed. The brush stroke is finalized once you release the mouse button. Scroll the mouse to change the shape size and Shift+scroll to change the feathering (hardness), either before you start drawing or at any time during the operation. Likewise you can use the “{” and “}” keys to decrease/increase hardness, and the “<” and “>” keys to decrease/increase opacity.

If you have a graphics tablet with pen pressure sensitivity, darktable can apply the recorded pen pressure to certain attributes of the brush stroke. This operation can be controlled in preferences > darkroom > pen pressure control for brush masks .

On lifting the tablet pen or releasing the left mouse button the brush stroke is converted into a number of connected nodes, which define the final shape. A configuration option (preferences > darkroom > smoothing of brush strokes) controls how much smoothing is applied. A higher level of smoothing leads to fewer nodes being created – this eases subsequent editing at the expense of lower accuracy.

Nodes and segments of a brush stroke can be modified individually. See the documentation on path shapes (below) for more details. Change the size or hardness of a node by scrolling and Shift+scrolling over a node, respectively.

Note: Rendering a complex brush shape can consume a significant number of CPU cycles. Consider using the circle, ellipse or path shapes instead where possible.



circle

Click on the image canvas to place the circle. Scroll while hovering over the circle to change its diameter. Scroll while hovering over the circle’s border to change the width of the feathering (the same effect as holding Shift while scrolling with the mouse wheel within the main shape).



ellipse

The general principle is the same as for the circle shape. In addition, four nodes are shown on the ellipse line. Click and drag the nodes to adjust the ellipse’s eccentricity. Ctrl+click and drag the nodes or use Shift+Ctrl+scroll (with the mouse wheel) to rotate the ellipse. Alt+click within the shape to toggle the gradual decay between equidistant and proportional mode.

path

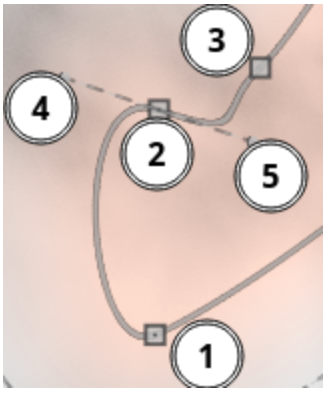
Click on the image canvas to place three or more nodes and generate a free-format enclosed shape. Terminate the path by right-clicking after having set the last point. By default, nodes are connected with smooth lines. If you want a node to instead define a sharp corner, you can do so by creating it with Ctrl+click.

In edit mode Ctrl+click on an existing node to convert it from smooth to sharp corners and vice versa. Shift+click on one of the line segments to insert an additional node. Right-click on a node to delete it. Take care to ensure that the mouse pointer is over the desired node and the node is highlighted, to avoid accidentally removing the whole path.

The size of the completed shape can be modified by scrolling. Normally resizing alters the boundary size by a constant distance, but an alternative centroid-based resize which resizes the whole outline proportionally is activated with Ctrl+Shift+scroll.

The width of the border (the area with a gradual opacity decay) can also be resized, with Shift+scroll (with the mouse wheel) from anywhere within the shape.

Single nodes as well as path segments can be moved by dragging them with the mouse. If a node is selected by clicking on it, two Bézier handles appear which allow you to modify the curvature of the line (reset to the default curvature by right-clicking on either of the handles). In the image below, the selected node (2) has two handles (4 and 5). Altering the position or length of the handles alters the curvature between node 2 and the adjacent nodes (1 and 3).



Dragging one of the control points on the border adjusts the border width just in that part of the shape.

Consider fine-tuning paths in restricted edit mode (enabled by Ctrl+clicking on the ‘show and edit mask elements’ icon). This allows you to adjust single nodes and segments without the risk of accidentally shifting or resizing the whole shape.

gradient

The gradient shape is a linear gradient which extends from a given point to the edge of the image.

Click the image canvas to place the gradient. The central line marks 50% opacity, the dotted lines to the sides indicate 100% and 0% opacity respectively. Between these dotted lines the opacity changes linearly. You can compress/expand the distance between the dotted lines by scrolling with your mouse wheel while pressing Shift. Note that the dotted lines will only appear while hovering over one of the lines with your cursor.

The line has two anchor nodes which you can drag to change the rotation of the gradient. You can also set the rotation angle when placing the gradient shape by clicking and dragging to place the shape.

Gradient lines can also be curved by scrolling with your mouse while hovering close to the center line. This can be useful to counteract the distortion caused by the *lens correction* module.

Depending on the module and the underlying image, using a gradient shape might provoke banding artifacts. You should consider activating the *dither or posterize* module to alleviate this.

AI object *(requires AI features)*

An AI-generated mask: instead of drawing the outline by hand, click on the subject and a segmentation model produces the mask for you. Iterate by adding more clicks until the mask covers exactly what you want, then apply it. Requires AI features to be enabled and a model for the *mask* task active in AI preferences – see the AI features overview for what runs where.

- **click** on the subject to add a foreground (positive) prompt point – the model produces a mask around the clicked region.
- **shift+click** to add a background (negative) point, telling the model "exclude this area from the mask". Useful when the mask spills onto a neighboring object you didn't want.
- **ctrl+shift+click** to clear all prompt points and start over.
- **right-click** to apply the mask. The result is vectorized into a group of path shapes and registered with the mask manager .

After applying, the result is *stored* as a regular path shape – the AI-segmentation output is vectorized into Bézier paths once and persisted in that form. Nodes are editable, the group can be combined with other shapes via set operators, polarity can be reversed, and so on. The AI model is not involved beyond the initial generation, so the mask renders just as cheaply as any other vector shape and works in modules and styles that have no AI dependency.

First click on a new image runs the encoder once and caches the result on disk; later clicks reuse the cache (even across restarts) so only the lightweight decoder re-runs per prompt point. Changing any module that distorts or crops the image (crop, rotate and perspective, lens correction, etc.) invalidates the cache and forces a re-encode on the next click.

Tips for better results: The model selects one visually distinct object at a time. If the first click picks only part of the subject (an arm rather than the whole person, a wheel rather than the whole car), add more clicks on *adjacent* parts to extend the mask. Clicking on a visually separate area usually switches the model to a new selection rather than extending the current one – when that happens, reset with **ctrl+shift+click** and start over instead of trying to repair the mask with corrections.

---

reversing the polarity of a drawn mask

Click on the "+/-" button to reverse the polarity of the entire drawn mask. For example, a circular mask will, by default, cause the module to be applied only to the area inside the drawn circle. Reversing its polarity will cause the module to apply to the whole image, *except for* that circle.

panning and zooming the image

While creating or editing a shape, mouse actions are applied to the current shape. If you need to move or zoom the portion of the image shown in the center view, hold down the 'a' key while dragging the mouse or using the scroll wheel. While the key is held down, the mouse actions will apply to the entire image rather than the current shape.

3.5.3.3. parametric masks

The parametric mask feature offers fine-grained selective control over how individual pixels are masked. It does this by automatically generating an intermediate blend mask from user-defined parameters. These parameters are color coordinates rather than the geometrical coordinates used in drawn masks.

For each data channel of a module (e.g. Lab, RGB) and several virtual data channels (e.g. hue, saturation) you can construct a per-channel opacity function. Depending on each pixel's value for a given data channel this function calculates a blending factor between 0 and 1 (100%) for that pixel.

Each pixel of an image thus has different blending factors for each of its data channels. All blending factors are finally multiplied together (pixel-by-pixel), along with the value of the global opacity slider, to form a complete parametric blend mask for the image.

If the blend mask has a value of 0 for a given pixel, the input of the module is left unchanged. If the blend mask has a value of 1 (100%) for a pixel, the module has its full effect.

channel tabs

Click on one of the channel tabs to select a data channel to use to build your mask.

Modules acting in (*display-referred*) *Lab* color space have data channels for L, a, b, C (chroma of LCh) and h (hue of LCh).

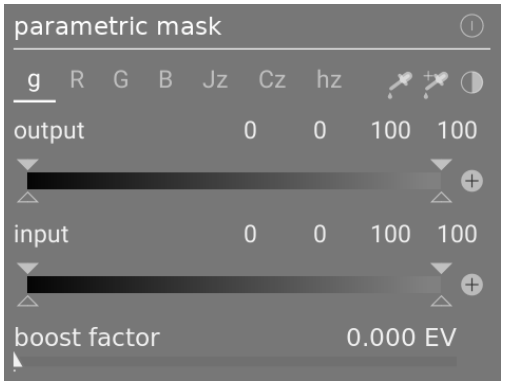
Modules acting in *display-referred RGB* color space have data channels for g (gray), R, G, B, H (hue of HSL), S (saturation of HSL), and L (lightness of HSL).

Modules acting in *scene-referred RGB* color space have data channels for g (gray), R, G, B, Jz (luminance component of JzCzhz), Cz (chroma, or saturation, of JzCzhz), and hz (hue of JzCzhz). The g (gray) value is calculated as a weighted average of the R, G & B channels, the exact weightings depending on the working color space being used. The JzCzhz color space is a polar representation of the Jzazbz color space, in the same way that LCh is a polar representation of the Lab space. Like the L in Lab color space, the Jz is a representation of the luminosity of a pixel that aligns with how we perceive brightness. However, the Jzazbz color space is much better for high dynamic range images and is less susceptible to hue shifts than Lab space.

See Wikipedia for more details about these color spaces.

Two sliders can be shown for each associated data channel: one that works on the *input data* that the module receives and one that works on the *output data* that the module produces prior to blending. The sliders for the output data channels are hidden by default and can be shown using the *show output channels* option in the blending menu.

The *boost factor* slider allows the range of values targeted by the parametric mask sliders to be extended. It may be used in scene referred editing, where luminance values may extend beyond 100%, to target highlights. This slider is only available for channels where it is meaningful.



inspecting data channels & masks

Press the letter C while hovering over a channel's input/output slider to view the input/output image data for that color channel. The center image changes to display that color channel either in gray-scale values or in false colors depending on the setting in preferences > darkroom > display of individual color channels .

Press the letter M to see the resulting mask for that slider overlaid on the image.

When the mouse pointer leaves the slider the image returns to normal after a short delay.

linear / log mode

Press the letter A while hovering over the a slider to change its display to 'log' mode. This provides more fine control in the shadows. Press A again to toggle back to 'linear' mode.

channel input/output sliders

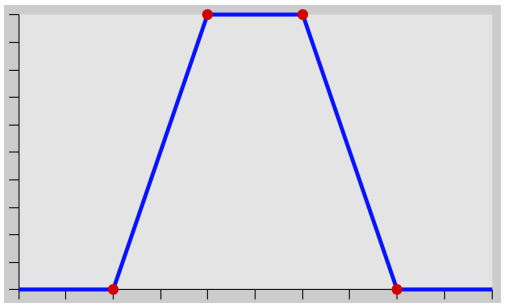
With each color channel slider you can construct a trapezoidal opacity function. For this purpose there are four markers per slider. Two filled triangles above the slider mark the range of values where opacity is 1. Two open triangles below the slider mark the range values where opacity is 0. Intermediate points between full and zero are given a proportional opacity.

The filled triangles, or inside markers, indicate the closed (mostly narrower) edge of the trapezoidal function. The open triangles, or outside markers, indicate the open (mostly wider) edge of the trapezoidal function. The sequence of the markers always remains unchanged: they can touch one another but they cannot switch position.

A polarity (+/-) button to the right of each the slider switches between “range select” and “range de-select” modes, with visual confirmation provided by exchanging the upper and lower triangle markers. These two types of trapezoidal functions are represented graphically in the following images.

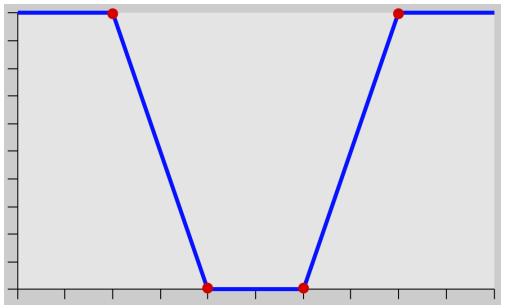
range select





range deselect





In their default state all markers are at their extreme positions.

In this state a range select function selects the whole range of values giving an “all at 100%” mask. Starting from there one can move the sliders inwards to gradually exclude more and more parts of the image except for the remaining narrow range.

Conversely a range de-select function (enabled by toggling the polarity) by default deselects the whole range of values, giving an “all-zero” mask as a starting point. Moving the sliders inwards gradually includes more and more parts of the image except for the remaining narrow range.

pickers

With the left-hand picker button you can select a point or area probe from your image. The corresponding values for the real and virtual data channels are then displayed within each color channel slider.

With the right-hand picker button you can automatically set the slider's values based on the selected range. Click and drag to set the parameters for the input slider from the drawn rectangle; Ctrl+click and drag to set the parameters for the output slider.

invert

Click the invert button above the sliders to invert the polarity of the entire parametric mask. This differs from the polarity buttons beside the individual sliders which just invert the parameters for the current slider/channel.

reset

Click the reset button above the sliders to revert all parametric mask parameters to their default state.

3.5.3.4. combining drawn & parametric masks

Drawn and parametric masks can be used in combination to form a single mask that can be applied to a module.

There are two main elements which control how individual masks are combined: the *polarity* setting of each individual mask (defined by the plus or minus buttons) and the setting in the "combine masks" combobox.

The "combine masks" combobox contains the following options, defining how the drawn and parametric masks will be combined:

exclusive

A straightforward method of combining masks, exclusive mode multiplies together the individual pixel values from each of the component masks.

For a given pixel, the final mask will have value of 0 if *any* of the individual masks are 0 at that location and it will only have a value of 1.0 if *all* masks have a value of 1.0 at that location.

Any individual mask can *exclude* a pixel by setting its value to 0, regardless of what the other masks do. Once a pixel is excluded by a mask there is no way for another mask to include it again.

inclusive

Inclusive mode first inverts each individual mask (subtracts its value from 1.0), multiplies the inverted masks together, and finally inverts the combined mask once again.

For a given pixel, the final mask will have a value of 1.0 if *any* of the individual masks are 1.0 at that location and it will only have a value of 0 if *all* masks have a value of 0 at that location.

Any individual mask can *include* a pixel by setting its value to 1.0, regardless of what the other masks do. Once a pixel is fully included by a mask (its value is 1.0) there is no way for another mask to exclude it again.

exclusive and inclusive inverted modes

Using the above combination methods alone would still be rather limiting. We gain maximum flexibility by allowing an additional inversion step for each individual mask. This is governed by the polarity buttons that you find close to the individual channels.

Toggling the polarity button of a mask inverts its values (subtracts the original value from 1.0).

Finally within the "combine masks" combobox you may select the *exclusive & inverted* or *inclusive & inverted* options. Each of these options is equivalent to the *exclusive* and *inclusive* modes, respectively, but with a final step that inverts the resulting mask.

typical use cases

inclusive mode

For this mode you set the "combine masks" combobox to inclusive mode and make sure that all polarity buttons of all the individual channels and of the drawn mask are set to negative (-). Your starting point is a mask where all pixels have a value of zero (no pixel is selected). You now adjust the parametric mask sliders to bring more and more pixels into the selection or you draw shapes on the canvas to select specific areas of your image.

exclusive mode

In the opposite case you set the “combine masks” combobox to exclusive mode and make sure that all polarity buttons are set to positive (+). Your starting point is a mask with all values at 1.0 (all pixels are selected). You now adjust the parametric mask sliders to exclude parts of your image as needed or directly draw shapes on the canvas to exclude those areas.

For your convenience the parametric masks GUI provides a toggle button that inverts all channel polarities and toggles between inclusive and exclusive mode in the “combine masks” combobox.

For novice users it is recommended that you stick to the above two use cases. This means that you should decide beforehand how you want to construct your mask.

3.5.3.5. mask refinement & additional controls

When a mask is active, several additional sliders are shown which allow the mask to be further refined.

details threshold

This control allows you to alter the opacity of the mask based on the amount of detail in the image. Use this slider to select either areas with lots of detail (positive values) or areas that are flat and lacking in detail (negative values). The default (zero) effectively bypasses details refinement. This is mostly useful to apply sharpening and blurring effects that ignore out-of-focus (bokeh) regions or to sharpen only blurry parts, preventing over-sharpening of in-focus regions.

Note: The data used for the detail mask refinement is taken from the demosaic or rawprepare stage in the processing pipeline, not from the module’s input (which is used for the other parametric mask criteria). None of the processing modules after demosaic will have any effect on the detail mask and it is not available for non-RAW images.

---

feathering guide

Mask feathering smooths a mask such that the mask’s edges automatically align with the edges of features in the image. The smoothing is guided either by the module’s input or output (before blending), and may happen before or after the mask blurring, depending on what is selected in the “feathering guide” combobox. Feathering is particularly sensitive to the choice of guide image when used with edge-modifying modules (modules for sharpening or blurring an image).

- *output before blur*: feathering is guided using the *output* image of the module and takes place *before* the mask is blurred
- *input before blur*: feathering is guided using the *input* image of the module and takes place *before* the mask is blurred
- *output after blur*: feathering is guided using the *output* image of the module and takes place *after* the mask has been blurred
- *input after blur*: feathering is guided using the *input* image of the module and takes place *after* the mask has been blurred

feathering radius

Adjust the strength of the feathering effect. Feathering works best if the mask’s edges already approximately match some edges in the guiding image. The larger the “feathering radius” the better the feathering algorithm can align the mask to more distant edges. If this radius is too large, however, the feathered mask may overshoot (cover regions that the user wants to exclude). Feathering is disabled when the feathering radius is set to 0.

blurring radius

Blurring the mask creates a softer transition between blended and unblended parts of an image and can be used to avoid artifacts. The blurring radius slider controls the radius of a gaussian blur applied to the final blend mask. The higher the radius, the stronger the blur (set to 0 for an unblurred mask). Mask blur is always applied after feathering if both kinds of mask adjustment are activated. This allows any resulting sharp edges or artifacts to be smoothed.

mask opacity

The strength of the module’s effect is determined by the mask’s local opacity. Feathering and blurring the mask may reduce the opacity of the original mask. The “mask opacity” slider allows you to readjust the mask opacity to compensate. If the mask opacity is decreased (negative slider values) less opaque parts are affected more strongly. Conversely, if the mask opacity is increased (positive slider values) more opaque parts are affected more strongly. As a consequence, completely opaque portions of the mask always remain opaque and completely transparent portions always remain transparent. This is to ensure that regions that have been fully excluded from or included in a module’s effect (by setting the mask’s opacity to 0% or 100%) remain fully excluded or included.

mask contrast

This slider increases or decreases the mask contrast. This allows you to adjust the transition between the opaque and transparent parts of the mask.

temporarily switch off mask

Sometimes it is useful to visualize a module's effect without the mask being active. Click on this icon to temporarily deactivate the mask (the selected blend mode and global opacity remain in effect).

display mask

Click on this icon to display the current mask as a yellow overlay over a black-and-white version of the image. Solid yellow indicates an opacity of 100%; a fully visible gray background image (without yellow overlay) indicates an opacity of 0%.

example: feathering a drawn mask



It can be rather tedious to create a drawn mask that precisely covers a particular feature in an image. In this example, we want to enhance the color contrast of the lion sculpture shown in the left image above without affecting the background.

1. The first image above shows the original, unaltered image.
2. The second image shows a rough selection of the sculpture created with a drawn mask. Note that the mask is rather fuzzy and does not precisely follow the outline of the lion sculpture.
3. The third image shows the effect of adjusting the feathering radius, mask opacity and mask contrast, leading to a well matched mask with little effort. In this example the feathering radius has been adjusted to 50 and a blur radius of 5 was chosen to smooth the mask to some degree. The mask opacity and mask contrast have been increased to 0.3 and 0.5, respectively.
4. The final image above shows the end result, where the color enhancement (via the *color contrast* module) is restricted to only the lion sculpture.

Mask feathering works particularly well in this example because the sculpture is well separated from the out-of-focus background. The distinct edge at the border of the sculpture guides the feathering mask adjustment to match the shape of the sculpture.

3.5.3.6. AI masking

AI masking requires AI features to be enabled in AI preferences , with a model active for the AI task the specific masking tool uses (each tool below states which one). See the AI features overview for what runs where.

Currently darktable provides one AI masking tool:

AI object

Uses the *mask* task. Click on a subject to produce a mask around it; iterate by adding more clicks until it covers exactly what you want. The result is vectorised into a group of path shapes and can be edited like any other drawn shape – see drawn masks: AI object for details.

3.5.3.7. raster masks

As described in the previous sections, the final output of a module's mask (the combined effect of any drawn and parametric masks) is a grayscale raster image representing the extent to which the module's effect should be applied to each pixel. This raster image is stored internally for active modules and can be subsequently reused by other modules in the pixelpipe.

As with any mask, if the opacity value for a pixel in a raster mask is zero, the module's input is passed through the module unchanged. If the opacity is 1.0 the module has its full effect. For each value between 0 and 1.0 the module's effect is applied proportionally at that location.

You can choose a raster mask from the combobox. Raster masks can be identified by the name of the module against which they were originally generated.

Raster masks can also be imported from images generated by other applications – see the *external raster mask* module documentation.

Note: Raster masks are generated as part of a module's internal processing. Once a module's processing is complete its mask then becomes available to subsequent modules in the pixelpipe.

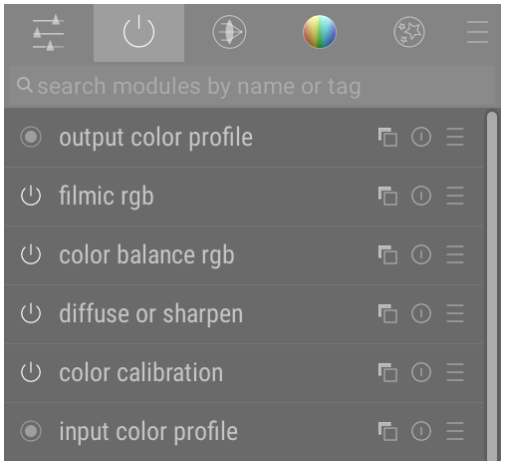
This has two implications:

1. Raster masks cannot be generated by disabled modules since they do not participate in pixelpipe processing. As soon as you disable a module, its mask is no longer available for use.
2. Raster masks are passed *up* the pixelpipe after module processing – they can only be used by modules that come later in the pipe than the generating module.

3.6. organization

3.6.1. overview

Processing modules are organized and accessed via the right-hand panel in the darkroom:



Click on the icons at the top of this panel to reveal, from left to right,

quick access panel

A customizable panel allowing quick access to commonly-used module controls.

active modules

A group containing the modules that are currently active in the pixelpipe. Right-click to show all modules that are present in the history stack within the active group, regardless of whether or not they are actually active.

module groups
One or more groups of processing modules. These groups are user-defined but some default groupings are provided as presets.

presets menu
A menu that allows you to access stored module layout presets and create your own (via the “manage presets..” option). You can also directly access the “manage presets” dialog by Ctrl+clicking on the presets menu.

Click once on a module group icon (including the active group) to show only the modules in that group. Click a second time to show a list of all modules that are currently active or present in any group.

You can change which widgets appear in the quick access panel and which modules appear in the module groups, by right-clicking the appropriate icon.

search

Below the module group icons is the search bar, which you can use to access any of the processing modules, regardless of whether they are currently in a group. This option allows you to search by module name, any user-defined instance name , as well as pre-defined module tags (for example, the color calibration rgb module can also be searched using the terms “hue”, “contrast” and “vibrance”).

3.6.2. module groups

A number of pre-defined module groups are shipped with darktable and are selectable as presets. These are summarized below.

All of these presets (with the exception of *modules: deprecated* and *search only*) also include the quick access panel. All except the *modules: deprecated* group include the search bar.

modules: all

This preset contains all modules, sorted according to the traditional module groupings used prior to darktable 3.4, as follows:

 **base modules**
The minimal set of modules normally required to render a presentable image.

 **tone modules**
Other modules related to tone levels and contrast.

 **color modules**
Modules related to color grading and color profiles.

 **corrective modules**
Modules related to correcting problems such as lens distortions, sensor noise, sharpening, etc.

 **(special) effects modules**
“Special effect” modules such as bloom, diffuse or sharpen, etc.

workflow: scene-referred & *workflow: display-referred*

These presets define groups of modules relevant to the scene-referred and display-referred workflows , sorted into groups as shown below:

 **base modules**
A basic set of modules to adjust the cropping/orientation, adjust the exposure, and apply tone mappings and contrast as appropriate to the workflow.

 **color modules**
Modules relating to color grading and color saturation.

 **corrective modules**

Modules relating to correcting problems relating to lens distortions, sensor noise, sharpening, retouching, etc.

 **(special) effects modules**

“Special effect” modules such as watermark, framing, vignetting, etc.

workflow: beginner

This preset provides a minimal set of modules targeted as a starting point for beginners. It is suggested that beginners start by copying this minimal preset, and add to it as they gain experience with other modules.

 **base modules**

A basic set of modules to adjust the cropping/orientation, adjust the exposure, and apply a basic tone mapping.

 **grading modules**

Modules dealing with creative tone and color grading.

 **(special) effects modules**

“Special effect” modules such as retouch, sharpen, watermark, etc.

search only

This preset does not include any module groupings. Modules may only be accessed using the search facility.

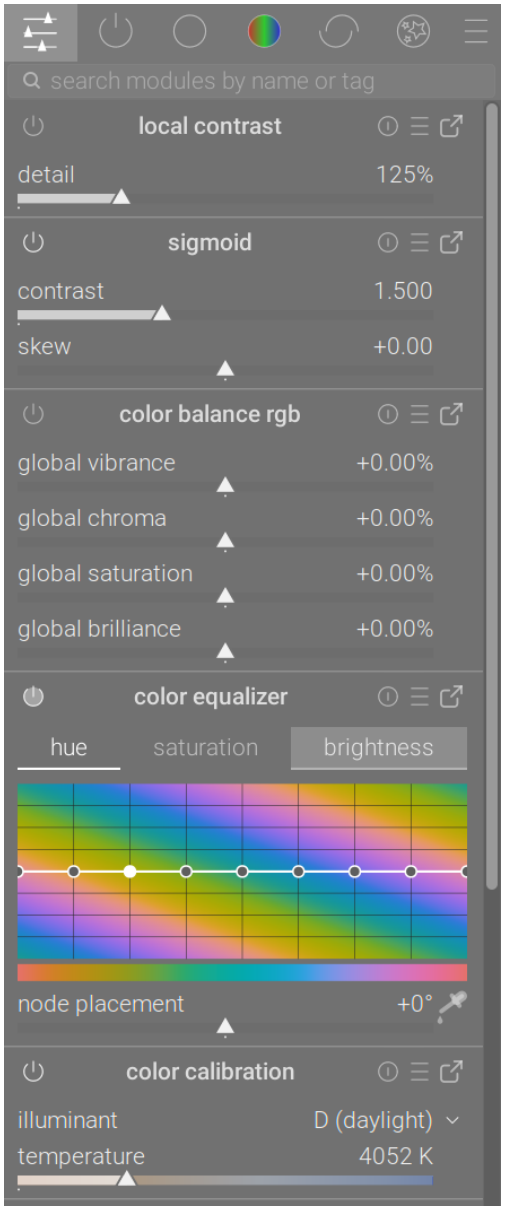
modules: deprecated

This preset contains a list of deprecated modules. This is the only way to access deprecated modules for new edits but be warned: these modules will be removed for new edits in the next release of darktable. This group cannot be duplicated and the modules within it cannot be added to user-created groups.

Note: Deprecated modules are assigned to this group only temporarily to support a smooth transition to their replacements. After the deprecation period (usually one year), these modules can no longer be used for new edits and are removed from this group. If the group then becomes empty, it will no longer be shown in the presets list.

3.6.3. quick access panel

The quick access panel allows you to access widgets from a number of processing modules all in one place.



The initial set of modules (and widgets) in the panel is determined by the *auto-apply pixel workflow defaults* setting in processing preferences .

You can add or remove widgets in the panel by:

- right-clicking on the .
- right-clicking on the module name.
- using the manage module layouts screen.
- Ctrl+clicking on a widget in a processing module while in visual shortcut mapping mode .

Click on the icon to the left of the module name to enable/disable the module.

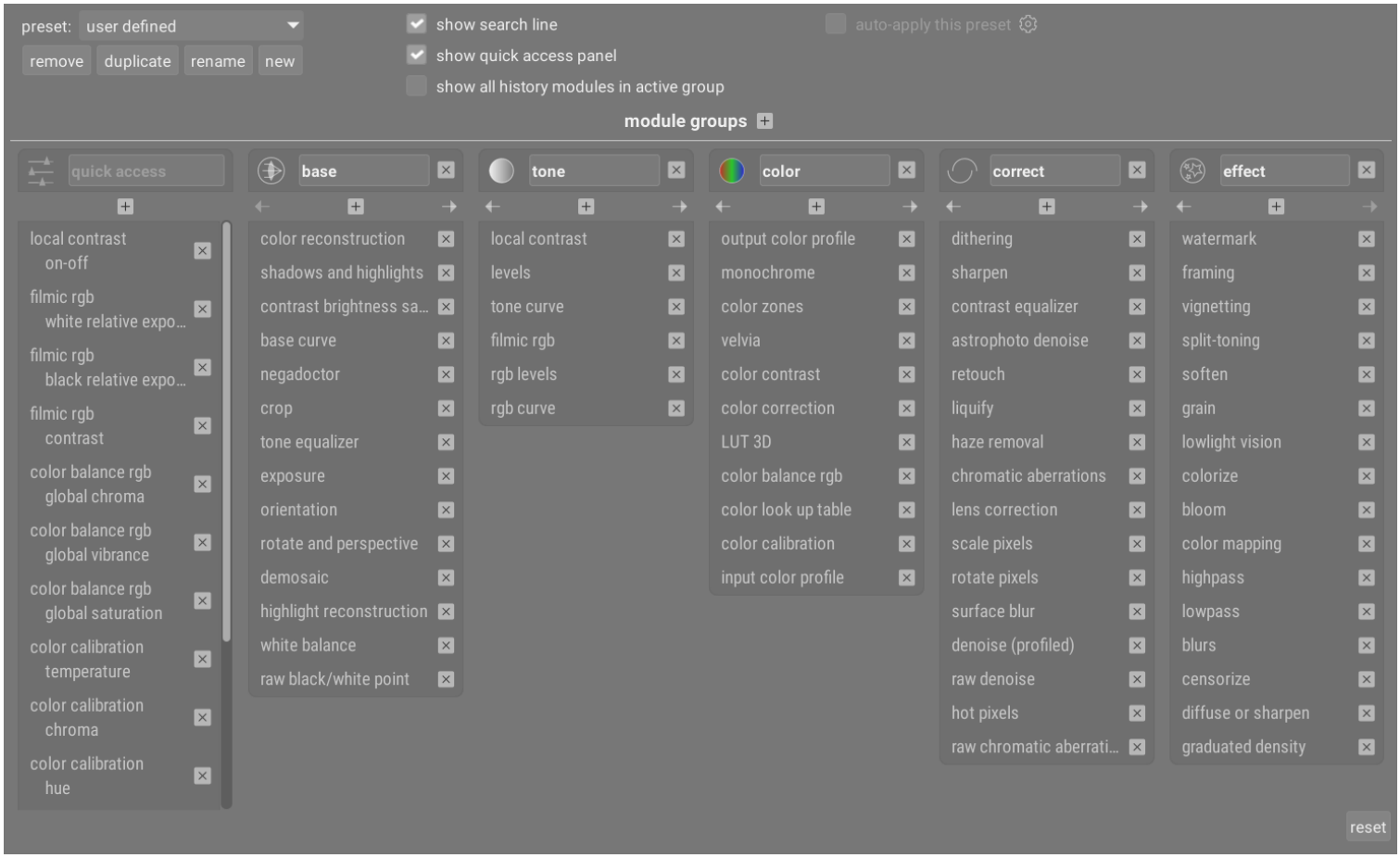
Icons to the right of the module name can be used to:

- reset the state of the module back to its original condition.
- open the presets menu of the module.
- open the full version of the module.

If any module has multiple instances enabled you will no longer be able to manage that module's widgets through the quick access panel and will have to use the full module interface.

3.6.4. manage module layouts

Manage the layout and grouping of processing modules and the quick access panel.



This maintenance screen can be accessed from the *presets* menu beside the module search box or module group icons (below the scopes module in the darkroom view). Ctrl+click on the preset menu to open this screen directly.

Settings are automatically saved when you exit the screen. Click *reset* to abandon any changes made in the current editing session.

module controls

global controls

The following global controls are available in the top panel of the screen.

preset
Select an existing module group preset.

remove
Remove the current preset (user-defined presets only).

duplicate
Duplicate the current preset with a new name. The above example shows a new preset named “user defined”, which has been created by duplicating the “modules: default” preset.

rename
Rename the current preset (user-defined presets only). Right-click to bring up a menu that can be used to copy, paste, select all, delete or insert an emoji).

new
Create a new preset containing a minimal list of modules.

show search line
Choose whether to display the search bar below the module group icons

show quick access panel
Choose whether to display the quick access panel. If checked, a new entry will appear in the bottom panel to allow you to add or remove widgets.

show all history modules in active group
Select this to show all modules that are present in the history stack within the active group, regardless of whether or not they are actually active.

auto-apply this preset
Module group presets can be automatically applied based on the type of image being worked on. The check box indicates whether this preset currently has any auto-apply rules. Click on the gear icon to amend the auto-apply settings. See presets for details.

module groups

The bottom panel of the screen allows you to alter the quick access panel and module groups for the selected preset (user-defined presets only).

add a group
Click on the + sign beside the “module groups” label to add a new group.

remove a group
Remove a group by clicking on the X button beside the group name.

add a module/widget
Add a module to a group, or a widget to the quick access panel, by clicking the + sign below the group name. Select the required module/widget from the displayed list.

remove a module/widget
Remove a module from a group, or a widget from the quick access panel, by clicking the X beside the module/widget name.

change a group’s icon
Change the icon assigned to a group by clicking on the existing group icon and selecting a new one from the displayed list.

change the group order
Change the order in which the groups are displayed by clicking on the < and > buttons below the group names.

rename a group
Rename a module group by clicking on the group name and typing. Right-click to bring up a menu that can be used to copy, paste, select all, delete or insert an emoji)

4. Tethering

4.1. overview

The tethering view allows you to capture images directly into darktable from a connected camera.

To use the tethering feature you first need to connect your camera to your PC using a USB cable. Your computer might ask to mount or view the connected camera. *Do not mount or view the camera*. If your camera is mounted or viewed automatically, you will need to “unmount/eject” the camera before darktable can access it. This unlocks the camera so that darktable can take control of it – darktable will then re-lock the camera so that it cannot be used by other applications.

After the USB cable is connected, go to the import module in the lighttable view. If your camera is available for use, a new section should appear in the import module containing the name of your camera and a “mount camera” button. Click this button to connect your camera and three additional buttons will appear: “copy & import from camera”, “tethered shoot” and “unmount camera”. Click “tethered shoot” to enter the tethering view. When you are finished, press the “unmount camera” button before physically disconnecting it.

darktable uses gphoto2 to interface with your camera. If you have problems finding the connected camera as described above, check the troubleshooting section in this chapter to verify that your camera has tethering support.

In the center view, images are shown while you capture them. You can capture an image by either using darktable’s user interface or by manually triggering a capture with your camera. If you are using Live View the image will be shown in darktable’s center view.

When entering tethering view, a film roll will be created using the same structure as defined for camera import (see preferences > import > session options). The job code will be predefined as “capture”.

If you want to group your captures into different film rolls, you should use the session module in the right-hand panel to set a different job code. When entering a new name and pressing Enter, a new film roll will be created and newly-captured images will be added into this new film roll.

darktable provides some useful tools to set up an image capture in the user interface. You can set up timelapse captures, brackets for HDR and even sequential captures of bracketed images. For more information read the documentation on the camera settings module and the examples later in this chapter.

4.2. tethering view layout

left panel

image information
Display image information.

right panel

From top to bottom:

scopes
A graphical depiction of the current image’s light levels and colors. This module can be moved to the left-hand panel if desired (see preferences > miscellaneous > position of the scopes module).

session
Session settings.

live view
Live view settings.

camera settings
Camera settings.

metadata editor
Edit metadata for selected images.
tagging
Tag selected images.

bottom panel

From left to right:

star ratings
Apply star ratings to images.
color labels
Apply color labels to images.

4.3. examples

The following sections show two typical use cases for the tethering view.

studio setup with screening

This is a pretty common use case. You have your studio and subject set up, the camera is connected to your computer and tethering view is active in darktable. You work at the camera and take images. Whenever you like, you can review the images directly on your computer monitor instead of using the camera LCD.

This workflow is efficient and effective, since you can immediately review your captures instead of waiting until after the shoot when everyone is gone. If you're shooting with a model this is a nice way to preview the captures with the client instead of fumbling around with your camera.

Working in the tethering view can save you time and aggravation. Set a session name, shoot your images, and they will be saved to the correct film roll for the session, for easy on-site review.

capturing a timelapse

A timelapse is a video clip composed of images taken in a timed sequence. A typical example is to take a timelapse of a cityscape to capture the movement of the clouds and traffic etc.

To set up a timelapse capture, create a new session . Now decide whether you want to shoot in manual or auto mode. It is recommended that you only use auto mode in situations where the ambient light will change significantly during the time of the shoot, for example, when shooting a timelapse over 24 hours.

The camera settings module can be used to define the delay (the number of seconds between capture) and sequence (the number of images to capture) of your timelapse.

To start the timelapse click the capture button in the same panel and watch the filmstrip fill up with images. The last captured image is always displayed in the center view.

storing images on the camera and computer

By default, the gphoto2 framework (used by darktable's tethering feature) will only download images to your computer, and will not store them on the camera's memory card. This setting can be changed outside of darktable by using the `gphoto2` command line interface. However, there are cases where this method fails, as it requires that darktable use the same configuration file as the gphoto2 command line tool. In particular, if darktable or gphoto2 are installed within a sandbox or container that hides user account settings, you may face this issue (for example, with snap packages and similar).

To allow captured images to be retained on the camera's memory card, connect the camera to your computer for a tethering session, but close darktable. Enter `gphoto2 --set-config capturetarget=1` on the command line. If this command is successful, start darktable again. Thereafter, images should be stored (duplicated) on the camera's memory card during a tethered capture.

4.4. troubleshooting

This troubleshooting guide can be used to verify whether or not your camera can be used with tethering. This is done using the same (gphoto2) tool that darktable uses to interface with your camera.

Before you start, you first need to find your camera port name. Usually the port “`usb:`” is sufficient and is therefore used in the following guide.

is your camera detected?

The following command will verify that your camera is connected to the computer and detected by gphoto2.

```
env LANG=C gphoto2 --auto-detect
```

check the camera’s driver abilities

Execute the following command and ensure that the “capture choices” ability supports “Image” and configuration support is “yes” – darktable will check these two abilities to decide whether to show the “tethered shoot” button.

```
env LANG=C gphoto2 --port usb: --abilities
```

remote capture

This step will verify that your camera can be remotely controlled – i.e. that it can capture an image, download it to your computer and display it within darktable.

```
env LANG=C gphoto2 --port usb: --capture-image-and-download
```

tethered capture

This last step tests if your camera supports “events”, are heavily used by darktable. Running this command will make the gphoto2 process wait for an image capture event which you must manually trigger on your camera. If successful, the image will be downloaded to your computer.

```
env LANG=C gphoto2 --port usb: --capture-tethered
```

now what?

If any of the above steps failed, there are problems with your specific camera and driver. Please file an issue on the gphoto2 github page . Add the following flags to the failed command for better support and attach the log output to your issue:

```
--debug --debug-file gphoto2_debug.log
```

If you successfully completed all of the above tests, your camera is probably supported by darktable. If these tests were successful but you nevertheless stumble upon a problem in darktable, please file an issue on the darktable github page . Please attach the log output from the commands above and the log file output produced by starting darktable with the following command:

```
darktable -d camctl 2>&1 >camctl.log
```

5. Map

5.1. overview

The map view allows you to see where your geo-tagged images were taken, and to add location information to non-geo-tagged images.

The map view shows a world map with the images in the current collection pinned to their geo-tagged location (if available). This requires that images are tagged with location information. Some newer cameras, including smartphones, are already equipped with GPS receivers. Other cameras may need additional GPS hardware to do this.

Even if your camera doesn't support this feature, there is an alternative method – darktable can match the Exif date/time in your image(s) to a separate GPX data tracking file created by a GPS tracker, that has recorded your movements. GPS trackers can be purchased as standalone handheld devices or you can install a GPS tracker app on your smartphone. Location tagging with GPS tracking data can be done using the geotagging module, in the lighttable and map views.

center map view

In the center of the map view you will see a world map.

Map data is taken from open map sources on the internet. As such, new map data is only available if you are connected to the internet – darktable keeps a disk cache of previously loaded map data.

You can navigate within the map using your mouse. Left-click and drag to move the map. Use the scroll-wheel to zoom in and out.

On-screen controls and displays are available to help you find your way. A navigation area is located on top left of the map – use this as an alternative to mouse-dragging and scrolling. The scale of the map is displayed at the bottom-left. At the bottom-right you can see the geographical coordinates for the center of the map.

Images that already have geo-location attributes in their metadata are displayed as small icons on the map. Images close to each other are grouped and a count of grouped images is displayed on the bottom-left corner.

In order to assign geo coordinates to an image, activate the filmstrip on the lower panel (press Ctrl+F). You can assign a geo location to an image by dragging the image icon from the film-strip and positioning it on the map – darktable will record the new location (latitude and longitude) as part of the image metadata. This data will be included in exported images.

In order to remove location data from an image simply drag it from the map and drop it onto the filmstrip.

Images close to each other are grouped under a single image group. You can use the map settings module to control the grouping as needed. The number displayed on the bottom left of the thumbnail is the number of images inside the group. A white number means that all images in the group are at exactly the same location, whereas a yellow number means they are not. Use the mouse scroll wheel while hovering over a group of images to scroll through the thumbnails of the images in that group.

Normally, images in the center map view have black borders. If an image is selected in the filmstrip, then the corresponding image on the map will be highlighted with a white border.

Click+drag to adjust the location of an image. Shift+click to move a complete group of images.

The left and right panels provide additional controls (see map view layout).

undo/redo

All image movements in the map view are recorded by darktable. It is possible to undo or redo such changes to recover a previous state. Note that this undo/redo facility is unlimited while moving images but it is reset each time you leave the map view.

Press Ctrl+Z to undo the last modification and Ctrl+Y to redo the last undone modification (if any).

5.2. map view layout

left panel

These modules are identical to the lighttable view. From top to bottom:

collections
Filter the list of images displayed in the map view.
recently used collections
View recently used collections of images.
image information
Display image information.

right panel

Here you can find the modules specific to the map view. From top to bottom:

find location
Search for a place on map.
locations
Manage a hierarchical list of location tags and their corresponding regions on the map.
map settings
Choose the map provider and set up various map display parameters.
tagging
Tag selected images.
geotagging
Apply GPX track data to selected images.

bottom panel

filmstrip
Drag images from the filmstrip onto the map as described in the map overview .

6. Slideshow

6.1. overview

The slideshow view allows you to watch a slideshow of your current collection with the associated filtering rules and sort order applied.

To learn more about how to define the collection and filtering rules, see the section on collections . Select the sort criteria and sort order of the images in the top panel .

The next section provides more details on the usage of the slideshow view.

6.2. usage

The slideshow view is still in an early stage of development with only a basic set of features.

If you don’t need the auto-advance mode, you could even use the sticky preview feature instead.

spacebar	start and stop auto-advance mode which automatically switches to the next images every five seconds by default.
ESC	leave slideshow mode and return to lighttable view.
+ or up arrow	increase delay between each image.
- or down arrow	decrease delay between each image.
left-click or right arrow or right shift-key	switch to the next image of the collection.
right-click or left arrow or left shift-key	switch to the previous image of the collection.

Hint: To take full advantage of your screen size, put darktable into fullscreen mode by pressing F11 and hide the border-controls by pressing the B key.

---

7. Print

7.1. overview

This view allows you to print your images. Because printing is not easy, there are many technical aspects to be taken into account.

After selecting an image in the lighttable view you can enter the print settings module to adjust printer settings and initiate printing.

This module supports the printer's ICC profile, which is somewhat mandatory if you want to obtain a high quality print close to the image displayed on the screen.

It is important to note that ICC profiles provided by the paper and/or printer manufacturers cannot be used on GNU/Linux as they are printer-driver dependent. On Linux and macOS, the darktable print module uses CUPS and there are no ready-to-use ICC profiles available for this driver.

On Windows, paper specific ICC profiles from paper manufacturers should work as expected. Always run a test print to confirm output is as desired.

7.2. print view layout

The central area displays the image layout on the paper (the white area). Some gray borders may be displayed around the image to represent the printable area (the page minus the borders) not filled by the image.

The filmstrip below the image allows you to select more images.

overlays

When the mouse is over the bounding box of an image, its width and height are shown along its top and left, respectively. Margins between the bounding box and the page edge are notated next to the dotted lines extending out from each side of the bounding box. All measurements are shown in the units as chosen in the print settings module.

Images are inset along one dimension of their bounding box when they do not match the aspect ratio of the box. The overlaid margin measurements should therefore only be used to understand the layout bounds, not the actual printed size of the image.

left panel

collections
Filter the list of images displayed in the lighttable.

image information
Display image information

right panel

print settings
Adjust print settings and initiate printing.

bottom panel

filmstrip
Select images to be printed from the filmstrip.

8. Module Reference

8.1. overview

The modules in this reference section are broken down into two distinct types:

processing modules
: Processing modules are used exclusively in the darkroom view. Each module performs a processing operation on the image before passing its output to the next module for further processing. Together this sequence of processing steps forms the pixelpipe .

  Processing modules have a *Technical information* section which gives a brief summary of the module (the information can also be seen as a tooltip when hovering over the module name). The following information is shown:

  - a brief outline of the purpose of the module
  - an indication of the purpose (e.g. corrective or creative)
  - the processing color space (e.g. non-linear, Lab)
  - information about input and output:
    - color space (e.g. linear, RGB)
    - display referred or scene referred workflow

utility modules
: Utility modules may be used in any darktable view. They are not directly involved in processing the pixels of an image but perform other ancillary functions (managing image metadata and tags, editing history, modifying pixel pipeline order, snapshots and duplicates, image export etc.).

The two types of modules have a few aspects in common, described below.

module header

Each module has a header at the top, normally consisting of the following elements:

module name
: Click on the name to expand or collapse the rest of the module and show/hide its controls.

reset parameters button
: This normally appears to the right of the module name and is used to reset the state of the module back to its original condition.

presets menu
: This normally appears at the far right of the module header. The presets menu is predominantly used in processing modules, but many of the utility modules allow presets to be defined as well. You can also access this menu by right-clicking anywhere on the module header.

Processing modules contain additional elements in their module header, as described in the processing module header section.

module resizing

utility modules

Some utility modules contain lists of information that can grow as more entries are added. To help manage screen real-estate, it is possible to increase or decrease the maximum number of entries that can be displayed before a scroll bar is added. To alter the maximum number of entries, place the mouse over an entry in the list, and hold Shift+Alt while scrolling your mouse wheel. If the list currently contains more entries than this maximum, a scrollbar will appear so that you can access the hidden entries. Alternatively you can click+drag the bottom of the scrollable area with your mouse.

Note: It is not possible to extend these areas beyond the number of entries currently shown. If you attempt to do so using Shift+Alt+scroll, the maximum number of entries *will* increase, and a toast message will appear informing you of the new maximum. However, the module itself will not be resized unless its content exceeds this maximum.

processing modules

Some processing modules contain drawn graphical elements that can take up too much or too little screen space depending on the width of your side panels. These drawing areas usually default to a 16:9 aspect ratio and can be similarly resized by hovering over them and holding Ctrl while scrolling.

visibility and ordering

utility modules

Right-click in a side panel (outside of any module) to list the available utility modules and toggle their visibility. Drag-and-drop a module to change its position. Select preferences > general > reset view panels to reset to default values.

processing modules

Please see mange module layouts for information about how to organize processing modules. Please note that you cannot reorder them without changing how an image is processed. See the pixelpipe and module order for more information.

8.2. processing modules

8.2.1. AgX

Technical information

description
applies a tone mapping curve. inspired by Blender’s AgX tone mapper.
purpose
corrective and creative.
input
linear, RGB, scene-referred.
processing
non-linear, RGB.
output
linear, RGB, display-referred.

Transform the colors and tonal range of an image to fit a display.

The primary innovation of the AgX module is an advanced color handling method that ensures a natural and film-like appearance, preventing the oversaturation artifacts common with simpler tone mapping techniques. It is derived from the display transform of the same name, used in the Blender 3D modeler by T. J. Sobotka, Eary_Chow, Sakari Kapanen and others.

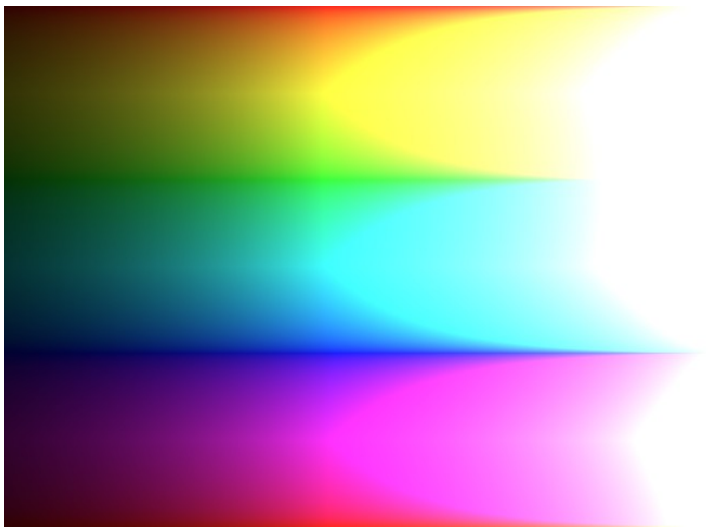
Modules placed before *AgX* in the pipeline operate in scene-referred space. Modules after *AgX* work in display-referred space.

You can automatically enable *AgX* in new images by setting the value of preferences > processing > auto-apply pixel workflow defaults to ‘scene-referred (AgX)’.

primaries

The “primaries” are the defining feature of AgX and the core of how it handles color. They determine the basic color appearance of the image *before* the tone mapping curve is applied. For precise definitions of the color terms used in this section, please refer to *darktable’s color dimensions* .

The core challenge in fitting a scene’s wide dynamic range into a display’s narrower range is managing color across the entire tonal scale. Without tone mapping, transitions would be abrupt, and lead to color distortions:



A simple per-channel curve often causes colorful objects to shift to pure, unnatural-looking primary or secondary colors as their intensity changes – an effect sometimes called the “Notorious 6”:



The primaries controls in *AgX* work to prevent this by building a custom color space for the tone curve. By adjusting the primaries *before* the curve is applied, *AgX* creates a more graceful “path to white,” allowing colors to desaturate and shift hue in a way that looks more natural and believable. This process influences the color rendering across the entire tonal range to create a cohesive final image:



usage

Please take note of the following guidelines while using this module within your workflow:

only use one tone mapper
AgX is a tone mapper ; you typically have exactly one tone mapper enabled for an image.

adjust for the mid-tones first
By default, the module preserves middle gray. Before using *AgX*, you should first use the *exposure* module to adjust the mid-tones to your liking.

A recommended workflow for using this module is provided at the end of this page.

module controls

The module's controls are divided into three categories:

- Tone mapping controls (input range and the curve)
- Post tone-mapping adjustments ("look")
- Color-related controls (primaries)

Throughout this module's documentation, references will be made to other tone mappers, namely *filmic rgb* and *sigmoid* . This is to support users already familiar with those modules, however, familiarity with these modules is not required to use *AgX*.

Due to the complex nature of this module, related controls are grouped together in collapsible sections, and the controls are distributed over two tabs.

the settings tab

The settings tab holds the most commonly-used controls.

input exposure range

This section provides controls similar to those of the *filmic rgb* module, allowing you to set the black and white points. The selected exposure range will then be projected into the [0, 1] range using a logarithmic transformation. This means that each 1 EV stop of dynamic range corresponds to an equal distance along the curve's horizontal axis. The mid-gray point is situated proportionally on this axis between the selected extremes. After this initial projection, the data will be further processed by the curve (below).

white relative exposure
Set the white point. Any value above this will be clipped to 1. Use the picker to automatically set this value.

black relative exposure
Set the black point. Any value below this will be clipped to 0. Use the picker to automatically set this value.

dynamic range scaling

Apply a safety margin when using pickers to select the white and black relative exposure.

auto tune levels

Select the auto tune picker icon to set both the white and black relative exposures at the same time (same as clicking both pickers above).

The *read exposure* (camera icon) button does not analyze the contents of the image, like the pickers do. Instead, it estimates the black and white relative exposures based on the settings in the *exposure* module. If there are several instances of *exposure*, the button will read the settings from the first enabled, unmasked instance. If all instances are masked, the first instance will be used.

Using the button only makes sense if the input to *AgX* is actually influenced by the exposure module – that is, if the *exposure* module comes before *AgX* in the pipeline (which is normally the case). This allows you to take into account any in-camera exposure compensation, in-camera highlight preservation (if supported by darktable), and manual adjustments. The mechanism is similar to that used by *filmic rgb*, but is not applied automatically.

basic curve parameters

In the following discussion, the parts of the tone curve are identified as follows:

- The *pivot* is the point on the curve around which contrast is adjusted (the point of highest contrast)
- The *shoulder* is the higher bend of the curve (the highlights)
- The *toe* is the lower bend of the curve (the shadows)
- The *contrast* is the slope of the curve (steeper curve = higher contrast)

If you do not understand these terms from the definitions, it may be helpful to show the curve while adjusting the related controls and observe the effect.

pivot relative exposure

Sets the input value of the pivot point (the point in the tonal range where the contrast will be highest), in EV relative to mid-gray. Since contrast is normally highest around the pivot, this slider allows you to choose which part of the input tonal range should have the most contrast. The corresponding color picker adjusts only this slider, but not *pivot target output* (below). You may be tempted to use it as a brightness control (moving the pivot towards black, without modifying its output value, will brighten the image, while moving it towards white will darken the image), but the *pivot target output* is better suited for that, as it directly influences brightness, without changing the point of highest contrast.

When adjusting the *black/white relative exposure* values, the *pivot relative exposure* will be maintained for as long as possible. This will cause it to move along the x-axis, as the preserved quantity is the exposure value *relative to mid-gray*, and its position on the x-axis depends on the endpoints of that axis (the black and white relative exposure values). If an adjustment would cause the pivot input to fall outside the exposure boundaries, it will be forced inside the available range.

pivot target output

Sets the target output linear value (power) for the tone selected by the pivot's input. The value is indicated on the y-axis of the curve. Note that the scale of the y-axis is not uniform, and depends on the *curve y gamma*, described in the section advanced curve parameters . The corresponding color picker adjusts both *pivot relative exposure* (the point of highest contrast) and *pivot target output*, attempting to preserve the average brightness of the selected region.

For more predictable results, you are advised to use the color pickers on uniform areas, as sudden tonal transitions in the selected area will cause the average values to change drastically, leading to fluctuations in both pivot input and output.

contrast

Sets the slope of the curve at the pivot point. This value is scaled internally to maintain a consistent final (linear) output contrast, compensating for changes to both the dynamic range and the *curve y gamma* setting.

shoulder power / toe power

These sliders determine how fast the contrast drops as the curve approaches black or white. Higher values result in a sharper bend, maintaining contrast for longer before a more abrupt roll-off. If the overall contrast is not sufficient to reach the black and/or white point, either or both ends of the curve may become "inverted," rendering these controls ineffective. Should this occur, a warning icon will appear next to the affected slider(s). Hovering over the warning provides a tooltip with suggested actions, and, if the *show curve* section is expanded (below), the affected part of the curve will be highlighted in yellow.

show curve

Expand the *show curve* section to display a plot of the curve. This can be a useful tool to learn about curve behavior by observing the effect of related controls, or when investigating issues such as loss of detail or “curve inversion” (see the *toe power / shoulder power* control above). The plot is not interactive – it simply illustrates the effect of the sliders.

The x-axis shows the selected input exposure range, measured in EV, with all values relative to mid-gray (indicated by the 0 EV mark).

The y-axis shows the linear output value, with 18% representing mid-gray. The scaling of the y-axis is not linear – horizontal grid lines help visualize this non-linearity. The degree of non-linearity is governed by a gamma value (default: 2.2). More information on the gamma is provided in the description of the advanced curve parameters .

The curve has 5 important points:

- The *black and white points* are at the left and right edges of the graph, respectively; their final linear output values can be controlled by the *target black* and *target white* sliders (see advanced curve parameters).
- The *pivot* is the point around which the curve is built, indicated by a dot. By default, it maps mid-gray to mid-gray. You may move this point to match your main subject using the provided picker.
- The *toe and shoulder starting points* define where the curve transitions from a linear section to the compressed toe (shadows) and shoulder (highlights). By default, these are set to the pivot point.

advanced curve parameters

shoulder start
Defines the point where the linear portion of the curve ends and the shoulder begins. Keeping the value at 0% allows the smooth transition to start at the pivot. Higher values push the transition point up towards the chosen *target white*. This may result in hard clipping, which can lead to a loss of detail in the highlights.

target white
The upper bound that the curve converges to, relative to the peak of the output. This can be used to limit the maximum output luminance. For SDR output the peak is display white; for HDR output it is the peak of the export (see HDR output).

follow output peak
This checkbox is only shown for edits made before darktable supported HDR output, and only while it is unticked. Such edits keep their white at the 203 nit reference white, so they render exactly as before, also when exported as PQ or HLG. Tick it to let the curve extend the highlights to the peak of an HDR output. SDR output does not change. Once ticked, the checkbox is hidden; new instances of the module always follow the output peak.

HDR purity
For HDR output, how much of the extra saturation and hue shift that comes from extending the highlights to the peak is kept (default 50%). This control has no effect on SDR output, and is hidden for edits that do not follow the output peak.

toe start
Defines the point where the linear portion of the curve ends and the toe begins. Keeping the value at 0% allows the smooth transition to start at the pivot. Higher values push the transition point down towards the chosen *target black*. This may result in hard clipping, which can lead to a loss of detail in the shadows.

target black
The lower bound that the curve converges to. This can be used to create a faded analog look, similar to the *lift* control in the *look* section or the *global offset* in *color balance rgb* .

keep the pivot on the diagonal
Automatically adjusts *curve y gamma* in an attempt to ensure the curve remains S-shaped, as long as the contrast setting is high enough. It works much like *auto adjust hardness* in *filmic rgb*. While the checkbox is ticked, *curve y gamma* cannot be adjusted manually, and is hidden.

curve y gamma
Shifts the representation of the pivot along the y-axis without changing its output luminance. This is an internal parameter of the algorithm and does not need to match your display’s gamma. Its purpose is mainly to enable you to keep the S-shape of the curve, thereby ensuring that the *shoulder power* and *toe power* controls remain effective. Manual adjustments are disabled while *keep the pivot on the diagonal* is enabled.

look

The “look” controls define how the image is processed after the tone mapping operation. This means that they are *display-referred* operations, and can result in clipping, so use them carefully.

slope
Multiplies the output values by this factor. Black is not affected (since it has a value of 0). Values above 1 brighten the image and increase contrast, whereas those below 1 darken the image and reduce contrast.

lift
Shifts values up or down. Only blacks are fully affected, with the effect gradually reduced for brighter tones. Whites are not affected if *slope* is at its default value of 1. Negative values crush shadows, whereas positive values can produce a faded look.

brightness
Applies a power function (gamma adjustment) to the image. The black and white points are not affected. Values above 1 will raise the mid-tones, opening up shadows, whereas those below 1 will darken the mid-tones and compress shadows.

saturation
Controls color intensity by adjusting the image's chroma. For a precise definition of chroma vs. saturation, please see *darktable's color dimensions* . A value of zero makes the image monochrome.

preserve hue
The tone mapping curve, being a per-channel curve, introduces color shifts, with color tending towards the primary (red, green and blue) and secondary (yellow, cyan and magenta) colors in the highlights. At a value of 0%, these color shifts are retained. By raising this slider, the input hues (those before the tone curve) can be partially or fully restored.

Note that the input hues themselves are affected by the "primaries manipulations" that are performed before tone mapping , and the final hues are affected by the "primaries manipulations" that are applied after tone mapping . For a detailed description of the order of the operations, see technical details below.

the primaries tab

The *primaries* tab controls are similar to, but more extensive than, those of the collapsible *primaries* section in the *sigmoid* module:

disable adjustments
Turns off all manipulation of primaries. You are not advised to tick this checkbox for actual processing – it is intended more as a learning tool for quick comparisons. For a starting point without adjustments that can be fine-tuned manually, use the *unmodified* configuration.

reset primaries
Clicking this button reveals a pop-up menu to apply one of the built-in sets of primaries, without altering any of the other settings.

base primaries
Defines which color space is used as the basis of the *AgX* processing space. The attenuation and rotation controls below are applied relative to this space. For especially problematic colors, you may find that wider spaces provide better control.

before tone mapping

The controls in this group affect the operations performed before the tone mapping curve is applied.

red/green/blue attenuation
Controls the amount of desaturation applied to the primary color. This adjusts the rate of the color's shift towards white as its intensity increases. Lower values result in a slower shift and more pronounced hue changes, at the risk of artifacts. The saturation of all colors will be reduced, regardless of luminosity, but this can be reversed for shadows and mid-tones using the *purity boost* sliders in the after tone mapping section, the overall effect being a desaturation of highlights.

red/green/blue rotation
Rotates the hue angle of the primary color. This affects the direction of the hue shift for colors as their intensity changes. For example, rotating red can influence whether it bends towards yellow or magenta.

after tone mapping

The controls in this group affect the processing performed after tone mapping. Use them as artistic controls to help you achieve the desired final look.

reverse all

Ticking this checkbox will hide the other controls in the group, and adjust processing settings as if the *master* reversal sliders, discussed below, were set to 100%, the *purity boost* parameters to match the values of the corresponding *attenuation* sliders, and the *reverse rotation* parameters to mirror the values set with the *rotation* sliders. The values of the hidden sliders will be retained, but not used for processing. Unchecking the checkbox will make them visible, and their settings effective, again.

The checkbox is intended to simplify the user interface (users may wish to create presets with the checkbox ticked), and for quick “what-if” checks.

set from above

Clicking the button will *copy* the values of the sliders from the before tone mapping section, and set the *master* controls to 100%, producing the same visual effect as ticking *reverse all*. However, this only provides a starting point – the sliders will not be hidden, and may be adjusted to taste.

master purity boost, master rotation reversal

These are multipliers that affect the individual red/green/blue controls below, allowing you to globally increase or decrease their effect.

red/green/blue purity boost

Restores color purity *after* the tone curve is applied. Higher values make the image look more colorful and chroma-laden, but can introduce artifacts if pushed too far. When combined with *attenuation* (above), the net effect is a selective desaturation of highlights (since the purity boost does not fully recover purity in highlights, due to them being strongly desaturated by the tone mapping process).

red/green/blue reverse rotation

Reverses the initial primary rotation after the tone curve. This offers a final control over the rendered hues. This can be used to partly or fully reverse the hue shifts introduced by the corresponding rotation sliders, mostly in the shadows and mid-tones. Since the curve is applied per-channel, it is subject to the “Notorious 6” shifts, which mostly affect highlights.

presets

The module provides built-in presets, based on Blender, and on the *smooth* preset of the *sigmoid* module. When adjusting primaries starting from those presets, you will need to take the following into account:

- In Blender, the rotation of primaries is not reversed, and purity is restored using values different from those used for attenuation. Therefore, the corresponding *blender-like* and *scene-referred default* presets of *AgX* come with *master purity boost = 100%, master rotation reversal = 0%*. This means that, although the per-channel rotation reversal sliders are set up for complete reversal, they have no effect. You can turn them on gradually by raising the *master rotation reversal* control, and, once that is no longer set to 0%, adjust the individual channel sliders according to taste.
- In the *sigmoid* module’s *smooth* preset, rotations are always completely reversed, but purity is not boosted at all by default. Therefore, the corresponding *AgX* presets, also called *smooth*, set *master purity boost = 0%, master rotation reversal = 100%*. The per-channel purity boost sliders are set to completely reverse the attenuation, but are effectively disabled by the master slider. The per-channel rotation reversal sliders mirror the settings of the reversal slider, and are fully enabled by the master control.

You are, of course, free to create your own presets with your favorite combination of master and per-channel controls (for example, having the master reversal controls at 100% and the per-channel controls set to your preferred values).

selective tuning for mid-tones and highlights

It is important to note that using the primaries controls, the properties of the S-curve (lower contrast in highlights), and per-channel application of the curve together make it possible to apply different adjustments to mid-tones and highlights, even though this may not be evident at the first glance.

Per-channel S-curves have the property of desaturating highlights, independently of the *attenuation* control. This means you *can* reverse the attenuation (restore saturation) for mid-tones using *purity boost*, but not so for highlights.

Another property of per-channel S-curves is the skewing of bright colors mentioned above: the “Notorious 6”. *AgX* cleverly uses this effect to exaggerate hue shifts introduced using the rotation of primaries. For example, a red primary rotated a few degrees towards green (yellow) will turn ever so slightly yellow, and this can be reversed later using the *red reverse rotation* slider, if desired. However, for brighter tones, the “Notorious 6” shift will skew the slightly yellow-tinted red more towards yellow, allowing you to obtain orange sunsets instead of pink ones. The reverse rotation will not recover this additional skew fully. To control how much *additional* skew you want to allow, use the *preserve hue* slider in the look section.

HDR output

When an image is exported to an HDR format – an output profile with a PQ or HLG transfer function, such as *PQ Rec2020* or *HLG Rec2020* – *AgX* adapts automatically, so the same edit can be exported to SDR and HDR without changes. darktable places the diffuse "reference" white of HDR output at 203 nits, and the room above it is called the *headroom* of the output:

- for PQ, the peak is set by the *PQ peak* setting of the *export* module (default 1000 nits, a headroom of about 4.9×)
- for HLG, the peak is always 1000 nits
- SDR output has no headroom, and is rendered exactly as before

With HDR output, *AgX* follows the HDR design of Blender's AgX: shadows, mid-tones and the pivot keep the brightness they have in SDR, and only the highlights are extended up to the peak of the output. The *target white* is relative to that peak. The *HDR purity* slider controls how saturated those extended highlights are.

The curve plot and the curve warnings use the headroom of the darkroom's own output, so they show the HDR curve when your display profile is PQ or HLG. The picker for *pivot target output* always works on the SDR curve.

Edits made before darktable supported HDR output do not follow the output peak, and keep their white at 203 nits. To let such an edit extend its highlights to the peak of HDR output as well, tick its *follow output peak* checkbox.

recommended workflow

The following workflow is recommended when using this module:

1. Set overall exposure for the mid-tones using the *exposure* module. Alternatively, use *contrast* and *toe / shoulder power* to fill the output tonal range.
2. Choose a preset as a starting point. If darktable is set to apply a scene-referred workflow by default (see preferences > processing > auto-apply pixel workflow defaults), the module will apply reasonable defaults automatically. In other cases (for example, when using the legacy display-referred workflow or if the workflow preference is set to "none"), manually select the preset *blender-like|base* or *smooth|base* for best results. The *punchy* presets are more contrasty and colorful.
3. Use the *auto tune levels* picker to set the desired exposure range.
4. Set the pivot on the subject using the picker next to *pivot relative exposure* (optional) – this ensures that contrast is maximized around the selected area.
5. Move the *pivot target output* slider to adjust the brightness of the pivot point (optional).
6. Set the *contrast*.
7. Adjust *toe* and *shoulder power* to set contrast in shadows and highlights, respectively (optional).
8. Finally, if desired, you may add "drama" by adjusting the controls in the look section – add *brightness*, set overall *saturation*, and adjust colors with *preserve hue*.

Please also note the following:

- The provided presets (with the exception of *unmodified base primaries*) provide carefully tuned primaries. When starting out, it is best to rely on those settings. Adjusting them, based on individual taste and/or special lighting conditions (e.g. LED lights, stage lighting) is part of a more advanced workflow.
- Unless high quality processing is enabled, the effect of adjusting the primaries cannot be judged properly, especially with highly saturated colors and narrow-spectrum light sources like LEDs.
- Don't forget that the *AgX* module is just another tone mapper. As flexible as it is, it is not intended to solve all image processing tasks related to color and contrast, many of which will be better resolved by using other modules.

additional options

The following additional options are "hidden" by default but can be toggled by manually editing your `darktablerc` file while darktable is closed.

"3 tab" mode

For people with small screens, where vertical screen-space is at a premium, an optional "3 tab" mode is provided in order to reduce the need to scroll the module. This mode moves the plot of the curve and advanced curve parameters from the *settings* tab into a dedicated *curve* tab, and duplicates some other controls from the settings tab for convenience.

You can enable this mode by setting `plugins/darkroom/agx/enable_curve_tab=TRUE` in `darktablerc`. When set to `FALSE`, all curve controls appear on the *settings* tab as normal.

disabling curve warnings

The warnings documented within the basic curve parameters section can be disabled by setting `plugins/darkroom/agx/enable_curve_warnings=FALSE` in `darktablerc`.

always expand the “look” section

By default, the *look* controls are placed inside a collapsible section . If you want to keep them visible at all times, you can set `plugins/darkroom/agx/look_always_visible=TRUE` in `darktablerc`.

---

technical details

The following is a detailed description of the steps taken when processing with the *AgX* module. This section is provided as a reference for interested readers, but is not required to use the module.

1. Linear, scene-referred data arrives in the pipe’s working space, defined in the *input color profile* module (e.g. Rec 2020)
2. If needed, data is converted to the selected “base” space
3. Colors that are out-of-gamut for the “base” space (RGB triplets that have any negative component in that space) are “slid” into gamut (in-gamut colors are not modified, though it is possible that a previously out-of-gamut color “lands on top of” an in-gamut color). The *sigmoid* module performs a similar operation, but the method is different.
4. An input matrix is calculated and applied, based on the RGB attenuation and primaries rotation values. This desaturates the colors (mixing channels into each-other) and rotates the primaries. This is the same as with *sigmoid*.
5. The HSV representation is calculated from the inset+rotated RGB. The hue (H) will be used later to restore the hue according to the *look / preserve hue* slider.
6. For each component (RGB channel), the same processing steps are executed independently (but, since the desaturation (primaries attenuation) mixes channels into each-other, this means that with respect to the original color, the processing is not independent at all):
    - Log encoding is applied: for input x, log2(x) is calculated (*x* would be the value of the red, green or blue channel)
    - The log value is scaled and shifted according to the selected exposure parameters, so the selected black point becomes 0, the white point 1, and the value corresponding to mid-gray (18%) ends up on the x-axis of the curve in a ratio *black relative exposure : white relative exposure* (by default, 10 : 6.5, or at about 0.61)
    - The curve is applied. This produces an output that is considered to be encoded according to the *gamma*. This can be considered as data ready to be displayed on a monitor using the specified gamma value (the gamma is only used for processing, and does not have to match the gamma of the display used for editing, or that of the output color space).
7. The *look* is applied in pure display-referred mode, like legacy editing of gamma-encoded images, and similar to editing sRGB JPGs in traditional editors. Except for *saturation*, these are also per-channel operations:
    - first the *slope* and the *lift*
    - then the *brightness*
    - finally the *saturation*
8. For each channel, the gamma-encoded data is linearized by applying `linear = gamma_encoded ^ gamma` (so it is scene-referred 0..1, but the encoding is linear, like always in darktable’s pipeline; mid-gray is at 0.18)
9. The HSV representation of the result is calculated. The resulting hue (from the H component) and the original hue are mixed together according to the *preserve hue* setting to produce the final hue. This does not guarantee complete preservation of color, as the “original” hue was already recorded after the input matrix, and after this mix, the output matrix (see the next step) modifies colors again.
10. The output matrix is applied to each channel, which:
    - performs the rotation reversal (this is a difference with *sigmoid*: the latter always completely reverses the rotation, in *AgX*, reversal is up to the user)
    - applies the purity boost (just like *sigmoid*)
    - converts the result back into the pipe working space.

For a deep dive into the theory and development behind AgX, the primary resource is the discussion thread on Blender Artists: Feedback & Development - Filmic - Baby Step to a v2 .

8.2.2. astrophoto denoise

Technical information

description
apply a poisson noise removal best suited for astrophotography.
purpose
corrective.
input
non-linear, Lab, display-referred.
processing
non-linear, Lab.
output
non-linear, Lab, display-referred.

Remove image noise while preserving structure.

This is accomplished by averaging each pixel with some surrounding pixels in the image. The weight of such a pixel in the averaging process depends on the similarity of its neighborhood with the neighborhood of the pixel being denoised. A patch with a defined size is used to measure that similarity.

As denoising is a resource-intensive process, it slows down pixelpipe processing significantly. Consider activating this module late in your workflow.

module controls

patch size
The radius of the patch used for similarity evaluation.
strength
The strength of the denoising.
luma
The amount of denoising to apply to luma. Select carefully in order not to lose too much structure.
chroma
The amount of denoising to apply to chroma. You can be much more aggressive with this parameter.

8.2.3. base curve

Technical information

description
apply a view transform based on personal or camera maker look, for corrective purposes, to prepare images for display.
purpose
corrective.
input
linear, RGB, display-referred.
processing
non-linear, RGB.
output
non-linear, RGB, display-referred.

Simulate the in-camera JPEG by applying a characteristic base curve to the image.

darktable comes with a number of base curve presets that attempt to mimic the curves of various camera manufacturers. These presets are automatically applied according to the “maker ID” found in the image’s Exif data. Camera-specific base curve presets are also available for some camera models.

This module will be enabled by default if preferences > processing > auto-apply pixel workflow defaults is set to “display-referred”. A second option in the preferences dialog allows you to choose whether darktable should attempt to apply a camera-specific base curve (if found) or the generic manufacturer one.

module controls

*Please refer to the curves section for more details about how to modify curves including the **scale for graph** and **preserve colors** controls.*

fusion
Trigger the exposure fusion feature (see this Wikipedia article). This feature allows you to merge the image with one or two copies of itself after applying the current base curve and boosting its exposure by a selected number of EV units. The resulting image is thus a combination of two or three different exposures of the original image.

Use this feature to compress the dynamic range of extremely underexposed images or for true HDR input. For best results, use the *exposure* module to apply a suitable adjustment for correctly exposed highlights.

exposure shift (fusion)
The exposure difference between the merged images in EV units (default 1). This slider is only visible if the *exposure fusion* feature is activated.

exposure bias (fusion)
Determines how the multiple exposures are computed. With a bias of 1 (the default), the image is fused with overexposed copies of itself. With a bias of -1, it is fused with underexposed copies. A bias of 0 attempts to preserve the overall lightness of the image by combining both over- and underexposed copies of the image. This slider is only visible if the *exposure fusion* feature is activated.

8.2.4. bloom

Technical information

description
apply Orton effect for a dreamy ethereal look.
purpose
creative.
input
non-linear, Lab, display-referred.
processing
non-linear, Lab.
output
non-linear, Lab, display-referred.

Create a soft bloom effect.

This module works by blurring the highlights and then blending the result with the original image.

***Note:** This module performs blurs in Lab color space, and is no longer recommended. Instead, use the tone equalizer module or the exposure module with a parametric mask .*

---

module controls

size
The spatial extent of the bloom effect.
threshold
The threshold for the brightness increase.
strength
The strength of the overlighting.

8.2.5. blurs

Technical information

description
simulate physically-accurate lens and motion blurs.
purpose
creative.
input
linear, RGB, scene-referred.
processing
linear, RGB.
output
linear, RGB, scene-referred.

Simulate physically-accurate blurs in scene-referred RGB space.

blur types

Three types of blur are provided:

1. *lens blur*: Simulates a lens diaphragm with a configurable number of blades and blade curvature to create synthetic bokeh.
2. *motion blur*: Simulates the effect of camera motion with a configurable path.
3. *gaussian blur*: This is not really an optical blur but can be used for denoising or for creative effects using blend modes

A diagram at the top of the module shows the shape of the blurring operator (known as the *point spread function*). The module will turn each luminous point from the scene into a blot shaped like the displayed blurring operator, with the size of the blot defined by the *blur radius*.

module controls

general

blur radius
The spreading size of the blur.
blur type
Choose between the different blur variants (above).

controls specific to lens blur

diaphragm blades
The number of blades that the diaphragm is composed of. Older lenses used typically 5 or 7 blades, newer lenses typically use 9 or 11 blades. In any case, real lenses have an odd number of blades and any number greater than 11 blades comes very close to producing a perfect disc. If you degenerate the diaphragm settings with the *concavity* to create a star or an asterisk, this control defines how many branches it has.
concavity
- a concavity of 1 ensures the diaphragm is a regular convex polygon (triangle, pentagon, heptagon, etc.).
- a concavity greater than 1 but lower than `number of blades - 1` turns the shape into a star.
- a concavity greater than `number of blades - 1` but lower than `number of blades` turns the shape into an asterisk, when decreasing *linearity* below 1.
- a concavity greater than or equal to `number of blades` degenerates the shape into a “burst pattern”.

linearity
- a linearity of 0 creates a disc, no matter the number of blades or the concavity.
- a linearity of 1 makes all the outer bounds of the shape straight.
- a linearity between 0 and 1 makes the outer bounds of the shape more or less curved.

rotation
Allows the shape to be rotated with respect to its center – mostly useful with a small number of blades, when a particular orientation is needed.

controls specific to motion blur

direction
The orientation of the motion's path in angular degrees. 0° is horizontal motion.

curvature
The curvature of the motion. Zero produces a straight line, a negative value produces a concave curvature, a positive value produces a convex curvature.

offset
Shifts along the motion path following its curve. This is useful to select a portion of the curved path that is symmetrical, which produces a coma shape (example 1: direction = -45°, curvature = +2, offset = +0.5 ; example 2 : direction = -45°, curvature = +1, offset = +1).

caveats

This module is implemented using a "naive" convolution, which is a slow algorithm. Faster approaches are available (using FFT) but not yet implemented. The GPU implementation, through OpenCL, should hide this issue somewhat. In any case, the runtime of the module will increase with the square of the blur radius.

The blurring process does not take scene depth and depth-of-field into account, but blurs the whole image as a flat object. It is therefore not suitable for creating fake depth-of-field. Using darktable's general masking will only partially work to isolate the foreground of an image, since it will still be blurred into the background.

tips and tricks

All images are usually (even a tiny bit) noisy. If you blur only a part of the image, the blurred region will look suspiciously clean compared to the rest of the image. It is therefore a good idea to add a bit of noise on top of the blurred part to blend it with the rest, using either the grain or the censorize modules.

8.2.6. censorize

Technical information

description
censorize license plates and body parts for privacy.

purpose
creative.

input
linear or non-linear, RGB, scene-referred.

processing
frequential, RGB.

output
special, RGB, scene-referred.

Degrade parts of the image in an aesthetically pleasing way, in order to anonymize people/objects or hide body parts.

Censorize works in linear RGB color space to apply a physically-accurate gaussian blur and gaussian luminance noise.

Aside from anonymization, this module can also be used for a wide range of creative purposes, for example:

- Combine a simple blur with a multiply blend mode to create a realistic bloom (Orton effect).
- Combine a simple blur with a subtract blending mode and low opacity to create an unsharp mask , similar to the sharpen module but in an RGB scene-referred space.
- Add noise to create artificial grain.

Note: The anonymizing methods provided by this module are not forensically safe in order to favor aesthetics. Some forensic techniques may still be able to reconstruct the censorized content based on its structure, especially for simple shapes and text (e.g. license plates, street numbers).

If forensically safe anonymization is required, the only way to achieve this is to paint the surfaces with a solid color.

The darktable team does not accept responsibility for poorly anonymized images leading to the identification of individuals or personal property.

workflow

You are advised to leave the module's controls at their default values while you mask the areas of the image that you wish to censorize, in order that the details of the image remain visible.

module controls

input blur radius
The strength of the first pass of the gaussian blur.

pixellation radius
The size of the "big pixels" created after the first pass of gaussian blur.

output blur radius
The strength of the second pass of the gaussian blur, applied after the pixellation.

noise level
The strength (standard deviation) of the luminance gaussian noise applied after the second pass of the gaussian blur. Adding noise can fake details in the blurred regions and make content detection more difficult for artificial intelligence algorithms.

8.2.7. chromatic aberrations

Technical information

description
correct chromatic aberrations.

purpose
corrective.

input
linear, RGB, scene-referred.

processing
linear, RGB.

output
linear, RGB, scene-referred.

Correct chromatic aberrations.

In contrast to the *raw chromatic aberrations* module, this module does not require raw data as input.

workflow

To obtain the best result, you are advised to proceed as follows:

1. Attenuate the chromatic aberrations as much as possible in the *lens correction* module using the TCA sliders.
2. Increase the strength slider in this module to better see its effect.
3. Increase the radius until the chromatic aberrations disappear. If this is insufficient, try enabling the “very large chromatic aberrations” setting.
4. Choose the guide that gives the best result in term of sharpness and artifacts.
5. Reduce the strength to avoid washing out the colors too much.

For more complicated cases you could also try the following:

- Use several instances with different correction modes – for example, a first instance in “brighten only” mode, and a second in “darken only” mode.
- Use several instances with low strength to correct the chromatic aberrations a little at a time without degrading colors too much.
- Use the module with parametric or drawn masks.
- Use RGB red, green and blue blend modes to restrict the effect to a particular channel.

module controls

guide
The color channel that will be used as a reference for the correction.

radius
The radius of the effect. Increase until chromatic aberrations are eliminated. This is the most important slider of the module.

strength
This slider acts as a safeguard and can help to preserve colorful areas that do not suffer from chromatic aberrations. Increase for stronger correction, decrease for stronger preservation.

correction mode
Allows you to restrict the effect to brighten or darken pixels only. For full control, this can be used in combination with R, G, B blend modes and multiples instances.

very large chromatic aberrations
Makes the algorithm iterative to help in reducing very large chromatic aberrations.

8.2.8. color balance

Technical information

description
shift colors selectively by luminance range.

purpose
corrective or creative.

input
linear, Lab, scene-referred.

processing
non-linear, RGB.

output
non-linear, Lab, scene-referred.

A versatile tool for adjusting the image’s color balance.

This module can be used to revert parasitic color casts or to enhance the visual atmosphere of an image using color grading, a popular technique in the cinema industry. For scene-referred workflow, you should consider using the improved *color balance rgb* module instead.

overview

The *color balance* module allows you to shift colors selectively by luminance range (shadows, mid-tones, and highlights). It can do this using two different methods:

lift, gamma, gain
: The classic method, which allows a more separated control of shadows versus highlights.

slope, offset, power
: The new standard defined by the American Society of Cinematographers Color Decision List (ASC CDL) and more suited to scene-referred editing.

The *master* settings affect the whole image. They are not available in *lift, gamma, gain (sRGB)* mode. The slider ranges are limited to usual values ([50%; 150%] for saturation, [-50%; 50%] for contrast), but higher and lower values can be defined via keyboard input after right-clicking on the corresponding slider.

For better efficiency, in *slope, offset, power* mode it is recommended that you set the slope first, then the offset, and finally the power, in that order. The name of the mode can be use as a mnemonic to remember the order.

The shadows parameter has a far heavier effect in *slope, offset, power* mode than in *lift, gamma, gain* mode. When switching from the former to the latter, you should adapt the saturation in shadows, dividing by around 10.

Note: Although this module acts on RGB colors its location in the pixelpipe puts it into the Lab color space. Accordingly the module converts from Lab to RGB, performs its color adjustments, and then converts back to Lab.

---

presets

Several presets are provided in this module to help you to better understand how it can best be used. The “teal/orange color-grading” preset is a very popular look in cinema and is a good showcase model. It is meant to be used with two instances combined with masks . The first instance will exclude skin tones and will shift neutral colors toward teal blue. The second will partially revert the first one and add more vibrance to skin tones only. Together they will create separation between subject and background. The masking and blending parameters will need to be tweaked to suit every image.

Other presets provide Kodak film emulations. In this way you can recreate any film look you like using *color balance*.

module controls

mode

lift, gamma, gain (sRGB) is the legacy mode from darktable 2.4 and earlier. In this mode, the color transformations are applied in sRGB color space encoded with the sRGB gamma (average gamma of 2.2).

lift, gamma, gain (ProPhoto RGB) is the same as the previous mode but works in ProPhoto RGB space, encoded linearly. In this mode, the RGB parameters are corrected in XYZ luminance (Y channel) internally so they affect only the color, and only the “factors” adjust the luminance.

slope, offset, power (ProPhoto RGB) applies the ASC CDL in ProPhoto RGB space, encoded linearly. As with the previous mode, the RGB parameters are corrected in XYZ luminance internally. In this mode, the slope parameter acts as an exposure compensation, the offset acts as a black level correction, and the power acts as a gamma correction. All parameters will have some impact on the whole luminance range but the slope will mostly affect the highlights, the offset will mostly affect the shadows, and the power will mostly affect the mid-tones.

color control sliders

This combobox selection affects the user interface used for the shadows, mid-tones and highlights controls.

RGBL controls allow direct access to the RGB parameters that will be sent to the algorithm and internally adjusted in XYZ luminance, depending on the mode used. They are the only ones stored in darktable's development history.

HSL controls are more intuitive, but are only an interface: the hues and saturations are computed dynamically from and to the RGB parameters and never stored. During the HSL to RGB conversion, the HSL lightness is always assumed to be 50%, so the RGB parameters are always balanced to avoid lightness changes. However, during the RGB to HSL conversion, the HSL lightness is not corrected.

As a consequence, editing in RGB, then in HSL, then again in RGB will not retain the original RGB parameters, but will normalize them so their HSL lightness is 50%. The difference is barely noticeable in most cases, especially using the modes that already correct the RGB parameters internally in XYZ luminance.

In both modes, additional "factor" sliders act on all RGB channels at once. Their effect is similar to the controls of the *levels* module and affect only the luminance.

input saturation

A saturation correction applied before the color balance. This can be used to dampen colors before adjusting the balance, to make difficult images easier to process. When you entirely desaturate the image, this creates a luminance-based monochrome image that can be used as a luminance mask, to create color filters with the *color balance* settings, like a split-toning or sepia effect (when used with blending modes).

output saturation

A saturation correction applied after the color balance. This is useful once you have found a proper hue balance but find the effect too heavy, so you can adjust the global saturation at once instead of editing each channel saturation separately at the expense of possibly messing up the colors.

contrast / contrast fulcrum

The contrast slider allows the luminance separation to be increased. The fulcrum value defines the luminance value that will not be affected by the contrast correction, so the contrast will roll over the fulcrum. Luminance values above the fulcrum will be amplified almost linearly. Luminance values below the fulcrum value will be compressed with a power function (creating a toe). This correction comes after the output saturation and is applied on all RGB channels separately, so hues and saturations might not be preserved in case of dramatic settings (shadows might be resaturated, highlights might be desaturated, and some color shift is to be expected).

shadows, mid-tones, highlights

Depending on the mode used, the shadows settings will control either the lift or the offset, the mid-tones settings will control either the gamma or the power, and the highlights settings will control either the gain or the slope. Parameters are transferred unaltered when you change the mode.

In *RGBL* mode, the RGB sliders' range is limited to [-0.5; 0.5]. In *HSL* mode, the saturation sliders range is limited to [0%; 25%]. Values outside of these bounds can be defined with keyboard input by right-clicking on the slider.

Note: The shadows, mid-tones and highlights sliders can take up a great deal of space in the *color balance* module. The overall layout of these sliders can therefore be cycled through three different layouts by clicking on the *shadows, mid-tones, highlights* heading.

---

optimize luma

The picker beside the optimize luma label will select the whole image and optimize the factors for shadows, mid-tones and highlights so that the average luminance of the image is 50% Lab, the maximum is 100% and the minimum is 0%, at the output of this module. This is essentially histogram normalization, similar to that performed by the *levels* module. The optimizer is only really accurate when used in *slope, offset, power* mode.

If you want more control, you can define three control patches by using the pickers beside each factor slider to sample luminance in selected areas. The *shadows* picker samples the minimum luminance, the *mid-tones* picker samples the average luminance, and the *highlights* picker samples the maximum luminance. The most sensitive parameter is the mid-tones factor, since selecting a slightly different area can lead to dramatic parameter changes. Using the factors pickers alone, without triggering the luma optimization, will allow you to perform adjustments without general optimization, but each parameter is always computed taking the other two into account. Once patches are selected, the label changes to read “optimize luma from patches”. To reset one patch, you can just redo the selection. Patches are not saved in the parameters and are retained only during the current session.

It is important to note that the luminance adjustment targets only the output of the *color balance* module and does not account for adjustments performed in other modules later in the pixelpipe (e.g. *filmic rgb* , *tone curve* , *color zones* , *levels*). Using the *color balance* module to remap the luminance globally on the image is not recommended because it does not preserve the original colors – modules such as *tone curve* or *filmic rgb* are better suited for this purpose. Luminance adjustments in *color balance* are better performed, in combination with color adjustments, using masks.

neutralize colors

In an image where some areas are exposed to direct sunlight and some areas are exposed to reflected light (shadows), or where several artificial light sources are present simultaneously, shadows and highlights often have different color temperatures. These images are particularly difficult to correct since no general white balance will match all the colors at once. The color neutralization optimizer aims at helping you find the complementary color for shadows, mid-tones, and highlights so that all the color casts are reverted, and the average color of the image is a neutral gray.

As with the luma optimization, the picker beside the neutralize colors label will trigger a general optimization over the whole image. This works fairly well in landscape photography, or for any photograph with a full spectrum of colors and luminances.

For night and events photography, this will most likely fail and you will need to manually input the sampling areas with the picker beside each hue slider. For the highlights sample, use a color exposed to spotlights that should be neutral white or light gray. For the shadows sample, use a color exposed to ambient light that should be neutral black or dark gray. For the mid-tones sample, use a color exposed by both ambient and spotlights.

The success of the optimization depends on the quality of the samples. Not every set of samples will converge to a good solution and you need to ensure that the color patches you choose are really a neutral color in real life. In many cases the optimizer will output the correct hue but an excessive saturation that will need some extra tweaking. In some cases, no valid optimization will be delivered and you will need to reset the saturation parameters and start over, or simply stop after the patches selection. Note that in the auto-optimization, the maximum saturation is 25%, which might not be enough in very few cases but will avoid inconsistent results in most.

If you select color patches from the hue pickers without triggering the optimization, the software will only perform one round of optimization and then stop. This allows you to control each luminance range separately and avoid divergence of the solution in corner cases. The hue and saturation corrections are computed taking into account the two other luminance ranges and three factors, and will always output the complementary color of the selected area. If you want to reinforce the color of the area instead, you can then add 180° to the computed hue. Once patches are selected, the label changes to read “neutralize colors from patches”. To reset one patch you can just redo the selection. Patches are not saved in the parameters and are retained only during the current session. The parameters found by the automatic neutralization are accurate only in *slope, offset, power* mode, but can work to some extent in *lift, gamma, gain* mode too.

8.2.9. color balance rgb

Technical information

description
color grading tools using alpha masks to separate shadows, mid-tones and highlights.

purpose
corrective or creative.

input
linear, RGB, scene-referred.

processing
non-linear, RGB.
output
non-linear, RGB, scene-referred.

An advanced module which brings color-grading tools from cinematography into the photographic scene-referred pipeline.

This module is not suitable for beginners with no prior knowledge of color theory, who might want to stick to the *global chroma* and *global vibrance* settings until they have a good understanding of the dimensions of color .

introduction

Color-grading is an important part of image editing. It can help to remove unwanted color casts and can also deliver a creative color twist that will add atmosphere to your images. In the days of film photography, most of the color ambiance was obtained with the film emulsion and the developing chemicals, with some color timing being performed under the enlarger with color heads. This consumed expensive resources and was mostly reserved for the cinema industry, where the job was performed by a colorist.

In the digital age, where raw images look flat and even, color-grading assumes the same role that film emulsions did, by re-introducing color shifts for aesthetic purposes. It can also serve to harmonize the color palette of a series of images (which may have been shot under different conditions) to achieve a consistent global look. For this task, the vectorscope is also extremely useful.

Colorists usually split color-grading into two distinct steps:

1. *Primary color-grading* aims to fix unwanted color casts and create a neutral starting point,
2. *Secondary color-grading* gives the image its final look and atmosphere.

Primary color-grading is best left to the *color calibration* module, which operates in a physical framework better suited to illuminant correction. *Color balance RGB*, on the other hand, is mostly concerned with secondary color-grading. Performing a truly neutral primary color-grading should make the secondary color-grading easy to transfer between images (via styles, presets or copy & paste) with a similar effect.

general principles

The *color balance RGB* module is an improvement over the American Society of Cinematographers Color Decision List (ASC CDL), and uses alpha masks to allow the effect to be properly split between shadows and highlights. The classic CDL acts on the entire luminance range, and each of its parameters is given more weight on some parts of the image only as a side-effect of the mathematics.

This module works, for the most part (*4 ways*, *chroma*, *vibrance*, *contrast*), in a linear RGB color space designed specifically for color-grading. This color space exhibits a uniform spacing of perceptual hues while retaining a physically-scaled luminance[1]. The perceptual part of the module (*saturation* and *brilliance*) works in the JzAzBz[2] color space, which provides a perceptual scaling of both lightness and chromaticity suitable for HDR images. Both color spaces ensure that saturation and chroma changes take place at constant hue, which is not the case for most other saturation operators in darktable (notably in the older *color balance* module).

The *color balance RGB* module expects a scene-referred linear input and produces a scene-referred RGB output, which may or may not be linear, depending on the module settings (*contrast* and *power* will delinearize the output).

At its output, *color balance RGB* checks that the graded colors fit inside the pipeline RGB color space (Rec. 2020 by default) and applies a soft saturation clipping at constant hue, aiming to retarget out-of-gamut color to the nearest in-gamut color by scaling both chroma and lightness. This prevents the chroma and saturation settings from pushing colors outside of the valid range and allows more drastic adjustments to be safely used.

Note that this module abides by the CIE definitions of chroma and saturation, as explained in the dimensions of color section.

module controls

master tab

hue shift

Rotate all colors in the image by an angle over the chromaticity plane, at constant luminance and chroma. You can use this control to remove spilled colored light on a subject or to quickly change the color of some object. This setting is usually best applied locally, using masks .

global vibrance

This affects the chroma dimension of color over the entire image, prioritizing those colors with low chroma. This allows the chroma of neutral colors to be increased without exaggerating already-colorful pixels.

contrast

This setting is applied on the luminance channel at constant hue and chroma. The fulcrum setting (in the *masks* tab, under *contrast gray fulcrum*) allows you to set the neutral point of the contrast curve:

- at the fulcrum, the contrast curve leaves the luminance unchanged,
- below the fulcrum, the contrast curve decreases the luminance for positive contrast values, or increases it for negative values,
- above the fulcrum, the contrast curve increases the luminance for positive contrast values, or decreases it for negative values.

The fulcrum has a value of 18.45% by default, which is consistent with the current scene-referred workflow and should fit most use cases (assuming that global brightness has been fixed as recommended using the *exposure* module).

The contrast algorithm gives natural results that mimic the central part of the contrast curve of analog film. However, it will also increase the image's dynamic range, which may void *filmic* settings in the pipe. For global contrast adjustments, you should normally use the *tone equalizer* module – the *color balance RGB* contrast slider is best used with masks, e.g. for selective corrections over the foreground or background.

linear chroma grading

Linear chroma grading affects the chroma dimension proportionally to its input value, at constant hue and luminance. It does this globally, with a flat coefficient (using the *global chroma*), as well as on each of the *shadows*, *mid-tones* and *highlights* masks (defined in the *masks* tab under *luminance ranges*).

perceptual saturation grading

Perceptual saturation grading affects both the luminance and the chroma dimensions, in a perceptual space, proportionally to its input value, at constant hue. It does this globally, with a flat coefficient (using the *global saturation*), as well as on each of the *shadows*, *mid-tones* and *highlights* masks (defined in the *masks* tab under *luminance ranges*).

perceptual brilliance grading

Perceptual brilliance grading affects both the luminance and the chroma dimensions, in a perceptual space, proportionally to its input value, at constant hue, and in a direction orthogonal to the saturation. Its effect is close to that of changing exposure, but scaled perceptually. It does this globally, with a flat coefficient (using the *global saturation*), as well as on each of the *shadows*, *mid-tones* and *highlights* masks (defined in the *masks* tab under *luminance ranges*). Brilliance Values outside the range -20% - 20% are likely to create artifacts, and are best avoided.

4 ways tab

Each of the settings in the 4 ways tab is composed of the same three components, which define a color using independent coordinates:

1. *luminance*,
2. *hue*,
3. *chroma*.

Color input like this defines a color shift applied to the image globally or over the specified luminance range.

Each hue slider has a picker , which may be used to compute the opponent color of the selected region. This is useful to revert unwanted color casts (e.g. skin redness), since shifting the color to its opponent cast neutralizes it. You can also select the current color by Ctrl+clicking on the picker.

global offset

This is equivalent to the ASC CDL *offset* and falls back to adding a constant RGB value to all pixels, quite like the *black offset* in the *exposure* module. This control does not use masking.

shadows lift

This is conceptually equivalent to the *lift* from *lift/gamma/gain*, although implemented differently, and falls back to multiplying the masked pixels by a constant RGB value. It is applied using the *shadows* mask.

highlights gain

This is equivalent to the ASC CDL *slope*, and falls back to multiplying a the masked pixels by a constant RGB value. It is applied using the *highlights* mask.

global power

This is equivalent to the ASC CDL *power*, and falls back to applying a constant RGB exponent. It is not masked and needs to be normalized, since the power function has a different behavior above and below 1, and we are in an unbounded pipeline where white is typically greater than 1. The normalization parameter is available in the *masks* tab under *white fulcrum*.

masks tab

This tab defines auxiliary controls for the previous tabs. You should at least set the *white fulcrum threshold*, as that value is used to normalize brightness levels; leaving it at the default may introduce artifacts in other modules later in the pipeline (*filmic rgb* highlight recovery being the prominent example).

luminance ranges

The graphs show the opacity (on the *y* axis) of the 3 luminance masks relative to the pixel luminance (on the *x* axis). The darkest curve represents the *shadows* mask, the brightest represents the *highlights* mask, and the third curve represents the *mid-tones* mask.

Only the *shadows* and *highlights* masks can be controlled directly – the *mid-tones* mask is computed indirectly from the others and acts as an adjustment variable.

shadows fall-off
: Control the softness or hardness of the transition from fully opaque (100%) to fully transparent (0%) for the shadows mask.

mask middle-gray fulcrum
: Set the luminance value where all three masks have 50% opacity. In practice, this is used to define how the image is separated into shadows and highlights.

highlights fall-off
Control the softness or hardness of the transition from fully opaque (100%) to fully transparent (0%) for the highlights mask.

For each of these settings, a mask button, provided to the right of the slider, displays the appropriate mask (shadows, mid-tones, highlights), overlaid as a checker board. The still-visible area of the image (not hidden by the mask) is the area that will be affected by the shadows, mid-tones and highlights sliders in the other tabs.

All mask previews display the output of the module, including any color changes made, so you can also activate them while editing, to see only the affected part of the image.

Luminance masks are computed at the input of the module, which means that they are insensitive to any luminance changes made inside the module.

thresholds

white fulcrum
Set the white point luminance in EV. This is used to normalize the *power* setting in the *4 ways* tab, and also to normalize brightness levels. Display-referred implementations of power functions assume that white is at 100%, which removes the need for normalization. For scene-referred purposes this needs to be taken into account.

The picker to the right of the slider automatically sets the white fulcrum to the maximum luminance from the selected region, which should be sufficient in most cases.

contrast gray fulcrum
Set the fulcrum for the *contrast* setting in the *master* tab. This corresponds to the luminance value that will be left unchanged by the contrast adjustment. This setting usually matches the middle-gray linear value. If you followed the scene-referred workflow recommendations and set the global brightness early in the pipeline, using the *exposure* module, the correct value should usually be around 18-20%.

The picker to the right of the slider automatically sets the contrast gray fulcrum to the average luminance from the selected region. This relies on the assumption that the average luminance is usually close to middle-gray, which is not true if you have specular highlights or primary light sources in the frame, or for low/high-key images.

saturation formula

Note that this setting is not really appropriate for the *masks* tab (since it is not technically related to the masks) but is placed here because it is not meant to be used regularly and in the spirit of saving some display real-estate. Two options are provided:

JzAzBz (2021)
This mode is the original saturation algorithm. It uses the JzAzBz uniform color space (UCS) to compute the saturation. This color space is not meant for color changes and its lightness does not account for the Helmholtz-Kohlrausch effect , which states that colorful colors will look brighter than neutral or near-neutral colors (greys and pastels) having the same luminance. It also suffers from non-smooth behavior near black, with colors being darkened too much.

darktable UCS (2022)
The darktable Uniform Color Space has been designed from the ground up , using psychoperceptual measurement datasets, for the sole purpose of the color manipulation (saturation) performed by this module. This color space does account for the Helmholtz-Kohlrausch effect and has a built-in gamut mapping formula that is more accurate and efficient than can be achieved in JzAzBz. It displays a smoother behavior which makes saturation changes more even across the lightness range.

mask preview settings

These settings apply to the mask previews, displayed by clicking the mask buttons in the *luminance ranges* section. These settings are saved globally, so will be applied to all subsequent images unless changed.

checker board color 1 and 2
Set the two colors for the background checker board mask underlay. You can set them to opponent colors of the current image to aid legibility.

checker board size
Set the width of the checker board cells in pixels (adjusted according to the display's DPI setting).

FAQ

saturation or chroma?

As described in the dimensions of color section, saturation and chroma roam the (lightness, chroma) plane in different directions. In addition, the chroma of *color balance RGB* uses a scene-referred linear space, while the saturation uses a perceptual space, which rescales color for even spacing.

In practice, you should use the chroma setting if you want to preserve the scene-linearity of the light emission and/or keep the luminance unchanged. However, these changes might affect some hues more heavily than others, due to the fact that the color space is not fully perceptually-scaled.

Saturation is closer to the effect of mixing white paint with some base color. Reducing the saturation of red will degrade it to pink, while reducing its chroma will degrade to a gray shade at the same luminance. Saturation is perhaps a more intuitive way to interact with color, due to its connection with painting.

Choosing one or the other is mostly a matter of deciding where on the (lightness, chroma) graph you want to push your colors, and where they are to begin with. To reach pastel colors, saturation is the way to go. To reach laser-like colors (almost monochromatic), at the risk of looking synthetic, chroma is the way to go.

what is the connection with lift/gamma/gain?

The *lift/gamma/gain* algorithm relies on a display-referred color space, since it assumes a bounded and symmetric dynamic range, with white point at 100% and gray at 50%. As such, it is simply unusable in a scene-referred space. However, the only incompatible part is the *lift*. The *gamma* is exactly the ASC CDL *power*, and the *gain* is exactly the ASC CDL *slope*.

The *color balance RGB* module simply has two slopes instead of one: the *gain*, applied on the highlights extracted from the whole image by a mask, and the *lift*, applied similarly but on the shadows.

changing contrast

While *color balance RGB* is mostly about color (other modules handle the global contrast in chromaticity-preserving ways) luminance is as much a part of color as hue or chroma, and it needs to be dealt with here too, because the perception of saturation relies on it. If you wish to turn red into pink, for example, reducing its chroma will turn it gray, so you need to increase its luminance as well.

There are several ways to change the contrast in *color balance RGB*, either locally (with masks) or globally (without):

- in the *master* tab, use the *contrast* setting (possibly alongside the *contrast gray fulcrum* in the *masks* tab). Be aware that this will raise the white point and therefore increase the dynamic range of the image, which may void filmic settings later in the pipeline.
- in *perceptual saturation grading* , desaturate highlights and resaturate shadows to produce a luminance contrast boost,
- in *perceptual brilliance grading* , add brilliance in the highlights and remove brilliance in the shadows to produce to a luminance contrast boost,
- in the *4 ways* tab, set the *shadows lift* luminance to negative values and the *highlights gain* luminance to positive values, which also produces a luminance contrast boost.

The difference between these methods is how the effect will be weighted relative to the input of the module. You are advised to do the majority of luminance contrast adjustments in the *filmic* and *tone equalizer* modules, and then undertake final changes in *color balance RGB* while examining the colors.

internal processing

The following is the internal order of operations within the module:

1. Transform from pipeline RGB to Kirk/Filmlight Ych space,
2. Apply hue shift at constant chroma and constant luminance,
3. Compute luminance masks with Y,
4. Apply the *linear chroma* and *vibrance* settings at constant hue and luminance,
5. Transform to Kirk/Filmlight RGB space,
6. Apply the *4 ways* settings (except *luminance power*),
7. Transform to Kirk/Filmlight Yrg space,
8. Apply *luminance power* and *contrast* on Y,
9. Transform to JzAzBz space,
10. Apply the *perceptual saturation* and *perceptual brilliance* settings,
11. Soft-clip the chroma using pipeline RGB gamut at constant hue and lightness,
12. Transform back to pipeline RGB.

caveats

Setting the global chroma to -100% will not produce a real monochrome image, as is customary with other algorithms. The reason for this is that the RGB space used has a D65 white point defined in CIE LMS 2006 space, while darktable uses a white point defined in CIE XYZ 1931 space, and there is no exact conversion between these spaces. The result will therefore be a slightly tinted black & white image. If your intent is to get a real black & white image using the luminance channel, the *color calibration* module offers a *B&W : luminance-based* preset that does exactly the same thing but without the white-point discrepancy.

This module has its gamut-mapping (against pipeline RGB) permanently enabled. This means that if your original image contains some largely out-of-gamut colors to start with, simply enabling *color balance RGB* with no particular setting will slightly alter its colors. This is probably for the best.

The maximum saturation allowed in the pipeline working RGB space is recorded for each hue when initializing the module, and is later cached in a LUT (look-up table) to save performance. If the working profile is later changed, *color balance RGB* is not notified, meaning that it will not update its cached hue/saturation LUT. To force a LUT update, you can simply change any setting in the *color balance RGB* module, then change it back again. It is not recommended that you change the working RGB space half-way through an editing session, as this could result in unexpected chroma and hue changes.

For performance reasons, the non-linear conversions from and to the working RGB space are bypassed, meaning that the internal colorimetry will be wrong when using non-linear color spaces. Note that there is no reason to use non-linear spaces as working RGB since they make alpha blending more challenging for no benefit.

---

1. Richard A. Kirk, Chromaticity coordinates for graphic arts based on CIE 2006 LMS with even spacing of Munsell colors, 2019. https://doi.org/10.2352/issn.2169-2629.2019.27.38 ↩

2. Safdar et al., Perceptually uniform color space for image signals including high dynamic range and wide gamut, 2017. https://doi.org/10.1364/OE.25.015131 ↩

8.2.10. color calibration

Technical information

description
perform color space corrections such as white balance, channels mixing and conversions to monochrome emulating film.

purpose
corrective or creative.

input
linear, RGB, scene-referred.

processing
linear, RGB or XYZ.

output
linear, RGB, scene-referred.

A fully-featured color-space correction, white balance adjustment and channel mixer module.

This simple yet powerful module can be used in the following ways:

- To adjust the white balance (chromatic adaptation), working in tandem with the *white balance* module. Here, the *white balance* module makes some initial adjustments (required for the *demosaic* module to work effectively), and the *color calibration* module then calculates a more perceptually-accurate white balance after the input color profile has been applied.
- As a simple RGB channel mixer , adjusting the R, G and B output channels based on the R, G and B input channels, to perform cross-talk color-grading.
- To adjust the color saturation and brightness of the image, based on the relative strength of the R, G and B channels of each pixel.
- To produce a grayscale output based on the relative strengths of the R, G and B channels, in a way similar to the response of black and white film to a light spectrum.
- To improve the color accuracy of the input color profile using a color checker chart.

White Balance in the Chromatic Adaptation Transformation (CAT) tab

Chromatic adaptation aims to predict how all surfaces in the scene would look if they had been lit by another illuminant. What we actually want to predict, though, is how those surfaces would have looked if they had been lit by the same illuminant as your monitor, in order to make all colors in the scene match the change of illuminant. White balance, on the other hand, aims only at ensuring that whites and grays are really neutral (R = G = B) and doesn't really care about the rest of the color range. White balance is therefore only a partial chromatic adaptation.

Chromatic adaptation is controlled within the Chromatic Adaptation Transformation (CAT) tab of the *color calibration* module. When used in this way the *white balance* module is still required as it needs to perform a basic white balance operation (connected to the input color profile values). This technical white balancing ("camera reference" mode) is a flat setting that makes grays lit by a standard D65 illuminant look achromatic, and makes the demosaicing process more accurate, but does not perform any perceptual adaptation according to the scene. The actual chromatic adaptation is then performed by the *color calibration* module, on top of those corrections performed by the *white balance* and *input color profile* modules. The use of custom matrices in the *input color profile* module is therefore discouraged. Additionally, the RGB coefficients in the *white balance* module need to be accurate in order for this module to work in a predictable way.

The *color calibration* and *white balance* modules can be automatically applied to perform chromatic adaptation for new edits by setting the chromatic adaptation workflow option (preferences > processing > auto-apply chromatic adaptation defaults) to "modern". If you prefer to perform all white balancing within the *white balance* module, a "legacy" option is also provided. Neither option precludes the use of other modules (such as *color balance RGB*) for creative color grading further along the pixel pipeline.

By default, *color calibration* performs chromatic adaptation by:

- reading the RAW file's Exif data to fetch the scene white balance set by the camera,
- adjusting this setting using the camera reference white balance from the *white balance* module,
- further adjusting this setting with the input color profile in use (standard matrix only).

For consistency, the *color calibration* module's default settings always assume that the standard matrix is used in the *input color profile* module – any non-standard settings in this module are ignored. However, *color calibration*'s defaults can read any auto-applied preset in the *white balance* module.

It is also worth noting that, unlike the *white balance* module, *color calibration* can be used with masks . This means that you can selectively correct different parts of the image to account for differing light sources.

To achieve this, create an instance of the *color calibration* module to perform global adjustments using a mask to exclude those parts of the image that you wish to handle differently. Then create a second instance of the module reusing the mask from the first instance (inverted) using a raster mask .

CAT tab workflow

The default illuminant and color space used by the chromatic adaptation are initialised from the Exif metadata of the RAW file “as set in camera”.

Alternatively you can use the picker (to the right of the color patch) to select a neutral color from the image or, if one is unavailable, select the entire image. In this case, the algorithm finds the average color within the chosen area and sets that color as the illuminant. This method relies on the “gray-world” assumption, which predicts that the average color of a natural scene will be neutral. This method will not work for artificial scenes, for example those with painted surfaces.

- Select “*as shot in camera*” to restore the camera defaults and re-read the RAW Exif.

The color patch shows the color of the currently calculated illuminant projected into sRGB space. The aim of the chromatic adaptation algorithm is to turn this color into pure white, which does not necessarily means shifting the image toward its *perceptual* opponent color. If the illuminant is properly set, the image will be given the same tint as shown in the color patch when the module is disabled.

To the left of the color patch is the *CCT* (correlated color temperature) approximation. This is the closest temperature, in kelvin, to the illuminant currently in use. In most image processing software it is customary to set the white balance using a combination of temperature and tint. However, when the illuminant is far from daylight, the CCT becomes inaccurate and irrelevant, and the CIE (International Commission on Illumination) discourages its use in such conditions. The CCT reading informs you of the closest CCT match found:

- When the CCT is followed by “*(daylight)*”, this means that the current illuminant is close to an ideal daylight spectrum ± 0.5 %, and the CCT figure is therefore meaningful. In this case, you are advised to use the “*D (daylight)*” illuminant.
- When the CCT is followed by “*(black body)*”, this means that the current illuminant is close to an ideal black body (Planckian) spectrum ± 0.5 %, and the CCT figure is therefore meaningful. In this case, you are advised to use the “*Planckian (black body)*” illuminant.
- When the CCT is followed by “*(invalid)*”, this means that the CCT figure is meaningless and wrong, because we are too far from either a daylight or a black body light spectrum. In this case, you are advised to use the *custom* illuminant. The chromatic adaptation will still perform as expected (see the note below), so the “*(invalid)*” tag only means that the current illuminant color is not accurately tied to the displayed CCT. This tag is nothing to be concerned about – it is merely there to tell you to stay away from the daylight and planckian illuminants because they will not behave as you might expect.

When one of the above illuminant detection methods is used, the module checks where the calculated illuminant sits using the two idealized spectra (daylight and black body) and chooses the most accurate spectrum model to use in its *illuminant* parameter. The user-interface will change accordingly:

- A temperature slider will be provided if the detected illuminant is close to a *D (daylight)* or *Planckian (black body)* spectrum, for which the CCT is meaningful.
- Hue and chroma sliders in CIE 1976 Luv space are offered for the *custom* illuminant, which allows direct selection of the illuminant color in a perceptual framework without any intermediate assumption.

Note: Internally, the illuminant is represented by its absolute chromaticity coordinates in CIE xyY color space. The illuminant selection options in the module are merely interfaces to set up this chromaticity from real-world relationships and are intended to make this process faster. It does not matter to the algorithm if the CCT is tagged “invalid” – this just means that the relationship between the CCT and the corresponding xyY coordinates is not physically accurate. Regardless, the color set for the illuminant, as displayed in the patch, will always be honored by the algorithm.

When switching from one illuminant to another, the module attempts to translate the previous settings to the new illuminant as accurately as possible. Switching from any illuminant to *custom* preserves your settings entirely, since the *custom* illuminant is a general case. Switching between other modes, or from *custom* to any other mode, will not precisely preserve your settings from the previous mode due to rounding errors.

Other hard-coded *illuminants* are available (see below). Their values come from standard CIE illuminants and are absolute. You can use them directly if you know exactly what kind of light bulb was used to illuminate the scene and if you trust your camera’s input profile and reference (D65) coefficients to be accurate. Otherwise, see *caveats* below.

CAT tab controls

adaptation

The working color space in which the module will perform its chromatic adaptation transform and channel mixing. The following options are provided:

- *Linear Bradford (1985)*: This is accurate for illuminants close to daylight and is compatible with the ICC v4 standard, but produces out-of-gamut colors for more difficult illuminants.
- *CAT16 (2016)*: This is the default option and is more robust in avoiding imaginary colors while working with large gamut or saturated cyan and purple. It is more accurate than the Bradford CAT in most cases.
- *Non-linear Bradford (1985)*: This can sometimes produce better results than the linear version but is unreliable.
- *XYZ*: This is the least accurate method and is generally not recommended except for testing and debugging purposes.
- *none (disable)*: Disable any adaptation and use the pipeline working RGB space.

illuminant

The type of illuminant assumed to have lit the scene. Choose from the following:

- *same as pipeline (D50)*: Do not perform chromatic adaptation in this module instance but just perform channel mixing, using the selected *adaptation* color space.
- *CIE standard illuminant*: Choose from one of the CIE standard illuminants (daylight, incandescent, fluorescent, equi-energy, or black body), or a non-standard “LED light” illuminant. These values are all pre-computed – as long as your camera sensor is properly profiled, you can just use them as-is. For illuminants that lie near the Planckian locus, an additional “temperature” control is also provided (see below).
- *custom*: If a neutral gray patch is available in the image, the color of the illuminant can be selected using the picker , or can be manually specified using hue and saturation sliders (in LCh perceptual color space). The color swatch next to the picker shows the color of the calculated illuminant used in the CAT compensation.
- *as shot in camera*: Calculate the illuminant based on the white balance settings provided by the camera.

temperature

Adjust the color temperature of the illuminant. Move the slider to the right to assume a more blue illuminant, which will make the white-balanced image appear warmer/more red. Move the slider to the left to assume a more red illuminant, which makes the image appear cooler/more blue after compensation.

This control is only provided for illuminants that lie near the Planckian locus and provides fine-adjustment along that locus. For other illuminants the concept of “color temperature” doesn’t make sense, so no temperature slider is provided.

hue

For custom white balance, set the *hue* of the illuminant color in LCh color space (derived from CIE Luv space).

chroma

For custom white balance, set the *chroma* (or saturation) of the illuminant color in LCh color space (derived from CIE Luv space).

gamut compression

Most camera sensors are slightly sensitive to invisible UV wavelengths, which are recorded on the blue channel and produce “imaginary” colors. Once corrected by the input color profile, these colors will end up out of gamut (that is, it may no longer be possible to represent certain colors as a valid [R,G,B] triplet with positive values in the working color space) and produce visual artifacts in gradients. The chromatic adaptation may also push other valid colors out of gamut, at the same time pushing any already out-of-gamut colors even further out of gamut.

Gamut compression uses a perceptual, non-destructive, method to attempt to compress the chroma while preserving the luminance as-is and the hue as close as possible, in order to fit the whole image into the gamut of the pipeline working color space. One example where this feature is very useful is for scenes containing blue LED lights, which are often quite problematic and can result in ugly gamut clipping in the final image.

clip negative RGB from gamut

Remove any negative RGB values (set them to zero). This helps to deal with bad black level as well as the blue channel clipping issues that may occur with blue LED lights. This option is destructive for color (it may change the hue) but ensures a valid RGB output no matter what. It should never be disabled unless you want to take care of the gamut mapping manually and understand what you are doing. In that case, use the *black level correction* in the *exposure* module to get rid of any negative RGB (RGB means light, which is energy, and which should always be a positive quantity), then increase the *gamut compression* until no solid black patches remain in the image. Proper denoising may help getting rid of odd RGB values too. Note that this approach may still be insufficient to recover some deep and luminous shades of blue.

Note 1: It has been reported that some OpenCL drivers don't play well when negative RGB values are present in the pixel pipeline, because many pixel operators use logarithms and power functions (*filmic*, *color balance*, all the CIE Lab <-> CIE XYZ color space conversions), which are not defined for negative numbers. Although the inputs are sanitized before sensitive operations, it is not enough for some OpenCL drivers, which will output isolated `NaN` (Not a Number) values. These `NaN` values may be subsequently spread by local filters (blurring and sharpening operations, like *sharpness*, *local contrast*, *contrast equalizer*, *low pass*, *high pass*, *surface blur*, and *filmic* highlights reconstruction), resulting in large black, gray or white squares.

In all these cases, you **must** enable the "clip negative RGB from gamut" option in the *color calibration* module.

Note 2: A common case for failure of the color algorithms in *color calibration* (especially the gamut compression) is pixels that have a luminance value of 0 (Y channel of the CIE 1931 XYZ space), but non-zero chromaticity values (X and Z channels of the CIE 1931 XYZ space). This case is a numerical oddity that matches no physical reality (a pixel with no luminance should have no chromaticity either), will produce a division by zero in xyY and Yuv color spaces, and will create `NaN` RGB values as a result. This issue is **not** corrected inside *color calibration* because it is a symptom of a bad input profiling and/or a bad black point level, and needs to be addressed manually either by adjusting the input color profile with the channel mixer or in the *exposure* module's *black level correction*.

CAT warnings

The chromatic adaptation in this module relies on a number of assumptions about the earlier processing steps in the pipeline in order to work correctly, and it can be easy to inadvertently break those assumptions in subtle ways. To help you to avoid these kinds of mistakes, the *color calibration* module will show warnings in the following circumstances.

- If the *color calibration* module is set up to perform chromatic adaptation but the *white balance* module is not set to "camera reference", warnings will be shown in both modules. These errors can be resolved either by setting the *white balance* module to "camera reference" or by disabling chromatic adaptation in the *color calibration* module. Note that some sensors may require minor corrections within the *white balance* module in which case these warnings can be ignored.
- If two or more instances of *color calibration* have been created, each attempting to perform chromatic adaptation, an error will be shown on the second instance. This could be a valid use case (for instance where masks have been set up to apply different white balances to different non-overlapping areas of the image) in which case the warnings can be ignored. For most other cases, chromatic adaptation should be disabled in one of the instances to avoid double-corrections.

  By default, if an instance of the *color calibration* module is already performing chromatic adaptation, each new instance you create will automatically have its adaptation set to "none (bypass)" to avoid this "double-correction" error.

The chromatic adaptation modes in *color calibration* can be disabled by either setting the *adaptation* to "none (bypass)" or setting the *illuminant* to "same as pipeline (D50)" in the CAT tab.

These warnings are intended to prevent common and easy mistakes while using the automatic default presets in the module in a typical RAW editing workflow. When using custom presets and some specific workflows, such as editing film scans or JPEGs, these warnings can and should be ignored.

Advanced users can disable module warnings in preferences > processing > show warning messages .

channel mixing

The remainder of this module is a standard channel mixer, allowing you to adjust the output R, G, B, colorfulness, brightness and gray of the module based on the relative strengths of the R, G and B input channels.

Channel mixing is performed in the color space defined by the *adaptation* control on the CAT tab . For all practical purposes, these CAT spaces are particular RGB spaces tied to human physiology and proportional to the light emissions in the scene, but they still behave in the same way as any other RGB space. The use of any of the CAT spaces can make the channel mixer tuning process easier, due to their connection with human physiology, but it is also possible to mix channels in the RGB working space of the pipeline by setting the *adaptation* to "none (bypass)". To perform channel mixing in one of the *adaptation* color spaces without chromatic adaptation, set the *illuminant* to "same as pipeline (D50)".

Note: The actual colors of the CAT or RGB primaries used for the channel mixing, projected to sRGB display space, are painted in the background of the RGB sliders, so you can get a sense of the color shift that will result from your altered settings.

Channel mixing is a process that defines a boosting/muting factor for each channel as a ratio of all the original channels. Instead of entering a single flat correction that ties the output value of a channel to its input value (for example, `R_output = R_input × correction`), the correction to each channel is dependent on the input of *all* of the channels for each pixel (for example, `R_output = R_input × R_correction + G_input × G_correction + B_input × B_correction`). Thus a pixel's channels contribute to each other (a process known as "cross-talk") which is equivalent to rotating the primary colors of the color space in 3D. This is, in effect, digital simulation of physical color filters.

Although rotating primary colors in 3D is ultimately equivalent to applying a general hue rotation, the connection between the RGB corrections and the resulting perceptual hue rotation is not directly predictable, which makes the process non-intuitive. "R", "G" and "B" should be taken as a mixture of 3 lights that we dial up and down, not as a set of colors or hues. Also, since RGB tristimulus does not decouple luminance and chrominance, but is an additive lighting setup, the "G" channel is more strongly tied to human luminance perception than the "R" and "B" ones. All pixels have a non-zero G channel, which implies that any correction to the G channel is likely to affect all pixels.

The channel mixing process is therefore tied to a physical interpretation of the RGB tristimulus (as additive lights), which makes it well-suited for primary color grading and illuminant corrections, and blends the color changes smoothly. However, trying to understand and predict it from a perceptual point of view (luminance, hue and saturation) is going to fail and is discouraged.

Note: The "R", "G" and "B" labels on the channels of the color spaces in this module are merely conventions formed out of habit. These channels do not necessarily look "red", "green" and "blue", and users are advised against trying to make sense out of them based on their names. This is a general principle that applies to *any* RGB space used in any application.

R, G and B tabs

At its most basic level, you can think of the R, G and B tabs of the *color calibration* module as a type of matrix multiplication between a 3x3 matrix and the input [R G B] values. This is in fact very similar to what a matrix-based ICC color profile does, except that the user can input the matrix coefficients via the darktable GUI rather than reading the coefficients from an ICC profile file.

$$\begin{bmatrix} R_out \\ G_out \\ B_out \end{bmatrix} = \begin{bmatrix} Rr & Rg & Rb \\ Gr & Gg & Gb \\ Br & Bg & Bb \end{bmatrix} X \begin{bmatrix} R_in \\ G_in \\ B_in \end{bmatrix}$$

If, for example, you've been provided with a matrix to transform from one color space to another, you can enter the matrix coefficients into the *channel mixer* as follows:

- select the *R* tab and then set the Rr, Rg & Rb values using the R, G and B input sliders
- select the *G* tab and then set the Gr, Gg & Gb values using the R, G and B input sliders
- select the *B* tab and then set the Br, Bg & Bb values using the R, G and B input sliders

By default, the mixing function in *color calibration* just copies the input [R G B] channels straight over to the matching output channels. This is equivalent to multiplying by the identity matrix:

$$\begin{bmatrix} R_out \\ G_out \\ B_out \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} X \begin{bmatrix} R_in \\ G_in \\ B_in \end{bmatrix}$$

For a more intuitive understanding of how the mixing sliders on the R, G and B tabs behave, consider the following:

- for the *R* destination channel, adjusting sliders to the right will make the R, G or B areas of the image more "red". Moving the slider to the left will make those areas more "cyan".
- for the *G* destination channel, adjusting sliders to the right will make the R, G or B areas of the image more "green". Moving the slider to the left will make those areas more "magenta".
- for the *B* destination channel, adjusting sliders to the right will make the R, G or B areas of the image more "blue". Moving the slider to the left will make those areas more "yellow".

R, G and B tab controls

The following controls are shown for each of the R, G and B tabs:

input R/G/B
Choose how much the input R, G and B channels influence the output channel relating to the tab concerned.

normalize channels
Select this checkbox to normalize the coefficients to try to preserve the overall brightness of this channel in the final image as compared to the input image.

brightness and colorfulness tabs

The brightness and colorfulness (color saturation) of pixels in an image can also be adjusted based on the R, G and B input channels. This uses the same basic algorithm that the *filmic rgb* module uses for tone mapping (which preserves RGB ratios) and for mid-tones saturation (which massages them).

saturation algorithm
This control allows you to upgrade the saturation algorithm to the new 2021 version, for edits produced prior to darktable 3.6 – it will not appear for edits that already use the latest version.

colorfulness tab controls

input R/G/B
Adjust the color saturation of pixels, based on the R, G and B channels of those pixels. For example, adjusting the *input R* slider will affect the color saturation of pixels containing a lot of “R” more than pixels containing only a small amount of “R”.

normalize channels
Select this checkbox to try to keep the overall saturation constant between the input and output images.

brightness tab controls

input R/G/B
Adjust the brightness of certain colors in the image, based on the R, G and B channels of those colors. For example, adjusting the *input R* slider will affect the brightness of colors containing a lot of R channel much more than colors containing only a small amount of R channel. When darkening/brightening a pixel, the ratio of the R, G and B channels for that pixel is maintained, in order to preserve the hue.

normalize channels
Select this checkbox to try to keep the overall brightness constant between the input and output images.

gray tab

Another very useful application of *color calibration* is the ability to mix the channels together to produce a grayscale output – a monochrome image. Select the *gray* tab, and set the R, G and B sliders to control how much each channel contributes to the brightness of the output. This is equivalent to the following matrix multiplication:

$$GRAY_{out} = \begin{bmatrix} r & g & b \end{bmatrix} \times \begin{bmatrix} R_{in} \\ G_{in} \\ B_{in} \end{bmatrix}$$

When dealing with skin tones, the relative weights of the three channels will affect the level of detail in the image. Placing more weight on R (e.g. [0.9, 0.3, -0.3]) will make for smooth skin tones, whereas emphasising G (e.g. [0.4, 0.75, -0.15]) will bring out more detail. In both cases the B channel is reduced to avoid emphasising unwanted skin texture.

gray tab controls

input R/G/B
Choose how much each of the R, G and B channels contribute to the gray level of the output. The image will only be converted to monochrome if the three sliders add up to some non-zero value. Adding more B will tend to bring out more details, adding more R will tend to smooth skin tones.

normalize channels
Select this checkbox to try to keep the overall brightness constant as the sliders are adjusted.

area color mapping

The area mapping feature is designed to help with batch-editing a series of images in an efficient way. In this scenario, you typically develop a single reference image for the whole batch and then copy&paste the development stack to all of the other images in the batch.

Unfortunately, the color temperature of the illuminating light often changes slightly between shots, even within the same series captured in the same conditions. This can be the result of a cloud passing by the sun in natural light, or a different ratio between colored bounce light and main light. Each image will still need some individual fine-tuning if you want a perfectly even look over the whole series, and this can be both time-consuming and frustrating.

Area color mapping allows you to define a target chromaticity (hue and chroma) for a particular region of the image (the control sample), which you then match against the same target chromaticity in other images. The control sample can either be a critical part of your subject that needs to have constant color, or a non-moving and consistently-lit surface over your series of images.

The mapping process consists of two steps.

step 1: set the target

There are two ways of setting the target chromaticity for your control sample:

1. if you know or expect an arbitrary color for the control sample (for example, a gray card, a color chart, a product or a logo of a specified color), you can set its L, h and c values directly, in Lch derived from CIE Lab 1976 space,
2. if you simply want to match the development of your reference image, set the *area mode* to *measure*, then enable the picker (to the right of the color patch) and draw a rectangle over your control sample. The *input* column will then be updated with the L, h, c values of the control sample before the color correction, and the *target* column will show the resulting L, h, c values of the control sample after the current calibration setting is applied.

If you reset the L, h, c values, the default value is a neutral color at 50% lightness (middle-gray) – this can be useful to quickly set the average white balance of any image. If you want to match the control sample against neutral gray, you only need to reset the chroma slider because the lightness and hue settings have no effect on chromaticity for neutral grays.

Note that the target value is not reset when you reset the module itself, but is stored indefinitely in darktable's configuration and will be available on next launch as well as for the next image you develop. Additionally, the position of the selected rectangle is also stored for ease of use when applying the picked correction value to multiple similar images (see below under "match the target"), though you may redraw the rectangle if the position of the target object changes between images.

The *take channel mixing into account* option lets you choose where the target is sampled. If disabled, the target color is measured immediately after the *CAT* (Chromatic Adaptation Transform) step, which takes place before any channel mixing. This means that if you have a calibrated profile in effect within the channel mixer, this profile will be discarded. If enabled, the target color is measured after the *CAT* and the channel mixing steps, including any calibrated profile. This is the recommended option for most use cases.

Note: If you are defining your target from a gray patch, you should know that the gray patch on color checkers is never entirely neutral. For example, Datacolor Spyder has a slightly warm gray (hue = 20°, chroma = 1.2) while X-Rite pre-2014 has a colder but more neutral gray (hue = 240°, chroma = 0.3) and X-Rite post-2014 is almost perfectly neutral (hue = 133°, chroma = 0.2). In general, it is not desirable to match the control sample against a perfectly neutral gray target, and it is actually wrong to do so when using gray cards and color checkers as a control sample.

---

step 2 : match the target

When you open a new image, the *area mode* is automatically reset to *correction*. Using the picker attached to the color patch, you can then directly reselect your control sample in the new image. The proper illuminant settings required for the control sample to match the memorized target chromaticity will be automatically computed, and the setting will be updated in the same operation.

The *take channel mixing into account* option will need to be set the same as when the measurement of the target was performed to ensure consistent results. Note that the target matching only defines the illuminant settings used in the Chromatic Adaptation Transform – it does not alter the channel mixer settings, since the calibration is handled in the color checker calibration tool. However, the channel mixer settings can be used or discarded in the computation of the illuminant settings, depending on this option.

This operation can be repeated as many times as you have images in your series with no further work.

step 3: deactivate color mapping

The settings you configured in *step 1* are “sticky” – they will stay active until you manually turn them off by resetting *lightness* to 50 and *chroma* to 0. Until then, *every* time you use this module (even after closing and restarting darktable), those settings will affect the results of an auto-whitebalance operation. To remind you that color mapping is active, especially while the section is collapsed, the heading will change from “area color mapping” to “area color mapping (active)” whenever *chroma* is nonzero or *lightness* is other than 50.

Note: Perfectly matching your control sample against the target chromaticity may still not yield a similar perceptual result, even if the numbers are exactly the same. The ratio of lightness between the control sample and its surrounding, as well as the color contrasts at play in the frame, will alter the perception of colors in ways that are very difficult to model. To build an intuition of this problem, see the gray strawberries illusion .

---

extracting settings using a color checker

Since the channel mixer is essentially an RGB matrix (similar to the *input color profile* used for RAW images) it can be used to improve the color accuracy of the input color profile by computing ad-hoc color calibration settings.

These computed settings aim to minimize the color difference between the scene reference and the camera recording in a given lighting situation. This is equivalent to creating a generic ICC color profile but here, the profile is instead stored as module settings that can be saved as presets or styles, to be shared and reused between images. Such profiles are meant to complement and refine the generic input profile but do not replace it.

This feature can assist with:

- handling difficult illuminants, such as low CRI light bulbs, for which a mere white balancing will never suffice,
- digitizing artworks or commercial products where an accurate rendition of the original colors is required,
- neutralizing a number of different cameras to the same ground-truth, in multi-camera photo sessions, in order to obtain a consistent base look and share the color editing settings with a consistent final look,
- obtaining a sane color pipeline from the start, nailing white balance and removing any bounced-light color cast at once, with minimal effort and time.

supported color checker targets

Users are not currently permitted to use custom targets, but a limited number of verified color checkers (from reputable manufacturers) are supported:

- X-Rite / Gretag MacBeth Color Checker 24 (pre- and post-2014),
- Datacolor SpyderCheckr 24 (pre- and post-2018),
- Datacolor SpyderCheckr 48 (pre- and post-2018),
- Datacolor SpyderCheckr Photo.

Users are discouraged from obtaining cheap, off-brand, color targets as color constancy between batches cannot possibly be asserted at such prices. Inaccurate color checkers will only defeat the purpose of color calibration and possibly make things worse.

IT7 and IT8 charts are not supported since they are hardly portable and not practical for use on-location for ad-hoc profiles. These charts are better suited for creating generic color profiles, undertaken using a standard illuminant, for example with Argyll CMS .

Note: X-Rite changed the formula of their pigments in 2014 and Datacolor in 2018, which slightly altered the color of the patches. Both formulas are supported in darktable, but you should be careful to choose the correct reference for your target. If in doubt, try both and choose the one that yields the lowest average delta E after calibration.

---

prerequisites

In order to use this feature you will need to take a test shot of a supported color checker chart, on-location, under appropriate lighting conditions:

- frame the chart in the center 50% of the camera’s field, to ensure that the image is free of vignetting,
- ensure that the main light source is far enough from the chart to give an even lighting field over the surface of the chart,
- adjust the angle between the light, chart and lens to prevent reflections and gloss on the color patches,
- for the best quality profile you should capture an image with the appropriate brightness. To achieve this, take a few bracketed images (between -1 and +1 EV) of your color checker and load them into darktable, ensuring that all modules between *color calibration* and *output color profile* are disabled. Choose the image where the white patch has a brightness L of 94-96% in CIE Lab space or a luminance Y of 83-88% in CIE XYZ space (use the global color picker). This step is not strictly necessary – alternatively you can take a single image and apply the exposure compensation as recommended in the profile report.

If the lighting conditions are close to a standard D50 to D65 illuminant (direct natural light, no colored bounced light), the color checker shot can be used to produce a generic profile that will be suitable for any daylight illuminant with only a slight adjustment of the white balance.

If the lighting conditions are peculiar and far from standard illuminants, the color checker shot will be only usable as an ad-hoc profile for images taken in the same lighting conditions.

usage

The settings used in color calibration depend on the chosen CAT space and on any color settings defined earlier in the pipe within the *white balance* and *input color profile* modules. As such, the results of the profiling (e.g. the RGB channel mixing coefficients) are valid only for a rigid set of *CAT space*, *white balance* and *input color profile* settings. If you wish to create a generic style with your profile, don't forget that you will need to include the settings from these modules as well.

Use the following process to create your profile preset/style:

1. Enable the *lens correction* module to correct any vignetting that might mislead the calibration,
2. On the bottom of the *color calibration* module, click on the arrow near the *calibrate with a color checker* label, to show the controls,
3. Pick the correct model and manufacturer of your color checker from the *chart* drop-down,
4. In the image preview, an overlay of the chart's patches will appear. Drag the corners of the chart so that they match the visual references (dots or crosses) around the target, to compensate for any perspective distortion,
5. Click the *refresh* button to compute the profile,
6. Check the *Profile quality report*. If it is "good", you can click on the *validation* button. If not, try changing the optimization strategy and refresh the profile again.
7. Save the profile in a preset or style, or simply copy & paste the module settings to all of the images taken under the same lighting conditions, from within the lighttable view or filmstrip.

Note: You don't *need* to use the standard matrix in the *input color profile* module when performing a calibration, but be aware that the "as shot in camera" default white balance will not work properly with any other profile, and that you will need to always use the same input profile whenever you reuse such calibration settings.

---

reading the profile report

The profile report helps you to assess the quality of the calibration. The settings in color calibration are only a "best fit" optimization and will never be 100% accurate for the whole color spectrum. We therefore need to track "how inaccurate" it is in order to know whether we can trust this profile or not.

Bad profiles can happen and will do more harm than good if used.

delta E and the quality report

The CIE delta E 2000 (ΔE) is used as a perceptual metric of the error between the reference color of the patches and the color obtained after each step of calibration:

- ΔE = 0 means that there is no error – the obtained color is exactly the reference color. Unfortunately, this will never happen in practice.
- ΔE = 2.3 is defined as the Just Noticeable Difference (JND).
- ΔE < 2.3 means that the average observer will not be able to tell the difference between the expected reference color and the obtained color. This is a satisfactory result.
- ΔE > 2.3 means that the color difference between the expected reference and the obtained color is noticeable for the average observer. This is unsatisfactory but sometimes unavoidable.

The quality report tracks the average and maximum ΔE at the input of the module (before anything is done), after the chromatic adaptation step (white balance only), and at the output of the module (white balance and channel mixing). At each step, the ΔE should be lower than at the previous step, if everything goes as planned.

profile data

The data generated by the profiling process comprises the RGB 3×3 matrix and the detected illuminant. These are expressed in the CAT *adaptation* space defined in the *CAT* tab and are provided in case you want to export these coefficients to other software. If the detected illuminant is *daylight* or *black body*, the matrix should be fairly generic and reusable for other *daylight* and *black body* illuminants with the addition of a small white balance adjustment.

normalization values

These are the settings that you should define, as-is, for the *exposure* and *black level correction* parameters in the *exposure* module, in order to obtain the lowest possible error in your profile. This step is optional and is useful only when the utmost precision is required, but beware that it can produce negative RGB values that will be clipped at various points in the pipeline.

overlay



The chart overlay displays a disc in the center of each color patch, which represents the expected reference value of that patch, projected into the display RGB space. This helps you to visually assess the difference between the reference and the actual color without having to bother with ΔE values. This visual clue will be reliable only if you set the *exposure* module as instructed in the *normalization values* of the profile report.

Once the profile has been calibrated, some of the square patches will be crossed in the background by one or two diagonals:

- patches that are not crossed have $\Delta E < 2.3$ (JND), meaning they are accurate enough that the average observer will be unable to notice the deviation,
- patches crossed with one diagonal have $2.3 < \Delta E < 4.6$, meaning that they are mildly inaccurate,
- patches crossed with two diagonals have $\Delta E > 4.6$ ($2 \times$ JND), meaning that they are highly inaccurate.

This visual feedback will help you to set up the optimization trade-off to check which colors are more or less accurate.

enhancing the profile

Because any calibration is merely a “best fit” optimization (using a weighted least-squares method) it is impossible to have all patches within our ΔE < 2.3 tolerance. Some compromise will therefore be required.

The *optimize for* parameter allows you to define an optimization strategy that attempts to increase the profile accuracy in some colors at the expense of others. The following options are available:

- *none*: Don’t use an explicit strategy but rely on the implicit strategy defined by the color checker manufacturer. For example, if the color checker has mostly low-saturation patches, the profile will be more accurate for less-saturated colors.
- *neutral colors*: Give priority to grays and less-saturated colors. This is useful for desperate cases involving cheap fluorescent and LED lightings, having low CRI. However, it may increase the error in highly-saturated colors more than not having any profile.
- *saturated colors*: Give priority to primary colors and highly-saturated colors. This is useful in product and commercial photography, to get brand colors right.
- *skin and soil colors*, *foliage colors*, *sky and water colors*: Give priority to the chosen hue range. This is useful if the subject of your images is clearly defined and has a typical color.
- *average delta E*: Attempt to make the color error uniform across the color range and minimize the average perceptual error. This is useful for generic profiles.
- *maximum delta E*: Attempt to minimize outliers and large errors, at the expense of the average error. This can be useful to get highly saturated blues back into line.

No matter what you do, strategies that favor a low average ΔE will usually have a higher maximum ΔE, and vice versa. Also, blues are always the more challenging color range to get correct, so the calibration usually falls back to protecting blues at the expense of everything else, or everything else at the expense of blues.

The ease of obtaining a proper calibration depends on the quality of the scene illuminant (daylight and high CRI illuminants should always be preferred), the quality of the primary *input color profile*, the *black point compensation* set in the *exposure* module, but first and foremost on the mathematical properties of the camera sensor’s filter array.

profile checking

It is possible to use the *color space check* button (first on the left, at the bottom of the module) to perform a single ΔE computation of the color checker reference against the output of the *color calibration* module. This can be used in the following ways:

1. To check the accuracy of a profile calculated in particular conditions against a color checker shot in different conditions.
2. To evaluate the performance of any color correction performed earlier in the pipe, by setting the *color calibration* parameters to values that effectively disable it (CAT *adaptation* to *none*, everything else set to default), and just use the average ΔE as a performance metric.

caveats

The ability to use standard CIE illuminants and CCT-based interfaces to define the illuminant color depends on sound default values for the standard matrix in the *input color profile* module as well as reasonable RGB coefficients in the *white balance* module.

Some cameras, most notably those from Olympus and Sony, have unexpected white balance coefficients that will always make the detected CCT invalid even for legitimate daylight scene illuminants. This error most likely comes from issues with the standard input matrix, which is taken from the Adobe DNG Converter.

It is possible to alleviate this issue, if you have a computer screen calibrated for a D65 illuminant, using the following process:

1. Display a white surface on your screen, for example by opening a blank canvas in any photo editing software you like
2. Take a blurry (out of focus) image of that surface with your camera, ensuring that you don’t have any “parasite” light in the frame, you have no clipping, and are using an aperture between f/5.6 and f/8,
3. Open the image in darktable and extract the white balance by using the picker tool in the *white balance* module on the center area of the image (non-central regions might be subject to chromatic aberrations). This will generate a set of 3 RGB coefficients.
4. Save a preset for the *white balance* module with these coefficients and auto-apply it to any color RAW image created by the same camera.

8.2.11. color contrast

Technical information

description
increase saturation and separation between opposite colors.
purpose
creative.
input
non-linear, Lab, display-referred.
processing
non-linear, Lab.
output
non-linear, Lab, display-referred.

A simplified control for changing the contrast or separation of colors between the green/magenta and blue/yellow axes in Lab color space. Higher values increase color contrast, lower values decrease it.

module controls

green-magenta contrast
Change the green-magenta color contrast. This is equivalent to raising or lowering the steepness of the a* curve in Lab. Lower values desaturate greens and magenta, while higher values increase their saturation.
blue-yellow contrast
Change the blue-yellow color contrast. This is equivalent to raising or lowering the steepness of the b* curve in Lab. Lower values desaturate blues and yellows, while higher values increase their saturation.

8.2.12. color correction

Technical information

description
correct white balance selectively for blacks and whites.
purpose
corrective or creative.
input
non-linear, Lab, display-referred.
processing
non-linear, Lab.
output
non-linear, Lab, display-referred.

Modify the global image saturation to give the image a tint or as an alternative method to split-tone the image.

Note: *Use the color balance rgb module for color modifications.*

---

module controls

color board
For split toning move the white dot to the desired highlight tint and then select a tint for shadows with the dark spot. For a simple global tint set both spots to the same color.
saturation
Correct the global saturation.

8.2.13. color equalizer

Technical information

description
change saturation, hue and brightness depending on local hue.
purpose
corrective and creative.
input
linear, RGB, scene-referred.
processing
quasi-linear, RGB.
output
quasi-linear, RGB, scene-referred.

Selectively adjust the hue, saturation, and/or brightness of pixels based on their current hue.

This module is an attempt to recreate *some* of the functionality of the *color zones* module in the scene-referred part of the pipeline whilst overcoming some of that module's limitations. Specifically, the *color zones* module is prone to increase chroma/luma noise, and introduce artifacts or harsh transitions. These are somewhat mitigated within *color equalizer* with the following additions:

- a fixed number of equally-spaced adjustment nodes means a smoother adjustment curve
- a guided filter is used, which attempts to smooth out changes by taking into account both the hue of the neighboring pixels and their overall saturation

The provided tabs can be used (either individually or in combination) to adjust the hue, saturation, and/or brightness at each of the color nodes. While the distance between nodes cannot be changed (see above), if you wish to target a specific hue that does not currently lie on a node, the "node placement" slider can be used to move all of the nodes simultaneously. The picker can be used to display the color of a "picked" pixel or area on the curve adjustment section of the module to allow specific colors to be targeted.

Note: While the above mitigations can reduce the noise/artifacts generated by this module, they will not always be entirely successful, and you are therefore advised to use this module with care. Try not to make very large changes, and look at some of the options for tweaking the guided filter (below) if introduced noise is excessive.

---

module controls

hue/saturation/brightness adjustment curves

The main controls of this module are the hue/saturation/brightness adjustment curves. Use each of these tabs to adjust the hue, saturation, or brightness of pixels based on their current hue. Click and drag the color nodes (up/down) to alter the properties of pixels with the given hue. You can also adjust nodes using the scroll wheel on your mouse and, as with sliders, you can change the speed of the adjustment by combining the scroll with Ctrl (to adjust more slowly) or Shift (to adjust more quickly).

You can also adjust the color nodes using sliders, which can be shown/hidden by middle-clicking on the curve adjustment section of the module with your mouse or by using the *control sliders* button.

As with sliders, you can also right-click on a node/slider to perform fine adjustments (see module controls/sliders for more details).

In addition, you can hover the cursor over the image and scroll to adjust the value (see *cursor indicator/control* for details).

Use the picker to choose a pixel or area on the image and show the hue of that pixel/area on the adjustment curve.

node placement
Adjust the position of all of the nodes simultaneously (move them left/right).

Note: The slider color names will not change when you move the nodes with the “node placement” slider.

options

general

white level

Set the upper bound of brightness corrections that can be made with the module – the module will not allow brightness of any pixels to be greater than this value. Use the picker to set based on a point/area on the image. This should usually be kept at its default value.

hue curve (*N/A for saturation/brightness curves*)

Control how the hue adjustment curve is interpolated between nodes. Values greater than 1 make the transitions between the nodes more gradual. Values less than 1 create sharper transitions, allowing for more precise adjustments to individual hues, while affecting neighboring areas/colors less. Be aware that lower values can increase the likelihood of introducing noise or artifacts.

guided filter

In order to reduce the amount of noise/artifacts, this module uses a guided filter by default. The guided filter attempts to smooth out changes by taking into account both the hue of the neighboring pixels and their overall saturation. This guided filter is ultimately a compromise between adjustment precision (targeting colors more specifically) and avoidance of artifacts (which will generally require less precise targeting). The following options allow the guided filter to be adjusted and visualised in order to better manage these compromises:

use guided filter

Toggle the guided filter on/off.

hue analysis radius

Choose how much to take neighboring pixels into account when accounting for hue in the guided filter. The default value (1.5px) is a compromise intended to ensure that individual “noisy” pixels (whose color differs significantly from their neighbors) don’t become more noisy as a result of any adjustments. Increase this value to smooth out any introduced noise (such as Moiré). However, be aware that very large values can cause halos and other artifacts at edges.

saturation threshold

Since this module is intended to perform operations based on a pixel’s hue, it is usually desirable to restrict its operation to only those pixels that have the highest saturation (and for it to have more of an effect on more highly-saturated pixels). The saturation threshold can be used to choose how saturated a pixel must be before its properties can be adjusted by the module. Decrease to allow adjustment of less-saturated pixels. Increase to restrict adjustment of less-saturated pixels.

visualize weighting function (saturation)

The first “visualization” button (to the right of the “saturation threshold” slider) can be used to see which pixels might be impacted by the operation of this module, based on their saturation, and by how much. Pixels with lower saturation, which will not be adjusted, are shown in shades of blue. Pixels with higher saturation, which might be adjusted (depending on the adjustment curves), are shown in shades of red. The more saturated the red/blue, the more/less (respectively) a pixel will be affected by any adjustment.

This visualization only depends on a pixel’s saturation, and will not change when you alter the hue/saturation/brightness adjustment curves. It is also affected by the “contrast” setting (below).

contrast

Regulate the steepness of the saturation adjustment curve. This is used in conjunction with the “saturation threshold” slider to determine how much a pixel might be adjusted based on its saturation. The generated curve is shown overlaid on the image when the “visualise weighting function” button is toggled. Positive values result in a steeper transition (highly-saturated pixels are more likely to be changed, low-saturation pixels are less likely to be changed). Negative values result in a shallower transition (highly-saturated pixels are less likely to be changed, low-saturation pixels are more likely to be changed)

effect radius

Set the radius of the applied parameters in the guided filter. Larger values will smooth the effects of any adjustments, and will better retain pre-existing detail within the image, though can cause bleeding at edges that will need to be mitigated by other means.

visualize changed output for the selected tab

The second “visualization” button (to the right of the “effect radius” slider) can be used to visualize the overall effect of the adjustments made in the current tab, taking into account both the guided filter and the adjustment curve. Red pixels indicate where the value of the chosen adjustment (hue/saturation/brightness) has increased. Blue pixels indicate where the value of the adjustment has decreased.

control sliders

Click *control sliders* to toggle the display of the sliders.

cursor indicator/control

When the *color equalizer* module is enabled and expanded, you can move the mouse pointer over the image to show a cursor that displays information about the pixel color under the pointer. It will also highlight the corresponding hue in the open tab. When this cursor is shown, the mouse wheel can be used to alter the corresponding hue, saturation, or brightness band in the module.



- the cross-hairs indicate the position of the pixel under the cursor.
- the text label shows the value of the change in hue, saturation, or brightness (depending on which tab is selected) for the pixel color under the cursor.
- if the color equalizer has altered the color at the current cursor position, the magnitude of the adjustment is indicated by an arc on the left-hand-side. The longer the arc, the greater the adjustment.

If you need to move or zoom the portion of the image shown in the center view while the module is expanded, hold down the ‘A’ key while dragging the mouse or using the scroll wheel. As long as the key is held down, mouse actions adjust the viewport rather than adjusting the color.

8.2.14. color harmonizer

Technical information

description
harmonize colors toward a selected palette in perceptual space.

purpose
creative color grading.

input
linear, RGB, scene-referred.

processing
darktable UCS / JCH (perceptual).

output
linear, RGB, scene-referred.

The color harmonizer reduces chromatic dissonance by pushing hues toward a selected color palette. This palette is defined by a set of hue angles called harmony nodes. Colors that are “off-palette” are gently pulled toward the nearest node; colors already on a node are left unchanged. A per-node saturation multiplier can additionally boost or reduce the colorfulness of colors near each node.

module controls

harmony rule

Select one of the harmony rules. All node positions are offsets from the anchor hue. Angles are in degrees.

Rule	Nodes	Node positions	Character
Monochromatic	1	0°	Single hue family
Analogous	3	0°, −30°, +30°	Adjacent neighbors; naturalistic, cohesive
Analogous complementary	4	0°, −30°, +30°, +180°	Analogous triad plus its opposite
Complementary	2	0°, +180°	Direct opposites; maximum contrast
Split complementary	3	0°, +150°, +210°	Anchor plus both neighbors of its complement
Dyad	2	−30°, +30°	Anchor is symmetry axis, not a node
Triad	3	0°, +120°, +240°	Evenly spaced; balanced, colorful
Tetrad	4	−30°, +30°, +150°, +210°	Two dyad pairs; anchor is symmetry axis, not a node
Square	4	0°, +90°, +180°, +270°	Four equally spaced hues
Custom	2–4	Freely placed	User-defined node positions

Note on dyad and tetrad: the anchor hue sets the symmetry axis of the pattern, not the position of a node. The palette is symmetric around the anchor. This matches the vectorscope harmony guide overlay.

 **infer from image *(standard modes only)***

Analyses the preview image's chroma-weighted hue histogram and automatically selects the harmony rule and anchor hue that best fit the image's existing color distribution — i.e. the combination that already covers the most chromatic energy and therefore requires the least correction.

anchor hue

The primary hue from which all node positions are derived (not shown in *custom* mode). Expressed as a normalized value displayed in degrees. An eyedropper  is available to sample a color directly from the image. The color swatch strip below the slider shows the actual node colors for quick visual feedback.

vectorscope two-way sync

When enabled (default), any change to the harmony rule, anchor hue, or custom node positions is immediately reflected in the vectorscope harmony overlay, and changes made in the vectorscope, i.e. changing the overlayed rule or rotating the overlay, are reflected back in the module. Disable to make adjustments without disturbing the vectorscope display. If enabled, using the module will automatically switch the scopes module to show the vectorscope.

 **import from vectorscope *(standard modes only)***

Imports the harmony rule and anchor hue currently configured in the vectorscope panel, then switches the histogram panel to the vectorscope view.

nodes *(custom mode only)*

Number of active harmony nodes in custom mode (2–4). Only the first N node rows are shown and active; the others are hidden.

node hue sliders *(custom mode only)*

One row per active node. Each row shows a color swatch and a hue slider with an eyedropper. The hue slider sets the node position on the hue wheel (0°–360°). Use the eyedropper to sample a desired hue directly from the image.

Effect controls

pull strength

Colors will be shifted depending on their proximity to the nodes. The shift is strongest for pixels between nodes, and weakest (or zero) for pixels exactly on a node.

Note: pull strength scales the **hue** correction only. The saturation correction (per-node saturation multipliers) is applied independently, at full strength regardless of this slider.

pull width

Scales the width of each node’s attraction zone.

- **< 1 (narrow):** quick decay; only hues very close to a node are attracted. The rest of the hue wheel is barely touched. Useful for images already close to harmonic, or when precise, surgical correction of specific hues is needed.
- **= 1 (default):** medium decay — clean zone separation with smooth transitions.
- **> 1 (wide):** no decay; all pixels are attracted noticeably regardless of how far they are from a node. Useful to suppress colors outside the target harmony.

  Increasing pull width never displaces pixels that are already exactly on a harmony node — their angular distance to the nearest node is zero, so their hue shift is always zero.

neutral color protection

Shields low-chroma pixels from correction.

The slider sets how aggressively low-chroma pixels are exempted: low values protect only near-absolute grays; high values extend protection to muted and pastel tones. Even at the maximum, vivid colors remain largely unaffected.

Default: 0.50.

smoothing

This smooths the spatial transitions at zone boundaries — visible as subtle color steps in smooth gradients like skies or skin when adjacent regions fall in different attraction zones.

The blur sigma scales with both this slider and the current pull width: wider attraction zones span larger hue ranges and generate larger corrections, which can produce sharper spatial boundaries when pixels near each other fall on opposite sides of a zone midpoint.

- **0 (default, off):** transitions are handled purely by the Gaussian interpolation in hue space. Maximum detail is preserved. Recommended for most uses.
- **0.1–0.5:** subtle spatial averaging to reduce color noise in smooth areas.
- **> 1.0:** broad spatial blending; can create a painterly effect but may bleed corrections across image edges.

Saturation section *(collapsible)*

One row per active harmony node. Each row shows a color swatch (the node’s hue) and a saturation multiplier slider.

node saturation sliders

Saturation multiplier for colors near the corresponding harmony node.

- **100 % (default):** saturation is unchanged.
- **< 100 %:** desaturates colors near this node, proportionally to their Gaussian proximity.
- **> 100 %:** boosts saturation near this node.

The effect is weighted by the pixel’s Gaussian proximity to the node and further modulated by **neutral color protection** so near-achromatic pixels are spared. At the node itself (maximum weight) the saturation is multiplied by exactly the slider value.

Usage guide

Basic workflow

When the module is first activated it is in a neutral state and so there will be no visible change in the image.

1. **Set the anchor hue.** Use the eyedropper to pick the dominant or most important color in the scene — a sky, a garment, a skin tone — or drag the slider manually while watching the swatch strip.
2. **Choose a harmony rule.** For portraiture, *complementary* (subject vs. background) or *analogous* (warm tones) are natural starting points. For landscapes, *triad* or *split complementary* often work well. Enable *vectorscope two-way sync* (under the harmony rule heading) to see the palette on the vectorscope while you choose.
3. **Raise pull strength.** Start around 0.2–0.3. The effect is graduated.
4. **Adjust pull width if needed.** If many off-palette hues are not moving enough, widen the zones. If you want to correct only specific hue ranges without touching the rest, narrow them. The result may be very subtle if the image already has an inherent harmony and the module is set to match. In this case, setting both strength and width to their maximum values and then reducing them gradually may be a more effective strategy.
5. **Use neutral color protection to taste.** Raise it for images with many desaturated tones (architecture, faded film looks) to exempt pastels and muted tones from correction.
6. **Use the saturation section for palette polishing.** Once the hue pull is set, open the saturation section to boost or calm chroma per node — e.g. boost the warm nodes and desaturate the cool ones for a cinematic split-tone look.

Custom harmony mode

Use *custom* when none of the geometric rules match the image's intended palette.

1. Set the harmony rule to *custom*. The previous rule's node positions are copied as a starting point.
2. Use *nodes* to select how many palette targets you need (2–4).
3. Drag each node's hue slider or use its eyedropper to position nodes at the desired palette hues.
4. Raise pull strength to pull off-palette colors toward the custom nodes.

Custom nodes are synced to the vectorscope as free-floating angle markers (the vectorscope shows no standard rule name for custom palettes).

Working with the vectorscope

With *vectorscope two-way sync* enabled, the vectorscope overlays the harmony guide on the image's color distribution. Colors inside the guide arcs are on-palette; colors outside are being corrected. Use this to verify the pull is moving in the intended direction. Aside from using the hue slider the anchor hue can be moved by scrolling with the cursor over the vectorscope.

To start from a palette already set in the vectorscope panel, click the *import from vectorscope* button ().

8.2.15. color look up table

Technical information

description
perform color space corrections and apply looks.
purpose
corrective or creative.
input
linear or non-linear, Lab, display-referred.
processing
defined by profile, Lab.

output
linear or non-linear, Lab, display-referred.

A generic color look up table implemented in Lab space.

The input to this module is a list of source and target points and the complete mapping is interpolated using splines. The resulting look up tables (LUTs) are editable by hand and can be created using the darktable-chart utility to match given input (such as hald-cluts and RAW/JPEG with in-camera processing pairs). See using darktable-chart for details.

module controls

color board
The color board grid shows a list of colored patches. The colors of the patches are the source points. The target color of the selected patch is shown as offsets which are controlled by sliders beneath the color board. An outline is drawn around patches that have been altered (where the source and target colors differ).

Click a patch to select it, or use the combo box or picker . The currently-selected patch is marked with a white square, and its number is displayed in the combo box below.

By default, the module will load the 24 patches of a classic color checker and initialise the mapping to identity (no change to the image). Configurations with more than 24 patches are shown in a 7x7 grid.

interaction
To modify the color mapping, you can change the source and target colors, though the main use is to change the target colors.

Start with an appropriate palette of source colors (either from the presets menu or from a style you have downloaded). You can then change the lightness (L), green-red (a), blue-yellow (b), or saturation (c) of the patches' target values with the sliders.

To change the source color of a patch you can select a new color from your image by using the picker and Shift+click on the patch you want to replace.

Double-click a patch to reset it; Right-click a patch to delete it; Shift+click on empty space to add a new patch (with the currently picked color as the source).

8.2.16. color mapping

Technical information

description
transfer a color palette and tonal repartition from one image to another.
purpose
creative.
input
linear or non-linear, Lab, display-referred.
processing
non-linear, Lab.

output
non-linear, Lab, display-referred.

Transfer the look and feel of one image to another.

This module statistically analyses the color characteristics of a source and target image. The colors of the source image are then mapped to corresponding colors on the target image.

Two steps are required to use this module:

1. Open the source image in the darkroom view and acquire its color characteristics by pressing the “acquire as source” button. A set of color clusters is generated and displayed in the “source clusters” area. Each cluster is represented by a set of color swatches with the mean value in the center surrounded by swatches indicating the variance within that cluster. The clusters are sorted in ascending order by their weight (the number of pixels that contribute to each of them).
2. Open the target image in the darkroom view. The previously collected source clusters should be stored; if they are not yet displayed, press the reset button. Now press the “acquire as target” button to generate a corresponding set of color clusters for your target image, which are then displayed in the “target clusters” area.

When both source and target clusters are collected an automatic color mapping is applied to the target image. In its default settings the overall effect is quite exaggerated. A set of sliders gives you control of the effect’s strength. You can also use the *normal* blend mode along with the *opacity* slider to tame the effect. As the color mapping module comes early in the pixelpipe, you can also finetune the colors with later modules.

module controls

acquire as source/target
Press these buttons to generate color clusters for the source and target image, respectively. The processing takes a few seconds during which the GUI will be unresponsive.

number of clusters
The number of color clusters to use. This should roughly correlate to the number of dominant colors in the image. The clusters are acquired through a statistical sampling process to represent the colors that occur most commonly throughout the image. The random nature of the sampling process means that you might get different results each time you acquire the clusters, especially if you have a larger number of clusters and/or a complex color palette in the image. If you change this parameter all the collected color clusters are reset and must be re-acquired.

color dominance
This parameter controls the mapping between source and target clusters. At the minimum value, the mapping is based on color proximity. This typically leads to very subtle effects on the target image. At the maximum value, the mapping is based on the relative weight of the color clusters – dominant colors of the source image are mapped to dominant colors of the target image. This typically leads to a very strong effect. Intermediate values incrementally shift between these extremes.

histogram equalization
Modify the target image’s contrast by matching its histogram with the histogram of the source image. This slider controls the extent of this effect.

8.2.17. color reconstruction

Technical information

description
recover clipped highlights by propagating surrounding colors.

purpose
corrective.

input
linear or non-linear, Lab, display-referred.

processing
non-linear, Lab.

output
non-linear, Lab, display-referred.

Recover color information in blown-out highlights.

Due to the nature of digital sensors, overexposed highlights lack valid color information. Most frequently they appear neutral white or exhibit some color cast, depending on what other image processing steps are involved. This module can be used to “heal” overexposed highlights by replacing their colors with better fitting ones. The module acts on pixels whose luminance exceeds a user-defined threshold. Replacement colors are taken from neighboring pixels. Both the spatial distance and the luminance distance (range) are taken into account for color selection.

Due to a limitation of the underlying algorithm, reconstructed colors may sometimes be displayed incorrectly if you zoom into the image in the darkroom view. If this happens you might observe a magenta shift in highlight areas close to high contrast edges, or you might see colorless highlight areas when used alongside the “reconstruct color” method of the *highlight reconstruction* module. These artifacts only influence on-screen image display – the final output remains unaffected. It is recommended that you finetune the parameters of this module only when viewing the fully zoomed-out image.

Note that similar functionality is also available in the *reconstruct* tab of the *filmic rgb* module.

module controls

threshold
The color reconstruction module replaces the color of all target pixels having luminance values above this threshold. Only pixels with luminance values below the threshold are taken as valid source pixels for replacement colors. Setting this parameter too high will cause the module to have no effect on any pixels. Setting it too low will minimize the “pool” of replacement colors – if no suitable colors are available, the original colors are retained. This parameter therefore exhibits a “sweet spot” characteristic with an optimum setting depending on the individual image.

spatial extent
The spatial distance that source pixels may have from target pixels in order for them to contribute to color replacement. Higher values cause ever more distant pixels to contribute. This increases the chance of finding a replacement color but can make the replacement color less well suited for the reconstruction.

range extent
The range difference (in luminance) that source pixels may have from target pixels in order for them to contribute to color replacement. Higher values cause more pixels to contribute even if their luminance differs more strongly from the target pixels. This again increases the chance of finding a replacement color but at the same time increases the risk of unfitting colors creeping in.

precedence
This combobox defines whether certain replacement colors shall take precedence over others as follows:

- *off* (default): All pixels contribute equally
- *saturated colors*: Pixels contribute according to their chromaticity – the more highly-saturated a color, the more it contributes
- *hue*: Give precedence to a specific hue (see below)

hue
This slider is only visible if you set the *precedence* to “hue”. It allows you to select a preferred hue for replacement colors. This only has an effect if the preferred hue is actually present within the selected spatial and range distance of the target pixels (see above). A typical use case is repairing highlights on human skin in situations where diverging colors are in close proximity (e.g. textiles or hair with a luminance close that of to skin). Setting a hue preference on skin tones prevents these other colors from creeping in.

8.2.18. color zones

Technical information

description
selectively shift hues, chroma and lightness of pixels.

purpose
creative.

input
linear or non-linear, Lab, display-referred.

processing
non-linear, Lab.
output
non-linear, Lab, display-referred.

Selectively adjust the lightness, chroma and hue of pixels based on their current lightness, chroma and hue.

This module works in CIE LCh color space, which separates pixels into *lightness*, chroma and *hue* components. It allows you to manipulate the lightness, chroma and hue of targeted groups of pixels through the use of curves .

You first need to choose whether you wish to adjust (select) pixels based on their lightness, chroma or hue. You can then use three curves, on their respective tabs, to adjust the lightness, chroma and hue of ranges of pixels selected via this method.

Note: *This module should be used with care, as transitions between colors may not be graceful. Instead, use the color balance rgb module with a parametric mask .*

---

pixel selection method

The *color zones* module offers three different methods for selecting which pixels you want to adjust. They are:

select by hue (default)
Select pixels to manipulate based on their hue. For example, you may want to darken a blue sky, or to change a red porche into a yellow one. The following image shows the full range of hues that you can choose to operate on:



select by lightness
Select pixels to manipulate based on their lightness. For example, you may want to make your shadows brighter, or to make your highlights more yellow. The following image shows the range of lightness levels that you can choose to operate on, from dark to light:



select by chroma
Select pixels to manipulate based on their chroma. For example, you may want to tone down the chroma of some already highly saturated pixels, or to change their hue. The following image shows the range of chroma levels that you can choose to operate on, from a completely unsaturated monochrome gray through to the most highly saturated color:



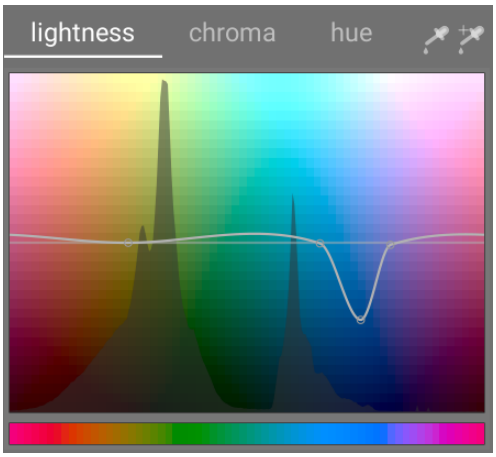
pixel manipulation curves

Once you have chosen a pixel selection method, the selected range of lightness, chroma or hue levels will appear along the horizontal axis of the three pixel manipulation curves, which can be viewed and adjusted by choosing the appropriate tab (lightness, chroma, hue).

If, for example, you were to choose to *select by hue* (the default), the horizontal axis (below the manipulation curves) would show the range of hues you can work with, and the three pixel manipulation curves would appears as follows:

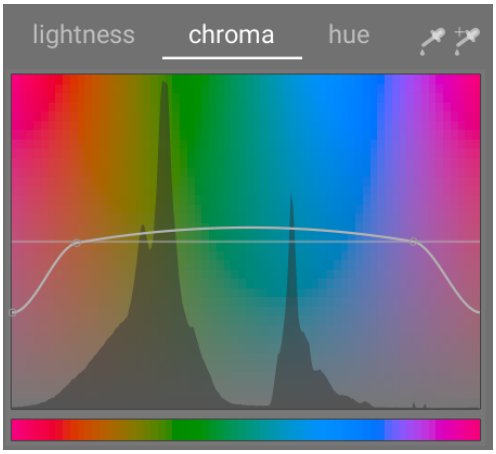
lightness

By adjusting the lightness curve up or down in a given (hue) location, you can brighten or darken pixels matching hues where the curve has been raised or lowered, respectively. The example below reduces the lightness of the blue parts of the image:



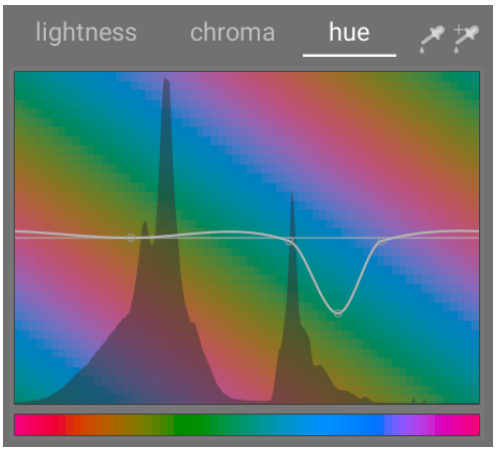
chroma

By adjusting the chroma curve up or down in a given (hue) location, you can desaturate (make less colorful) or resaturate (make more colorful) pixels matching hues where the curve has been raised or lowered, respectively. The example below reduces the chroma of the red parts of the image:



hue

By adjusting the hue curve up or down in a given (hue) location, you can shift the hue of pixels matching hues where the curve has been raised or lowered, allowing you to replace one color with another. The example below shifts the blue parts of the image to green:



The curves work similarly in the lightness- and chroma-based selection modes as well. See the section on curves to see how spline curves work in general.

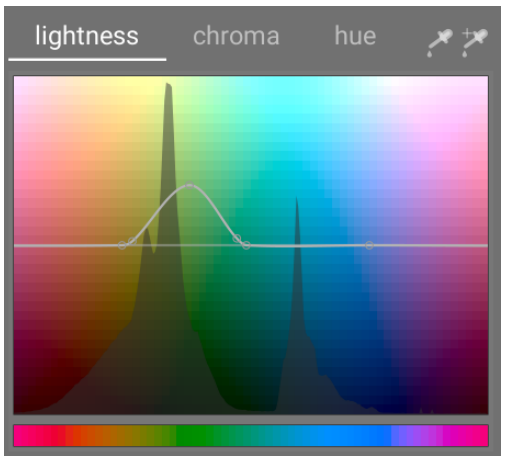
Note that these examples are somewhat contrived in order to illustrate the module's usage. In practical use, they would likely need to be combined with drawn and/or parametric masks to further isolate their effect.

range selection

When adjusting the pixel manipulation curves, it can sometimes be difficult to judge exactly where on the horizontal axis pixels will fall. To the right of the tab controls are a pair of pickers that can be used to assist with this.

If you click the left-hand picker and choose a pixel in the image, you will see a dark vertical line showing where that pixel falls on the horizontal axis. If you Ctrl+click or right-click on the same picker you can choose a rectangular area from the image – the range of values represented within the selected rectangle will be shaded vertically, with a similar dark line showing the median value.

If you click on the right-hand picker, you can similarly choose a rectangular area on the image and the display will be shown as described above (a shaded area with a dark vertical line). However, in this case the picker will also automatically add some control points to the curve for you, representing the highlighted range (see below). Simply drag on the center node to raise or lower the curve within the selected range. Alternatively, hold Ctrl while selecting a range to automatically create a positive curve (push up the selected range) or hold Shift while selecting to create a negative curve (pushed down).



module controls

The following controls are available in the *color zones* module:

lightness, chroma & hue tabs
Each tab displays a pixel manipulation curve to allow you to alter "lightness", "chroma", or "hue" based on the pixel selection method.

edit by area
Choose how to interact with the curve. This setting is disabled by default, allowing the control points for the curve to be freely placed. Check the box to fall back to the legacy “edit by area” mode, which functions in a similar way to the spline curve controls used in wavelet modules.

 **mask display**
Enable the *mask display* to highlight pixels that have been affected by *color zones* adjustments in yellow.

select by
Define the horizontal axis (the range of values to work on). You can choose between “lightness”, “chroma”, and “hue” (the default). Changing this parameter resets all pixel manipulation curves to their default state (horizontal straight lines). If you want to work on multiple ranges, you need to create additional instances of the module.

process mode
Choose between a “smooth” (default) or “strong” processing mode. The default mode is less likely to cause artifacts.

mix
Use this parameter to tune the strength of the overall effect.

interpolation method
Define how the curve is interpolated using the user-defined control points. See the curves section for more details.

8.2.19. colorize

Technical information

description
overlay a solid color on the image.

purpose
creative.

input
linear or non-linear, Lab, display-referred.

processing
non-linear, Lab.

output
non-linear, Lab, display-referred.

Add a solid layer of color to the image.

module controls

hue
The hue of the color layer.

saturation
The saturation of shadow tones.

lightness
The lightness of the color layer.

source mix
Controls how the lightness of the input image is mixed with the color layer. Setting this to zero will result in a uniformly colored plane.

8.2.20. composite

Technical information

description
combine the image with elements from another processed image (move this module to after output color profile if you see banding).

purpose
corrective and creative.

input
linear, RGB, scene-referred.
processing
linear, RGB.
output
linear, RGB, scene-referred.

Create a composite image by overlaying an already-processed image on top of the current image.

Drag and drop a processed image from the filmstrip onto the “drop image from filmstrip here” box to overlay the chosen image, and then alter the various placement and adjustment attributes of the image being overlaid.

Please note the following points:

- This module does not include the entire pixelpipe of the overlaid image – only that part of the pixelpipe up to the composite module. If you wish to include any modules after that point, you should move the composite module within the overlaid image first and complete your edit before dropping it onto your target image.
- Because the darktable will now be processing two images, this could significantly degrade performance, so try to avoid too many large edits on both images.
- The overlaid image is imported with only eight bits per pixel rather than the full floating point values normally used within darktable’s internal pipeline. This means that any major changes made after the composite module might introduce banding. In such cases you are advised to do as much processing as possible before the composite module in the pixelpipe. Move the composite module as needed to achieve this.

module controls

drop image from filmstrip here
The image to overlay onto the currently edited image (dragged from the filmstrip).
opacity
The percentage of transparency applied to the overlaid image.
rotation
The amount to turn the overlaid image. Rotation is around the alignment point.
scale
The amount to shrink the overlaid image.
scale on
The criteria on which to base the overlaid image scaling.
alignment
The area on which to align the overlaid image.
x offset
The amount to move the overlaid image horizontally.
y offset
The amount to move the overlaid image vertically.

8.2.21. contrast & texture

Technical information

description
enhance local contrast by boosting details while preserving edges.
purpose
creative.
input
linear, RGB, scene-referred.
processing
linear, RGB.

output
linear, RGB, scene-referred.

contrast & texture is the scene-referred counterpart to the display-referred *local contrast* module. It can be used for both adding punch to, for example, clouds or foliage and for smoothing busy areas (like softening skin or calming down a cluttered background). It can also be used to manage highlights and shadows as the scene referred equivalent of the display referred *shadows and highlights* .

The module uses the same *exposure-independent guided filter (eigf)* used by the *tone equalizer* module for its guided mask. The affected scale of detail can be tuned with the module's controls, and three separate scale levels can be targeted individually. The underlying filter is exposure-independent, so the strength of the effect stays consistent across shadows and highlights alike.

A noise bias control lets you tame the amplification of shadow noise, which would otherwise be boosted along with genuine detail since both look like local contrast to the algorithm.

To build up a more elaborate effect, multiple instances of *contrast & texture* can be used together, each targeting different detail scales and possibly masked to different parts of the image.

Multiple controls have a mask display button  available - only one mask can be displayed at a time, so clicking on a mask display button turns off any other mask currently being displayed.

Note: This module is a first step towards a more fully-featured scene-referred local contrast tool including multiple scales within one module instance and shadows/highlights control. Feedback is welcome at the pixls.us forum .

---

module controls

base detail level
Adjust the base detail level, which determines the scale targeted by highlights, shadows, and coarse, medium and fine details.

- higher values: affect local contrast in finer details.
- lower values: affect local contrast in coarser details.

Press the mask display button to preview the low pass filter result used for shadows (blue) and highlights (yellow).

The detail scales targeted by the three local contrast sliders can be previewed with their respective mask buttons (see below). Use them in conjunction with adjusting *base detail level* to tune the detail level you want to affect.

highlights
Adjust highlights at the base detail level size.

shadows
Adjust shadows at the base detail level size.

Note: When adjusting *highlights* and *shadows* a lower *base detail level* will tend to enhance local contrast in those respective regions while a higher *base detail level* will result in flatter local contrasts in those regions.

local contrast

coarse details
Adjust coarse, low frequency local contrasts. Click the mask display button to preview the size of the coarse details to adjust.

medium details
Adjust medium frequency local contrasts. Click the mask display button to preview the size of the medium details to adjust.

fine details
Adjust fine, high frequency local contrasts. Click the mask display button to preview the size of the fine details to adjust. Can be used for sharpening.

filter settings

halo control

Adjust the halo control of the filter.

- higher values: suppress halos at the expense of details around edges. Helps with making the effect on local contrast smoother
- lower values: allow more halos to get more local contrast and details.

noise bias

Helps reduce amplification of noise in the shadows by adding a bias to the luminance estimate before filtering. A higher value suppresses more shadow noise but can also reduce genuine fine detail in those areas.

8.2.22. contrast equalizer

Technical information

description
add or remove local contrast, sharpness, acutance.

purpose
corrective and creative.

input
linear, Lab, scene-referred.

processing
frequential, RGB.

output
linear, Lab, scene-referred.

Adjust luminance and chroma contrast in the wavelet domain.

This versatile module can be used to achieve a variety of effects, including bloom, denoise, clarity, and local contrast enhancement.

It works in the wavelet domain and its parameters can be tuned independently for each wavelet detail scale. The module operates in CIE LCh color space and so is able to treat luminosity and chromaticity independently.

A number of presets are provided, which should help you to understand the capabilities of the module.

module controls

The *contrast equalizer* module decomposes the image into various detail scales. On each detail scale, you can independently adjust the contrast and denoise splines for lightness (“luma”) and chromaticity (“chroma”, or color saturation), as well as adjusting the edge-awareness (“edges”) of the wavelet transform. The luma, chroma and edges splines are provided on separate tabs, and some examples of their usage are given in the following sections.

Below the spline graphs is a *mix* slider, which can be used to adjust the strength of the effect, and even to invert the graph (with negative values). While your mouse is over the spline graph, the curve will be displayed as if the *mix* slider was set to 1.0, to allow for easier editing. When you move your mouse away, the graph will be re-adjusted to account for the *mix* slider.

In the background of the curve you can see a number of alternating light and dark stripes. These represent the levels of detail that are visible at your current zoom scale – any details without these stripes are too small to be seen your current view. Adjustments made to control points within the striped section *may* produce a visible effect (depending on the strength of the adjustment). Adjustments outside of the striped region *will not*. Zoom in to see higher levels of detail and make adjustments to the more detailed areas of the image.

Tip: if you are having trouble visualising which parts of the curve will affect which details in the image, you can set the blend mode to “difference”. This will make the image go black except for those areas where the output of the module differs from the input. By raising the curve at one of the control points, you will be able to see which details in the image are represented by that point.

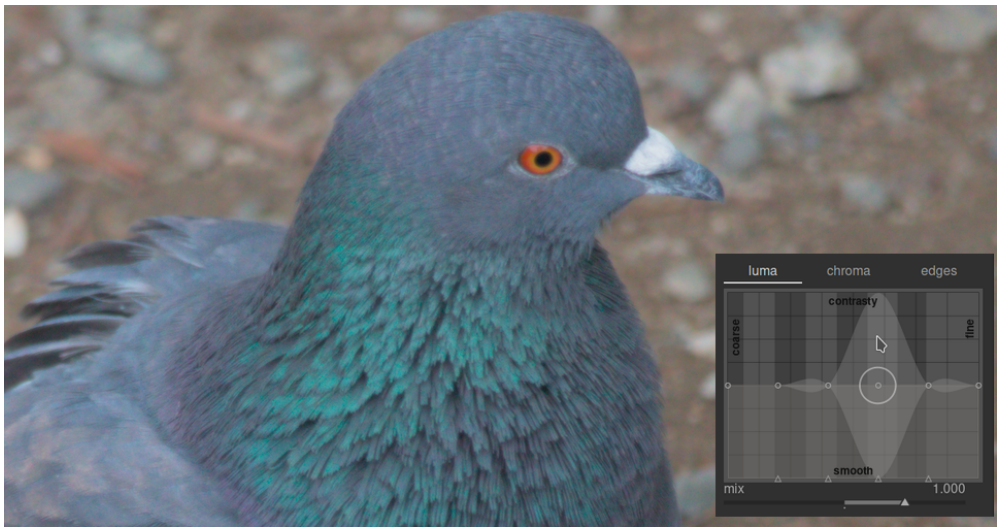
---

luma tab

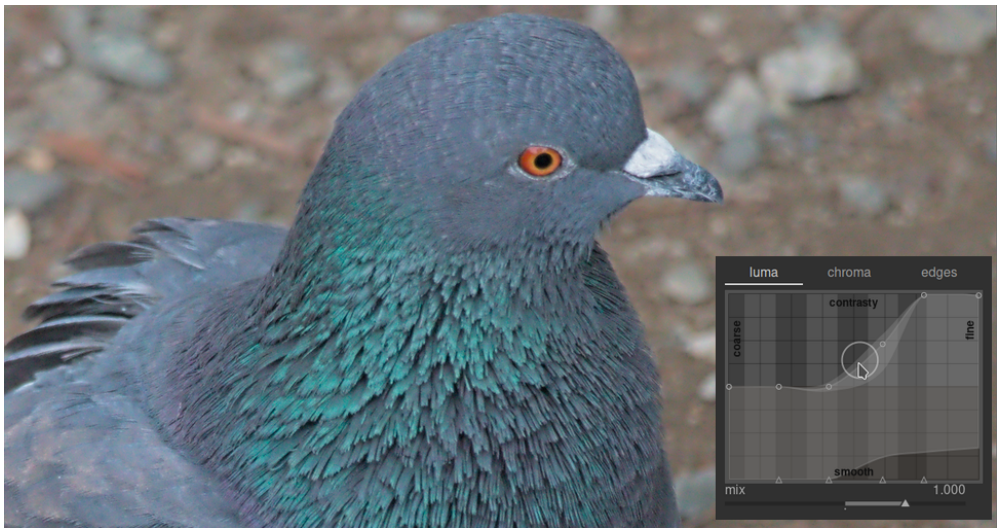
The luma tab allows you to adjust the local contrast in the image's luminance (brightness). Adjustments are represented by a white spline that begins as a horizontal line running across the centre of the graph (indicating that no change will be made). Raise or lower this spline at the left end of the graph to increase or decrease the local contrast of coarse detail in the image. Perform similar adjustments towards the right side of the graph to adjust the local contrast of the fine details in the image.

When you hover the mouse pointer over the graph, a white circle indicates the radius of influence of the mouse pointer – the size of this circle can be adjusted by scrolling with the mouse wheel. The larger the circle of influence, the more control points will be affected when you adjust the curve. A highlighted region in the background shows what the spline would look like if you pushed the currently-hovered control point all the way to the top or bottom on the graph – see the screenshot below for examples of these features. For more information see the wavelets section.

The following image shows the default state of the contrast equalizer module before any adjustments have been made:



Raising the two control points at the right-hand end of the graph will increase the sharpness of the fine details (the eye and feathers of the bird) while leaving coarser details (the rocks in the background) largely unaffected. The example below has been exaggerated to better illustrate the effect.



Increasing the local contrast can also amplify the luma noise in the image. A second spline located at the bottom of the graph can be used to denoise the selected detail scales. Raise this spline (by clicking just above one of the triangles at the bottom of the graph and dragging the line upwards) to reduce noise at the given wavelet scale. In the example above, the dark denoising spline has been raised at the fine-detail end of the graph.

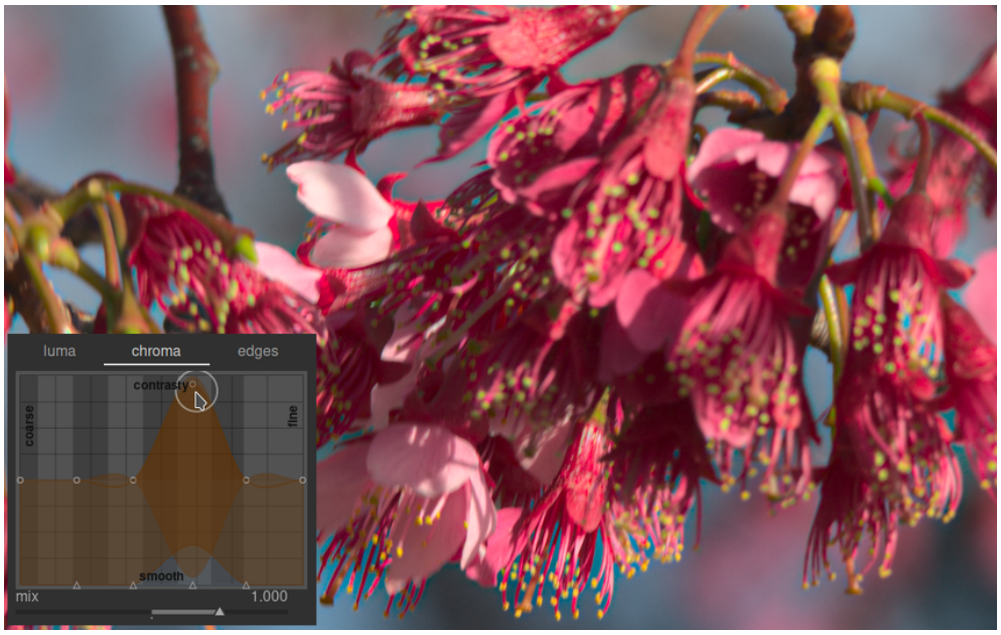
chroma tab

The chroma tab allows the color contrast or saturation to be adjusted at the selected wavelet scales. You'll usually see more saturation when dragging the nodes up, and less saturation when dragging them down. Keeping an eye on the vectorscope can help visualize the changes.

See the following example:



Say you wanted to boost the color contrast around the anthers at the end of the stamen. The pink petals of the flowers are already quite saturated, but using contrast equalizer you can selectively boost the saturation on the small scale of the anthers without impacting the saturation of the petals. The third control point from the right corresponds to the detail scale of the anthers. By raising that control point, you can target anthers saturation only:



As in the luma tab, the chroma tab also has a denoising spline at the bottom of the graph. This can be used to handle chroma noise at different scales within the image. Chroma denoising can generally be more aggressive on larger wavelet scales and has less effect on a smaller scale.

edge tab

The basic wavelet *à trous* transform has been enhanced in the *contrast equalizer* to be “edge-aware”, which can help to reduce the gradient reversals and halo artifacts that the basic algorithm can produce. The *edges* tab does not directly act on the edges in an image; rather it adjusts the edge awareness feature of the wavelet transform. If you have not adjusted the luma or chroma splines, adjusting the edge awareness spline will have no effect.

To see the sorts of artifacts that the *edges* curve tries to combat, here is an example taken from the original paper “Edge-Optimized À-Trous Wavelets for Local Contrast Enhancement with Robust Denoising” (Hanika, Damertz and Lensch 2011):



In the image on the left, the edges spline has been reduced to a minimum, effectively disabling the edge awareness and resulting in halos. In the middle image, the edges spline has been increased too much, resulting in gradient reversals. In the image on the right the edges spline has been set somewhere in between the two extremes, resulting in nice clean edges.

Usually the default central position of the spline is a good starting point, but if there are objectionable artifacts around the edges, this control can be helpful in mitigating them. Some experimentation will be required.

8.2.23. crop

Technical information

description
change the framing.
purpose
corrective or creative.
input
linear, RGB, scene-referred.
processing
geometric, RGB.

output
linear, RGB, scene-referred.

Crop an image using on-screen guides.

This module appears later in the pipeline than the deprecated *crop and rotate* module, meaning that the full image can remain available for source spots in the *retouch* module. For best results, you are advised to use the *rotate and perspective* module to perform rotation and perspective correction (if required), and then perform final creative cropping with this module.

Whenever this module is in focus, the full uncropped image will be shown, overlaid with crop handles and optional guiding lines.



Resize the crop by dragging the border and corner handles. During resizing, the dimensions (in pixels) and aspect ratio of the crop are shown in the center of the crop area.

Move the crop rectangle by clicking and dragging inside the crop area. Constrain movement to the horizontal/vertical axis by holding Ctrl/Shift, respectively while dragging. Commit changes by giving focus to another module.

module controls

The *crop* module controls are split into two sections as follows:

crop settings

aspect

Set the aspect ratio of the crop, constraining the width:height ratio of the crop rectangle to the chosen aspect. Many common numerical ratios are pre-defined. A few special aspect ratios deserve explanation:

- *freehand*: Crop without any restrictions
- *original image*: Retain the aspect ratio of the original image
- *square*: Constrains the aspect ratio to be 1:1
- *golden cut*: The golden ratio (1.62:1)

You can also enter any other ratio after opening the combobox by typing it in the form of “x:y” or as a decimal (e.g. “0.5” to apply a ratio of 2:1).

The button beside the aspect combobox allows you to switch between portrait and landscape orientation if you have selected a rectangular aspect ratio.

If you want to add an aspect ratio to the pre-defined drop-down list you can do this by including a line of the form “`plugins/darkroom/clipping/extra_aspect_ratios/foo=x:y`” in darktable’s configuration file `darktablerc` in the configuration directory . Here “`foo`” defines the name of the new aspect ratio and “`x`” and “`y`” the corresponding numerical values (x and y must be integers). Note that you can only add new entries for ratios not already present in the drop-down list. You can not use fractions as the actual crop size, however you can for the name. For example, if you want an aspect ratio of 1.91:1, add the following to your darktablerc file: `plugins/darkroom/clipping/extra_aspect_ratios/1.91:1=191:100`

Note 1: When resizing an image in freehand mode you may retain the currently-set aspect ratio by holding Shift while dragging on any of the resize controls.

Note 2: The crop area is retained when changing *orientation* .

Note 3: Some RAW files (for example, from Canon or Olympus) can store in-camera aspect ratio information, for when the user selects an aspect ratio in the camera settings. If this information is present, the *crop* module will be automatically enabled and initially set to the in-camera aspect ratio when you start to edit the image. The full image is still available for manual cropping when the crop module is active.

---

margins

These sliders allow you to directly set how much of the image to crop from each side. They are automatically updated if you move or resize the crop area on the image using the mouse.

As this section is rarely used, it is collapsed by default.

left

The percentage of the image that should be cropped from the left side.

right

The percentage of the image that should be cropped from the right side.

top

The percentage of the image that should be cropped from the top.

bottom

The percentage of the image that should be cropped from the bottom.

guides

Tick the box to show guide overlays whenever the module is activated. Click the icon on the right to control the properties of the guides. See guides & overlays for details.

8.2.24. demosaic

Technical information

description
reconstruct full RGB pixels from a sensor color filter array reading.
purpose
mandatory.
input
linear, raw, scene-referred.
processing
linear, raw.
output
linear, RGB, scene-referred.

Control how raw files are demosaiced and optionally apply capture sharpening.

bayer filters

The sensor cells of a digital camera (commonly named “photosites”) are not color-sensitive – they are only able to record different levels of lightness. In order to obtain a color image, each cell is covered by a colored filter (red, green or blue) that primarily passes light of that color. This means that each photosite (and therefore each “pixel” of the raw image) only contains information about a single color channel.

The array of color filters covering the sensor, known as a color filter array (CFA), is commonly arranged in a mosaic pattern known as a Bayer filter (though other patterns are also used). A demosaic algorithm reconstructs the other color channels by interpolation with data from the neighboring photosites. For further reading see the Wikipedia articles on demosaicing and the Bayer filter .

Darktable offers several demosaic algorithms, each with its own characteristics. The differences between them are often very subtle and might only be visible while pixel-peeping. However, as the program works on a pixel-by-pixel basis and demosaic generates the base data for the other modules, the choice of the algorithm can have a visually significant effect on the quality of very fine details in the image. This can include the appearance of false maze patterns as well as the rendering quality of colored edges.

Demosaic interpolation algorithms are often prone to produce artifacts, typically visible as Moiré patterns when zooming into the image. The chosen algorithm might handle *pre-existing* Moiré- or Maze-like patterns in the raw data in a better or worse way. In these circumstances *VNG4* and *LMMSE* are often more stable.

The following demosaic algorithms are available for sensors with Bayer filters:

- *PPG* used to be darktable’s default demosaic algorithm. It is fast, but other algorithms generally yield better results.
- *AMaZE* and *RCD* offer better reconstruction of high-frequency content (finer details, edges, stars) but might struggle with color reconstruction overshoots or added noise in areas of low contrast. While *AMaZE* often retains more high-frequency details it is also more prone to color overshoots than *RCD*. Since *RCD* now offers similar performance to *PPG*, but with better results, it is now the default algorithm.
- *LMMSE* is better suited for use on high ISO and noisy images than *AMaZE* or *RCD*, both of which tend to generate overshooting artifacts when applied to such images. It can also be useful to manage images that exhibit Moiré patterns with other algorithms.
- *VNG4* is better suited for use on images with low-frequency content (e.g. low contrast regions such as sky or snow-fields). Compared to *AMaZE* and *RCD*, it causes loss of some details but is very stable for maze artifacts. VNG4 is no longer recommended – for most images, other algorithms provide better results.

Note: The performance of the demosaic algorithms differs significantly, *AMaZE* being by far the slowest.

---

sensors without bayer filters

There are a few cameras whose sensors do not use a Bayer filter. Cameras with an “X-Trans” sensor have their own set of demosaic algorithms. The default algorithm for X-Trans sensors is *Markesteijn 1-pass*, which produces fairly good results. For slightly better quality (at the cost of much slower processing), choose *Markesteijn 3-pass*. Though *VNG* is faster than *Markesteijn 1-pass* on some computers, it is prone to artifacts.

Images from monochrome cameras do not require demosaicing, however, since darktable 5.4 they use this module to support capture sharpening .

special algorithms

passthrough (monochrome) is only useful for cameras that have had the color filter array physically removed from the sensor (e.g. scratched off). Demosaic algorithms usually reconstruct other color channels by interpolation with data from the neighboring pixels. However, if the color filter array is not present, there is nothing to interpolate, so this algorithm simply sets all the color channels to the same value, resulting in a monochrome image. This method avoids the interpolation artifacts that the standard demosaic algorithms might introduce.

photosite_color is not meant to be used for image processing. It takes the raw photosite data and presents it as red, blue or green pixels. This is designed for debugging purposes in order to see the raw data and can assist with analysis of errors produced by the other demosaic algorithms.

monochrome is only available for monochrome cameras.

dual demosaic algorithms

Some images have areas best demosaiced using an algorithm that preserves high frequency information (like *AMaZE* or *RCD*) and other areas that might profit from an algorithm more suited to low frequency content (like *VNG4*).

In dual demosaic algorithms (e.g. *RCD + VNG4*) the sensor data is demosaiced twice, first by *RCD*, *AMaZE* or *Markesteijn 3-pass* and then by *VNG4*. Both sets of demosaiced data are retained for subsequent processing.

The data from the high frequency algorithm is then analysed for *local data change* and, using a threshold (there is a bit more of maths involved here), the output image is written pixel-by-pixel for each color channel using data from each demosaic algorithm weighed by the local data change.

In general, areas with greater detail are demosaiced by the algorithm best suited to that purpose (*RCD, AMaZe, Markesteijn 3-pass*) and any flat areas (like blue sky) are demosaiced using the second algorithm (*VNG4*).

The ’local data change’ is technically implemented as a gaussian-blurred single channel selection mask calculated from a combination of the threshold value and the pixels’ luminance.

selecting the threshold

An automatically-calculated threshold is difficult to implement. Instead, the “display blending mask” button can be used to display the *selection mask* so you can control the selection of the algorithm manually. The brighter the pixel in the displayed mask, the more the output is taken from the high-frequency algorithm.

capture sharpen

Capture sharpening recovers details lost due to in-camera or lens blurring, which can be caused by diffraction, the anti-aliasing filter or other sources of gaussian-type blur.

Capture sharpening is performed inside the demosaic module because some parts of the algorithm require raw CFA data. The sharpening itself is performed on demosaiced RGB data and is available for bayer, xtrans and monochrome sensors.

Capture sharpening is not intended to be used as a general sharpening / local contrast enhancing tool. It should instead be understood as a way to increase micro-contrast in structures with lots of detail, and leads to better results for further processing. Excellent examples would be details of surfaces like wood, brick walls, hair etc.

Prerequisites for optimal results are:

- Decent white balance parameters (same requirement as for *highlight reconstruction* or *demosaic*) – in very rare cases this could lead to inferior sharpening quality.
- Low chromatic aberration – you might want to use the *raw chromatic aberrations* module to reduce problems – there will still likely be more and stronger halo artifacts.
- Acceptable sensor noise, as noise will generally be amplified by capture sharpening (see *contrast sensitivity* below).

Capture sharpening works in an iterative process using specialized gaussian kernels using a different sigma (in the range of 0.0 -> 1.5) for every single pixel location.

module controls

method

The demosaic algorithm to use (see above).

edge threshold (*PPG* only)

The threshold for an additional median pass. Defaults to “0” which disables median filtering.

LMMSE refine (*LMMSE* only)

Refinement steps for use with the LMMSE demosaic algorithm. Median steps average the output. Refinement steps add some recalculation of red and blue channels. While the refinement options work well for luma noise, they may decrease quality on images with heavy chroma noise.

color smoothing

Activate a number of additional color smoothing passes. Defaults to “off”.

match greens

In some cameras the green filters have slightly varying properties. This parameter adds an additional equalization step to suppress artifacts. Available options are “disabled”, “local average”, “full average” and “full and local average”. This option is not shown for X-Trans sensors.

dual threshold *(dual demosaic modes only)*

Set the contrast threshold for dual demosaic modes. Lower values favor the high frequency demosaic algorithm and higher values favor the low frequency algorithm.

display blending mask *(dual demosaic modes only)*

Show the blending mask that is used to differentiate between high and low frequency areas (adjusted by the “switch dual threshold” parameter). For each pixel, the brighter the mask, the more the module’s output is taken from the high frequency demosaic algorithm.

capture sharpen

Enable capture sharpening. Selecting this option will enable additional controls (see below) and attempt to auto-calculate the *radius* and *contrast sensitivity* based on the current image’s raw sensor data (see below).

iterations (capture sharpen)

The strength of the effect increases with more iterations. For most images a setting of 8 will be sufficient.

radius (capture sharpen)

Defines the mathematics used (sigma) for the special gaussian kernels. Analysis of many images (also from the RawTherapee community) show that, on modern sensors, using a good lens and a high quality demosaicer, a typical radius will be between 0.5 and 0.8.

To auto-calculate the *radius* for the current image (based on raw sensor data), click the button on the right of the radius slider. The underlying algorithm has been used for some time with reliable results, though it might fail on images with lots of chroma noise. This calculation works better when the image is fully-zoomed out and uncropped in the darkroom view.

Note: Do not manually increase the radius much further than the auto-calculated value, as this will soon lead to halo artifacts.

contrast sensitivity (capture sharpen)

As sensor noise will typically be amplified by capture sharpening, we try to counter this problem by performing an analysis of local variance, luminance and a user-defined threshold. Sharpening is applied only to regions that are considered by this analysis to be safe. This analysis results in a mask that can be visualised by clicking the button to the right of the contrast sensitivity slider.

For a good default we check the image ISO and sensor readout precision as a starting point.

Very noisy images will require larger values than default. For very low sensor noise you can decrease the threshold to also improve capture sharpening in very dark parts of the image.

corner boost / sharp center (capture sharpen)

Most lenses are sharper in the central part than the corners of the image. Increasing the *corner boost* parameter causes a larger per-pixel radius to be used in the capture sharpen kernels for the image corners. This extra radius is controlled by the *corner boost* control. Adjusting the *corner boost* will cause an additional slider (*sharp center*) to appear. This controls the affected area using the distance from the image center. Again, the mask button to the right of the slider allows you to visualize the effect.

presets, styles and copy/paste

Defining demosaic presets or using demosaic parameters for styles or copying/pasting history stacks is a bit more tricky than for other modules. Please bear in mind that demosaic works for a range of sensors each offering different algorithms and some specific options. Here are some reminders:

1. It should always be safe to create an automatic preset that only applies to a specific camera.
2. If a preset was defined while working on an image with a different sensor than the current image, darktable translates the algorithm as follows:
   - bayer sensors fall back to *RCD*
   - xtrans sensors fall back to *markjestejn*
   - bayer4 sensors always use *vng4*.
   - dual demosaicing will also be available on bayer and xtrans sensors
3. Extra options like *green equalizing* will only be applied where they are supported by the current sensor.
4. In auto-applied presets with capture sharpen enabled, you might want to use the auto-calculated radius (or contrast sensitivity) instead of using hard-coded values. To achieve this, you must set the radius (or contrast sensitivity) to zero before saving the preset. If this preset is then applied to another image, darktable accepts this as a request for these values to be automatically recalculated.

Capture sharpening is not enabled by default but you could achieve this by defining an auto-applied preset after setting radius and contrast threshold to zero. You are advised to leave the iterations setting at its default setting of 8, as this leads to good sharpening with very low risk for artifacts when used with an auto-calculated radius.

It is generally safe to auto-apply capture sharpening to images with ISO values below 1000. If you have a good sensor, with low noise even on high ISO values, you might be able to use it for ISOs as high as 3200.

8.2.25. denoise (profiled)

Technical information

description
denoise using noise statistics profiled on sensors.

purpose
corrective.

input
linear, RGB, scene-referred.

processing
linear, RGB.

output
linear, RGB, scene-referred.

An easy to use and highly efficient denoise module, adapted to the individual noise profiles of a wide range of camera sensors.

One issue with a lot of denoising algorithms is that they assume that the variance of the noise is independent of the luminosity of the signal. By profiling the noise characteristics of a camera's sensor at different ISO settings, the variance at different luminosities can be assessed, and the denoising algorithm can be adjusted to more evenly smooth out the noise.

Currently, darktable has sensor noise profiles for over 300 popular camera models from all the major manufacturers. If you generate your own noise profile for a camera that is not yet supported by darktable, be sure to share it with the darktable development team so they can include it in the next release! Please see darktable's camera support page for more information.

For an AI-based alternative applied later in the workflow, on the already-demosaiced and edited image, see neural restore and the AI features overview .

modes

The *denoise (profiled)* module implements two algorithms, each of which is available in either an easy-to-use "auto" mode, or a more advanced manual mode with additional controls:

non-local means
This algorithm works in the spatial domain in much the same way as the *astrophoto denoise* module. It averages each pixel with some surrounding pixels in the image. The weight of such a pixel in the averaging process depends on the similarity of its neighborhood with the neighborhood of the pixel being denoised. A patch with a defined size is used to measure that similarity.

Note that this algorithm is quite resource-intensive.

wavelets (default)
This algorithm works in the wavelet domain, and provides a simplified user interface. Wavelet decomposition allows you to adjust the denoise strength depending on the coarseness of the noise in the image. This mode can be used in either *Y0U0V0 color mode* (which allows you to independently control luminance and chroma noise) or *RGB color mode* (which allows you to independently control noise for each RGB channel).

The wavelet algorithm is less resource-intensive than *non-local means*.

luma versus chroma noise

Both "non-local means" and "wavelet" algorithms can efficiently tackle luma (lightness) noise and chroma (color) noise.

In the past, it was suggested that you use two separate instances of this module to tackle *chroma* and *luma* noise independently (using *chroma* and *lightness* blending modes). This is no longer recommended, since the *denoise (profiled)* module is placed before the *input color profile* module in the pixelpipe (so that the profile parameters are accurate) and color blending modes should only be used after the input color profile has been applied.

The new algorithms in this module now provide their own methods to separately handle luma and chroma noise, and in both cases this can be handled with a single module instance.

module controls

The *denoise (profiled)* module provides some controls that are independent of the algorithm used. These are described first, before moving on to the algorithm-specific controls.

When describing the controls specific to an algorithm, we will first cover the simplified interface, and then move on to the more advanced controls for that algorithm.

Note that sliders are provided with minimum and maximum values by default. However, these are only soft limits and, where needed, higher values can be entered by right-clicking on the slider and typing a new value.

common controls

profile

darktable automatically determines the camera model and ISO based on the Exif data of your raw file, and searches for a corresponding profile in its database. If your image has an intermediate ISO value, settings will be interpolated between the two closest datasets in the database, and this interpolated setting will show up as the first line in the combo box. You can also manually override this selection if necessary. Re-selecting the top-most entry in the combo box will return you to the default profile.

compensate highlight preservation

Some cameras underexpose the raw file in specific modes such as HDR, dynamic range expansion, highlight tone priority, HLG or AutoLightingOptimizer to protect the highlights. When enabled, reduces the ISO used for denoise by the factor of the camera's hidden exposure bias. This control appears only when such a mode is detected in the image's Exif data.

whitebalance-adaptive transform

As the white balance amplifies each of the RGB channels differently, each channel exhibits different noise levels. This checkbox makes the selected algorithm adapt to the white balance adjustments. This option should be disabled on the second instance if you have applied a first instance with a color blend mode.

mode

Choose which denoising algorithm to use (see above), and whether to present the simplified ("auto") or full manual interface for that algorithm.

adjust autoset parameters (auto modes only)

Automatically adjust all the other parameters on the current denoising algorithm using a single slider. This is particularly useful when you have had to increase the exposure on an underexposed image, which normally introduces additional noise (as if you had taken the shot with a higher ISO). This control compensates for that by using settings similar to those used for a higher ISO image. The "effective ISO" used by the denoise algorithm is the actual ISO used, multiplied by the value of this slider.

strength

Fine-tune the strength of the denoising. The default value has been chosen to maximize the peak signal-to-noise ratio. It's mostly a matter of taste – whether you prefer a low noise level at the cost of fine details, or you accept more noise to better preserve fine detail.

preserve shadows (advanced mode only)

Lower this control to denoise the shadows more aggressively. Usually, as noise increases, you will need to decrease this parameter.

bias correction (advanced mode only)

Correct any color cast that may appear in the shadows. Increase this value if dark shadows appear too green, decrease if they appear too purple.

non-local means auto sliders

central pixel weight (details)

Control the amount of detail that should be preserved by the denoising algorithm. By default, this will have a low value, meaning that the algorithm will treat both luma and chroma noise equally. Move this slider to the right to reduce the amount of luma denoising, so that the algorithm will primarily affect chroma noise. By adjusting this slider together with the *strength* slider, you can find a good balance between luma and chroma denoising.

non-local means advanced sliders

When you take non-local means out of auto mode, the *adjust autoset parameters* slider is replaced with the following controls. You can use the auto-adjust slider to arrive at some initial settings then, when you switch to manual mode, the sliders will show the equivalent manual settings. You can then continue to fine-tune the manual settings from the auto-set starting point.

patch size

Control the size of the patches being matched when deciding which pixels to average – see *astrophoto denoise* for more information. Increase this for images with more noise but be aware that high values may smooth out fine edges. The effect of this slider on processing time is minimal.

search radius

Control how far away from a pixel the algorithm will attempt to find similar patches. Increasing the value can give better results for very noisy images when coarse grain noise is visible, but the processing time is hugely impacted by this parameter (the processing time increases with the square of this parameter). A lower value will make execution faster, a higher value will make it slower. It most cases it is better to use the scattering parameter, which has a similar effect but without the heavy processing cost.

scattering (coarse-grain noise)

Like the *search radius*, this slider controls how far from a pixel the algorithm will attempt to find similar patches. However, it does this without increasing the number of patches considered. As such, processing time will stay about the same. Increasing the value will reduce coarse grain noise, but may smooth local contrast. This slider is particularly effective at reducing chroma noise.

wavelet curves

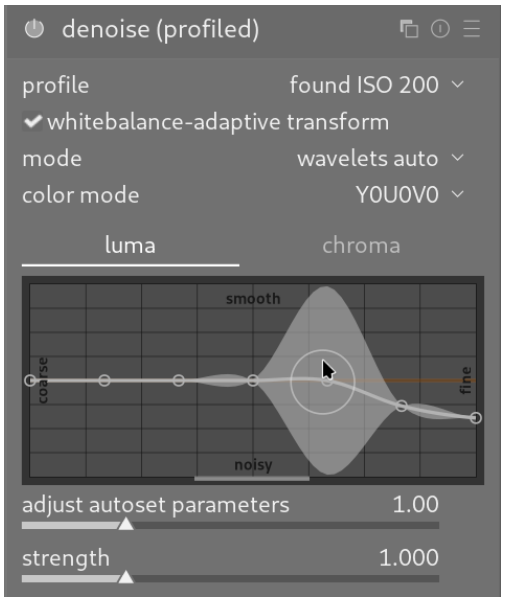
Wavelet curves are shown when one of the “wavelet” modes is selected.

The noise in an image usually consists of both fine grain and coarse grain noise, to varying degrees. The wavelet curves allow the strength of the denoising to be adjusted depending on the coarseness of the visible noise. The left end of the curve will act on very coarse grain noise, while the right of the curve will act on very fine grain noise. Raising the curve will increase the amount of smoothing, while lowering the curve will decrease it.

For example, you can preserve very-fine-grain noise by pulling the right-most part of the curve down. When tackling chroma noise (e.g. on a chroma curve, or on a second module instance with a color blend mode) you can safely raise the right side of the curve, since colors do not change a lot on fine scales. This can be useful if you see some noisy isolated pixels with the wrong color.

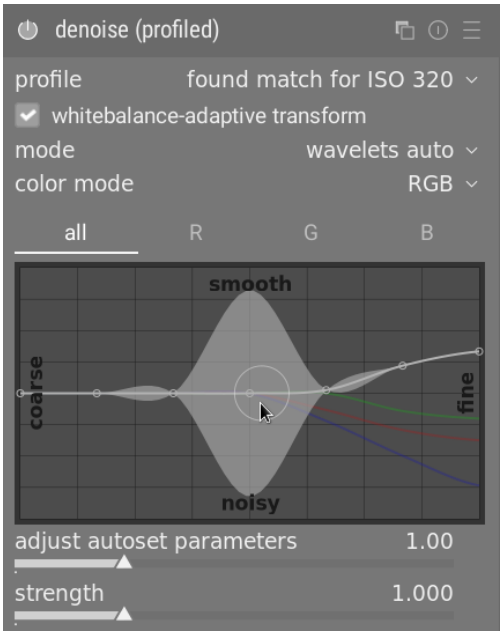
wavelets *Y0U0V0* color mode

The preferred way to use wavelets is with the *Y0U0V0* color mode. This mode separates the denoising curves into luminance (*luma*) and color (*chroma*) components. You can then use the *luma* curve to control the level of luma denoising, and the *chroma* curve to control the level of chroma denoising.



wavelets *RGB* color mode

Before the *Y0U0V0* color mode was introduced, wavelet-based denoising could only be performed directly on the *R*, *G* and *B* channels, either together or individually.



If you want to denoise the RGB channels independently, the best way to do this is to use an instance of the *color calibration* module placed immediately before the *denoise (profiled)* module so that it outputs a gray channel based on the *red* channel only, then denoise that monochrome image using the *red* wavelet curve. Repeat this procedure for the blue and green channels. This procedure is time-consuming, but gives the best result because looking at the color of a noisy pixel is not a reliable way to determine which channel to adjust. For example, a noisy red pixel could be due to a noise peak on the red channel, but could also be due to a noise lull on the blue and green channels.

The issue with independently denoising the RGB channels is that there can still be some residual chroma noise at the end that requires excessive smoothing to eliminate. This was in fact one of the key motivations behind implementing the *Y0U0V0* color mode.

wavelets advanced sliders

When you take *wavelets* out of *auto* mode, the *adjust autoset parameters* slider is replaced with the *preserve shadows* and *bias correction* controls listed above in the common controls section.

8.2.26. diffuse or sharpen

Technical information

description
simulate directional diffusion of light with heat transfer model to apply an iterative edge-oriented blur, inpaint damaged parts of the image, or to remove blur with blind deconvolution.

purpose
corrective and creative.

input
linear, RGB, scene-referred.

processing
linear, RGB.

output
linear, RGB, scene-referred.

Diffusion is a family of physical processes by which particles move and spread gradually with time, from a source that generates them. In image processing, diffusion mostly occurs in two places:

- diffusion of photons through lens glass (blur) or humid air (hazing),
- diffusion of pigments in wet inks or watercolors.

In both cases, diffusion makes the image less sharp by “leaking” particles and smoothing local variations.

The *diffuse or sharpen* module uses a generalized physical model to describe several kinds of diffusion, and can be used by image makers to either simulate or revert diffusion processes.

As it is a highly technical module, several presets are provided to demonstrate its use for various purposes.

Diffusion can be removed in order to:

- recover the original image from sensors with an anti-aliasing filter or mitigate the blur created by most demosaicing algorithms (use one of the *sharpen demosaicing* presets and move the module before the *input color profile* module in the pipeline),
- reverse static lens blurring/defocusing (use one of the *lens deblur* presets),
- remove atmospheric haze (use the *dehaze* preset),
- add extra acutance for better legibility (use the *local contrast* preset).

Note that motion blurs cannot be reverted by undoing the diffusion process, as they are not diffusive in nature.

Diffusion can be added in order to:

- create a bloom or Orton effect (use the *bloom* preset),
- inpaint missing or damaged parts of an image (use the *inpaint highlights* preset),
- denoise in an edge-preserving way (use one of the *denoise* presets)
- apply a surface blur (use the *surface blur* preset).

Since the process is physical, even its glitches may be used for creative purposes. For example, you can:

- simulate line drawing or watercolor (use the *simulate line drawing* and *simulate watercolor* presets),
- create random patterns and textures by increasing noise (over time, with iterations, noise will connect with neighbors to create random specks).

Note: This module is highly resource-intensive, as it is actually an anisotropic, multiscale, partial differential equation solver. The module’s runtime increases with the number of iterations and OpenCL is therefore strongly recommended. Some “fast” presets are also provided for use on systems without OpenCL.

---

concepts

time

Diffusion is a time-dependent process: the more time it has, the further the particles can spread. In this module, time is simulated using the number of iterations (the number of times the algorithm runs on top of itself). More iterations can make reconstruction (deblurring, denoising, dehazing) more accurate if properly set, but can also cause it to degenerate.

direction

Natural diffusion usually takes place from points with a high potential (high energy or high concentration of particles) to those with a low potential (low energy or low concentration of particles). In an image, this means that diffusion always occurs from the brightest pixels to the darkest.

This particular implementation can simulate natural diffusion, using what is called an isotropic diffusion (all directions have the same weight, like heat diffusion), but can also force a weighted direction parallel to the gradients (forcing diffusion across object edges and creating ghost edges), or a weighted direction perpendicular to the gradients, called isophote (forcing diffusion to be contained inside edges, like in a droplet of watercolor). The relative weight of each direction (gradient and isophote) is user-defined and can be found in the *direction* section of the module.

speed

Depending how fluid the environment is, particles can move more or less freely and therefore more or less fast. The speed of diffusion can be set in the *speed* section of the module.

When performing reconstruction (denoising, deblurring, dehazing), it is advisable to use smaller speeds for better accuracy. This prevents numerical overshoots (and therefore degeneration of the solution) and may require more iterations. For small numbers of iterations, higher speeds may be used. Note that large blurs need many iterations for proper reconstruction, so the speed should be adjusted to avoid degenerating the solution.

All speeds are added (first to fourth orders), and the sums "`first order + second order`" and "`third order + fourth order`" should never exceed ±100%, unless you want to produce glitch art .

scale

Natural diffusion is supposed to happen only to the closest neighboring coordinates. That is, at each iteration, each pixel should only interact with its 9 nearest neighbors.

Here, we fast-track things a bit to save time and reuse the multi-scale wavelets scheme from the *contrast equalizer* module, so that we can diffuse at different scales. The maximal scale of diffusion is defined by the *radius span* parameter.

Regardless of the diffusion, a *sharpness* parameter allows you to increase or decrease the details at each scale, much like the spline controls of the *contrast equalizer*. Along with the *edge sensitivity* slider, this provides the same features as the *contrast equalizer* module (*luma* and *edges* tabs) but in a scene-referred RGB space.

module controls

properties

iterations
: The number of times the algorithm should run on top of itself. High values slow the module down but allow more accurate reconstructions, provided that the diffusion speeds are low enough.

central radius
: The main scale of the diffusion. Zero causes the diffusion to act more heavily on fine details (used for deblurring and denoising). Non-zero values define the size of details to be heavily diffused (used to increase local contrast).

radius span
: This allows you to select the band of details radii to act on, around the central radius. The span of diffusion defines a range of detail scales (between `center - span` and `center + span`) within which the diffusion is confined. High values diffuse on a large band of radii, at the expense of computation time. Low values diffuse closer around the central radius. If you plan to deblur, the radius span should be approximately the width of your lens blur and the central radius should be zero. If you plan to increase the local contrast, but don't want to affect sharpness or noise, the radius span should be 3/4 of your central radius maximum.

The radii are expressed in pixels of the full-resolution image, so copy+pasting settings between images of different resolution may lead to slightly different results, except for pixel-level sharpness.

For electrical engineers, what is set here is a band-pass filter in wavelets space, using a gaussian frequential window centered on `central radius` with a fall-off (standard deviation) of `radius span`. Wavelet scales are analogous to harmonic frequencies and each wavelet scale defines the radius of the details to act on.

speed (sharpen ↔ diffuse)

In the following controls, positive values apply diffusion, negative values undo diffusion (i.e. sharpen) and zero does nothing.

1st order speed (gradient)
: The speed of diffusion of the low-frequency wavelet layers in the direction defined by the *1st order anisotropy* setting.

2nd order speed (laplacian)
: The speed of diffusion of the low-frequency wavelet layers in the direction defined by the *2nd order anisotropy* setting.

3rd order speed (gradient of laplacian)
: The speed of diffusion of the high-frequency wavelet layers in the direction defined by the *3rd order anisotropy* setting.

4th order speed (laplacian of laplacian)
The speed of diffusion of the high-frequency wavelet layers in the direction defined by the *4th order anisotropy* setting.

direction

In the following controls, positive values cause diffusion to avoid edges (isophotes), negative values make diffusion follow gradients more closely, and zero affects both equally (isotropic).

1st order anisotropy
The direction of diffusion of the low-frequency wavelet layers relative to the orientation of the gradient of the low-frequency (*1st order speed* setting).

2nd order anisotropy
The direction of diffusion of the low-frequency wavelet layers relative to the orientation of the gradient of the high-frequency (*2nd order speed* setting).

3rd order anisotropy
The direction of diffusion of the high-frequency wavelet layers relative to the orientation of the gradient of the low-frequency (*3rd order speed* setting).

4th order anisotropy
The direction of diffusion of the high-frequency wavelet layers relative to the orientation of the gradient of the high-frequency (*4th order speed* setting).

edge management

sharpness
Apply a gain on wavelet details, regardless of properties set above. Zero does nothing, positive values sharpen, negative values blur. This is mostly useful as an adjustment variable when blooming or blurring, to retain some sharpness while adding a glow around edges. You are not advised to use this for sharpening alone, since there is nothing to prevent halos or fringes with this setting.

edge sensitivity
Apply a penalty over the diffusion speeds when edges are detected. This detection uses the local variance around each pixel. Zero disables the penalty, higher values make the penalty stronger and more sensitive to edges. Increase if you notice edge artifacts like fringes and halos.

edge threshold
Define a variance threshold, which affects mostly low-variance areas (dark or blurry areas, or flat surfaces). Positive values will increase the penalty for low-variance areas, which is good for sharpening or increasing local contrast without crushing blacks. Negative values will decrease the penalty for low-variance areas, which is good for denoising or blurring with a maximal effect on black and blurry regions.

diffusion spatiality

luminance masking threshold
This control is useful if you want to in-paint highlights. For values greater than 0%, the diffusion will only occur in regions with a luminance greater than this setting. Note that gaussian noise will be added in these regions to simulate particles and initialize the in-painting.

workflow

The main difficulty with this module is that while its output can vary dramatically depending on its input parameters, these parameters have no intuitive link to everyday life. Users are likely to be overwhelmed, unless they are already familiar with Fourier partial differential equations. This section proposes some ways to approach this module without the burden of having to understand the underlying theory.

general advice

If you intend to deblur your image using this module, always start by properly correcting any chromatic aberrations and noise in the image, since the deblurring may magnify these artifacts. It is also important that you don't have clipped black pixels in your image. These can be corrected with the *black level correction* of the *exposure* module.

Since it works on separate RGB channels, it is better to apply this module after *color calibration* , so that you start with a fully neutral, white-balanced, input image. Note that increasing local contrast or sharpness will also lead to a slight color contrast and saturation boost, which is usually a good thing. Since it uses a variance-based regularization to detect edges, it is also better to put this module before any non-linear operation.

starting with presets

The provided presets have been tuned by the developer and tested on a range of images for typical purposes. The easiest way is simply to start from the presets, and then tweak them as needed:

- if the effect seems too strong, decrease the number of iterations,
- if edge artifacts appear, increase the edge sensitivity,
- if deblurring starts to affect valid blurry parts (bokeh), reduce the radius,
- if deblurring seems correct in bright areas but excessive in dark areas, increase the edges *threshold*,
- if deblurring clips black pixels, lower the *black level correction* in *exposure* module,
- fine-tune the *sharpness* to your taste.

starting from scratch

The module's default settings are entirely neutral and will do nothing to your image. The spirit of the module is that each order affects the texture of the image in a particular way.

Start by tuning the first order parameters (speed and anisotropy) to get an initial base. Then adjust the radius. This will affect coarser textures (either blur or sharpen them). Remember that the first order acts on the low frequencies of the wavelet scale and follows a direction parallel or perpendicular to the gradient of the low frequencies.

Next, start to tune the second order parameters (speed and anisotropy). The second order also acts on the low frequencies of the wavelet scale but this time follows a direction parallel or perpendicular to the gradient of the *high* frequencies, which can either be the direction of maximal sharpness or of noise. This can be used to reduce noise (using the second order in diffusion mode, with positive values) when you used the first order in sharpening mode (with negative values).

These two steps can be performed on the zoomed-out image. Remember that, while great care has been taken to make the algorithm's visual result fairly scale-invariant, the preview will be exact only when zoomed 1:1. In any case, anything happening at pixel level (radius < 2px) will not be visible for zoom levels lower than 50%.

At this point, you may want to tweak the edge sensitivity to take care of any edge artifacts. In theory, diffusing in the isophote direction ensures that diffusion is contained inside edges, but this is not sufficient when corners and sharp convex shapes are present in the image.

When the edge sensitivity control has been adjusted to produce satisfying results, the image usually becomes quite soft. In most cases it will be necessary, at this point, to increase the number of iterations in order to compensate. This will come with a performance penalty so tread carefully with the performance/quality trade-off depending on your hardware. If you can't increase the number of iterations, you will have to increase the diffusing speed.

The final step is to fine-tune the third and fourth order, which take care of the high frequencies of each wavelet scale. You will need to be a lot more gentle with these settings than for the first and second orders, as they can cause noise to blow-up really fast.

The third order follows the gradient or isophote direction of the low frequency layer, so can be used to guide the high frequency diffusion in a direction that is more likely to be legitimate regarding real edges (and less prone to catch noise).

The fourth order follows the gradient or isophote direction of the high frequency layer and is more likely to catch noise. Diffusing on the fourth order is the best way to reduce noise without affecting sharpness too much, either as a stand-alone denoiser, or as a regularization step in a deblurring process.

using multiple instances for image reconstruction

Noise post-filtering may benefit from **introducing** a diffusion process – this can be applied as an extra step after the *denoise (profiled)* module.

Conversely, the following optical issues may benefit from reconstruction by **undoing** the diffusion process:

1. blur introduced by a sensor's low-pass filter (LPF) and/or anti-aliasing performed by the *demosaic* module,
2. static lens blur,
3. haze/fog,
4. light diffusion (using a diffuser that is too large), leading to even lighting and lack of local contrast on the subject.

While more than one of these issues can affect the same image at the same time, it is better to try to fix them separately using multiple instances of the module. When doing so, ensure the issues are corrected from coarse scale to fine scale, and that denoising always happens first. That is, your instances should appear in the following pipe order :

1. denoise,
2. local contrast enhancement,
3. dehaze,
4. lens blur correction,
5. sensor and demosaic correction.

Starting with the coarser-scale reconstructions reduces the probability of introducing or increasing noise when performing the finer-scale reconstructions. This is unintuitive because these processes don't happen in this order during the formation of the image. For the same reason, denoising should always happen before any attempt at sharpening or increasing acutance.

notes and warnings

While this module is designed to be scale-invariant, its output can only be guaranteed at 100% zoom and high quality or full-size export. Results at lower zoom levels or export dimensions may or may not match your expectations.

When setting a deblurring algorithm, try to bear in mind that many of the greatest images in the history of photography were taken with lenses that were not remotely as sharp as those available today. Although the current trend is to build and sell increasingly sharp lenses and have software apply insane amounts of sharpening on top, this fashion does *not* lead to better imagery and makes the retouching process more tedious. Soft focus and a bit of blurriness have some poetic merits too, which surgically-sanitized HD images may fail to convey.

It should be noted that global contrast (using simple tone curves or black/white levels) also affects our perception of sharpness, which is quite different from optical sharpness (optical resolution). Human eyes are only sensitive to local contrast, which may come from optical sharpness (e.g absence of diffusion – thin edges) as well as from amplified tonal transitions. If some global tone mapping is in place to increase the contrast, the image will look sharper. If a tone mapping is used to decrease the contrast, the image will look more blurry. In none of these cases are the actual edges of objects affected in any way, and the perceptual consequences are pure illusion.

Part of the aging process is a loss of eyesight. The amount of sharpening that people over 50 find pleasing may not be the same as for people in their 20s. It is worth considering sharpening to obtain a *plausible* result (matching your everyday perception) rather than a *pleasing* result (that may look good only to people with the same eyesight as yours).

Finally, assessing the sharpness of images zoomed to 1:1 (100%) or more is a foolish task. In museums, exhibitions and even on screen, the general audience looks at images as a whole, not with a magnifying glass. Moreover, in most practical uses, photographs rarely exceed a resolution of 3000×2000 pixels (roughly a 300 DPI print at A4/Letter dimensions) which, for 24 megapixel sensors, means downscaling by a factor of 4. When examining a 24 megapixel file at 1:1, you are actually looking at an image that will never exist. Sharpening at pixel level, in this context, is a waste of time and CPU cycles.

8.2.27. dither or posterize

Technical information

description
reduce banding and posterization effects in output JPEGs by adding random noise, or reduce bit depth.

purpose
corrective, artistic.

input
non-linear, RGB, display-referred.
processing
non-linear, RGB.
output
non-linear, RGB, display-referred.

This module eliminates some of the banding artifacts that can result when darktable's internal 32-bit floating point data is transferred into discrete 8-bit or 16-bit integer output format for display or export. It can also be used for creative posterization effects.

Although not an inherent problem in any of darktable's modules, some operations may provoke banding if they produce a lightness gradient in the image. To mitigate possible artifacts you should consider activating dithering when using the *vignetting* or *graduated density* modules. This is especially relevant for images with extended homogeneous areas such as cloudless sky. Also watch out for banding artifacts when using a gradient drawn mask .

Viewing an image dithered into a very low bit depth from some distance (e.g. "Floyd-Steinberg 1-bit b&w") will give the impression of a homogeneous grayscale image. darktable attempts to mimic this impression when rendering zoomed-out images in the center view, the navigation window and thumbnails. This is accomplished by dithering those images into a higher number of grayscale levels. Note that, as a consequence, the scopes module – the data for which is derived from the navigation window – will show this increased number of levels and is therefore not a full match to the output image.

module controls

method
Choose the dithering method to use.

Floyd-Steinberg (default): Systematically distribute quantization errors over neighboring pixels. This method can be selected with some typical output bit depths. Alternatively, you can select *Floyd-Steinberg auto*, which automatically adapts to the desired output format.

random dithering: This method just adds some level of randomness to break sharp tonal value bands.

posterization: This method quantizes the pixel values into the indicated number of distinct levels per color channel (similar to Floyd-Steinberg) but does not redistribute the quantization errors, producing a posterization effect. Selecting two levels per channel produces eight possible output colors, three levels produces 27 colors (3x3x3), and so on. Use other modules affecting colors or tone curves (such as *tone equalizer* or *color balance rgb*) to fine-tune which pixels produce which posterized colors. You can also use various blending modes to control the colors of the result.

damping ("random" method only)
Controls the level of added random noise expressed as a damping factor in a 10*log 2 basis. A value of -80 is a good fit for 8-bit output formats as it corresponds to a maximum change of one of the 256 possible levels; -160 is a good fit for for 16-bit output.

8.2.28. enlarge canvas

Technical information

description
add empty space to the left, top, right or bottom.
purpose
corrective and creative.
input
linear, RGB, scene-referred.
processing
linear, RGB.
output
linear, RGB, scene-referred.

Increase the size of the image canvas and fill the additional area with the selected color.

module controls

percent left
The amount of enlargement to the left of the image.
percent right
The amount of enlargement to the right of the image.
percent top
The amount of enlargement to the top of the image.
percent bottom
The amount of enlargement to the bottom of the image.
color
The color with which to fill the additional area.

8.2.29. exposure

Technical information

description
redo the exposure of the shot as if you were still in-camera using a color-safe brightening similar to increasing ISO setting.
purpose
corrective and creative.
input
linear, RGB, scene-referred.
processing
linear, RGB.
output
linear, RGB, scene-referred.

Increase or decrease the overall brightness of an image.

This module has two modes of operation:

manual
Set the *exposure* and *black level* manually
automatic (RAW images only)
Use an analysis of the image's histogram to automatically set the exposure. Darktable automatically selects the exposure compensation that is required to shift the selected *percentile* to the selected *target level* (see definitions below). This mode is particularly useful for automatically altering a large number of images to have the same exposure. A typical use case of automatic mode is deflickering of time-lapse photographs.

module controls

mode
Choose the mode of operation (automatic/manual).
compensate camera exposure (manual mode)
Automatically remove the camera exposure bias (taken from the image's Exif data).
highlight preservation mode (manual mode)
Some cameras underexpose the raw file in specific modes such as HDR, dynamic range expansion, highlight tone priority, HLG or AutoLightingOptimizer to protect the highlights. When enabled, it restores the withheld exposure (the amount, in EV, is shown on the button). This control appears only when such a mode is detected in the image's Exif data.
exposure (manual mode)
Increase (move to the right) or decrease (move to the left) the exposure value (EV). To adjust by more than the default limits shown on the slider, right click and enter the desired value up to +/-18 EV (see module controls).
The picker tool on the right sets the exposure such that the average of the selected region matches the target lightness defined in area exposure mapping options.

percentile (automatic mode)

Define a location in the histogram to use for automatic exposure correction. A percentile of 50% denotes a position in the histogram where 50% of pixel values are above and 50% of pixel values are below that exposure.

target level (automatic mode)

Define the target level for automatic exposure correction (EV) relative to the white point of the camera.

black level correction (manual and automatic modes)

Adjust the black level point to unclip negative RGB values.

Note: Do not use the black level correction to add more density in blacks as this can clip near-black colors out of gamut by generating negative RGB values. This can cause problems with some modules later in the pixelpipe. Instead, use a tone mapping curve to add density to the blacks. For example, you can use the *relative black exposure* slider on the *scene* tab of the filmic rgb module, or establish a deeper toe in the base curve module.

---

area exposure mapping

The area mapping feature is designed to help with batch-editing a series of images in an efficient way. In this scenario, you typically develop a single reference image for the whole batch and then copy&paste the development stack to all of the other images in the batch.

Unfortunately, the light often changes slightly between shots, even within the same series captured in the same conditions. This can be the result of a cloud passing by the sun in natural light, surface reflections having less “shine” from a different angle, or simply due to unavoidable variability in the mechanical diaphragm aperture. Each image will still need some individual fine-tuning if you want a perfectly even look over the whole series, and this can be both time-consuming and frustrating.

Area exposure mapping allows you to define a target brightness, in terms of exposure, for a particular region of the image (the control sample), which you then match against the same target brightness in other images. The control sample can either be a critical part of your subject that needs to have constant brightness, or a non-moving and consistently-lit surface over your series of images.

The mapping process consists of two steps.

step 1: set the target

There are two ways of setting the target brightness for your control sample:

1. if you know or expect an arbitrary lightness for the control sample (for example, a gray card, a color chart, a product or a logo of a specified brightness), you can set its L value directly, in CIE Lab 1976 space,
2. if you simply want to match the development of your reference image, set the *area mode* to *measure*, then enable the picker (to the right of the *exposure* slider) and draw a rectangle over your control sample. The *input* column will then be updated with the lightness value of the control sample before the exposure correction, and the *target* column will show the resulting lightness of the control sample after the current exposure setting is applied.

If you reset the lightness value, the default value is 50% (middle-gray) – this can be useful to quickly set the average exposure of any image.

Note that the target value is not reset when you reset the module itself, but is stored indefinitely in darktable’s configuration and will be available on next launch as well as for the next image you develop.

step 2 : match the target

When you open a new image, the *area mode* is automatically reset to *correction*. Using the picker attached to the exposure slider, you can then directly reselect your control sample in the new image. The proper exposure setting required for the control sample to match the memorized target lightness will be automatically computed, and the setting will be updated in the same operation.

This operation can be repeated as many times as you have images in your series with no further work.

Note 1: Trying to match lightness of moving parts of a subject across frames can prove tricky because they can have legitimate changes in their illumination as their orientation changes with respect to the main light source. For example, a part of the face can be fully lit in some frames and partially shaded in others. An inconsistently-lit control sample will generally not provide a robust reference for lightness matching across a series and may result in more work than manually matching it with visual feedback.

Note 2: The exposure module works (by default) in the scene-referred, linear, camera RGB part of the pixel pipeline, before the input color profile is applied. However, the conversion from camera RGB to CIE Lab 1976 space relies on the input color profile. All of the lightness L metrics given in the *area mapping* settings will use the input profile defined later in the *input color profile module* to perform the conversion accurately, but the conversion itself assumes a linear (RAW) signal and will not work for JPEG and PNG images (which are non-linearly encoded before the *input color profile* module). If you want to use this feature on non-RAW images, you will need to move the exposure module to after *input color profile* or to use the *module order* preset *v3.0 for JPEG/non-RAW input*.

Note 3: Perfectly matching your control sample against the target lightness may still not yield a similar perceptual result, even if the numbers are exactly the same. For example, if your subject sits in front of a background made of some bright parts and some dark parts, the ratio of bright areas / dark areas will affect the perception of contrast and brightness. If this ratio changes across your series, the subject brightness will not appear constant even though the lightness value is exactly constant. For more details, see the checker shadow illusion and the Chubb illusion .

8.2.30. external raster mask

Technical information

description
read PFM/PNG files recorded for use as raster masks.
purpose
corrective or creative.
input
linear, raw, scene-referred.
processing
linear, raw.
output
linear, raw, scene-referred.

Generate a raster mask from a `.pfm` to be used by subsequent processing modules.

Note: `.pfm` files are not very common but darktable and GIMP can both generate such files. Both grayscale and RGB files are supported.

usage

Raster masks are automatically created whenever you define a mask within a processing module, for subsequent re-use in modules further up the pixelpipe. In some cases you might want such a mask to be generated by other software, for example to make use of software that can isolate a subject from its background, or simply to add some background pattern.

Using this module, external raster mask files are made available as a raster mask very early in the pixel pipeline. They can be of any dimension/size and are always scaled to the size of the currently developed image. Please note, however, that since raster masks are distorted by modules like *orientation* the external raster mask dimensions should always have the same orientation as the original raw file sensor data.

module controls

file selection
Choose the external raster file to use. The initial raster mask files root folder can be set in preferences > processing . The raster mask is stored in the image's associated DTDATA sidecar file unless preferences > storage > XMP sidecar files is set to *never*, in which case the raster mask is referenced from the selected path.

mode
Choose any of the RGB channels, or a combination of them, to generate the raster mask from.

vectorize
Create a separate path mask for each shape identified in the current raster mask, and add all the resulting path masks to the mask manager.

8.2.31. filmic rgb

Technical information

description
apply a view transform to prepare the scene-referred pipeline for display on SDR screens and paper prints while preventing clipping in non-destructive ways.

purpose
corrective and creative.

input
linear or non-linear, RGB, scene-referred.

processing
non-linear, RGB.

output
non-linear, RGB, display-referred.

Remap the tonal range of an image by reproducing the tone and color response of classic film.

This module can be used either to expand or to contract the dynamic range of the scene to fit the dynamic range of the display. It protects colors and contrast in the mid-tones, recovers the shadows, and compresses bright highlights and dark shadows. Highlights will need extra care when details need to be preserved (e.g. clouds).

darktable offers three modules for this task: *filmic rgb*, *sigmoid* and *AgX* . Only one of them should be enabled at a time. *filmic rgb* gives you the most direct control over where the black and white points of the scene are placed, at the cost of a larger number of parameters. The module is derived from another module of the same name in the Blender 3D modeller by T. J. Sobotka.

You can automatically enable this module in new images by setting the value of preferences > processing > auto-apply pixel workflow defaults to 'scene-referred (filmic)'.

Note: The controls of this module look technical, but they are meant to be set by eye. Judge the result on the image, not on the numbers in the GUI.

---

before you start

filmic rgb expects a linear, scene-referred image with a sensible mid-tone level. Prepare the image as follows:

set the mid-tones
In the *exposure* module, raise or lower the exposure until the mid-tones look right. Ignore the highlights at this stage – *filmic rgb* will bring them back.

Avoid negative pixel values in dark areas, as they make the calculations in *filmic rgb* unpredictable. On some camera models (mainly Canon), rawspeed – the raw decoding library of darktable – sets an exaggerated black level, which crushes the blacks and produces negative values. In that case, enter a negative black level correction in the *exposure* module.

use only one tone mapper
filmic rgb is a tone mapper ; you typically have exactly one tone mapper enabled for an image.

prepare the image if you use the pickers
The pickers in the *scene* tab measure luminance in RGB, where luminance and chrominance are linked. For reliable readings, first neutralize any color cast in the *color calibration* or *white balance* module, use a high quality *demosaic* algorithm and denoise a noisy image beforehand. None of this is required if you set the exposure sliders by hand.

Note: The module works best on images that use the full dynamic range of the sensor. In-camera, expose as brightly as the highlights allow (a technique known as “expose to the right”). Some clipping is acceptable – for example on specular reflections – but clipped data is lost and can only be approximated, using the *reconstruct* tab.

how filmic rgb works

filmic rgb maps the dynamic range of the photographed scene to the dynamic range of the display. The mapping is defined in five tabs:

- The *scene* tab defines the input: which luminances in the scene are to be treated as white and as black.
- The *reconstruct* tab handles blown highlights.
- The *look* tab defines the artistic intent. It applies an S-shaped curve that increases mid-tone contrast and maps the scene’s middle-gray to the middle-gray of the display.
- The *display* tab defines the output. These parameters rarely need to be changed.
- The *options* tab contains advanced settings. *filmic rgb* compresses local contrast. Once you are happy with the tone mapping, you may want to compensate for this in the *local contrast* module, add saturation in the *color balance rgb* module, and refine the tonal distribution with the *tone equalizer* .

The sliders are limited to typical and safe values. To enter a value outside these limits, right-click the slider and type it in.

Note: *filmic rgb* always changes the image to some degree – there is no neutral setting. The following comes close, and performs little more than a logarithmic mapping between the bounds set in the *scene* tab:

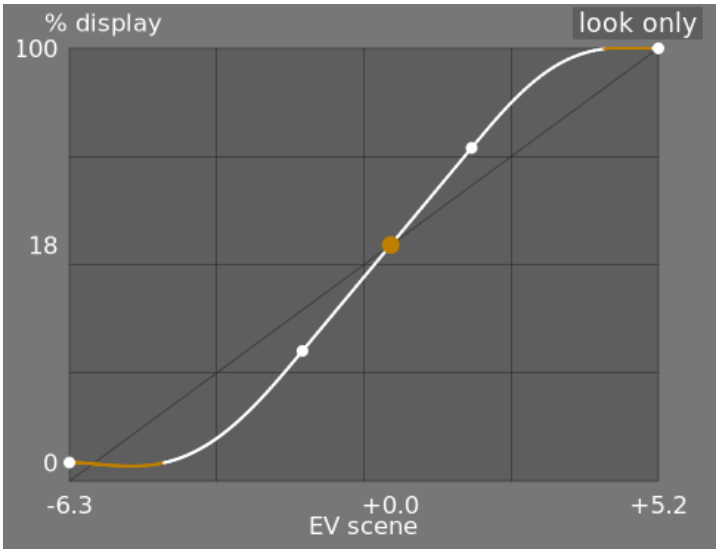
- in the *look* tab, set *contrast* to 1.0, *linear region* to 99% and *highlights saturation mix* to 0%,
- in the *options* tab, set *contrast in shadows* and *contrast in highlights* to *soft*.

graphic display

The graph at the top of the module offers several views. Cycle through them with the  icon to the right of the graph, and toggle the axis labels with the  icon.

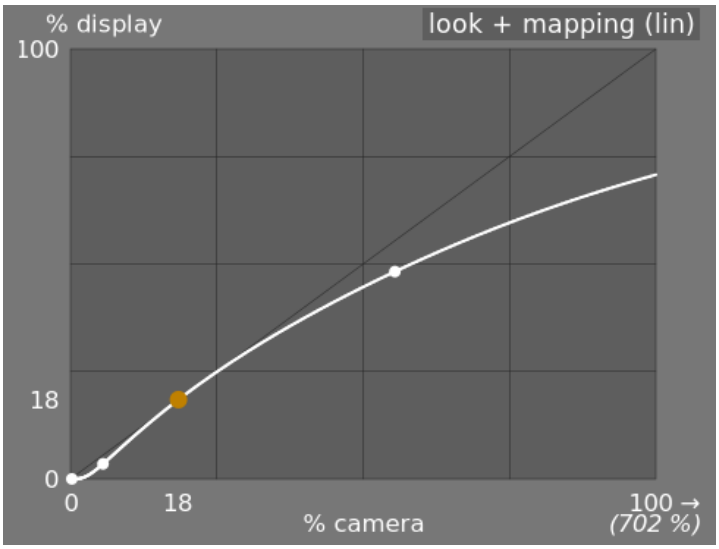
look only

This is the default view. The main bright curve shows how the dynamic range of the scene (in EV) is compressed into the display-referred output range. The orange dot shows the middle-gray point and the white dots either side mark out the linear region.



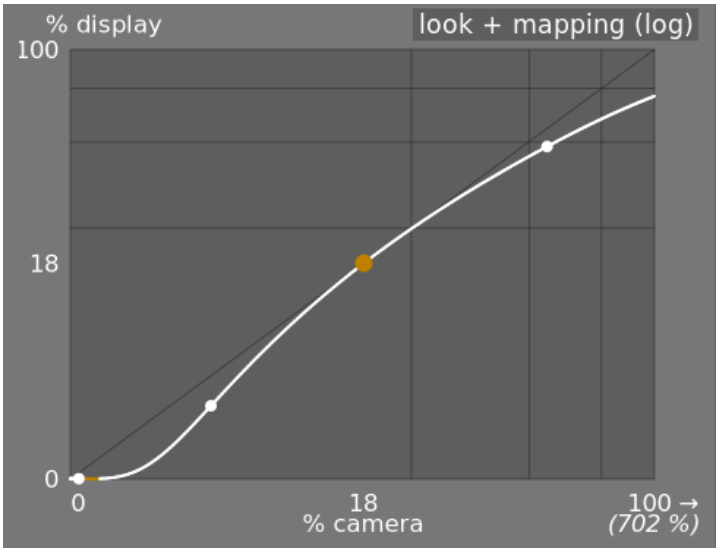
look + mapping (linear)

This view shows the mapping of input values [0,1] to output values in linear space, including the dynamic range mapping and the output transfer function. In a scene-referred workflow input values may exceed 1, but the graph is limited to the interval [0,1]. The actual value of the scene white point is shown in brackets on the *X axis*, expressed as a percentage of an input value of 1.



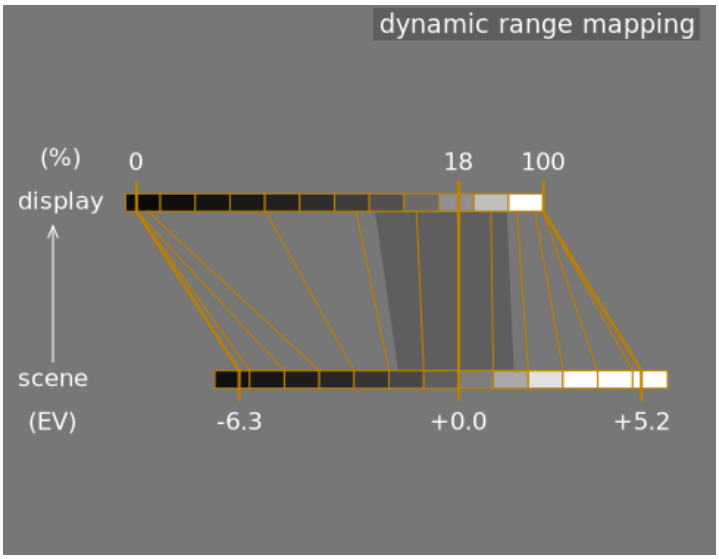
look + mapping (log)

The same as the previous view, but plotted in log space.



dynamic range mapping

This view is inspired by the Ansel Adams Zone System, showing how the zones in the input scene (EV) are mapped to the output. Middle gray from the scene is always mapped to 18% in the output (linear) space, and the view shows how the tonal ranges towards the extremes of the scene exposure range are compressed into a smaller number of zones in the display space, leaving more room for the mid-tones to be spread out over the remaining zones. The linear region is represented by the darker gray portion in the middle.



The look views also warn you when the curve is not usable:

- An **orange section** at the bottom or top of the curve means that the spline overshoots the limits of the display.
- A **red dot** means that the linear region is pushed too far towards the top or the bottom.
- A **dot drawn as a half circle** means that contrast is too low for the dynamic range of the image. In each case *filmic rgb* sanitizes the parameters internally, but the result is no longer what the sliders suggest. See troubleshooting for the corrective actions.

module controls

scene

The *scene* tab defines black, gray and white for the photographed scene. It is comparable to a levels tool, except that it works on scene-referred values in EV rather than on display-referred percentages.

white relative exposure

The number of stops (EV) between the scene middle-gray luminance and the scene luminance to be remapped to display white (peak-white). This is the right bound of the scene dynamic range that will be represented on the display – everything brighter than this value on the scene will be clipped (pure white) on the display. The picker tool reads the maximum luminance in RGB space over the drawn area, assumes it is pure white, and sets the white exposure parameter to remap the maximum to 100% luminance.

black relative exposure

The number of stops (EV) between the scene middle-gray luminance and the scene luminance to be remapped to display black (maximum density). This is the left bound of the scene dynamic range that will be represented on the display – everything darker than this value on the scene will be clipped (pure black) on the display. This slider therefore controls how far the shadows are recovered.

The picker tool reads the minimum luminance in RGB space over the drawn area, assumes it is pure black, and sets the black exposure parameter to remap the minimum to 0% luminance. This measurement is very sensitive to noise and cannot tell actual black apart from noise. It works best on low ISO images with high quality demosaicing. A resulting value of –16 EV means that the measurement has failed and the value has to be set by hand.

dynamic range scaling and auto-tune

The auto-tune picker combines the above pickers, and allows you to set the white and black exposures at the same time, using the maximum of the drawn region as the white and the minimum as the black. This gives good results in landscape photography but usually fails for portraits and indoor scenes.

When no true white and black are available on the scene, the maximum and minimum RGB values read on the image are not valid assumptions any more. Dynamic range scaling symmetrically shrinks or enlarges the detected dynamic range and the current parameters. This works with both pickers, and adjusts the current values of white and black relative exposures.

middle-gray luminance (hidden by default)

This slider defines which luminance in the scene is treated as middle-gray and remapped to 18% on the display. Use the *exposure* module for this instead – that is why the slider is hidden. To show it, enable *use custom middle-gray values* in the *options* tab.

Changing this value acts like a brightness correction. High values preserve the highlights but recover little shadow detail; low values recover the shadows but compress the highlights more harshly and reduce local contrast. The white and black relative exposures are adjusted automatically to keep the dynamic range from clipping, and can be corrected afterwards.

Note: The dynamic range shown here (white EV minus black EV) is not the dynamic range of your sensor as measured by DxOMark or PhotonsToPhotos. Several modules process the image before *filmic rgb*, so its input has a theoretically infinite dynamic range. The value only describes pixel encoding, not sensor capability.

The scene-referred workflow applies a black level correction of –0.0002 in the *exposure* module, which usually results in a dynamic range of around 12.3 EV at the input of *filmic rgb*. If a black relative exposure of –16 EV still fails to unclip the blacks, lower that correction further.

---

reconstruct

The controls in this tab blend the transition between clipped and unclipped areas and can reconstruct color from adjacent pixels. They are designed for small light sources that could not be exposed differently – a bare light bulb or the sun in the frame – and diffuse their edges in the way film would. They are not designed to recover large clipped areas or to in-paint missing parts of the image.

Disabling the *highlight reconstruction* module gives the algorithm more data to work with, since that module clips highlight data by default. This may produce magenta highlights, which can be corrected with the *gray ↔ colorful details* slider.

Reconstruction works in two steps: first define a mask that identifies the areas to be treated, then balance the trade-offs of the reconstruction algorithm.

highlights clipping

These controls define which areas of the image the reconstruction algorithms are applied to.

enable highlight reconstruction

Highlight reconstruction is disabled by default, for performance reasons, and should only be enabled when required. It is also enabled automatically as soon as the *threshold* or *transition* sliders are adjusted.

threshold

Pixels brighter than this threshold are treated by the reconstruction algorithm. The unit is EV, relative to the white point set in the *scene* tab. Enable *display highlight reconstruction mask* and adjust the threshold until the areas you want to reconstruct are shown in white. The raw overexposed warning shows you beforehand which pixels are clipped in the raw file, and whether they are clipped in one channel or in all of them.

transition

This control softens the transition between clipped and valid pixels. Moving it to the right blurs the mask, producing a smoother blend; moving it to the left sharpens the mask and reduces the feathering.

display highlight reconstruction mask

Click the icon to the right of this label to show the reconstruction mask. Keep it enabled while adjusting the two controls above.

balance

These controls balance the trade-offs between the reconstruction algorithms.

structure ↔ texture

This control decides whether the algorithm paints in a smooth color gradient (structure) or reconstructs texture from sharp detail found in unclipped pixels (texture). The default of 0% favors both equally. Where all three channels are clipped, no texture is available – move the slider to the left. Where only one or two channels are clipped, the remaining channels still carry detail – move the slider to the right.

bloom ↔ reconstruct

This control decides whether the algorithm reconstructs sharp detail in the clipped areas (reconstruct) or applies a blur that approximates the blooming of traditional film (bloom). The default of 100% maximizes sharpness. Moving the slider to the left adds blur, which tends to darken the highlights and may make the reconstruction more colorful.

gray ↔ colorful details

This control decides whether the algorithm recovers monochromatic highlights (gray) or colorful ones. Move it to the right for more color in the highlights, to the left to desaturate them. Desaturating helps when magenta or out-of-gamut colors appear.

look

The *look* tab shapes the S-curve. Watch the *look only* graph while adjusting it: the curve should rise smoothly from the black point at the bottom left to the white point at the top right. If it does not, see troubleshooting .

contrast

This slider controls the slope of the middle, linear part of the curve, and therefore mostly affects the mid-tones. The larger the dynamic range of the scene, the more contrast is needed for a natural-looking image. Note that global contrast also affects acutance – a low-contrast image looks unsharp even when it is optically sharp in the sense of the Optical Transfer Function (OTF) .

A value of 1 almost completely disables the S-curve, leaving only a small residual effect from the splines in the shadows and highlights.

The curve is built by computing virtual nodes from the module parameters and interpolating between them. It consists of a linear middle part and two ends that transition smoothly to the limits of the exposure range. Unlike in a curve module, the nodes cannot be moved manually.

linear region

The range around middle-gray that is mapped with the slope set by *contrast*, expressed as a percentage of the dynamic range defined in the *scene* tab. This is the luminance range that is rendered with priority – and also the range that is not desaturated at the extremes (see *highlights saturation mix*).

As a rule of thumb, make the linear region as large as the curve allows. Whatever is not covered by it has to be handled by the two compressed ends of the curve, so a larger linear region means more of the image is rendered at the contrast slope.

There is a limit, though: enlarging the linear region leaves less room for the ends of the curve, which then start to overshoot. Enlarge it until the graph warns you, then back off. If you need more room at one end only, shift the region with *shadows ↔ highlights balance* instead of shrinking it; reducing the contrast also relaxes the constraint.

shadows ↔ highlights balance

By default the linear region is centered on the dynamic range. This slider shifts it towards the shadows or the highlights, giving one end of the tonal range more room than the other.

highlights saturation mix

At the extremes of the tonal range, pixels tend towards pure white or pure black, neither of which can carry saturation. This slider controls how that transition is made. Positive values keep the highlights saturated – suitable for skies, but to be used with care on skin tones. Negative values bleach the highlights. The default of 0 mixes both strategies.

Since the gamut mapping introduced in *v6* handles the extremes accurately, there is no longer a separate desaturation curve to shape. In practice this slider is now a bleaching control for the highlights.

It is not intended for creative saturation. Handle saturation earlier in the pipeline instead, for example with the “add basic colorfulness” preset in the *color balance rgb* module. The technical background is described under gamut mapping .

hardness (hidden by default)

The power function applied to the output transfer function. It is calculated automatically from the values in the *scene* tab. To set it by hand, disable *auto adjust hardness* in the *options* tab.

This parameter is often called “gamma”, a term that is ambiguous in imaging and is avoided here. It raises or compresses the mid-tones to account for display non-linearities and to avoid quantization artifacts in 8-bit output. Output ICC profiles normally perform this step themselves, but the signal leaving *filmic rgb* is logarithmically encoded, which those profiles do not account for. *filmic rgb* therefore pre-compensates so that middle-gray is mapped to 45% instead of 76%.

display

The parameters in this tab rarely need to be changed. They define the target luminances of the output medium. Note that darktable uses the same pipeline for the display preview and for the exported file.

target black luminance
The black level of the target medium. The default is the lowest non-zero value that the output color space can encode. Setting it to zero maps some non-zero luminances to 0 and may lose detail in the deepest shadows. Raising it produces faded blacks for a “retro” look. Non-zero values also constrain the S-curve and can make it harder for *filmic rgb* to find a smooth spline.

target white luminance
The white level of the target medium. Values below 100% produce muted whites for a “retro” look.

target middle-gray (hidden by default)
The middle-gray of the output medium, used as the target for the central node of the S-curve. On gamma-corrected media the effective value is computed as middle-gray^(1/gamma), so 18% with a gamma of 2.2 results in an actual target of 45.87%. This duplicates the *exposure* control in the *exposure* module, which should be used instead.

Note: The default values are correct for almost all purposes. The exception is output in a linear RGB space (Rec. 709, Rec. 2020) intended for media that handle log-encoded data. To remove the compression described under *hardness* , set the hardness to 1.0 and the target middle-gray to 45%.

---

options

color science
Defines the algorithms used by *filmic rgb*. New edits use *v7 (2023)*, which is the version described on this page. Images edited with an earlier version keep that version so that existing edits are not changed. See legacy color science for the older variants.

contrast in highlights
The curvature at the highlights end of the spline. *safe* never over- or undershoots but has muted contrast near white. The default, *hard*, constrains the slope more tightly, giving a sharper curve and more tonal compression in the highlights. *soft* relaxes the constraint for a gentler curve with less compression.

contrast in shadows
The same for the shadows end of the spline. *safe* never over- or undershoots but has muted contrast near black, *hard* (the default) compresses the shadows more, and *soft* compresses them less.

Which setting to choose: Keep *hard* unless the graph shows a problem – it suits most scenes. Switch the affected end to *soft* if that end looks over-compressed, or if you want to enlarge the linear region further. Switch it to *safe* if the spline keeps overshooting and the other adjustments have not helped: *safe* always produces a usable curve, at the price of muted contrast near black or white. The two ends are set independently, so change only the one that causes the problem.

use custom middle-gray values
Shows the *middle-gray luminance* slider in the *scene* tab. Disabled by default, since middle-gray should be set in the *exposure* module.

auto-adjust hardness
Enabled by default. When disabled, the *hardness* slider appears in the *look* tab.

iterations of high-quality reconstruction
The number of passes of the highlight reconstruction algorithm. The default reconstruction works on separate RGB channels in a single pass. The high-quality reconstruction works on RGB ratios and propagates color from neighboring pixels into the clipped areas over several passes, which produces more neutral highlights at a higher processing cost. It is useful where channel clipping has produced magenta highlights.

Too many iterations make the reconstruction degenerate: colors from further away bleed into the clipped areas – white clouds filled with blue sky, or the sun disc behind trees filled with leaf green.

add noise in highlights
Adds noise to the reconstructed highlights so that they do not look unnaturally smooth next to areas that still contain sensor noise.

type of noise
The statistical distribution of the added noise. *poissonian* noise is closest to natural sensor noise, *gaussian* is less faithful but visually more pleasing and closer to film grain. Most denoising modules turn sensor noise from poissonian to slightly gaussian, so choose whichever blends better into the noise actually present in the image.

troubleshooting

the curve is not smooth

If the graph shows an orange section, a red dot or a half-circle dot (see graphic display), the requested curve cannot be realized. Try the following, in this order:

- check that *white relative exposure* and *black relative exposure* in the *scene* tab actually match the scene,
- reduce the *linear region* and/or the *contrast* in the *look* tab, or increase the contrast if a dot is drawn as a half circle,
- shift the linear region with *shadows ↔ highlights balance*,
- set *contrast in shadows* and/or *contrast in highlights* in the *options* tab to *safe* or *hard*,
- reduce *target black luminance* in the *display* tab if it is not zero.

the highlights have no color

Work through the causes in this order, because each one makes the next impossible to judge:

- Check the *white relative exposure* in the *scene* tab. If it is set below the brightest tones of the scene, those tones are clipped to pure white before any saturation control can act on them. No later setting can bring their color back.
- Check whether the highlights are clipped in the raw file itself, using the raw overexposed warning . Clipped channels carry no color information; see *reconstruct* .
- Only then raise *highlights saturation mix* in the *look* tab. High values can produce fringes around bright areas.
- If the highlights are neither clipped nor bleached but the image as a whole looks flat, add saturation in the *color balance rgb* module rather than here.

color artifacts

filmic rgb preserves hue much more strictly than a simple per-channel curve. It is therefore less forgiving of **invalid** colors such as chromatic aberrations and clipped magenta highlights, which a per-channel curve would have hidden – though not solved.

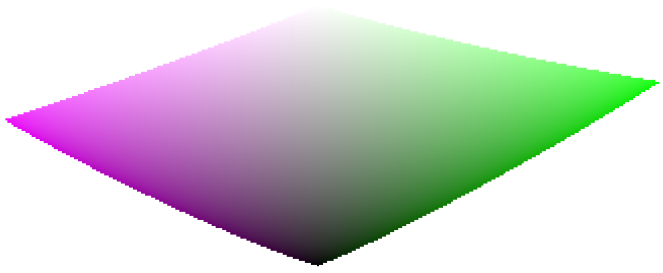
A tone mapping operator cannot repair a damaged signal. Correct such defects earlier in the pipeline with the modules provided for the purpose. As a quick workaround, any color brighter than the *white relative exposure* degrades to pure white, so setting the white relative exposure slightly below the exposure of the clipped area will render it as clean white: if it is clipped at the input, let it be clipped at the output.

different output for different color spaces

If you export the same image to sRGB and to Adobe RGB and compare both on a wide-gamut screen, the sRGB version has more desaturated highlights. This is expected. The sRGB gamut boundary lies closer to the neutral axis, so less chroma is available at a given luminance. It is a sign that the gamut mapping is working, not a bug.

gamut mapping

Very bright and very dark colors cannot be highly saturated at the same time. The graph below shows the gamut of the sRGB color space at the constant hue of its green primary, with lightness on the vertical axis and chroma on the horizontal axis:



As lightness approaches pure black or pure white, the available chroma shrinks to zero. If colors are allowed to leave the gamut, they are clipped when converting to the display color space. That clipping preserves neither hue nor luminance, so highlights typically shift towards yellow and appear darker than their surroundings.

filmic rgb therefore performs a test conversion into the display color space. If the result falls outside the [0; 100]% range, it computes the maximum saturation available in gamut at that luminance and hue, and clips the color to that value. This keeps the hue constant throughout the whole tone mapping and gamut mapping operation while making full use of the available gamut. The hue used as a reference is the hue at the input of *filmic rgb*, before any tone mapping.

The gamut is taken from the output color profile, so the mapping adapts to any output space. Only *matrix* and *matrix + curve(s)* ICC profiles are supported. With *LUT* profiles the gamut mapping falls back to the pipeline working space (Rec. 2020 by default).

The remaining trade-off – how much saturation to keep in the highlights – is controlled by the *highlights saturation mix* slider in the *look* tab. It mixes between two strategies: preserving the ratios between the RGB channels relative to the *max RGB* norm, which keeps highlights saturated, and applying the curve per channel, which bleaches them. Whatever the setting, the output saturation is never allowed to exceed the input saturation.

legacy color science

Images edited with an earlier version of *filmic rgb* keep the color science they were edited with, so that their rendering does not change. The *color science* setting in the *options* tab makes these versions available: *v3 (2019)*, *v4 (2020)*, *v5 (2021)* and *v6 (2022)*. They differ in how they desaturate colors close to pure black and pure white:

- *v3* to *v5* desaturate the extremes of the tonal range along fixed curves, which is a conservative approximation of the gamut boundary. The *look only* graph shows an additional, darker “desaturation curve” for these versions.
- *v6* introduced the measured gamut mapping described above.
- *v7* replaced the *preserve chrominance* setting with the single *highlights saturation mix* slider. In *v3* to *v6*, the *preserve chrominance* setting selects the norm N by which the RGB channels are divided before the S-curve is applied to N, so that the ratios between the channels are preserved:
- *no* applies the curve to each channel independently. This saturates the shadows and desaturates the highlights, which can help with out-of-gamut blues or reds.
- *max RGB* uses the maximum of the R, G and B channels. This is the behavior of the original version of the module. It tends to darken blues, especially skies, and may produce halos or fringes where channels are clipped.
- *luminance Y* uses a linear combination of the channels. It darkens reds and increases local contrast in them, and behaves poorly with saturated or out-of-gamut blues.
- *RGB power norm* uses $(R^3 + G^3 + B^3)/(R^2 + G^2 + B^2)$ and is usually a good compromise between *max RGB* and *luminance Y*.
- *RGB euclidean norm* is independent of the working color profile and gives the same result in any RGB space. It desaturates the highlights more than the power norm and is probably closest to a color film look. Several controls have been renamed over time. The tooltips in darktable always describe the behavior of the version currently selected.

current name	**former name(s)**
linear region	latitude
hardness	target power factor function
highlights saturation mix	extreme luminance saturation (v3, v6), mid-tones saturation (v4, v5)

8.2.32. framing

Technical information

description
add solid borders or margins around the image.

purpose
creative.

input
linear or non-linear, RGB, display-referred.
processing
geometric, RGB.
output
linear or non-linear, RGB, display-referred.

Generate a frame around the image.

The frame consists of a border (with a user-defined color) and a frame line within that border (with a second user-defined color). Various options are available to control the geometry and color of the frame.

module controls

border size
The size of the frame as a percentage of the underlying full image.
aspect
The aspect ratio of the final module output (i.e. the underlying image plus the frame). Right-click on the slider to enter a custom aspect as a ratio (e.g. “6:5”).
orientation
The orientation of the frame (portrait/landscape). Select ‘auto’ for darktable to choose the most reasonable orientation based on the underlying image.
horizontal/vertical position
Select from a set of pre-defined ratios to control where the underlying image will be positioned on the horizontal/vertical axis. You can also right click and enter your own ratio as “x/y”.
frame line size
The percentage of the frame line size, relative to the border size at its smallest part.
frame line offset
The position of the frame line, relative to the underlying image. Choose 0% for a frame line that touches the image. Choose 100% for a frame line that touches the outer border.
border color / frame line color
A pair of color selectors which allow the border and frame line colors to be defined. Clicking on the colored field will open a color selector dialog which offers a choice of commonly-used colors, or allows you to define a color in RGB color space. You can also activate a picker to take a color probe from the image.
show guides
Tick the box to show guide overlays whenever the module is activated. Click the icon on the right to control the properties of the guides. See guides & overlays for details.

8.2.33. graduated density

Technical information

description
simulate an optical graduated neutral density filter.
purpose
corrective and creative.
input
linear or non-linear, RGB, scene-referred.
processing
non-linear, RGB.

output
non-linear, RGB, display-referred.

Simulate a graduated density filter in order to correct exposure and color in a progressive manner.

A line is shown on screen allowing the position and rotation of the gradient to be modified with the mouse.

This module is known to provoke banding artifacts under certain conditions. You should consider activating the *dither or posterize* module to alleviate these issues.

module controls

density
Set the density of the filter (EV). A low value underexposes slightly whereas a high value creates a strong filter.
hardness
The progressiveness of the gradient. A low value creates a smooth transition, whereas a high value makes the transition more abrupt.
rotation
The rotation of the filter. Negative values rotate clockwise. The rotation can also be set by dragging the end of the gradient line with the mouse.
hue
Choose a hue to add a color cast to the gradient.
saturation
The saturation of the color cast to add to the gradient (defaults to a neutral color cast of 0)

8.2.34. grain

Technical information

description
simulate silver grains from film.
purpose
creative.
input
non-linear, Lab, display-referred.
processing
non-linear, Lab.
output
non-linear, Lab, display-referred.

Simulate the grain of analog film. The grain is processed on the L channel of Lab color space.

module controls

coarseness
The grain size, scaled to simulate an ISO number.
strength
The strength of the effect.
mid-tone bias
The bias toward the mid-tones. As the bias increases, the grain becomes less visible in the shadows and highlights.

8.2.35. haze removal

Technical information

description
remove fog and atmospheric hazing from images.

purpose
corrective.
input
linear, RGB, scene-referred.
processing
frequential, RGB.
output
linear, RGB, scene-referred.

Automatically reduce the effect of dust and haze in the atmosphere. This module may also be employed more generally to give images a color boost specifically in low-contrast regions of the image.

Haze absorbs light from objects in the scene but it is also a source of diffuse background light. The haze removal module first estimates, for each image region, the amount of haze in the scene. It then removes the diffuse background light according to its local strength and recovers the original object light.

Setting both of the module's controls to unity maximizes the amount of haze removal but is also likely to produce some artifacts. Removing the atmospheric light entirely may render the image flat and result in an unnatural looking style. Optimal values are typically below unity and are rather image dependent, but also a matter of personal aesthetic preferences.

Note: While this module does increase contrast, is not intended for this purpose, and will fail on many images that do not contain actual haze. To increase contrast on non-hazy images, you can use *local contrast* , *diffuse or sharpen* , or use the 4-ways tab in *color balance rgb* (reduce the shadows and lift luminance).

module controls

strength
The amount of haze removal. At unity, the module removes 100 percent of the detected haze up to the specified distance. Negative values increase the amount of haze in the image.
distance
Limit the distance up to which haze is removed. For small values, haze removal is restricted to the foreground of the image. Haze is removed from the entire image if the distance parameter is set to unity. If the *strength* is negative the distance control has no effect.

8.2.36. highlight reconstruction

Technical information

description
avoid magenta highlights and try to recover highlights colors.
purpose
corrective.
input
linear, raw, scene-referred.
processing
reconstruction, raw.
output
linear, raw, scene-referred.

Attempt to reconstruct color information for pixels that are clipped in one or more RGB channel.

clipping

Clipping occurs when the amount of captured light exceeds either the capacity of a camera's sensor to record that light (photosite saturation) or the capacity of the Raw file to store it (digital clipping). Once a pixel is clipped we can no longer know the precise brightness of that pixel – only that it is equal to or greater than the maximum value that pixel can store.

Ideally, the photosite saturation point would be the same as the value at which digital clipping occurs (to make maximum use of the camera's dynamic range) but these values often differ between cameras. Darktable uses a camera's "white point" to determine whether or not a given channel is clipped. If the white point is set incorrectly for a given camera, this can lead to valid pixels being clipped and can adversely impact the effectiveness of this module.

When a camera captures light (using a normal Bayer sensor) each pixel represents a single color (R,G,B), which is then interpolated by the demosaic module to calculate the color of neighboring pixels. The result will often be pixels (in the demosaiced image) that are clipped in some (R,G,B) channels but not others.

If these pixels are left partially clipped it can result in unrealistic colors appearing in the image. These incorrect colors can then be further skewed by the white balance module, which adjusts the ratios of the R, G, and B channels to account for the overall color of the scene. For example, where only the G channel is clipped (and R and B are close to clipping) the white balance module can cause the R and B channels to be adjusted above the clipping point of the G channel leading to pink highlights that would otherwise have been white.

The crude method to resolve this is to clip the R and B channels to the clipping point of the G channel (the "clip highlights" reconstruction method), but this can result in the loss of valid pixel data that may be useful in highlight reconstruction, and may also cause other artifacts and hue shifts.

highlight reconstruction methods

A number of highlight reconstruction methods are provided within this module. These methods all use unclipped channels and/or adjacent pixels to reconstruct the missing data.

inpaint opposed (default)

Restore clipped pixels by using an average of adjacent unclipped pixels to estimate the correct color. This works well for the majority of images but may fail where the clipped areas are adjacent to areas of a different color.

segmentation based

A more sophisticated algorithm that uses adjacent unclipped pixels to estimate the correct color, by treating each clipped area separately (as an individual segment). The color of each clipped segment is estimated by analysing the color ratios of the adjacent pixels. Pixels that are too dark or appear to be an edge are rejected by the algorithm. If all surrounding pixels are rejected, that segment is reconstructed using the "inpaint opposed" method (above). Segments that are close together are often parts of the same object and so can be treated as if they were a single segment.

Segmentation based reconstruction is able to rebuild large areas where all channels are clipped by examining the surrounding gradients. However, you should think of this method more as a way to "disguise" clipped areas with something plausible, rather than a way to "magically" repair them.

If you see irregular patterns or magenta casts you should reduce the threshold value. After selecting a proper threshold you might also need to modify the candidating value. There is no OpenCL code for the segmentation based mode, so there is a performance penalty when using it.

guided laplacians

Use an algorithm (derived from the *diffuse or sharpen* module) to replicate details from valid channels into clipped channels and to propagate color gradients from valid surrounding regions into clipped regions. This is a computationally-intensive method designed for maximum smoothness and seamless blending of the reconstructed regions into their neighborhood, and is designed primarily to reconstruct spotlights and specular reflections. This mode is available for Bayer sensors only.

clip highlights

Clamp all pixels to the white level (i.e. clip the remaining color channels). This method is most useful in cases where clipped highlights occur in naturally desaturated objects (e.g. clouds).

reconstruct in LCh

Analyse each pixel with at least one clipped channel and attempt to correct the clipped pixel (in LCh color space) using the values of the other (3 for Bayer or 8 for X-Trans) pixels in the affected sensor block. The reconstructed highlights will still be monochrome, but brighter and with more detail than with "clip highlights". This method works fairly well with a high-contrast base curve, which renders highlights desaturated. As with *clip highlights* this method is a good option for naturally desaturated objects.

Note: When using the highlight reconstruction included with the *filmic rgb* module it may be better to avoid using this module in *clip highlights* mode (so that *filmic rgb* has more information to work with).

module controls

common controls

method
The method used to reconstruct highlights.

clipping threshold
Pixels above this value are considered to be clipped.

clipping mask
Click the icon beside the slider to visualise what areas of the image are considered to be clipped (the clipping mask). If the clipping mask does not match the RAW overexposed warning , you may need to correct the threshold value.

Note: The clipping mask is also available as a raster mask for use in subsequent modules. This can be used, for example, to manage any color cast that remains after highlight reconstruction.

“guided laplacians” mode

noise level
Add Poisson noise (natural photon noise such as you would find in sensor readings) to the clipped regions. For high-ISO images, the valid regions of the image will be noisy, but the reconstructed clipped areas will be smooth, which may look odd. Adding some noise in the reconstruction helps to visually blend the result with the rest of the image.

iterations
The *guided laplacians* mode is an iterative process that extrapolates gradients and details from the neighborhood. Each new iteration refines the previous reconstruction but adds more computations that will make the module slower. The default number of iterations should provide reasonable results but you can increase if magenta highlights are not completely recovered – increase this parameter gradually but carefully, to manage the speed/quality trade-off.

inpaint a flat color
Inpainting a flat color is an algorithmic trick that may help recover magenta highlights in difficult cases (large blown areas) by smoothing RGB ratios. It can be seen as a “reconstruction booster” that may reduce the number of iterations required to entirely remove magenta in clipped highlights. However, this also makes the reconstruction less accurate and can lead to non-smooth reconstructed edges and unrelated colors being inpainted (e.g. blue sky or green leaves bleeding into white clouds). Use this setting with caution.

diameter of the reconstruction
The *guided laplacians* mode uses a multi-scale algorithm that tries to recover details from each scale independently. The *diameter of the reconstruction* is the largest scale used by the algorithm. Large scales will increase memory consumption as well as runtimes, and may also cause unrelated colors or details to be inpainted in clipped regions. You are advised to use a diameter roughly twice as large as the largest clipped area to be reconstructed. It is also possible that a given diameter may not suit all clipped areas, in which case you should use several instances at different scales and mask the clipped areas accordingly.

“segmentation based” mode

clipping threshold
Since this controls the locations of pixels that are considered to be clipped, it also changes the size of the resulting segments and the location of the adjacent pixels used for the reconstruction. For accurate adjustment, you can use the exposure module to ensure that no highlights are clipped in the histogram (or the image you see on screen). Then raise the clipping threshold until the highlights are no longer white and slowly lower it again until they look acceptable.

combine
The radius at which close segments are combined and considered to be part of the same segment. Increase (to combine more segments) when different parts of the same object have been incorrectly reconstructed to different colors. Decrease (to separate segments) when different objects have been incorrectly reconstructed to the same color. Click on the button beside the slider to see the outlines of the resulting segments.

candidating
Choose whether to prefer choosing candidate pixels (used to obtain color data) with segmentation analysis (high values) or inpaint opposed (low values). Click the button beside the slider to show the segments that are considered to have good candidates.

rebuild
Choose how to rebuild areas that have all channels clipped. The “small” and “large” modes are tuned for segment sizes less than 25 and greater than 100 pixels in diameter, respectively. The “flat” modes attempt to ignore narrow un-clipped features (powerlines, branches) to avoid gradients. Finally, the “generic” modes attempt to find the best settings for each segment.

“guided laplacians” mode and filmic’s highlight reconstruction

It is important to note that the *highlight reconstruction* module is quite early in the pixel pipeline – before *input color profile* and the full chromatic adaptation in *color calibration* (if you use the *modern chromatic adaptation* workflow). A common trick to solve clipped highlights is to simply desaturate them to white but, because white is not defined before the full chromatic adaptation and the input color profiling, it is not possible to use this trick here. Technically, there is no color yet at this point in the pipeline, only an arbitrary 3D signal.

The *guided laplacians* approach has been designed specifically to be immune to white-balance discrepancies and to avoid any concept or method related to color (so there is no explicit desaturation). It only handles gradients (transitions) in the signal and aims at connecting them smoothly, in order to fill the missing parts. This process is quite heavy though, since it falls into the category of supervised machine learning (gradient-based optimization through multi-scale curvature), which is a sub-branch of artificial intelligence.

Filmic’s highlight reconstruction uses a simpler color propagation algorithm coupled with a desaturation option that can favor an *achromatic* reconstruction. Not only does it *know* about color (because it comes after the full color profiling and chromatic adaptation) but it also uses a simplified and faster version of the algorithm used by the *guided laplacians* approach. Namely, this variant will not try as hard to restore details and will favor a smooth blur instead.

The filmic reconstruction is good enough for very large clipped patches and offers the benefit of being able to degrade to white as a last resort. It is also better and faster to inpaint solid color into clipped areas, at the expense of details. Its main drawback is that it is not as selective in the source of the colors being inpainted in clipped parts, so it may inpaint unrelated colors.

All in all, you are advised to use the *guided laplacians* highlight reconstruction mode to:

1. smooth the boundaries of clipped areas,
2. recover spotlights and clipped areas of diameter below approximately 256px (on the full-resolution RAW),
3. remove chromatic aberrations, which can occur during demosaicing (the next module in the pipeline) at the boundary between clipped and valid regions.

If you find yourself having to increase the *diameter of reconstruction* past 512px to get a full recovery from magenta, the best approach is usually to cap the diameter to 512px, do the most you can with this setting, and then enable filmic’s highlight reconstruction to finish the work. This will give more bearable run-times with a very similar result.

8.2.37. highpass

Technical information

description
isolate high frequencies in the image.

purpose
creative.

input
linear or non-linear, Lab, scene-referred.

processing
frequential, Lab.

output
special, Lab, scene-referred.

A high pass filter.

This module is primarily intended to be used in combination with a *blend mode* . For example, try using the module with a blend mode of “soft light” for high pass sharpening.

Note: *This module performs blurs in Lab color space, which can result in undesirable effects, and is no longer recommended. Instead, use the contrast equalizer module for fine sharpness or the local contrast module for general sharpness.*

Module Controls

sharpness
The sharpness of the filter
contrast boost
The contrast boost

8.2.38. hot pixels

Technical information

description
remove abnormally bright pixels by dampening them with neighbors.
purpose
corrective.
input
linear, raw, scene-referred.
processing
reconstruction, raw.
output
linear, raw, scene-referred.

Automatically detect and eliminate hot pixels.

Hot pixels are pixels which have failed to record a light level correctly. Detected hot pixels are replaced by an average of their neighbors.

module controls

threshold
The minimum brightness for a pixel to be considered a hot pixel candidate.
strength
The sensitivity of detection; higher values will fix more pixels by requiring less deviation from their neighbors.
detect by 3 neighbors
Extend the detection of hot pixels – regard a pixel as hot if a minimum of only three (instead of four) neighbor pixels deviate by more than the threshold level.
mark fixed pixels
Visually mark the corrected pixels on the image and display a count of hot pixels that have been fixed.

8.2.39. input color profile

Technical information

description
convert any RGB input to pipeline reference RGB using color profiles to remap RGB values.
purpose
mandatory.
input
linear or non-linear, RGB, scene-referred.
processing
defined by profile.
output
linear, RGB, scene-referred.

Define how darktable will interpret the colors of the image.

This module takes the color space used by the image source (e.g. camera, scanner) and converts the pixel encodings to a standardized working color space. This means that subsequent modules in the pipeline don't need to be concerned with the specifics of the input device, and can work with and convert to/from a common working color space.

Where an image has been captured in a raw file, the input color profile module will normally apply either a standard or enhanced color matrix specific for that camera model, which will be used to map the colors into the working profile color space. If color space information is embedded in the image, the *input color profile* module will use this information when mapping the colors to the working profile color space. The user can also explicitly specify a color space for the incoming image, and can even supply a custom ICC color profile specifically made for the input device.

As part of the mapping from the input color space to the working profile space, the colors can be confined to a certain gamut using the *gamut clipping* options, which can help to mitigate some (infrequent) color artifacts. This is also influenced by the chosen rendering intent .

Note that the final color profile that will be used when exporting the image is controlled by the *output color profile* module.

module controls

input profile
The profile or color matrix to apply. A number of matrices are provided along with an enhanced color matrix for some camera models. The enhanced matrices are designed to provide a look that is closer to that of the camera manufacturer.

You can also supply your own input ICC profiles and put them into `$DARKTABLE/share/darktable/color/in` (where `$DARKTABLE` is the darktable installation directory) or `color/in` in the darktable configuration directory . Note that these `color/in` directories are not created by the darktable install; if you need to use one, you must create it yourself. One common source of ICC profiles is the software that is shipped with your camera, which often contains profiles specific to your camera model. You may need to activate the *unbreak input profile* module to use your own profiles.

If your input image is a low dynamic range file like JPEG, or a raw file in DNG format, it might already contain an embedded ICC profile, which darktable will use by default. You can restore this default by selecting "embedded icc profile". If you hover your mouse over the input profile combobox on such an image, details of the embedded profile will be shown in a tooltip.

working profile
The working profile used by darktable's processing modules. Each module can specify an alternative space that it will work in, and this will trigger a conversion. By default darktable will use "linear Rec. 2020 RGB", which is a good choice in most cases.

gamut clipping
Activate a color clipping mechanism. In most cases you can leave this control in its default "off" state. However, if your image shows some specific features such as highly saturated blue light sources, gamut clipping might be useful to avoid black pixel artifacts. See possible color artifacts for more information.

Choose from a list of RGB profiles. Input colors with a saturation that exceeds the permissible range of the selected profile are automatically clipped to a maximum value. "linear Rec. 2020 RGB" and "Adobe RGB (compatible)" allow for a broader range of unclipped colors, while "sRGB" and "linear Rec. 709 RGB" produce a tighter clipping. Select the profile that prevents artifacts while still maintaining the highest color dynamics.

8.2.40. lens correction

Technical information

description
correct lenses optical flaws.
purpose
corrective.
input
linear, RGB, scene-referred.
processing
geometric and reconstruction, RGB.
output
linear, RGB, scene-referred.

Automatically correct for (or simulate) lens distortion, transverse chromatic aberrations (TCA) and vignetting.

You can choose to either use lens correction data embedded in your RAW file (where present/supported) or correction data provided by the external Lensfun library .

Additional controls are also provided for manual vignetting correction, in case the profiles for your lens are insufficient or missing.

Note that if TCA correction is enabled in this module, also using *raw chromatic aberrations* may cause artifacts from over-correction.

Lensfun correction data

If your system's Lensfun library has no correction profile for the automatically identified camera/lens combination, the controls for the three photometric parameters (below) are replaced with a warning message. You may try to find the right profile yourself by searching for it in the menu.

If your lens is present in the list but has not been correctly identified, this may require some adjustment within the exiv2 program (see this post for details). Note that you may need to re-import the images once such adjustments have been made as the lens name is retrieved as part of the import process.

By default, only lenses that are directly compatible with your camera's mount are listed and automatically identified. If you are using lenses for a different mount with an adapter (for example a Canon EF lens adapted to a Nikon Z body), then you must run the `lensfun-add-adapter` tool to enable those lenses.

If you can't find your lens, check if it is in the list of currently supported lenses , and try running the `lensfun-update-data` tool. If there is still no matching profile for your lens, a lens calibration service is offered by Torsten Bronger, one of darktable's users. Alternatively you may visit the Lensfun project to learn how to generate your own set of correction parameters. Don't forget to share your profile with the Lensfun team!

module controls

correction method
Choose which method to use to correct distortions. Additional controls will be provided depending on the option selected:
- "Lensfun database": Use corrections provided by the Lensfun project.
- "embedded metadata": Use corrections embedded in the metadata of the RAW file. This is only available if supported metadata is found.
- "only manual vignette": Do not perform any automatic correction but provide manual vignette correction.

corrections
Choose which corrections (distortion, TCA, vignetting) to apply. Change this from its default "all", if your camera has already performed some internal corrections (e.g. vignetting), or if you plan to undertake some corrections with a separate program.

corrections done
Occasionally, for a given camera/lens combination, only some of the possible corrections are supported. This message box appears at the bottom of the module to indicate which corrections have actually been applied to the image.

show guides
Tick the box at the bottom of the module to show guide overlays whenever the module is activated. Click the icon on the right to control the properties of the guides. See guides & overlays for details.

Lensfun controls

The following controls are provided for the “Lensfun” correction method only:

camera
The camera make and model as determined by the image’s Exif data. You can override this manually and select your camera from a hierarchical menu. Only lenses with correction profiles matching the selected camera will be shown.

lens
The lens make and model as determined by the image’s Exif data. You can override this manually and select your lens from a hierarchical menu. This is mainly required for pure mechanical lenses, but may also be needed for off-brand / third party lenses.

photometric parameters (focal length, aperture, focal distance)
Lens corrections depend on certain photometric parameters that are read from the image’s Exif data: focal length (for distortion, TCA, vignetting), aperture (for TCA, vignetting) and focal distance (for vignetting). Many cameras do not record focal distance in their Exif data, in which case you will need to set this manually.

You can manually override all automatically selected parameters. Either take one of the predefined values from the drop-down menu or, with the drop-down menu still open, just type in your own value.

target geometry
In addition to correcting lens flaws, this module can change the projection type of your image. Set this combobox to the desired projection type (e.g. “rectilinear”, “fisheye”, “panoramic”, “equirectangular”, “orthographic”, “stereographic”, “equisolid angle”, “Thoby fisheye”). To correct the aspect ratio of an anamorphic lens, use the *rotate and perspective* module.

scale
Adjust the scaling factor of your image to avoid black corners. Press the auto scale button (to the right of the slider) for darktable to automatically find the best fit.

mode
The default behavior of this module is to *correct* lens flaws. Switch this combobox to “distort” in order to instead *simulate* the flaws/distortions of a specific lens (inverted effect).

TCA override
Check this box to override the automatic correction parameters for TCA. This will expose the TCA red and TCA blue parameters below. Un-check the box to revert back to automatic corrections.

TCA red; TCA blue
Override the correction parameters for TCA. You can also use these sliders to manually set the parameters if the lens profile does not include TCA correction. Look out for colored seams at features with high contrast edges and adjust the TCA parameters to minimize those seams.

Note: TCA corrections will not be applied to images that have been identified as monochrome (see developing monochrome images for more information).

Note: The lens correction module will fill in missing data at the borders by repeating the borders’ pixels. For strong corrections, this filling can be visible (especially on noisy images). Crop the image if necessary.

---

embedded metadata controls

The following controls are provided for the “embedded metadata” correction method only:

use latest algorithm
This control appears for images using an older version of the “embedded metadata” correction algorithm. Check this box to irreversibly change to the newer algorithm.

The following controls can be revealed by clicking the “fine-tuning” button:

distortion
Fine-tune the distortion / chromatic aberration correction.

vignetting
Fine-tune the vignetting correction.
TCA red
Fine-tune the red chromatic aberration correction.
TCA blue
Fine-tune the blue chromatic aberration correction.
image scale
Override the image scaling.

manual vignette correction

Full vignetting correction is unavailable or inadequate for many lenses, whether using embedded metadata or the Lensfun database. Click on the “manual vignetting correction” button to provide additional adjustments via the following controls.

strength
the overall strength of the effect.
radius
the amount of the image that is unchanged by the correction.
steepness
the steepness of the correction effect outside of the radius.

The effect of the correction can be visualized by clicking on the mask button beside the strength slider.

manual distortion correction

Manual distortion corrections are not natively supported.

A workaround can help to make manual corrections:

- Switch *correction method* to *Lensfun database*, then pick the “generic” profiles, choose a “generic” rectilinear lens (e.g. *“Generic, Crop-factor 1.0 (Full Frame)”*) and set *target geometry* to *orthographic*.
- Use *mode* to set the direction of the applied effect (*correct* for barrel; *distort* for pincushion) and the *mm* adjustment to tune the magnitude of the correction.
- Use the *scale* slider to make the image fit and if need be fine tune with the *target geometry*.
- If you find a good fit for your lens (or the respective zoom range of a lens with variable focal length) you can then save the settings as a preset and set it up to auto apply if appropriate exif metadata is available.

8.2.41. liquify

Technical information

description
distort parts of the image.
purpose
creative.
input
linear, RGB, scene-referred.
processing
geometric, RGB.
output
linear, RGB, scene-referred.

Move pixels around by applying freestyle distortions to parts of the image using points, lines and curves.

As you might want to use source data from any part of the image you will be shown the uncropped image (possibly with the cropping rectangle overlaid as a guide) while the module is active.

nodes

Each of liquify's tools is based on nodes. A point consists of a single node and a line or curve consists of a sequence of linked nodes defining a path.

Each instance of the liquify module is limited to a maximum of 100 nodes – for more nodes, use additional instances. However, please note that the liquify module consumes a lot of system resources.

Drag the central point of a node to move the node around. The radius describes the area of the effect (distortion occurs only within this radius). To change the radius drag the handle at the circumference. A strength vector starting from the center describes the direction of the distortion, and its strength is depicted by the length of the vector. Change the vector by dragging its arrow head.

points

A point consists of a single node and strength vector.

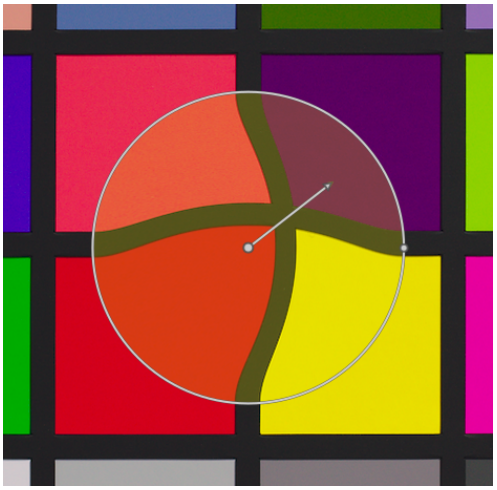
Click the point icon to activate the point tool and then click on the image to place it. Hold Ctrl while clicking on the point icon to add multiple points without having to click the icon again each time. Right-click to exit creation mode.

point modes

The strength vector of a point has three different modes. These can be toggled by holding Ctrl and clicking on the arrowhead of the strength vector.

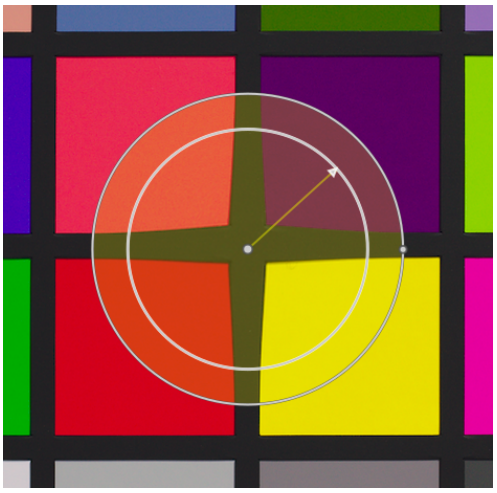
linear

A linear distortion inside the circle, starting from the opposite side of the strength vector and following the vector's direction. This is the default mode.



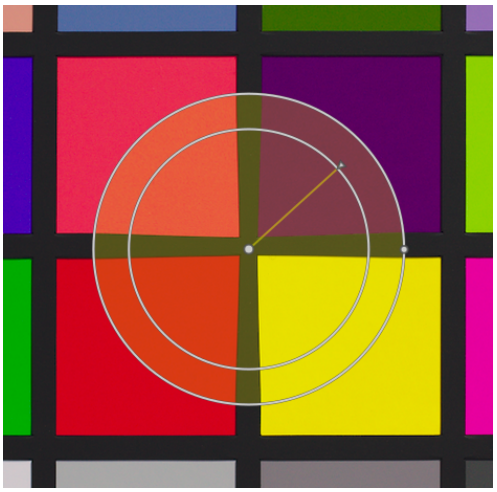
radial growing

The strength vector's effect is radial, starting with a strength of 0% in the center and increasing away from the center. This mode is depicted by an additional circle with the arrow pointing outwards.



radial shrinking

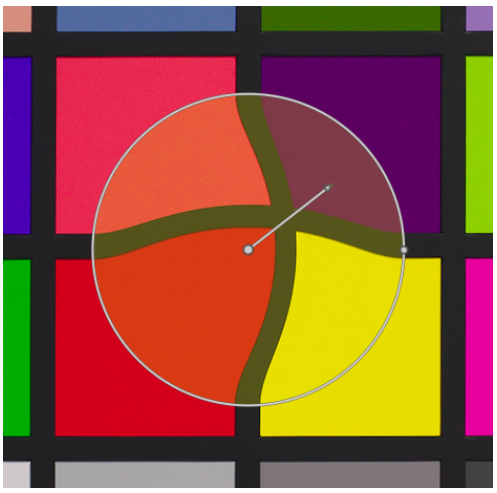
The strength vector’s effect is radial, starting with a strength of 100% in the center and decreasing away from the center. This mode is depicted by an additional circle with the arrow pointing inwards.



feathering

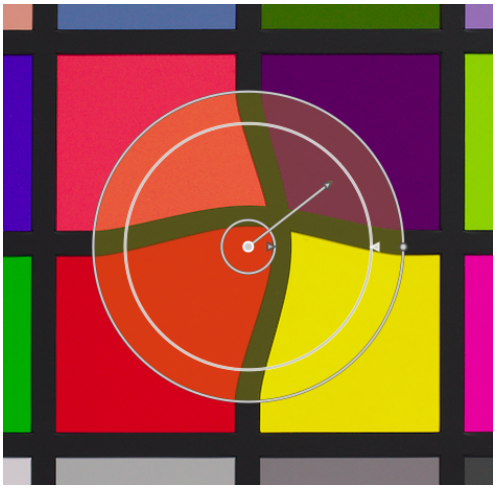
default mode

By default the strength varies linearly from 0% to 100% between the center and the radius of the control point. It is possible to modify the feathering effect by clicking on the center of the circle.



feathered mode

In “feathered” mode, two control circles are displayed, which can be modified independently to feather the strength of the effect. Note that clicking the center of the circle again hides the feathering controls but does *not* return to the default mode.



removing points

A point can be removed by right-clicking on the center of the node.

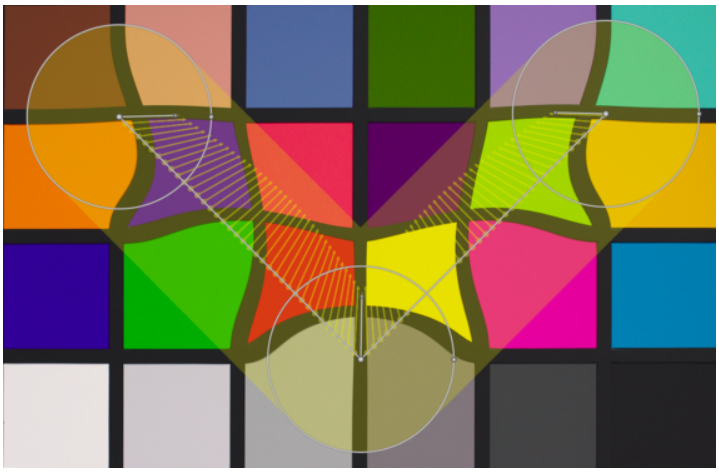
lines and curves

Lines and curves are sequences of points linked together by straight or curved lines. The effect is interpolated by a set of associated strength vectors.

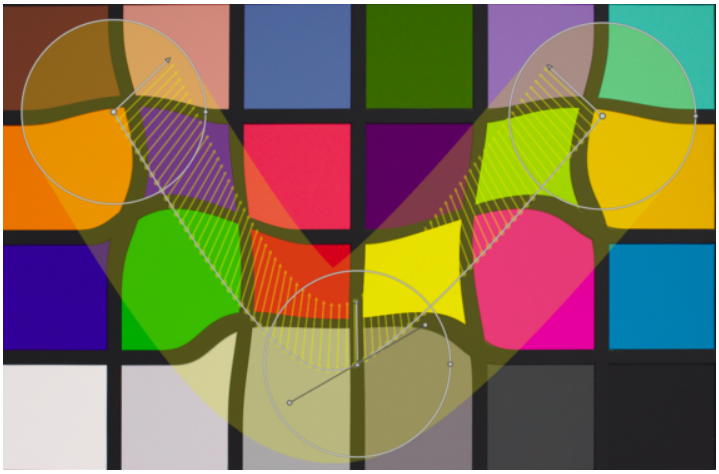
Click the appropriate icon to activate the line or curve tool and then click on the image to place a sequence of points forming the path. Right-click anywhere when the last point has been placed in order to finish drawing the line/curve.

Hold Ctrl while clicking on the line/curve icon to add multiple lines/curves without having to click the icon again each time. Right-click a second time to exit creation mode after the final line or curve has been completed.

lines



curves



Ctrl+click on a line or curve segment to add a new control point. Ctrl+right-click on the center of a node to remove a control point.

Right-click on a segment to remove the shape completely. Ctrl+Alt+click on a segment to change that segment from a line to a curve and vice versa.

link modes

Ctrl+click on the center of a node to change the way the points of a curve are linked together. There are four modes, which correspond to different ways of handling the steepness of the bezier curve using control handles:

autosmooth
This is the default mode, in which control handles are not displayed – controls are automatically computed to give a smooth curve.

cusp
Control handles can be moved independently. This mode is depicted by a triangle symbol in the node's center.

smooth
Control handles always give a smooth curve. This mode is depicted by a diamond symbol in the node's center.

symmetrical
Control handles are always moved together. This mode is depicted by a square symbol in the node's center.

view and edit nodes

Click the node tool icon to activate or deactivate the node edit tool. This displays all currently-defined distortion objects and their controls. Alternatively you can right-click on the image at any time for the same effect.

warps and nodes count

This information field displays the number of warps (individual distortion objects) and nodes currently in use.

show guides

Tick the box to show guide overlays whenever the module is activated. Click the icon on the right to control the properties of the guides. See guides & overlays for details.

8.2.42. local contrast

Technical information

description
manipulate local and global contrast separately.

purpose
creative.

input
non-linear, Lab, display-referred.
processing
non-linear, Lab.
output
non-linear, Lab, display-referred.

Enhance the image’s local contrast.

This is achieved using either a *local laplacian* (default) or *unnormalized bilateral* filter. Both modes work exclusively on the L channel from Lab. The *local laplacian* filter has been designed to be robust against unwanted halo effects and gradient reversals along edges.

module controls

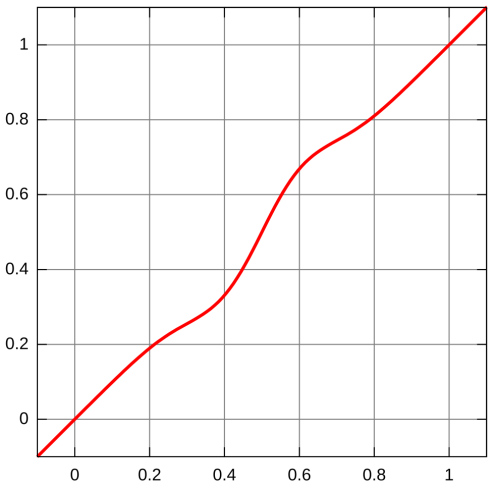
mode
Choice of *local laplacian* filter or *bilateral grid*. The following sections define the controls available for each of these modes.

bilateral grid

coarseness
Adjust the coarseness of the details to be adjusted.
contrast
Control how strongly the algorithm distinguishes between brightness levels. Increase this parameter for a more contrasty look.
detail
Add or remove detail. Higher values increase local contrast.

local laplacian

To understand the parameters of the local laplacian filter, one can think of it as applying a curve to the image, similar to the following graph:



This curve is applied to the image in a way that works locally and avoids halo artifacts.

The local laplacian mode also supports shadow lifting and highlight compression, similar to the *shadows and highlights* module.

detail
Add or remove detail. Higher values increase local contrast. This inserts an S shaped element in the center of the curve, to increase or decrease local contrast.

highlights
This affects one end of the S shaped contrast curve, effectively increasing or compressing contrast in the highlights. A low value pulls the highlights down.

shadows
Similar to the highlights parameter, this affects the other end of the contrast curve, and increases or decreases contrast in the shadows. A higher value gives more contrast in the shadows. A lower value lifts the shadows and can effectively simulate a fill light. Note that this is done with local manipulation of the image. However, this means that a completely dark image cannot be brightened in this way – only dark objects in front of bright objects are affected.

mid-tone range
This controls the extent of the S shaped part of the contrast curve. A larger value makes the S wider, and thus classifies more values as mid-tones and fewer values as highlights and shadows. In higher dynamic range settings it can be useful to reduce this value to achieve stronger range compression, by lowering the contrast in the highlights and the shadows. Note however, that for really strong HDR scenarios this may work best in combination with a *base curve* that pre-compresses the range, perhaps with an approximately logarithmic curve. The exposure fusion feature in the *base curve* module may lead to more pleasing results at times, but is also more prone to halo effects.

This setting can cause banding artifacts in the image if pushed to extreme values. This is due to the way in which darktable computes the fast approximation of the local laplacian filter.

8.2.43. lowlight vision

Technical information

description
simulate human night vision.
purpose
creative.
input
non-linear, Lab, display-referred.
processing
linear, XYZ.
output
non-linear, Lab, display-referred.

Simulate human lowlight vision. This module can also be used to perform a day-to-night conversion.

The idea is to calculate a scotopic vision image, which is perceived by the rods rather than the cones in the eyes under low light. Scotopic lightness is then mixed with photopic value (regular color image pixel) using some blending function. This module can also simulate the Purkinje effect by adding some blueness to the dark parts of the image.

This module comes with several presets, which can be used to get a better idea of how the module works.

module controls

curve
The horizontal axis represents pixel lightness from dark (left) to bright (right). The vertical axis represents the kind of vision from night vision (bottom) to day vision (top).
blue
Set the blue hint in shadows (Purkinje effect).

8.2.44. lowpass

Technical information

description
isolate low frequencies in the image.
purpose
creative.

input
linear or non-linear, Lab, scene-referred.
processing
frequential, Lab.
output
special, Lab, scene-referred.

Apply a low pass filter (for example, a Gaussian blur) to the image, while controlling the output contrast and saturation.

This module is primarily intended to be used in combination with a blend mode . For example, try using the *local contrast mask* preset with the *overlay* blend mode.

Note: *This module performs blurs in Lab color space, which can produce undesirable effects, and is no longer recommended. Instead, use the contrast equalizer module for blurring or the tone equalizer module for dynamic range compression.*

---

module controls

radius
The radius of the blur.
soften with
The blur algorithm to use:
- *gaussian blur*: Blur all image channels (L, a, b)
- *bilateral filter*: Blur the L channel only, while preserving edges

contrast
Higher absolute values increase contrast. Lower absolute values reduce contrast. Negative values result in an inverted negative image. A value of zero leads to a neutral plane.
brightness
Negative values darken the image. Positive values lighten the image.
saturation
The color saturation. Negative values result in complementary colors by inverting the a/b channels. Higher absolute values increase color saturation. Lower absolute values reduce color saturation. A value of zero fully desaturates the image.

8.2.45. LUT 3D

Technical information

description
perform color space corrections and apply look.
purpose
corrective or creative.
input
linear, RGB, display-referred.
processing
defined by profile, RGB.

output
linear or non-linear, RGB, display-referred.

Transform RGB values with a 3D LUT file.

A 3D LUT is a tridimensional table that is used to transform a given RGB value into another RGB value. It is normally used for film simulation and color grading.

This module accepts `.cube`, `.3dl`, `.png` (haldclut) and `.gmz` files. Uncompressed 3D LUT data is not saved in the database or the XMP file, but is instead saved to the 3D LUT file path inside the 3D LUT root folder. It is therefore important to back up your 3D LUT folder properly – sharing an image with its XMP is pointless if the recipient doesn't also have the same 3D LUT file in their own 3D LUT folder.

The compressed format `.gmz` is only available when GMIC is installed. This format can hold a full library of LUTs, and LUT data loaded from this type of file *can* be saved to the database and XMP files.

Note: the module clips all values outside of the range [0,1]. You may have to reduce the range of the input before applying.

---

usage

LUTs are most commonly used in darktable for color grading or film look simulation. For this reason, by default, the module is placed after the *filmic rgb* module in the pixelpipe and should be applied to a neutral image (without first applying a specific look). While you can find hundreds of free LUTs on the internet, you should note that not all of them are compatible with the darktable environment and workflow – incompatible LUTs will not produce the advertised look. To limit the risk, a color grading LUT should have been created to work with one of the available "application color spaces" (see below), for both the input and the output of the module.

Camera log LUTs (as F-log or S-Log3) are different to color-grading and film-look-simulation LUTs, and are intended to convert the camera log raw data into something (linear raw data or other color space) that darktable is able to understand. In this case the *LUT 3D* module should be manually placed between the *demosaic* and *input color profile* modules. Once you have done this, you can no longer choose an "application color space". The "input profile" of input color profile module should be aligned with the output of the LUT. *Please note that this use case has not yet been tested.*

module controls

file selection
Choose the 3D LUT file to use. File selection is inactive if the 3D LUT root folder has not been defined in preferences > processing .

application color space
A 3D LUT is defined relative to a specific color space. Choose the color space for which the selected 3D LUT file has been built. Cube files are usually related to Rec. 709 while most others are related to sRGB.

interpolation
This defines how to calculate output colors when input colors are not exactly on a node of the RGB cube described by the 3D LUT. There are three interpolation methods available: tetrahedral (default), trilinear and pyramid. Usually you won't see any difference between interpolation methods except with smaller sized LUTs.

8.2.46. monochrome

Technical information

description
quickly convert an image to black & white using a variable color filter.

purpose
creative.

input
linear or non-linear, Lab, display-referred.

processing
non-linear, Lab.

output
non-linear, Lab, display-referred.

Quickly convert an image to black & white using a variable color filter.

To use this module, first reduce the filter size (to concentrate its effect) and move it across the hue palette to find the best filter value for your desired image rendition. Then gradually expand the filter to include more hues and achieve more natural tonality.

Although this module is easy to use, better results can usually be obtained by using the film emulation presets in the *color calibration* module.

Note: Under certain conditions, especially highly saturated blue light sources in the frame, this module may produce black pixel artifacts. Use the gamut clipping option of the *input color profile* module to mitigate this issue.

---

module controls

filter size/position
The default central location of the filter has a neutral effect but dragging it to an alternate position applies a filter analogous to taking a b&w photograph through a conventional color filter.

A picker can be activated to automatically set the position and size of the filter based on the selected portion of the image. Scroll the mouse wheel to change the filter size, making the filter's range of hues more or less selective.

highlights
Control how much to retain highlights.

8.2.47. negadoctor

Technical information

description
invert film negative scans and simulate printing on paper.
purpose
corrective and creative.
input
linear, RGB, display-referred.
processing
non-linear, RGB.
output
non-linear, RGB, display-referred.

Process scanned film negatives.

You can obtain an image of a negative using a film scanner, or by photographing it against a white light (e.g. a light table or computer monitor) or off-camera flash.

preparation

If the image of the negative was obtained using a digital camera, then in order to obtain accurate colors in the final image, you will need to take the following points into consideration:

- When taking the photograph, adjust the exposure to fully utilise the entire dynamic range of your camera sensor – i.e. “expose-to-the-right”, so that the histogram in your camera touches the right hand side without clipping the image.
- Ensure the white balance is correctly set up to compensate for the light source used to illuminate the negative. You can take a profiling image of the light source with no film negative in front of it, and then use the “from image area” feature in the *white balance* module to obtain a reference white-balance setting. This reference white balance setting can then be made into a style or simply pasted onto the images taken from your film negatives.
- Apply the standard or enhanced camera matrix in the *input color profile* module.

When scanning or photographing your film negative, make sure you include some unexposed part of the film within the captured image. This is required to set the *Dmin* parameter (see below). If this is not possible (e.g. your film holder completely obscures the unexposed parts of the film), you can take a separate image of an unexposed part of the film, measure the *Dmin* parameter from that image, and then paste that setting to the rest of the images from that film.

When developing the scanned/photographed film negatives, it is recommended that you disable any tone mapping modules such as *filmic rgb* and *base curve* .

The *working profile* parameter in darktable’s *input color profile* module should be set to either *linear Rec. 2020 RGB*, or to an ICC profile representing the actual color space of your film emulsion. Some examples of such ICC profiles may be found in the following forum posts:

- https://discuss.pixls.us/t/any-interest-in-a-film-negative-feature-in-rt/12569/177
- https://discuss.pixls.us/t/input-color-profile-to-use-for-negatives/20271/13

Note: if you want to use the *tone equalizer* with *negadoctor*, you’ll need to move the *tone equalizer* module **after** *negadoctor* in the pixelpipe , since the *tone equalizer* is not designed to work with negatives.

---

module controls

It is strongly recommended that you set the parameters following the order in which they are presented in the GUI. Start by setting the *film stock*, then work through each of the tabs (*film properties* , *corrections* , *print properties*) in order, working from top to bottom in each tab.

When using pickers , be careful to avoid including dust and scratches, which can skew the data taken from the sampled region.

film stock

The first step is to choose “color” or “black and white” in the *film stock* drop-down. If you select “black and white”, any sliders that are only used for color will be hidden from view.

film properties

This tab contains a number of basic settings. If, after adjusting these settings, your image is still not quite as you would like it, you can make further adjustments on the *corrections* tab. These are technical settings, and serve a similar purpose to the *scene* tab in the *filmic rgb* module, in that they adjust the black and white points and hence define the dynamic range of the negative.

color of the film base

Sample an area of the base film stock from your scan. This is the area just outside of the image (an unexposed part of the film). If you are working with black and white negatives, you can leave this slider at its default value (white). If working on color film, click the picker to the right of the color bar, which will create a bounding box covering about 98% of your image. Then, click and drag across an area of your negative which contains only unexposed film stock. This will automatically calculate values for the *D min* slider(s). It is likely at this point that your image will still look too dark, but you can correct this later.

D min

If the *film stock* is set to “black and white”, this slider indicates the minimum value corresponding to the unexposed film stock. If the *film stock* is set to “color”, this control will consist of 3 separate sliders, one for each of the red, green and blue channels.

D max

This slider represents the dynamic range of your film, and it effectively sets the film’s white point. Dragging this slider to the left will make the negative brighter. Dragging it to the right will make the negative darker. When adjusting this slider manually, it’s a good idea to closely watch your histogram to ensure that you don’t clip the highlights (where the histogram has been pushed over too far off the right hand side of the graph). If you click the picker icon (on the right) negadoctor will automatically calculate this value to ensure maximal use of the histogram without clipping. To use the picker, click and drag to draw a rectangle across only the exposed parts of the negative. Don’t include the unexposed film stock, as this will skew the result.

scan exposure bias

This slider allows you to set the black point. It is a technical adjustment that ensures a proper zeroing of the RGB values and a spreading of the histogram between [0, 1] values for robustness in the operations that follow. Dragging this to the left will make the negative brighter. Dragging to the right will make the negative darker. When adjusting this slider manually, it’s a good idea to closely watch your histogram to ensure that you don’t clip the shadows (where the histogram is pushed too far off the left hand side of the graph). If you click the picker negadoctor will automatically calculate any required offset. To use the picker, select a region in the darkest lowlights, or select the entire image without including any unexposed film stock. Double-check the histogram to ensure the left part of it doesn’t clip.

corrections

This tab contains sliders that allow you to make color cast corrections within both the shadow and highlight regions.

The settings on this tab should not be needed for most well-preserved negatives. It is mostly useful for old and poorly-preserved negatives with a decayed film base that introduces undesirable color casts.

The other case where these color cast corrections may be needed is if the white balance properties of the light used to scan the film negative are significantly different to the light source under which the original film camera took the shot. For example, if you illuminate the film with an LED light, but the original shot was taken under daylight, this may require some additional color cast corrections.

shadows color cast

These three sliders allow you to correct for color casts in the shadows by adjusting the *red*, *green* and *blue* channels individually. Use the picker to set the sliders automatically by selecting a neutral gray shadow region requiring correction. Because the shadows sliders can also affect color casts in the highlights, it is important to finish setting the shadows sliders before moving on to the highlights sliders.

highlights white balance

These three sliders allow you to correct the white balance in the highlights by adjusting the *red*, *green* and *blue* channels individually. Use the picker to set the sliders automatically by selecting a neutral gray region in the highlights that is not properly balanced.

print properties

This tab contains settings that mimic the tonal effect of the photochemical papers that would have been used to create the hard copy image if you were not developing the photo digitally. These are creative settings, and serve a similar overall purpose to the creative tone curve settings on the *look* tab of the *filmic rgb* module.

The *print exposure*, *paper black* and *paper grade* are analogous to the *slope*, *offset* and *power* controls in the *color balance* module (which in turn is based on the ASC CDL standard). These settings define a creative tone curve to enforce your contrast intent after the inversion, at the end of the module. The equation governing this slope/offset/power behavior is:

$$RGB_out = (RGB_in \times exposure + black)^{grade}$$

paper black (density correction)

For this slider, select the picker and click and drag to select a region that encompasses only the exposed part of the film negative. If you can see unexposed film stock around the edges of your image, ensure that these areas are excluded from the drawn rectangle when calculating the *paper black* setting. Paper black represents the density of the blackest silver-halide crystal available on the virtual paper. In the analog development process, this black density always results in non-zero luminance, but the digital pipeline usually expects black to be encoded with a zero RGB value. This slider setting lets you remap paper black to pipeline black via an offset. You can use the picker to select a region of the image that should be mapped to black in the final image.

paper grade (gamma)
This slider is your gamma (contrast) control, and it defaults to a value of 4. If all has gone well, this value (4) minus the value of D max (from the “film properties” tab) should normally leave you with a value between 2 and 3.

paper gloss (specular highlights)
This slider is essentially a highlights compression tool. As you drag this slider to the left, you will see in the histogram that the highlight values are being compressed (pushed to the left). Adjust this accordingly, so that your highlights are not clipped in the histogram. You can also use this to simulate a matte photo print with low-contrast highlights.

print exposure adjustment
This slider offers one final opportunity to correct any clipping of the highlights. If you have followed all the previous instructions carefully, you shouldn’t need to adjust this setting. Note that you can increase the print exposure while at the same time decreasing the paper gloss, which allows you to brighten the mid-tones without losing any highlights. You can use the picker to select the brightest highlights, or select the entire image without including any unexposed film stock. This will set the exposure so that the brightest part of the selected region is not clipped. Double-check the histogram to make sure that the right part of the histogram doesn’t clip.

8.2.48. orientation

Technical information

description
flip or rotate image by step of 90 degrees.
purpose
corrective.
input
linear, RGB, scene-referred.
processing
geometric, RGB.
output
linear, RGB, scene-referred.

Rotate the image 90 degrees at a time or flip the image horizontally and/or vertically.

The module is enabled by default and the orientation (rotation) is automatically set based on the image’s Exif data.

The orientation can also be set using the actions on selection module in the lighttable view.

Note: The *crop* area is retained when the orientation is changed.

module controls

transform
Double click the label to reset to the default transformations


rotate counter-clockwise
Rotate the image 90 degrees counter-clockwise


rotate clockwise
Rotate the image 90 degrees clockwise


flip horizontally
Flip the image (mirror) horizontally


flip vertically
Flip the image (mirror) vertically

show guides
Tick the box to show guide overlays whenever the module is activated. Click the icon on the right to control the properties of the guides. See guides & overlays for details.

8.2.49. output color profile

Technical information

description
convert pipeline reference RGB to any display RGB using color profiles to remap RGB values.
purpose
mandatory.
input
linear or non-linear, Lab, display-referred.
processing
defined by profile.
output
non-linear, RGB or Lab, display-referred.

Manage the output profile for export and the rendering intent to be used when mapping between color spaces.

darktable comes with pre-defined profiles *sRGB*, *Adobe RGB*, *XYZ* and *linear RGB*. You can provide additional profiles by placing them in `$DARKTABLE/share/darktable/color/out` (where `$DARKTABLE` is the darktable installation directory) or `color/out` in the darktable configuration directory . Note that these `color/out` directories are not created by the darktable install; if you need to use one, you must create it yourself.

The output color profile may also be defined within the export module.

module controls

output intent
The rendering intent for output/export. Rendering intent can only be selected when using LittleCMS2 to apply the output color profile (this can be changed in preferences > processing). If darktable's internal rendering routines are used, this option is hidden. For more details see rendering intent .

output profile
The profile used to render colors for output/export. The profile data will be embedded into the output file (if supported by the file format) allowing other applications to correctly interpret its colors. As not all applications are aware of color profiles, the general recommendation is to stick to sRGB unless you know what you are doing and have a good reason to do otherwise.

HDR output

The *PQ Rec2020*, *PQ P3*, *HLG Rec2020* and *HLG P3* profiles produce HDR images, for display on HDR screens. They use the PQ (SMPTE ST 2084) and HLG (ARIB STD-B67) transfer functions of Rec. ITU-R BT.2100 , together with Rec2020 or Display P3 primaries. To keep their brightness and color, export them in a format that can carry them with enough precision, such as 10-bit or higher AVIF, HEIF, JPEG XL or 16-bit PNG. For an introduction to HDR video and display standards, see high-dynamic-range television on Wikipedia.

darktable maps a pipeline value of 1.0 (display white for SDR output) to the HDR "reference white" of 203 nits, as recommended by Report ITU-R BT.2408 , so mid-tones look as bright as in an SDR export. Values above 1.0 are highlights brighter than reference white:

- PQ can encode up to 10000 nits. The brightest highlights reach the *PQ peak* set in the *export* module (default 1000 nits).
- HLG is converted for a nominal 1000 nit display, so highlights reach at most 1000 nits, about 4.9 times reference white.

The ratio between the peak of the output and reference white is the *headroom* of the output. The *AgX* and *sigmoid* tone mappers follow it, so that a single edit can be exported to SDR, HLG and PQ without changes.

8.2.50. raw black/white point

Technical information

description
sets technical specificities of the raw sensor touch with great care!.
purpose
mandatory.
input
linear, raw, scene-referred.
processing
linear, raw.
output
linear, raw, scene-referred.

Define camera-specific black and white points.

This module is activated automatically for all raw images. Default settings are applied for all supported cameras. Changes to the defaults are not normally required.

module controls

black level 0-3
The camera-specific black level of the four pixels in the RGGB Bayer pattern. Pixels with values lower than this level are not considered to contain valid data.
white point
The camera-specific white level. All pixels with values above this are likely to be clipped and will be handled accordingly in the *highlight reconstruction* module.
flat field correction
Use flat field correction to compensate for lens shading. This field only appears for applicable Raw files and will automatically use any GainMap embedded in the Raw. You can choose to disable this correction if desired.

8.2.51. raw chromatic aberrations

Technical information

description
correct chromatic aberrations for Bayer sensors.
purpose
corrective.
input
linear, raw, scene-referred.
processing
linear, raw.
output
linear, raw, scene-referred.

Correct chromatic aberrations of raw images.

This module currently only works for raw images recorded with a Bayer sensor (the sensor used in the majority of cameras) – for other types of image, you should use the *chromatic aberrations* module instead.

The module will also not apply any corrections to any photos that have been identified as monochrome (see developing monochrome images for more information).

This module expects good white balance data (provided by the *white balance* module) for best results. In most cases the *default* settings are sufficient.

Note that if this module is enabled, then TCA correction in the *lens correction* module should be disabled, as the two modules will conflict with one another.

module controls

iterations
The number of iterations. For most images, “twice” is sufficient, and is the default value. Occasionally, increasing this control can give better results.

avoid colorshift
If the module causes tinting – often pink or greenish – tick this box to apply a correction.

8.2.52. raw denoise

Technical information

description
denoise the raw image early in the pipeline.

purpose
corrective.

input
linear, raw, scene-referred.

processing
linear, raw.

output
linear, raw, scene-referred.

Perform denoising on raw image data before it is demosaiced .

This module has been ported from dcraw .

For an AI-based alternative that denoises the raw data as part of a combined denoise/demosaic inference pass, see neural restore and the AI features overview .

module controls

noise threshold
The threshold for noise detection. Higher values lead to stronger noise removal and greater loss of image detail.

coarse/fine curves
The noise of an image is usually a combination of fine-grained and coarse-grained noise. These curves allow the image to be denoised more or less depending on the coarseness of the visible noise. The left of the curve will act on very coarse grain noise, while the right of the curve will act on very fine grain noise.

Raising the curve will result in more smoothing, while lowering it will result in less smoothing. As an example, you can preserve very fine-grained noise by pulling down the rightmost point of the curve to the minimum value.

If you are tackling chroma noise with a blend mode , you can raise the rightmost part of the curve quite high, as colors do not change a lot on fine grain scales. This will help especially if you see some isolated pixel left un-denoised.

The best way to use the R, G, and B curves is to examine each of the channels in turn using the *color calibration* module in gray mode, denoise that channel, and then repeat for the other channels. This way, you can take into account the fact that some channels may be noisier than others. Beware that guessing which channel is noisy without actually seeing the channels individually is not straightforward and can be counterintuitive. A pixel which is completely red may not be caused by noise on the R channel, but actually by noise on G and B channels.

See the wavelet section for more details.

8.2.53. retouch

Technical information

description
remove and clone spots, perform split-frequency skin editing.

purpose
corrective.
input
linear, RGB, scene-referred.
processing
geometric and frequential, RGB.
output
linear, RGB, scene-referred.

Remove unwanted elements from your image by cloning, healing, blurring and filling using drawn shapes.

This module extends the capabilities of the deprecated *spot removal* module (equivalent to this module's "clone" tool) by including a "heal" tool (based on the heal tool from GIMP), as well as "fill" and "blur" modes. It can also take advantage of wavelet decomposition , allowing the image to be separated into layers of varying detail (from coarse to fine) which can be selectively retouched before being recombined to produce the output image.

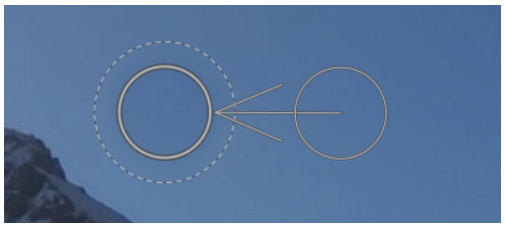
As you might want to use source data from any part of the image you will be shown the uncropped image (possibly with the cropping rectangle overlaid as a guide) while the module is active.

clone and heal

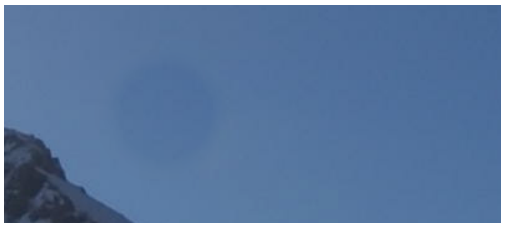
Cloning allows a part of the image (the *target*) to be hidden by replacing it with an area copied from elsewhere in the image (the *source*). For example, you may wish to remove a small cloud from a blue sky:



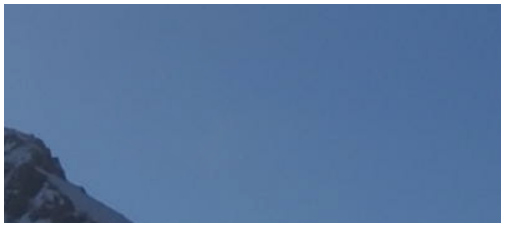
The simplest way to do this is with the *cloning* tool  . The following example uses a circular shape to clone out the cloud using the circle of blue sky beside it:



In many cases, the edges of the *source* shape won't precisely match the surroundings of the *target*, leading to unnatural looking results. In this example, the sample of sky we chose as the source was slightly darker than the target, leaving a faint outline of the circular shape used in the cloning process:



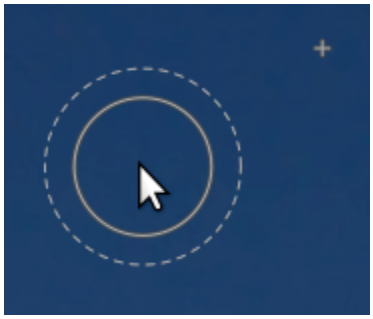
In such cases, the *heal* tool  is more appropriate. With this tool, the color and luma of the sample is blended to fit better with the surroundings. In this example, using *heal* instead of *clone* produces a much more pleasing result:



source and target shapes

Once you have chosen *heal* or *clone* mode, you must choose the shape you wish to use (*circle*, *ellipse*, *path* or *brush* – see the drawn masks section for details). The source and target patches will both use the same shape.

When you hover over the image with your mouse, a plus symbol (+) will appear to indicate where the *source* shape will be placed by default. Your normal mouse cursor will indicate the position of the target shape:



It is recommended that you choose the position of the source shape first, followed by the position of the target, as follows:

- Shift+click to set the position of the source shape in “relative mode”. The position of the “plus” (+) symbol will move to the clicked location and will remain in a fixed position *relative to your cursor* until you click on the image to begin placing your target shape. If you place subsequent shapes without first changing the target location, the source shape will be placed at the same offset from the target shape as was used for the first stroke.
- Ctrl+Shift+click to change the position of the source shape in “absolute” mode. As above, the position of the “plus” (+) symbol will move to the clicked location, but will *remain in the same absolute position* even if you move your mouse. You can then click on the image to begin placing the target shape. If you place subsequent shapes without first changing the target location, exactly the same source position will be used, fixed in the absolute coordinate system of the image.

Once you have placed your source and target shapes on the image, they can be adjusted manually with your mouse.

Ctrl+click and drag within either the source or target shape to rotate only that specific shape. Ctrl+Shift+click and drag within either shape to rotate both the source and target shapes together.

Note: For *circle* and *ellipse* shapes only, you can place both the source and target shapes with a single click-and-drag motion: Click on the desired target location and then drag, releasing your mouse button when you reach the desired source location. This operation does not affect the placement of subsequent shapes.

---

fill and blur

The *clone* and *heal* tools both require the use of another part of the image to “fill in” the region being hidden. Sometimes there is no suitable sample in the image to use for this purpose. In such cases, the *retouch* module offers two further options:

 **fill tool**

Fill the drawn region with a selected color.

blur tool

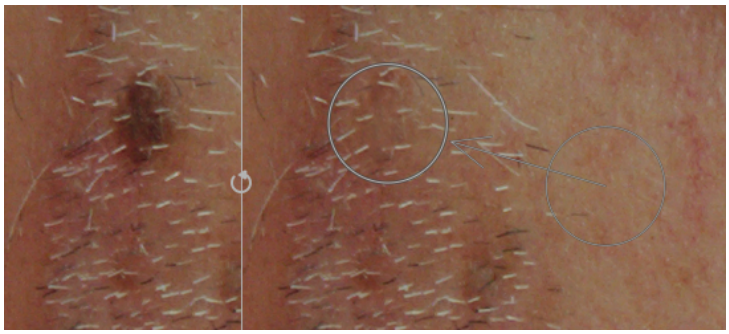
Apply a blur to the drawn region, smoothing out any details.

These two options are most useful when used together with wavelet decomposition, where they can be used to smooth over features within a selected detail layer.

wavelet decomposition

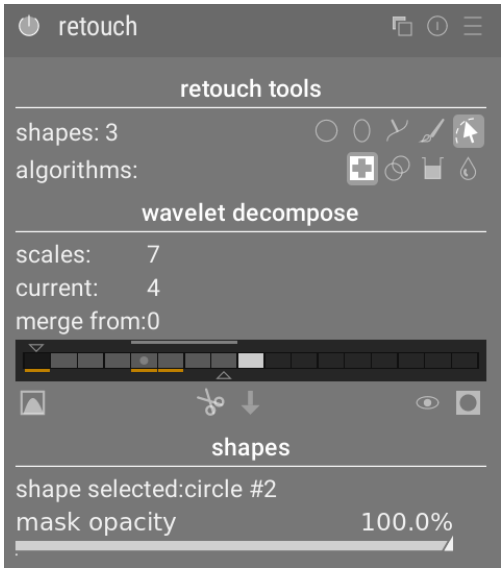
Wavelets allow an image to be decomposed into a number of layers each containing varying levels of detail, so that you can work on each detail layer independently and then recombine them at the end. This is particularly useful in portrait photography, where you can deal with skin blotches and blemishes on a coarse layer of detail, leaving the finer skin texture untouched. The *wavelets* section provides more information on the decomposition process.

This method can be used with the healing tool, for example, to paint over a spot that appears in one of the coarse detail layers, while leaving the hairs in the fine detail layers intact:



It can also be used with the blur tool to even out coarse blotches in the skin, again without impacting the finer details (see the *wavelets* section for details on this technique).

module controls



retouch tools

The *retouch tools* section consists of two items:

shapes

The number after the *shapes* label indicates how many shapes have been placed on the image, either directly or within a wavelet layer.

Click on one of the shape icons to draw a new shape on the image (see drawn masks for details).

Ctrl+click on a shape icon to draw multiple shapes continuously (right-click to cancel).

Click the *show and edit shapes*  button to show and edit any existing shapes for the currently-selected wavelet scale.

algorithms

Choose a retouching algorithm (clone, heal, fill or blur). Ctrl+click to change the algorithm used for the currently-selected shape. Shift+click to set the default algorithm (used for new images or when you reset module parameters).

wavelet decompose

The *wavelet decompose* section centres around a bar graph that shows how the image has been decomposed into detail (scale) layers. The key features of the bar graph are:

- The black square on the left represents the entire image before decomposition.
- The gray squares show the various wavelet detail layers, with fine details to the left, and coarse details to the right.
- The white square on the far right represents the residual image (the remainder of the image after the detail layers have been extracted).
- A light gray dot in a square indicates the currently-selected layer. Click on another square to select a different layer.
- The light gray bar running along the top indicates which levels of detail are visible at the current zoom level. Zoom in to see finer details.
- The triangle at the bottom shows how many layers the image has been decomposed into. Drag the triangle to the right to create more layers. Drag it to the left to decrease the number of layers. By default no wavelet decomposition is performed.
- The triangle at the top shows the current value of the “merge from” parameter (see below).
- The orange lines under the squares indicate which layers have retouch edits applied.

The remaining items in this section are:

scales

Shows how many detail layers the image has been decomposed into. Zero indicates that no wavelet decomposition has been performed.

current

Shows which layer is currently selected (also marked with the light gray dot on the bar graph).

merge from

This setting allows a single edit to be applied to multiple consecutive scales within a group starting from the coarsest scale (excluding the residual image) down to the scale selected. For example, if “merge from” is set to 3 and the maximum scale is 5 then all edits that are added to scale 5 will be applied to scales 3, 4 and 5. Edits added to scale 4 will be applied to scales 3 and 4, and edits added to scale 3 will be applied to scale 3 only. If you set *merge from* to 0, then merging is disabled, and all edits apply only to the scale that they are defined in. Setting *merge from* to the highest scale available (in this example, 5) also disables merging.

```
          merge_from
              v
  0   1   2   3   4   5    scale
              <-------o    scale 5 edits
              <---o        scale 4 edits
              o            scale 3 edits
          o                scale 2 edits
      o                    scale 1 edits
```

 **display wavelet scale**

Display the currently-selected wavelet layer on the center image. Selecting this option brings up an additional control – *preview single scale*.

preview single scale
An additional control that allows the black, white and gray points of the wavelet scale preview to be adjusted to make it easier to see. Click the  to set these values automatically. This does not affect the module's operation – only the wavelet scale preview.

 **cut**
Cut all shapes from the currently-selected layer and place them into the clipboard.

 **paste**
Move the shapes on the clipboard to the currently-selected layer.

 **temporarily switch off shapes**
Toggle all shapes (whether on the current layer or not) on or off, temporarily removing the module's effect.

 **display masks**
Show the target shapes associated with the currently-selected layer with a yellow overlay.

shapes

This section allows you to modify settings related to the currently-selected shape:

shape selected
Indicates which shape is currently selected, and what type of shape it is.

fill mode
If the *fill* algorithm has been chosen for the currently-selected shape, choose whether to "erase" or fill the selected shape with a chosen "color".

fill color
If a *fill mode* of "color" has been chosen, select the color to fill the shape with. You can click to select or enter a custom rgb value or use the picker to take a sample from the image.

brightness
If the *fill* algorithm has been chosen for the currently-selected shape, fine-tune the color's brightness. This slider also works in "erase" mode.

blur type
If the *blur* algorithm has been chosen for the currently-selected shape, choose whether to apply a "gaussian" or "bilateral" blur.

blur radius
If the *blur* algorithm has been chosen for the currently-selected shape, choose the radius of the blur.

mask opacity
Alter the opacity of the mask associated with the currently-selected shape. An opacity of 1.0 indicates that the shape is completely opaque and the module's effect is fully applied, whereas a value less than 1.0 indicates that the effect applied by the shape is blended with the underlying image to the degree indicated by the slider.

panning and zooming the image

While creating or editing a shape, mouse actions are applied to the current shape. If you need to move or zoom the portion of the image shown in the center view, hold down the 'a' key while dragging the mouse or using the scroll wheel. While the key is held down, the mouse actions will apply to the entire image rather than the current shape. Holding down the key will also temporarily suppress generating new shapes in continuous-creation mode.

8.2.54. rgb curve

Technical information

description
alter an image's tones using curves in RGB color space.

purpose
corrective and creative.

input
linear, RGB, display-referred.

processing
non-linear, RGB.
output
linear, RGB, display-referred.

A classic digital photography tool to alter an image's tones using curves.

This module is very similar to the *tone curve* module but works in RGB color space.

Activate the picker on the left to show the picked values in the graph (Ctrl+click or right-click to use the picker in area mode). Numerical (Lab) values of the input and output (see below) at the selected spot or area are shown at the top left of the widget.

A second picker to the right can be used to automatically create new nodes based on the sampled area. Ctrl+click+drag to alter the created nodes to have a positive curve for the selected area; Shift+click+drag to create a negative curve.

module controls

*Please refer to the curves section for more details about how to modify curves including the **interpolation method** and **preserve colors** controls.*

mode
RGB is a linear color space designed to capture and display images in additive synthesis. It is related to the capture and display media and does not isolate color and lightness information in the same way that Lab and XYZ color spaces do. This module works in ProPhoto RGB. Adding contrast in RGB space is known to desaturate highlights and boost saturation in the shadows, but this has been proven to be the most reliable way to edit contrast, and is the standard method in most software

Depending on the desired intent you can apply the RGB curve in two different modes

- *RGB, linked channels*: Apply the L-channel curve to all three channels in the RGB color space.
- *RGB, independent channels*: R, G and B curves can be adjusted independently.

compensate middle gray
Select this to change the histogram display in the module. This option does not alter the processing but may assist when editing the curve.

8.2.55. rgb levels

Technical information

description
adjust black, white and mid-gray points in RGB color space.
purpose
corrective and creative.
input
linear, RGB, display-referred.
processing
non-linear, RGB.

output
non-linear, RGB, display-referred.

Adjust black, white and mid-gray points in RGB color space. This module is similar to the *levels* module, which works in Lab color space.

The rgb levels tool shows a histogram of the image, and displays three bars with handles. Drag the handles to modify the black, middle-gray and white points in lightness (in “RGB, linked channels” mode) or independently for each of the R, G and B channels (in “RGB, independent channels” mode).

Moving the black and white bars to match the left and right borders of the histogram will make the output image span the full available tonal range. This will increase the image’s contrast.

Moving the middle bar will modify the mid-tones. Move it to the left to make the image look brighter and move it to the right to make it darker. This is often referred to as changing the image’s gamma.

Three pickers are available for sampling the black, white and gray points from the image.

Note: Under certain conditions, especially with highly saturated blue light sources, the *levels* module may produce black pixel artifacts. See the “gamut clipping” option of the *input color profile* module for information about how to mitigate this issue.

---

module controls

mode
The mode of operation. “RGB, linked channels” (default) provides a single levels tool which updates all channels, taking into account the selected color preservation method (see “preserve colors” below). “RGB, independent channels” provides separate levels controls for each of the R, G and B channels.

auto
Auto-adjust the black and white point and put the gray point exactly in the mean between them. Use the picker to auto-adjust based on a selected region of the image.

preserve colors
Choose a color preservation method when using “RGB, linked channels” mode (default “luminance”).

8.2.56. rgb primaries

Technical information

description
adjustment of the RGB color primaries for color grading.

purpose
corrective or creative.

input
linear, RGB, scene-referred.

processing
linear, RGB.

output
linear, RGB, scene-referred.

Adjust the hue and purity of the RGB primary colors (i.e. *which* red, green and blue they represent), while leaving uncolored (gray) pixels unchanged. In addition to preserving gray pixels, the opponency relationships between the colors are also preserved under this adjustment: If you increase the purity of the blue primary, the opponent yellow's intensity increases to balance things out; If you twist the blue hue toward cyan, the opponent yellow is twisted toward orange.

This module is essentially a channel mixer (as in the color calibration module) but with a different interface. Even though the sliders are named "red", "green" and "blue", all adjustments are global and affect the overall colorimetry of the image, just like a channel mixer does.

When applied before the *filmic rgb* or *sigmoid* tone mapping modules *rgb primaries* can be used to make small adjustments to colorimetry. When applied after the tone mapping modules it may be used to apply creative edits such as tinting.

module controls

red hue
Shift red towards yellow (positive values) or magenta (negative values)
red purity
The purity of the red primary
green hue
Shift green towards cyan (positive values) or yellow (negative values)
green purity
The purity of the green primary
blue hue
Shift blue towards magenta (positive values) or cyan (negative values)
blue purity
The purity of the blue primary
tint hue
When applied after tone mapping, this control allows the gray (achromatic) parts of the image to be tinted. When applied before tone mapping, it acts like a white balance control.
tint purity
The purity of the tint that is applied to the image

8.2.57. rotate and perspective

Technical information

description
rotate or distort perspective.
purpose
corrective or creative.
input
linear, RGB, scene-referred.
processing
geometric, RGB.

output
linear, RGB, scene-referred.

Automatically correct for converging lines, a form of perspective distortion. The underlying mechanism is inspired by Markus Hebel's *ShiftN* program. This module also allows for the rotation of the image to be adjusted.

Perspective distortions are a natural effect when projecting a three dimensional scene onto a two dimensional plane and cause objects close to the viewer to appear larger than objects further away. Converging lines are a special case of perspective distortions frequently seen in architectural photographs – parallel lines, when photographed at an angle, are transformed into converging lines that meet at some vantage point within or outside of the image frame.

This module is able to correct converging lines by warping the image in such a way that the lines in question become parallel to the image frame. Corrections can be applied in a vertical and horizontal direction, either separately or in combination. In order to perform automatic correction the module is able to analyze the image for suitable structural features consisting of line segments. You can also set the line structures manually by drawing a "perspective rectangle" or drawing multiple horizontal and vertical lines onto the image. Based on these (automatic or manually-drawn) line segments a fitting procedure is initiated, which determines the best values for the module's parameters.

As the most common use case for this module is for rotation, the perspective correction controls are hidden by default. Click the "perspective" header to expand the controls.

While the module is active (and none of the structure buttons are selected) you can right-click and drag anywhere on the image to define a horizontal or vertical line. This will cause the rotation parameter to be automatically adjusted to make the drawn line horizontal/vertical with respect to the image frame.

Note: This functionality (right-click and drag to set horizontal/vertical) is also available when the module is inactive, as long as no other function (e.g. drawn mask creation) claims the right-hand mouse button.

---

perspective correction workflow

structure

The first step is to obtain details about the horizontal and/or vertical structures in the image. Three alternative methods are provided to do this:

manually draw structure lines

Click on the  icon to enable line drawing mode and then click-and-drag on the image to draw lines that you wish to become horizontal or vertical. The module will automatically detect whether the lines are horizontal or vertical and color them green or blue, respectively. Draw as many lines as you wish (the more lines, the better the fit mechanism will work) and then click on one of the "fit" icons to complete the process. You can re-enter this mode to edit your drawn lines at any time. Edit a line by clicking and dragging on the line or the end nodes, and right-click a line to delete it. Once you are happy with your changes, re-select a "fit" icon to complete the process.

manually define a perspective rectangle

Click on the  icon to enable perspective rectangle drawing mode. This will draw a rectangle on the screen and you can grab and move the corners of the rectangle so that the left and right sides fall on lines you wish to make vertical, and the top and bottom fall on lines you wish to make horizontal. Once you are happy with your rectangle, click one of the "fit" icons to complete the process. You can re-enter this mode to edit your drawn rectangle at any time. Once you are happy with your changes, re-select a "fit" icon to complete the process.

This method is similar to how "keystone" correction works in the *crop and rotate* module.

automatically detect structure

Click the  icon to analyze the image for structural elements – darktable will automatically detect and evaluate line elements. Shift+click to apply a contrast enhancement step before performing further analysis. Ctrl+click to apply an edge enhancement step before performing further analysis. Both variations can be used alone or in combination if the default analysis is not able to detect a sufficient number of lines.

Only lines that form a set of vertical or horizontal converging lines are used for subsequent processing steps. Line segments are displayed as overlays on the image canvas, with the type of line identified by color as follows:

green
Vertical converging lines

red
Vertical lines that do not converge

blue
Horizontal converging lines

yellow
Horizontal lines that do not converge

gray
Other lines that are not of interest to this module

Lines marked in red or yellow are regarded as outliers and are not taken into account during the automatic fitting step. This outlier elimination involves a statistical process using random sampling which means that each time you press the “get structure” button the color pattern of the lines will look slightly different.

You can manually change the status of line segments: Left-Click on a line to select it (turn the color to green or blue) and Right-click to deselect it (turn the color to red or yellow). If you keep the mouse button pressed, you can use a sweeping action to select/deselect multiple lines in a row. The size of the select/deselect brush can be changed with the mouse wheel. Hold down the Shift key and keep the left or right mouse button pressed while dragging to select or deselect all lines in the chosen rectangular area.

Once you are happy with the detected lines, select a “fit” icon to complete the process.

fit

Once you are happy with the identified horizontal and vertical lines, using one of the methods above, click on one of the “fit” icons to automatically set the module’s parameters based on the defined structure. The image and the overlaid lines are then displayed with perspective corrections applied.

You may choose to automatically apply just the vertical corrections  , just the horizontal corrections  , or both together  . Ctrl+click on any of the icons to apply a rotation without the lens shift. Shift+click on any of the icons to apply the lens shift without any rotation.

rotate

Once you are happy with the applied perspective corrections, you may wish to perform a final rotation correction either by adjusting the rotation parameter or right-clicking and dragging the image to define a horizontal/vertical line.

module controls

rotation
Control the rotation of the image around its center to correct for a skewed horizon. To rotate by more than the default soft limit of ten degrees, right click and enter the desired value up to 180 degrees (see module controls).

automatic cropping
When activated, this feature crops the image to remove any black areas at the edges caused by the distortion correction. You can either crop to the “largest area”, or to the largest rectangle that maintains the original aspect ratio (“original format”). In the latter case you can manually adjust the automatic cropping result by clicking in the clip region and moving it around. The size of the region is modified automatically to exclude any black areas.

lens shift (vertical)

Correct converging vertical lines (i.e. to make the *green* lines parallel). In some cases you can obtain a more natural looking image if you correct vertical distortions to an 80 ~ 90% level rather than to the maximum extent. To do this, reduce the correction slider after having performed the automatic correction.

lens shift (horizontal)

Correct converging horizontal lines (i.e. to make the *blue* lines parallel).

shear

Shear the image along one of its diagonals. This is required when correcting vertical and horizontal perspective distortions simultaneously.

lens model

This parameter controls the lens focal length, camera crop factor and aspect ratio that used by the correction algorithm. If set to “generic” a lens focal length of 28mm on a 35mm full-frame camera is assumed. If set to “specific”, the focal length and crop factor can be set manually using the sliders provided.

focal length

If the *lens model* is set to “specific”, set the lens focal length. The default value is taken from the image’s Exif data, and can be overridden by adjusting the slider manually.

crop factor

If the *lens model* is set to “specific”, set the camera crop factor. You will normally need to set this value manually.

aspect adjust

If the *lens model* is set to “specific”, this parameter allows for a free manual adjustment of the image’s aspect ratio. This is useful for “unsqueezing” images taken with an anamorphic lens (which changes the ratio of image height to width).

structure

Define horizontal and vertical lines in the image using a manual or automatic method (see workflow section for details).

fit

Set the distortion correction sliders automatically based on the identified structure (see workflow section for details).

show guides

Tick the box to show guide overlays whenever the module is activated. Click the icon on the right to control the properties of the guides. See guides & overlays for details.

examples

Here is an image with a skewed horizon and converging lines caused by directing the camera upwards:



Here is the image after having corrected for vertical and horizontal perspective distortions using automatic structure detection. Note the framing adjustment made by the automatic cropping feature and the still-visible overlay of structural lines:



8.2.58. rotate pixels

Technical information

description
null.
purpose
null.
input
null.
processing
null.
output
null.

The sensors of some cameras (such as the Fujifilm FinePix S2Pro, F700, and E550) have a diagonally oriented Bayer pattern instead of the usual orthogonal layout.

Without correction this would lead to a tilted image with black corners. This module applies the required rotation.

darktable detects images that need correction using their Exif data and automatically activates this module where required. For other images the module always remains disabled.

The module has no controls.

8.2.59. saturation curve

Technical information

description
remap saturation and brilliance with curves.

purpose
corrective or creative.
input
linear, RGB, scene-referred.
processing
darktable UCS or JzAzBz, perceptual.
output
linear, RGB, scene-referred.

Remaps the saturation and brilliance of an image as a function of their own input value, using two independent, editable curves. Unlike a flat saturation or vibrance slider, each curve lets you decide exactly how much correction is applied at every point along the saturation range – for example, protecting already-vivid colors while boosting near-neutral ones, or the reverse.

An optional edge-aware guided filter can smooth the value the curves read from, so the correction varies gradually across saturation edges instead of switching abruptly.

introduction

Global saturation and vibrance controls apply a single coefficient, sometimes weighted by chroma, to the whole image. This is often too blunt an instrument: skin tones, skies and foliage usually sit at different points on the saturation range, and pushing all of them by the same amount rarely gives a natural result.

Saturation curve plots a curve over the full range of a perceptually normalized saturation axis, from 0 to 100% of the local gamut boundary, and lets you add, move or remove control nodes anywhere along it. A second, independent curve does the same for brilliance, the perceptual dimension that behaves like a saturation-aware exposure change. Both curves are read using each pixel's own saturation as the lookup value, so the correction is local to the pixel's starting saturation rather than to a spatial mask, unless the guided filter is enabled (see *guided filter*).

general principles

Each pixel's RGB value is converted to a perceptual color space and reduced to two numbers: a hue angle and a normalized saturation between 0 (neutral) and 1 (the gamut boundary for that hue and lightness). This normalized value is looked up on the saturation and brilliance curves, and the resulting curve heights become multiplicative correction factors, applied at constant hue.

Two perceptual formulas are available, from the *saturation formula* setting:

JzAzBz (2021)
The original formula. Uses the JzAzBz uniform color space, which does not fully account for the Helmholtz-Kohlrausch effect and can behave less smoothly near black.
darktable UCS (2022)
The darktable Uniform Color Space, designed specifically for saturation manipulation. It accounts for the Helmholtz-Kohlrausch effect and includes a built-in, more accurate gamut-mapping model, giving smoother results across the lightness range. This is the default and recommended formula.

Colors that would fall outside the destination gamut after correction are soft-clipped at constant hue toward the nearest in-gamut saturation, instead of being hard-clipped. This allows stronger curve adjustments without posterization or clipping near the gamut boundary. The soft-clip is applied once after the saturation curve and again after the brilliance curve.

module controls

The module has two tabs, one per curve, a row of shared controls below the curve graph, and a collapsible *guided filter* section.

saturation tab

saturation curve

Plots the output saturation multiplier (*y* axis) against the normalized input saturation (*x* axis). The default curve is a flat line at 50%, a neutral, unity multiplier. Moving a node above the midline increases saturation at that point on the input range; moving it below decreases it.

- click the curve to add a node, drag an existing node to reposition it
- Ctrl+right-click a node to reset it to neutral
- right-click a node to remove it, when more than two nodes are present
- double-click anywhere on the graph to reset the active curve
- scroll over a selected node to nudge its value; Ctrl+scroll uses larger steps

The graph background shows a log-compressed histogram of the image's own input-saturation distribution, so you can see which parts of the range are populated before placing nodes. A gradient strip below the graph shows the corresponding hue-less perceptual gradient for reference.

brilliance tab

brilliance curve

Works identically to the saturation curve, but its output multiplier applies to the perceptual brilliance dimension. Brilliance changes act at constant hue and combine a luminance and chroma shift orthogonal to pure saturation, closer to a local, saturation-aware exposure adjustment.

The same node interactions apply as for the *saturation curve* . Switching tabs does not reset the other curve – both are always active and can be combined. Both curves are looked up at the same input value for a given pixel, whether that is the pixel's raw saturation or, with the guided filter enabled, its smoothed counterpart.

shared controls

color picker

Samples the normalized saturation at the selected point and overlays it on the active curve's graph as a vertical guide, using the standard picker behavior. Use it to locate where a subject, such as skin or sky, sits on the input axis before shaping the curve. Always reads the raw, unfiltered per-pixel saturation, regardless of the guided filter setting.

show saturation mask

Toggles a grayscale preview of the normalized input saturation used to look up both curves, shown in place of the image. Brighter areas correspond to higher input saturation. With the guided filter disabled, this shows the raw per-pixel saturation. With it enabled, it shows the exact coordinate described under *protect near-neutrals* that both curves are looked up with – raw near neutral, fully smoothed above the protected zone. *noise protection* never shifts this coordinate, in the mask or in the actual correction; it only damps the resulting saturation and brilliance factors after the curve lookup, so it has no separate visual signature in this mask. Either way, the mask reflects the module's input, not any correction the curves apply.

saturation formula

Selects the perceptual normalization model, described in general principles . Affects both curves and is not meant to be changed regularly, hence its place in the shared row rather than duplicated per tab.

guided filter

Optional edge-aware smoothing of the saturation value that both curves are looked up with. Only available with the *darktable UCS* saturation formula ; the whole section is disabled when *JzAzBz* is selected.

use guided filter

Enables the smoothing. Disabled by default. Off, each pixel's curve lookup uses its own raw, normalized saturation. On, that value is blended toward a fast scalar guided filter's output before being used as the lookup for both curves, so neighboring pixels with similar saturation are pulled toward a common, locally smoothed value. Reduces halos and stair-stepping at boundaries between saturated and neutral regions, especially with aggressive curves and brilliance changes.

filter radius

Neighborhood radius, in pixels, used for the guided filter's local statistics. Larger values give a more strongly smoothed lookup value with broader spatial influence; smaller values keep the smoothing tighter around local structures.

edge feathering

Controls how strongly the filter follows edges in the saturation signal. Lower values preserve harder edges and keep the smoothed value aligned to the underlying saturation structure; higher values give softer, less edge-anchored transitions.

protect near-neutrals

Center, as a percentage of the normalized saturation range, of a zone near zero saturation that is protected from the guided filter. Pixels at or below the protected side of this zone keep their own raw, unfiltered saturation as the curve lookup value instead of the smoothed one; pixels above it are looked up with the fully smoothed value. Raise this to keep more of the near-neutral range untouched by the smoothing.

protection transition width

Width, as a percentage, of the smooth ramp centered on *protect near-neutrals* over which the lookup value fades from raw to fully smoothed. Wider values give a gentler, more gradual handover instead of a sharp cutoff between the protected and filtered parts of the range.

noise protection

Reduces the strength of the saturation and brilliance corrections – not the curve lookup value itself – for pixels where the guided filter's local residual looks like noise rather than a real color edge, based on a per-pixel confidence estimate computed alongside the filter. Higher values damp the applied factors further toward no change (unity) in those regions. Only has an effect while *use guided filter* is enabled.

Unlike *filter radius*, *edge feathering*, *protect near-neutrals* and *protection transition width*, this control has no visible footprint in the *saturation mask* preview: those four all shift the coordinate the curves are looked up at, which the mask shows directly, while *noise protection* acts after that lookup, on the curves' output. Judge its effect on the image itself rather than on the mask.

Note: *filter radius*, *edge feathering*, *protect near-neutrals*, *protection transition width* and *noise protection* are only active when *use guided filter* is enabled.

usage guide

basic workflow

When first enabled, both curves default to a flat, neutral line and the image is unchanged.

1. use the color picker on a subject you want to target, for example skin or sky, to see where its saturation falls on the graph,
2. briefly enable the saturation mask to confirm which regions of the image correspond to which part of the curve's *x* axis,
3. add a node near the picked position and drag it up to boost, or down to reduce, saturation in that range; the histogram overlay shows how much of the image is affected,
4. add further nodes to protect one end of the range, for example keeping already-vivid colors near the right edge untouched, while adjusting another,
5. switch to the brilliance tab if the corrected colors need a luminance/chroma trim at constant hue to look balanced, which is often useful after a strong saturation boost,
6. if strong curve settings produce halos or stair-stepping around saturated edges, enable the guided filter and re-check the mask preview – which reflects filter radius, edge feathering, protect near-neutrals and protection transition width directly – then separately judge noise protection against the image itself, since it has no footprint in the mask.

choosing a formula

Leave the formula on *darktable UCS* unless matching the behavior of an older edit or style built with *JzAzBz*. Switching formulas changes how input saturation is normalized and therefore changes the effect of existing curve nodes. Switching to *JzAzBz* also disables the *guided filter* section entirely; its settings are kept but have no effect until you switch back to *darktable UCS*.

Note: You should re-check the curves against the histogram overlay after switching formulas.

using the guided filter

The guided filter is only available with the *darktable UCS* formula. It does not change what the curves do; it mainly changes which saturation value they are evaluated at for each pixel. Start with the default radius and increase it if halos persist across large color transitions; increase edge feathering only if transitions still look too abrupt afterward. Because the smoothing sits upstream of both curves, it affects the saturation and brilliance results together; there is no separate guided-filter control per curve. If near-neutral areas such as skies or skin start looking patchy or mottled once the guided filter is on, raise *protect near-neutrals* (and soften the handover with *protection transition width*) to keep them on their own raw saturation instead. If strong curve settings amplify visible noise in otherwise flat, low-detail regions, raise *noise protection* to damp the correction there; judge this against the corrected image, not the saturation mask, since this control has no coordinate to show there.

FAQ

why does the curve reference a normalized value instead of raw chroma?

Raw chroma or saturation values are not bounded consistently across hues and lightness levels – the same numeric chroma can be barely visible for one hue and heavily clipped for another. Normalizing input to the local gamut boundary, from 0 to 1, for the current hue and lightness makes the curve behave consistently across the image: a node placed at 80% always means 80% of the way to the most saturated color physically displayable at that hue and lightness.

how is this different from color balance RGB's perceptual saturation grading?

Color balance RGB applies a single perceptual saturation and brilliance coefficient, optionally split between shadows, mid-tones and highlights via luminance masks. *Saturation curve* instead lets the correction vary continuously as a function of the pixel's own saturation, independent of its luminance. The two modules are complementary and can be used together: luminance-based masking in *color balance RGB*, saturation-based shaping here.

does the guided filter blend the corrected image with the original?

No. It does not mix the untouched and corrected pixels after the fact. It only smooths the saturation value fed into the saturation and brilliance curves for each pixel, before any correction is computed. A pixel is always fully processed through the curves; what changes is which point on the curve it is evaluated at.

internal processing

The following is the internal order of operations within the module, for each active pixel:

1. convert from pipeline RGB to XYZ, then to the selected perceptual space, JzAzBz or darktable UCS JCH,
2. compute the hue angle and the saturation normalized to the local gamut boundary,
3. if the guided filter is enabled (darktable UCS only), computed on CPU or, where available, GPU via OpenCL: run the guided filter over a full-image scalar mask to get a smoothed value, estimate a per-pixel noise confidence from the local residual and color-edge energy, and blend the raw value toward the smoothed one using the *protect near-neutrals*/*protection transition width* zone as the blend weight; otherwise use the value from step 2 directly,
4. look up the saturation and brilliance curves at that (possibly blended) value to obtain two correction factors,
5. if the guided filter is enabled, damp both factors toward unity in proportion to *noise protection* times the pixel's noise confidence,
6. apply the saturation factor at constant hue, soft-clipping against the gamut boundary,
7. apply the brilliance factor at constant hue, orthogonal to saturation,
8. soft-clip the result again against the destination gamut,
9. convert back to pipeline RGB.

caveats

Changing the working RGB space mid-session will change how existing curve nodes behave, since 80% of the gamut boundary under one profile is not the same absolute saturation as under another. If you change the working profile, re-check the curves against the histogram overlay.

The saturation mask preview shows the module's input to the curves, raw or guided-filter-smoothed, not the corrected output. It is a diagnostic aid for shaping the curve, not a preview of the final result.

Note: enabling the guided filter adds computation cost that scales with filter radius. Very large radii will noticeably increase processing time on high-resolution images.

The guided filter, and its *protect near-neutrals*, *protection transition width* and *noise protection* controls, only take effect with the *darktable UCS* saturation formula; they are inert under *JzAzBz*.

8.2.60. scale pixels

Technical information

description
module for setting pixel aspect ratio useful for certain sensor types and anamorphic desqueeze.
purpose
corrective.
input
linear, RGB, scene-referred.
processing
linear, RGB.
output
linear, RGB, scene-referred.

Some cameras (such as the Nikon D1X) have rectangular instead of the usual square sensor cells. Without correction this would lead to distorted images. This module applies the required scaling.

darktable detects images that need correction using their Exif data and automatically activates this module where required.

For other images, the module can also be used to de-squeeze the image when using anamorphic lenses.

module controls

pixel aspect ratio
Aspect ratio of the pixels. Values greater than 1 stretch the image horizontally, values less than 1 stretch the image vertically.

8.2.61. shadows and highlights

Technical information

description
modify the tonal range of the shadows and highlights of an image by enhancing local contrast.
purpose
corrective and creative.
input
linear or non-linear, Lab, display-referred.
processing
non-linear, Lab.

output
non-linear, Lab, display-referred.

Modify the tonal range of the shadows and highlights of an image by enhancing local contrast.

Note: *This module performs blurs in Lab color space, which can result in a number of issues when the parameters are pushed hard, including halos, high local contrast in highlights, and hue shifts towards blue in the shadows. You are advised to use the tone equalizer module instead.*

---

module controls

shadows
Control the effect on shadows. Positive values will lighten shadows while negative values will darken them.

highlights
Control the effect on highlights. Positive values will lighten highlights while negative values will darken them.

white point adjustment
By default the algorithm of this module leaves the black and white points untouched. In some cases an image might contain tonal variations beyond the white point, i.e. above a luminance value of 100. A negative shift in the white point adjustment can bring these values back into the proper range so that further details in the highlights become visible.

soften with
The underlying blurring filter: gaussian or bilateral. Try the bilateral filter if the gaussian blur generates halos.

radius
The radius of the blurring filter used by the algorithm. Higher values give softer transitions between shadows and highlights but might introduce halos. Lower values will reduce the size of halos but may lead to an artificial look. The bilateral filter is much less prone to halo artifacts.

compress
Control how strongly the effect extends to the mid-tones. High values limit the effect to only the extreme shadows and highlights. Lower values also cause adjustments to the mid-tones. At 100% this module has no visible effect as only absolute black and absolute white are affected.

shadows color adjustment
Control the color saturation adjustment made to shadows. High values cause saturation enhancements on lightened shadows. Low values cause desaturation on lightened shadows. It is normally safe to leave this at its default of 100%. This gives a natural saturation boost on shadows – similar to what you would expect in nature if shadows were to receive more light.

highlights color adjustment
Control the color saturation adjustment made to highlights. High values cause saturation enhancements on darkened highlights. Low values cause desaturation on darkened highlights. Often highlights do not contain enough color information to give convincing colors when darkened. You might need to adjust this parameter in order to find the best fitting value depending on your specific image, but be aware that sometimes the results might not be fully satisfying.

8.2.62. sharpen

Technical information

description
sharpen the details in the image using a standard UnSharp Mask (USM).

purpose
corrective.

input
linear or non-linear, Lab, display or scene-referred.

processing
frequential, Lab.

output
quasi-linear, Lab, display or scene-referred.

Sharpen the details in the image using a standard UnSharp Mask (USM).

This module works by increasing the contrast around edges and thereby enhancing the *impression* of sharpness of an image. This module is applied to the L channel in Lab color space.

Note: *The USM algorithm used in this module performs blurs in Lab color space, which can produce undesirable effects, and is no longer recommended. Instead use the presets offered by the contrast equalizer module for deblurring or the local contrast module for general sharpness.*

---

module controls

radius
The unsharp mask applies a gaussian blur to the image as part of its algorithm. This parameter controls the radius of that blur which, in turn, defines the spatial extent of the edge enhancement. Very high values will lead to ugly over-sharpening.

amount
The strength of the sharpening.

threshold
Contrast differences below this threshold are excluded from sharpening. Use this to avoid amplification of noise.

8.2.63. sigmoid

Technical information

description
apply a view transform to make an image displayable on a screen or print using a robust and smooth tone curve with optional color preservation methods.

purpose
corrective and creative.

input
linear, RGB, scene-referred.

processing
non-linear, RGB.

output
linear, RGB, display-referred.

Remap the tonal range of an image using a modified generalized log-logistic curve.

This module can be used to expand or contract the dynamic range of the scene to fit the dynamic range of the display.

Note: Modules placed before sigmoid in the pipeline operate in scene-referred space. Modules after sigmoid work in display-referred space.

You automatically enable this module in new images by setting the value of preferences > processing > auto-apply pixel workflow defaults to ‘scene-referred (sigmoid)’ (this is the default setting for new installations).

usage

Please take note of the following guidelines while using this module within your workflow:

only use one tone mapper
sigmoid is a tone mapper ; you typically have exactly one tone mapper enabled for an image.

adjust for the mid-tones first

The sigmoid curve pivots around middle gray. Before using sigmoid, you should first use the *exposure* module to adjust the mid-tones to your liking.

less is more

You should try to accomplish the majority of your processing using modules in the scene-referred section of the pixelpipe and not rely on the display transform module (*sigmoid*, *filmic rgb*, *base curve*) to do all the work.

preserve hue to taste

This module provides a number of methods to preserve hues. You are advised to use the “per channel” mode and tune the hue preservation to your liking on a per-image basis. Sunsets and fire are two examples where users commonly reduce the hue preservation to achieve a “hotter” look.

keep skew at bay

For images like portraits, it is best to refrain from using skew values above zero. This way you will avoid harsh transitions in the skin tones. This is particularly important when hue preservation is not enabled, as the skew also affects the hue path of the skin tones.

module controls

contrast

Adjust the aggressiveness of the compression while leaving middle-gray unchanged. Higher values require lower exposure to reach display white, and shadows become darker. Lower contrast is able to display a larger dynamic range.

skew

Lean the compression towards shadows or highlights. Skew can be used to transfer some contrast from shadows to highlights or vice versa without changing the amount of contrast at middle gray. Positive skew flattens shadows and compresses highlights. Negative skew creates darker shadows and duller highlights.

color processing

The mode used to map pixel values from scene to display space.

- *per channel* mode applies the sigmoid curve to each rgb channel separately, affecting luminance, chroma, and hue. Hue can be optionally preserved using the *preserve hue* option (below). This mode is in line with the behavior of the color layers in analog film, and handles smooth roll-off to bright areas very well.
- *rgb ratio* is similar to *preserve color* in filmic rgb . It maps the rgb triplet uniformly using the sigmoid curve, which preserves the spectral color of the pixel. Bright colorful pixels are desaturated along spectral lines as they would otherwise end up outside the display gamut.

preserve hue *(per channel mode only)*

Choose how much to preserve hue – 100% preserves the spectral hue of the image (identical to using the “rgb ratio” color processing mode); 0% uses the per-channel mode with heavy hue skewing (see below). An acceptable approximation of preserved perceptual hue is usually somewhere between the two extremes.

All colors that lie between the primary colors (red, green, and blue) converge towards the closest secondary colors (yellow, magenta, and cyan). The *per channel hue skew* effect creates yellow sunsets and fires, magenta-looking blue lights, and cyan skies. The skew is stronger for brighter and more saturated pixels.

target black

Lower bound that the sigmoid curve converges to as the scene value approaches zero – this should normally be left unchanged. You *can* use this to give a faded analog look, but should instead prefer to use the “global offset” slider in *color balance rgb* to achieve a similar effect.

target white

Upper bound that the sigmoid curve converges to as the scene value approaches infinity – this should normally be left unchanged. You can use this to clip white at a defined scene intensity.

For HDR output (an output profile with a PQ or HLG transfer function), target white is multiplied by the headroom of the output – its peak divided by the 203 nit reference white – so that the highlights extend to the peak of the export. HLG peaks at 1000 nits, and PQ at the *PQ peak* set in the *export* module (default 1000 nits). The result is capped at 1600% (3248 nits). SDR output is not affected.

follow output peak

This checkbox is only shown for edits made before darktable supported HDR output, and only while it is unticked. Such edits keep *target white* as set, so they render exactly as before, also when exported as PQ or HLG. Tick it to scale *target white* to the peak of an HDR output, as described above. SDR output does not change. New instances of the module always follow the output peak.

primaries

Expand this section to set custom primaries. You are advised to use the “smooth” preset as a starting point and then adjust using the following controls:

Note: These settings apply only to the *per channel* processing mode.

---

base primaries
Choose the set of primaries to use as the base for adjustments. This is a bit like locally overriding the working profile, and is necessary to allow presets to be created that don’t change even if the user amends the working profile used in the pixel pipeline.

red/green/blue attenuation
Attenuate (decrease) the purity of the red, green or blue primaries before the signal is processed through *sigmoid*’s per-channel curves. An important consequence is that now even the brightest and most pure inputs get smoothly degraded to achromatic at the high end. This avoids posterization and flat-looking patches, which are often seen with, for example, blue LED lights.

red/green/blue rotation
Rotate the primaries where the per-channel curves are applied. This affects the hue paths when approaching white in the high end. These controls should not normally need large adjustments from the starting values given in the “smooth” preset.

recover purity
Recover some of the original purity. A value of 100 causes all of the attenuations to be restored after the per-channel process is done. This lands the middle range values near their original purities. A value of 0 doesn’t restore the purity at all, so the more you apply attenuation, the less purity there is in the final image. The rotations are always restored regardless of the value of this slider. When this slider is at 0, the output of the module is guaranteed to remain within the gamut footprint of the chosen base primaries.

Bear in mind that unlike the *rgb primaries* module, this is not a tool for creative color grading but rather a set of controls to provide a reasonable starting point for further edits. The effect of these adjustments is not the same as the rgb primaries module even though the interface looks similar.

8.2.64. soften

Technical information

description
create a softened image using the Orton effect.
purpose
creative.
input
linear, RGB, display-referred.
processing
linear, RGB.
output
linear, RGB, display-referred.

Create a softened image using the Orton effect .

Michael Orton achieved his results on slide film by using two exposures of the same scene, one well exposed and one overexposed. He then used a darkroom technique to blend these two images into a final image where the overexposed image was blurred.

This module is a near-copy of Orton’s analog process into the digital domain.

module controls

size
The size of blur of the overexposed image. The bigger the size, the softer the result.
saturation
The saturation of the overexposed image.
brightness
The brightness (EV) of the overexposed image.
mix
How the two images are mixed. 50% represents an equal mix of the well exposed and overexposed images.

8.2.65. spektrafilm

Technical information

description
simulates the physical process of developing and printing analog film, using spectral emulsion and paper data from the spektrafilm project.
purpose
creative.
input
linear, RGB, scene-referred.
processing
non-linear, RGB.
output
non-linear, RGB, display-referred.

Spektrafilm aims to recreate the look of an analog film stock printed onto darkroom paper for digital images. It is not intended to replace *negadoctor* , but rather provides a software approach to emulate the process of developing film in the darkroom

Most film emulations are a color recipe: someone photographed a scene on film, measured how the colors moved, and saved that as a curve or a LUT. This module works the other way round. It starts from laboratory measurements of what real film and real paper are made of – how sensitive each color layer is to each wavelength of light and how much dye each one produces – and simulates what physically happens to a photograph, step by step:

- light from the scene falls on the film and exposes its three-color layers,
- development turns that exposure into dye, with the layers chemically interfering with one another as they do in a real tank,
- an enlarger shines light through the developed negative onto a sheet of paper,
- the paper develops in turn, and the finished print is scanned.

Grain, halation and diffusion filters are simulated as things that physically happen along the way.

Because the module simulates the physical properties of light and its interaction with film exposure and the printing process, the same film and paper render differently under different lighting. Tungsten light and midday sun will not simply differ in white balance.

The measurement data comes from the spektrafilm project by Andrea Volpato.

Note: *spektrafilm* is a tone mapper and takes the place of *filmic rgb* , *sigmoid* or *AgX* . To use it for every new edit, set preferences > processing > auto-apply pixel workflow defaults to *scene-referred (spektrafilm)*.

Like other view transforms, modules placed before *spektrafilm* in the pipeline operate in scene-referred space. Modules after it work in display-referred space.

---

usage and FAQ

install the data first

The module needs a data pack – the film and paper measurements it works from – and cannot render anything without one. If none is installed, the module shows a single button that downloads it, and the rest of the controls appear once that finishes. Packs are checked against a checksum before being installed and are stored in darktable's data folder rather than the cache, so they survive clearing the cache (see *where data packs are stored* below).

edits stay on the data pack they were made with

Each data pack has its own identity, and an edit records the pack it was first rendered with. Installing a newer pack does not change how existing edits look: new packs are installed beside the old ones, and only new edits (or an edit where you choose a different *spectral upsampling* method) use them. Edits made before packs had an identity are treated as made with the first published pack (spektrafilm 0.3.3).

If the pack an edit was made with is not installed, the module still renders it with the closest installed pack, shows a warning naming the pack it wants, and offers a button to download that pack.

where data packs are stored

Downloaded packs live in `darktable/spektrafilm/packs` inside your user data folder – `~/.local/share/darktable/spektrafilm/packs` on Linux and macOS, `%LOCALAPPDATA%\darktable\spektrafilm\packs` on Windows. Each pack has its own subfolder, so several packs can be installed at once. The subfolder is named after the pack's identity, an 8-digit hexadecimal hash. The first published pack (spektrafilm 0.3.3) has no identity of its own, so its subfolder is named after its spectral upsampling table instead (`565f4ec4`, the *hanatos2025* table).

A pack can also be installed by hand by placing its files directly in `darktable/spektrafilm` (one level above `packs`). The module never overwrites that folder, and a pack placed there takes precedence over downloaded ones. This is the way to use a pack that is not published for download, such as a testing pack.

The module itself does not list installed packs; the *spectral upsampling* list shows the methods they provide.

updating the data

New data packs are never downloaded automatically. Use *check for data pack updates* at the bottom of the *advanced* section of the *film* tab to see whether a newer pack is published, and download it from the same button.

only use one tone mapper

spektrafilm is a tone mapper ; you typically have exactly one tone mapper enabled for an image.

start simple

Choosing a film stock is enough. Everything else already carries a sensible value taken from that film's own measurements, and the paper follows the film automatically. The tabs are for fine-tuning.

auto print exposure changes what film exposure does

Out of the box, *film exposure* behaves like leaving the enlarger on for a fixed time: expose the film more and the print comes out brighter. When *auto print exposure* is on, the module sets the print exposure for you, changing it by exactly the amount needed to cancel out the brightness change that *film exposure* would otherwise cause. This is what a real printer does, aiming for a consistent print density however the negative was exposed. *film exposure* then changes only color, contrast and grain, by moving the scene to a different part of the film's response.

some print stocks need manual print exposure

A few print stocks are not papers but print films made for projection – Kodak 2302, 2383 and 2393 – with a far greater density range than any paper. *auto print exposure* can misjudge these stocks. If a print looks implausibly dark or bright with it on, correct it with *print exposure compensation*, or leave auto off for those stocks.

why doesn't the tone range hit 100% white?

Physical characteristics of the film simulation: Highlights sit on the shoulder of the film and print response curves, where additional exposure produces progressively less additional density. This means the brightest tones compress and lose separation as they approach white rather than clipping cleanly to it.

How to resolve: Either use *pre-compression boost* or *post-compression scale* in the *scanner* tab, place the white point with the *black and white point* controls in the *scanner* tab, disable *gamut compression* in the *film* tab (under advanced) or use an instance of *tone curve* or *levels* set after *spektrafilm*.

why doesn't the tone range hit 100% black?

Shadows sit where the density curve flattens out towards its maximum, so no more density can build up past a certain point and the darkest tones never quite reach black. If *viewing glare* in the *scanner* tab is raised, the room light reflected off the print lifts them further.

How to resolve: Experiment with different print stocks – the print films *Kodak 2383* and *Kodak 2393* reach far deeper blacks than any of the papers – place the black point with the *black and white point* controls in the *scanner* tab, or use an instance of *tone curve* or *levels* set after *spektrafilm*.

how to set up a clean digital look?

Disable *grain* and *halation*. This will reveal fairly harsh local contrasts/sharpening due to the *DIR couplers*. Control these with *same-layer inhibiton* or *inhibitor spread*.

how to control contrast?

Either use modules in darktable's pipeline before/after spektrafilm or use spektrafilm's various built in options to modulate contrast. E.g. by using push/pull (work with *film exposure* and *push/pull* in the *film* tab) or with *print gamma* in the *print* tab. Other options include *diffusion* in *film* or *print* tab, *preflash* and the *black and white point* controls in the *scanner* tab.

The scientifically inclined readers can find more detailed information in the Spektrafilm OFX plug-in reference guide .

module controls

Film stock, print paper and film format are always visible at the top of the module. Everything else is grouped into tabs, one per stage of the physical process: *film*, *print*, *grain*, *halation*, and *scanner*.

header

film stock

The film to simulate.

print paper

The paper to print onto. Left on *auto* it follows the film stock's own intended paper, and names which one that currently is – for example *auto (Kodak Portra Endura)*. A black & white film that names no paper gets a black & white one. Choose a paper explicitly and it stays put when you change film.

Choose *none (scan the film)* to look at the developed negative or slide directly instead of printing it. This is selected by itself for slide and reversal stocks, and you can switch back to a paper afterward.

format

A preset picker for common frame sizes (half-frame, 35mm, 6x6, 6x7, 6x9, 4x5, 8x10, Super 8, 16mm, Super 16, Super 35, VistaVision, 65mm 5-perf, IMAX 15-perf, or custom), which sets *frame long edge* below. The *frame long edge* slider is only shown when *custom* is selected. The preset names a film *gauge* (35mm) while the slider gives the frame's long edge (36mm); both describe the same format.

frame long edge

The real-world width of the simulated frame, in mm. Only available when *format* is set to *custom*. This is the physical scale everything else is measured against, so grain, scatter, halation, diffusion and the DIR coupler spread all come out proportionally larger on a smaller format at the same print size.

film

exposure

film exposure

Exposure adjustment at the film stage, in EV. Unlike the *exposure* module, which only scales pixel values, this changes how much light reaches the simulated film. Besides brightness, it moves the scene to a different part of the film's response, so contrast, color and grain change too.

Note: With *auto print exposure* on, *film exposure* no longer changes overall brightness (see usage above), but it still changes color and grain, because it moves the scene to a different part of the film's response.

push/pull

Shooting the film at a speed other than its box speed (the ISO rating printed on the film's packaging), then compensating in development, in stops. Positive values push (shoot darker, develop longer), negative values pull. Development is adjusted automatically: each stop of push lowers the exposure by one stop and raises development contrast by about 15%, and pull does the opposite. This comes on top of *development gamma* below, which does not move. This is an approximation, since the real result depends on the specific developer, which is not simulated.

chemistry

development time

How long the film is developed, in minutes. Of the film stocks, only Kodak Double-X has measurements for more than one development time (4, 5, 6.5, 9, and 12 minutes), so the slider is grayed out for every other film. Values snap to the times actually measured, and 0 means the stock's standard development.

development gamma
Overall development contrast, as though you had developed for longer or shorter. 1.0 is normal.

fast layer gamma
Contrast of the most light-sensitive layer alone. Real film has several layers of differing sensitivity, and a change in development does not affect them equally.

slow layer gamma
Contrast of the less sensitive layers.

developer exhaustion
In a real developing tank the developer is locally depleted where the film was most heavily exposed. Those areas then stop gaining density, however much brighter the scene gets, so highlights run into a ceiling instead of climbing indefinitely – and the approach to that ceiling steepens, so highlight contrast tends to rise rather than soften. Mid-gray is held still, so only the bright end moves. The simulation models this as a change to the film's response curve rather than as a spatial effect. 0 switches it off.

DIR couplers

As one color layer develops, it releases a chemical that slows development in its neighbors. This is where the saturation of color negative film comes from, and it sharpens edges too: a bright area holds back development just beyond its own boundary, which reads as extra definition.

DIR couplers
Overall strength of the effect. 1.0 matches the real film, 0 switches it off. The slider stops at 1.0 because the simulation has to be able to work backwards from the film's measured curves, and beyond film-accurate strength that becomes impossible – for some stocks well before 2.0. The module reduces the amount further for stocks where it would break down sooner.

same-layer inhibition
How much each layer holds back its own development. Raising it flattens contrast within each color channel. Can be used for tuning local contrast and sharpening independently of the interlayer effect below.

interlayer inhibition
How much each layer holds back the other two. This is the part that produces the saturation the effect is known for; at 0 the three colors develop independently.

inhibitor spread
How far the released chemical spreads through the emulsion, in thousandths of a millimeter. The couplers are more of an edge effect rather than a purely tonal one: the further it spreads, the further a bright area reaches past its own boundary, and the more the result looks like added sharpening. Shorten it for a gentler, more purely tonal response. Each film carries its own value, which is what the slider returns to on reset; changing film keeps the value you set.

spread tail
A small part of the chemical travels much further than the rest. This is what gives large bright areas a broad, gentle falloff rather than a glow that stops abruptly.

tail weight
How much of the chemical travels in that long tail. 0 removes the tail entirely.

diffusion

Simulates a diffusion filter – the physical piece of glass placed in the optical path to bloom highlights and soften contrast. Two diffusion stages are modeled at different points in the analog film process: one in front of the camera lens during exposure (this one in the film tab), and another in the enlarger when the developed negative is projected onto photographic paper (in the print tab).

enable diffusion filter
Switch on the camera-stage filter.

diffusion filter type
Which filter to imitate:

- *black pro-mist*: a concentrated, punchy halo that keeps blacks deep.
- *glimmerglass*: tight and subtle, preserving sharpness.
- *pro-mist*: broader and more pastel, a softer atmospheric look.
- *cinebloom*: a wide, slowly fading veil across the frame.

diffusion strength

How much light is diverted into the halo. 0 switches it off. The diverted light is taken from the direct picture and spread into the halo, so raising this lifts shadows and lowers contrast as well as making highlights glow.

diffusion size

How far the halo spreads – the same light carried further. Use *diffusion strength* for more.

diffusion halo warmth

Warms (positive) or cools (negative) the outer halo, on top of whatever bias the chosen filter type already has.

advanced

spectral upsampling

How a pixel's color is turned into the spectrum of light the film is exposed to. A digital image only records three colors, so the spectrum has to be reconstructed before the film's measured sensitivities can be applied to it. Many different spectra produce the same three values and each method picks a different one, so switching changes the render even with the same film.

Methods come in two kinds:

- irradiance methods, such as *hanatos2025*, reconstruct the light reaching the film directly.
- reflectance methods reconstruct the color of the surface instead, as recovered under the scene's light, and relight it with the film's reference illuminant.

The list shows the methods of every installed data pack. *hanatos2025* is the original one, and newer packs can add more. Grayed out when only one method is installed. Choosing a method from a newer pack moves the edit onto that pack.

bandwidth adaptation

Trims the extreme violet and deep red ends of the film's sensitivity as part of how each stock is characterized. On by default and best left on as it belongs to the film's description. Only applies to irradiance methods such as *hanatos2025* and is grayed out for reflectance methods.

surface adaptation

An optional per-color exposure correction carried in the film data, worth up to two stops for strongly colored light and nothing at all for neutral. Off by default, because it shifts saturated colors noticeably and the reference implementation does not apply it either. Stocks whose data does not include the correction are unaffected either way. Like *bandwidth adaptation*, it is grayed out for methods it does not apply to.

gamut compression

Keeps colors inside the working color profile's gamut. When enabled, colors from the simulated film and print are pulled towards the gamut along two smooth OkLCh curves: one for chroma, which already eases strongly saturated colors before they reach the gamut boundary, and one for lightness, which rolls the highlights off to white from the upper midtones up. When disabled, out-of-gamut colors pass through unchanged and are hard-clipped later in the pipeline instead, which can shift hues and merge distinct tones, and the highlights no longer roll off. This is useful for spotting which colors the simulation is pushing out of gamut.

check for data pack updates

Asks the data repository whether a newer data pack is published. Nothing is downloaded until you press the button again, which is now labeled *download data pack* followed by the pack's version. A newer pack is installed beside the ones you have and does not change existing edits (see usage above).

print

exposure

print exposure compensation

How long the enlarger stays on, in EV. More exposure prints denser, so positive values darken the print and negative values lighten it. It always acts as an offset: with *auto print exposure* on, it shifts the automatic result.

auto print exposure

Compensate print exposure automatically for changes in film exposure, the way a real printer aims for consistent density whatever the negative. Does nothing while *print paper* is set to *none (scan the film)*.

print gamma

Contrast of the print, achieved by reshaping the paper's own response.

Slide and reversal stocks are viewed directly rather than printed. *print paper* switches to *none (scan the film)* by itself when you choose one, and the entire print tab then has no effect, along with *viewing glare* on the scanner tab as there is no print surface.

chemistry

development time

How long the print is developed. Works exactly like the film tab's version and is set separately from it. Only Kodak Print Film 2302 was measured at more than one time (2, 3.5, 5, 7 and 9 minutes), so the slider is grayed out for every other paper – and while *print paper* is set to *none (scan the film)*, since there is no print to develop. When the paper changes, the slider moves to that paper's standard time.

print gamma red / print gamma green / print gamma blue

Contrast of each of the paper's three dye layers alone, on top of the overall *print gamma*. 1.0 leaves the layer unchanged. Splitting contrast per channel corrects a negative whose layers developed to different contrasts (crossover). Filtration cannot fix that, because it shifts every tone the same way, while crossover pushes shadows towards one color and highlights towards another.

filtration

filtration M / filtration Y

Magenta and yellow filters in the enlarger, in Kodak CC units away from neutral – the color balance of the print, adjusted the way it is in a darkroom.

preflash

preflash exposure

A brief, even pre-exposure of the paper before the main one, through the blank part of the film. It adds density where the main exposure is weakest, so it tones the print's highlights and lowers contrast. Use it to bring detail into highlights that would otherwise print as blank paper, or to tame a contrasty negative.

preflash M filter shift / preflash Y filter shift

Magenta and yellow filtration for the preflash alone, in Kodak CC units, independent of the main filtration above.

print diffusion

enable print diffusion

A second, independent filter at the enlarger rather than the camera, diffusing the light at the print exposure rather than the film exposure. Its controls work like the ones above and are set separately: *print diffusion filter type*, *print diffusion strength*, *print diffusion size*, and *print diffusion halo warmth*.

Because it acts on the light passing through the negative, its effect on the print is reversed: it spreads the print's dark areas into its lighter ones instead of making highlights glow, and *print diffusion halo warmth* prints in the complementary color, so positive values give a cyan halo and negative values a yellow one.

grain

Film grain comes from the silver crystals the picture is made of. The module simulates such an emulsion, blurs detail and grain together the way a real one does, and then restores the lost edge definition with the sharpening under *texture* below. The blur and the recovery are tuned as a pair, so a picture with grain on is slightly softer than the same picture with grain off. That is how film behaves. If you want it sharper, lower *grain blur* or switch grain off; *grain strength* scales only the grain, not the blur.

The controls are split in two. *emulsion* changes what the film is made of and rebuilds the crystals. *texture* changes only how those crystals are drawn at your output size.

enable grain

Switch grain simulation on.

emulsion

granularity

Controls the size of the film crystals relative to the measured value for this film stock. 1.0 corresponds to the stock's default crystal size as specified in its datasheet.

Raising the value grows the crystals, which means there are fewer of them and each one counts for more, so the grain becomes coarser *and* stronger together – the same way a faster film differs from a slower one.

grain strength

How far the grain is allowed to move each pixel. 1.0 matches the real stock, 0 switches it off.

Unlike *granularity*, this scales the result: a negative's grain gets amplified a second time by the print stage, while a slide is scanned directly with nothing to amplify it. The same value therefore reads much weaker on slide film; to strengthen grain there, raise *granularity* instead. Right-click for values up to 8.

uniformity

Controls how evenly the film crystals are distributed, relative to the stock's measured value. Higher values concentrate grain toward the mid-tones, causing it to taper off in the densest areas. Lower values let grain keep growing all the way into the densest areas.

sublayer particle scale

A real emulsion layers coarse crystals over finer ones. This scales the finer sub-layers against the coarsest. The stock's overall granularity is held, so lower values make the fine layers finer and the coarse layer coarser, letting it dominate; at 0 only the coarsest layer carries grain. No effect on stocks measured as a single layer.

density floor

Scales the density each crystal sits at even where the film received no light – the reason grain does not disappear entirely in clear areas. 1.0 is the stock's own measured floor, 2.0 twice it. The floor is measured per channel, so the balance between channels is kept.

texture

grain blur

How much the grained film is blurred, as a scale on a reference radius of 0.89 pixels. This softens grain and the image detail under it alike: the crystals are unchanged, so raising it makes grain smoother and less distinct, not coarser. Use *granularity* above for that.

The value that looks right depends on output size since it is measured in pixels rather than on the film. Raise it for large prints where one crystal covers several pixels; lower it for small output.

dye cloud size

Each developed crystal leaves a small cloud of dye rather than a hard dot. This scales how far that cloud spreads, measured on the film itself, so it only becomes visible at magnifications where a single crystal covers more than a pixel.

recovery radius

How wide the recovery sharpening reaches, in pixels. 0 switches it off. A broad radius puts the crispness back into mid-sized detail. This stops the pixel grid itself from showing through.

recovery strength

How strongly it sharpens, restoring the definition the grain blur took away. It moves density around without adding any, so it cannot invent detail and is not a noise reduction. 0 leaves the softening in place with nothing countering it; pushing well beyond the default sharpens more than was ever lost, which makes grain look crunchy rather than photographic.

halation

Two separate things happen to bright light inside film. Some of it scatters sideways within the emulsion as it passes through. Some goes all the way through, reflects off the film base behind, comes back, and puts a reddish glow around very bright highlights. Both are simulated and adjusted independently.

enable halation and scatter

Switch both effects on.

scatter

scatter amount

How much light scatters sideways within the emulsion. 1.0 matches the film, 0 switches it off.

scatter size

How far the light scatters. 1.0 matches the film and is the value the simulation normally works at. Above that, the whole frame softens quickly since the radius grows directly with the value. Drags to 1.5, right-click for more.

halation

halation strength
How strong the reflected glow is. 1.0 matches the chosen film, which means it looks very different from stock to stock: most modern film has a layer specifically to absorb this light and shows little, while consumer stocks such as *Kodak Gold 200* or *Kodak UltraMax 400* glow about five times as strongly as *Kodak Portra*, *Kodak Ektar* or *Vision3*. Drags to 2, right-click for up to 8.

halation size
How far the glow spreads. 1.0 matches the film.

highlight boost

The highlight boost is not part of halation and is not toggled by *enable halation and scatter*. It acts on the film exposure before any light spreads, so it also feeds the film's *diffusion* filter. It sits on this tab because its visible result is the glow scatter, halation and diffusion produce from it.

highlight boost
Brightens everything above *boost protect* so that highlights, including ones clipped in the original file, glow into the scatter, halation and diffusion effects. It does not detect clipping as such. In EV; 0 switches it off. The boost applies over a fixed range, so it gives the same result regardless of image size, zoom level, or how the export was processed.

boost range
How sharply the boost rises above the *boost protect* threshold. Lower keeps it to the very brightest areas; higher spreads it more evenly over everything above the threshold. Nothing below the threshold is affected either way.

boost protect
Keeps everything below this many stops above mid-gray out of the boost entirely.

scanner

How the finished print – or the film itself, when *print paper* is set to *none (scan the film)* – is digitized, and the conditions it is looked at under. These act on the final image, after everything else.

output

pre-compression boost
Brightens the image with a focus on the midtones to highlights. Use it to lift midtones into the roll-off and use *post-compression scale* to define where highlights and white lands. This acts at the end of the module, so the picker measures the processed image.

post-compression scale
Scales the finished picture after the gamut compressor. This only changes the level, leaving the relationship between the channels alone, similar to what a tone curve does by moving its white point.

sharpness

scanner blur
Softening from the scanner's own optics, in pixels. 0 switches it off.

scanner sharpen radius
How wide the scanner's sharpening reaches, in pixels.

scanner sharpen strength
How strongly it sharpens. 0 switches it off – worth doing if you would rather sharpen later with darktable's *sharpen* or *diffuse or sharpen* modules.

glare

viewing glare
A faint veil of room light reflecting off the print surface, as a percentage. Lifts the deepest blacks very slightly, the way a real print in a real room never quite reaches black. Does nothing while *print paper* is set to *none (scan the film)*, since there is no print surface.

black and white point

Sets where the darkest and lightest tones of the image end up. Two cases need this, because their range is set by the film rather than by the medium being viewed:

- a scan of slide or other positive film: the film always carries some base density and never reaches its full maximum, so the scan looks washed out until its ends are set. Both corrections switch on by themselves when you choose a slide or reversal stock.
- a print on negative paper: the paper's range is set by the negative behind it, so the print may not reach full black or white. Both corrections are off by default here, as in the reference implementation.

The controls are grayed out for a scan of negative film and for a print on positive paper, which carry their own black and white. Mid-gray is held in place by adjusting the exposure, so *film exposure*, *print exposure compensation* and *print gamma* keep working as usual with the corrections on.

black point correction
Set the darkest tone of the image with *black level*. Off, the shadows stay where the film puts them, which is often short of black.

black level
Lifts or crushes the shadows. 0 is true black; higher values lift the blacks and soften the image, and below 0 the deepest shadows are crushed together.

white point correction
Set the lightest tone of the image with *white level*. Off, the highlights stay where the film puts them.

white level
Brightness of the lightest tone. 1 is full white. Because mid-gray is held in place, lowering it also darkens the shadows a little.

With only one correction switched on, the other end stays where the film puts it.

8.2.66. split-toning

Technical information

description
use two specific colors for shadows and highlights and create a linear toning effect between them up to a pivot.

purpose
creative.

input
linear, RGB, scene-referred.

processing
linear, RGB.

output
linear, RGB, scene-referred.

Create a two color linear toning effect where the shadows and highlights are represented by two different colors.

The split-toning module does not convert images to black and white and has limited benefits on color images. In order to perform traditional split-toning, the input to this module should therefore be monochrome.

module controls

shadows and highlights color
Set the desired hue and saturation for both shadows and highlights. Clicking on the colored squares will open a color selector dialog which offers you a choice of commonly used colors, or allows you to define a color in RGB color space.

balance
The ratio of toning between shadows and highlights. When set to 50%, half of the lightness range in image is used for shadows toning and the other half for highlights toning.

compress
The percentage of total (mid-tone) lightness range that is not affected by color toning. This compresses the effect on the shadows and highlights while preserving the mid-tones.

8.2.67. surface blur

Technical information

description
apply edge-aware surface blur to denoise or smoothen textures.
purpose
corrective and creative.
input
linear, RGB, scene-referred.
processing
linear, RGB.
output
linear, RGB, scene-referred.

Smooth image surfaces while preserving sharp edges using a bilateral filter.

This module *can* be used to denoise images, however you should be aware that bilateral filters are susceptible to overshoots and darktable offers much better alternatives. For example, the *astrophoto denoise* module uses a non-local means denoising algorithm, and the *denoise (profiled)* module provides a choice between non-local means and wavelet denoising algorithms.

The *surface blur* module blurs noise within the surfaces of an image by averaging pixels with their neighbors, taking into account not only their geometric distance but also their distance on the range scale (i.e. differences in their RGB values). It can be particularly useful if one RGB channel is more noisy/needs more smoothing than the others. In such a case, use the *color calibration* module to examine the channels one by one, in order to set the blur intensities accordingly.

In chromaticity blending mode , this module can be used to reduce the appearance of color moire resulting from fine repeated patterns on the object being photographed. You will likely need to increase the values of the Red, Green, and Blue sliders from their defaults for optimal effectiveness.

The module can also be used as a creative filter to provide interesting effects, for example to lend an image a cartoon-like appearance. When used with chrominance blending, it can also be used to even out the color of a surface (for example, to remove skin redness).

This module is resource-intensive and slows down pixelpipe processing significantly. You should consider activating it late in your workflow.

module controls

radius
The spatial extent of the gaussian blur.
red, green, blue
The blur intensity for each of the RGB channels.

8.2.68. tone curve

Technical information

description
alter an image’s tones using curves.
purpose
corrective and creative.
input
linear or non-linear, Lab, display-referred.

processing
non-linear, Lab.
output
non-linear, Lab, display-referred.

A classic digital photography tool to alter the tones of an image's using curves.

This module is very similar to the *rgb curve* module but works in Lab color space.

Activate the picker to show the picked values on the displayed histogram. Numerical (Lab) values of the input and output (see below) at the selected spot or area are shown at the top left of the graph.

module controls

*Please refer to the curves section for more details about how to modify curves including the **interpolation method**, **scale for graph** and **preserve colors** controls.*

color spaces

Depending on the desired intent, the tone curve can be applied in one of three color spaces:

- *Lab (linked or separated channels)*: Lab is a perceptual color space that is designed to approximate the way humans perceive colors and lightness. It represents color information independently of lightness. In "Lab, separated channels" mode, you are given fully independent control over the luminance (L-channel) and the chrominance (a/b-channels). In "Lab, linked channels" mode, only the luminance (L-channel) control is available. The color saturation correction is automatically computed, for each pixel, from the contrast correction applied to the luminance channel. This works better in cases where a subtle contrast correction is applied, but gives increasingly inaccurate saturation correction as the contrast is more dramatically altered.
- *XYZ (linked channels)*: XYZ is a linear technical color space designed to link the physiological light response of the human eye to RGB color spaces. As with Lab, it separates lightness from color information, but does so in a way that does not account for the role of the brain's correction in human perception. The "XYZ, linked channels" mode offers an alternative to "Lab, linked channels", and works by applying the L-channel curve to all three channels in the XYZ color space. Look at blend mode "coloradjustment" if you want to tune the strength of automatic chroma scaling. This mode is known to produce a slight hue shift towards yellow.
- *RGB (linked channels)*: RGB is a linear color space designed to capture and display images in additive synthesis. It is related to capture and display media and does not isolate color and lightness information. The "RGB, linked channels" mode works in ProPhoto RGB and applies the L-channel curve to all three channels in the RGB color space. Adding contrast in RGB space is known to desaturate highlights and boost saturation in the shadows, but this has proven to be the most reliable way to edit contrast, and is the standard method used in most software. This mode makes the tone curve module behave in much the same way as the base-curve module, except that the latter works in camera RGB space.

Note that the module works in Lab color space in all cases. This means that the middle-gray coordinate is always 50% in the graph, no matter what color space is used. The same applies to the inset histogram displayed in the background of the curve. The controls are converted to the relevant color space before the corrections are applied – in RGB and XYZ, the middle-gray is therefore remapped from 50% to 18%.

L-channel curve

The tone curve in L-channel works on Lightness. To provide a better overview, a lightness histogram is displayed in the module. When working in one of the "linked channels" modes, the L-channel curve is the only one available.

The horizontal axis represents the lightness of the input image pixels. The vertical axis represents the lightness of the output image pixels. The default straight diagonal line does not change anything. A curve above the default diagonal increases the lightness, whereas a curve below decreases it. An S-like curve will enhance the contrast of the image. You can move the end-points of the default diagonal to change the black and white points and hence alter the overall contrast – this has the same effect as changing the black and white points in the *levels* module.

a/b-channel curves

The curves in the a and b channels work on color values and are available only in the “Lab, separated channels” mode. The horizontal axis represents the color channel value of the input image pixels. The vertical axis represents the color channel value of the output image pixels. Positive a-values correspond to more magenta colors; negative a-values correspond to more greenish colors. Positive b-values correspond to more yellowish colors; negative b-values correspond to more blueish colors.

The default diagonal straight line does not change anything. Shifting the center of the curve will give the image a color tint: shifting the a-channel upwards gives a magenta tint; shifting the b-channel upwards gives a yellow tint; shifting the a-channel downwards gives a green tint; shifting the b-channel downwards gives a blue tint.

Increasing/decreasing the steepness of a curve, without shifting its center, will increase/decrease the color saturation of the respective channel. With properly defined curves you can exert fine control on color saturation, depending on the input pixel’s colors.

8.2.69. tone equalizer

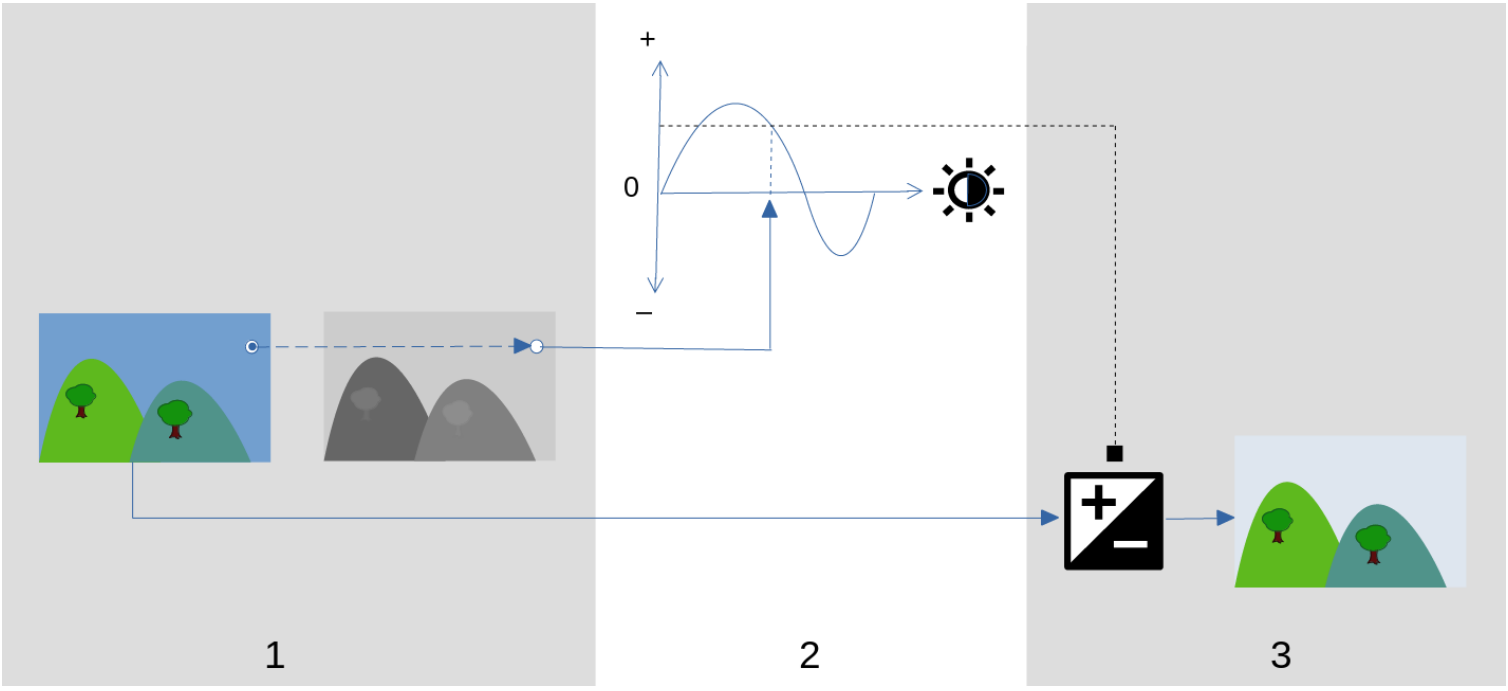
Technical information

description
relight the scene as if the lighting was done directly on the scene.
purpose
corrective and creative.
input
linear, RGB, scene-referred.
processing
quasi-linear, RGB.
output
quasi-linear, RGB, scene-referred.

Dodge and burn while preserving local contrast.

The *tone equalizer* module replaces the need for other tone-mapping modules such as the *shadows and highlights* , *tone curve* , *zone system (deprecated)* , and similar modules. It works in linear RGB space and utilizes a user-defined mask to guide the dodging and burning adjustments, helping to preserve local contrast within the image.

The following diagram describes how the tone equalizer works:



1. Create a monochrome *guided mask* that divides the input image into regions of similar luminosity. The resulting mask should blur the fine details within the image so that pixels within each region are all treated similarly, preserving local contrast.

2. Adjust the sliders in the *simple* tab or the equalizer graph in the *advanced* tab to alter the brightness of the underlying image, based on the brightness of the mask. Exposure can also be adjusted by scrolling with the mouse while hovering the cursor over the preview image (see *cursor indicator/control* for details).

   In the simple tab, each slider corresponds to a single brightness zone (EV) in the mask, which can be raised or lowered to adjust the exposure of the image where the mask's brightness lies in that zone. Similarly, in the equalizer tab, the horizontal axis of the graph corresponds to the brightness level of the mask, and the vertical axis specifies the required exposure adjustment to pixels where the mask matches that brightness level.

3. The exposure of each pixel of the input image is adjusted using the mask and the equalizer graph. For each pixel, the module looks up the brightness of the mask at that point, finds the matching brightness on the horizontal axis of the equalizer graph, and increases or decreases the exposure accordingly (using the vertical axis of the graph).

It is important that the mask separates the image into regions of similar brightness, and that a suitable amount of blur is applied within those regions. This means that all of the pixels in each region will have their exposure adjusted similarly, without adversely affecting local contrast. Examine your image beforehand to identify which regions you wish to dodge and burn, and use the controls on the *masking* tab to ensure that those areas are reasonably separated in tone within the final mask. This will allow those regions to be adjusted independently.

module controls

The controls of the *tone equalizer* module are divided between three tabs.

display exposure mask
Click on the icon to the right of this label to show/hide the module's guided mask over the top of the image. This control is available in all three tabs.

simple tab

This tab splits the brightness of the guided mask into nine zones (from –8 to 0 EV) and allows you to alter each zone independently. This is a simplified interface and is used to generate the same tone adjustment curve as shown in the *advanced* tab.

–8 EV … 0 EV
Each slider adjusts the exposure of all pixels where the *guided mask* has the given brightness. If the mask's histogram is evenly spread over the entire tonal range, sliders towards the top generally affect the shadows, whereas sliders towards the bottom generally affect the highlights. You can check the spread of the histogram within the *advanced* tab.

advanced tab

This tab allows you to control the same intensity levels as in the simple tab, though here they are represented as control points on a curve. Behind the curve is a histogram representing the intensity levels of the *guided mask* (not the underlying image). If the histogram is too bunched up, this means your mask doesn't have a good spread of intensity levels, which makes it harder to independently control the brightness of the different parts of your image. It is therefore recommended that the histogram be adjusted so that it extends the entire range, covering as many control points as possible for maximum flexibility. You can adjust the mask using the controls in the *masking* tab.

Click+drag the control points on the curve to adjust the brightness of all pixels where the mask has the given intensity. If the mask's histogram is evenly spread over the entire tonal range, control points to the left will generally affect the shadows, and control points to the right will generally affect the highlights. Moving a single control point will also affect control points on either side to ensure the curve remains smooth. This behavior can be adjusted using the *curve smoothing* control.

curve smoothing
Control how the curve is interpolated between control points. Move the slider to the right to make the transitions between the control points more gradual, but beware that going past about 0.6 can introduce some instability (oscillations) in the curve due to mathematical constraints. Move the slider to the left for a more well-behaved curve, but beware that this can result in harsher tonal transitions that may damage local contrast.

mask exposure compensation

Adjust the mask's histogram to the left or right. If you have used the *exposure* module to adjust the image brightness, you may need to offset that adjustment by using this slider to re-centre the mask's histogram. Click on the wand icon to the right of the slider to set the exposure compensation such that the average of the mask's histogram will coincide with the central –4EV control point. The slider can then be fine-tuned as required.

mask contrast compensation

Dilate (spread out) or compress the mask's histogram. The wand icon to the right of the slider will propose a reasonable starting point, which can then be fine-tuned to optimize the spread of the histogram under the tone equalizer control points.

masking tab

This tab contains controls for adjusting the guided mask.

The purpose of the guided mask is to separate out areas with different tonal ranges so that they can be independently brightened or darkened by the tone equalizer. The masking filters are designed to allow sharp edges between these areas to be preserved, while blurring details within a given tonal range, so that the brightness can be adjusted without adversely impacting local contrast. Ideally the mask histogram shown in the *advanced* tab should be spread out across all of the control points.

When setting up the guided mask you will need to strike a balance between obtaining a smooth blur within tonal regions (to preserve local contrast) and preservation of the boundaries between those regions. Some experimentation will be required to find the best settings. Often the key controls to adjust are the *exposure/contrast compensation* sliders at the bottom of the module.

Displaying the guided mask while making these adjustments will help you to understand these controls and better judge the quality of the mask.

luminance estimator

Choose the method by which the luminance of a pixel will be estimated when mapping it to a mask intensity value (default *RGB euclidean norm*).

preserve details

Choose the smoothing algorithm used to blur the mask:

- *no:* Do not smooth the mask (the effect is the same as using normal tone curves). When the module is used to compress dynamic range, this option can cause compression of local contrast. This can be useful when increasing (local and global) contrast.
- *guided filter:* Use the original guided filter algorithm to blur the mask while attempting to preserve edges. One of the limitations of this algorithm is that the guided filter is exposure-sensitive, meaning that shadows tend to be blurred more than highlights. Note that this limitation can sometimes be an asset: if one wants to lighten shadows a lot, the guided filter can provide very good local contrast preservation.
- *average guided filter:* Use this option in cases where the effect of the guided filter is too strong. In this mode, a geometric mean is taken between the output of the original *guided filter* algorithm and the output given by the *no* option.
- *eigf (default):* The *exposure-independent guided filter* solves the problem of the original *guided filter*, in that it makes the degree of blurring independent of the exposure. This means the degree of blurring applied to the highlights and the shadows regions should be about the same. This improved algorithm is now the default option.
- *averaged eigf:* This option takes the geometric mean between the *eigf* mask and the mask generated by the *no* option, and is useful in cases where the degree of blurring in the mask needs to be mitigated.

filter diffusion

By default this is set to a value of 1, meaning that the filtering algorithm is run once on the input image to produce the blurred monochrome mask.

If you increase this to 2, the filtering algorithm will be run once on the input image to produce an intermediate mask, and then a second time on the intermediate mask. As a result, the final mask will be more blurred than when using a single iteration. Progressively higher values will diffuse the mask even further. However, because the mask filtering algorithm is run multiple times, each iteration will increase the processing time.

smoothing diameter

This controls how much of the surrounding image to take into account when calculating the mask's blur at a particular point, defined as a percentage of the length of the image's longer side (default 5%). With lower values, the transition between darker and lighter areas of the mask will be more pronounced. As the value is increased, transitions become smoother/softer. For the default *exposure-independent guided filter (eigf)*, you should typically use blurring radii around 1-10%. With the original *guided filter*, blurring radii around 1-25% will typically yield better results.

edges refinement/feathering

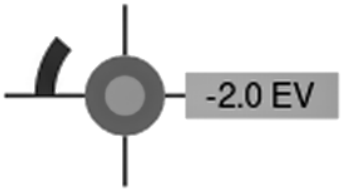
Higher values force the mask to follow high contrast edges more closely. Lower values give smoother gradients, but may introduce halos. If required, feathering can be set to values as high as 10,000.

mask quantization

Apply a degree of posterization to the mask, so that it tends to centre round a few discrete levels. In some cases, this may be useful to help separate out areas of your image into distinct masking levels.

cursor indicator/control

When the *tone equalizer* module is enabled and expanded, you can move the mouse pointer over the preview image to show a cursor that displays information about the pixel under the pointer. When this cursor is shown, the mouse wheel can be used to brighten or darken the areas of your image that match the mask intensity level at that point. This provides a convenient way to quickly brighten or darken specific parts of the image.



- the cross-hairs indicate the position of the pixel under the cursor
- the text label shows the intensity of the guided mask at that point, in EV
- the shade of the outer circle indicates the intensity of the mask at that point
- if the tone equalizer has brightened or darkened pixels matching this mask intensity, the magnitude of the adjustment is indicated by an arc on the left-hand-side. The longer the arc, the greater the brightness adjustment,
- if there has been an exposure adjustment, the shade of the inner circle indicates the amount of brightening or darkening, relative to the mask's intensity at that point (as indicated by the outer gray circle). That is to say, if the pixel under the crosshairs has been brightened, the inner circle will be a lighter shade of gray than the outer circle; if the pixel has been darkened, the inner circle will be a darker shade of gray than the outer circle.

If you need to move or zoom the portion of the image shown in the center view while the module is expanded, hold down the 'a' key while dragging the mouse or using the scroll wheel. As long as the key is held down, mouse actions adjust the viewport rather than adjusting the tone curve.

presets

The *tone equalizer* comes with several presets that can use used to compress shadows and highlights. Each comes in two variants, using either the guided filter (gf) or the exposure-independent guided filter (eigf). The variants using the guided filter tend to preserve local contrast in the shadows better those that use the exposure-independent guided filter, but at the price of reducing the local contrast in the highlights. Either of these variants may lead to better results, depending on the image. In both cases, the presets preserve middle-gray, so you shouldn't need to adjust the global exposure after activating the tone equalizer.

8.2.70. unbreak input profile

Technical information

description
correct input color profiles meant to be applied on non-linear RGB.

purpose
corrective.

input
linear, RGB, display-referred.

processing
non-linear, RGB.
output
non-linear, RGB, display-referred.

Add a correction curve to image data. This is required if you have selected certain input profiles in the *input color profile* module.

If you decide to use an ICC profile from the camera manufacturer in the *input color profile* module, a correction curve frequently needs to be pre-applied to image data to prevent the final output from looking too dark. This extra processing is not required if you use darktable's standard or enhanced color matrices.

The correction curve is defined with a linear part extending from the shadows to some upper limit and a gamma curve covering mid-tones and highlights. For further information please see darktable's neighboring project UFRaw .

linear
The upper limit for the region counted as shadows and where no gamma correction is performed. Typically values between 0.0 and 0.1 are required by the profile.
gamma
The gamma value required to compensate the input profile. Often the required value is 0.45 (the reciprocal of the 2.2 gamma used by some manufacturer's profiles).

8.2.71. velvia

Technical information

description
resaturate giving more weight to blacks, whites and low-saturation pixels.
purpose
creative.
input
linear, RGB, scene-referred.
processing
linear, RGB.
output
linear, RGB, scene-referred.

Resaturate pixels in a weighted manner that gives more weight to blacks, whites and less saturated pixels.

Note: *This module causes hue and brightness shifts that can be difficult to manage. Instead, use the color balance rgb module for color modifications.*

---

module controls

strength
The strength of the effect
mid-tones bias
Reduce the effect on mid-tones in order to avoid unnatural skin tones. Reducing the mid-tone bias decreases mid-tone protection and makes the overall velvia effect stronger.

8.2.72. vignetting

Technical information

description
simulate a lens fall-off close to edges.

purpose
creative.
input
non-linear, RGB, display-referred.
processing
non-linear, RGB.
output
non-linear, RGB, display-referred.

Apply a vignetting effect to the image.

Vignetting is a modification of the brightness and saturation at the borders of the image in a specified shape. Many of the parameters listed below can also be modified with a graphical control that overlays the image when the module has focus, showing the shape and extent of the effect.

Note: This module is known to provoke banding artifacts under certain conditions. You should consider activating the *dither or posterize* module to alleviate this.

Also note: This module can produce unnatural-looking results and should be used with care. Instead, use the exposure module with an elliptical mask with a large transition area and, if necessary, use the color balance rgb module with the same mask to reduce saturation at the edges.

---

module controls

fall-off start
The radius of the central vignetting area.
fall-off radius
The progressiveness of the fall-off. Lower values cause a steeper transition.
brightness
The intensity of brightening (positive values) or darkening (negative values).
saturation
Control how strong colors become when desaturated/saturated in the darkened/brightened vignetting area.
horizontal/vertical center
Shift the center of the vignetting area horizontally/vertically.
shape
The shape of the vignetting effect. The default value of 1 creates a circular or elliptical area. Smaller values shift the shape to being more square; higher values turn it into a cross-like shape.
automatic ratio
Automatically adjust the width/height ratio of the vignetting area to match the aspect ratio of the underlying image. This typically causes the vignetting area to become elliptical.
width/height ratio
Manually adjust the width/height ratio of the vignetting area.
dithering
Activate random noise dithering to alleviate banding artifacts caused by vignette gradients. Select “8-bit output” to prevent banding on monitor display and for JPEGs. When set to “16-bit output”, only a small amount of dithering is applied, just strong enough to compensate for banding on the fine grained 16-bit level. It is now recommended that you instead use the *dither or posterize* module to alleviate banding artifacts.

8.2.73. watermark

Technical information

description
overlay an SVG watermark like a signature on the image.

purpose
creative.
input
non-linear, RGB, display-referred.
processing
non-linear, RGB.
output
non-linear, RGB, display-referred.

Render a vector-based overlay onto your image. Watermarks are standard SVG documents and can be designed using Inkscape .

You can also use bitmap (PNG) images.

The SVG processor in darktable can also substitute strings within an SVG document, allowing image-dependent information to be included in the watermark.

User-designed watermarks should be placed into the `watermarks` folder in the darktable configuration directory . Once in place, use the reload button update the list of available watermarks.

The following is a list of variable strings that are supported for substitution within an SVG document.

In addition to this list you can also use the variable strings defined in the variables section.

```
$(WATERMARK_TEXT)             A short free text (max. 511 characters)
$(WATERMARK_COLOR)            The color to use for $WATERMARK_TEXT
$(WATERMARK_FONT_FAMILY)      The font family to use for $WATERMARK_TEXT
$(WATERMARK_FONT_STYLE)       The font style (normal, oblique, italic)
$(WATERMARK_FONT_WEIGHT)      The font weight (boldness)
```

module controls

marker
Choose the watermark to apply. Use the reload button to update the list to include any newly-added watermarks. The extension of the file (png/svg) is shown in brackets.
text
A free text field in which you can enter up to 511 characters to be printed where referenced by the corresponding watermark. Any variables contained in the string will be replaced by their current values before the watermark is applied. An example watermark using the text from this field is supplied as `simple-text.svg`.
font
The font to use (default “DejaVu Sans Book”). Click on the field to open a dialog box showing the fonts available on your system. Fonts can be searched by name and a preview is shown next to the font name. You may specify your own sample text.
color
The color of the text. Click on the colored field to open a color selector dialog which offers you a choice of commonly used colors, or allows you to define a color in RGB color space.
opacity
The opacity of the watermark’s rendering.
rotation
The rotation angle of the watermark.
scale
The scale of the watermark (in per cent), with respect to the option selected in the “scale on” parameter.

scale on

The reference for the scale parameter – how the watermark is scaled relative to the image:

- “image” (default): Scale the watermark relative to the whole image,
- “larger border”: Scale the larger border of the watermark relative to the larger border of the image,
- “smaller border”: Scale the smaller border of the watermark relative to the smaller border of the image,
- “height”: Scale the height of the watermark relative to the height of the image (useful for text with a constant font height),
- “advanced options”: activate additional options (below) to allow you to choose which image dimension to scale to which watermark dimension. For example, you can choose to scale the watermark’s height relative to the image’s width.

scale marker to (“advanced options” only)

The image reference against which the watermark should be scaled – “image width”, “image height”, “larger image border” or “smaller image border”.

scale marker reference (“advanced options” only)

The dimension of the unrotated watermark (“marker width” or “marker height”) to use as a scaling reference.

alignment

Use these controls to align the watermark to any edge or the center of the image.

x offset

Pixel-independent offset relative to the choice of alignment on the x-axis.

y offset

Pixel-independent offset relative to the choice of alignment on the y-axis.

Note: When using the watermark in combination with the framing module, the added frame affects the watermark’s position. For example, if you align the watermark to an edge, it may appear within the frame instead of on the actual image. To ensure the watermark is positioned relative to the image itself, you can manually adjust the module order so that the framing module comes after the watermark module in the pixelpipe.

8.2.74. white balance

Technical information

description
scale raw RGB channels to balance white and help demosaicing.

purpose
corrective.

input
linear, raw, scene-referred.

processing
linear, raw.

output
linear, raw, scene-referred.

Adjust the white balance of the image by altering the temperature and tint, defining a coefficient for each RGB channel, or choosing from list of predefined white balance settings.

The default settings for this module are derived from the camera white balance stored in the image’s Exif data.

White balance is not intended as a “creative” module – its primary goal is to technically correct the white balance of the image ensuring that neutral colored objects in the scene are rendered with neutral colors in the image. For creative color operations, it is usually better to use other modules such as *color calibration* or *color balance rgb* .

Note: The *color calibration* module now provides a more modern and flexible method of controlling white balance. The *color calibration* module can be enabled by default for new images by selecting “scene-referred” (filmic or sigmoid) in preferences > processing > auto-apply pixel workflow defaults . Some basic settings are still required (and applied automatically) in the *white balance* module, so that *demosaic* works correctly.

---

module controls

scene illuminant temp

This section provides scene-illuminant *temperature* and *tint* controls to adjust the white balance of the image. Click on the 'scene illuminant temp' section label to cycle between 3 different color modes for the temperature/tint sliders.

temperature
Set the color temperature in kelvin.

tint
Alter the color tint of the image, from magenta (tint < 1) to green (tint > 1).

white balance presets

setting
Choose from a predetermined list of white balances. The available settings are derived from the presets available in the camera used to take the photograph. The following options are provided in addition to any camera-defined white balance presets.

- *as shot*: The white balance as reported by the camera
- *from image area*: Draw a rectangle over a neutral color in the image to calculate white balance from that area.
- *user modified*: The most recently modified setting. Manual adjustment of temperature, tint or r/g/b channel coefficients will automatically select this option. Choose this setting after selecting any other preset to return parameters to the most recent user-modified state
- *camera reference*: Set the temperature to the camera reference white point, which is assumed to be D65 (or ~6502K). The white balance channel multipliers are calculated such that pure white in the camera colorspace is converted into pure white in sRGB D65 (where pure white means that each color channel has an equal value).
- *as shot to reference* (default): The white balance as reported by the camera is used in processing modules before *input color profile* for slightly better results in *highlights reconstruction* , *raw chromatic aberrations* , *demosaic* or *denoise (profiled)* . This is corrected later in the pipe in *input color profile* to the camera reference white point as in *camera reference*. Please note that, in a few cases, the coefficients reported by the camera for 'as shot' are bad so you might prefer *camera reference* for those cameras.

For convenience these modes can also be set by clicking on one of the buttons in the button bar above the *setting* drop-down.

finetune
Finetune a camera-specific white balance preset. This is only shown if it is available for the camera in question. The direction of adjustment is dependent on the provided presets. If your camera doesn't have white balance presets available, check this guide to see how you can submit your own.

channel coefficients

The RGB channel coefficients are automatically calculated from the above parameters and, as such, are hidden by default. You can expand/collapse the channel coefficients section by clicking on either the 'channel coefficients' label or the adjacent triangular button.

red/green/blue
Set the value of each RGB channel coefficient from 0 to 8

additional functionality

colored sliders

By default the module's sliders are monochrome. However, two flavors of colored sliders can be enabled by clicking on the 'scene illuminant temp' section label in the module.

no color (default)
The background of the sliders is not colored.

illuminant color
The slider colors represent the color of the light source (the color you are adjusting to in order to achieve neutral white).
effect emulation
The slider colors represent the effect the adjustment would have had on the scene. This is how most other raw processors show temperature/tint sliders colors.

button bar

The button bar is simple addition that allows one-click access to the internal white balance settings. You can disable this by editing your `darktablerc` file. Find the line that says

```
plugins/darkroom/temperature/button_bar=TRUE
```

and change `TRUE` to `FALSE`.

usage warning

The only parameters that are used internally by this module's operation are the *rgb channel coefficients*. The *temperature* and *tint* sliders are provided as a more user-friendly way to adjust those parameters. The relationship between the channel coefficients and temperature/tint sliders depends on characteristics specific to the camera used to take the photograph. This means that applying white balance settings from an image made with one camera to an image made with another will, in general, not give consistent results.

The mathematical relationship between the two sets of values is not straightforward. It is possible to set the channel coefficients such that there is no valid equivalent temperature and tint setting (mainly where very high temperature values are calculated from the sliders). Editing white balance using temperature and tint on an image previously edited using channel coefficients may therefore give odd results, at least where high temperature values are involved.

8.2.75. (deprecated) basic adjustments

Technical information

description
apply usual image adjustments.
purpose
creative.
input
linear, RGB, scene-referred.
processing
non-linear, RGB.
output
non-linear, RGB, scene-referred.

Please note that this module is deprecated from darktable 3.6 and should no longer be used for new edits. Please use the quick access panel instead.

A convenience module that combines controls from *exposure* , *highlight reconstruction* , *color balance* and *vibrance* into a single module.

While this module can provide a quick and convenient way to make simple adjustments to an image, it must be used with care. Normally exposure adjustments should come before input color profile in the pixelpipe, and color adjustments should come after. Because the *basic adjustments* module combines all of these functions into a single operation in the pixelpipe, it may not play nicely with other modules. Therefore, if you plan to use *basic adjustments* with other modules, please instead consider using the *exposure* + *base curve* / *tone curve* / filmic rgb + *color balance* modules so that these operations occur in the correct places in the pixelpipe.

module controls

black level correction

Equivalent to black level correction in the *exposure* module, this slider defines the threshold at which dark gray values are cut off to pure black. Reducing this can bring some dark colors back into gamut. Increasing this slider may appear to increase the contrast and pop of an image, but it can push dark colors out of gamut, and clipped data cannot be recovered further down the pixel pipe. To control the contrast in the shadows, it is better to use other modules such as *tone curve* or *levels* , which mitigate these negative impacts further down the pixel pipeline.

exposure

Adjust exposure compensation. Adding 1 EV of exposure compensation is equivalent to doubling the exposure time in camera, opening the aperture by 1 stop, or doubling the ISO.

Positive exposure corrections will make the image brighter. As a side effect noise level is increased. Depending on the basic noise level of your camera and the ISO value of your image, positive exposure compensations with up to 1EV or 2EV should still give reasonable results.

Negative exposure corrections will make the image darker. Given the nature of digital images this can not correct for fully blown out highlights but allows you to reconstruct data if only some of the RGB channels are clipped (see the *highlight reconstruction* module for information on how to deal with clipped pixels).

highlight compression

Compress the highlights of the image in order to recover detail.

contrast

Equivalent to the *contrast* slider in the *color balance* module, this is used to increase the separation of luminance values around a fulcrum point, in effect making the tone curve steeper. The *middle gray* slider sets the fulcrum point for the contrast, and any pixels whose luminance matches this middle gray point will be unaffected. Pixels brighter than the middle gray point will become even brighter, and pixels darker than the middle gray point will become even darker.

preserve colors

If a non-linear tone curve is applied to each of the RGB channels individually, then the amount of tone adjustment applied to each color channel may be different, and this can cause hue shifts. Therefore, the *preserve colors* menu provides different methods of calculating the “luminance level” of a pixel. The amount of tone adjustment is calculated based on this luminance value, and then this same adjustment is applied to all three of the RGB channels. Different luminance estimators can affect the contrast in different parts of the image, depending on the specific characteristics of that image. The user can therefore choose a particular estimator that provides the best results for the given image.

middle gray

Set the fulcrum point for the *contrast* slider. This is equivalent to the *contrast fulcrum* slider on the *color balance* module. If the contrast slider is set to 0 this slider will not have any effect.

brightness

Equivalent to increasing the *gamma factor* slider in the *color balance* module. Moving the slider to the right will increase the brightness of the image, with an emphasis on the mid-tones.

saturation

Equivalent to the *output saturation* slider on the *color balance* module, this slider affects the saturation, or brilliance and intensity of the colors. Moving the slider to the right will increase the amount of color in the image; moving the slider to the left will reduce the amount of color in the image.

vibrance

Accentuate the colors of the image without adding unnatural colors, as it’s often the case with the *saturation* slider. It works by reducing the lightness of already saturated pixels to make the colors more vivid. You can also achieve some interesting effects by combining it with the saturation slider to target more or less saturated areas of the image.

auto

Automatically adjust the exposure, taking into account the entire image, or use the picker to select a rectangular area of the image – the exposure will be automatically adjusted based on the selected region. This allows you to prioritise which parts of the image should be well-exposed.

clip

This affects the number of pixels that will be clipped to black or white during the auto-exposure calculation. Moving this slider to the right will allow more pixels to be clipped and increase the contrast; moving this slider to the left will compress the image more and lower the contrast.

8.2.76. (deprecated) channel mixer

Technical information

description
perform color space corrections such as white balance, channels mixing and conversions to monochrome emulating film.

purpose
corrective or creative.

input
linear, RGB, display-referred.

processing
linear, RGB.

output
linear, RGB, display-referred.

Please note that this module is deprecated from darktable 3.4 and should no longer be used for new edits. Please use the *color calibration* module instead.

A simple yet powerful tool to manage color channels.

This module accepts red, green and blue channels as an input and provides red, green, blue, gray, hue, saturation and lightness channels as output. It allows you to independently control how much each input channel contributes to each output channel.

RGB matrix multiplication

You can think of the *channel mixer* as a type of matrix multiplication between a 3x3 matrix and the input [R G B] values.

```
┌ R_out ┐     ┌ Rr Rg Rb ┐     ┌ R_in ┐
│ G_out │  =  │ Gr Gg Gb │  X  │ G_in │
└ B_out ┘     └ Br Bg Bb ┘     └ B_in ┘
```

If, for example, you've been provided with a matrix to transform from one color space to another, you can enter the matrix coefficients into the *channel mixer* as follows:

- set the destination to *red* then set the Rr, Rg & Rb values using the red, green and blue sliders
- set the destination to *green* then set the Gr, Gg & Gb values using the red, green and blue sliders
- set the destination to *blue* then set the Br, Bg & Bb values using the red, green and blue sliders

By default, *channel mixer* just copies the input [R G B] channels straight over to the matching output channels. This is equivalent to multiplying by the identity matrix:

```
┌ R_out ┐     ┌ 1  0  0 ┐     ┌ R_in ┐
│ G_out │  =  │ 0  1  0 │  X  │ G_in │
└ B_out ┘     └ 0  0  1 ┘     └ B_in ┘
```

As an example use case, the following matrix is useful for taming ugly out-of-gamut blue LED lights by making them more magenta:

```
┌  1.00  -0.18  0.18 ┐
│ -0.20   1.00  0.20 │
└  0.05  -0.05  1.00 ┘
```

In this case it is useful to use a parametric mask to limit the effect of the *channel mixer* to just the problematic colors.

A more intuitive take for what the *channel mixer* sliders do:

- for the *red* destination, adjusting sliders to the right will make the R, G or B areas of the image more red. Moving the slider to the left will make those areas more cyan.
- for the *green* destination, adjusting sliders to the right will make the R, G or B areas of the image more green. Moving the slider to the left will make those areas more magenta.
- for the *blue* destination, adjusting sliders to the right will make the R, G or B areas of the image more blue. Moving the slider to the left will make those areas more yellow.

monochrome

Another very useful application of the *channel mixer* is the ability to mix the channels together to produce a grayscale output – a monochrome image. Use the *gray* destination, and set the red, green and blue sliders to control how much each channel contributes to the brightness of the output. This is equivalent to the following matrix multiplication:

```
GRAY_out  =   [ r  g  b ]  X  ┌ R_in ┐
                              │ G_in │
                              └ B_in ┘
```

When dealing with skin tones, the relative weights of the three channels will affect the level of detail in the image. Placing more weight on red (e.g. [0.9, 0.3, -0.3]) will make for smooth skin tones, whereas emphasising green (e.g. [0.4, 0.75, -0.15]) will bring out more detail. In both cases the blue channel is reduced to avoid emphasising unwanted skin texture.

Different types of traditional black and white film have differing sensitivities to red, green and blue colors, and this can be simulated by setting the gray destination coefficients appropriately. The *channel mixer* module has a number of built-in presets that can be used to achieve this.

module controls

destination
Select the destination channel that will be affected by the slider settings immediately below. The red, green and blue destination channels are used for color mixing as described in the *matrix multiplication* section above. The gray channel is used for making grayscale images as described in the *monochrome* section above. It is also possible to define the R, G & B input channels to produce HSL (hue, saturation and lightness) values on the output, although this is a very specialised application.

red
Defines how much the red input channel should contribute to the selected destination channel.

green
Defines how much the green input channel should contribute to the selected destination channel.

blue
Defines how much the blue input channel should contribute to the selected destination channel.

8.2.77. (deprecated) contrast brightness saturation

Technical information

description
adjust the look of the image.
purpose
creative.
input
non-linear, Lab, display-referred.
processing
non-linear, Lab.
output
non-linear, Lab, display-referred.

Please note that this module is deprecated from darktable 4.4 and should no longer be used for new edits. Please use the *color balance rgb* module instead.

A very basic tool for adjusting the *contrast*, *brightness* and *saturation* of an image. Please note that a number of other modules provide much more versatile methods of adjusting these parameters.

All module controls default to a neutral position (zero) and provide the ability to increase or decrease the relevant parameter

Note: *This module works in Lab color space and is prone to halos. Instead, use the color balance rgb module.*

module controls

contrast
The contrast
brightness
The brightness
saturation
The saturation

8.2.78. (deprecated) crop and rotate

Technical information

description
change the framing and correct the perspective.
purpose
corrective or creative.
input
linear, RGB, scene-referred.
processing
geometric, RGB.
output
linear, RGB, scene-referred.

Please note that this module is deprecated from darktable 3.8 and should no longer be used for new edits. Please use the *crop* module to crop the image, the *rotate and perspective* module to perform rotation and keystone correction, and the *orientation* module to flip the image on the horizontal/vertical axes.

Crop, rotate, and correct perspective distortions using on-screen guides.

Whenever the user interface of this module is in focus, the full uncropped image will be shown, overlaid with crop handles and optional guiding lines.



Resize the crop by dragging the border and corner handles.

Move the crop rectangle by clicking and dragging inside the crop area. Constrain movement to the horizontal/vertical axis by holding Ctrl/Shift, respectively while dragging. Commit changes by giving focus to another module.

If you intend to use the *retouch* module, you are recommended to use *rotate and perspective* for rotation and/or keystone correction only, performing creative cropping in the *crop* module. This process ensures that the entire image is available for source spots in *retouch*, since the *crop* module is placed after *retouch* in the pixelpipe.

Note: Some of the tools in this module (angle adjustment and perspective distortion correction) require the original image data to be interpolated. For best sharpness results set “lanczos3” as the pixel interpolator in preferences > processing .

---

module controls

The *crop and rotate* module controls are split between two tabs as follows:

main tab

flip
Flip the image on the horizontal, vertical or both axes.

angle
Correct the rotation angle to level an image by setting a numerical value in degrees or using the mouse. To use the mouse, right-click and drag to draw a line along a suitable horizontal or vertical feature. As soon as you release the mouse button the image is rotated so that the line you drew matches the horizontal or vertical axis.

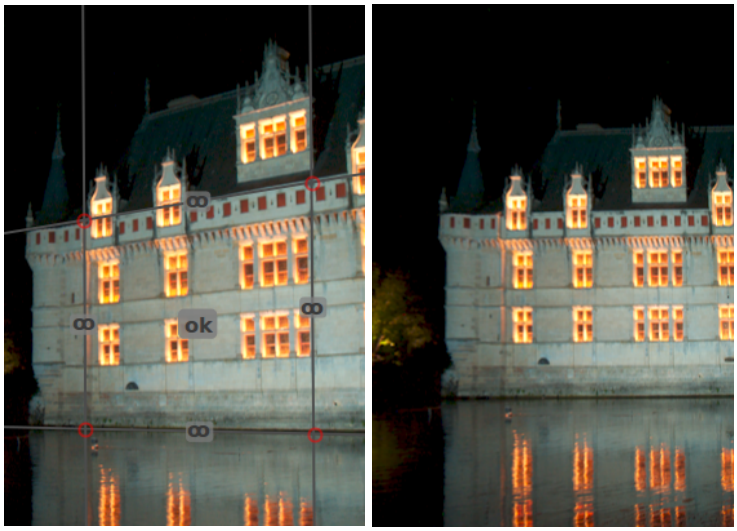
keystone

Correct perspective distortions. This tool is useful, for example, when you shoot a high building from the ground with a short focal length, aiming upwards with your camera. The combobox allows you to select the type of correction you want to use:

- *vertical*: Limit the correction to vertical lines
- *horizontal*: Limit the correction to horizontal lines
- *full*: Correct horizontal and vertical lines at the same time

Depending on the selected correction type you will see two or four straight adjustment lines overlaid on your image. Two red circles on each line allow you to modify the line positions with your mouse. Each line additionally carries a “symmetry” button. If activated (and highlighted in red) all movements of the affected line will be mirrored by the opposite line.

In order to correct perspective distortions, you need to find suitable horizontal and/or vertical features in your image and align the adjustment lines with them. When finished, press the “OK” button located close to the center of the image – corrections will be applied immediately. You can return to the module and refine your corrections by selecting “correction applied” in the keystone combobox.



automatic cropping

Automatically crop the image to avoid black edges on the image borders. This is useful when rotating the image.

aspect

Set the aspect ratio of the crop, constraining the width:height ratio of the crop rectangle to the chosen aspect. Many common numerical ratios are pre-defined. A few special aspect ratios deserve explanation:

- *freehand*: Crop without any restrictions
- *original image*: Retain the aspect ratio of the original image
- *square*: Constrains the aspect ratio to be 1:1
- *golden cut*: The golden ratio (1.62:1)

You can also enter any other ratio after opening the combobox by typing it in the form of “x:y” or as a decimal (e.g. “0.5” to apply a ratio of 2:1).

If you want to add an aspect ratio to the pre-defined drop-down list you can do this by including a line of the form “`plugins/darkroom/clipping/extra_aspect_ratios/foo=x:y`” in darktable’s configuration file `darktablerc` in the configuration directory . Here “`foo`” defines the name of the new aspect ratio and “x” and “y” the corresponding numerical values (x and y must be integers). Note that you can only add new entries for ratios not already present in the drop-down list.

Finally, the button beside the aspect combobox allows you to switch between portrait and landscape orientation if you have selected a rectangular aspect ratio.

Note: When resizing an image in freehand mode you may retain the currently-set aspect ratio by holding Shift while dragging on any of the resize controls.

---

show guides
Tick the box to show guide overlays whenever the module is activated. Click the icon on the right to control the properties of the guides. See guides & overlays for details.

margins tab

These sliders allow you to directly set how much of the image to crop from each side. They are automatically updated if you move or resize the crop area on the image using the mouse.

left
The percentage of the image that should be cropped from the left side.
right
The percentage of the image that should be cropped from the right side.
top
The percentage of the image that should be cropped from the top.
bottom
The percentage of the image that should be cropped from the bottom.

8.2.79. (deprecated) defringe

Technical information

description
attenuate chromatic aberration by desaturating edges.
purpose
corrective.
input
linear or non-linear, Lab, display-referred.
processing
non-linear, Lab.
output
non-linear, Lab, display-referred.

Please note that this module is deprecated from darktable 3.6 and should no longer be used for new edits. Please use the *chromatic aberrations* module instead.

Remove color fringing, which often results from Longitudinal Chromatic Aberrations (LCA), also known as Axial Chromatic Aberrations.

This module uses edge-detection. Where pixels are detected as a fringe, the color is rebuilt from less saturated neighboring pixels.

module controls

operation mode
global average: This mode is usually the fastest but might show slightly incorrect previews at high magnification. It might also protect the wrong regions of color too much or too little by comparison with local average.

local average: This mode is slower because it computes local color references for every pixel. However it might protect color better than global average and allows for rebuilding color where actually required.

static threshold: This mode does not use a color reference but directly uses the threshold as given by the user.

edge detection radius
The algorithm uses the difference between the input image and a gaussian-blurred version of the same image to detect edges. This parameter controls the spatial extent of the gaussian blur used in the edge detection. Try increasing this value if you want a stronger detection of the fringes or the thickness of the fringe edges is too high.

threshold
The threshold over which a pixel is counted as a “fringe”. Try lowering this value if there is not enough fringe detected and increasing it if too many pixels are desaturated.

8.2.80. (deprecated) fill light

Please note that this module is deprecated from darktable 3.4 and should no longer be used for new edits. Please use the *tone equalizer* module or the *exposure* module with a drawn mask .

Locally modify exposure based on pixel lightness.

This module pushes exposure by increasing lightness with a Gaussian curve of a specified width, centered on a given lightness.

module controls

exposure
The fill-light exposure (EV)
center
The median lightness impacted by the fill-light. The lightness must be set manually but a picker can be used to provide a guide based on a sample from the image. The lightness of the selected point/area is displayed in the gradient bar.
width
The width of the Gaussian curve. This number is expressed in zones, with the full range being 10 zones.

8.2.81. (deprecated) global tonemap

Please note that this module is deprecated from darktable 3.4 and should no longer be used for new edits. Please use the *filmic rgb* module instead.

Compress the tonal range of an HDR image into the limited tonal range of a typical LDR output file.

Global tonemap processes each pixel of an HDR image, without taking the local surrounding into account. This is generally faster than local *tone mapping (deprecated)* , but might lead to less convincing results with very high-dynamic-range scenes.

module controls

operator
The operator to use. *Reinhard*, *Filmic* and *Drago* global tonemap operators are available. Depending on the selected operator, different parameters can be adjusted.

Some operators are fully self-adjusting, and do not require specific controls.

bias (*Drago* operator only)
This parameter influences the contrast of the output image. It is an essential parameter for adjusting the compression of high values and the visibility of details in dark areas. A value of 0.85 is recommended as a starting point.
target (*Drago* operator only)
A scale factor to adjust the global image brightness to the brightness of the intended display. It is measured in cd/m, and should match the brightness of your output device. Higher values lead to a brighter image, while lower values lead to a darker image.
detail (all operators)
This parameter controls how much detail is preserved from the original input image and transferred back into the output image after tonemapping.

8.2.82. (deprecated) invert

Technical information

description
invert film negatives.
purpose
corrective.
input
linear, raw, display-referred.
processing
linear, raw.
output
linear, raw, display-referred.

Please note that this module is deprecated from darktable 3.4 and should no longer be used for new edits. Please use the *negadoctor* module instead.

Invert scanned negatives.

module controls

color of film material
Clicking on the colored field will open a color selector dialog which offers you a choice of commonly used colors, or allows you to define an RGB color. You can also activate the picker to take a color probe from your image – preferably from the unexposed border of your negative.

8.2.83. (deprecated) levels

Technical information

description
adjust black, white and mid-gray points.
purpose
creative.
input
linear or non-linear, Lab, display-referred.
processing
non-linear, Lab.
output
non-linear, Lab, display-referred.

Please note that this module is deprecated from darktable 4.4 and should no longer be used for new edits. Please use the *rgb levels* module instead.

Adjust black, white and mid-gray points.

The levels tool offers two modes of operation:

manual
The levels tool shows a histogram of the image, and displays three bars with handles. Drag the handles to modify the black, middle-gray and white points in absolute values of image lightness (the L value from Lab).

Moving the black and white bars to match the left and right borders of the histogram will make the output image span the full available tonal range, increasing the image’s contrast.

Moving the middle bar will modify the mid-tones. Move it to the left to make the image look brighter and move it to the right to make it darker. This is often referred to as changing the image’s gamma.

Three pickers are available for sampling the black, white and gray points from the image. The “auto” button auto-adjusts the black and white point and puts the gray point exactly in the mean between them.

automatic
The module automatically analyses the histogram of the image, detects the left and right histogram borders, and lets you define the black point, the gray point and the white point in terms of percentiles relative to these borders.

Note: Under certain conditions, especially with highly saturated blue light sources, the *levels* module may produce black pixel artifacts. See the “gamut clipping” option of the *input color profile* module for information about how to mitigate this issue.

---

module controls

mode
The mode of operation (automatic or manual).
black (automatic mode only)
The black point in percentiles relative to the left border of the histogram.
gray
The gray point in percentiles relative to the left and right borders of the histogram after having applied the black point and white point corrections.
white
The white point in percentiles relative to the right border of the histogram.

8.2.84. (deprecated) spot removal

Technical information

description
remove sensor dust spots.
purpose
corrective.
input
linear, RGB, scene-referred.
processing
geometric, raw.
output
linear, RGB, scene-referred.

Please note that this module is deprecated from darktable 3.6 and should no longer be used for new edits. Please use the “cloning” tool in the *retouch* module instead.

Correct areas of an image (the target) using details from another area of the image (the source).

This module uses some of the shapes that are available for drawn masks (circles, ellipses and paths), and the user interface is similar.

To use this module, first select the desired shape by clicking the corresponding shape icon (Ctrl+click to enable continuous shape creation mode). Next, you may optionally choose the source for the correction by using Shift+Ctrl+click or Shift+click on the image canvas to choose absolute or relative mode, respectively (see below). Finally click on the canvas to choose the area to be healed (the target area). A cross is displayed where the source area will be positioned.

The source positioning has two modes

absolute mode
Before placing a shape on the image canvas, Shift+Ctrl+click on the desired position for the source. From this point on, all new shapes will have their source created at the same absolute position (in image co-ordinates).

relative mode
Before placing a shape on the image canvas, Shift+click on the desired position for the source. The current shape will have the source created at that position and subsequent shapes will have their source created at the same position, *relative to the target shape*.

After creation the source and target areas of each shape can be shifted independently until the result matches your expectations. An arrow points from the source area of each shape to its target.

Use the shape-specific controls to adjust each shape's size, border width, and other attributes.

Right-click on a shape to delete it.

This module is equivalent to the "cloning" tool in the *retouch* module; see the description of the cloning tool for additional details and examples. If *spot removal* produces unsatisfactory results, you may want to try the "heal" tool in *retouch* instead.

8.2.85. (deprecated) tone mapping

Please note that this module is deprecated from darktable 3.4 and should no longer be used for new edits. Please use the *tone equalizer* module instead.

Compress the tonal range of HDR images so that they fit into the limits of an LDR image, using Durand's 2002 algorithm.

The underlying algorithm uses a bilateral filter to decompose an image into a coarse base layer and a detail layer. The contrast of the base layer is compressed, while the detail layer is preserved, then both layers are re-combined.

module controls

contrast compression
The contrast compression level of the base layer. A higher compression will make the image fit a lower dynamic range.

spatial extent
The spatial extent of the bilateral filter. Lower values cause the contrast compression to have stronger effects on image details.

8.2.86. (deprecated) vibrance

Technical information

description
saturate and reduce the lightness of the most saturated pixels to make the colors more vivid.

purpose
creative.

input
linear or non-linear, Lab, display-referred.

processing
non-linear, Lab.

output
non-linear, Lab, display-referred.

Please note that this module is deprecated from darktable 3.6 and should no longer be used for new edits. Please use the vibrance control in the *color balance rgb* module instead.

Saturate and reduce the lightness of the most saturated pixels to make the colors more vivid.

module controls

vibrance
The amount of vibrance to apply to the image.

8.2.87. (deprecated) zone system

Please note that this module is deprecated from darktable 3.4 and should no longer be used for new edits. Please use the *tone equalizer* or multiple instances of the *exposure* module with parametric masks to narrow down on a zone.

Adjust the lightness of an image using Ansel Adams' zone system .

The lightness of the image (the L channel in Lab color space) is divided into a number of zones ranging from pure black to pure white. These zones are displayed in a zonebar beneath a preview image. The number of zones can be changed by mouse-scrolling on that bar (default 10 zones).



The zonebar is split horizontally with the lower part showing the zones of the module's input and the upper part the zones of the module's output. In its default state both parts are fully aligned. While the output zones are static you can left click and drag a control point in the input zones to modify the mapping between input and output.

Shifting a control point proportionally expands the zones on one side and compresses the zones on the other side of that point. All pre-existing control points stay in place, effectively preventing changes to the zones beyond the next control point. Use right click to remove a control point.

The preview image above the zonebar shows how the image's zones are broken down. When hovering above a zone, that zone – either from input or output – is highlighted in yellow on the preview image.

8.3. utility modules

8.3.1. darkroom

8.3.1.1. clipping warning

Highlight areas of the image that may exhibit luminance or gamut clipping.

When an image is sent to a display device, each pixel is normally represented as a set of 3 numbers, representing the intensity of the red, green and blue primary colors in the output color space. Because the output color space is usually closely related to hardware with physical limitations, there is a maximum permitted value for the [R,G,B] channels, representing the maximum available intensity for that color space. Similarly, there is also a minimum value below which pixel values will be mapped to zero. When we try to convert from a larger color space to the final output color space, any values exceeding this maximum will be clamped to the maximum value, and any values below the minimum will be clamped to zero. This process is called “clipping” and it will lead to lost detail, or “incorrect” colors for any pixels with clipped channels.

Click the  icon to enable the clipping warning.

There are two ways in which a pixel might become clipped when represented in the output color space.

- *luminance clipping*: This can occur when a pixel is too bright to be represented in the output color space. The pixel luminance is calculated as a weighted average of the [R,G,B] channels. If this average exceeds the maximum allowed value, it is an indication of over-exposure. The overall luminance of a pixel can also be too dark to be represented by an [R,G,B] value in the output color space, in which case it will simply be shown as black. We normally deal with luminance clipping by carefully adjusting tone mappings and exposure levels.

- *gamut clipping*: The output color space defines a set of primary colors that, mixed together in certain ratios, produce the final output color. However, there are only so many colors that can be produced by mixing together a combination of those three primary colors. Highly saturated colors in particular can be difficult to represent, especially for pixels that are very bright or very dark. If there is no set of positive [R,G,B] values that can represent a given color at a given level of brightness, we say that the color is “out of gamut”, and we need to settle for another color instead that *can* be represented by permitted [R,G,B] values within the color space. We can handle gamut clipping by being careful not to over-saturate colors in the highlights and shadows, and possibly by using some color grading/color mapping techniques.

The “clipping warning” module is used to highlight those pixels that cannot be accurately represented in the output color space, either due to luminance or gamut clipping. Prior to darktable 3.4, the clipping highlighted any pixels that exceeded the maximum allowed value on any of the [R,G,B] channels, or that had been completely crushed to black. From darktable 3.4 onwards, the clipping warning indicator has some additional modes to help you to differentiate between luminance and gamut clipping, so that you can make better decisions about how to address any issues.

As the clipping warning runs at the end of the preview pixelpipe, it receives data in display color space then converts it to histogram color space. If you are using a display color space that is not “well behaved” (this is common for a device profile), then colors that are outside of the gamut of the display profile will clip or distort.

The clipping warning module, described here, warns you about clipping caused by image processing and the limitations of the output color space. It should not be confused with the following similar tools:

- The raw overexposed warning indicates where pixels in the original raw file are clipped due to physical limitations in the dynamic range of the camera sensor. This module highlights information that was permanently lost at the point of image capture, and you need to deal with it as best you can using highlight recovery techniques.

- The gamut check module also provides information about clipping arising from image processing. It is based on the external littleCMS library, and is more or less equivalent to the *full gamut* mode in the clipping warning module. The downsides of the gamut check module are that it doesn’t allow you to distinguish between clipping caused by luminance and gamut mapping, and it is much slower than the clipping warning indicator.

module controls

Right-click on the clipping icon to show the following options:

clipping preview mode

Choose the type of clipping that you want to highlight:

- *any RGB channel*: Provides an over-clipping indication if any one of the three [R,G,B] channels exceeds the maximum permitted value for the histogram color space, or an under-clipping indication if the three [R,G,B] channels are too dark and are all forced to black. This was the default mode prior to darktable version 3.4.
- *luminance only*: Indicates any pixels that are clipped because their luminance falls outside of the range set in the "upper threshold" and "lower threshold" sliders. If this happens, it generally means that tone mapping or exposure settings have been poorly set
- *saturation only*: Indicates where over-saturated colors have pushed one or more of the [R,G,B] channels towards a value outside the permitted range of the histogram color space, even though the overall luminance of the pixel may lie within acceptable limits. This means that the pixel's color is impossible to represent in the histogram color space, and can arise from poorly set gamut mapping or saturation settings,
- *full gamut*: Shows a combination of the three previous options. This is the default mode from darktable 3.4 onwards, and it gives the most complete indication of potentially problematic pixels.

color scheme

By default, the indicator marks pixels with *red* where the upper threshold is exceeded (over-clipping) and with *blue* where the lower threshold is breached (under-clipping). This color scheme can be changed to *black & white* or *purple & green* for "over & under" indicators, which may be useful to improve visibility for some images.

lower threshold

Expressed in EV relative to the white point (which is nominally EV = 0). If the [R,G,B] channels all fall below this value, an under-clipping indicator is shown warning that the pixel may be end up being crushed to black. Use the following reference to set this threshold, depending on your intended output medium:

- *8-bit sRGB* clips blacks at –12.69 EV
- *8-bit Adobe RGB* clips blacks at –19.79 EV
- *16-bit sRGB* clips blacks at –20.69 EV
- *fine art matte prints* typically produce black at –5.30 EV
- *color glossy prints* typically produce black at –8.00 EV
- *black & white glossy prints* typically produce black at –9.00 EV

upper threshold

How close a pixel should be to the upper limit before being flagged by the clipping warning, expressed as a percentage (default 98%). In the case of gamut checks, this controls how close the saturation of the pixel is allowed to get to the limits of the color space's gamut before a clipping indication is flagged.

Note: This clipping indication may not be optimal if you want to inspect *where* data is clipped, in which case use the highlight reconstruction clipping mask .

8.3.1.2. color assessment

Assess colors and brightness in your image under neutral viewing conditions to avoid perceptual issues.

When developing an image, the way we perceive brightness, contrast and saturation is influenced by the surrounding ambient conditions. If an image is displayed against a dark background, this can have a number of adverse effects on our perception of that image:

- Exaggeration of the perceived exposure makes the image seems brighter than it really is. This is nicely illustrated by the Adelson checkerboard shadow effect .
- A decrease in the perceived saturation in the image makes the colors seem less rich than they really are (the Hunt effect).
- A decrease in the perceived contrast in the image makes the tones seem flatter than they really are (Bartleson-Breneman effect 3)

The end result is that the final image can end up being too dark and overly-processed in terms of contrast and color saturation. To avoid this, the *color assessment* module in the darktroom places a grey and white frame around the image to help the user better assess the colors in the image



When the color assessment button is selected in the bottom panel, the image is zoomed out so that a thick mid-gray border appears around the image to act as a reference against which to compare the image's tones. A thinner white border is placed immediately around the image to give the eyes a basis for comparison when looking at parts of the image that are meant to be a bright white.

Although the color assessment mode provides a mid-gray surrounding to the image, it is recommended that you also set your user interface (in preferences > general) to one of the "grey" themes. These themes are designed to provide a user interface that is close to middle gray (it is actually slightly darker to allow better contrast with the text in the user interface). When one of these themes is used together with the color assessment mode, this will help to avoid the above perception issues.

Color assessment mode can also be toggled by pressing Ctrl+B.

The total size of the grey and white frame and the relative size of the white frame may be adjusted by right-clicking . This will bring up a control panel with two slider controls:

total border width relative to screen
: Adjust the size of the combined grey and white border. Increase to grow the total border (and shrink the image). Decrease shrink the border (and grow the image).

white border ratio
: Adjust the white border relative to the screen. Increase this slider to enlarge the the white border. Decrease to shrink the white border. The image size will remain unaffected.

8.3.1.3. duplicate manager

View and create multiple versions of the current image. Each version can be edited independently without affecting other versions – all versions use the same underlying image file, but the editing history of each version is stored in its own independent XMP sidecar file.

The duplicate manager lists each version of the current darkroom image along with its preview thumbnail. Hold down the left mouse button on a thumbnail to temporarily show that version in the center view. Double-click to switch to that version and edit it.

Buttons to manage the duplicates are shown to the right of each preview thumbnail:



duplicate with history

Create a new duplicate with the same history stack (only shown if the selected duplicate has been edited).



duplicate original

Create a ‘virgin’ duplicate of this image without any development.



delete

Delete the duplicate’s sidecar files (only shown if there are multiple versions in the list).

A version number is displayed to the right of each thumbnail. Click in the area below the version number to enter a description against that version of the image. This description is stored in the *version name* metadata tag, which can also be edited within the metadata editor module in the lighttable view.

8.3.1.4. gamut check

Highlight areas of the image that may exhibit gamut clipping.

Click the  icon to activate the gamut check display mode for your image. This function highlights, in cyan, all pixels that are out of gamut with respect to the selected softproof profile. You can also activate gamut check with the keyboard shortcut Ctrl+G. A message “gamut check” on the bottom left of your image tells you that you are in gamut check display mode. Gamut check and soft proof are mutually exclusive modes.

Right-click on the icon to open a dialog with configuration parameters – these are the same as for the soft proof option.

You might also like to consider using the clipping warning , which also provides under- and over-exposure warnings as well as a gamut check similar to that offered by this module.

8.3.1.5. global color picker

Take color samples from the current darkroom image, display their values in multiple ways and compare colors from different locations.

The color picker is activated by pressing the picker icon. The module’s parameters will remain in effect until you leave the darkroom mode.

Besides the global color picker described here, many darktable modules (e.g. *tone curve*) also contain local pickers which are used to set individual module parameters. You should be aware that these two forms of picker do not always work in the same color space. The global color picker works in the histogram color space and takes its samples after the complete pixelpipe has been processed. Local pickers run in the color space of the module in which they are activated and reflect the input or output data of that module within the pixelpipe.

You can right-click on the sampled color values to copy them to the clipboard.

As the global color picker runs at the end of the preview pixelpipe, it receives data in display color space then converts it to histogram color space. If you are using a display color space which is not “well behaved” (this is common for a device profile), then colors that are outside of the gamut of the display profile will clip or distort.

Hover over any of the color values to show a tooltip containing more detailed information about the picked color or live color sample. This information includes RGB and Lab values as well as an approximate color name. An attempt is also made to detect skin tones and provide an appropriate description. Skin tone detection needs proper Lightness scaling (44 to 48% for African and 58 to 64% for all others) and neutral white balance.

module controls

point/area mode

The global color picker can be activated in point or area mode with the picker icon. In point mode only a small spot under your cursor is taken as a sample. In area mode darktable samples the area within a drawn rectangle.

mean/min/max

If samples are taken in area mode, darktable will calculate mean, minimum and maximum color channel values. This combobox allows you to select which of those are displayed. For obvious statistical reasons mean, min and max are identical for the single sample of point mode.

color swatch / color values

A color swatch representing the sampled point or area is displayed alongside numerical values. Clicking on the swatch will toggle on/off a much larger swatch for easier viewing.

The global color picker works in display RGB color space, though you may choose (by selecting from the drop-down) to translate these numerical values into another color space. Beware that values in other color spaces are approximated here – depending on the display color profile there may be some deviations from the actual values.

live samples

The sampled colors (in either area or point mode) can be stored as live samples by pressing the “add” button. A color swatch and numerical values will be shown for each stored sample. You can change the numerical value (mean, min, max) and color space to display.

Newly created live samples are not locked. If you change your image processing those changes will be reflected in the live samples. This can be used to see how an altered parameter affects different parts of the image. Clicking on a live sample’s color swatch locks it and a lock symbol is displayed. Further image changes will then no longer affect the sample. You can use this to compare two live samples by locking just one of them, thus providing a before and after comparison.

If you hover the mouse over the “delete” button of one of the live sample entries, the selected region for that sample will be highlighted in the image preview.

To update a live sample, first right-click on it to copy the selected live sample color to the primary sample – an icon  is shown on the live sample to indicate that editing is in progress. Next, make any desired new color selection on the primary sample – it is not necessary to click the picker to start the new color selection, but you can still use the picker to alter the picker mode (point or area). Note that using the picker to select the existing (point/area) picker mode will cancel the edit. Finally, to save the new color selection, right-click on *any* live sample to update that sample (it does not have to be the originally selected live sample). The save icon will be removed once the update has been made.

display samples on image/vectorscope

When this checkbox is ticked, live sample locations are visually indicated on the image and the vectorscope view of the scopes module.

restrict scope to selection

When this checkbox is ticked, only the values of the selected area or point are taken into account by the regular and waveform views of the scopes module. This allows you to see what tonal values are present in the selected area. When using a picker in a processing module, this option restricts the scope to the picked area from the processing module instead of the global color picker.

8.3.1.6. guides & overlays

A number of commonly-used compositional guides can be overlaid on your image while you are editing. These can be enabled either globally (all the time) or locally (when certain modules are active).

Other darkroom functionality also draws colored overlay lines on the image (for example, drawn masks). An option is also provided to change the color of those overlays (see below).

global guides

Left-click the  icon in the bottom bar to globally display guide overlays. The overlays will remain switched on until you click the button a second time to switch them off.

Right-click the icon to show the settings dialog (see below).

local guides

A more common use is to switch the guides on only when a specific module is activated. The following control is added by default to all modules that crop/distort the image (currently crop , crop and rotate , orientation , framing , liquify , lens correction and rotate and perspective):



Tick the box to show guide overlays whenever the module is active. Click the icon on the right to show the settings dialog (see below).

global guide overlay settings

Please note that, while you can choose to switch guide overlays on and off either globally or locally, the following settings are stored globally and cannot be set independently for each module.

type
The type of compositional guide lines to display.

flip
Some guides are asymmetrical. This option allows you to flip such guides horizontally/vertically.

horizontal lines, vertical lines, subdivisions
When the *grid* overlay type is selected, set the parameters of the grid.

overlay color
The color of the overlay lines. Note that this impacts *any* lines that are drawn directly over the image, for example, drawn masks.

contrast
The contrast between the lightest and darkest parts of any overlays – usually, the contrast between the “on” and “off” parts of dashed lines.

8.3.1.7. high quality processing

Process the image in the darkroom view using “high quality mode”, so that the displayed image reflects the appearance of a full-sized export.

Activate this mode by clicking the  icon at the bottom of the darkroom view.

Note that while this mode provides the most accurate representation of the processed image, it also results in significant performance degradation. Please see the types of pixelpipe section for a full discussion.

8.3.1.8. history stack

View and modify the history stack of the current darkroom image.

This module lists every change of state (activate/de-activate/move/change parameters) for all processing modules that have been modified for the current image. Select a point in the stack to return to that point in the development history of the image. Shift+click an item in the history stack to expand that module in the right-hand module panel without changing the current edit.

Caution: If you select a module in the history stack and then make further modifications to the image, all edits above the currently selected step will be discarded. It is easy to lose development work on an image in this way! You can usually use Ctrl+Z to undo such changes.

It is safe to quit the program, leave the darkroom mode, or switch to another image after having selected some earlier state in the history stack module. When returning to that image you will find the history stack panel in the state where you left it.

Hover the mouse over an item in the history stack to show a tooltip with details of all changes that were made in that module compared to its previous or default state. This can help you to track down adjustments that were made unintentionally.

Click “compress history stack” to generate the shortest history stack that reproduces the current image. This causes all edits above the currently-selected step to be discarded. If any module appears multiple times in the remainder of the stack, these will be *compressed* into a single step in the history.

Click “compress history stack” while holding the Ctrl key to truncate the history stack without compressing it i.e. discard all modules above the currently selected one but leaving the remainder of the history stack unchanged.

Click the “reset parameters” button in the module header to discard the entire history stack and reactivate any default modules. This can also be achieved by selecting the “original image” history item and clicking “compress history stack”.

The button to the right of the “compress history stack” button allows you to create a new style from the history stack of the current image, which can then be applied to other images. Use the first line of the popup dialog window to name your style and the second to add a searchable description. You will be prompted to choose which of the modules from the current history stack to include in the style.

Once created, styles can be managed and applied to other images with the lighttable’s styles module. You can also assign shortcut keys to your styles (see preferences > shortcuts) and apply the associated style to all selected images by pressing the shortcut key whenever you are in the lighttable or darkroom view.

8.3.1.9. image information line

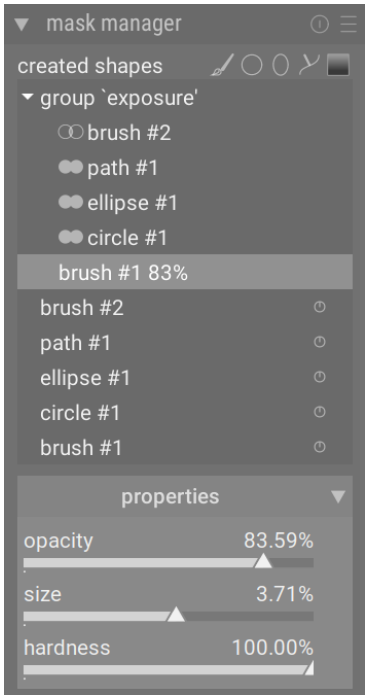
Display information about the current image in a darkroom panel.

The contents of this line can be set in preferences > darkroom . See the variables section for information about the variables that can be used here. You can also insert a newline with `$(NL)`.

The image information line can be displayed in the top, bottom, left or right panel depending on an additional configuration item in preferences > darkroom .

8.3.1.10. mask manager

Manage all drawn masks and shapes for the current image.



This module can be used to create, rename, edit and delete shapes. You can alter the shapes in a drawn mask, group shapes together, and combine them using set operators.

The top line of the mask manager panel contains buttons that can be used to create new shapes. These are the same as in the drawn mask interface of the processing modules.

The panel below these buttons displays a list of all masks and individual shapes defined for the current image.

Groups of shapes forming a drawn mask are displayed with a headline in the form “group <module_name>” to indicate the module in which they are used, with the constituent shapes listed below. The list of mask groups is followed by a list of all individual shapes that have been generated in the context of the current image (which also repeats the shapes that are part of a group). If a shape is in use by any drawn masks this is indicated by a symbol to the right of the shape name.

shapes

By default each shape receives an automatically generated name, consisting of the shape type (“brush”, “circle”, “ellipse”, “path”, “gradient”) and an automatically-incremented integer. You can rename a shape by double-clicking on its current name. It is a good habit to give shapes and groups meaningful names, especially if you intend to reuse the same selection in different masks.

If any shape has a reduced opacity this will be shown to the right of the shape name. An icon representing the chosen set operator (mode) is shown to the left (except for the first shape in a group, which has no prior shape to be combined with).

Click on a shape name to show the selected shape on the image canvas with all of its controls, allowing you to edit the properties of just that shape. This is especially useful where there are many overlapping shapes within a mask, making it difficult to select the right one with the mouse. Similarly if you select a shape on-screen from within the mask controls of a processing module, that shape will also be selected in the mask manager.

Right-click on a shape name to show a menu containing additional options (explained below).

Note: darktable retains all shapes that have ever been defined for the current image unless you explicitly remove them. If you choose to include develop history when exporting an image, all defined shapes will be exported with the image. Beware that if the list of shapes is very long the space required to store those shapes might exceed the size limit of certain file formats. This can cause XMP tag creation to fail during export.

---

masks & groups

Drawn masks are constructed by adding a group of shapes to the image in the order that they are listed (from bottom to top – the same order as processing modules are displayed). Each shape adjusts the existing mask using one of five logical set operators (see below). Because order is important it is also possible to move shapes up and down the list via the right-click menu.

Click on the name of a group in the mask manager to expand that group, showing a list of its constituent shapes – the corresponding shapes will be shown on the center image. Similarly if you choose to show a drawn mask from within a processing module, the corresponding group will be expanded within the mask manager.

Right-click on a group name to display a menu with options to add new or existing shapes to the group, or to clean up unused shapes. You can also choose to delete the group.

Right-click on any of the constituent shapes to control how that shape contributes to the overall group mask as follows:

remove from group
Remove the shape from the current mask.

use inverted shape
Invert the polarity of the selected shape.

mode
Choose how this shape will be combined with the preceding mask by selecting one of the five set operators defined below. The currently-selected mode is highlighted with a check mark.

move up/down
Move the shape up or down the list.

You can also create your own groups using existing shapes by selecting the shapes you wish to group, right-clicking them and choosing “group the forms”.

properties

Expand the *properties* section to change the properties appropriate for the selected shape(s) (some or all of: *opacity*, *size*, *shrink or grow*, *rotation*, *feather*, *hardness*).

If a *group* is selected, the soft limits of the sliders are automatically adjusted in an attempt to prevent irreversible distortions (where some of the shapes are clamped at their extreme values but others can still be adjusted, so that reversing the move does not return all shapes to their previous configuration). However, like any soft limits, these can be overridden (forced) if you are happy to accept the consequences.

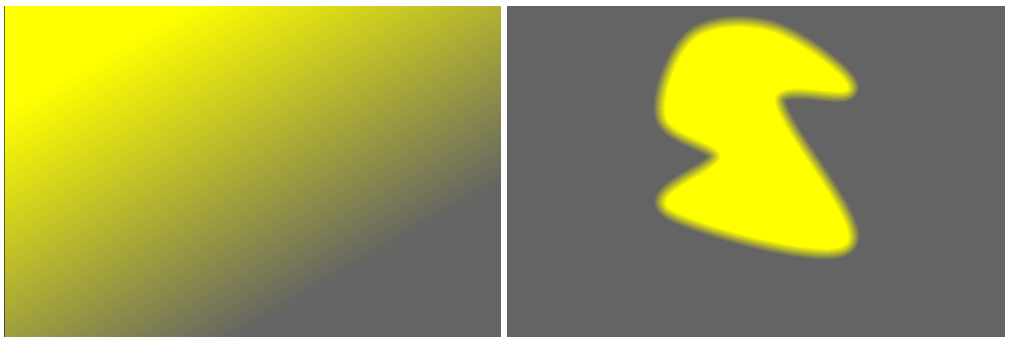
If a pen (e.g. Wacom) is detected, some additional options are also displayed to control how it is used by darktable:

pressure
Controls how the pressure reading of a graphics tablet impacts newly generated drawn mask brush strokes. You can control the brush width, hardness and opacity. “Absolute” control means that the pressure reading directly defines the attribute with a value between 0% and 100%. “Relative” means that the pressure reading adjusts the attribute between zero and the pre-defined default value (default off).

smoothing
Sets the level for smoothing of drawn mask brush strokes. Stronger smoothing leads to fewer nodes and easier editing at the expense of lower accuracy.

set operators (modes)

Set operators are used to define how grouped shapes are combined. In the following examples (with the exception of “sum”) we will use a mask that combines a gradient followed by a path, to demonstrate the effect of each set operator when applied to the path shape:



As a convention in the following explanations we say that a pixel is “selected” in a mask or shape if it has an opacity greater than zero.

sum (default for brush shapes)
The shape adds to the existing mask by increasing its opacity by the opacity of the drawn shape. This allows multiple shapes (e.g. brush strokes) with low opacity to be layered on top of one another to increase the strength of the overall mask (e.g. for dodge and burn operations). The resulting opacity of a given pixel is the sum of the opacity of the individual shapes that intersect with that pixel, up to a maximum of 100%.

union (default for non-brush shapes)
The shape adds to the existing mask in such a way that the resulting mask contains the pixels that are *either* selected in the existing mask *or* in the added shape. In overlapping areas the maximum value is taken:



intersection

The shape adds to the existing mask in such a way that the resulting mask contains only pixels that are selected in *both* the existing mask *and* the added shape. In overlapping areas the minimum value is used. In the given example we use this operator to “imprint” the path with a gradient:



difference

In the non-overlapping area the existing mask is unchanged. In the resulting mask, pixels are selected only if they are selected in the existing mask but *not* in the added shape. This set operator can be chosen if you want to “cut out” a region from within an existing selection:



exclusion

The resulting mask selects all pixels that are selected in the existing mask and not in the added shape or vice versa. This is equivalent to an “exclusive or”:



8.3.1.11. module order

Change the order of the processing modules in the darkroom using presets.

When processing an image, the active modules are applied in a specific order, which is shown in the right-hand panel of the darkroom view. This module provides information about the current ordering of the processing modules in the pixelpipe . The name of the currently-selected preset is shown in the module header (or “custom” if the user has manually modified the order). The following presets are available for selection.

v5.0 RAW
This is the default module order for scene-referred RAW development workflow in darktable 5.0 and later.

v5.0 JPEG
This is the default module order for JPEG development in darktable 5.0 and later.

v3.0 RAW
This is the default module order for scene-referred RAW development workflow between darktable 3.0 and darktable 4.8.

v3.0 JPEG
This is the default module order for JPEG development between darktable 3.0 and darktable 4.8.

legacy
This module order is used for the legacy display-referred workflow. You will also see this order displayed for any images you previously edited in darktable prior to version 3.0.

custom
The user has manually modified the module order.

Note: changing the order of modules in the pixelpipe is not a cosmetic change to the presentation of the GUI – it has real consequences to how the image will be processed. Please do not change the order of the modules unless you have a specific technical reason and understand why this is required from an image processing perspective.

For more information about changing module order please refer to the pixelpipe & module order .

8.3.1.12. navigation

Zoom and pan the current image.

The navigation panel (displayed on the top left hand side of the darkroom) shows a full preview of the current image with a rectangle showing the area that is currently visible in the central panel. You can interact with this module using your mouse in the same way as you can interact with the main image (scroll to zoom in/out, middle-click to cycle through zoom levels, click+drag to pan). Alternatively, the current zoom scale is displayed in a combobox to the bottom-right of the preview – click to choose from a number of common zoom levels or type a number to set the zoom level manually.

The navigation panel can be toggled on/off with a keyboard shortcut (Ctrl+Shift+N by default) or in the left panel’s right-click menu.

8.3.1.13. raw overexposed warning

Highlight areas of the image where color channels of the raw input file are clipped.

Clipped color channels imply an overexposed image with loss of information in the affected areas. Some of this information may be recoverable using the *highlight reconstruction* , *color reconstruction* or *filmic rgb* modules.

Click on the  icon to show/hide the warning overlay. Right-click on the icon to open a dialog containing the following configuration parameters.

mode
Choose how to mark clipped areas:
- *mark with CFA color*: Display a pattern of the respective primary colors (red, green, and blue) to indicate which color channels are clipped.
- *mark with solid color*: Mark clipped areas with a user defined solid color (see below) independent of the affected color channels.
- *false color*: Set clipped color channels to zero in the affected areas.

color scheme

Choose the solid color for the *mark with solid color* mode.

clipping threshold

Set the threshold defining what values are considered to be overexposed. You can safely leave this slider at its default value of 1.0 (white level) in most cases.

8.3.1.14. snapshots

Store development snapshots and compare with the current edit.

Snapshots can be taken at any point in the development process and later overlaid onto the current center view or shown side-by-side. This allows you to undertake a comparison (by default left=snapshot, right=active edit) while you are tuning parameters of a module. It can also be combined with the history stack module to compare a snapshot against different stages of development.

To take a snapshot, click on the *take snapshot* button. Above the button you will see a list of the snapshots that have been taken for this editing session. The name of each snapshot reflects the name of the module selected in the history stack and its position at the time the snapshot was taken. You can choose your own label for a snapshot by Ctrl+clicking on the snapshot name – this will be displayed after the module name, separated with a bullet.

Click on the name of a snapshot to show it.

At all times, an arrow containing the letter “S” is displayed to indicate which side of the image is the snapshot and which is the current edit.

You can pan or zoom the image while using the snapshot view using your keyboard or mouse. To pan with the mouse you will need to hold the “a” key while clicking and dragging (this key can be configured in preferences > shortcuts > views > darkroom > force pan & zoom with mouse).

Click on the name of the snapshot again to disable the overlay and return to your editing session. Click the module’s reset button to remove all existing snapshots.

You can restore a snapshot to the current history stack (which will replace the history stack of the current image with the stack taken from that snapshot) by clicking the  icon to the right of the snapshot name.

overlay view

By default, selecting a snapshot enables a split view between the saved snapshot image and the current state of the processed image. You can control the position of the split-line by clicking and dragging the line with your mouse. If you hover over the split-line with your mouse, a small rotation icon will appear on the center of the line. Click this icon to change between vertical and horizontal split view – the positions of the snapshot and current image will be rotated anti-clockwise allowing you to choose whether they appear to the top, bottom, left or right of the screen.

side-by-side view

You can toggle a side-by-side snapshot view by clicking on the  icon to the right of the *take snapshot* button.

Selecting a snapshot then splits the center viewport into two equal panels, one showing the snapshot and one showing the current state of the processed image. The dividing line can not be moved and any click without “a” rotates the layout – switching between showing the snapshot to the left, below, right, or above the current image.

Note: Snapshots are retained for the duration of your darktable session. This means that you can also use snapshots to compare with another image. Just navigate to that image and enable the snapshot view as normal. When you do this, an arrow appears in the snapshot name to indicate it was not taken from the current image – hover over the name with your mouse to see the originating image name in a tooltip.

8.3.1.15. soft proof

View your image rendered using a selected color profile.

Click the  icon to activate the soft proof display mode on your image. This allows you to preview your image rendered using a printer profile to see how colors will end up on the final print. You can also activate soft proof with the keyboard shortcut Ctrl+S. A message “soft proof" on the bottom left of your image tells you that you are in soft proof display mode.

Right-click on the icon to open a dialog with the following configuration parameters. For each of these parameters, the list of available profiles is read from `$DARKTABLE/share/darktable/color/out` (where `$DARKTABLE` represents darktable’s installation directory) and `color/out` in the darktable configuration directory . Note that these `color/out` directories are not created by the darktable install; if you need to use one, you must create it yourself.

display profile
: Set the color profile for the display. The option “system display profile” is the preferred setting when working with a calibrated display. The profile is taken either from your system’s color manager or from your X display server. The method darktable uses to detect your system display profile can be changed in preferences > miscellaneous . For more information see the display profile section.

preview display profile
: Set the color profile for the preview image if you are using a second window.

intent
: Set the rendering intent for your display – only available if rendering with LittleCMS2 is activated. See rendering intent for a list of available options. This option appears twice – once for the “display profile” and once for the “preview display profile”.

softproof profile
: Set the color profile for soft proofing. Typically these profiles are supplied by your printer or generated during printer profiling.

histogram profile
: Set the color profile of the histogram. None of the available options are ideal, however, “system display profile” is probably the *least bad* setting, since all other profiles are derived from the display color space and at least the values will conform to what you see on screen.

8.3.2. lighttable

8.3.2.1. (deprecated) lua scripts installer

after 5.6

Lua scripts are bundled with darktable and do not need to be installed anymore.

before 5.6

As described in lua scripts , darktable provides powerful scripting capabilities to extend its functionality and integrate with other software. Many such scripts are already available. The Lua script installer module helps to install them without the need for manual configuration. The first time it is run, instructions are displayed in the module.

For the module to be able to install the scripts, you will need to have git installed and available on your path. You can get it from git-scm.com .

The officially supported scripts are documented here .

The module can be disabled with an option in preferences > lua options .

8.3.2.2. actions on selection

Perform actions on images that have been selected in the lighttable view.

module controls

The module controls are spread over two tabs for manipulating the image files and the related metadata.

images tab

remove

Remove the selected images from the darktable library without deleting them. Removed images can no longer be viewed or edited within darktable but the image files remain on the filesystem along with any XMP and DTDATA sidecar files. As darktable keeps the XMP files up-to-date with your latest development history, you can fully restore your work later by re-importing the images.

delete / delete (trash)

Remove the selected images from the darktable library and remove any associated XMP and DTDATA sidecar files from the filesystem. If no duplicates of a removed image remain in the darktable library, the image file itself is also deleted. You can control whether this action irrevocably deletes the files or attempts to put them into your system's trash bin with a configuration item in preferences > security . A second configuration item in the same tab allows you to control whether or not to be prompted before deleting images.

move

Physically move selected images (the image files plus all associated XMP and DTDATA sidecars) to another folder on the filesystem. If an image with the given filename already exists in the target folder the source image will not be moved. After the move the target folder becomes the selected collection if the source folder is empty. If a collection is manually selected during a move, then darktable will not change the selection.

copy

Physically copy selected images (the image file plus all associated XMP and DTDATA sidecars) to another folder on the filesystem. If an image with the given filename already exists in the target folder it will not be overwritten – instead a new duplicate will be generated with the same history stack as the source image. After the copy the target folder becomes the selected collection. If a collection is manually selected during a copy, then darktable will not change the selection.

create hdr

Create a high dynamic range image from the selected images, and add the result to the library as a new image in DNG format. Image alignment can be done by darktable, and this is controlled by a set of preferences . You can also generate HDRs with programs like Luminance HDR, and later import them into darktable for further processing. Note that darktable can only create HDR images from raw files.

duplicate

Create duplicates of the selected images within darktable. Duplicate images share the same image file, but each duplicate has its own XMP and DTDATA sidecar files and a separate entry in darktable's library database. This allows you to test different edits on the same image.

rotation

Perform a clockwise or counter-clockwise rotation on the selected images. A third button can be used to reset the image rotation to the value stored in the image's Exif data. This feature is directly linked to the *orientation* processing module – adjustments made here are automatically converted into a history stack item for that module.

copy locally

Create local copies of the selected images on the local drive. These copies will then be used when the original images are not accessible (see local copies).

resync local copy

Synchronize the XMP sidecar of the local copy of each selected image with the copy in external storage, and remove the local copies. Note that if a local copy has been modified and the external storage is not accessible the local copy will not be deleted (see local copies).

group

Create a new group from the selected images (see image grouping).

ungroup

Remove the selected images from their group (see image grouping).

metadata tab

metadata type checkboxes

Choose the types of metadata (ratings, tags, metadata, colors, geo tags) that you want to operate on.

copy

Copy the chosen types of metadata from the selected image onto the clipboard. If you have more than one image selected, or no images selected, then this button is unavailable.

paste

Paste any metadata in the clipboard onto the selected images.

clear

Clear the chosen types of metadata from the selected images.

mode
When pasting metadata onto images, this option controls whether the metadata on the clipboard should be merged with the existing metadata (*merge*), or should replace it entirely (*overwrite*).

refresh exif
Refresh the Exif data from the source file. Warning: this may overwrite some tags and metadata that you have altered in darktable (such as star ratings).

monochrome
Flag the image as monochrome, meaning that it will receive any monochrome-specific workflow treatment that is offered by the processing modules. Refer to the developing monochrome images section for more details.

color
Remove the monochrome flag from the image so that it will receive the default workflow treatment that is normally used when developing color photos.

8.3.2.3. history stack

Manipulate the history stack of one or more selected images.

module controls

selective copy…
Copy parts of the history stack from the selected image.

copy
Copy the complete history stack from the selected image.

selective paste…
Paste parts of a copied history stack onto all selected images.

paste
Paste all items of a copied history stack onto all selected images.

Note: These are only brief descriptions, for an extensive explanation see copy & paste .

---

compress history
Compress the history stack of the selected image. If any module appears multiple times in the history stack, these occurrences will be *compressed* into a single step in the history. *Beware: this action can not be undone!*

discard history
Physically delete the history stack of the selected images. *Beware: this action can not be undone!*

load sidecar file
Open a dialog box which allows you to import the history stack from a selected XMP file. The imported history stack is used to *completely replace* the current history stack(s) of the selected image(s). *Caution: this operation can not be undone!*

Images that were exported by darktable typically contain the full history stack if the file format supports embedded metadata (see the export module for details of this feature and its limitations). You can load an exported image as a sidecar file in the same way as you can with an XMP file. This feature allows you to recover all parameter settings if you have accidentally lost or overwritten the XMP file. All you need is the source image, typically a raw, and the exported file.

write sidecar files
Write XMP sidecar files for all selected images. The filename is generated by appending ".xmp" to the name of the underlying input file.

By default darktable generates and updates sidecar files automatically whenever you work on an image and change the history stack. You can disable automatic sidecar file generation in preferences > storage . This can be useful when you are running multiple versions of darktable (so that edits in each version do not conflict with one another) however, in general, disabling this feature is not recommended.

8.3.2.4. import

Add images to the darktable library, optionally copying them from another location on the filesystem or from a connected camera.

See supported file formats for more information.

module controls

The following buttons are shown in the module's UI by default:

add to library
Add existing images to the darktable library without copying or moving files. If you only add a single image to the library it will be automatically loaded in the darkroom.

copy & import
Create copies of images from the filesystem and then add those copies to the darktable library.

When a camera is detected, a new section will appear in the module for that device. If you hover your mouse over the camera label, a tooltip will display information about the camera (model, firmware version etc.)

Depending on the capabilities of the camera, the following additional buttons may be displayed:

mount camera
Mount the camera for exclusive use by darktable. This button only appears if the camera is not currently mounted and is not locked by another process.

copy & import from camera
Create copies of images from the connected camera and then add those images to the darktable library. This button only appears if the camera is currently mounted.

tethered shoot
Open the tethering view so that you can take images with your connected camera using darktable. This button only appears if the camera is currently mounted.

unmount camera
Unmount the camera and release it for use by other applications. This button only appears if the camera is currently mounted.

module parameters

Click on the "parameters" label or the expander button beside it to display the following additional options.

All parameters are retained from one session to the next and can be saved as module presets.

ignore exif rating
Check this option to ignore any rating stored within the image's Exif data and instead use a hard-coded value (below).

initial rating
Initial star rating (from 0 to 5) to use for all newly-imported images (default 1).

apply metadata
Check this option to automatically set metadata fields and/or tags at import time (see below).

metadata
If the "apply metadata" box is checked, a list of the *visible* metadata fields is presented for completion (see metadata editor for details). Any populated strings are automatically added to the imported images. You can also choose from any presets saved within the metadata editor module.

Double-click on a label to reset the corresponding field. Double-click on the "metadata presets" label to reset all fields.

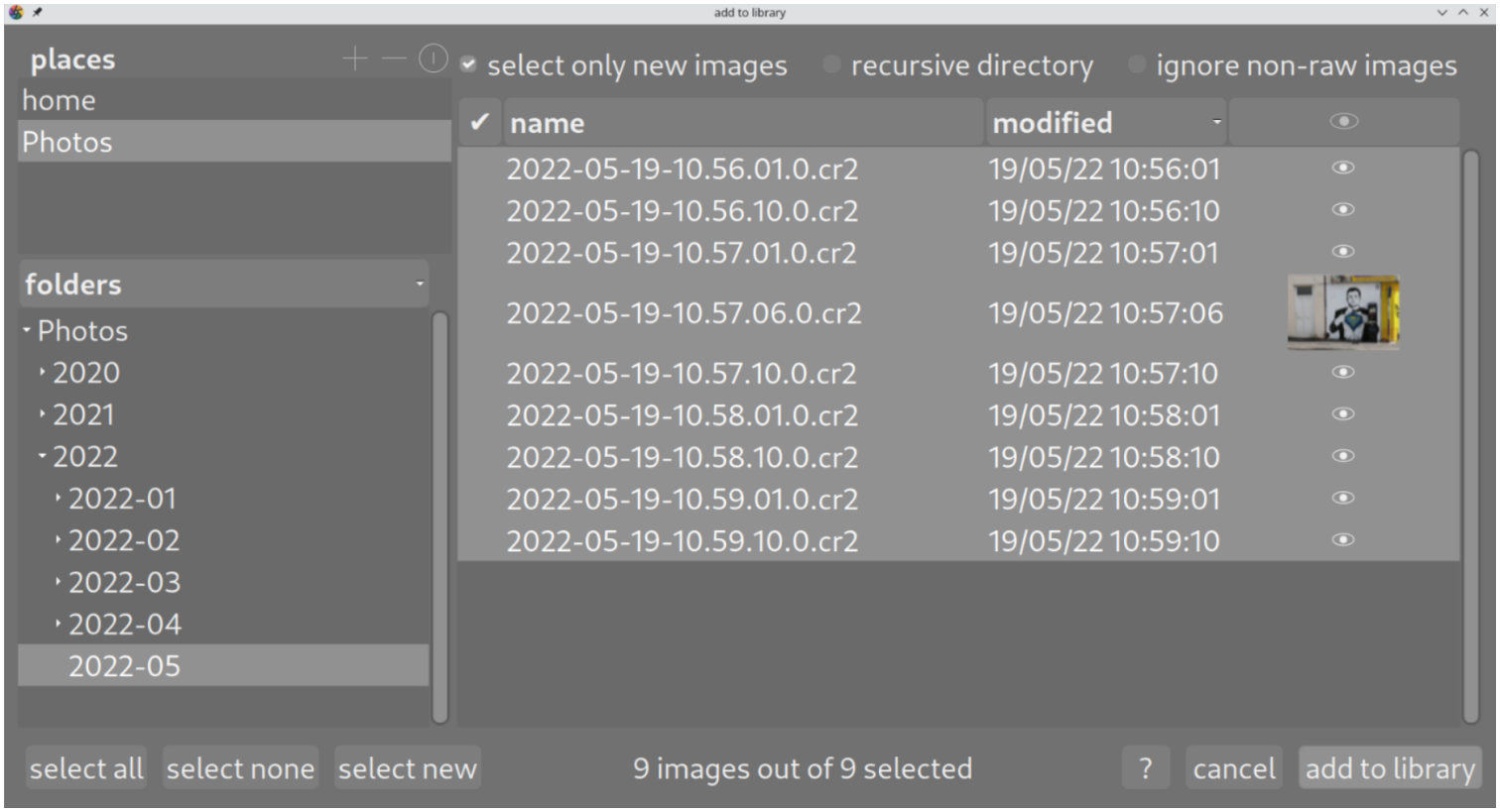
If preferences > storage > create XMP files is set to "never", an additional column "from XMP" is presented so that you can choose to prevent the import of metadata from XMP files.

tags
If you want to add further tags by default when importing images, you can provide them here as a comma separated list. As with metadata you can also choose from any presets saved within the tagging module.

import dialog

Each of the three import buttons (add to library, copy & import, copy & import from camera) uses a similar dialog for the import process, described in this section.

The following example screenshot is taken from the "add to library" button:



common functionality

places and folders

The import dialog is intended to allow you to set up common import locations, to make subsequent imports as simple as possible. When you first open the dialog, darktable attempts to add some common locations (home, pictures, mounted devices) to the places pane. You can add new places to the list by clicking on the “+” button and you can remove places from the list by clicking on the “–” button. If you wish to restore a default location that you have deleted, you can do this with the reset button.

When you choose a place, the folder tree is automatically populated (into the folders pane) from the root directory of the selected place. You can then navigate the folder tree and select a folder for import. The last selected place/folder is automatically reloaded the next time you open the dialog.

In the example screenshot above, you can see that a “place” has been created for the root of the Photos folder and a sub-folder within that structure has been selected. This is the recommended workflow for the import process – you should not have to create new places very often.

files

Once you have selected a folder, the files pane will automatically be populated with a list of the files found in that folder. By default, all files in the chosen folder are selected.

You can view thumbnails for any of the images by clicking on the icon. In addition, buttons are available at the bottom of the screen to select “all” files or “none”.

Once you are happy with your selection, press Enter or click on the button at the bottom-right of the screen to import (this button will be named differently depending on the type of import being performed).

Press Escape or click the “cancel” button to exit without importing.

common options

The following additional options are common to all import dialogs:

recursive directory
Check this option to import images in the selected folder *and all subfolders*. It is recommended that you not use this option to import a large number of images at the same time. The import process causes darktable to generate thumbnails for all of the imported images, but in the end it will only be able to keep the most recent in its cache. It is therefore better to import images in smaller chunks to avoid the performance penalty this imposes.

ignore non-raw images
Check this option if there are non-RAW images in the same folder that you do not wish to import. This option is usually used where the camera stores RAW+JPEG and you only want to work on the RAW files, leaving the JPEG images untouched.

add to library

The “add to library” button allows you to add one or more existing images to the darktable library from the local filesystem. This process does not copy or move images but merely adds their details to the library database and creates XMP sidecar files for them.

select only new images
Tick this box to restrict the *initial* selection (when the dialog is loaded) to only those images that have not already been loaded into the darktable library. If you attempt to reload existing images into the darktable library, data for those images will be reloaded from the XMP sidecar files. A button is also available at the bottom of the dialog to select only “new” images for the currently-selected folder.

Note: “Add to library” does not create duplicates of your image files in a separate folder structure but processes them in-situ. The “add to library” process simply adds details of those images to darktable’s library database (and creates an XMP sidecar file if applicable) allowing the images to be viewed and developed.

This means that if you delete images from disk after having added them, darktable cannot access them any more. Moreover, darktable does not watch for changes in the filesystem. Any new images will not be shown until they are explicitly imported.

---

copy & import

This option copies images from another location on your filesystem (including mounted storage devices) and then adds the copied images to the darktable library. Using this option, if an existing XMP sidecar file is available for the image, it will *not* be read or copied and a new XMP will be created.

The following additional options are available to control the file and directory naming of the copied files. By default, only the “import job” option is shown – click on the “naming rules” label or the expander icon beside it to show additional options:

import job
The name of the import job (populated into the `$(JOBCODE)` variable).

override today’s date
Enter a valid date/time (in `YYYY-MM-DD[Thh:mm:ss]` format) if you want to override the current date/time used when expanding the variables `$(YEAR)`, `$(MONTH)`, `$(DAY)`, `$(HOUR)`, `$(MINUTE)` and `$(SECONDS)`. Leave the field empty otherwise.

base filmroll’s directory
The base directory part of the naming pattern (default `$(PICTURES_FOLDER)/Darktable`). Click on the icon beside the input field to choose a directory manually.

filmroll name
The sub directory part of the naming pattern (default `$(YEAR)$(MONTH)$(DAY)_$(JOBCODE)`).

keep original filename
Check this box to keep the original filename instead of using the pattern below when importing.

file naming pattern
The file name part of the naming pattern (default `$(YEAR)$(MONTH)$(DAY)_$(SEQUENCE).$(FILE_EXTENSION)`).

keep this window open
Keep the window open after the import is complete, allowing multiple imports but with different naming options.

Most of these options can also be set in preferences > import . See this section for more information about the available variables.

copy & import from camera

This option copies files from a connected camera to the local filesystem and then adds the copied images to the darktable library. It provides the same naming options as the “copy & import” dialog but does not allow places or folders to be selected.

8.3.2.5. scripts

The *scripts* module is created by the *script manager* and lets you start, stop, configure, install and update darktable’s Lua scripts directly from the lighttable. See basic principles for an explanation of how the script manager loads scripts.

Since darktable 5.6 a collection of scripts is bundled with the program, so this module is available without any manual installation. Documentation for the individual bundled scripts is available in the lua docs .

Installing or updating scripts requires git to be installed and available on your path.

The module can be disabled in preferences > Lua options by disabling Lua scripts.

module controls

The **action** dropdown at the top of the module switches between three views: *start/stop scripts*, *configure* and *install/update scripts*. The module remembers the last selected view.

start/stop scripts

Lists the scripts contained in the selected folder, each preceded by an on/off button used to start or stop it.

folder
Select which group of scripts to display. The bundled scripts are grouped into *contributed*, *examples*, *official* and *tools*. Scripts placed in `[configuration directory]/lua/FOLDER/file.lua` can be selected here. See basic principles on how darktable loads added scripts.

page navigation
When a folder holds more scripts than fit on a single page, use the < and > buttons to move between pages. The current position is shown as “page x of y”. The number of scripts shown per page is set in the *configure* view.

script buttons
Each entry shows the script’s name preceded by an on/off button. Click it to start or stop that script. A started script stays active for the current session and is started again automatically the next time darktable runs.

configure

scripts per page
Set how many lines are shown per page, between 5 and 20 (default 10).

change number of buttons
Apply the value selected above. The button list updates immediately; no restart is required.

install/update scripts

This view is split into three sections.

update scripts

scripts to update
Select an installed script collection to update. The bundled scripts appear here as *lua-scripts*; any additional scripts you have added are listed by their folder name.

update scripts
Pull the latest version of the selected collection from its git repository.

add more scripts

URL to download additional scripts from
Enter the URL of a git repository that contains the scripts you want to add.

new folder to place scripts in
Enter a name for the folder the scripts will be cloned into (created under `lua/` in your configuration directory).

install additional scripts
Clone the repository into the folder named above. Once installed, the scripts appear under that folder name in the *start/stop scripts* view.

disable scripts

enable “disable scripts” button
A safety check that must be ticked before the *disable scripts* button becomes active, to prevent it from being clicked by accident.

disable scripts
Renames the `luarc` file in your configuration directory to `luarc.disabled`. Note that in the bundled installation (5.6 onwards) this does *not* stop the bundled scripts from loading (see this issue). They are loaded independently of your `luarc`. To actually prevent the Lua scripts from running, use the *disable Lua scripts* option in preferences > lua options instead. (To undo the rename, manually rename change `luarc.disabled` back to `luarc`.)

8.3.2.6. selection

Select images in the lighttable according to simple criteria.

module controls

select all
Select all images in the current collection.

select none
De-select all images.

invert selection
Select all images in the current collection that are not currently selected.

select film roll
Select all images in the current collection that are in the same film roll as the currently-selected images.

select untouched
Select all images in the current collection that have not yet been developed.

8.3.2.7. timeline

View your images by the date/time they were taken.



In the lighttable filemanager view, you can show/hide the timeline module in the bottom panel with the shortcut Ctrl+F.

Within the timeline, you can show the next and previous dates by scrolling your mouse; Ctrl+scroll to zoom in/out.

You can also use the timeline to select images by date range by clicking and dragging your mouse.

8.3.3. map

8.3.3.1. find location

Search for a location on the map. You must be connected to the internet to use this feature.

To use this module, type in a place name or address, press Enter and a list of results will be shown. Click on an item in the list and the map will zoom to that location. An outline covering that location or a pin pointing at the location will be displayed.

An outline (polygon) can be used to create a user location. Check the *max polygon points* parameter in the map settings module to ensure that sufficient points are available for a polygon to be displayed.

8.3.3.2. locations

Create areas or locations and organize them as hierarchical tags.

A location is shown as a shape on the map when selected. Initially each location is represented as a square or circle and can be changed to a rectangle or ellipse by adjusting the shape's width and/or height.

A location can also be created from an OpenStreetMap region (city/country) polygon. To achieve this, first make sure the *max polygon points* parameter is large enough (some country polygons use more than 150,000 points). Then select the desired location in the find location module. When the corresponding region shape is displayed, a polygon symbol becomes available in the "shape" control (see below). Select it to create the new location.

Each location is stored as tag entry under the geotagging collection in the collections module. The pipe "|" character can be used to insert a new level (a group of locations).

module controls

shape
Select the circle or rectangle symbol to choose the default shape for new locations. A polygon symbol is available when a shape is displayed by the find location module.

new location / new sub-location
When no location is selected you can use the *new location* button to create a location at root level. When a location is selected use the *new sub-location* button to create a sub-location within the selected location.

show all
Show or hide all locations that lie within the visible area of the current map.

actions on the locations list

click
Select or de-select a location. If the location is not currently visible on the map, the map is automatically centered on that location.

Ctrl+click
Edit the name of the location. Press Enter to save changes or Esc to close the editing window without saving.

right-click
Open a sub-menu which allows you to:

- Edit the name of the location.
- Delete the location.
- Update the filmstrip – the filmstrip will be populated with all images in the selected location.
- Switch to the lighttable view and show a collection that contains all images in the selected location.

actions on locations in the map

Note: The following actions have no effect on polygon locations.

click and drag
Move the location shape to a new position on the map.

Ctrl+click or Ctrl+Shift+click
Move an image or a group of images and place them inside a location shape.

mouse scroll
When inside a location shape (but not over an image), increase or decrease the size of that shape.
When hovering over an image, cycle through the thumbnails of images located at that position on the map.
When outside a location shape (and not over an image) zoom the map in or out.

Shift+scroll
Increase or decrease the width of the location shape.

Ctrl+scroll
Increase or decrease the height of the location shape.

click on a location shape
Select a different location when the *show all* checkbox is selected.

8.3.3.3. map settings

Select preferred map data from various providers. Some will provide additional layers (satellite view etc.) which you can toggle.

module controls

map source
Choose the provider to source map information from.

max polygon points
The find location module doesn't display polygons with more points than this for performance reasons. Usually a country polygon has between 50,000 and 150,000 points.

show OSD
Choose whether to display the OSD controls at the top-left of the center view.

filtered images
Check this box to display only the images from the current collection (those shown in the filmstrip) in the center view. Uncheck the box to display all images in the current library, where those images have associated GPS data. You can also toggle this option by pressing Ctrl+S.

max images
The maximum number of thumbnails to display on the map.

group size factor
Increase or decrease the size of area that causes images to be grouped.

min images per group
The minimum number of images that need to be placed in the same position in order to automatically create an image group for them.

thumbnail display
Define what information to show on the map display

- *thumbnails*: Display image thumbnails along with a counter.
- *count*: Just display the number of images (to free space on the map). Hover over the number of images to show the corresponding thumbnail(s). A count-only marker behaves the same way as a normal image thumbnail, in terms of color coding, scrolling, drag and drop etc.
- *none*: Show nothing.

You can also cycle through these options by pressing Shift+S.

8.3.4. print

8.3.4.1. print settings

Manage settings for the print view and initiate printing.

module controls

printer

printer
Select one of the installed printers.

media
The type of media loaded on the printer (Plain Paper, Luster Photo Paper, etc.).

profile
The printer's ICC profile for the loaded paper. This is the profile specific to the printer and paper. This profile is the last color space transformation applied to the image whose goal is to create a high quality print.

intent
The print rendering intent ("perceptual", "relative colorimetric", "saturation" or "absolute colorimetric"). See rendering intent for more details.

black point compensation
Whether to adjust the black point of the output profile, which is often lighter than the input profile. This should be "on" when the *intent* is set to "relative colorimetric".

page

paper size
The size of the paper on which to print.

orientation
Portrait or landscape (note that darktable chooses the best fit by default).

units
The unit to use for setting the margins: “mm”, “cm”, or “inch”.

margins
Set each margin separately, or all together by clicking on the middle “lock” button.

display grid
Select the grid size using the entry field (expressed in the currently selected unit). Tick the option to display the grid on the canvas.

snap to grid
Help setting the image areas by snapping them to the grid for proper alignment.

borderless mode required
Indicates whether the printer borderless mode is to be activated. This item is activated when the user’s margins are smaller than the printer hardware margins. Note that it is only an indicator as it does not activate the borderless mode automatically.

image layout

image width/height
This information field displays the actual image width and height (given with the selected units) on the paper.

scale factor
This information field displays the scaling of the image to fit on the paper. If this value is less than 1 the image is down-scaled, otherwise it is up-scaled. This is an important factor to watch – a value that is too large (up-scale) may result in a low quality print. The corresponding dpi (dots per inch) is also displayed.

alignment
Select the alignment of the image on its area.

new image area button
Create a new image area. Drag and drop on the canvas to place it. If the option snap to grid is activated the area can be easily aligned to the grid lines. An image can be placed into this area by dragging it from the filmstrip and dropping it on the new area.

delete image area button
Remove the currently selected image area from the composition.

clear layout button
Remove all the image areas leaving the canvas empty.

The following four fields represent the position of currently selected area on the page – the top/left corner on the first line and the width/height of the area on the second line.

When hovering an image area its position and size are displayed. It is also possible to grab the side and corner of the area to change the size or to drag the whole area to change its position.

The page layout can be recorded using a preset.

print settings

profile
The export profile to use. This profile is the entry point used for the next transformation using the printer’s ICC profile. Usually it is better to prefer a large gamut (e.g. Adobe RGB) rather than a smaller one (e.g. sRGB).

intent
The rendering intent to use when exporting the image. For more information see rendering intent .

style
Choose a style to apply when exporting the image – defaults to “none”. See the export module for a more detailed discussion of applying a style during export.

mode
Whether the chosen style should be appended to the existing history stack or replace it completely. See the export module for more details.

print button

When clicked, the images are first exported using the selected options, then composed on the page and finally sent to the printer.

8.3.5. shared

8.3.5.1. collection filters

Filter the *collection* of images displayed in the lighttable view and filmstrip panel using image attributes, optionally pinning filters to the top panel for quick access.

Once you have defined a *collection* of images with the collections module, the collection filters module allows you to define additional filters and sort criteria. For example, you might wish to show all images in a given folder using the collections module and then define additional quick filters to narrow-down by star rating and/or color label.

filtering attributes

For information about the image attributes you can select, please see the documentation for the collections module.

Note that only a subset of filters is currently implemented – more will be added in future versions of darktable.

default settings

By default, three filters (range rating , color label and text search are defined within this module. By default these filters select all images (i.e. show images with any star rating or color label) and are all pinned to the top panel for quick access. This default setting is also available as the “initial setting” preset.

module controls

adding new filters

To add a new filter to the module, click on the “new rule” button and select an image attribute to use.

combining multiple filters

When you use multiple filters, the first item on the filter header allows you to define how the filter is combined with the previous one.

and
Narrow down search. An image is only retained if it also fulfills the new criteria.
or
Add more images. Images from the *collection* that fulfill the new criteria are added.
and not
Exclude images. Images that fulfill the new criteria are removed.

Note: The and/or/not operators have a defined order of precedence such that “not” is executed first, then “and” and finally “or”. This means, for example, that if you define complex filters like “A and B or C and not D” these will be implemented as “(A and B) or (C and (not D))”.

---

removing or deactivating filters

You can remove or temporarily deactivate a specific filter using the buttons on the right of the filter header (see screenshots in the following sections). If a filter is pinned to the top panel, you will need to unpin it before removing or deactivating.

pinning into the top toolbar

The pin button on the right of the filter header allows you to pin a filter to the top panel. To avoid unwanted actions, pinned filters can’t be removed or deactivated.

resetting filters

Clicking on the module reset button will remove all un-pinned filters and reset all others to their default values. If you want to also remove the pinned filters, you can Ctrl+click on the reset button.

returning to a previous set of filters

You can return to a previously-defined set of filters by clicking on the history button at the bottom of the module and selecting from the resulting list.

filter widgets

A number of filter widgets have been created for use within this module. Since some of these widgets use non-standard interfaces, their usage is explained in the following sections:

color labels

The following image shows the color labels widget, set to filter images having yellow or green color labels:

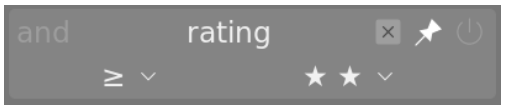


You can interact with this filter widget as follows:

- Click on a color label to include images with that label
- Ctrl+click on a color label to include images without that label
- Click or Ctrl+click on the gray icon to act on all color labels simultaneously
- Use the last button to define how to handle the selection of multiple color labels. Select  (and) to filter images having *all* of the selected color labels; Select  (or) to filter images with *at least one* of the selected color labels.

rating

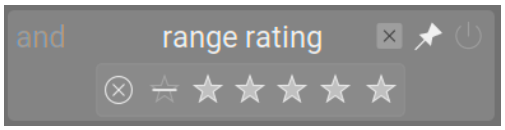
This is the classic rating selection widget that used to be shown in darktable's top panel by default.



This widget is composed of a pair of comboboxes. The combobox on the right (always visible) is used to choose a number of stars, plus some additional options ("all", "unstarred only", "rejected only", "all except rejected"). The combobox to the left (only shown when a star rating is chosen in the right-hand combobox) is used to choose an operator (<, <=, =, >, >=, ≠).

range rating

This new widget also allows you to select images by star rating, this time using a new "range" widget. The following image shows the widget with ratings of 2-4 stars selected.

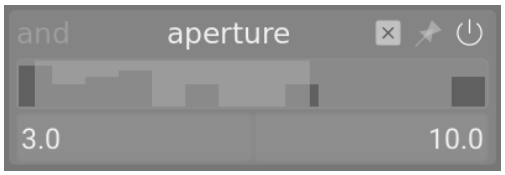


You can choose a new range of ratings to filter by clicking and dragging over the widget's interface. You can also access pre-defined ranges by right-clicking and selecting from the popup menu. The number of applicable images is also shown against each entry in this menu.

range filters for numeric attributes

Numeric attributes (aperture, exposure, focal length, iso, aspect ratio) are filtered using a widget that shows a histogram of the number of images available depending on the value of the given attribute (similar to the *timeline* in the lighttable view).

For example, the following image shows the widget when used for selecting by aperture:

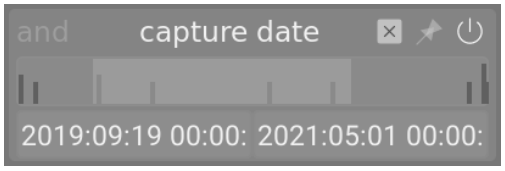


As with the range rating filter, you can select a range of values to filter by clicking and dragging on the widget. You can also access predefined ranges by right-clicking and selecting from the popup menu. The number of applicable images is also shown against each entry in this menu.

You can also use the “min” and “max” text entry fields in the main module interface to manually define the bounds of the selection.

date attributes

As with numeric and rating range filters, date/time filters are represented using a “range” widget. You can select a range of values by clicking and dragging on the widget, and use the right-click menu for more options.



You can also use the “min” and “max” text entry fields in the main module interface to manually define the bounds of the selection (right-click for more options).

The date/time format used by this module is `YYYY:MM:DD HH:MM:SS` (only the year values are mandatory).

The keyword `now` is allowed in the max field (to define the current date/time).

You can also use “relative” date/time values by preceding text entries with `+` or `-`. This allows you to set the maximum or minimum bound of the range relative to the other bound. As an example if you set the min value to `-0000:01` and the max value to `2022:04:15`, this will select images taken during the month before April 15th 2022.

filename

The filename filter allows you to search images by their filename and/or extension. You can either enter the name or extension (with the leading `.`) manually or use the right-click menu to select.

Within the extension field, you can also use the keywords `RAW` (to select all handled RAW file extensions), `NOT RAW` (to select all non-RAW file extensions), `LDR` (to select low-dynamic-range extensions) or `HDR` (to select high-dynamic-range extensions).

camera, lens

The camera and lens filters are presented text entry fields into which you can enter text to filter by. Alternatively, you can right-click on the text box to see a list of cameras or lenses to select from.

duplicates

This filter allows you to narrow down the collection to images that contain either at least one duplicate or exclusively duplicates. Filtering for only duplicates makes it easy to work with duplicates without touching the originals.

image dimensions

The image dimension filter allows you to filter the collection down to images by their original dimensions. You can use exact dimension parameters (width x height) or partial dimensions, such as `6000x` (for images with a width of 6000 pixels) or `x4000` (for images with a height of 4000 pixels).

text search

You can search images using any of their text properties (path, filename, filmroll, tags, metadata, camera, maker) using the text search filter. The search is performed “on-the-fly” and the widget is dimmed during the search process.

By default darktable performs a “fuzzy” search (wildcards are automatically added to the start and the end of the text). If you want to search for an exact match, you can enclose it with double-quotes (`"match this exactly"`).

sorting

The bottom part of the module allows you to define the sort order of the images when displayed in the lighttable and filmstrip views. As with the filter criteria, you can add multiple rules to this section. However, only the first-selected sort criteria is pinned to the top panel (this cannot be unpinned).

As with filter criteria, the history button allows you to access any previously-used sort criteria.

The button to the right of the attribute selection allows you to choose whether to sort that criteria in ascending or descending order.

custom sort

Setting the sort order to “custom sort” presents images in any sequence you define. Images are initally ordered by an image property (import order, image capture time, file name, etc.) but you can then incrementally alter “custom sort” sequences by dragging-and-dropping images within the lighttable view. Be careful: an erroneous drag-and-drop can easily ruin one or more painstakingly-created image sequences.

warnings

Please bear the following warnings in mind when using custom sort

- “Undo” does not reverse drag-and-drop resequencing.
- Drag-and-drop resequencing is *always* enabled in lighttable filemanager and in the filmstrip, but any resulting changes *are invisible* if “custom sort” is not currently in effect.
- Ensure any image you drag is *not selected* unless you want to reposition the entire currently-selected set.
- Aim your “drops” accurately: verify the correct target is *highlighted* before dropping!
- Custom sort positions are lost if images are removed from darktable or their tags are detached or deleted (since sort order is not stored in an image’s sidecar file).
- Do not use “custom sort” for tag-based collections that specify a wild-card ("*" or “%”) expression.

  When a wild-card tag expression does not resolve to a *single* tag, drag-and-drop operates only on the sequence of the first matching tag. If that tag is attached to a source image, that image is resequenced. If the same tag is also attached to the target image, the drop occurs there; if not, resequenced images appear at the top or bottom of the collection’s “custom sort”. If the first matching tag is not attached to a source image, the image is not repositioned. The “tag case sensitivity” preference is also relevant here: “insensitive” treats “%” (but not “*”) as a string-match wild card. The global sequence is not affected in any of these cases.

which custom sort sequence?

The collection type determines which image sequence drag-and-drop changes. If the collection’s selection criteria includes *tag*, then a *tag-based* “custom sort” sequence is altered. Otherwise, the *global* “custom sort” sequence is altered. Collection filters (even if tag-based) do not influence which sequence is changed by drag-and-drop.

When “custom sort” is in effect, dragged-and-dropped images are immediately presented in their new (tag-based or global) “custom sort” sequence.

what to drag, where to drop?

While viewing a collection in filemanager or the filmstrip, you may drag any hovered-over (highlighted) image to any other image in the displayed collection. Empty space before, between, or after images is not a recognized drop target.

You may relocate a single image (unselected or single selection) or a selected set of images by dragging one of them to the desired position as follows:

- When *sort order* is “ascending” the dragged image/set will be placed *before* the image you drop it on to.
- When *sort order* is “descending” the dragged image/set will be placed *after* the image you drop it on to.
- Abort an accidental drag by dropping the dragged image *outside* of the thumbnail display. If all peripheral panels are hidden, then drop the dragged image on *itself*. If multiple images were selected, the entire selected set will be relocated to that spot.
- If the “dropped on” image is subsequently resequenced, filtered out, or even deleted, the dropped image(s) remain where the “dropped on” image was located.

which image is last (or, in descending sort, first)?

- An image dropped after the last image on lighttable is regarded as dropped *on the last image*, thus positioned before it.
- To change the last image in an “ascending” “custom sort” display, drop it on the existing last image, then drag-and-drop that last image on its intended replacement.
- If you expect to use a “descending” sort, do the opposite to change the first image.

suggested process

1. If you intend to present a collection in more than one “custom sort” sequence or if some images in a collection are included in other collections to be displayed with a different global “custom sort”, then tag-based collections should be used.
2. For tag-based “custom sort” only:
    - Set *sort by* to the image attribute and *sort order* (ascending or descending) you want as your tag’s initial sequence,
    - Select the images to be sequenced, and
    - in the right-hand panel at *tagging* (bottom half) add a new tag or click on an existing tag.
    - The tag is now attached to the selected images in the displayed sequence. Note the tag’s increased image count.
    - Change *collections* selection crierion to “tag” and choose the newly-applied tag. Note that the newly-tagged images are shown.
3. Long-distance drags are difficult because neither *filemanager* nor *filmstrip* scroll while you drag. Try to fit drag sources and targets onto the same screen by maximizing the the window and scrolling the display. If that’s not sufficient:
    - increase the number of images displayed (thus shrinking thumbnails) or
    - hide images between drag source and drop target using *collection filters* (e.g. by ranges of “datetime captured” or “image id”).
4. Rearrange the images as you desire:
    - Clear any active selection (*selection* > *select none*).
    - Set *sort by* > “custom sort” so you can verify your changes immediately.
    - Remember the **warnings** above as you drag-and-drop.

8.3.5.2. collections

Filter the images shown in the lighttable view and filmstrip panel using image attributes. This set of filtered images is known as a *collection*.

Importing images into darktable stores information about them (filename, path, Exif data, data from XMP sidecar files etc.) in darktable's library database. A collection may be defined by applying filtering rules to these attributes, thus creating a subset of images to display in the lighttable view and the filmstrip module.

The default collection is based on the *film roll* attribute and displays all images of the last imported or selected film roll.

The sort order of the listed collections (ascending/descending) can be toggled by clicking  .

filtering attributes

The images in a collection can be filtered using the following image attributes:

files

film roll
The name of the film roll to which the image belongs (which is the same as the name of the folder in which the image resides). Ctrl+Shift+click on a film roll to switch to the corresponding folder. Right-click to remove the contents of the film roll from the darktable library or tell darktable that its location has changed in the file system.

folder
The folder (file path) where the image file is located. Click on a folder to include the contents of that folder and all sub-folders in the collection; Shift+click to include only the images in the selected folder; Ctrl+click to show only the images from any sub-folders; Ctrl+Shift+click to switch to the corresponding film roll. Right-click on a folder name to remove its contents from the darktable library or tell darktable that its location has changed in the file system.

filename
The image's filename.

metadata

tag
Any tag that is attached to the image. Untagged images are grouped under the "not tagged" entry. When activated, a hierarchical list of known tags is displayed

title
The image's metadata "title" field.

description
The image's metadata "description" field.

creator
The image's metadata "creator" field.

publisher
The image's metadata "publisher" field.

rights
The image's metadata "rights" field.

notes
The image's metadata "notes" field.

version name
The image's metadata "version name" field.

rating
The image's star rating

color label
Any color label attached to the image ("red", "yellow", "green", "blue", "purple").

geotagging
The geo location of the image (see locations).

times

capture date
The date the photo was taken, in the format `YYYY:MM:DD`.
capture time
The date & time the photo was taken, in the format `YYYY:MM:DD hh:mm:ss`.
import time
The date/time the file was imported, in the format `YYYY:MM:DD hh:mm:ss`.
modification time
The date/time the file was last changed, in the format `YYYY:MM:DD hh:mm:ss`.
export time
The date/time the file was last exported, in the format `YYYY:MM:DD hh:mm:ss`.
print time
The date/time the file was last printed, in the format `YYYY:MM:DD hh:mm:ss`.

capture details

camera
The Exif data entry describing the camera make and model.
lens
The description of the lens, as derived from Exif data.
aperture
The aperture, as derived from Exif data.
exposure
The shutter speed, as derived from Exif data.
exposure bias
The exposure bias, as derived from Exif data.
focal length
The focal length, as derived from Exif data.
ISO
The ISO, as derived from Exif data.
aspect ratio
The aspect ratio of the image, including any cropping within darktable.
white balance
The white balance, as derived from Exif data.
flash
Whether flash was used or not, as derived from the Exif data.
exposure program
The exposure program, as derived from Exif data.
metering mode
The metering mode, as derived from Exif data.

darktable

group
Choose specific group(s) of imagee
local copy
Show files that are, or are not copied locally.
history
Choose images whose history stacks have been altered or not.
module
Filter based on the processing modules that have been applied to the image.
module order
Choose images with “v5.0”, “v3.0”, “legacy” or “custom” module orders.

module controls

defining filter criteria

The top line of the module can be used to define filter criteria for your collection as follows:

image attribute
The combobox on the left allows you to choose which attribute to use to filter your collection.

search pattern
In the text field to the right of the attribute combobox, you can write a matching pattern. This pattern is compared against all database entries with the selected attribute. The filtering mechanism detects a match if any image's attribute contains the pattern in its full text. You may use `%` as wildcard character. The collection will be limited to only the matching images. Leave the text field empty to match all images having that attribute. Where applicable, a tooltip will appear if you hover over the attribute or search pattern to provide additional information.

Attributes with numerical or date/time attributes can be used in combination with comparative and range operators. Use <, <=, >, >=, <>, or = to select images with attributes less than, less than or equal, greater than, greater than or equal, not equal, or equal to the given value, respectively. An expression in the form `[from;to]` can be used to select using a range of values.

select by value
As well as defining filter criteria using a search pattern, you can also manually choose from a list of values (for the chosen attribute) taken from the currently-matching set of images. Making such a selection will automatically populate the "search pattern" field.

The box below the *search pattern* will list values for the selected attribute that are present in the currently-selected images. This list is continuously updated as you type. You may also choose a sorting criteria by scrolling through the list and double-clicking.

If you enable single-click mode (see preferences > lighttable) you can select with a single-click rather than double-click. This mode also allows you to select a range of values with the mouse. This only works for numerical and date-time attributes.

combining multiple filters

You can combine multiple filters together to create more complex collections of images using a series of rules. A rule is a combination of a filter criteria along with a logical operation that defines how that criteria is combined with any previous rules.

Click on the  button (to the right of the search field) to open a menu with the following options:

clear this rule
Remove the current rule, or reset it if this is the only rule defined.

narrow down search
Add a new rule, which is combined with the previous rule(s) in a logical AND operation. An image is only retained as part of the collection if it also fulfills the new criteria.

add more images
Add a new rule, which is combined with the previous rule(s) in a logical OR operation. Images that fulfill the new criteria are added to the collection.

exclude images
Add a new rule, which is combined with the previous rule(s) in a logical EXCEPT operation. Images that fulfill the new criteria are removed from the collection.

The logical operators defining how rules are combined are indicated with icons to the right of each added rule: AND by the  symbol, OR by the  symbol, and EXCEPT by the  symbol. Click on any of these icons to change the logical operation for that rule.

updating the folder path of moved images

While it is best to not touch imported files behind darktable's back, this module can help you to recover from situations when you have moved or renamed image folders after importing them. The collections module has a feature that allows you to update darktable's library database with the new folder location. The process is as follows:

1. Set the *image attribute* combobox to "folder" or "film roll".
2. The original film roll or folder name will be displayed with ~~strikethrough~~ formatting to emphasize that it can not be located.
3. Right-click on the folder or film roll name, select "update path to files…", and then select the new location of the folder.

returning to a previous collection

You can return to a previously-defined collection by clicking on the history button at the bottom of the module or using the recently used collections module – see the preferences section for more details.

preferences

The "preferences…" option in the presets menu allows you to adjust the behavior of the collections module as follows:

do not set the 'uncategorized' entry for tags
Do not set the 'uncategorized' category for tags that do not have children (default off).

tag case sensitivity
Set whether tags should be case sensitive or not – without the `sqlite` ICU extension this will only apply to the 26 latin letters (default insensitive).

number of collections to be stored
Set the number of recent collections to show in the history popup (if present).

number of folder levels to show in lists
The number of folder levels to show in film roll names, starting from the right (default 1).

sort film rolls by
Sort film rolls by one of the following options:

- "folder name" (path)
- "display name" (the roll label shown in the list)
- "import time" (the date the film rolls were first imported, default)

8.3.5.3. copy and paste

Copying and pasting edits between images is available in multiple views. In the lighttable, the history stack module provides buttons for these operations; the keyboard shortcuts listed below can also be used. In other views, such as the darkroom, copying and pasting is done through keyboard shortcuts.

copy (`Ctrl+c`)
Copy the complete history stack from the selected image, including the module order. If more than one image is selected, the history stack is taken from the image that was selected first.

Information relating to internal display encoding and mask management is considered unsafe to automatically copy to other images and is therefore not included in this operation.

The following modules are excluded by default from the *copy* operation:

- *orientation*
- *lens correction*
- *raw black/white point*
- *rotate pixels*
- *scale pixels*
- *white balance*
- deprecated modules

If you wish to include these modules use *selective copy* and *selective paste*.

selective copy (`Ctrl+shift+c`)

Copy parts of the history stack from the selected image. A dialog appears, from which you can select which history stack items you want to copy.

For any module, you may also choose to “reset” that module’s parameters – this will cause the module to be copied but with all controls set to their initial (default) state (as if you had clicked the module reset button). Double-click a history item to copy only that item and immediately close the dialog.

You can also choose to include or exclude the module order during copying.

If more than one image is selected, the history stack is taken from the image that was selected first.

paste (`Ctrl+v`)

Paste all items of a copied history stack onto all selected images, preserving the existing history stack by appending the edits. If *selective copy* was used to copy selected modules, only those modules are pasted.

Pasting is always performed from the current state of the source image. If you change modules in the source image after copying and then paste, the changed (most recent) settings will be pasted.

selective paste (`Ctrl+shift+v`)

Paste parts of a copied history stack onto all selected images. As with *selective copy*, a dialog appears from which you may choose items to paste (or *reset*) from the copied history stack. Choose to *append* or to *overwrite* the history stack of the target image(s).

You can also choose to include or exclude the module order during pasting.

A copied history stack can have multiple instances of the same module (with identical or different instance names) and pasting behaves differently for these entries depending on whether you use *overwrite* or *append*:

- *overwrite* deletes the history of the destination image before adding every module instance from the copied history stack. Choosing *select all* in both the *selective copy...* and *selective paste...* dialogs will precisely duplicate the copied history stack to the destination image(s) (including any duplicate occurrences).
- *append* takes each module from the copied history stack and if there is a module in the destination image with the same name, it is replaced and added to the top of the history stack. If there is no such module, a new instance will be created and added to the top of the history stack. If a module has multiple instances in either history stack, only the last occurrence of that module will be added.

sync the last change to all selected images (`Ctrl+x`)

Only available in the darkroom through its keyboard shortcut.

Sync the module that was changed last from the active (open) image to all selected images: Select multiple images and apply an edit to the opened image. Hit `Ctrl+x`: The last changed module is synced from the currently open image to all selected images (you can also do the edit first and then select the images to sync the change to).

Note:

- Using *selective paste* with *overwrite* is a destructive process. Use it with care. When using *append*, the existing history stack items are preserved and can be used to restore the image’s state from before the paste. With *overwrite* all the image’s previous edits are irrevocably lost.
- Automatic module presets are added only to an image when it is first opened in the darkroom or its history stack is discarded. If you use *overwrite* to paste history stack entries to images that haven’t previously been opened in the darkroom, then the next time that image is opened in the darkroom, automatic presets will be applied to the image. It may therefore seem as if the *overwrite* behavior did not accurately duplicate the existing history stack, but in this case, those automatic modules were applied after the paste action.

8.3.5.4. export

Export selected images.

When used in the darkroom view, the currently-edited image will be exported if no other images are selected in the filmstrip.

Files can be exported to a file on disk, email, various online storage locations, a web album, or a book template.

module controls

storage options

target storage
The type of location to store your selected images. A number of different back-ends are implemented, including file on disk, system clipboard, LaTeX book template and various web albums. Depending on the selected target, you will be asked to provide additional information, such as filenames or account name and password.

filename template
Define the folder and file to which the image will be exported. This can be automatically generated using several pre-defined variables. For example, `$(SEQUENCE)` in a template records sequence number of the image exported to the target. See the variables section for details.

output directory selector
The button beside the filename template entry opens a dialog to select the parent directory for export.

on conflict
Choose what to do if the generated filename conflicts with an existing file on export:

- *create unique filename*: Automatically choose a unique new file name by appending an integer to name of the conflicting file.
- *overwrite*: Automatically overwrite existing files. This option will present you with a confirmation dialog in order to protect you from accidental data loss – you can disable this in preferences > security > ask before exporting in overwrite mode . ***Note:*** *This dialog is not presented per-file but as a one-off confirmation before the export job starts.*
- *overwrite if changed*: Automatically overwrite existing files if the last export timestamp stored in darktable's database does not match the last changed date/time on the existing file.
- *skip*: Do not export images where the destination filename already exists.

format options

file format
Choose the file format for the exported image. Additional options will appear (below) depending on the selected format. Note that for some export formats there are caveats concerning metadata inclusion .

quality
The quality of the exported file. Higher values lead to larger file sizes. The default quality (95) is a good setting for very high quality exports (e.g. for archiving or printing purposes). If you need a good compromise between size and quality (e.g. for online image display or uploads) you should consider a value of "90" instead.

bit depth
The number of bits used for each color channel. More bits means less posterization/color banding.

compression
The type of compression to use.

compression level
For export formats where *compression* can be specified, the *compression level* specifies how much compression to apply. The higher the level, the more the data will be compressed, at the cost of more CPU cycles.

b&w image
For TIFF export format, it is possible to save a monochrome image. This setting controls whether the resulting file encodes the shades of gray as separate RGB channels, or as a single grayscale channel. The latter option will result in smaller files.

global options

set size
Choose how to measure the maximum size of your exported image

- *in pixels (for file)*: Enter the maximum width and height in pixels.
- *in cm (for print)*: Enter the maximum width and height in cm and define the image's dpi. The equivalent size in pixels will be automatically calculated.
- *in inch (for print)*: Enter the maximum width and height in inches and define the image's dpi. The equivalent size in pixels will be automatically calculated.
- *by scale (for file)*: Enter a multiplier to specify by how much the exported image should be scaled compared to the input image. For example, entering a value of 0.5 will result in an output image with half the width and height (in pixels) of the original image.

dpi

If units of cm or inches are chosen, set the dpi of the output image. The dpi will also be stored in the Exif data of the exported image. It will be automatically set to 300 if “in pixels” or “by scale” is chosen.

max size

Set the maximum width and height of the exported image(s) in pixels, cm or inches (depending on the selected unit) – zero means that no constraint will be set on that dimension. Exported images will be constrained so as not to exceed either of these values, while retaining the correct aspect ratio. Set both to zero to export with the original dimensions (after cropping). If the entered values exceed the original dimensions darktable will either export with the original dimensions or upscale the image, depending on the “allow upscaling” parameter.

allow upscaling

Set to “yes” to perform an upscaling step if the user-defined maximum width and height exceed the original dimensions of the image. If set to “no” the exported image’s dimensions will not exceed the dimensions of the original image (after cropping). This uses the pixel interpolator set in preferences; for higher-quality super-resolution upscaling using an AI model, see the neural restore module.

high quality resampling

Set this to ‘yes’ to perform high quality resampling on the image. The image will be processed in full resolution and only downscaled at the very end. This can sometimes result in better quality, but will always be slower.

store masks

Store masks as additional layers (for TIFF format) or channels (for EXR and XCF formats) in the exported image.

profile

The output color profile. Select “image settings” if you want the settings in the *output color profile* module of the individual images to be respected.

PQ peak

The brightest highlight of a PQ (HDR) export, in nits (203 to 10000, default 1000). Tone mappers that follow the output, such as *AgX* and *sigmoid* , map their white to this peak, while mid-tones keep their SDR brightness. Set it to the peak brightness of the display the image is meant for. This option is only shown when *profile* is a PQ profile or “image settings”. HLG exports always peak at 1000 nits, and SDR exports are not affected. The setting is stored in export presets, and an export keeps the peak it was started with, even if the setting is changed while it runs. The darkroom also uses this setting when your display profile is PQ.

PQ exports also record how bright the image actually is, so that HDR displays and viewers can tone map it: PNG, AVIF and HEIF files carry its content light level (MaxCLL and MaxFALL), and JPEG XL files carry its measured peak luminance as their intensity target. AVIF needs darktable built with libavif 1.0.0 or newer, and HEIF with libheif 1.15.0 or newer.

intent

This option lets you define the intent – the way in which darktable will handle out-of-gamut colors. See rendering intent for a more detailed description of the available options.

style

Choose a style which darktable will combine with the existing history stack to generate the output image. These history items are only added temporarily – the original history stack is not overwritten. You can use this feature to add processing steps and parameters that you want to be applied specifically to images before export. For example you may define a style that adds a stronger level of sharpening when you produce scaled-down JPEG files for the internet or add a certain level of exposure compensation to all of your output images.

mode

When applying a style during export this option defines whether the history stack items of that style replace the original history stack of the image or are appended to it. Technically speaking, in append mode history stack items of the style will constitute separate instances of the respective modules on top of any existing ones (see also multiple instances). As a consequence the original history stack will remain in effect with the new items being applied in addition. This way you can apply an overall adjustment (e.g. exposure) to a number of exported images while respecting the settings of each individual image.

start export

Press this button to start a background job to export all selected images. A bar at the bottom of the left hand panel displays the progress of the export job. Furthermore a notification message pops up reporting the completion of each individual export. You may click on the pop-up to make it disappear. You may abort the export job by clicking on the “x” icon located close to the progress bar.

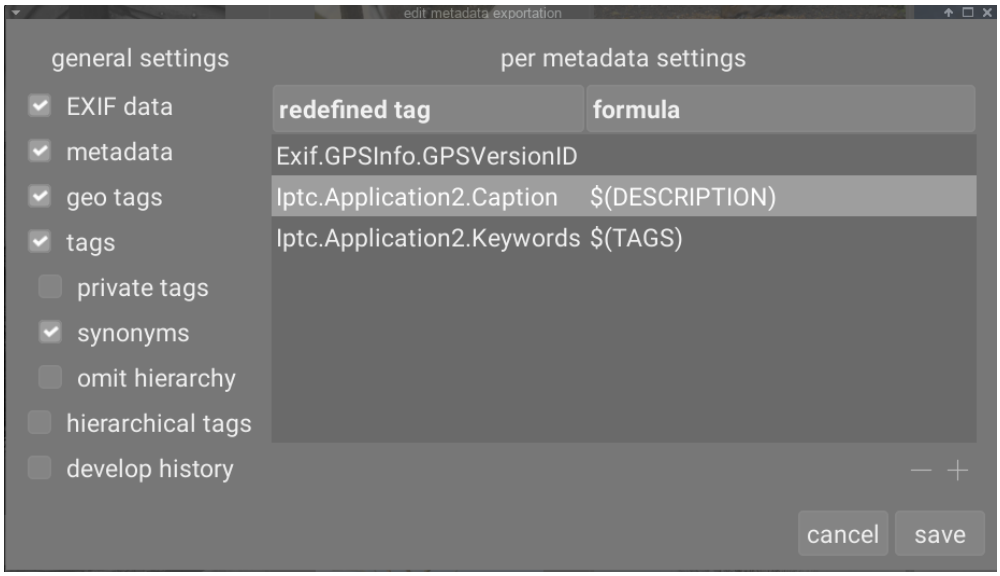
multi-preset export

Expanding this section allows you to easily export each of the selected images with several different presets in a single export run. For example, you may wish to export each image in multiple sizes at the same time for use in a website. Simply create a preset for each type of export you wish to make, ensuring that the file path settings for each preset is unique (so that the exported images don’t overwrite one another). Select the desired presets and press the “start export” button as normal.

Note: Images that are selected but currently hidden (because they are members of a collapsed group) will not be exported.

metadata preferences

The “preferences...” option in the presets menu brings up a dialog where you can configure what metadata to include in exported files:



The parameters entered into this dialog are saved along with other export parameters to user presets and the last entered values are retained when darktable is closed.

Caution: For **AVIF, EXR, HEIF, JPEG XL and XCF selecting specific metadata is not supported**. For these formats, *no* metadata fields will be included unless the user selects *all* of the metadata checkboxes in the export module preferences.

general settings

The left-hand-side of this dialog allows you to choose which groups of metadata are to be exported with the image. The following options are available:

exif data
Export the source image’s Exif data.

metadata
Export metadata defined in the metadata editor module. Only metadata fields that are tagged as *visible* and are not tagged as *private* will be exported.

geo tags
Export geo tags.

tags
Export tags created in the tagging module (to `Xmp.dc.Subject`). Three additional options can also be selected:

- *private tags*: Export private tags
- *synonyms*: Export tag synonyms
- *omit hierarchy*: Only export the last part of hierarchical tags

hierarchical tags
Export hierarchical tags (to `Xmp.lr.Hierarchical Subject`)

develop history
Export the entire development history (history stack and shapes) where supported (e.g. JPEG, JPEG2000, TIFF). The development history will be stored as XMP tags within the output file. This information can later be used to reconstruct the parameters and settings that were used to produce the exported image.

Caution: For various reasons embedding XMP tags into output files may fail without notice, e.g. if certain size limits are exceeded. Users are therefore advised to not rely on this feature for their backup strategy. To back up your data always make sure to save your input (raw) file as well as all of darktable's XMP sidecar files.

per metadata settings

The right-hand-side of this dialog allows you to define formulas to populate image metadata. The formulas defined here have priority over the settings in the left-hand-side of the dialog. The first column identifies the entry to be edited. The second column allows you to define how to calculate the value for that metadata entry using a formula.

See the variables section for details of the variables you can use in your metadata formula. Press Enter to validate the formula. Leave the formula empty to prevent a given metadata entry from being exported (`Exif.GPSInfo.GPSVersionID` in the above example).

Use the "–" icon to remove a metadata entry from the list and the "+" icon to add a new one from a predefined list of available metadata tags.

Click on the "add" button to add a metadata entry to the list.

The formulas allow you virtually define all the metadata you need to qualify your images in tagging and export the values in the XMP or IPTC tags of your choice. The exported tags can be different from one export to the next depending on the destination of the images. Tags and Categories are displayed separately in image information.

Remember that a tag set up as a category is never exported.

tips

- To prevent a specific metadata field from being exported, add it to the list and leave the formula empty.
- To force a specific exif metadata field to be exported when exif export is disabled, add it to the list and enter = into the formula.

examples

example 1

A first level tag called places is set as a category, and is followed by four levels of information (or keywords): country, region, city and location (e.g. `places|France|Nord|Lille|rue Nationale`). Each level can be retrieved (when it is defined) by one of the variables `$(CATEGORY0(places))`, `$(CATEGORY1(places))`, `$(CATEGORY2(places))` and `$(CATEGORY3(places))`. In this example, the returned values are "France", "Nord", "Lille" and "rue Nationale", respectively. These keywords can also be retrieved as simple tags using the variable `$(TAGS)`. The last keyword level defined (the leaf) is displayed in image information , here "rue Nationale".

example 2

A first level tag called creator is followed by the name of the photographer, both set as categories: `creator|firstname lastname`. The formula copyrights `($(YEAR) $(CATEGORY0(creator)))` builds the text associated with image rights. Here, image information displays "creator: firstname lastname" as categories. Neither creator nor "firstname lastname" appear in the tags list and they are not exported as simple tags.

Note: tagging is not appropriate to define free text metadata, like a title or a description, which may be specific to each image. Use the metadata editor for this type of information.

8.3.5.5. filmstrip

The filmstrip can be used to quickly switch between images. The images shown are the same as those displayed in the lighttable view and are defined by the currently-selected collection.



The filmstrip can be switched on and off using the shortcut Ctrl+F. The height of the filmstrip panel can be changed by clicking and dragging its top border.

Quickly navigate through images in the filmstrip by scrolling with the mouse. Increase scroll speed with Shift+scroll. Change the height of the filmstrip with Ctrl+scroll or by clicking+dragging the top of the panel. Change the custom sort order of images by dragging and dropping image thumbnails.

In the darkroom, the image currently being processed is selected and highlighted. Double-click on an image to open it for editing. Single clicking a currently not-opened image will also select that image in addition to the currently open one.

By hovering inactive images you can act on them with a keyboard shortcut without changing the image being processed. Placing the cursor over the main viewport in effect hovers the current image.

If you wish to select multiple images in the filmstrip, click to select the first image followed by either Ctrl+click to select or de-select further images, or Shift+click to select a range of images.

Note: The currently open image will be kept selected, even when selecting other images. This is different from the behavior in lighttable view and from what you might be used to from your file explorer. It is something to be aware of to avoid e.g. accidentally rejecting the currently open image.

The following shortcuts can be used to select images in the filmstrip:

- `Ctrl+A` selects all images in the filmstrip
- `Ctrl+Shift+A` deselects all images
- `Ctrl+I` inverts the current selection

The following shortcuts can be used to perform operations on the selected images

- `F1, F2, F3, F4, F5` adds or removes a color label (red, yellow, green, blue, purple, respectively). A color label will be added if *any* selected image does not currently have the label; otherwise the label will be removed
- `0, 1, 2, 3, 4, 5` sets the star rating
- `R` rejects the image(s)
- `Ctrl+D` duplicates the image(s)
- `Ctrl+C` copies the full history stack
- `Ctrl+V` pastes all of the copied history stack
- `Ctrl+X` copies the settings of the of the last changed module to all selected images
- `Ctrl+Shift+C` selectively copies the history stack
- `Ctrl+Shift+V` selectively pastes from the copied history stack

See copy & paste for detailed information about the copy and paste functionality.

The filmstrip image under the cursor can be pinned to the second darkroom window by assigning a shortcut to utility modules > filmstrip > pin in second window .

8.3.5.6. focus peaking

Identify which parts of the image contain high contrast details such as edges and textures, which are usually a useful guide to sharpness and therefore focus.

Activate the module by clicking on the icon. The sharpest parts of the image will be highlighted with a yellow, green and blue overlay:

Focus peaking works by filtering out most of the image noise, measuring the intensity gradients in the image and calculating average and standard deviation statistics. When the gradient of an edge differs significantly from the mean, the associated pixels are marked with a “heat map” indicating how sharp the edge is.

- *yellow* represents a large (6σ) jump in gradient, indicating a very sharp edge.
- *green* represents a medium (4σ) jump in gradient, indicating a reasonably sharp edge.
- *blue* represents a small (2σ) jump in gradient, indicating a slightly sharp edge.

Note: While the algorithm in this module generally does a good job of locating the sharpest parts of an image, it does not necessarily detect *whether* an image is sharp. Additionally, as it uses local contrast to detect sharpness, it will also highlight the edges of dark objects against bright backgrounds (and vice versa) even if those edges are blurred. Because it runs at the end of the pixelpipe, it may also detect the results of any sharpening you have performed within darktable.

The following example image was taken with a wide aperture to give a shallow depth of field, and you can see how the camera has focused on the chinese character written on the second red lantern along from the front. There are also stems of the pink flowers that fall within the area of acceptable sharpness around the focal plane, and these have also been marked up with yellow and green.



8.3.5.7. geotagging

Import and apply GPX track data to a selection of images.

This module is common to the lighttable and map views. The map view provides an enhanced mode that allows you to preview the position of the images along the GPS tracks while adjusting the images’ date/time offset and time zone.

workflow overview

A GPS receiver calculates its current position based on information it receives from satellites, and records it in a GPX file together with the current date and time. The Exif data of the images also contains a time stamp defined by the camera settings. The *geotagging* module takes the time stamp of the image, looks up the position in the GPX file at that time, and stores the appropriate coordinates (latitude/longitude/elevation) in its database and the image's XMP sidecar file.

Two problems may occur during this process:

- In contrast to GPS devices, most cameras don't record the time accurately.
- The time stored in the Exif data does not include the time zone. Most people set their camera to local time, whereas GPS devices store the time in the UTC (Universal Time, Coordinated, i.e. Greenwich (London)) time zone. If the time zones of the camera and GPX file differ, than the related location will be wrong.

If your image already shows the correct date/time and carries the UTC time stamp, you can directly apply the GPX track file without further adjustments.

Otherwise follow the following process to correlate the time of the images and GPS track files

1. Fix the camera time setting for a single image by manually entering the correct date/time for that image into the *date/time* field. A good way to do this accurately is to take a photograph of a reliable time source. This can be any precise clock or, even better, the time displayed on your GPS device (bearing in mind that GPS devices normally *show* the local time, even though they *store* universal time). The difference (offset) between the time entered and the time stored in the image's Exif data will be shown in the *date/time offset* field.

2. Press the *lock* button to lock the calculated offset in the module.

3. Select all of the images you wish to adjust and click *apply offset* to apply the calculated offset to those images.

4. If the camera time zone is not UTC, set the time zone in the *camera time zone* field.

With the time setting corrected, you can now apply GPX tracking data using the *apply GPX track file* button.

module controls (common)

date/time
This field is initialized with the date/time read from the first selected image (format `yyyy:mm:dd hh:mm:ss`) and can be modified to correct the date/time for that image. The individual date/time fields can be altered by scrolling over them with your mouse. If any field reaches its limit, the neighboring fields are automatically updated. For example, if you go over 60 on the minute field, the hour field will automatically be incremented. It is also possible to use milliseconds in this module if you enable preferences > lighttable > show image time with milliseconds .

original date/time
The original date/time of the image is shown here for reference.

date/time offset
The calculated difference between the original date/time and that keyed in the *date/time* field. If the calculated difference is greater than 9999 days 23 hours 59 minutes and 59 seconds, the offset is invalidated.

lock button
If the lock button is activated, the offset value is frozen. If you subsequently change the selected image(s), the new image date/time is updated using the locked offset. This allows you to apply the same offset to multiple groups of images.

apply offset button
Apply the offset to selected images.

apply date/time button
It is sometimes useful to be able to set the absolute date/time for an image, for example where this information is missing. This button allows you to apply the date/time entered in the *date/time* field to the selected images, without considering the previous value. You can use Ctrl+Z to undo any unwanted changes.

camera time zone
Select the camera's time zone. Start typing to show a list of permitted values.

apply GPX track file (lighttable view only)
Apply a GPX track file. Click the corresponding button and navigate to the GPX file. You can use Ctrl+Z to undo any unwanted changes. Within the file chooser window, the preview button lists the tracks of the selected GPX file along with the following information: track name, start and end date/time (local time), number of track points and number of selected images that will be geotagged.

module controls (map view)

GPX file
The path of the GPX file selected.

track list
This table shows the start date/time of each track, along with the number of track points and the number of matching images. When a check button is activated, the related track is displayed on the map. A check button in the table header allows you to select or de-select all of the tracks at once. Hover over a row with your mouse to display the start and end times both in local time (LT) and UTC. To reset the map zoom and show the complete set of tracks, double-click on a track in the list, or single-click on the track list header. You can also assign a keyboard shortcut (preferences > shortcuts > utility modules > geotagging > view entire track).

preview images
If checked, the matching images are displayed on the map along the visible tracks.

select images
If you don't wish to apply an offset to all of the selected images, but only to the matching images, use this button to select images. If you don't want to lose the current offset you may want to lock it before changing the selection.

counts
A count of the number of matching images and the number selected is displayed to the right of the *select images* button.

apply geo-location
This button is displayed when the offset is null. The *apply geo-location* button applies the GPX data to matching images on selected tracks.

apply offset and geo-location
This button is displayed once an offset has been entered. As a reminder, the *apply offset* button applies the offset to all selected images. Unlike *apply offset*, the *apply offset and geo-location* button applies the offset *and the GPX data* to matching images on selected tracks.

You can use Ctrl+Z to undo any unwanted changes (twice in case of *apply offset and geo-location*).

8.3.5.8. image information

Display information embedded within an image's Exif data as well as a number of additional data fields defined by darktable.

When hovering with the mouse over image thumbnails, the displayed data is automatically updated to show information about the image currently under the mouse cursor.

When several images are selected and the focus is not on a single image, the module only displays information that is the same for all images. If any fields differ between the images, the text "<*various values*>" is displayed instead.

While you are in the lighttable view, you can double-click on the *filmroll* field for a given image to show all images in that image's film roll.

preferences

The "preferences..." option in the presets menu brings up a dialog with a list of all fields that are available for display.

The *visible* checkbox allows you to choose which fields to display. You can also drag and drop one row at a time to change the display order.

These preferences can be saved as module presets. Press the module's reset button to make all available information visible and displayed in its default order.

8.3.5.9. metadata editor

Edit the metadata of selected images.

Metadata is freeformat text (title, description, creator, publisher, rights etc.) that describes your images.

module controls

reset
Delete visible metadata (see below) from the selected images.

metadata entry fields
A separate entry field is displayed for each visible metadata item, containing its current value (if any). Press Ctrl+Enter to insert a new line in a field. Double-click on a field's label to delete the contents of that field.

When several images are selected that have differing values for a given field, the module displays "*<leave unchanged>*" for that field – if you choose to apply changes, these fields will not be changed. If you right-click on such a field, the different values are listed at the end of the contextual menu. Select one of those values to apply that value to all of the selected images.

Hold Ctrl while scrolling with your mouse to increase the height of a field's entry box.

apply
Apply new settings from the metadata entry fields to the selected images.

keyboard

You may use the keyboard to navigate and apply changes while any of the metadata entry boxes have focus, as follows:

- The Tab key saves the current field and moves the cursor to the next field. When the last field is reached, the Tab key returns focus to the first field.
- Shift+Tab works the same as Tab, but in the opposite direction.
- The Enter key saves the current field without moving the cursor.

preferences

The "preferences" option in the presets menu brings up a dialog where you can configure what metadata tags to display and how they are handled within darktable. For each metadata item, two check boxes allow you to restrict how metadata is handled as follows:

visible
Show or hide this metadata field. Hidden fields are not included in exported images.
private
Keep this metadata field private. Private fields are not included in exported images.

The display name of a metadata field can be changed by clicking on it. Drag-and-drop the rows to change the display order.

To add new metadata fields to the module, press the "+" button at the bottom of the dialog. This will allow you to search for the tags you want to add in a new dialog. Tags can be added by either double-clicking on a tag name or pressing the "add" button. Press "done" to update the metadata editor with the new tags. Exif tags which are added in the editor are read from the image file on import . For already imported images they can be read using refresh exif on selected images.

All tags in this module are available for display in the image information module, updating in the import module and filtering in the collections module.

The field values can be used in variable substitution using the full case-sensitive Xmp tag name as the variable name (e.g. `Xmp.iptc.location`).

8.3.5.10. neural restore

Apply AI-powered restoration to selected images: raw denoise, RGB denoise, and 2x / 4x upscaling. Available in both the lighttable and darkroom views as a panel with a tabbed task picker, an interactive before/after preview, and a process button that runs the chosen task on every selected image. Outputs go to a new file – the original image is never modified.

The AI features this module drives are configured in AI preferences ; see the AI features overview for what these tasks do and don't do.

Neural restore can be significantly sped up with GPU acceleration .

prerequisites

Before you can use this module:

1. **Enable AI features** in AI preferences . The master switch is off in a fresh install.
2. **Activate a model for each task you want to use.** In the same preferences tab, tick the *enabled* column for one model per task – `rawdenoise` for the raw denoise tab, `denoise` for the denoise tab, `upscale` for the upscale tab. If no model is enabled for the current tab, the preview area shows *model not available* and the process button refuses to run.

Switching tabs without an active model for the new task is harmless – the module just disables itself for that tab until you enable one.

tabs

The notebook at the top of the module selects the task to run.

raw denoise

AI denoise applied early in the raw pipeline. The exact stage depends on the sensor:

- **Bayer sensors** are denoised directly on the CFA mosaic, with denoise and demosaic combined into a single inference pass. The result is saved as a Bayer CFA DNG.
- **X-Trans sensors** (and any other non-Bayer color CFA the Bayer model can't handle) are demosaicked first by darktable's pipeline, then the AI denoiser operates on the resulting linear Rec.2020 RGB image. The result is saved as a LinearRaw DNG.

Either output can be re-imported into darktable as a regular raw – downstream darkroom processing is unchanged.

Limitations. Monochrome sensors are not currently supported – the available raw-denoise models are trained on color-CFA data and have no monochrome equivalent yet. Use *denoise* in neural restore module on a monochrome image instead.

- *strength* – linear blend between the source raw (0%) and the AI-denoised output (100%) at the raw sensor level. Lower values let some of the original noise back in, in direct proportion.

denoise

Machine-learning RGB denoise applied to the already-demosaicked and mostly-edited image; writes a TIFF.

- *strength* – at 100% the output is the full AI model result. Lower values do not just blend back the source – the difference between source and denoised is decomposed into wavelet (DWT) detail bands and the higher-frequency texture is brought back selectively, so you recover grain and surface detail without re-introducing the color noise the model removed.

upscale

Super-resolution to 2x or 4x the input pixel dimensions. Writes a TIFF at the upscaled size. 4x on a high-resolution image is memory-hungry; tile processing is enabled automatically to keep peak memory bounded, at the cost of some extra processing time.

- *scale* – output magnification factor applied to each side of the image (*2x* or *4x*). *2x* doubles the width and height, producing four times as many pixels overall; *4x* quadruples each side, producing sixteen times as many. Larger factors are slower and need more GPU memory.

Note: 4x upscaling is significantly more demanding on the model than 2x, since it must invent 16x as many pixels of new detail as the original. Fine textures and high-frequency content may be less convincing than at 2x – use 2x when it meets your resolution requirements.

where the tasks sit in your workflow

raw denoise and *denoise* both remove sensor noise – the difference is **at which point in the workflow** you decide to apply them, and you normally use one *or* the other, not both:

- reach for **raw denoise** when you already know the shot is noisy (high ISO, deep push, low light) and haven't started editing. It removes noise at the source, on the raw data, which is where a model can remove it most cleanly – so it gives the strongest result on difficult captures. You commit to it up front and then edit the cleaned DNG.
- reach for **denoise** when you have already developed an image and noise is bothering you in the *result*. It cleans the noise you can actually see in your finished edit, without making you redo that edit.

raw denoise – *before* darkroom editing

Operates on the source raw (Bayer CFA, or X-Trans / non-Bayer demosaicked to linear Rec.2020) and writes a DNG. The intended workflow is:

1. select the noisy raw in lighttable;
1. run *raw denoise* – the output DNG is grouped with the source;
1. open the new DNG and edit it normally.

As it works with the original sensor data it removes noise most cleanly, before demosaic spreads it across the color channels. It is the AI counterpart to the classical raw denoise module – both operate on the raw CFA data before demosaic. The original raw file stays untouched.

denoise – *late* in the workflow, *after* darkroom editing

Exports your image through the full darkroom pipeline (every active module is applied) and then runs the AI denoiser on the result, writing a TIFF. The intended workflow is:

1. develop the image up to and including the step that switches the pipeline from *scene-referred* to *display-referred* space (*filmic rgb* , *sigmoid* or *AgX*) and any display-referred color grading – the point where the noise visible to the viewer has emerged;
1. run *denoise* – the output is a TIFF with the edit so far baked in, plus noise cleaned up;
1. continue editing the TIFF if you want to add sharpening, local adjustments, output sizing, etc., or deliver it directly.

This cleans exactly the noise visible in your finished edit, without redoing any work – your edit is baked into the TIFF and you carry on from there. The trade-off: the noise is already shaped by the whole pipeline, so on a very noisy capture *raw denoise* gives a cleaner result. The TIFF is fully editable – a clean midpoint, not necessarily the last stop.

upscale – *last step before delivery*

Runs the full darkroom pipeline, then enlarges the result 2x or 4x and writes a TIFF. The intended workflow is:

1. finish your edit, including any sharpening and grading;
1. run *upscale* at the chosen factor – the output is a TIFF at the upscaled size;
1. use the output for print or large-format display.

Don’t upscale early in the workflow: any subsequent denoise, sharpening or grading would then operate on the synthesised pixels rather than the originals, and the output quality suffers. Typical uses are enlarging a small crop for a large print, or pulling a low-resolution source up to web-deliverable size.

preview

Below the tab controls is a split before/after preview of a small patch of the currently-displayed image. Drag the divider to wipe between the original (left) and the AI-restored output (right). The preview is generated on demand by clicking the preview area; it uses the same model and settings the process button would use, so it’s a faithful guide to what the final output will look like.

To choose a different patch of the image, click the picker button (to the right of *process*) and then click within the preview widget – the preview re-generates centred on the point you clicked.

Hover over the preview to get a 2x magnified tooltip view, useful for evaluating per-pixel noise / detail behavior.

process

The *process* button runs the active task on every currently-selected image, in turn. Progress is reported in the status bar; each finished image lands in the output folder and (if *add to the current collection* is enabled) is imported into the database automatically.

If no model is active for the current task, the button refuses to run (the preview shows *model not available*). If the model is active but its execution provider fails to initialise – or fails on one specific image at runtime – the affected image is skipped with a message in the status bar, and the rest of the batch continues.

output parameters

The collapsible *output parameters* section controls how the result is written to disk. Settings apply to whichever task is currently active.

bit depth *(denoise and upscale only)*

Bit depth of the output TIFF (*8 bit*, *16 bit*, *32 bit (float)*). profile *(denoise and upscale only)*

Color profile embedded in the output TIFF. *image settings* uses the working profile of the source image. The other entries match the standard export dialog.

compression *(denoise and upscale only)*

TIFF-compression of the output-file. Offers *uncompressed*, *deflate* and *deflate with predictor* (default). The default reduces file size by about 50-60%.

preserve wide-gamut colors *(denoise only)*

When on, pixels whose color falls outside sRGB gamut pass through the model unchanged – wide-gamut colors are preserved exactly, but those specific pixels are not denoised. When off, every pixel is denoised but wide-gamut colors may be clipped to sRGB.

add to the current collection

Import the output into darktable automatically when it's written. The new image is grouped with the source image so they appear together in darktable.

output folder

Where the output file is written. Supports darktable variables – `$(FILE_FOLDER)` (the default) writes next to the source image. Use the folder-icon button to pick a directory.

performance

Processing time depends on the AI acceleration backend, the image resolution, and the task. Raw denoise and 4x upscale are the most demanding. The first inference run after starting darktable also pays a one-time graph-compilation cost – see GPU acceleration , how AI works , and performance notes for the details.

8.3.5.11. recently used collections

Display a list of recently used collections generated by the collections module.

Click on an entry to reopen the selected collection. This also updates the collections module with the appropriate filter criteria and rules.

preferences

The "preferences..." option in the presets menu allows you to adjust the behavior of the recent collections module as follows:

number of recent collections to be stored

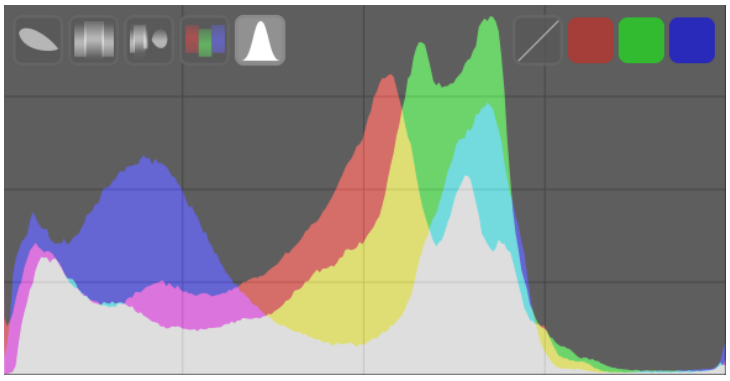
The number of recent collections to store and display in the recent collections module (default 10)

prefer a history button in the collections module

Set whether you prefer to use this module or the history button in the collections module.

8.3.5.12. scopes

This module provides various graphical depictions of the developed image's light levels or chromaticity .



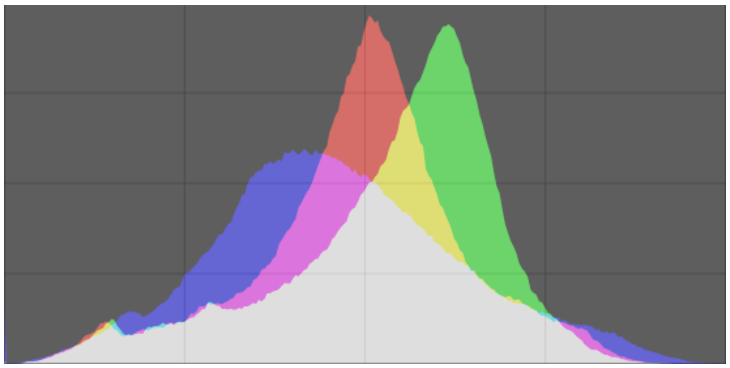
Hover the mouse over the module to reveal buttons that can be used to adjust the display. The buttons on the left-hand-side are used to choose the display mode, from left-to-right: vectorscope, waveform, split (waveform/vectorscope), rgb parade and histogram. The buttons on the right-hand-side control how the plot for the current scope is drawn.

The height of the scopes module can be changed by clicking-and-dragging on the bottom edge or by holding down Shift+Alt while scrolling the mouse. The visibility of the module can be toggled with a keyboard shortcut (default Ctrl+Shift+H).

The scopes module is at the top of the right-hand panel by default but can be moved to the left-hand panel with preferences > miscellaneous > position of the scopes module .

For performance reasons, scopes are calculated from the image preview (the image displayed in the navigation module) rather than the higher quality image displayed in the center view. The preview is calculated at a lower resolution and may use shortcuts to bypass more time-consuming image processing steps. Hence the display may not accurately represent fine detail in the image, and may deviate in other ways from the final developed image.

histogram



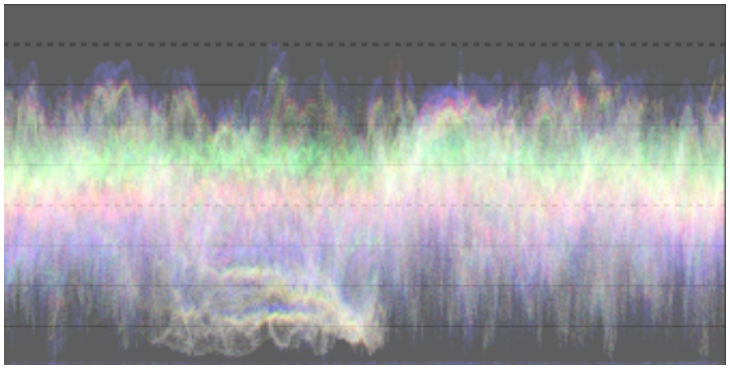
The histogram shows the distribution of pixels by lightness for each color channel.

The horizontal axis runs from 0% to 100% lightness for each channel. The vertical axis gives the count of pixels having the given lightness.

In its default state, data for all three RGB color channels is displayed. The red, green and blue colored buttons on the right-hand-side can be used to toggle the display of each color channel.

The first button on the right-hand-side toggles between logarithmic and linear rendering of the vertical axis (pixel count) data.

waveform



The waveform scope shows similar data to the histogram, but allows you to view that data in a spatial context.

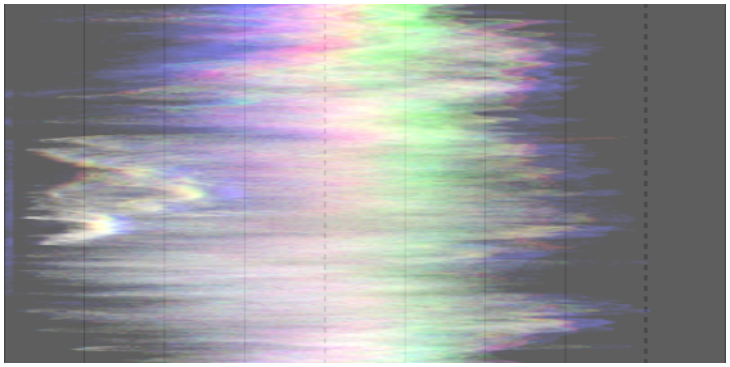
In the “standard” horizontal waveform, the horizontal axis of the waveform represents the horizontal axis of the image – the right-hand side of the waveform represents the right-hand side of the image and the left-hand side of the waveform represents the left-hand side of the image.

The vertical axis represents the distribution of pixels by lightness for each channel – the dotted line at the top represents 100% lightness (values above this may be clipped), the dotted line in the middle represents 50% lightness, and the bottom of the waveform represents 0% lightness.

The brightness of each point on the waveform represents the number of pixels at the given position (the horizontal axis) having the given lightness (the vertical axis). The brighter the point, the more pixels at that position have that lightness.

As with the histogram, you can selectively display each of the red, green, and blue channels, by clicking on the appropriate buttons.

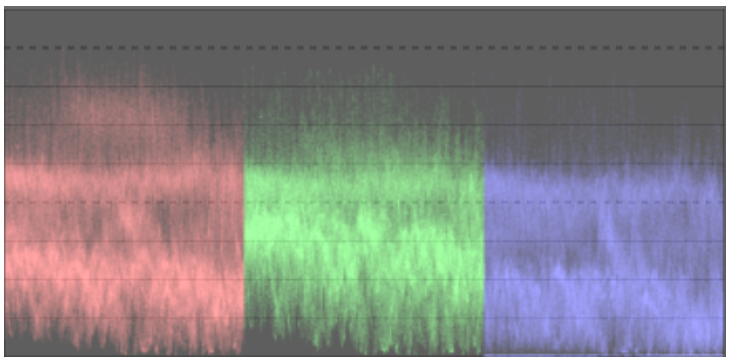
The first button on the right-hand-side toggles the display between horizontal (above) and vertical (below) mode:



In the vertical waveform, the vertical axis of the plot represents the vertical axis of the image, and the horizontal axis represents the distribution of pixels by lightness. The vertical waveform can be useful for portrait-format images, or simply to understand an image in a different way.

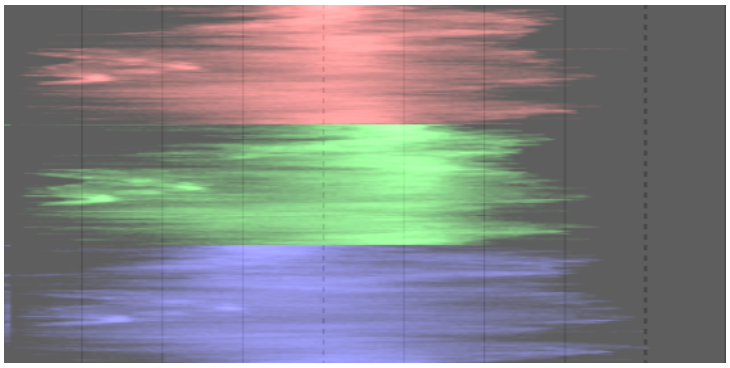
See Of Histograms and Waveforms for more on darktable’s waveform scope.

rgb parade



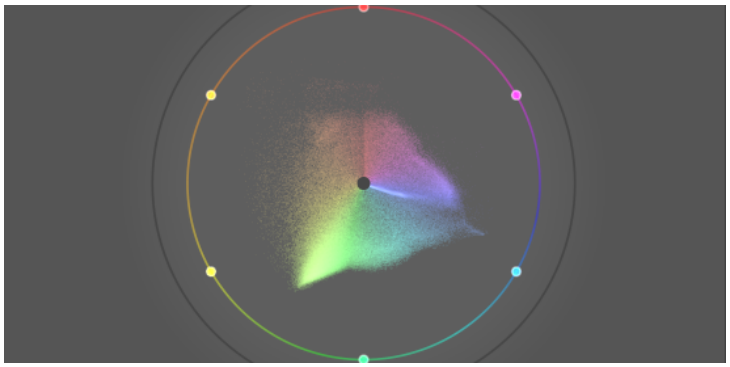
The RGB parade scope shows the same data as the waveform, but with the red, green, and blue channels presented side-by-side.

As with the waveform, clicking the button on the right-hand-side of the module toggles between horizontal (above) and vertical (below) mode:



The RGB parade can be useful for matching the intensities of the red, green, and blue channels. It can also help to understand the differences between, and individual qualities of, each channel.

vectorscope

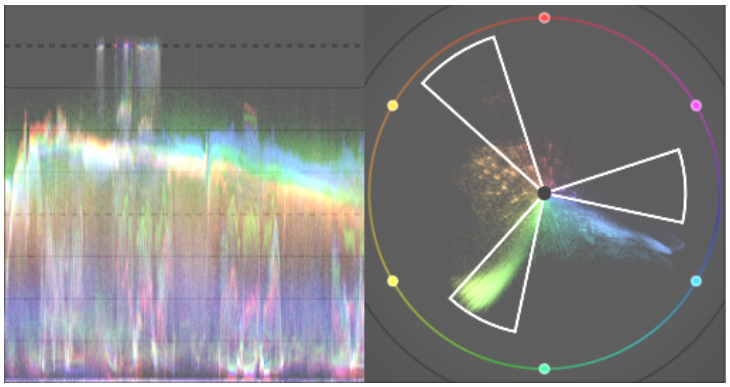


The vectorscope shows chromaticity without regard to either lightness or spatial data.

The distance from the center of the graph represents chroma and the angle represents hue. Areas of the graph are colored depending on the chromaticity of the color to which they correspond in the image. The graph represents color “volume” by rendering the more frequently used colors in lighter tones.

The graph includes a “hue ring” representing the maximum chroma of each hue (in bounded RGB) of the current histogram profile. The RGB primary/secondary colors are marked by circles around the edge of the hue ring.

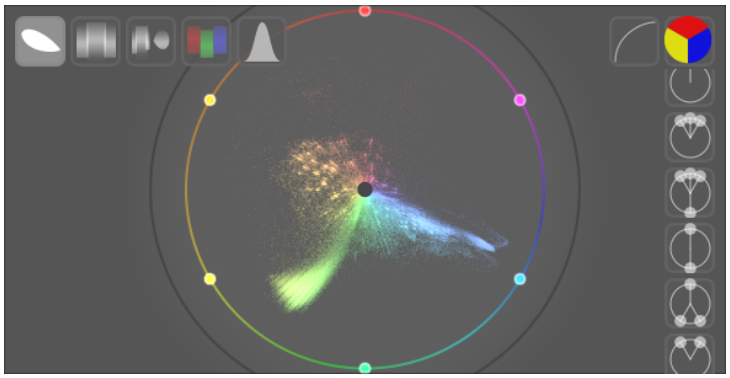
split (waveform/vectorscope)



Shows a combined mode with the waveform on the left and the vectorscope on the right. The available controls are as described for the individual modes.

color harmony and additional vectorscope controls

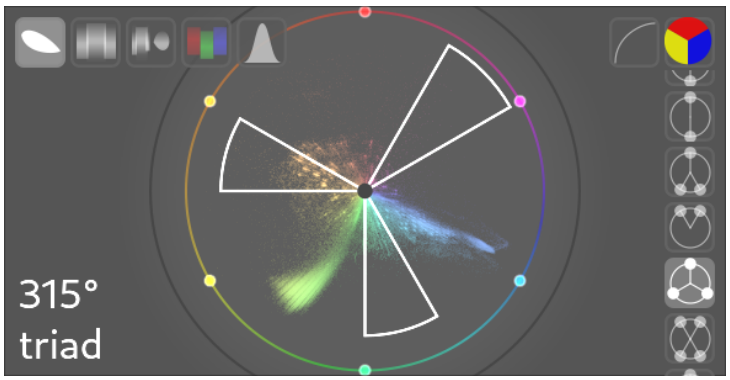
The vectorscope provides some additional controls beyond those provided by the other modes, which deserve separate discussion. Hovering over the scopes module while in vectorscope mode shows the following additional buttons:



Click the second-to-right-most button to toggle the chroma scale between linear and logarithmic mode.

Click the right-most button to toggle the color space of the vectorscope between CIELUV , JzAzBz or RYB – as previously mentioned, these color spaces exclude any luminosity component in this module. The CIELUV graph will be faster to calculate, and is a well-known standard, though JzAzBz may be more perceptually accurate.

Finally, along the right-hand edge of the module are a series of buttons that allow the selected color harmony indicator to be overlaid on the vectorscope graphic when in RYB color space. For example, the following shows the “triad” color harmony:



Rotate the display of the overlaid harmony guides by hovering over the scopes module and scrolling with the mouse scroll wheel. Hold Ctrl while scrolling to rotate more slowly. Hold Shift while scrolling to change the area covered by the guide overlays.

The type and position of the chosen harmony guides is saved alongside each image's editing data (in the XMP file and database) so can be restored whenever you return to the image for further editing.

A full description of how to use this functionality is out of scope of this reference section, however, the following overview gives a basic guide as to how to use this mode to improve the color harmonies within the image.

1. With the global color picker , take live samples of the key colored areas of the image. The choice of these is purely artistic, depending on the feeling you wish to convey. Choose (in the global color picker) to display the chosen samples on the RYB vectorscope.

2. Choose the color harmony type that you wish to implement (or the closest to the actual distribution of picked live samples). Rotate (with Ctrl+scroll) until the picked samples fall roughly inside (or near) the harmony guides.

3. If some of the picked samples fall outside of the guide lines, perform manipulations of the image's colors until they fall within the guides. You can use any processing module to achieve this but multiple masked instances of *color balance rgb* usually provide the best results.

Bear in mind that these controls are provided as a basic guide to achieving color harmony – do not try to force colors into the guides at the expense of the overall look of the image. For some useful examples of how to use this functionality, please see the pull request that implemented the original change.

caveats

- The hue ring is not a gamut check , as a color can be within the hue ring, yet out of gamut due to its darkness/lightness.
- When adjusting an image based upon a color checker, faster and more accurate results will come from using calibrate with a color checker in the *color calibration* module.
- The vectorscope does not have a "skin tone line", which is a flawed generalization rather than a universal standard.
- The vectorscope represents a colorimetric encoding of an image, which inevitably diverges from a viewer's perception of the image.

exposure adjustment

The histogram, waveform, and RGB parade scopes can be used to directly alter the *exposure* and *black level* of the *exposure* module.

For the histogram, click towards the right-hand side of the histogram and then drag right to increase or drag left to decrease the exposure. In a similar manner you can control the black level by clicking and dragging in the left-hand side.

For horizontal waveform and RGB parade, drag the upper portion of the scope up/down to increase/decrease exposure. Drag the lower portion up/down to increase/decrease the black level.

For vertical waveform and RGB parade scopes, the corresponding regions are to the right (exposure) and left (black level).

You can also scroll in the appropriate area – rather than dragging – to adjust exposure and black point.

Finally, you can override the default adjustment behavior as follows:

- hold Ctrl+Shift while adjusting to override the soft limits (e.g. to significantly overexpose the image)
- hold Ctrl while adjusting to fine-tune the adjustment. Alternatively, right-click to enter a precise value for the adjustment
- double-click on the desired area to reset to the original value

histogram profile

Image data is converted to the *histogram profile* before the scope data is calculated. You can choose this profile by right-clicking on the soft-proof or gamut check icons in the bottom module and then selecting the profile of interest. When soft-proof or gamut check is enabled, the scope is shown in the soft proof profile.

As the scopes module runs at the end of the preview pixelpipe, it receives data in display color space. If you are using a display color space that is not "well behaved" (this is common for a device profile), then colors that are outside of the gamut of the display profile may be clipped or distorted.

8.3.5.13. styles

Create named styles from selected images' history stacks and apply styles to selected images.

Styles can either be created within this panel or in the history stack module in the darkroom.

A list of all available styles is displayed in the module. A search field above the list allows you to locate a style by name or description. This module also allows styles to be edited and deleted.

Double-click on a style name to apply that style to all selected images. A style may also be applied to all selected images by pressing a shortcut key assigned to it (see preferences > shortcuts) while in the lighttable or darkroom view.

Hover over the style name with your mouse to view a preview of the first selected image with that style applied.

Remove all styles by clicking on the module's reset button.

Please note that styles also include the active state of each module. You can use this to create your own default settings, which you can then activate on-demand. Simply set your desired defaults for each module, disable the module, and save the style.

A large selection of "camera" styles is provided with darktable. These styles are designed to approximate the out-of-camera JPEG look for supported camera models and can be found within this module under the "darktable camera styles" style group. You can automatically apply these styles on import using the companion Lua script official/camera style .

module controls

hide preview

Use this checkbox to hide the preview on the tooltip entirely. This can be useful on slower computers, where rendering the preview may take additional time.

create duplicate

When applying a style to images, tick this box to create a duplicate of each image before applying the chosen style to that duplicate. Disable this option to apply the chosen style directly to the selected images. Beware that in this case any existing history stack is overwritten (depending on the mode – see below) and cannot be recovered.

mode

As with the history stack module, this combobox allows you to either "append" the style to the current history stack or to "overwrite" the history stack of the target image.

create

Create new style(s) using the history stack of the selected image(s). For each selected image a style creation window is shown. You must supply a unique name for each new style and you can also add an optional description.

You will be provided with the option to select which history stack items you want to include in the created style. For any module, you may also choose to "reset" that module's parameters – this will cause the module to be included in the style but with all controls set to their initial (default) state (as if you had clicked the module reset button).

The panel supports a hierarchical view, meaning that you can create categories using the pipe symbol "|" as a separator. For example "print|tone curve +0.5 EV" will create a "print" category with a style of "tone curve +0.5 EV" below it.

You can cancel the creation process by pressing the Cancel button or pressing the Esc key on your keyboard.

edit

Select "edit" to bring up a dialog which allows you to edit an existing style. The upper portion of the dialog shows modules currently included in the style, the lower portion all modules available from the source images history stack. For the latter to appear *one* image has to be selected – ff no or multiple images are currently selected no source history stack will be offered.

You can add or remove modules and pick modules to update from the source history stack by setting the respective checkboxes.

Check the "duplicate" option if you want to create a new style, instead of overwriting the existing one, in which case you will need to provide a unique name for the new style.

remove

Delete the selected style, without any further prompt.

import

Import a previously saved style. Styles are stored as XML files with the extension `.dtstyle`. If a style with the same name already exists, you will be asked if you wish to overwrite.

export
Save the selected style to disk as a `.dtstyle` file. This allows styles to be published and shared with other users.

preferences

The “preferences…” option in the presets menu allows you to adjust the following settings:

preview size
Set the size of the preview on the tooltip. The available options are: default and large.

8.3.5.14. tagging

Manage tags attached to images.

Tags provide a means of adding information to images using a keyword dictionary. You can also manage tags as a hierarchical tree, which can be useful when their number becomes large.

Tags are physically stored in XMP sidecar files as well as in darktable’s library database and can be included in exported images.

definitions

The following definitions assume that you have set up a single tag named “`places|France|Nord|Lille`”.

tag
A tag is a descriptive string that may be attached to an image. A tag can be a single term or a sequence of connected terms forming a path, separated by the pipe symbol. For example, “`places|France|Nord|Lille`” defines a single tag, where each term in the path forms a smaller subset of those before it. You can attach as many tags to an image as you like.

You can assign properties (name, private, category, synonyms and image order) to a tag.

You can use tag names to define collections that are displayed, processed, and exported in a *durable* custom sort order, which does not change when other collections (containing some of the same images) are reordered.

node
Any path that forms part of a tag is a node. In the above example, “`places`”, “`places|France`”, “`places|France|Nord`” and “`places|France|Nord|Lille`” are all nodes. In the hierarchical tree view, the nodes form the branches and leaves of the tree.

free node
Any node that is not explicitly defined as a tag is called a free node. In the above example, “`places`”, “`places|France`” and “`places|France|Nord`” are all free nodes. You cannot set any properties, except “name”, for a free node and you cannot add a free node to an image. See the “multiple tags” section below for more information.

category
Any tag can be flagged by the user as being a “category”.

multiple tags

The above definitions considered a simple example – a single tag and its properties. Consider instead the scenario where the following four pipe-delimited tags are each separately defined in darktable.

```
places|France|Nord|Lille
places|France|Nord
places|France
places|England|London
```

In this case the nodes are

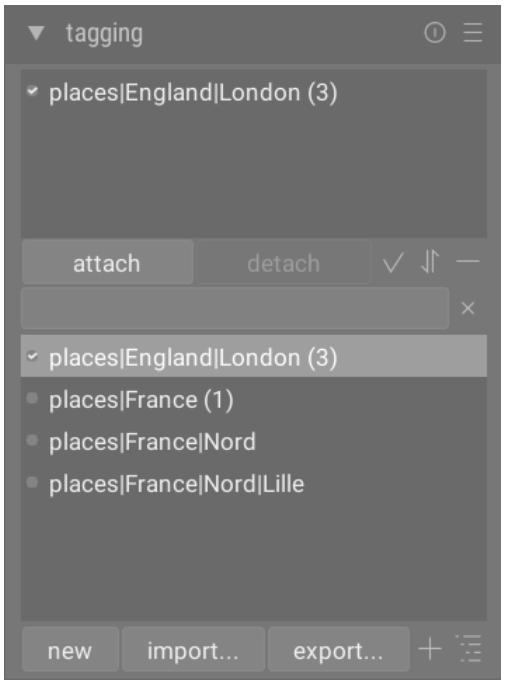
```
places
places|France
places|France|Nord
places|France|Nord|Lille
places|England
places|England|London
```

The only free nodes are “places” and “places|England”. Both of these free nodes are also (by implication) categories.

You can attach any of these tags to any image. Any tags attached to an image, except category tags, can be included when that image is exported .

If you attach the “places|France|Nord|Lille” tag to an image, the “places|France|Nord” and “places|France” tags are also implicitly attached to that image (you don’t need to attach them manually). Note that this is only true here because those additional tags have been separately defined – the “places” node is not included because it is a “free node” (not a tag).

module sections



The tagging module consists of two sections

1. The upper *attached tags* section (with the *attach/detach* buttons under it)
2. The lower *tag dictionary* section (with the *new/import.../export...* buttons under it)

attached tags section

The *attached tags* section displays tag(s) attached to image(s), where those images are

- under your mouse cursor (if hovering over an image in the lighttable view); or
- currently selected (if not hovering over an image)

Automatically generated darktable tags (with names starting “darktable|”) cannot be selected.

At the bottom of the *attached tags* section are the following buttons, from left to right:

attach
If a tag is selected in the *tag dictionary* section, attach this tag to the selected images.

detach
If a tag is selected in the *attached tags* list, detach that tag from the selected images. A tag can also be detached if you right click on the tag name and select *detach* from the pop-up menu.



hidden tags
Choose whether to view any hidden tags that darktable has automatically attached to the selected images.



sort
Choose whether to sort the *attached tags* list alphabetically or by the count shown in brackets next to the tag (this count indicates how many of the selected images have that tag attached to them).



parents
Choose whether or not to show the parent categories of the tag.

You can adjust the height of the *attached tags* window by holding Ctrl and scrolling with your mouse wheel.

tag dictionary section

The *tag dictionary* section displays all of the tags that are available in darktable's database. At the top of the *tag dictionary* section is a text box where tag names can be entered. Below this is a list of available tags, which may also include indicator symbols to the left of the tag names. The meanings of these symbols are as follows:

- a check mark [✓] indicates that the tag is attached to all of the selected images
- a minus sign [–] indicates that the tag is attached to at least one of the selected images. If the symbol is next to a node name in the hierarchical *tree* view, it means that one of the child tags under that node is attached to at least one of the selected images.
- if no indicator symbol is present, this means that the tag is not attached to any of the selected images, or that the node has no child tags attached to any of the selected images.

In the hierarchical *tree* view, a name in italics represents either a free node or a category.

Below the *tag dictionary* list are the following buttons, from left to right:

new
Create a new tag, using the name that has been entered into the text entry box at the top of the *tag dictionary* section.

import...
Import tags from a Lightroom keyword file.

export...
Export all tags to a Lightroom keyword file.



suggestions
Show a list of suggested keywords based on the keywords already associated with the selected images (see the preferences section). CAUTION: This view queries the database so might be slow.



list/tree
Toggle the display of tags between the straight *list* view and hierarchical *tree* view.

You can adjust the height of the *tag dictionary* window by holding Ctrl while scrolling with your mouse wheel.

usage

The following sections describe the operations that can be performed with tags.

text entry

The text entry box (shown under the *attach*/*detach* buttons) has multiple purposes.

- If the *tag dictionary* list is in *list* view mode (rather than *tree* view mode), then typing the first few characters of a tag will bring up a list of suggestions. You can then scroll down with the arrow keys and press Enter to select one of the suggestions. Pressing Enter a second time will attach it to the selected images. You can also edit the name of the tag before pressing Enter – in this case the tag will be created if it doesn't already exist in the database.
- Typing some partial text into the text entry box allows you filters the set of tags shown in the *tag dictionary* window to those whose name or synonym matches the entered text. Press Enter to attach a tag with the entered name to the selected images. If that tag name does not yet exist in the database, it will be created before being attached.
- The pop-up menu entry "copy to entry" can be used to copy a selected tag to the text entry box. You can then edit this name and press Enter to create a new tag with that name, making it easy to create multiple tags with similar names.

create tag

There are several ways to create a new tag:

- *Import a text file*. You can import one or more text files in the Lightroom tagging file format. You can also export your tags, edit the exported file, then re-import it. The import function updates existing tags and creates new tags as required. If you change the name of a tag in an imported file, it will be treated as a new tag.
- *Import already-tagged images*. This method does not offer any flexibility to change tag names or categories during the import process.
- *Use the "create tag" sub-menu*. A tag can be created manually, under an existing one (hierarchical) or at the root level.
- *Use the "set as a tag" sub-menu*. You can set a free node (e.g. "`places|England`") as a tag so that it gets implicitly attached to all images tagged with its sub-tags (e.g. "`places|England|London`").
- *Type into the text box and press the "new" button* or the Enter key. Hierarchical tags are created using the pipe symbol "`|`" to separate nodes. Note that the entered tag is also attached to any selected images.

A number of tags are automatically generated by darktable when certain actions are undertaken. For example, the tags "`darktable|exported`" and "`darktable|styles|your_style`" can be used to identify images that have been exported and had styles applied, respectively.

edit/rename tag

The *tag dictionary* can be maintained via the “edit…” and “change path…” items in the right-click pop-up menu.

The “edit…” operation allows you to change the name of a tag, though you cannot change which node it belongs to (you cannot use the pipe “|” symbol in the tag name field). The command is aborted if you try to enter a tag name that already exists. You can set the *private* and *category* flags and define *synonyms* for the tag (see below). These attributes are recorded in the `XMP-dc Subject` and `XMP-lr Hierarchical Subject` metadata entries, respectively. You can control which tags are included in exports by changing settings in the export module.

- A tag set as “private” is, by default, not exported.
- A tag set as “category” is not exported in `XMP-dc Subject`. However it is exported in `XMP-lr Hierarchical Subject` as this XMP metadata holds the organization of your tags.
- “synonyms” enrich the tag information and are mainly used to assist search engines. For example “juvenile”, “kid” or “youth” can be set as synonyms of “child”. Synonyms can also be used to translate tag names to other languages.

The “change path…” operation is only available in the *tree* view mode, and it shows the number of tagged images which would be impacted by a change to the name of this node. The change path window lets the user change the full path of the node, including the nodes to which it belongs (nodes can be specified using the pipe “|” symbol). This operation is powerful, but please take care as it can have a significant impact on the metadata of your images. The operation is aborted if the requested change causes a conflict with an existing tag.

A quick way to organize the tag structure is to drag and drop the nodes. In the *tree* view mode, you can drag any node and drop it on top of any other node. The first node and its descendants, if any, become descendants of the second node. Dragging over a node automatically opens that node (to avoid opening a node, drag over the node selection indicator instead). To place a node at the root level, drag it onto the top of the tagging window. If the requested change causes a conflict with an existing tag, the operation is aborted.

attach tag

There are a number of ways to attach an existing tag to a group of selected images:

- click on a tag in the *tag dictionary* window to select it, then click on the *attach* button.
- right-click on a tag in the *tag dictionary* window, to show a pop-up menu, then select the “attach tag” menu item.
- double-click on a tag in the *tag dictionary* window.
- right-click on a tag shown in the *attached tags* view to show a pop-up menu. If some of the selected images do not currently have that tag, the “attach tag to all” menu item can be used to attach that tag to all the selected images.
- Type into the text box and press the “new” button or the Enter key. This will create the tag if it doesn’t already exist, and attach it to the selected images.
- Press Ctrl+T to open a small text box at the bottom of the central view of the lighttable. Type in the name of a tag and press Enter. The tag will be created if it doesn’t exist, and attached to all the selected images.
- Drag an image or group of images and drop it onto the desired tag.

When hovering over the images in the lighttable you can check which tags are attached to the image, either by looking at the *attached tags* window in the *tagging* module, or in the *tags* attribute in the image information module.

detach tag

There are several ways to remove a tag from a group of selected images:

- click on a tag in the *attached tag* window of the *tagging* module to select the tag, then click on the *detach* button.
- double-click on a tag in the *attached tag* window.
- right-click on a tag in the *attached tag* window, to show a pop-up menu, and select “detach”.

delete tag

It is possible to completely remove a tag from all images (whether selected or not) and delete the tag from the database. Because this could impact a large number of images, a warning will be displayed indicating how many images currently have this tag attached. Take this warning seriously as there is no way to undo this action (except by restoring your database and/or XMP sidecar files from a backup). A tag in *tag dictionary* window can be deleted in the following ways:

- right-click on a tag in the *tag dictionary* window, to show a pop-up menu, and choose “delete tag”.
- right-click on a branch node in the *tag dictionary* window, to show a pop-up menu, and choose “delete node” to delete the selected node together with any child nodes.

import / export

The “import” button allows you to choose a text file (which must comply with the Lightroom tag text file format) and import its content. If a tag in the imported file already exists, its properties will be updated, otherwise a new tag will be created.

The “export” button exports your entire tag dictionary into a text file of your choice (Lightroom tag text file format).

copy tag to clipboard

It is possible to copy a single selected tag to the clipboard from both *attached tags* and *tag dictionary* sections by using the pop-up menu entry “copy to clipboard”.

find tag

If you right-click on a tag name in the *attached tags* section and click *find tag*, then the *tag dictionary* will show all matching tags.

keyboard

The following keys can be used within the tagging module:

- The Tab key can be used to navigate between the *attached tags* view, the text entry box, and the *tag dictionary* view. Pressing Tab from within the text entry field selects the first matching tag in the *tag dictionary* view (if any).
- The Down arrow key is equivalent to the Tab key, when pressed while inside the text entry field.
- Shift+Tab does the same as the Tab key but navigates in the opposite direction. Pressing Shift+Tab from within the text entry field selects the first user tag in the *attached tags* view (if any).
- The Enter key can be used within the *tag dictionary* view to attach the selected tag, keeping focus on the attached tag (similar to when a tag is attached using the mouse).
- Pressing Shift+Enter from within the *tag dictionary* view returns focus to the text entry field.
- The Delete / BackSpace keys can be used within the *attached tags* view to detach the selected tag.

navigation

To see the images bearing a particular tag in the *tag dictionary* window, right-click on the tag name and choose “go to tag collection” in the resulting pop-up menu. This opens a collection in the collections module showing all images containing this tag. You can also select other tags in the *collections* module by double-clicking on the collection for the other tag.

To return to the collection that was selected before opening a tag collection select the “go back to work” item from the pop-up menu. This will allow you to return to the original collection, as long as you haven’t selected any other collections in the meantime.

preferences

The “preferences…” option in the presets menu brings up a dialog where you can tweak the behavior of the tag suggestions list. The tag suggestions list comprises two parts, with the best-matched tags on one side and the most-recently-attached tags on the other. The following options are available:

suggested tags level of confidence

Level of confidence used to include the tag in the suggestions list (default 50):

- 0: display all associated tags,
- 99: match tags with a 99% confidence level,
- 100: this is essentially an unreachable level of confidence and so returns no matching tags. Use 100% to disable the best-matched tag suggestion list (faster).

number of recently attached tags

Number of recently attached tags to include in the suggestions list (default 20). A value of “-1” can be used to disable the most-recently-attached suggestions list.

8.3.6. tethering

8.3.6.1. camera settings

Set up an image capture job.

This can include sequence, bracket and delayed captures. You are also able to control other camera settings such as focus mode, aperture, shutter speed, ISO and white balance.

8.3.6.2. live view

Control your camera’s live view mode.

Functionality such as focus control, rotation, guides and overlays are supported.

8.3.6.3. session

Set a session jobcode to allow the creation of a new film roll.

A session is a sequence of exposures taken in tethering mode and placed into a single film roll. A new session means the creation of a new film roll.

The session module allows you to set a jobcode, which can be used to create your new film roll. See preferences > import > session options for details about how to create new film rolls using the `$(JOBCODE)` variable in the session module.

9. Preferences & Settings

9.1. overview

darktable comes with a number of user-configurable preferences. These can be adjusted in the preferences dialog, which can be reached by clicking the gear icon on the right of the top panel . Preferences are separated into tabs, each of which is described in more detail in the following sections.

When opened from within the lighttable or darkroom views, the appropriate tab will be loaded by default to allow you to modify that view's settings.

Any altered settings (i.e. those that differ from their default state) are highlighted with a bullet beside them. If you change a setting that needs a restart to take effect, a message will appear as a reminder after you exit the preferences dialog.

Double click on a setting's label to reset to its default value.

The preferences dialog can be closed by clicking on the close button in your window manager or pressing the Escape key.

9.2. backup settings & edits

If you want to back up darktable's settings and edits you can save a copy of the contents of the configuration directory. See configuration directory for descriptions of what the individual files contain.

Transferring the contents of this directory also allows for migration between variants (e.g. between Flatpak and AppImage) of darktable.

Darktable also maintains a number of backups of `data.db` and `library.db` inside the configuration directory: 15 daily snapshots of each, plus a snapshot created when updating to a new version (e.g. `data.db-pre-5.4.1`).

If you configure darktable to create sidecar files, your RAW files will also be accompanied by XMP files containing your edits (see preferences/storage).

9.3. general

Control the overall look and feel of darktable.

interface language
Set the language of the user interface. This includes the option to enable sentence case (`en@truecase`) for English. The system default is marked with an * (needs a restart)

theme
Set the theme for the user interface. Aside from any aesthetic considerations, the recommended interface color for color evaluation is middle gray. Visual perception is affected by ambient brightness, and a low user interface brightness causes all kinds of illusions. Using a dark interface to retouch photos can therefore lead to excessive retouching (abuse of contrast and saturation) and to a photo that is too dark when printed. It is therefore highly recommended that you use one of the "grey" themes for retouching work as these are designed so that the user interface approximates middle gray. For those who have difficulty reading text in the default theme, darktable includes two "highcontrast" themes with white text on a dark background, but the caveats mentioned previously apply if you select one of them (default "darktable-elegant-grey").

Some elements of a theme are customizable by these additional preferences:

condensed module controls
When enabled, controls in modules are made smaller and spacing between them is reduced (default off).

rounded module headers
When enabled, one corner (the corner furthest from the application's edge) of each module's header is rounded (default off).

focused module border
When a color is selected, the currently focused processing module has a border of the selected color (default none).

expanded module border

When a color is selected, each expanded processing module has a border of the selected color (default none). If *expand a single processing module at a time* is enabled, then *expanded module border* has no effect as the expanded module is also the focussed module.

accent color

When a color is selected, the on/off button of a module is tinted with the selected color (default none).

use system font size

Select this option to use the font size defined by your system. If unchecked, you may enter a custom font size in the box below (default on).

font size in points

If the “use system font size” option is switched off, enter a font size (in points) for darktable to use. The font size will be changed immediately.

GUI controls and text DPI

Adjust the global GUI resolution to rescale controls, buttons, labels, etc. Increase for a magnified GUI, decrease to fit more content in the window. Set to -1 to use the system-defined global resolution. The default is 96 DPI on most systems. (needs a restart)

reset view panels

The order and visibility of utility modules can be manually changed by the user (see modules overview). Click this button to reset to default values.

CSS theme modifications

In addition to selecting a pre-built theme you can also apply additional CSS customizations of your own to tweak the look-and-feel of darktable.

Two different methods are provided for this:

create a custom theme

If you wish to make a large number of changes to darktable’s UI you may wish to create your own theme (in a `.css` file) and place it in the `themes` folder in the darktable configuration directory . Your new theme will automatically appear in the *theme* selection list the next time you restart darktable.

Please note that the structure of darktable’s internal CSS changes frequently and you may need to make significant changes to your own themes when new versions of darktable are released. For this reason (among others) we do not recommend creating complex custom themes unless you are willing to devote a lot of time to ongoing maintenance. If your theme loads any of darktable’s pre-built themes using the `@import url` directive, note that your CSS theme file may not be portable between installations (`@import url` uses relative paths and the location of the pre-built themes is system-dependent).

create theme tweaks

A text box is provided at the bottom of the general tab within which you can enter your own CSS tweaks. When using this option, darktable will first load your selected theme (the “base” theme, chosen in the *theme* drop-down) and then apply your custom CSS on top. This means that you can easily make minor alterations to the look-and-feel, while still keeping mostly up-to-date with core theme changes when a new version of darktable is released. It also means that you can usually change your base theme without affecting your custom CSS tweaks.

When you have finished entering your CSS, click the “save CSS and apply” button. This will save your CSS to `user.css` in the darktable configuration directory . and immediately apply it to the current darktable session.

If you notice any issues after applying your CSS, you can uncheck the “modify selected theme with CSS tweaks below” box to revert. This will immediately restore the base theme but will leave your tweaks in the editor so that you can re-edit them and try again. Simply press “save CSS and apply” again when you are ready to retry. This will automatically re-check the “modify selected theme with CSS tweaks below” checkbox and apply the new CSS.

Note: If you have any issues while using custom CSS tweaks please retry with the “modify selected theme with CSS tweaks below” option unchecked to be certain that they were not caused by any of your alterations.

---

understanding darktable's themes

All of darktable's pre-built themes are provided as CSS files in `$DARKTABLE/share/darktable/themes/` (where `$DARKTABLE` is darktable's installation directory). The `darktable.css` theme contains the bulk of the code used to control the look-and-feel of darktable. A number of other themes are also provided but most of them use `darktable.css` as a base (by importing `darktable.css` using the `@import url` directive).

If you choose to create your own custom theme file you are advised to follow a similar approach – import one of darktable's existing theme files using `@import url` (this directive expects relative paths) and then apply your own customizations on top. You do not need to do this when using the CSS text box in the preferences dialog – attempting to use `@import url` in the CSS tweaks text box will not work correctly.

Themes use the same basic CSS principles as in html browsers (with some minor exceptions – see the Gtk documentation for details):

- The majority of the style properties are assigned to broad groups of UI elements, for example, Gtk buttons and text entry fields
- Next, related groups of darktable-specific UI elements are given *class names* allowing them to be styled as a group
- Finally, some unique UI elements are assigned a CSS *id* so that they can be styled independently

You are encouraged to explore the existing themes (`darktable.css` in particular is very well commented) and to make use of the Gtk Inspector tool to figure out how to select the specific UI element (or class of elements) you wish to modify. Some experimentation will be required.

Please note that darktable themes are grayscale by default in order that users not be distracted by strong colors while editing images. You are advised to retain this practice in your own themes and to keep the average shade as close to middle-gray as possible. In addition you are advised to review your custom CSS each time darktable is updated, to ensure that changes to the application have not adversely affected your tweaks.

9.4. import

Control default file naming conventions used when importing images.

session options

The following options define the default naming pattern for use in the "copy & import" or "copy & import from camera" options in the import module, or when taking photos in the tethering view.

The naming pattern consists of three parts: a base part defining the parent folder, a session part defining a filmroll (which is specific to the individual import session), and a file name part defining the filename structure for each imported image.

Several pre-defined variables can be used in the pattern as placeholders:

```
$(HOME)              the home folder as defined by the system
$(PICTURES_FOLDER)   the pictures folder as defined by the system (usually "$HOME/Pictures")
$(DESKTOP)           the desktop folder as defined by the system (usually "$HOME/Desktop")
$(USERNAME)          your user account name on the system
$(FILE_NAME)         basename of the imported image
$(FILE_EXTENSION)    extension of the imported image
$(JOBCODE)           unique identifier of the import job
$(SEQUENCE)          a sequence number within the import job
$(MAX_WIDTH)         maximum image width to limit within export session
$(MAX_HEIGHT)        maximum image height to limit within export session
$(ID)                unique identification number of the image in darktable's database
$(YEAR)              year at the date of import
$(MONTH)             numerical month at the date of import
$(DAY)               numerical day at the date of import
$(HOUR)              hour at the time of import
$(MINUTE)            minute at the time of import
$(SECOND)            second at the time of import
$(EXIF_YEAR)         year the photo was taken (from Exif data)
```

```
$(EXIF_MONTH)        numerical month the photo was taken (from Exif data)
$(EXIF_DAY)          numerical day the photo was taken (from Exif data)
$(EXIF_HOUR)         hour the photo was taken (from Exif data)
$(EXIF_MINUTE)       minute the photo was taken (from Exif data)
$(EXIF_SECOND)       seconds the photo was taken (from Exif data)
$(EXIF_ISO)          ISO value of the photo (from Exif data)
```

base filmroll's directory naming pattern
The base directory part of the naming pattern (default `$(PICTURES_FOLDER)/Darktable`).

filmroll naming pattern
The filmroll part of the naming pattern (default `$(YEAR)$(MONTH)$(DAY)_$(JOBCODE)`).

keep original filename
Check this box to keep the original filename instead of using the pattern below when importing from a camera or card (default off).

file naming pattern
The file name part of the naming pattern (default `$(YEAR)$(MONTH)$(DAY)_$(SEQUENCE).$(FILE_EXTENSION)`).

9.5. lighttable

Control functionality in the lighttable view and modules.

general

hide built-in presets for utility modules
If enabled, only user-defined presets will be shown in presets menu for utility modules – built-in presets will be hidden (default off).

use single-click in the collections module
Enable "single click" mode in the collections module, which allows ranges to be selected (default off).

prioritize the hovered image over the selected images
Controls which images are affected by an action (e.g. using the keyboard to set the number of stars). If this option is enabled, only the image hovered over is affected by the action. If disabled, all the selected images are affected (i.e. selection is prioritized) and the hovered-over image is only affected if it is also one of the selected images. (default on)

expand a single utility module at a time
Controls how utility modules are expanded. If this option is enabled, expanding a module by clicking collapses any other currently expanded panel. If you want to expand a panel without collapsing the others you can do so with Shift+click. Disabling this option inverts the meaning of click and Shift+click (default off).

scroll utility modules to the top when expanded
With this option enabled the side panerepresent will scroll a utility module to the top of the panel when it is expanded. (default off)

rating an image one star twice will not zero out the rating
Normally clicking a one star rating twice will set a zero star rating to that image. Check this option to disable this functionality (default off).

show scrollbars for center view
Choose whether to show scrollbars in the center view of the lighttable (default on).

show image time with milliseconds
Choose whether to include milliseconds when displaying time values (default off). If set, milliseconds are shown in the image information module and can also be used in the geotagging module.

thumbnails

use raw file instead of embedded JPEG from size
When generating thumbnails for images that have not yet been processed in the darkroom, if the thumbnail size is greater than this value, generate it by processing the raw image data. If the thumbnail is below this size, use the JPEG preview image embedded in the raw file. Once an image has been processed in the darkroom, thumbnails will always be generated from raw data (you can revert back to the JPEG preview by discarding history). To render thumbnails with the best quality choose "always".

high quality processing from size

If the thumbnail size is greater than this value and is being generated from raw data, it will be processed using the full quality rendering path, which is better but slower (default 720p). To render thumbnails with the best quality, choose “always”.

enable disk backend for thumbnail cache

If activated, darktable stores all thumbnails on disk as a secondary cache, and thereby keeps thumbnails accessible if they are dropped from the primary cache. This needs more disk space but speeds up the lighttable view as it avoids the reprocessing of thumbnails (default on).

enable disk backend for full preview cache

If enabled, darktable writes full preview images to disk in the cache directory when evicted from the memory cache. Note that this can take a lot of storage (several gigabytes for 20k images) and darktable will never delete cached images. It’s safe to delete these manually if you want. Enabling this option will greatly improve lighttable performance when zooming an image in full preview mode (default off).

enable smooth scrolling for lighttable thumbnails

If enabled, use fractional scrolling when navigating through the thumbnails in the filemanager layout for a smoother experience. When enabled, the top row of thumbnails may only show partial images as the scrolling is not aligned with each row. This setting is most useful when scrolling via a touchpad or other input device rather than a mouse with a scroll wheel (default off).

generate thumbnails in background

Choose whether to automatically generate thumbnails in the background when the user is in the lighttable view and is inactive (no keyboard/mouse activity). Thumbnails will be generated up to the selected size (default never).

Note: *You can set the period of inactivity required before thumbnail generation will start by altering the* `backthumbs_inactivity` *setting in* `darktablerc` *in the darktable configuration directory (default 5 seconds).*

reset cached thumbnails

Force thumbnails to be regenerated by resetting the database. This may be needed if some thumbnails have been manually removed or have become corrupted (default off).

delimiters for size categories

Size categories are used to allow different thumbnail overlays to be shown depending on the thumbnail size. A pipe delimited set of values defines at what image sizes the category changes. The default value of “120|400” means that there will be 3 categories of thumbnail: 0-120px, 120-400px and >400px.

pattern for the thumbnail extended overlay text

If you have chosen to show extended overlay text over thumbnail images, this setting allows you to define what information is displayed. This pattern can use any of the variables defined in the variables section. You can also include formatting (bold, italic, colors etc).

pattern for the thumbnail tooltip (empty to disable)

Defines what information is displayed in the tooltip when the mouse hovers over image thumbnails. This pattern can also use the same variables and formatting as the extended overlay text.

9.6. darkroom

Control functionality in the darkroom view and associated modules.

general

scroll down to increase mask parameters

By default, scrolling your mouse up increases the value of the relevant shape parameters in drawn masks . Set this preference to reverse the behavior (default off).

middle mouse button zooms to 200%

When enabled, clicking the middle mouse button on the image canvas causes the zoom level to toggle between fit, 100% and 200%. When disabled, the middle mouse button toggles between fit and 100%. In the latter case, you can use Ctrl+middle-click to access the 200% zoom level (default on).

pattern for the image information line

Set the information to be displayed in the image information line . You can use any variables in the variables section as well as `$(NL)` for a new line. You can also include formatting (bold, italic, colors etc).

position of the image information line

Choose the darkroom panel in which the image information line is displayed. Choose between “top left” “top right” “top center” “bottom” and “hidden” (default “bottom”).

border around image in darkroom mode

Display the center image in darkroom mode with an outside border of the given number of pixels (default 20).

show scrollbars for center view

Choose whether to show scrollbars in the center view of the darkroom (default off).

show loading screen between images

Show gray loading screen when navigating between images in the darkroom. Switch this option off to just show a simple toast message and leave the previous image in place until the next image is loaded. Note that switching this option off can be very useful to quickly compare duplicate images, however, there might be issues with long loading times (leading you to think the next image has already loaded) and you may observe visual artifacts while the next image is loading (default on).

modules

display of individual color channels

Control how individual color channels are displayed when activated in the parametric masks feature. You can choose between “false color” and “gray scale” (default “false color”).

hide built-in presets for processing modules

If enabled, only user-defined presets will be shown in the presets menu for processing modules – built-in presets will be hidden (default off).

show the guides widget in modules UI

Enable this to show the local guides & overlays interface directly within the UI of the modules that support it (default on).

expand a single processing module at a time

Control how processing modules are expanded in the darkroom. If this option is enabled, expanding a module by clicking collapses any currently expanded module. If you want to expand a module without collapsing the others you can do so with Shift+click. Disabling this option inverts the meaning of click and Shift+click (default on).

only collapse modules in current group

When choosing to expand a single processing module at a time (using the logic defined in the previous setting), only collapse other modules that appear in the current visible group. Disable this option to ensure that modules in non-visible groups are also collapsed (default on).

expand the module when it is activated, and collapse it when disabled

Select this option for the darkroom to automatically expand or collapse processing modules when they are enabled or disabled. (default off)

scroll processing modules to the top when expanded

With this option enabled processing modules are scrolled to the top of the right-hand panel when expanded. (default on)

swap the utility and processing modules panels

If enabled, swap the left and right side panels in the darkroom view (default off).

show right-side buttons in processing module headers

Choose whether to show the four buttons (mask indicator, multi-instance menu, reset, presets menu) on the right-hand-side of the module header for processing modules. These buttons will always appear when the mouse is over the module. At other times they will be shown or hidden according to this preference selection:

- *always*: always show all buttons
- *active*: only show the buttons when the mouse is over the module
- *dim*: buttons are dimmed when mouse is not over the module
- *auto*: hide the buttons when the panel is narrow
- *fade*: fade out all buttons when the panel narrows
- *fit*: hide all the buttons if the module name doesn’t fit
- *smooth*: fade out all buttons in one header simultaneously
- *glide*: gradually hide individual buttons as needed

(default *always*)

show mask indicator in module headers

If enabled, an icon will be shown in the header of any processing modules that have a mask applied (default on).

prompt for name on addition of new instance

If enabled, when creating a new instance of a processing module, a prompt will be immediately displayed allowing you to set a name for the new instance (default off).

automatically update module name

Automatically update the name (label) on processing modules if the current parameters of the module match those of a saved preset . The module name will only be updated if it hasn't already been manually amended for the current image. It will be set either to the module name from which the preset was made (if the module name was manually adjusted before the preset was created) or to the name of the preset itself (default on).

9.7. processing

Control how images are processed.

image processing

always use LittleCMS 2 to apply output color profile

If this option is activated, darktable will use the LittleCMS 2 system library to apply the output color profile instead of its own internal routines. This is significantly slower than the default but might give more accurate results in some cases.

If the given ICC is LUT-based or contains both a LUT and a matrix, darktable will use LittleCMS 2 to render the colors regardless of this parameter's value (default off).

pixel interpolator (warp)

The pixel interpolator used for rotation, lens correction, liquify, crop and final scaling.

Whenever we scale or distort an image we have to choose a pixel interpolation algorithm (see wikipedia for details). For warping modules, darktable offers bilinear, bicubic or lanczos2. In general, bicubic is a safe option for most cases and is the default value.

pixel interpolator (scaling)

The pixel interpolator used for scaling. The same options are provided as for the warp modules, but with the addition of lanczos3.

lanczos3 can cause pixel overshoots leading to artifacts but sometimes gives a more crisp visual appearance. This option is therefore only provided for transforming (scaling) algorithms and is the default value.

LUT 3D root folder

Define the root folder (and sub-folders) containing LUT files used by the *LUT 3D* module.

Raster mask files root folder

Define the root folder (and sub-folders) containing raster mask files used by the external-raster-mask module.

auto-apply pixel workflow defaults

Choose which modules and module order are applied to new RAW image edits by default:

- *scene-referred (filmic)* assumes that most processing will be performed in a linear RGB color space. Selecting this option automatically enables the *filmic rgb* , *exposure* and *color calibration* modules for new edits and sets the module order to *v3.0 RAW*.

  The *exposure* module includes an automatic exposure adjustment of +0.7 EV (to provide a midtone brightening comparable to the +0.5 to +1.2 EV typically added by in-camera tone curves), and automatically enables the “compensate camera exposure” option for the filmic workflow. Both of these settings are intended to provide a reasonable starting-point for RAWs produced by a broad range of SLR and mirrorless cameras, but they can be overridden with an automatically-applied preset if the defaults produce consistently dark images for your camera.

  In the scene-referred workflows, the *color calibration* module acts in conjunction with the *white balance* module as the modern way to handle white balance and chromatic adaptation with improved color science. Note that when using the color calibration module, the white balance module needs to be active and set to “Camera Reference” mode (this will be done automatically and warnings will appear if the two modules have inconsistent settings). When using both modules as prescribed, it is still possible to auto-detect white-balance from a specific area of the image by selecting the CCT picker tool in the CAT tab of color calibration.

- *scene-referred (AgX)* follows the same assumptions and overall flow as *scene-referred (filmic)*, with the exception that it auto-enables the *AgX* module for tone mapping in place of *filmic rgb*.

- *scene-referred (sigmoid)* (default) also follows the same assumptions and overall flow as *scene-referred (filmic)*, with the exception that it auto-enables the *sigmoid* module for tone mapping in place of *filmic rgb*.

- *scene-referred (spektrafilm)* also follows the same assumptions and overall flow as *scene-referred (filmic)*, with the exception that it auto-enables the *spektrafilm* module for tone mapping in place of *filmic rgb*.

- *display-referred (legacy)* is the legacy mode (used by default in darktable 2.6 and earlier) and assumes that most processing will be performed in the Lab color space. Selecting this option automatically enables the *base curve* module for tone mapping and sets the module order to *legacy*. This workflow uses only the *white balance* module for chromatic adaptation.

- *none* sets the module order to *v3.0 RAW* and uses the *white balance* module for chromatic adaptation. No other exposure or tone mapping modules are enabled by default.

auto-apply per camera basecurve presets

Use a per-camera base curve by default (if available) instead of the generic manufacturer one. This should only be used in conjunction with the *display-referred* workflow defined above (default off).

detect monochrome previews

Enable this option to analyse images during import and tag them with the `darkroom|mode|monochrome` tag if they are found to be monochrome. The analysis is based on the preview image embedded within the imported file. This makes for a more convenient workflow when working with monochrome images, but it slows down the import, so this setting is disabled by default.

show warning messages

Enable this option to display warning messages in processing modules where non-standard and possibly harmful settings have been used in the pipeline. Such messages can sometimes be false-positives (because of intentional non-standard settings) and can be disregarded if you know what you are doing. Disable to hide these warnings. (default on).

HDR alignment

Note: These preferences are used for the *create hdr* option, and are only effective when darktable is built with OpenCV.

auto-align frames when merging HDR brackets

When enabled, darktable will attempt to align the images being used to create an HDR image (default on). To do this, it will register and warp exposure brackets onto a reference frame before merging, removing ghosting caused by a shaky tripod or handheld shots.

pick the HDR alignment reference automatically

Before merging, automatically examine every bracket and align to the one richest in detail instead of the first frame. Improves alignment when the first exposure is a poor template, at the cost of one extra decode pass over the frames (default off).

alignment detail (proxy scale)
Controls the fraction of the full sensor resolution at which feature detection and alignment are computed. Higher values find more distinctive features (better on repetitive or low-contrast scenes) at the expense of more time and memory (default 0.625). 0.5 restores the legacy speed.

shadow lift for feature detection (gamma)
The display-gamma applied to the linear raw before feature detection, to lift shadow detail into the detector's range (default 2.2). 1.0 disables it. Higher values help under-exposed frames but can amplify shadow noise; the percentile stretch already does part of this, so the effect is scene-dependent.

local contrast for feature detection (CLAHE clip)
Clip limit of the local contrast enhancement (CLAHE) applied before feature detection. 0 disables it (default 0). Enabling it (e.g. 2.0) helps detect features on extremely under-exposed frames, but on repetitive textures (e.g. façades, railings, foliage) it can change descriptor signatures between exposures and cause misalignment, so it is off by default.

feature budget per frame (keypoints)
Maximum number of Scale-Invariant Feature Transform (SIFT) keypoints kept per frame after spatial balancing, before matching (default 5000). Balancing both frames to a common budget keeps a feature-dense frame from flooding the matcher with ambiguous candidates.

save alignment debug images
Write per-frame diagnostic images while merging: the feature-detection input, the detected keypoints and all feature matches (green = inlier, red = outlier) (default off). Saved as Netpbm (PBM and PPM) files in a darktable_hdr_align_debug folder in the system temp directory (or the folder named by the DT_HDR_DEBUG_IMAGE_DIR environment variable). Only used for diagnosing alignment problems.

CPU / memory

darktable resources
Choose how much of your system and graphics card (GPU) memory will be used by darktable. Four options are provided by default:

- *small* takes roughly 20% of your system memory and 40% of your GPU memory. This might be acceptable on very large systems, especially if you're not exporting images. Mostly, though, this can only be recommended if you are using a lot of other demanding applications at the same time as darktable.
- *default* takes roughly 60% of your system memory and 70% of your GPU memory. This mode is recommended if you're not exporting a lot of images, have at least 16Gb of system memory and 4Gb of GPU memory, and also are running a lot of other application at the same time as darktable.
- *large* takes roughly 75% of your system memory and 90% of your GPU memory. This is the best option if you are only using darktable on your system and/or are exporting a lot of images.

See the memory & performance tuning section for more information.

OpenCL

This section configures GPU acceleration for darktable's regular image-processing pipeline. It is unrelated to the AI features' GPU acceleration, which is configured separately in AI preferences – see AI GPU acceleration for that setup.

activate OpenCL support
Your GPU can be used by darktable to significantly speed up processing. The OpenCL interface requires suitable hardware and matching OpenCL drivers on your system. If one of those is not found this option is grayed out. OpenCL support can be switched on and off at any time and takes immediate effect (default on).

OpenCL fast mode
If set, the OpenCL compiler uses aggressive optimizing for better performance with reduced precision while having some very subtle quality loss.

OpenCL scheduling profile
Defines how preview and full pixelpipe tasks are scheduled on OpenCL enabled systems:

- *default*: the GPU processes the center view pixelpipe; the CPU processes the preview pipe.
- *very fast GPU*: both pixelpipes are processed sequentially on the GPU.
- *multiple GPUs*: both pixelpipes are processed in parallel on different GPUs – see the multiple devices section for more information.

tuned GPU memory
If enabled on a system with multiple OpenCL devices you may specify a safety margin per device (headroom, default is 600MB).

OpenCL drivers
In most cases darktable is able to find the correct OpenCL driver but this depends on how your operating system handles installation. Generally speaking, darktable must:

- not use unreliable drivers
- only have one active driver per hardware device. Problems are typically found where the vendor-provided driver is installed *as well as* another driver (like `rusticl`) or (on Windows) where the `OpenCLon12` driver has been installed via the `OpenCL Compatibility Pack`.

Toggle-switches are offered for most available drivers, though on more exotic hardware like ARM boards you will have to enable the “other platforms” fallback option. Select the drivers you want to use from the list. If you suspect that a driver is malfunctioning you can explicitly disable it here.

See the memory & performance tuning section for more information.

9.8. security

Control whether warning messages are shown before undertaking certain activities.

general

ask before removing images from the library
Always ask before removing image information from darktable’s library database, where the xmp file is retained (default on).

ask before deleting images from disk
Always ask before deleting an image file (default on).

ask before discarding history stack
Always ask before discarding the history stack of an image (default on).

try to use trash when deleting images
Instead of physically deleting images from disk, attempt to put them into the system’s trash bin (default on).

ask before moving images from film roll folder
Always ask before moving an image file (default on).

ask before copying images to new film roll folder
Always ask before copying an image file to a new location (default on).

ask before removing empty folders
Always ask before removing any empty folder. This can happen after moving or deleting images (default off).

ask before deleting a tag
Always ask before deleting a tag from an image (default on).

ask before deleting a style
Always ask before deleting a style (default on).

ask before deleting a preset
Always ask before deleting a preset (default on).

ask before exporting in overwrite mode
Always ask before exporting images in overwrite mode.

other

password storage backend to use
The backend to use for password storage. Options: “auto” (default), “none”, “libsecret”, “kwallet”, “apple_keychain” (on macOS), “windows_credentials” (on Windows).

auto login to storage server
When exporting images to an external storage server (Piwigo) select this option to automatically log in to the server. You must set a password storage backend (above) to use this option.

executable for playing audio files

Define an external program which is used in the lighttable view to play audio files that some cameras record to keep notes for images (default “aplay”).

9.9. storage

The following options are related to darktable’s library database and XMP sidecar files .

database

allow for multiple workspaces

Allows for multiple workspaces that can be selected via a popup window at startup. In the popup, it is possible to create a new workspace, delete workspaces, set a workspace as the default (used on startup until changed again), or to work from memory (no database). If **remember selection and don’t ask again** is checked when confirming a workspace, that selection is kept for the next launch without showing the popup, and **allow for multiple workspaces** is turned off so the popup stays hidden until you enable **allow for multiple workspaces** again here. When this option is enabled, a workspace button also appears in the top panel so you can switch or create workspaces without restarting first (switching or creating still relaunches darktable into the chosen workspace). A workspace is a combination of a library database and a `darktablerc` configuration file using the same shared configuration folder settings (data database and shortcuts). New workspaces can optionally be created by copying an existing workspace as a template; workspace-specific configuration is duplicated while paths, labels, collection history and similar keys are cleared so the new workspace keeps its own library database. See darktable invocation for more details around `library.db`, `data.db` and `darktablerc`. (default off)

create database snapshot

Specifies how often darktable should create database snapshots. Options are “never”, “once a month”, “once a week”, “once a day” and “on close” (default “once a week”)

how many snapshots to keep

Number of snapshots to keep after creating a new snapshot, not counting database backups taken when moving between darktable versions. Enter “-1” to store an unlimited number of snapshots. (default 10)

XMP sidecar files

create XMP files

XMP sidecar files provide a redundant method of saving the changes that you have made to an image, in addition to the changes saved to darktable’s database. This option allows you to choose when files will *first be written*. Once written, they will subsequently be updated each time you edit or add a tag to the image. Choose from:

- *never*: Don’t write sidecar files. This can be useful if you are running multiple version of darktable for development/testing purposes but is not normally recommended,
- *on import*: A sidecar file will be written as soon as you add an image to darktable’s library,
- *after edit*: A sidecar will not be written until the first time you edit or add a tag to an image.

It’s strongly recommended that you choose either “on import” or “after edit”. Sidecar files provide a useful fail-safe to prevent data loss if your database becomes corrupted. Backing up your raw file plus the accompanying sidecar file will allow you to fully restore your work at a later date by re-importing your edit history back into darktable (default “on import”).

This option also affects the creation of DTDATA sidecar files - DTDATA files will be created when required unless this option is set to *never*.

store XMP tags in compressed format

Entries in XMP tags can get rather large and may exceed the available space to store the history stack in some output files on export. This option allows binary XMP tags to be compressed in order to save space. Available options are “never”, “always”, and “only large entries” (default).

auto-save interval

This preference sets the interval (in seconds) after which the processing history for an image will be automatically saved (while in the darkroom view). Set to zero to disable auto-saving. Note that this option might be ignored for slow drives (default 10s).

look for updated XMP files on startup

Scan all XMP files on startup and check if any have been updated in the meantime by some other software. If updated XMP files are found, a menu is opened for the user to choose which of the XMP files to reload (replacing darktable's database entries by the XMP file contents) and which of the XMP to overwrite from darktable's database. Activating this option also causes darktable to check for text sidecar files that have been added after import time (default off).

9.10. miscellaneous

interface

show splash screen at startup

When activated, darktable will display a splash screen indicating progress loading and initializing its code and data and scanning for updated XMP sidecars until it is ready to display its main window. When disabled, darktable displays nothing until loading is complete and it can display the lighttable – or darkrooom, if invoked with a single filename on the command line (default on).

show welcome screen on next run

When activated, the *Welcome* setup pages will be shown the next time darktable is started (default on, but is set to off after the welcome pages are shown).

load default shortcuts at startup

When launching the application, darktable loads default shortcuts first, and then loads user-defined shortcuts on top. This allows default shortcuts to be overridden with a new action but prevents them from being deleted (since the deleted shortcut will be automatically reloaded on the next restart). Deactivate this preference to stop loading default shortcuts on startup – only load the user-defined ones (including any defaults that you have not subsequently deleted or overridden). This makes deletion easier but also means that you will not benefit from new shortcuts added in future versions without first re-enabling this preference (default on).

scale slider step with min/max

When activated, the default step-size, when altering sliders, will depend on the current min/max values for that slider (default on).

marker shape

Select the shape of the slider marker from triangle, circle, diamond or bar (default triangle).

automatically update module name

If enabled, darktable will automatically update the module name to match a preset name or preset instance name, if present (default on).

sort built-in presets first

Choose how the presets menu is sorted. If this option is enabled, built-in presets are shown first. If the option is disabled, user presets are shown first (default on).

mouse wheel scrolls modules side panel by default

When enabled, the mouse wheel scrolls side panels by default and Ctrl+Alt+wheel scrolls data entry fields. When disabled, this behavior is reversed (default off).

enable touchpad gestures in darkroom

When enabled, use two-finger touchpad gestures in darkroom for panning (two-finger touchpad scrolling) and pinch-to-zoom. When disabled, touchpad gestures are ignored and darkroom uses only the mouse scrolling behavior for zooming in and out (default on).

constrain darkroom zoom between screen fit and 100%

When enabled, limits scroll-wheel and pinch-to-zoom in the darkroom to the range between screen fit (zoom out) and 100% (zoom in). Hold *Ctrl* while zooming to temporarily lift the limit and zoom beyond 100%. When disabled, zoom is never constrained and the additional *Ctrl* key press is not needed to zoom beyond 100% (default on).

always show panels' scrollbars

Defines whether the panel scrollbars should be always visible or activated only depending on the panel content (default on). (needs a restart)

duration of the ui transitions in ms

Defines how long modules and other ui elements will take to transition between states (expand/collapse). Set to 0 to disable animation (default 250ms).

auto-scroll filmstrip to center on selected image

When enabled, the filmstrip in culling mode and in the darkroom automatically scrolls to center on the selected image (default on).

position of the scopes module

Choose whether to show the scopes module in the left or right panel (default right). (needs a restart)

method to use for getting the display profile

This option allows the user to force darktable to use a specific method to obtain the current display profile for color management . In the default setting “all”, darktable will choose to query the X display server’s xatom or the colord system service. You can set this option to “xatom” or “colord” to enforce a specific method if the two methods give different results. You can run the darktable-cmstest binary to examine your color management subsystem.

order or exclude midi devices

Semicolon-separated list of device-name fragments. If matched, load the midi device at the id given by the location in the list. If preceded by “-”, prevent the matched device from loading. You can also add encoding and number of knobs. For example “BeatStep:63:16”. Please see midi device support for more information.

tags

omit hierarchy in simple tag lists

When exporting images any hierarchical tags are also added as a simple list of non-hierarchical tags to make them visible to some other programs. When this option is checked darktable will only include the last part of the hierarchy and ignore the rest. So `foo|bar|baz` will only add `baz`.

shortcuts with multiple instances

It is possible to create multiple instances of many processing modules. In this scenario it is not always obvious which instance should be controlled by keyboard shortcut operations. The following options control rules that are applied (in order) to decide which module instance keyboard shortcuts should be applied to.

These rules are also used when clicking and dragging on the scopes module to change exposure.

prefer focused instance

If an instance has focus, apply the shortcut to that instance and ignore any other rules. Note that this option does not impact blending shortcuts, which are always applied to the focused instance (default on).

prefer expanded instances

If instances of the module are expanded, ignore collapsed instances (default off).

prefer enabled instances

After applying the above rule, if remaining instances of the module are active, ignore inactive instances (default off).

prefer unmasked instances

After applying the above rules, if remaining instances of the module are unmasked, ignore masked instances (default off).

selection order

After applying the above rules, apply the shortcut to the first or last instance remaining (default “last instance”).

map / geolocalization view

pretty print the image location

Show a more readable representation of the geo-location in the image information module (default on).

slideshow view

waiting time before each image in slideshow

The number of seconds before displaying the next image (default 5) in the slideshow view .

9.11. shortcuts

Shortcuts are a way to control darktable using your keyboard, mouse, trackpad, or other *input devices*, such as MIDI and game controllers (as detailed in the midi device support section). They allow you to perform *actions* without interacting directly with a UI element.

An *action* is usually an operation that you might undertake using darktable's point-and-click user interface. See common actions for a list of the most common ones. For example:

- Increase, decrease or reset sliders
- Scroll through dropdown lists
- Enable, expand or focus modules
- Click buttons
- Switch between views

The recommended way to assign shortcuts to visual elements is the visual shortcut mapping mode.

in a nutshell

If you just want to assign a specific key to toggle a button, the process is the following:

1. Click on the  icon in the top panel
2. Hover the button that you want to control
3. Press the key that you want to use to control the button

Similarly, to control a slider with a shortcut you can:

1. Click on the  icon
2. Hover the slider that you want to control
3. Press the key that you want to use to control the slider, and scroll the mouse wheel or move the mouse horizontally, vertically or diagonally

The system is very flexible and powerful, and this is just scratching the surface. Keep on reading to understand it better and learn how to make the most of it.

types of shortcuts and actions

There are two types of actions:

Discrete actions
E.g., executing a command, focusing a UI element or resetting a slider

Directional actions
E.g., adjusting the value of a slider or scrolling through a list of presets

As the two action types are inherently different, so are the shortcuts you can use to trigger them:

- *Discrete actions* can be triggered by *discrete shortcuts* consisting only of key presses and optionally mouse clicks
- *Directional actions* need also a direction (and, in the case of sliders, a magnitude). Hence, the corresponding *directional shortcuts* will also incorporate a mouse movement

For example:

- The key e can be used to focus a module, or to toggle it on/off
- The key e, combined with up/down movements or the mouse, can be used to control the value of a slider

---

Note: A *directional shortcut* is sufficient to trigger a *discrete action*. The opposite is not true. However, it is possible to use a combination of two *discrete shortcuts* to control a *directional action*. For example, you can use two different shortcuts to trigger the up and down effects of a slider (see anatomy of an action).

If you are using keyboard and mouse, **all shortcuts must start with one or more key presses**, as mouse actions in isolation are used to navigate and interact with the UI.

If you have an input device, you can trigger discrete actions by pressing one or more buttons on the controller. Directional actions can use a combination of buttons and a knob/joystick movement, or just knob/joystick movement.

anatomy of a shortcut

The simplest shortcut consists of just one key press (e.g., e).

The same key can be repeated up to three times, so also `e double-press` (i.e., e pressed and released twice in quick succession) and `e triple-press` (i.e., e pressed and released three times in quick succession) are valid, and distinct, shortcuts.

You can further extend a shortcut with up to three clicks of different mouse buttons. So, `e double-press + left click` or `e + right click + left click` are also valid shortcuts.

Finally, if your fingers are very nimble and you really need to go there, you can repeat the last click up to three times. So, also `e + left click + middle double-click` and `e double press + middle click + left click + right triple-click` are valid shortcuts.

Hence, the most complex shortcut can consist of 8 key/button presses:

- Three key presses of the same key, (e.g., e), followed by
- Three presses of the three mouse buttons, followed by
- 2 more repetitions of the last click.

Your shortcut can include one or more modifiers (`shift`, `ctrl` and `alt`). In this case, for a directional shortcut, both the last key and the modifier must be pressed down when you execute the movement. For example, `ctrl + e double-press + pan` means that while moving the mouse left and/or right both e and `ctrl` must be pressed down.

Note: If you are a macOS user, your shortcuts will use `cmd` instead of `ctrl` and `option` instead of `alt`.

If you are defining a directional shortcut, then the movement part of the shortcut must be executed while the last key, mouse or controller button is held down.

For example, the shortcut `e double-press + pan`, can be activated by pressing e twice, holding down e on the second press and moving the mouse horizontally while the key is pressed.

short and long presses

By default, all key presses and button clicks in a shortcut are *short*, i.e., the key/button is pressed and immediately released. However, the last repetition of a key/button press in a shortcut can also be a *long* one. You can turn a normal key/button press into a long press by simply holding it a little longer before releasing it (longer than double-click speed but shorter than twice double-click speed). This applies to key presses and mouse clicks alike. Hence `e double-press` and `e long double-press` are two distinct shortcuts that can be assigned to different actions. The associated action triggers when the key is released, which entails that a shortcut ending with a long press cannot be used for a directional shortcut.

movements in directional shortcuts

The following movements are supported when defining directional shortcuts:

- Movement of the mouse scroll wheel
- Horizontal, vertical or diagonal movement of the mouse cursor
- Movement of a knob/joystick on an input device

When using an input device, you can directly assign a control knob or joystick to an action. However, this will significantly reduce the flexibility of such devices, as you will be able to use the knob/joystick only to control one action. By prefixing the movement with a key or button press you can use the same knob/joystick to control multiple actions.

triggering multiple shortcuts at once

If both `e + scroll` and `f + scroll` are mapped to a slider (or if you have fallbacks enabled), then scrolling the mouse wheel while holding down both `e` and `f` will move both sliders. Similarly, you could map `e double-press + scroll` and `f double-press + scroll` to two nodes in a graph (say, `yellow` and `green` in color equalizer). You can then move both sliders at once by:

1. pressing and releasing `e` once;
2. pressing and holding `e`;
3. pressing and holding `f`;
4. using the scroll wheel while both `e` and `f` are pressed.

This works because the repeat key counter is per-shortcut, not per key, and it is reset only after the current shortcut ends (i.e., when the repeated key is released). Thus it is possible to trigger as many actions at once as your fingers can handle, provided that all the shortcuts involved have the same number of key presses (i.e., single, double or triple presses).

shortcuts must be unique within a view

A single action may have multiple shortcuts but a single shortcut can only be linked to one action in a given darktable view – you can't chain actions together except by applying a preset or style. You can, however, set up a single shortcut that does one thing in the lighttable view and another in the darkroom view.

additional modifiers

The only valid modifiers are the `shift`, `ctrl` and `alt` keys on the keyboard. You can assign any key (or input device buttons) to function as one of the existing modifier types by assigning keys/buttons to the "global/modifier" action. However, you cannot define new modifier types.

anatomy of an action

Actions apply an *effect* to an *element* of a *widget*, where these terms are defined as follows:

widget
: Each visible part of the user interface is a *widget*. For example, the darktable application window is a widget, containing side panel widgets, each of which contains module widgets, each of which contains button, slider and dropdown list widgets etc. When assigning a shortcut to an action, you must first decide which widget it will apply to.

element
: An *element* is the part of a UI widget that is affected by your shortcut. For example, for a slider that has a picker , you can make a shortcut activate the picker *button* element or change the *value* element of the slider. For a row of tabs (the row is a single widget) you can select which tab element to activate or use your mouse scroll wheel to scroll through the tabs.

effect
: A shortcut can sometimes have multiple possible *effects* on a given *element*. For example, a button can be activated as if it was pressed with a plain mouse-click or as if it was pressed with Ctrl+click. A slider's value can be edited, increased/decreased or reset.

For example, you can assign the shortcut `e + scroll` to change the value (the *effect*) of the exposure slider (the *element*) of the exposure correction *widget*.

assigning shortcuts to actions

There are two primary methods of assigning shortcuts to actions: visual shortcut mapping (recommended) and the shortcut mapping screen . The former makes it easier to remap multiple shortcuts at once, while the latter allows for finer control and more flexibility.

visual shortcut mapping

Click on the  icon in the top panel of any darktable view to enter visual shortcut mapping mode. If you hold Ctrl while clicking the button, no confirmation will appear when overwriting an existing shortcut mapping.

The mouse cursor will change as you hover over UI widgets, to indicate whether or not a mapping can be created:

-  appears when you hover over a module header, to indicate that you can click to expand the module.
-  indicates that, in addition to assigning a shortcut, you can also add the widget to the quick access panel in the darkroom (by Ctrl+clicking on it).
-  indicates that the widget is already in the quick access panel (Ctrl+click to remove it).
-  indicates that a shortcut can be defined for the widget under the cursor but it cannot be added to or removed from the quick access panel.
-  indicates that there is no mappable widget under the cursor.

To define new shortcuts: Press a key combination while hovering over a mappable widget. A *default action* will be assigned to that shortcut based on the type of widget and whether it is directional. See the common actions section for examples of some of the defaults. You can assign as many shortcuts as you like in a single mapping session and then exit mapping mode when you are finished by clicking the  icon again or right-clicking anywhere on the screen.

To explore already defined shortcuts: Left-click on a mappable widget to open the shortcut mapping screen for that widget (see below). Left-click anywhere else on the screen to open the shortcut mapping screen, expanded (where possible) based on the part of the screen you have clicked on. This screen can be used to alter the action assigned to a shortcut and to configure shortcuts for non-visual actions. Entering the shortcut mapping screen exits visual shortcut mapping mode.

To update existing shortcuts: Define a second shortcut against the same widget. If you attempt to reallocate an existing shortcut to a new action, you will be notified of the conflict and prompted to replace the existing shortcut.

To change the default speed of a slider: Scroll with your mouse wheel while in visual mapping mode (without pressing any other buttons/keys) when hovering over the slider – scroll up to increase and down to decrease. When you leave mapping mode, normal mouse scrolls over that slider will change its value with the adjusted speed.

shortcut mapping screen

The shortcut mapping screen can be accessed from the global preferences dialog or by right-clicking on the visual mapping button. This screen allows access to all available actions, including some that are not directly linked to a UI widget.

The top panel of the shortcut mapping screen shows a list of available UI widgets/actions and the bottom panel shows the shortcuts currently assigned to them. Fields that can be changed by user action are shown in bold.

Note: You can search the top and bottom panels using the text entry boxes at the bottom of the screen, and use the up/down arrow keys to navigate between matches.

To create a new shortcut for an item: Double-click an item in the top panel, and then enter your desired shortcut (right-click to cancel). Once you have done this, a new entry will appear in the bottom panel showing the shortcut you have created. You can then manually alter the *element*, *effect*, *speed* or *instance* of the assigned action against that shortcut in the bottom panel. To delete a user-defined shortcut, select it in the bottom panel and press the Delete key.

Selecting an existing shortcut in the bottom panel will highlight (in bold) the matching action and its parents in the top panel. You can use this to navigate the top panel and find related actions.

The following additional options are provided in the shortcut mapping screen:

export...
Export the current shortcut mappings for one or all of your devices (keyboard/mouse, midi, game controller) to an external file. The dialog will show you how many shortcuts exist for each device.

import extras
Import extended shortcuts from a system file (`shortcutsrc`, with an optional language suffix, from your darktable installation – not the local config file of the same name). Existing shortcuts matching any in the system file will be replaced. These shortcuts will be localized to your system language if a localized version of the system file is available – currently, only German (`shortcutsrc.de`) in addition to the default English.

import...
Import shortcut mappings from an external file for one or all of your devices. When loading a device, you can chose to assign it a different number. This can for example be used to exchange midi layouts. Before loading, you can chose to wipe the specific device first. When loading all from an empty file, this will effectively delete all your shortcuts.

restore...
Restore your shortcut mappings to (a) The mappings shipped with darktable by default, (b) The start of your current session, or (c) The point at which the shortcut mapping screen was last opened. When restoring, you can choose to leave any additional shortcuts that were added after the relevant checkpoint as they are, so that only changed shortcuts are restored to their previous meaning. Or you can choose to first clear all shortcuts and just load the restore point.

deleting default shortcuts

When you delete a shortcut that has been created by darktable by default, that shortcut is moved to a separate “disabled defaults” category, in order to prevent it from being reloaded the next time darktable is launched. To reinstate a deleted shortcut, simply delete the shortcut from that category. You will be prompted if reinstating this shortcut overwrites another one that has been created in the meantime.

Alternatively, you may disable preferences > miscellaneous > interface > load default shortcuts at startup to prevent default shortcuts from being loaded on startup. While this option is disabled, darktable will only load user-defined shortcuts and any defaults that you have not subsequently deleted or overridden.

common actions

The following is a list of some of the actions to which you can assign shortcuts, organized by widget type. This is not an exhaustive list and you are encouraged to browse the shortcut mapping screen for a complete list of available actions.

Note: Some actions may have no effect. For example, all sliders include a *button* element, regardless of whether such a button is actually present alongside a given slider.

---

global actions

Actions in the “global” section of the shortcut mapping screen can be executed from any darktable view. Most of these actions do not have specific *elements* as they are used to perform one-off operations.

For example, the predefined shortcut `tab` triggers the action `globals/panels/all`, which toggles the visibility of all the side panels in the current view.

view-specific actions

Actions in the “views” section can only be executed from the specified darktable view. As with global actions, most do not have specific *elements* as they are used to perform one-off operations.

For example, the predefined shortcut `ctrl + b` triggers the action `views/darkroom/guide lines/toggle`, which toggles guide lines in the darktable view. It has an `effect` setting which allows you to directly simulate a right-click on the button, which in this case opens the guide lines configuration popup.

actions on modules

All *utility* and *processing* modules have the following elements:

show
Acts as a *toggle* that expands and collapses the module.

reset
Acts as a *button* that resets all module parameters when activated. The *ctrl-activate* action can be used to re-apply any automatic presets for that module.

presets
Allows you to select actions from the presets menu (e.g. edit, update, previous, next). The default action, when assigning a discrete shortcut to a *preset* element, is to display a list of the available presets for selection.

When a shortcut is assigned to a utility module, the default action is to *toggle* the *show* element (expand/collapse the module).

processing modules

Additionally, processing modules have the following elements:

enable
Acts as a *toggle* that switches the module on and off.

focus
Acts as a *toggle* that focuses or unfocuses the module. This is useful for modules such as *crop* or *tone equalizer* , whose on-screen controls are only activated when those modules have focus. For *crop*, changes are saved only when the module loses focus.

instance
Allows you to select actions from the multiple-instance menu (e.g. move up/down, create new instance). The discrete action associated to the *instance* element displays a list of the available options for selection; a directional action is also available and will move the *preferred module instance* (see below) up and down the pixelpipe.

If an action affects a processing module that can have multiple instances, you can choose which instance to adjust with a given shortcut. By default, all actions will affect the “preferred” instance, as defined using the settings in preferences > miscellaneous > shortcuts with multiple instances .

Additional options are available in the shortcuts mapping screen to adjust the blend parameters (the <blending> section) and module controls (the <focused> section) for the currently-focused module. The latter section allows you to assign shortcuts to the first, second, third (etc.) button, drop-down, slider and tab on the module. The shortcuts will affect different module controls depending on which module currently has focus (as the available list of controls changes).

For example, you if you assign `' + scroll` to `processing modules/\<focused\>/sliders` and set `element` to `1st`, you will be able to use the scroll wheel while holding down `'` to adjust the value of the first slider of the currently focused module. If `exposure` is focused you will affect the `exposure correction` slider, if `denoise (profiled)` is focused you will affect the `denoising strength` slider.

actions on specific widget types

buttons

A button is a clickable icon in the darktable interface. The default action is to activate that button as if clicked with the left mouse button. You can modify this action to activate the button as if clicked while holding a modifier key.

toggles

A toggle is a button that has a persistent on/off state. It therefore has additional *effects* to allow you to toggle it or explicitly set its state. As with a normal button, the default action is to activate the toggle as if clicked with the left mouse button (which toggles the button on/off).

For example, the predefined shortcut `shift + 0` is associated to `views/darkroom/raw overexposed/toggle`, which toggles on/off the raw overexposure indicator in the darkroom.

dropdowns

A dropdown is a multi-selection box and has the following elements available:

selection
: Allows values to be selected from the dropdown list in various ways. The default action, when assigning a discrete shortcut to a dropdown, is to display a popup *edit* box with a list of the available values for selection; A directional shortcut (i.e., including a mouse movement) will scroll through the available values.

button
: A standard *button* element that allows the button to the right of the dropdown (if present) to be activated. For example, the *aspect* dropdown in the *crop* module has a button that allows the crop controls to be changed from portrait to landscape and vice versa.

sliders

A slider allows you to continuously alter an integer or decimal value, and has the following elements available:

value
: Allows the current value of the slider to be altered. The default action, when assigning a discrete shortcut to a slider, is to display a popup *edit* box so you can enter a value; A directional shortcut will change the value up and down. Value elements are also used for modifying some on-screen graphs. When modifying the *value* element with a shortcut you may not exceed the bounds set in the visual slider.

force
: This is the same as the *value* element described above, but it allows you to exceed the bounds set in the visual slider.

zoom
: Allows you to change the upper and lower bounds of the visual slider without altering the current value.

button
: A standard *button* element that allows the button to the right of the slider (if present) to be activated. For example, a slider may include a picker to visually set its value based on selected elements of the image.

You can alter the value of a slider more quickly or slowly than normal by defining the *speed* of the action in the shortcut mapping screen. By default a *value* (or *force*) effect is given a speed of 1.0, which means that it is changed at the default rate defined by the given slider. You can alter the slider more quickly by increasing the speed (a speed of 10 makes the action 10x faster) or more slowly by decreasing it (a speed of 0.1 makes the action 10x slower).

fallbacks (a.k.a. generalized shortcut extensions)

As discussed above, a *widget* can have multiple different *actions* applied to it, and you may want to set shortcuts to control several of them. For example, to toggle a module on/off, expand/collapse it, and reset it to its default state. Normally, this would require setting 3 shortcuts for each module. If you want to control 5 modules in this fashion, you have to define 15 shortcuts. Setting up all these shortcuts one by one can be tedious and error prone.

To make this process simpler and to ensure consistency across all modules, you can create a single discrete shortcut for each module and then have a number of effects made available by default as *extensions* to that shortcut. These *generalized shortcut extensions* are called shortcut *fallbacks*.

Note: Fallbacks are a powerful way to quickly set up multiple actions using predefined and consistent shortcuts. They will assign a lot of actions automatically, which may be confusing if you are not aware of them. Fallbacks are therefore *disabled* by default and you will need to click on the `enable fallbacks` check box in the shortcuts setup window to enable them.

If they are enabled, when you create a discrete shortcut (e.g., `ctrl + r`) against a processing module the following shortcuts will be defined automatically:

- `ctrl + r` to show/hide the module (the default fallback)
- `ctrl + r + left click` to enable/disable the module
- `ctrl + r + left double-click` to reset the module
- `ctrl + r + right click` to show the module's preset menu
- `ctrl + r + right double-click` to show the module's multiple instance menu

In each case (except the first) you should hold the initial shortcut while clicking with your mouse. The final single/double mouse-click will apply the corresponding action.

Similar fallbacks are defined for many common UI elements. As with any other shortcut, fallback settings are fully customizable.

To see a list of *all* of the default fallbacks, click the "enable fallbacks" checkbox in the shortcut mapping screen and select the "fallbacks" category in the top panel. To see the fallbacks for a given widget (e.g. a slider) just select that widget in the top panel. In both cases an additional item (also named "fallbacks") will then appear in the bottom panel containing full details of the available fallbacks.

Fallbacks are only applied if no other shortcut using the resulting combination has been explicitly created. In the above example, if you were to explicitly assign `ctrl + r + left click` to another action, the "enable/disable module" fallback would be ignored.

Some fallback actions are defined using modifier keys (usually `ctrl +` and `shift +`). In this case you must define an initial shortcut without such a modifier in order to be able to use these fallbacks. For example, if you assign `ctrl + r` to an action, you cannot use a `ctrl +` fallback. Some default fallbacks of this type are provided for the *value* element and for horizontal/vertical movements in the (zoomed) central area – in this case, `shift +` increases the speed to 10.0 and `ctrl +` decreases the speed to 0.1.

9.12. presets

This menu provides an overview of the presets that are defined for darktable's modules, and allows you to modify some of their properties.

Pre-defined presets (those that are included by default within darktable) are shown with a lock symbol. Their properties cannot be changed.

User-defined presets can be imported from exported `.dtpreset` files using the "import" button at the bottom of the screen. You can export *all* user-defined presets to a single directory using the "export" button.

Delete a user-defined preset by selecting it and pressing the Delete key.

Edit a user-defined preset's properties by selecting it and pressing Enter or double-clicking. This will open a dialog that allows the preset to be edited, exported to an external `.dtpreset` file or deleted.

See the creating and editing presets section for more information about the other properties that can be edited in this dialog.

9.13. AI

Control darktable's AI features: enable or disable them, choose a hardware accelerator for model inference, point darktable at an ONNX Runtime library, and manage the AI models available on disk.

AI features are powered by ONNX Runtime . What runs out of the box and what GPU acceleration can be added on each platform:

OS	Bundled	Optional GPU acceleration
Linux	CPU only	NVIDIA CUDA, AMD ROCm/MIGraphX, Intel OpenVINO
Windows	DirectML (any DirectX 12 GPU)	NVIDIA CUDA, Intel OpenVINO
macOS	CoreML (Apple Silicon, integrated GPU)	–

Install scripts for the GPU runtimes are described below; see GPU acceleration for the full instructions.

For a deeper explanation of how AI features work – the inference runtime, execution providers, model layout, GPU acceleration, and troubleshooting – see the AI features special topic.

general

enable AI features

Master switch for all AI features. When disabled, no ONNX Runtime is loaded, no model directories are scanned, and AI-dependent modules hide themselves. Turn this off to eliminate AI-related memory and startup cost when you don't need it (default off).

AI acceleration

The hardware accelerator used for model inference. The list only shows the currently available providers as supported by the loaded ONNX Runtime. Installing a different ONNX Runtime updates the list.

- *auto* picks the best available provider for the loaded library, falling back to CPU if no accelerator initialises.
- *CPU* runs inference on the processor. Always available.
- *NVIDIA CUDA* uses NVIDIA GPUs via cuDNN (Linux, Windows; requires installing a CUDA-enabled ONNX Runtime).
- *AMD MIGraphX* uses AMD GPUs via ROCm (Linux only; requires installing a MIGraphX-enabled ONNX Runtime).
- *Intel OpenVINO* uses Intel GPUs and iGPUs via OpenVINO (Linux, Windows; requires installing an OpenVINO-enabled ONNX Runtime).
- *Windows DirectML* uses any DirectX 12 capable GPU (Windows; bundled by default).
- *Apple CoreML* uses the Apple Neural Engine and integrated GPU (macOS; bundled by default).

Double-click the label to reset to the default. If the selected provider cannot be initialised with the currently-loaded ONNX Runtime (for example, if the library was built without that provider) a *not available, will fall back to CPU* notice appears next to the combo box. After swapping the ONNX Runtime library, the notice becomes *restart to apply* until darktable is restarted.

ONNX Runtime library

Path to a shared library (`libonnxruntime.so*` on Linux, `onnxruntime.dll` on Windows) that darktable should load in place of the bundled one. Leave empty to use the bundled library (see the table above for what that is per OS). To use a GPU runtime, install one via the scripts described below and point this field at it. Double-click the label to reset to empty. Changes take effect after restarting darktable. *Hidden on macOS*, where ONNX Runtime is statically linked with CoreML support.

To the right of the path field are two buttons:

- *browse* (folder icon) – open a file chooser to pick a custom ONNX Runtime library. Selection is validated by loading the library and probing its version and available execution providers; if the file is not a valid ONNX Runtime build it is rejected and the previous path is restored.
- *detect* – scan standard system locations for an ONNX Runtime library (package-manager installs, `/usr/lib`, `/usr/local/lib`, `~/.local/lib`). If multiple candidates are found you will be asked to choose; if exactly one is found it is applied automatically. Useful when you have installed ONNX Runtime via your distribution's package manager or via the install script described below.

installing a GPU ONNX Runtime

GPU acceleration on Linux and Windows (NVIDIA CUDA, AMD MIGraphX, Intel OpenVINO) requires a GPU-enabled ONNX Runtime build, which darktable does not ship in-app. Install scripts handle this end-to-end – detecting your GPU, downloading the matching runtime, verifying it, and installing under your user account.

See GPU acceleration for the full instructions: one-line installers for Linux and Windows, prerequisites, GPU and driver requirements, alternative manual installation, building AMD ONNX Runtime from source, and troubleshooting.

After installation, restart darktable, then use the *detect* button above to point darktable at the newly-installed library (or pick the file manually with the browse button).

models

The lower half of the tab lists AI models known to darktable – both already-downloaded ones and those available from the darktable-ai model repository.

model list

Each row has the following columns:

- *select* (leftmost checkbox) – include the model in bulk operations (*download selected*, *delete selected*). Independent of the enabled state. The header checkbox toggles all rows at once.
- *name* – model name.
- info button (*i* icon) – click to open the model card with full metadata: scope, author, source, paper, license, training data, data license, notes.
- *version* – model version string from the repository manifest.
- *task* – which task this model serves: *mask*, *denoise*, *rawdenoise*, *upscale*.
- *enabled* – marks the model as the active model for its task. Only one model can be active per task at a time: ticking it on a second model of the same task automatically un-ticks the previously active one, like a radio button. Un-ticking the active model leaves the task with no active model until another is picked. AI-dependent modules use whichever model is currently active for their task.
- *status* – *downloaded*, *not downloaded*, *update available*, or *update required*.
- *default* – *yes* if the model is in the developer-recommended default set (selected by *download default*), *no* otherwise.

download / update default

Download or update the subset of models the darktable developers recommend as defaults for each task. Useful to get going quickly with a fresh install, and to pull in any newer versions of those models that the repository has published since.

download / update selected

Download or update the models whose *select* checkbox is ticked. Use this when you've ticked one or a few specific models in the list and only want to operate on those.

import from file…

Install a model from a local `.dtmodel` archive. The file chooser is filtered to `*.dtmodel`; once imported, the model immediately becomes available. Use this for models you've been given by hand, models you've built yourself, or extra per-task alternatives from the darktable-ai repository – the in-app list shows only one or two models per task, but the repository may publish additional alternatives for the same task (different denoise model trained on a different dataset, alternative segmentation backbone, etc.). Download the `.dtmodel` from the repository's GitHub releases and import it here.

delete selected

Remove the models whose *select* checkbox is ticked from disk. Models that were bundled with darktable cannot be deleted.

Downloads run in the background with a progress dialog and can be cancelled. Failed downloads are reported and the affected models remain in their previous state.

9.14. lua options

From 5.6 onward

These settings control how lua scripts are handled.

check for updated lua scripts on start up
automatically update (bundled) scripts to current version.

disable lua scripts
do not run lua scripts (bundled and user defined).

Note: If you have a previous lua scripts installation (e.g. Lua API 9.6.0 from darktable 5.4) in your configuration directory the option *disable scripts* will not be displayed. After (automatically) updating the scripts through the script manager the option will appear.

per-script configuration options (dropdown)
allows selecting lua scripts to adjust settings they might contain. The available settings will then appear below. The lua scripts have to be *activated* in the scripts-module to become available here.

Pre 5.6

These set up the behavior of the Lua scripts installer .

lua scripts installer log debug messages
Write debugging messages to the log if the -d lua flag is specified

lua scripts installer don't show again
Check this box to hide the Lua scripts installer in the lighttable if no lua scripts are installed.

9.15. configuration directory

darktable stores its settings, presets, styles, library database, etc. in a default directory. This directory's location depends on your operating system and installation method. It can be changed by using the `--configdir` option (see darktable invocation).

The location of the configuration directory (and of other darktable directories and files) can be found by running darktable (or darktable-cli) with the `--print-paths` (or `--print-paths-as-flags`) option.

Default configuration paths

- Linux
  - Installed via package manager or AppImage: `$HOME/.config/darktable`
  - Flatpak: `$HOME/.var/app/org.darktable.Darktable/config/darktable`
  - snap: `$HOME/snap/darktable/current/`
- Windows: `%LOCALAPPDATA%\darktable`
  - Typically `C:\Users\<username>\AppData\Local\darktable`
- macOS: `$HOME/.config/darktable`

Notable contents:

- `lua` (folder): Lua scripts. Note: from 5.6 onwards Lua scripts are also bundled and this folder is not required. But it will be respected if present. Custom scripts can be added into a subfolder. See Lua: basic principles .
- `styles` (folder): User created styles
- `darktablerc`: Main configuration file
- `data.db` and `library.db`: Central libraries: Image File Locations, metadata, edits etc.
- `luarc`: Defines Lua scripts to be loaded upon startup. Optional from 5.6 onwards, see Lua: basic principles .
- `shortcutsrc`: Keyboard shortcut configuration
- `user.css`: User-defined CSS-tweaks

Cache directory

- Linux
  - Installed via package manager or AppImage: `$HOME/.cache/darktable`
  - Flatpak: `$HOME/.var/app/org.darktable.Darktable/cache/darktable`
- Windows: `%LOCALAPPDATA%\Microsoft\Windows\INetCache\darktable`
  - Typically `C:\Users\<username>\AppData\Local\Microsoft\Windows\INetCache\darktable`
- macOS: `$HOME/.cache/darktable`

The cache directory path can be overridden using the `--cachedir` option (see darktable invocation).

10. Scripting with Lua

darktable comes with a versatile scripting interface language for functionality enhancement.

10.1. overview

Lua scripts can be used to define actions for darktable to perform when an event is triggered. One example might be calling an external application during file export in order to apply additional processing steps outside of darktable.

Lua is an independent project founded in 1993, providing a powerful, fast, lightweight, embeddable scripting language. Lua is widely used by many open source applications, in commercial programs, and for games programming. darktable uses Lua version 5.4. Describing the principles and syntax of Lua is beyond the scope of this manual. For a detailed introduction see the Lua reference manual .

darktable comes with a number of scripts included . The central orchestrator for scripts is the script manager (see basic principles).

The following sections will provide you with a brief introduction to how Lua scripts can be used within darktable. The comprehensive documentation on Lua Scripting in darktable can be found in the darktable lua documentation .

10.2. darktable's scripts

from 5.6 onward

A collection of scripts is included with darktable. They can be disabled in preferences > lua .

Starting with darktable 5.6, the lua-scripts are included with the release, so they no longer have to installed separately.

See basic principles for an explanation of how darktable loads Lua scripts.

Documentation for darktable’s scripts is available in the lua docs .

pre 5.6

Pre 5.6 the scripts were not bundled and had to be installed using the following instructions.

Download and Install

A collection of scripts is available in the lua-scripts repository . This also includes the script manager, which makes it easier to enable or disable individual scripts from the GUI. Documentation for these lua scripts is included in the lua docs .

To use these existing scripts in versions before 5.6, they must be installed in darktable. This can in most cases be done within darktable's GUI, but depends on your OS and chosen method of installation.

In any case, darktable should be started at least once to set up its directories. The config directories for most installation types are listed on the configuration directory page.

Documentation for darktable's scripts is available in the lua docs .

Linux Package, AppImage and Flatpak

The Lua scripts can be installed using darktable's GUI. Ensure git is installed on your system. If it isn't, use the package manager to install it. In the lighttable view, expand the lua scripts installer module in the lower-left corner. Ensure that select action: install scripts is selected, then click execute. Restart darktable once for the scripts module to appear in the lower-left corner.

Linux Snap Packages

Installation through darktable's GUI is currently broken , so you will have to take the following approach:

Ensure git is installed on your system. If it isn't, use the package manager to install it. You will have to download the Lua scripts using git, then create the `luarc` to let darktable load the script-manager. Close darktable, open a terminal and:

```
cd ~/snap/darktable/current
git clone https://github.com/darktable-org/lua-scripts.git lua
echo 'require "tools/script_manager"' >> luarc
```

Restart darktable twice, and the scripts module should appear in the lower-left corner.

macOS

The Lua scripts can be installed by using darktable's GUI. In lighttable view, expand the lua scripts installer module in the lower-left corner. Ensure that select action: install scripts is selected, then click execute. Restart darktable once for the scripts module to appear in the lower-left corner.

Alternatively you could use the console. Ensure git is installed on your system. If it isn't, use the package manager to install it. Then open a terminal and:

```
cd ~/.config/darktable/
git clone https://github.com/darktable-org/lua-scripts.git lua
```

Windows

Ensure git is installed on your system. Git can be obtained from git for windows , or using Windows package manager winget inside a command prompt with `winget install git.git`. If you use the gitforwindows.org distribution, install the Git Bash Shell also as it will aid in debugging the scripts if necessary.

The Lua scripts can now be installed by using darktable's GUI. In lighttable view, expand the lua scripts installer module in the lower-left corner. Ensure that select action: install scripts is selected, then click execute. Restart darktable once for the scripts module to appear in the lower-left corner.

Updating

To update the script repository, either use the scripts module GUI or open a terminal or command prompt and run the following commands:

Snap

```
cd ~/snap/darktable/current/lua
git pull
```

Flatpak

```
cd ~/.var/app/org.darktable.Darktable/config/darktable/lua
git pull
```

Linux and MacOS

```
cd ~/.config/darktable/lua/
git pull
```

Windows

```
cd %LOCALAPPDATA%\darktable\lua
git pull
```

10.3. basic principles

Scripts can register callbacks to perform actions on various darktable events. This callback mechanism is the primary method of triggering lua actions.

The central orchestrator for lua scripts is the *script manager*, which spawns the utility module **scripts** in the lighttable and loads darktable's bundled scripts.

Lua scripts can be disabled in preferences/lua options .

how darktable loads scripts (and the script manager)

darktable treats two kinds of locations differently: files that are *executed* as Lua code, and directories that are *scanned* for scripts:

- `luarc`-files executed at startup:
  - `$DARKTABLE/share/darktable/luarc`: run by darktable itself. This is the entry point that launches the (bundled) script manager. It serves internal purposes only and should not be edited. $DARKTABLE is the program's installation directory.
  - `[configuration-directory]/luarc`: If present – this is one way of calling your own scripts (see below).
- Scanned for `.lua` scripts (which then appear in the scripts-module):
  - `$DARKTABLE/share/darktable/lua-scripts/`: darktable's bundled scripts.
  - `[configuration-directory]/lua/[folder]`: your own scripts (see below).

The configuration directory is specific to your OS.

Thus you can **add your own scripts** as follows:

- Have the scripts-module pick them up (recommended)
  - Place your script e.g. in `[configuration-directory]/lua/user/myscript.lua`
  - Activate it in the scripts-module by selecting the appropriate folder (in this example "user") and clicking the on/off button next to the corresponding script.
  - Note: For the scripts-module to pick up scripts, they will need to be in a *subfolder* of `[configuration-directory]/lua`.
- Use `luarc` (runs at startup automatically)
  - Either write the code directly into `[configuration-directory]/luarc`. This works, but is not practical for anything beyond a few lines.
  - Or keep the code in its own file and `require` through `luarc`: E.g. a script present in `[configuration-directory]/lua/userscript.lua` could be loaded by adding `require "userscript"` (note that .lua is omitted here).

a note on lua-script installations from previous darktable versions

Before version 5.6, darktable's scripts had to be installed manually (see darktable's scripts).

To remain compatible with such an installation, `require` looks for a script manager in your configuration directory before loading the bundled one: if `[configuration-directory]/lua/tools/script_manager.lua` is present (for example from a manual pre-5.6 installation), *that* script manager is run instead of the bundled one.

10.4. a simple lua example

Let's start with a simple example that will print some code on the console. Create a file called `luarc` in darktable's configuration directory and add the following line to it:

```
print("Hello World !")
```

Start darktable from the console and you will see the sentence "`Hello World !`" printed on the console. Nothing fancy but it's a start.

At this point, there is nothing darktable-specific in the script. We simply use Lua's standard print function to print a string. That's nice and all, but we can do better than that. To access the darktable API you first need to `require` it and save the returned object in a variable. Once this is done you can access the darktable API as subfields of the returned object. All of this is documented in darktable's Lua API reference manual.

```
local darktable = require "darktable"
darktable.print_error("Hello World !")
```

Run the script and ... nothing happens. The function `darktable.print_error` is just like `print` but will only print the message if you have enabled lua traces by running darktable with "`darktable -d lua`" on the command line. This is the recommended way to do traces in a darktable lua script.

10.5. printing labeled images

The first example showed us the very basics of lua and allowed us to check that everything was working properly. Now let's do something a little bit more complex. Let's try to print a list of images that have a "red" label attached to them. But first of all, what is an image?

```
local darktable = require "darktable"
local debug = require "darktable.debug"
print(darktable.debug.dump(darktable.database[1]))
```

Running the code above will produce a lot of output. We will look at it in a moment, but first let's look at the code itself.

We know about `require darktable`. Here, we need to separately `require darktable.debug` which is an optional section of the API that provides helper functions to help debug lua scripts.

`darktable.database` is a table provided by the API that contains all images in the library database. Each entry in the database is an image object. Image objects are complex objects that allow you to manipulate your image in various ways (all documented in the `types_dt_lua_image_t` section of the API manual). To display our images, we use `darktable.debug.dump` which is a function that will take anything as its parameter and recursively dump its content. Since images are complex objects that indirectly reference other complex objects, the resulting output is huge. Below is a cut down example of the output.

```
toplevel (userdata,dt_lua_image_t) : /images/100.JPG
   publisher (string) : ""
   path (string) : "/images"
   move (function)
   exif_aperture (number) : 2.7999999523163
   rights (string) : ""
   make_group_leader (function)
   exif_crop (number) : 0
   duplicate_index (number) : 0
```

```
    is_raw (boolean) : false
    exif_iso (number) : 200
    is_ldr (boolean) : true
    rating (number) : 1
    description (string) : ""
    red (boolean) : false
    get_tags (function)
    duplicate (function)
    creator (string) : ""
    latitude (nil)
    blue (boolean) : false
    exif_datetime_taken (string) : "2014:04:27 14:10:27"
    exif_maker (string) : "Panasonic"
    drop_cache (function)
    title (string) : ""
    reset (function)
    create_style (function)
    apply_style (function)
    film (userdata,dt_lua_film_t) : /images
       1 (userdata,dt_lua_image_t): .toplevel
       [......]
    exif_exposure (number) : 0.0062500000931323
    exif_lens (string) : ""
    detach_tag (function): toplevel.film.2.detach_tag
    exif_focal_length (number) : 4.5
    get_group_members (function): toplevel.film.2.get_group_members
    id (number) : 1
    group_with (function): toplevel.film.2.group_with
    delete (function): toplevel.film.2.delete
    purple (boolean) : false
    is_hdr (boolean) : false
    exif_model (string) : "DMC-FZ200"
    green (boolean) : false
    yellow (boolean) : false
    longitude (nil)
    filename (string) : "100.JPG"
    width (number) : 945
    attach_tag (function): toplevel.film.2.attach_tag
    exif_focus_distance (number) : 0
    height (number) : 648
    local_copy (boolean) : false
    copy (function): toplevel.film.2.copy
    group_leader (userdata,dt_lua_image_t): .toplevel
```

As we can see, an image has a large number of fields that provide all sort of information about it. Here, we are interested in the “red” label. This field is a boolean, and the documentation tells us that it can be written. We now just need to find all images with that field and print them out:

```
darktable = require "darktable"
for _,v in ipairs(darktable.database) do
  if v.red then
    print(tostring(v))
  end
end
```

This code should be quite simple to understand at this point, but it contains a few interesting aspects of lua that are worth highlighting:

- `ipairs` is a standard lua function that will iterate through all numeric indices of a table. We use it here because darktable's database has non-numeric indices which are functions to manipulate the database itself (adding or deleting images, for example).
- Iterating through a table will return both the key and the value used. It is conventional in lua to use a variable named "_" to store values that we don't care about.
- Note that we use the standard lua function `tostring` here and not the darktable-specific `darktable.debug.dump`. The standard function will return a name for the object whereas the debug function will print the content. The debug function would be too verbose here. Once again, it is a great debug tool but it should not be used for anything else.

10.6. adding a simple shortcut

So far, all our scripts have done things during startup. This is of limited use and doesn't allow us to react to real user actions. To do more advanced things we need to register a function that will be called on a given event. The most common event to react to is a keyboard shortcut.

```
darktable = require "darktable"

local function hello_shortcut(event, shortcut)
darktable.print("Hello, I just received '"..event..
       "' with parameter '"..shortcut.."'")
end

darktable.register_event("hello shortcut",
       "shortcut",hello_shortcut,
       "A shortcut that prints its parameters")
```

Now start darktable, go to "preferences > shortcuts > lua > A shortcut that prints its parameters", assign a shortcut and try it. You should see a nice message printed on the screen.

Let's look at the code in detail. We first define a function that takes two strings as input parameters. The first one is the type of event triggered ("shortcut") and the second is the name of the shortcut ("A shortcut that prints its parameters"). The function itself calls `darktable.print`, which will print the message as an overlay in the main window.

Once that function is defined, we register it as a shortcut callback. To do that we call `darktable.register_event` which is a generic function for all types of events. We tell it that we are registering an event of type "shortcut" with the name "hello shortcut", then we give the callback to call and finally, we give the string to use to describe the shortcut in the preferences window.

Let's try a shortcut that is a little more interactive. This one will look at the images the user is currently interested in (selected or moused-over) and increase their rating:

```
darktable = require "darktable"

darktable.register_event("increase rating","shortcut",
  function(event,shortcut)
    local images = darktable.gui.action_images
    for _,v in pairs(images) do
      v.rating = v.rating + 1
    end
  end,"Increase the rating of an image")
```

At this point, most of this code should be self explanatory. Just a couple of notes:

- Instead of declaring a function and referencing it, we declare it directly in the call to `darktable.register_event` this is strictly equivalent but slightly more compact.
- `image.rating` is a field that gives the star rating of any image (between 0 and 5 stars, -1 means rejected).
- `darktable.gui.action_images` is a table containing all the images of interest. darktable will act on selected images if any image is selected, and on the image under the mouse if no image is selected. This function makes it easy to follow darktable's UI logic in lua.

If you select an image and press your shortcut a couple of times, it will work correctly at first but when you have reached five stars, darktable will start showing the following error on the console:

```
<![CDATA[
LUA ERROR : rating too high : 6
stack traceback:
   [C]: in ?
   [C]: in ?
   [C]: in metamethod 'newindex'
   ./configdir/luarc:10: in function <./configdir/luarc:7>
   [C]: in ?
   [C]: in ?
  ]]>
```

This is lua's way of reporting errors. We have attempted to set a rating of 6 to an image, but a rating can only go as high as 5. It would be trivial to add a check, but let's go the complicated way and catch the error instead:

```
darktable.register_event("increase rating","shortcut",
  function(event,shortcut)
    local images = darktable.gui.action_images
    for _,v in pairs(images) do
      result,message = pcall(function()
        v.rating = v.rating + 1
        end)
      if not result then
        darktable.print_error("could not increase rating of image "..
          tostring(v).." : "..message)
      end
    end
  end,"Increase the rating of an image")
```

`pcall` will run its first argument and catch any exception thrown by it. If there is no exception it will return `true` plus any result returned by the function. If there is an exception it will return `false` and the error message of the exception. We simply test these results and print them to the console.

10.7. exporting images with lua

So far we have learned to use lua to adapt darktable to our particular workflow. Let's look now at how to use lua to easily export images to an online service. If you are able to upload an image to a service via the command line then you can use lua to integrate this into darktable's user interface.

In this next example we will use lua to export via `scp`. A new storage type will appear in darktable's UI that will export images to a remote target via the copy mechanism in `ssh`.

```
darktable = require "darktable"

darktable.preferences.register("scp_export","export_path",
  "string","target SCP path",
  "Complete path to copy to. Can include user and hostname","")

darktable.register_storage("scp_export","Export via scp",
  function( storage, image, format, filename,
     number, total, high_quality, extra_data)
    if not darktable.control.execute("scp "..filename.." "..
      darktable.preferences.read("scp_export",
         "export_path","string")) then
      darktable.print_error("scp failed for "..tostring(image))
    end
end)
```

`darktable.preferences.register` will add a new preference to darktable's preferences menu, `scp_export` and `export_path` allow us to uniquely identify our preference. These fields are reused when we read the value of the preference. The `string` field tells the lua engine that the preference is a string. It could also be an integer, a filename or any of the types detailed in the API manual relating to `types_lua_pref_type`. We then have the label for the preference in the preference menu, the tooltip when hovering over the value and a default value.

`darktable.register_storage` is the call that actually registers a new storage type. The first argument is a name for the storage type, the second is the label that will be displayed in the UI and last is a function to call on each image. This function has a lot of parameters, but `filename` is the only one we use in this example. It contains the name of a temporary file where the image was exported by darktable's engine.

This code will work but it has a couple of limitations. This is just a simple example after all:

- We use preferences to configure the target path. It would be nicer to add an element to the export UI in darktable. We will detail how to do that in the next section.
- We do not check the returned value of `scp`. That command might fail, in particular if the user has not correctly set the preference.
- This script can't read input from the user. The remote `scp` must use password-less copy – `scp` can't be provided with a password easily, so we will leave it that way.
- There is no message displayed once the example is done, only the progress bar on the lower left side tells the user that the job is complete.
- We use `darktable.control.execute` to call an external program. The normal `os.execute` would block other lua codes from happening.

10.8. building user interface elements

Our previous example was a bit limited. In particular the use of a preference for the export path wasn't very nice. We can do better than that by adding elements to the user interface in the export dialog.

UI elements are created via the `darktable.new_widget` function. This function takes a type of widget as a parameter and returns a new object corresponding to that widget. You can then set various fields in that widget to set its parameters. You will then use that object as a parameter to various functions that will add it to the darktable UI. The following simple example adds a lib in the lighttable view with a simple label:

```
darktable = require "darktable"

local my_label = darktable.new_widget("label")
my_label.label = "Hello, world !"

darktable.register_lib("test","test",false,false,{
    [darktable.gui.views.lighttable] = {"DT_UI_CONTAINER_PANEL_LEFT_CENTER",20},
  },my_label)
```

There is a nice syntactic trick to make UI elements code easier to read and write. You can call these objects as functions with a table of key values as an argument. This allows the following example to work. It creates a container widget with two sub-widgets: a label and a text entry field.

```
local my_widget = darktable.new_widget("box"){
  orientation = "horizontal",
  darktable.new_widget("label"){ label = "here => " },
  darktable.new_widget("entry"){ tooltip = "please enter text here" }
}
```

Now that we know that, let's improve our script a bit.

```
darktable = require "darktable"

local scp_path = darktable.new_widget("entry"){
  tooltip = "Complete path to copy to. Can include user and hostname",
  text = "",
  reset_callback = function(self) self.text = "" end
}

darktable.register_storage("scp_export","Export via scp",
  function( storage, image, format, filename,
     number, total, high_quality, extra_data)
    if not darktable.control.execute("scp "..filename.." "..
      scp_path.text
    ) then
      darktable.print_error("scp failed for "..tostring(image))
    end
  end,
  nil, --finalize
  nil, --supported
  nil, --initialize
  darktable.new_widget("box") {
    orientation = "horizontal",
    darktable.new_widget("label"){ label = "target SCP PATH " },
    scp_path,
})
```

10.9. sharing scripts

So far, all of our lua code has been in *luarc*. That's a good way to develop your script but not very practical for distribution. We need to make this into a proper lua module. To do that, we save the code in a separate file (`scp-storage.lua` in this case):

```
--[[
SCP STORAGE
a simple storage to export images via scp

AUTHOR
Jérémy Rosen (jeremy.rosen@enst-bretagne.fr)

INSTALLATION
* copy this file in $CONFIGDIR/lua/
* add the following line in the file $CONFIGDIR/luarc
  require "scp-storage"
```

```
USAGE
* select "Export via SCP" in the storage selection menu
* set the target directory
* export your images

LICENSE
GPLv2

]]

darktable = require "darktable"
dtutils = require "lib/dtutils"

dtutils.check_min_api_version("7.0.0", ...)

local scp_path = darktable.new_widget("entry"){
  tooltip = "Complete path to copy to. Can include user and hostname",
  text = "",
  reset_callback = function(self) self.text = "" end
}

darktable.register_storage("scp_export","Export via scp",
  function( storage, image, format, filename,
     number, total, high_quality, extra_data)
    if not darktable.control.execute("scp "..filename.." "..
      scp_path.text
    ) then
      darktable.print_error("scp failed for "..tostring(image))
    end
  end,
  nil, --finalize
  nil, --supported
  nil, --initialize
  darktable.new_widget("box") {
    orientation = "horizontal",
    darktable.new_widget("label"){ label = "target SCP PATH " },
    scp_path,
})
```

darktable will look for scripts (following the normal lua rules) in the standard directories plus `$CONFIGDIR/lua/*.lua`. So our script can be called by simply adding `require "scp-storage"` in the *luarc* file. A couple of extra notes...

- `$CONFIGDIR` represents the darktable configuration directory .
- The function `dtutils.check_min_api_version` will check compatibility for you. `"7.0.0"` is the API version you have tested your script with and the "`...`" will turn into your script's name.
- Make sure to declare all your functions as `local` so as not to pollute the general namespace.
- Make sure you do not leave debug prints in your code – `darktable.print_error` in particular allows you to leave debug prints in your final code without disturbing the console.
- You are free to choose any license for your script but scripts that are uploaded on darktable's website need to be GPLv2.

Once you have filled all the fields, checked your code, you can upload it to our script page here .

10.10. calling lua from dbus

It is possible to send a lua command to darktable via its DBus interface. The method `org.darktable.service.Remote.Lua` takes a single string parameter which is interpreted as a lua command. The command will be executed in the current lua context and should return either nil or a string. The result will be passed back as the result of the DBus method.

If the Lua call results in an error, the DBus method call will return an error `org.darktable.Error.LuaError` with the lua error message as the message attached to the DBus error.

10.11. using darktable from a lua script

Warning: This feature is very experimental. It is known that several elements don't yet work in library mode. Careful testing is highly recommended.

The lua interface allows you to use darktable from any lua script. This will load darktable as a library and provide you with most of the lua API (darktable is configured headless, so the functions relating to the user interface are not available).

As an example, the following program will print the list of all images in your library:

```
#!/usr/bin/env lua
package = require "package"
package.cpath=package.cpath..";./lib/darktable/lib?.so"

dt = require("darktable")(
"--library", "./library.db",
"--datadir", "./share/darktable",
"--moduledir", "./lib/darktable",
"--configdir", "./configdir",
"--cachedir","cachedir",
"--g-fatal-warnings")

require("darktable.debug")

for k,v in ipairs(dt.database) do
   print(tostring(v))
end
```

Note the third line that points to the location of the `libdarktable.so` file.

Also note that the call to require returns a function that can be called only once and allows you to set darktable's command line parameter. The `:memory:` parameter to `--library` is useful here if you don't want to work on your personal library.

10.12. lua API

darktable's Lua API is documented in its own manual with a detailed description of all data structures and functions. You can download the API manual from here .

11. Guides & Tutorials

11.1. developing monochrome images

Photography has a long history of producing monochrome images, and many still enjoy this aspect of photography. While there are some specialized/modified cameras with a truly monochrome sensor, most still use a regular camera to capture a color image and transform it into a monochrome image during post-processing.

There are two main approaches to this conversion:

- A *physical approach*, where we attempt to simulate how a silver-based photographic film emulsion would react to the light captured at the scene.
- A *perceptual approach*, where we develop a color image and reduce the color saturation in a perceptual color space such as *CIE Lab*.

These approaches, and other monochrome-related features in darktable, are discussed in the following sections.

importing and flagging images as monochrome

When importing an image, there are a number of properties that can be used to indicate that the image requires a monochrome treatment:

- If the image was taken using an achromatic camera, the image will be automatically flagged as monochrome.
- When you capture a scene from which you would like to produce a monochrome image, it can be helpful to put your camera into a “black & white” creative mode. This allows you to visualise what the scene would look like in monochrome through your camera’s liveview screen or electronic viewfinder. The camera will still capture the full color data in the raw file, but the embedded JPEG preview image will be monochrome. When you import such an image, darktable can automatically flag the image as monochrome based on the preview image.

  Checking whether the preview is monochrome slows down the import process, so this is disabled by default. You can enable this check in preferences > processing > detect monochrome previews
- When processing a raw file, one of the first steps is to *demosaic* the image. If you set the *demosaicing method* to “passthrough (monochrome)”, this discards color information during the demosaicing process, and darktable will flag the image as monochrome.

  Note: You should only use this for images taken on a camera where the color filter array has been removed.
- After you have imported the image, you can manually flag an image as monochrome (or not) using the *metadata* tab on the lighttable’s actions on selection module,

If any of the above methods result in an image being flagged as monochrome, darktable modules can use this information to present the user with some monochrome-specific module controls and/or apply special processing to the image.

The `darktable|mode|monochrome` tag will be automatically applied to any images flagged as monochrome and, if you have enabled any permanent overlay information on your lighttable thumbnails, such images will be marked with a visual indicator `B&W` next to the file type information. By automatically applying this tag and visual indicator, darktable makes it easy to set up filters to single out images for monochrome treatment, and to see at a glance which images in the current collection bear the *monochrome* tag.

If darktable detects a true monochrome image or one from a monochrome-converted camera (using the “passthrough monochrome” demosaicer) some modules (e.g. demosaic, white balance) are automatically disabled.

monochrome conversion

physical approach

This approach tends to work with linear scene-referred data from the sensor, and attempts to mimic the response of a photographic film with a silver emulsion. It consists of three steps:

1. Map the color channels from the sensor into a single monochrome channel. Different types of monochrome photographic film have different levels of sensitivity to various wavelengths of light, and this can be simulated by giving the three color channels different weightings when combining them together into a single monochrome channel. The *color calibration* module allows the three channels to be mixed into a gray channel by varying amounts, and it includes a number of presets that are designed to emulate the characteristics of some well-known types of photographic film.

2. Apply a luminosity saturation curve. As a piece of photographic film is exposed to more intense light, its response will fall off as the silver emulsion becomes saturated. This saturation curve can be simulated within the *filmic rgb* module.

3. Developing a monochrome film in the darkroom traditionally involves “dodging and burning” to control the level of exposure across different parts of the image. This can be emulated in darktable by using either the *exposure* module with manually created masks , or by using the *tone equalizer* module, which generates a mask using a guided filter.

perceptual approach

The other option for producing a monochrome image is to reduce the color saturation in the image, which can be done in a linear colorspace, or in a color space oriented towards modelling human perception.

- The *color balance* module operates in linear RGB, and allows you to reduce the color saturation in the image using either the input or output color saturation slider – which you choose depends on whether you want to make any other adjustments to either the color or monochrome image in the color balance module. The color balance module tends to give a predictable and perceptually-uniform result.

- The *monochrome* module works in Lab color space, and it allows the user to graphically define a weighted combination of colors to determine the density of the blacks in the monochrome image. The interface can be somewhat sensitive to the settings, with small changes producing large effects, and you may experience problems with the global contrast and/or black pixel artifacts.

- Other modules such as *color zones* can also be used to remove color saturation from the image, but these don’t offer any real advantage over the simplicity of the *color balance* module’s saturation sliders.

11.2. batch-editing images

Batch-editing is the process of developing images in a semantically-related series that are expected to have a consistent final look, often with the intent of publishing the images in catalogues, magazines or books. It can be a tedious, frustrating and unexciting task, so darktable includes functionality to help make it faster and more reliable.

preparation

shooting color checkers

Shooting a color checker on-location can save a tremendous amount of time during batch post-processing of a series of images. The image of the color checker can be quickly used as a color reference in the post-processing workflow to neutralize any color cast – Datacolor and X-Rite color checkers (24 and 48) are natively supported.

prefer consistent lighting conditions

If possible, use controlled artificial lighting to maintain consistent lighting over the series of images. This means that you won't have to worry about light color temperature and intensity changes between images. Shoot a new image of the same color checker every time your lighting conditions change.

prefer manual mode

If possible, shooting in manual mode with constant exposure settings will help to remove some variability in the series. In terms of post-processing, any variation means that individual adjustments will be needed on each image, which will hinder your productivity.

post-processing

concepts

The post-processing needs to be split in 2 fully separated parts:

1. primary color-grading,
2. secondary color-grading.

Primary color-grading is performed first in the pixel pipeline **and** in the editing workflow, with modules such as *exposure* , *input color profile* and *color calibration* . Its purpose is to normalize each image to the same neutral ground-truth, in terms of overall brightness, color accuracy and white balance. This stage aims at making all images look similarly boring by removing any color cast and ensuring a perfectly neutral white, and is particularly important if your series uses different cameras. Primary color-grading is not about artistic intent or expression, but simply about preparing a sane and consistent basis for the next stage.

Secondary color-grading happens next in the pixel pipeline **and** in the editing workflow. This is where all the artistic expression happens, with modules such as *color balance rgb* . **If and only if** the primary color-grading was successful at perfectly neutralizing all images to the same ground-truth, copying and pasting the secondary color-grading steps between images should have exactly the same visual effect on each image, no matter if the images were shot with a different camera or under *slightly* different lighting conditions.

In a nutshell, the whole purpose of the primary color-grading is to ensure the repeatability of the secondary color-grading between images. For example, if a non-neutral white balance is expected in the series, it is much easier to re-introduce non-neutrality in *color balance rgb* (using the same color shift) on fully-neutralized images, than to fine-tune the primary color-grading individually on each image, especially if several different cameras were used.

method

profiling the series

You will first need to extract a color calibration profile from the image(s) of your color checker. This profile can then be applied to all images shot under the same lighting conditions with the same camera by copying and pasting the development history stack in the lighttable view. This step needs to be repeated for each camera and lighting setup.

Note: this process only works with the *modern chromatic adaptation* workflow, which assumes hard-setting the *white balance* to a D65 (*camera reference*) illuminant. See the documentation in the *color calibration* module for more information.

---

editing the reference image

Choose a reference image that was taken in the lighting conditions closest to those of the color checker image that served as your profiling reference. Your primary color-grading should already be handled by the profile used in the *color calibration* module (in conjunction with the *input color profile* module). What remains to complete this stage is to adjust the *exposure* setting to match the overall brightness that you expect.

Next, proceed with the *filmic rgb* white and black relative exposures, as well as the contrast setting. Finish with the secondary color-grading.

When this is done, you can measure the brightness and chromaticity of a control sample, preferably located on a non-moving surface that is consistently lit across your series (and appearing in all the frames). These measurements are taken using the *area exposure mapping* and *color calibration area mapping* tools. They will be memorized and will serve as a target for individual images as needed.

propagating the look

You can then propagate your secondary color-grading (including *filmic rgb*) to all the other images of the series, no matter whether or not they were shot in the same lighting conditions, since you should already have propagated your primary color-grading (calibration profile) to the relevant images. Don’t forget to paste the history using *append* mode or you will overwrite your primary color-grading as well.

However, this alone will not ensure a consistent look for the whole series.

fine-tune individual settings

If there was some variability in your lighting conditions, each image will need some further fine-tuning adjustments. Fortunately, if you have applied the proposed method so far, this should be relatively straightforward.

First, homogenize the exposure using your control sample and the *area exposure mapping* tool.

Then, adjust the *filmic rgb* white relative exposure if needed, preferably using the picker . The contrast should not require any adjustment since it does not depend on the dynamic-range of the image.

Finally, homogenize the chromatic adaptation, using your control sample and the *color calibration area mapping* tool.

This should get you close enough in most cases. However, if the background has changed, it is possible that these fine adjustments (aimed towards technically normalizing the primary color-grading on an individual basis) are not sufficient to give a perceptually-even look. In this case, you will need an extra step of secondary color-grading, which you are advised to perform on top of the previous one (shared with the other images of the series), in new instances of the relevant modules located later in the pipe. This ensures that the base secondary color-grading stays constant for all images and makes for a better workflow. You are not advised to make large changes in the primary color-grading to manage perceptual discrepancies involving contrast with the background.

controlling your series

The culling mode in the *lighttable* view will help you to compare images side-by-side when the work is done. To view your reference edit in the darkroom view, you may display the filmstrip and increase its height, or use a snapshot of the reference overlaid on the current image (which will not necessarily have the same size).

11.3. other resources

For help and support with using darktable you are encouraged to ask questions on the main discussion forums at discuss.pixls.us .

The official places to obtain information concerning darktable are the following:

- darktable.org
- GitHub wiki

The following articles provide some useful background information about darktable's scene-referred workflow:

- darktable 5.4 - A Simple Beginner Workflow and Interactive Walkthrough
- How do I ... in darktable 5.x?
- darktable 4.0: 3763 Days Later

Legacy articles:

- darktable 3.0 for dummies in 3 modules
- darktable 3.0 for dummies – hardcore edition
- darktable 3.0: RGB or Lab? Which modules? Help!
- darktable's filmic FAQ
- Introducing color calibration module

The following resources can provide a deeper understanding of the functionality of some specific processing modules:

- Noise reduction (profiled) with non-local means in darktable 3.0
- Noise reduction (profiled) with wavelets in darktable 3.0
- Contrast equalizer modules in darktable

Some other YouTube channels with recent content on using darktable are:

- Bruce Williams Photography
- Boris Hajdukovic

A continuously updated list of recent darktable resources:

- planet darktable

12. Special Topics

12.1. color management

12.1.1. overview

darktable employs a fully color managed workflow:

- Input color specifications are taken from embedded or user-supplied ICC profiles or (in the case of raw files) from a library of camera-specific color matrices.
- darktable automatically reads the display profile of your monitor (if properly configured) for accurate on-screen color rendition. Multiple-screen setups are fully supported as long as a system service like colord is in place and properly configured to inform darktable of the correct monitor profile.
- Output files can be encoded in one of darktable's built-in profiles, including sRGB and Adobe RGB, or into a color space supplied by the user as an ICC profile.

12.1.2. display profile

For darktable to faithfully render colors on screen it needs to find the correct display profile for your monitor. In general this requires your monitor to be properly calibrated and profiled, and it needs the profile to be correctly installed on your system. darktable queries your X display server's xatom as well as the system service colord (if available) for the right profile. If required you can enforce a specific method in preferences > miscellaneous .

To investigate your display profile configuration you can run the `darktable-cmstest` binary (Linux only) which prints out useful information (e.g. profile name per-monitor) and tells you if the system is correctly configured.

In rare cases you may need to manually select the display profile. This is possible from within the soft proof and gamut check option dialogs in the darkroom view and the display profile dialog in the lighttable view.

Bear in mind that high-tier consumer-grade screens usually don't need a user-made display profile unless you need to perform soft-proofing with professional expectations, since they are properly calibrated to sRGB in the factory.

A poorly-made display profile will be more harmful than sticking to the default sRGB profile, since the default might be slightly inaccurate but will at least be reliable. Advanced and professional users are advised to proceed with the production of custom display profiles only if they have the training to assess the quality of the resulting profile and understanding of the profiling options.

12.1.3. rendering method

darktable can render colors either with its own internal algorithms or by using the external library LittleCMS2. darktable's internal method is, by an order of magnitude, faster than the external one. The external option gives you a choice of the rendering intent and might offer a slightly higher accuracy in some cases.

You can change the default method in preferences > processing > always use LittleCMS 2 to apply output color profile

---

Note: If the given ICC is LUT-based or contains both, a LUT and a matrix, darktable will use LittleCMS2 to render the colors regardless of the configuration parameter's value.

12.1.4. rendering intent

If rendering with LittleCMS2 is activated (see rendering method) you can define how to handle out-of-gamut colors when converting between color spaces. A selection box in the export , *output color profile* , and soft proof modules gives you a choice of the following rendering intents:

perceptual
Best suited to photographs as it maintains the relative position of colors. This is usually the best choice.

relative colorimetric
Out-of-gamut colors are converted to colors having the same lightness, but different saturation. Other colors remain unmodified.

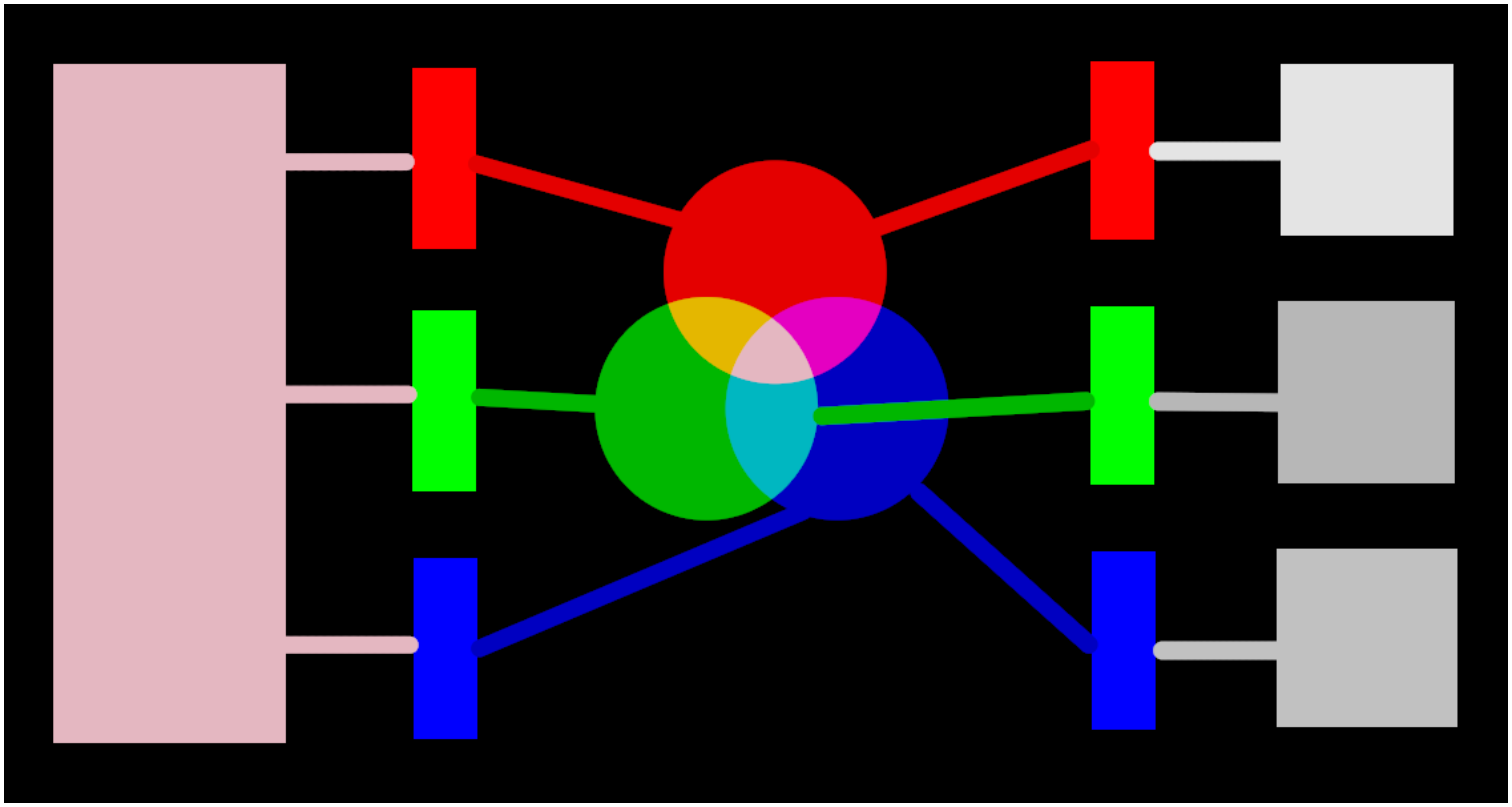
saturation
Saturation is retained but lightness is slightly changed.

absolute colorimetric
Keep the white point.

12.1.5. darktable's color spaces

Input images are either RGB files (like JPEGs or TIFFs) or camera RAWs. Both store visual information as a combination of primary colors (e.g. red, green and blue) which together describe a light emission to be recreated by a display.

The following image illustrates this concept.



The left-hand-side of the image depicts a colored light that we need to represent digitally. We can use three ideal color filters to decompose this light into three colored primary lights at different intensities. In order to recreate the original colored light from our ideal decomposition (as illustrated in the center of the image) we simply need to recombine those three primary lights by addition.

It should be possible to reproduce the original colored light by taking a set of white lights at the correct intensities and projecting those lights through appropriately colored filters. This experiment can be performed at home using gels and dimmable white bulbs. This is roughly what old color CRT displays did and is how video projectors still work.

In photography, the initial decomposition step is performed by the color filter array that sits on top of your camera's sensor. This decomposition is not ideal, so it isn't possible to precisely recreate the original emission with simple addition – some intermediate scaling is required to adjust the three intensities.

On screens, the LED bulbs are dimmed proportionally to each intensity, and the emissions of the three lights are physically added to reconstruct the original emission. Digital images store the intensities of these primary lights as a set of three numbers for each pixel, depicted on the right-hand side of the above image as shades of gray.

While a set of display intensities can be easily combined to recreate an original light on a screen (for example, if we created a synthetic image in-computer) the set of captured intensities from a sensor needs some scaling in order for the on-screen light addition to reasonably reproduce the original light emission. This means that every set of intensities, expressed as an RGB set, must be linked to a set of filters (or primary LED colors) that define a *color space* – any RGB set only makes sense with reference to a color space.

Not only do we need to temper the captured intensities to make them summable again, but if we are to recompose the original light on a display that does not have the same colored filters or primaries as the space in which our RGB set belongs, these intensities need to be rescaled to take into account the differing filters on the display. The mechanism for this scaling is described in *color profiles*, usually stored within `.icc` files.

Note: Color is not a physical property of light – it exists only in the human brain, as a product of the decomposition of a light emission by the cone cells in the retina, again very similar in principle to the above filtering example. An "RGB" value should be understood as "light emissions encoded on 3 channels connected to 3 primaries", but the primaries themselves may look different from what humans would call "red", "green" or "blue".

The filters described here are overlapping band-pass filters. Since they overlap, summing them back together would not preserve the energy of the original spectrum so (long story short) we need to dial them down with regard to the retina cone response

Most of darktable's actual image processing takes place in a large RGB "working profile" space, with some (mostly older) modules internally working in the CIELab 1976 color space (often just called "Lab"). The final output of the image processing pipeline is once again in an RGB space shaped for either the monitor display or the output file.

This process implies that the pixelpipe has two fixed color conversion steps: *input color profile* and *output color profile* . In addition there is the *demosaic* step for raw images, where the colors of each pixel are reconstructed by interpolation.

Each module has a position in the pixelpipe that tells you which color space the module lives in:

- up to *demosaic* : The raw image information does not yet constitute an "image" but merely "data" about the light captured by the camera. Each pixel carries a single intensity for one primary color, and camera primaries are very different from primaries used in models of human vision. Bear in mind that some of the modules in this part of the pipe can also act on non-raw input images in RGB format (with full information on all three color channels).
- between *demosaic* and *input color profile* : Image is in RGB format within the color space of the specific camera or input file.
- between *input color profile* and *output color profile* : Image is in the color space defined by the selected working profile (linear Rec. 2020 RGB by default). As darktable processes images in 4x32-bit floating point buffers, we can handle large working color spaces without risking banding or tonal breaks.
- after *output color profile* : Image is in RGB format as defined by the selected display or output ICC profile.

12.1.6. unbounded colors

Screens and most image file formats can only encode RGB intensities confined within a certain range. For example, images encoded on 8 bits can only contain values from 0 to 255, images on 10 bits from 0 to 1023, and so on… Graphic standards postulate that the maximum of that range, no matter its actual value, will always represent the maximum brightness that the display medium is able to render, usually between 100 and 160 Cd/m² (or nits) depending on the actual standard. We generally call this maximum “100 % display-referred”. The minimum of the range, encoded 0 no matter the bit-depth used, becomes then “0 % display-referred”. 100 % encodes pure white, 0 % encodes pure black.

This is a limitation for image processing applications, because it means that any pixel lying outside of this range will be clipped to the nearest bound, resulting in non-recoverable loss of data (colors and/or textures).

For the longest time, image processing software too was bounded to this limitation for technical reasons, and some still is, but now by design choice. As a result, they would clip RGB intensities at 100 % display-referred between image operations.

darktable uses floating-point arithmetic inside its color pipeline, which means it can handle any RGB value internally, even those outside the display-referred range, as long as it is positive. Only at the very end of the pipeline, before the image is saved to a file or sent to display, are the RGB values clipped if needed.

Pixels that can take values outside of the display range are said to have “unbounded colors”. One could choose to clamp (i.e. confine) those values to the allowed range at every processing step or choose to carry on with them, and clamp them only at the last step in the pipeline. However, it has been found that processing is less prone to artifacts if the unbounded colors are not clamped but treated just like any other color data.

At the end of the pipeline, modules like *filmic rgb* can help you to remap RGB values to the display-referred range while maximizing the data preservation and avoiding hard clipping, which is usually not visually pleasing.

However, at all times in the pipeline, you must ensure that you do not create negative RGB values. RGB intensities encode light emissions and negative light does not exist. Those modules that rely on a physical understanding of light to process pixels will fail if they encounter a non-physical light emission. For safety, negative RGB values are still clipped whenever they might make the algorithms fail, but the visual result might look degraded. Negative values can be produced when abusing the *black level* in *exposure* or the *offset* in *color balance* and care should be taken when using these modules.

12.1.7. possible color artifacts

There are some infrequent situations that still can lead to problematic results unless the user takes some action. Some modules in Lab color space, like *levels* and *monochrome* , rely on the fact that the L channels carries all lightness information, with the a and b channels purely representing chroma and hue. Unbounded colors with negative L values are especially problematic to these modules and can lead to black pixel artifacts.

It has been found that highly saturated blue light sources in the image frame are likely candidates for pixels with negative L values. If you are engaged in stage photography you should pay close attention to such lights appearing in images.

In order to mitigate this issue the *input color profile* module has a gamut clipping option. This option is switched off by default but can be activated if artifacts are observed. Depending on the settings, colors will be confined to one of the available RGB gamuts. In effect black pixel artifacts are prevented at the costs of losing some color dynamics.

12.1.8. darktable's color dimensions

This section defines the perceptual properties of color, both conceptually and quantitatively, in order to characterize and quantify the creative and corrective adjustments made to color in darktable.

definitions

Color properties like “saturation”, “brightness” or “lightness” have passed into common usage but are largely misused and often used to mean different things. In color science, each of these terms has a precise meaning.

There are two frameworks within which color properties may be analyzed and described:

- A scene-linear, physiological, framework, that mostly focuses on the response of the retina cone cells, using color spaces such as CIE XYZ 1931 or CIE LMS 2006,
- A perceptual, psychological, framework that stacks the brain’s corrections on top of the retina signal, using color spaces such as CIE Lab 1976, CIE Luv 1976, CIE CAM 2016 and JzAzBz (2017).

These two frameworks provide us with metrics and dimensions to analyze color and allow us to change some of its properties while preserving others.

The following dimensions of color are used by darktable:

hue
: An attribute of visual perception in which an area appears to be similar to one of the colors red, yellow, green, or blue, or to a combination of adjacent pairs of these colors considered in a closed ring. [1] Hue is a shared property between the perceptual and scene-linear frameworks.

luminance
: The density of luminous intensity with respect to a projected area in a specified direction at a specified point on a real or imaginary surface. [2] Luminance is a property of scene-referred frameworks, and is expressed by the Y channel of the CIE XYZ 1931 space.

brightness
: An attribute of visual perception according to which an area appears to emit, transmit or reflect more or less light. [3]

lightness
: The brightness of an area judged relative to the brightness of a similarly illuminated area that appears to be white or highly transmitting. [4] Lightness is the perceptual, non-linear homologue of luminance (roughly equal to the cubic root of luminance Y). Lightness is expressed by the L channel in CIE Lab and Luv 1976 and the J channel in JzAzBz.

chroma
: The colorfulness of an area judged as a proportion of the brightness of a similarly illuminated area that appears gray, white or highly transmitting. [5] *Warning: chroma is not short for chrominance, which is the color part of a video signal (the Cb and Cr channels in YCbCr, for example).*

brilliance
: The brightness of an area judged relative to the brightness of its surroundings. [6]

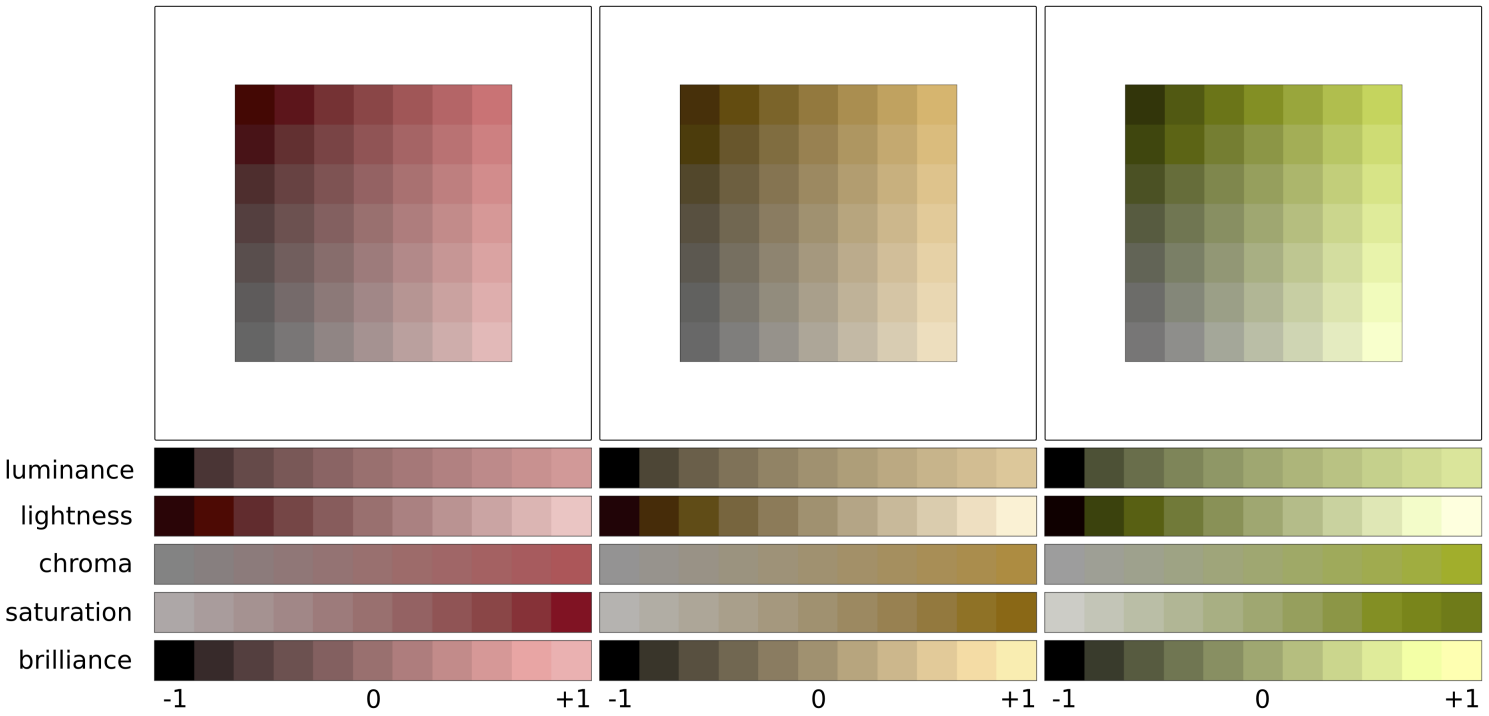
saturation
: The colorfulness of an area judged in proportion to its brightness. [7]

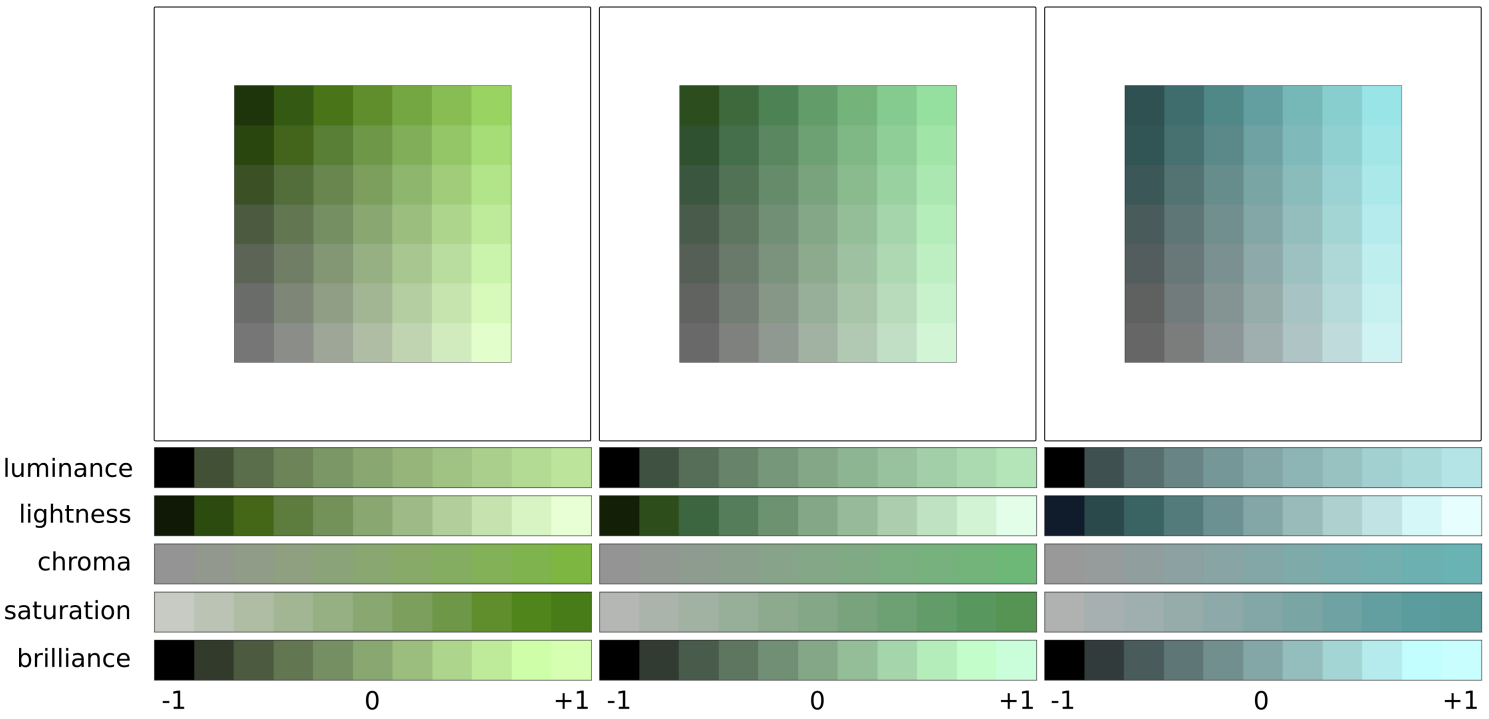
purity
: The distance of the pixel chromaticity from the white point in the *xy* chromaticity plane. Zero purity means the light is achromatic. High purity, at the other extreme, means the light is laser-like, composed of a single wavelength.

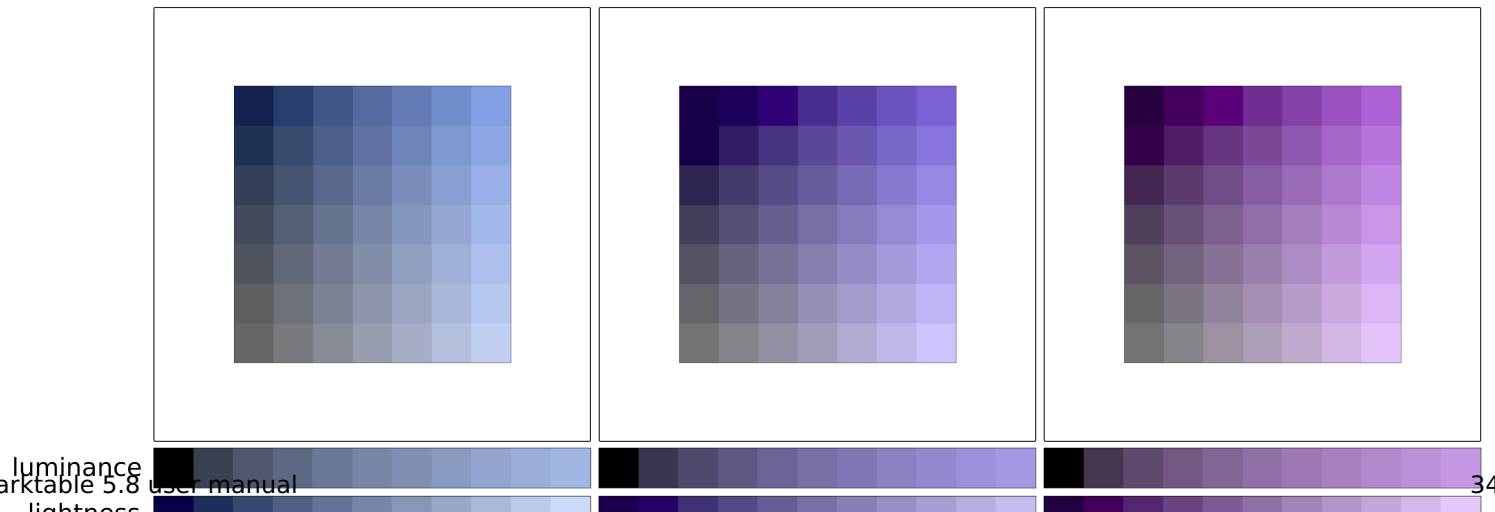
Colors can be described in many different color spaces, but no matter the color space, each color needs at least 3 components: some metric of luminance or brightness, and 2 metrics of chromaticity (hue and chroma, or opponent color coordinates).

illustrations

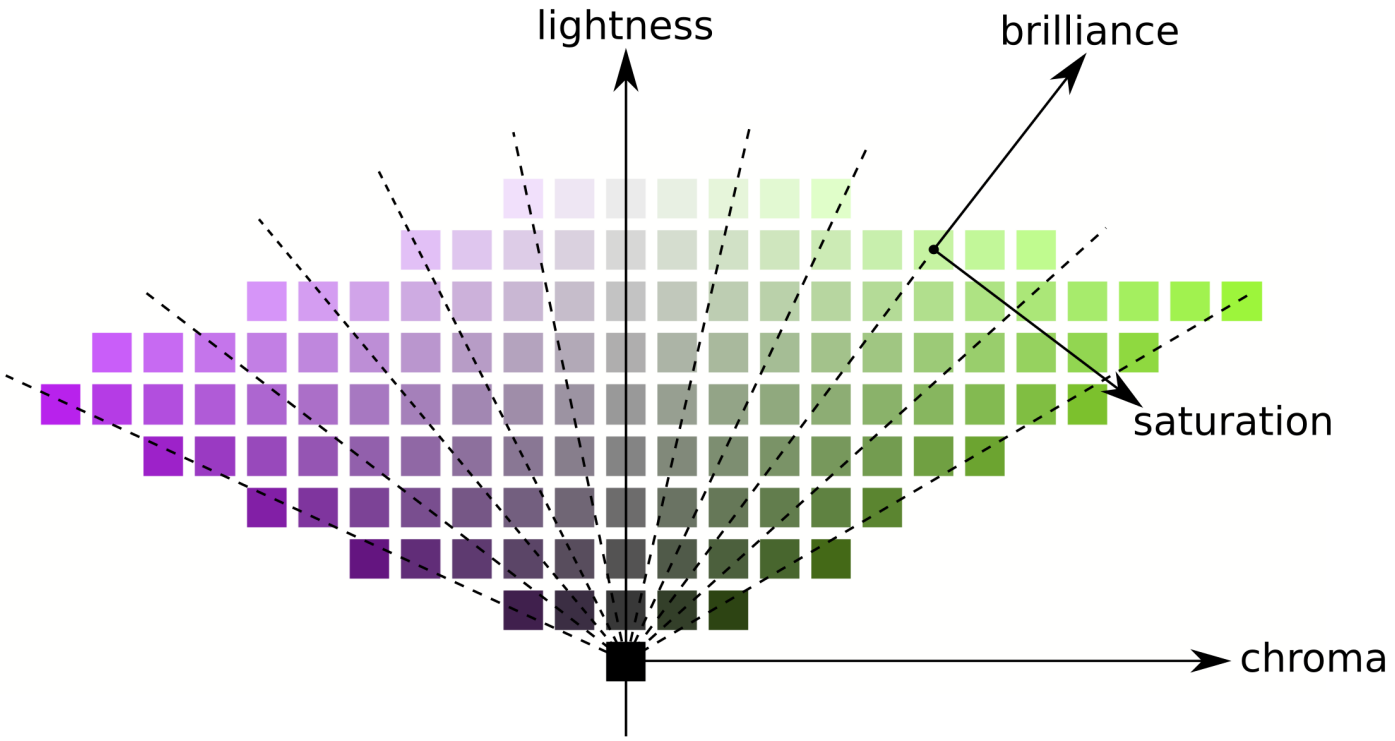
While the previous definitions are useful to give a meaning to the words, they don’t tell us what we should be looking at. The following charts show luminance, lightness, chroma, brilliance/brightness and saturation varying from a “0” base color and how the resulting colors degrade:


luminance
lightness
chroma
saturation
brilliance
-1
0
+1
-1
0
+1
-1
0
+1


luminance
lightness
chroma
saturation
brilliance
-1
0
+1
-1
0
+1
-1
0
+1


luminance
lightness

(Lightness + Chroma) or (Brilliance/Brightness + Saturation) are two different ways to encode the same reality. They are orthogonal spaces that can be converted from one to another by a simple rotation of the base. This means that chroma evolves at constant lightness, saturation evolves at constant brilliance/brightness, and vice versa:



Lines of equal chroma are vertical (following the patches grid), meaning that chroma has the same direction for all colors in the gamut (see below). However, lines of equal saturation are oblique (drawn dashed on the graph) and all go from black through each color patch, meaning that their directions are particular to each color.

Increasing the chroma will therefore move all colors uniformly away from the central gray axis horizontally, while increasing the saturation will close or open the angle of the oblique dashed lines like a flower.

Similarly, increasing the lightness will move all colors uniformly up from the horizontal axis, while increasing brilliance/brightness will move them along the lines of equal saturation.

On both of the above charts, lightness, chroma, saturation and brilliance are drawn in the JzAzBz color space, which is a perceptual color space suited for an HDR signal, and is used in parametric masks and the color balance RGB module. Luminance is drawn in the CIE XYZ 1931 color space, and represents the effect of an exposure compensation. It shows the same behavior as brilliance, except that the step size is not perceptually scaled.

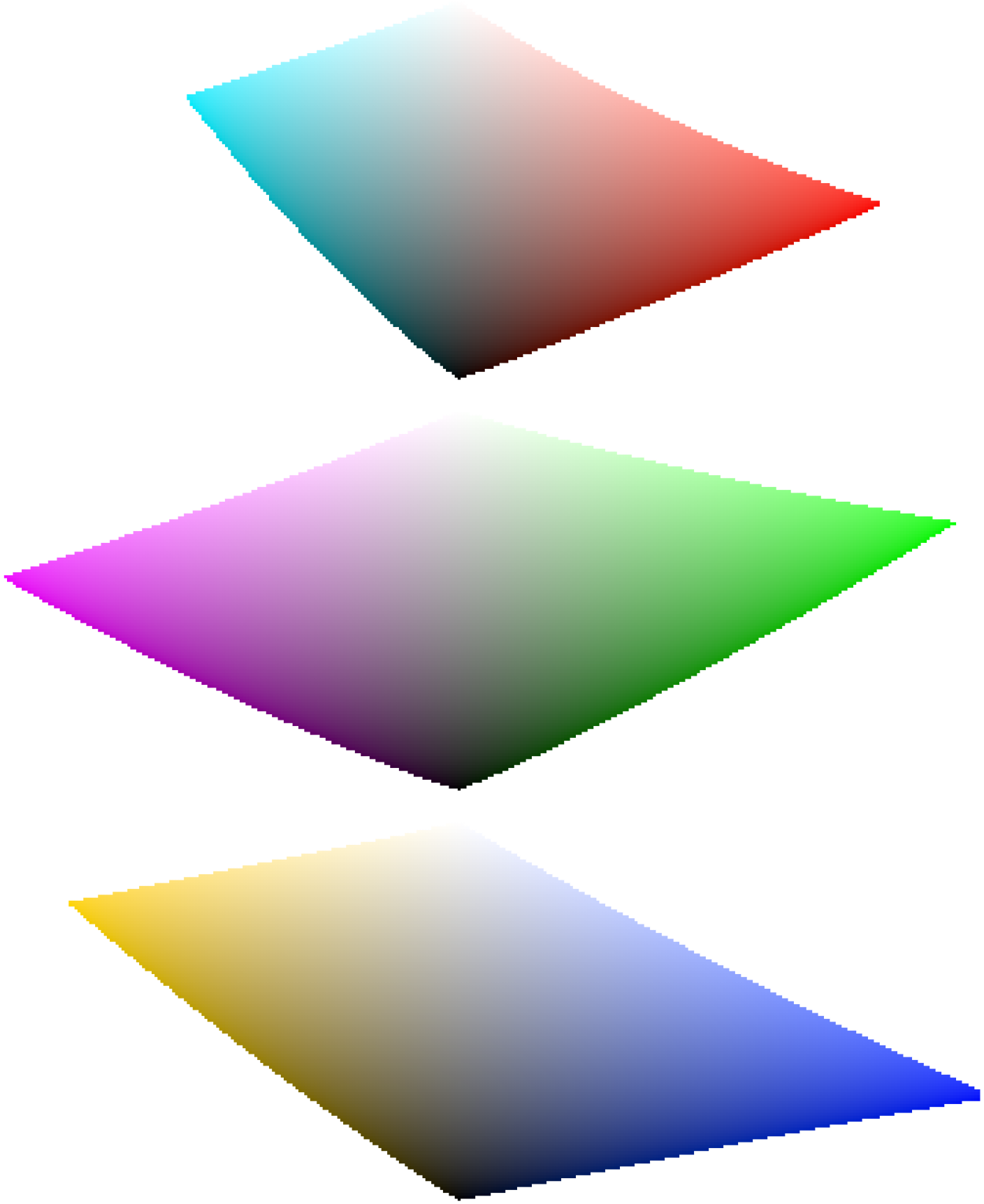
Note: In this section, *brilliance* and *brightness* are both used to describe the same dimension. In all rigor, brightness is an absolute metric, whereas brilliance is the brightness of some surface relative to the brightness of its surroundings (that is, how much a surface “pops” out of its surroundings and looks fluorescent). But in image editing, increasing the brightness of some surface will indeed increase its brilliance too, so the term *brilliance* is preferred in darktable’s user interface for clarity and in reference to its visual effect.

---

color dimensions and gamut

The gamut is the volume of colors that a certain color space can encompass and encode. It is important to note that, once converted to perceptual spaces, the gamut of any RGB space is not uniform along hues.

The following examples show the gamut volume of the sRGB space on hue slices containing the primary red, green and blue lights of the sRGB space, over a lightness-chroma plane with a uniform scale:

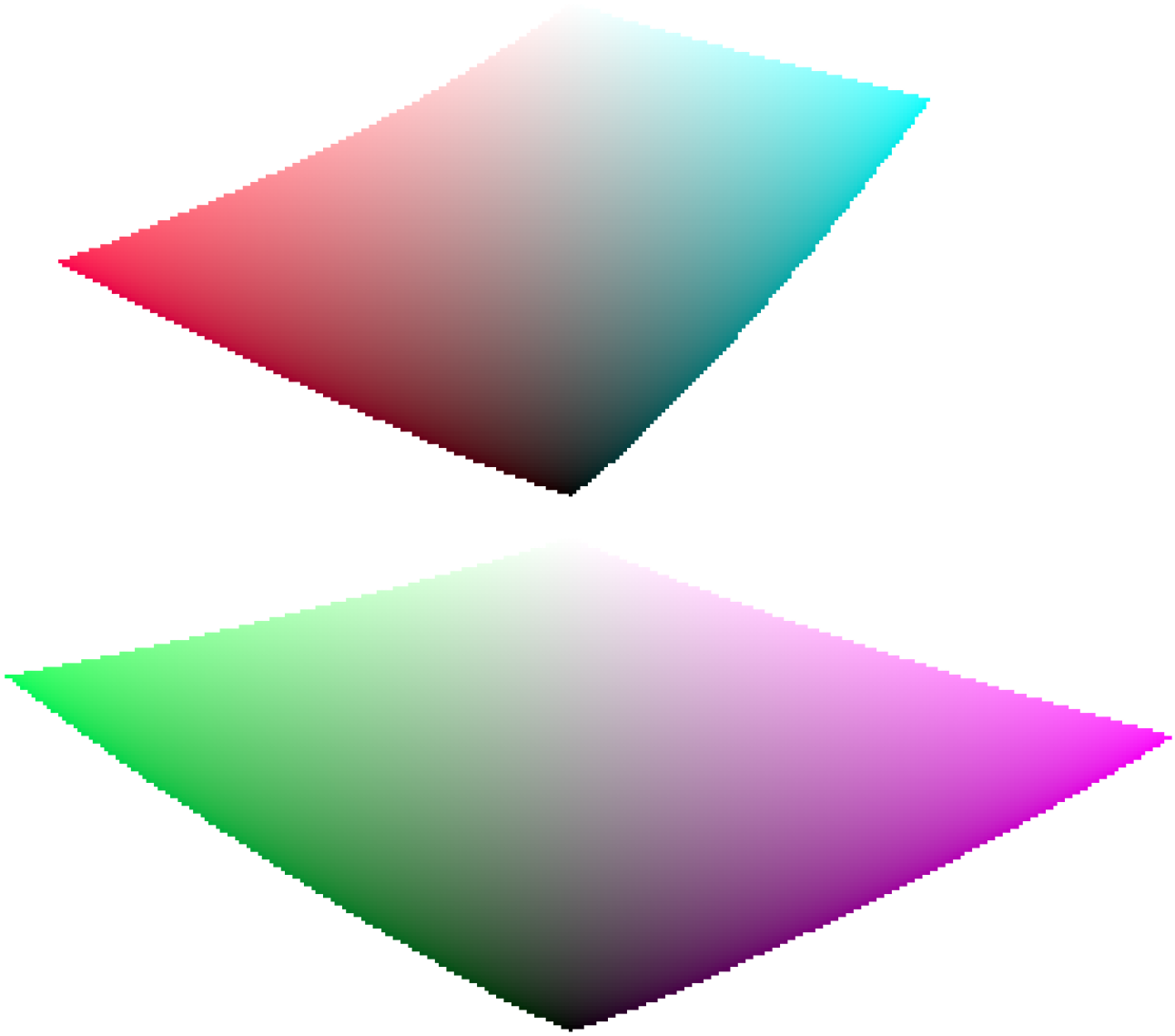


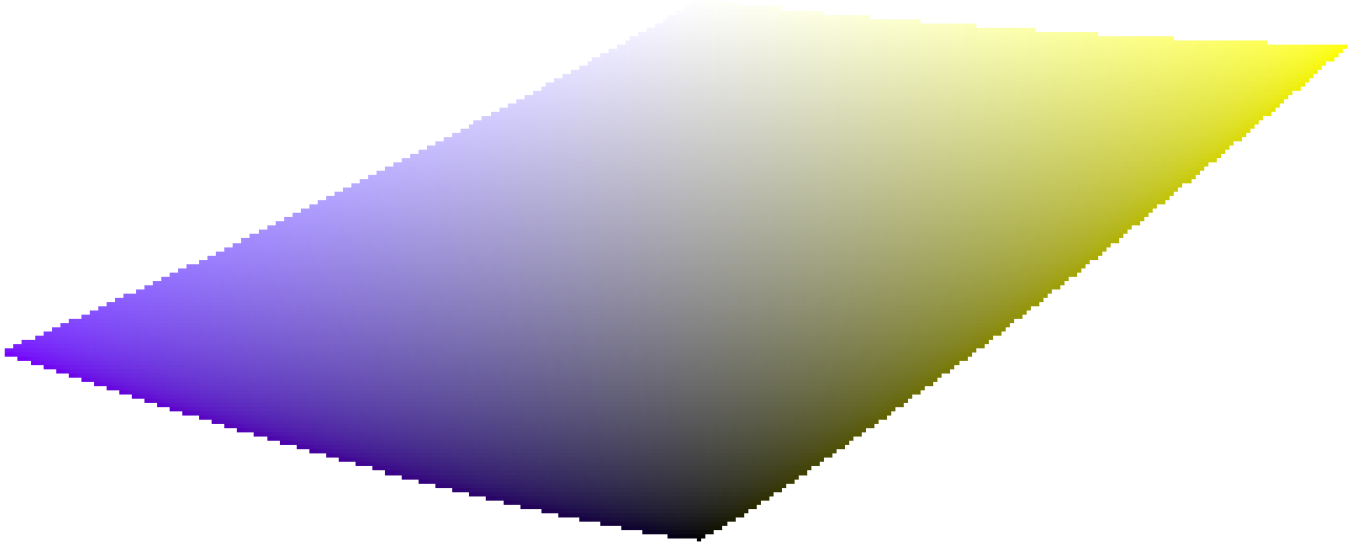
This shows that increasing the chroma (displacement over the horizontal axis) of some quantity can be safe for some hues at some lightness, but can push other hue-lightness coordinates way out of gamut. For example, we have much more margin in green or magenta than in cyan.

Many gamut issues at export are actually user-induced and the result of harsh chroma enlarging. For that reason, using brightness-saturation color models may be safer.

color dimensions and complementary colors

Cyan, magenta, yellow (CMY) are complementary colors of red, green, blue (RGB). However, the complementary CMY spaces computed from RGB spaces are not perceptually complementary. To show this, we create a CMY space from sRGB, where cyan has sRGB coordinates (0, 1, 1), magenta (1, 0, 1) and yellow (1, 1, 0), and display it in a lightness-chroma space:





By comparing with the hue slices of the primary colors in the previous section, it is easy to see not only that the gamuts don’t have the same shapes, but that the colors do not match.

This is one more reason to avoid using HSL/HSV spaces (derived from RGB spaces) to perform color editing: since these RGB spaces are not perceptually uniform in the first place, the resulting HSV/HSL spaces are not uniform either. While RGB spaces have some merit based on their connection to physical light, any process involving hue should go directly to perceptual spaces.

color dimensions and settings

Many applications, including darktable, call any settings that affect chroma “saturation” (for example, in color balance, “contrast/brightness/saturation”). This is a symptom of software trying to be accessible to non-professionals by using a common language. This is misleading, since saturation does exist and is quite different from chroma. In addition, many video specifications improperly call chroma “saturation”. Whenever darktable reuses such specifications, it uses the incorrect term from the specification rather than the proper color dimension term.

---

1. CIE definition of *hue*: https://cie.co.at/eilvterm/17-22-067 ↩
2. CIE definition of *luminance*: https://cie.co.at/eilvterm/17-21-050 ↩
3. CIE definition of *lightness*: https://cie.co.at/eilvterm/17-22-059 ↩
4. CIE definition of *brightness*: https://cie.co.at/eilvterm/17-22-063 ↩
5. CIE definition of *chroma*: https://cie.co.at/eilvterm/17-22-074 ↩
6. Article about *brilliance* (paywall): https://doi.org/10.1002/col.20128 ↩
7. CIE definition of *saturation*: https://cie.co.at/eilvterm/17-22-073 ↩

12.2. OpenCL

12.2.1. the background

Processing high resolution images is a demanding task requiring a modern computer. In terms of both memory and CPU power, getting the best out of a typical 15, 20 or 25 megapixel image can quickly take your computer to its limits.

darktable's requirements are no exception. All calculations are performed on 4 x 32bit floating point numbers. This is slower than "ordinary" 8 or 16 bit integer algebra, but eliminates all problems of tonal breaks or loss of information.

A great deal of optimization has been undertaken to make darktable as fast as possible. If you run a current version of darktable on a modern computer, you might not notice any "slowness". However, there are conditions and certain modules where you will feel (or hear from the howling of your CPU fan) how much your poor multi-core processor has to struggle.

That's where OpenCL comes in. OpenCL allows darktable to take advantage of the enormous power of modern graphics cards. Gamers' demand for highly detailed 3D worlds in modern shooters (as well as cryptocurrency mining) has fostered rapid GPU development. AMD, NVIDIA and Co had to put enormous processing power into their GPUs to meet these demands. The result is modern graphics cards with highly parallelized GPUs that can quickly calculate surfaces and textures at high frame rates.

You are not a gamer and you don't take advantage of that power? Well, then you should at least use it in darktable! For the task of highly parallel floating point calculations modern GPUs are much faster than CPUs. This is especially true when you want to repeat the same few processing steps millions of times. Typical use case: processing high megapixel images.

Note that this is separate from GPU acceleration of darktable's AI features , which run through their own ONNX Runtime execution providers rather than OpenCL – see AI GPU acceleration for that setup.

12.2.2. how OpenCL works

As you can imagine, the hardware architecture of GPUs can vary significantly. There are different manufacturers, and even different generations of GPUs from the same manufacturer may not be comparable. At the same time GPU manufacturers don't normally disclose all the hardware details of their products to the public. One of the consequences of this is the need to use proprietary drivers under Linux, if you want to take full advantage of your graphics card.

Fortunately an industry consortium lead by The Khronos Group has developed an open, standardized interface called OpenCL, which allows your GPU to be used as a numerical processing device. OpenCL offers a C99-like programming language with a strong focus on parallel computing. An application that wants to use OpenCL will need OpenCL source code that it hands over to a hardware-specific compiler at run-time. This way applications can use OpenCL on different GPU architectures (even at the same time). All of the hardware "secrets" are hidden in this compiler and are normally not visible to the user (or the application). The compiled OpenCL code is loaded onto your GPU and – with certain API calls – it is ready to perform calculations for you.

12.2.3. activating OpenCL in darktable

Using OpenCL in darktable requires that your PC is equipped with a suitable graphics card and that it has the required libraries in place. Most modern graphics cards from NVIDIA, Intel or AMD come with full OpenCL support. The OpenCL compiler is normally shipped as part of the proprietary graphics driver and is used as a dynamic library called `libOpenCL.so`. This library must be in a folder where it can be found by your system's dynamic linker.

When darktable starts, it will first try to find and load `libOpenCL.so` and – on success – checks if the available graphics card comes with OpenCL support. A minimally-sufficient amount of graphics memory (1GB+) needs to be available for darktable to take advantage of the GPU. If that check passes, darktable tries to setup its OpenCL environment: a processing context needs to be initialized, a calculation pipeline to be started, OpenCL source code files (extension is `.cl`) needs to be read and compiled and the included routines (OpenCL kernels) need to be prepared for darktable's modules. If all of that completes successfully, the preparation is complete.

By default, OpenCL support is activated in darktable if all the above steps were successful. If you want to de-activate it you can do so in preferences > processing > OpenCL . This configuration parameter is grayed out if the OpenCL initialization failed.

You can switch OpenCL support off and on at any time without requiring a restart. Depending on the type of modules you are using, you will notice the effect as a general speed-up during interactive work and export. Most modules in darktable can take advantage of OpenCL but not all modules are demanding enough to make a noticeable difference. In order to feel a real difference, use modules like *diffuse or sharpen* , and *denoise (profiled)* .

If you are interested in profiling statistics, you can start darktable with command line parameters `-d opencl -d perf`. After each run of the pixelpipe you will be shown details of processing time for each module plus an even more fine-grained profile for all used OpenCL kernels.

Apart from the speed-up you should not see any difference in the results between CPU and GPU processing. Except for some rounding errors, the results are designed to be identical. If, for some reason, darktable fails to properly finish a GPU calculation, it will normally detect the failure and automatically (and transparently) fall back to CPU processing.

12.2.4. setting up OpenCL

The huge diversity of systems and the marked differences between OpenCL vendors and driver versions makes it impossible to give an comprehensive overview of how to setup OpenCL. We only can give you an example, in this case for NVIDIA driver version 542.29.06 on Fedora 39. We hope that this will serve as a basic introduction and will help you to solve any problems specific to your setup.

The principle OpenCL function flow is like this:

```
darktable > libOpenCL.so > libnvidia-opencl.so.1 > kernel driver module(s) > GPU
```

- darktable dynamically loads `libOpenCL.so` – a system library that must be accessible to the system's dynamic loader (`ld.so`).
- `libOpenCL.so` reads the vendor-specific information file (`/etc/OpenCL/vendors/nvidia.icd`) to find the library that contains the vendor-specific OpenCL implementation.
- The vendor-specific OpenCL implementation comes as a library `libnvidia-opencl.so.1` (which in our case is a symbolic link to `libnvidia-opencl.so.545.29.06`).
- `libnvidia-opencl.so.1` needs to talk to the vendor-specific kernel modules `nvidia` and `nvidia_uvm` via device special files `/dev/nvidia0`, `/dev/nvidiactl`, and `/dev/nvidia-uvm`.

At system startup the required device special files (`/dev/nvidia*`) need to be created. If this does not happen on your system by default, the easiest way to set them up and make sure all modules are loaded is by installing the `nvidia-modprobe` package.

A user account that needs to make use of OpenCL from within darktable must have read/write access to NVIDIA's device special files. On some systems these files allow world read-write access by default, which avoids permission issues but might be debatable in terms of system security. Other systems restrict the access to a user group, e.g. "video". In this case your user account has to be member of that group.

To summarise, the packages that needed to be installed in this specific case were:

```
xorg-x11-drv-nvidia
xorg-x11-drv-nvidia-libs
xorg-x11-drv-nvidia-cuda
xorg-x11-drv-nvidia-cuda-libs
xorg-x11-drv-nvidia-power
akmod-nvidia
nvidia-settings
nvidia-modprobe
nvidia-persistenced
opencl-headers
opencl-filesystem
ocl-icd
ocd-icd-devel
```

On Linux systems you might also want the `clinfo` package giving you a lot of information about your OpenCL system and settings.

12.2.5. possible problems & solutions

darktable will detect OpenCL run-time errors automatically. On detecting an error, it will then reprocess everything on the CPU. While this will slow down processing it should not affect the end result.

There can be various reasons why OpenCL might fail during the initialization phase. OpenCL depends on hardware requirements and on the presence of certain drivers and libraries. In addition all these have to fit in terms of maker, model and revision number. If anything does not fit (e.g. your graphics driver – loaded as a kernel module – does not match the version of your `libOpenCL.so`) OpenCL support will likely not be available.

In this case, the best thing to do is start darktable from a console with `darktable -d opencl`.

This will give additional debugging output about the initialization and use of OpenCL. First, if you find a line that starts with `[opencl_init] FINALLY ...` that should tell you whether OpenCL support is available for you or not. If initialization failed, look at the messages above for anything that reads like `could not be detected` or `could not be created`. Check if there is a hint about where it failed.

Here are a few cases that have been observed in the past:

- darktable states that no OpenCL aware graphics card is detected or that the available memory on your GPU is too low and the device is discarded. In that case you might need to buy a new card if you really want OpenCL support.
- darktable finds your `libOpenCL.so` but then tell you that it couldn't get a platform. NVIDIA drivers will often give error code -1001 in this case. This happens because `libOpenCL.so` is only a wrapper library. For the real work further vendor-specific libraries need to be loaded. This has failed for some reason. There is a structure of files in `/etc/OpenCL` on your system that `libOpenCL.so` consults to find these libraries. See if you can find something fishy in there and try to fix it. Often the required libraries cannot be found by your system's dynamic loader. Giving full path names might help.
- darktable states that a context could not be created. This often indicates a version mismatch between the loaded graphics driver and libOpenCL. Check if you have left-over kernel modules or graphics libraries from an older installation and take appropriate action. When in doubt, perform a clean reinstall of your graphics driver. Sometimes, immediately after a driver update, the loaded kernel driver does not match the newly installed libraries. In this case reboot your system before trying again.
- darktable crashes during startup. This can happen if your OpenCL setup is completely broken or if your driver/library contains a severe bug. If you can't fix it, you can still use darktable with option `--disable-opencl`, which will skip the entire OpenCL initialization step.
- darktable fails to compile its OpenCL source files at run-time. In this case you will see a number of error messages looking like typical compiler errors. This could indicate an incompatibility between your OpenCL implementation and darktable's interpretation of the standard. In that case please raise an issue on github and we will try to assist. Please also report if you see significant differences between CPU and GPU processing of an image.
- you have installed a number of OpenCL drivers meant for the same hardware, this will always lead to severe problems and must strictly be avoided. On Windows systems you often have the `Microsoft OpenCLon12` driver installed via the `OpenCL Compatibility Pack`. Inspect and check at preferences > processing > OpenCL .
- A few emulated-on-CPU implementations of OpenCL also exist, coming as drivers provided by INTEL or AMD. We have observed that they do not provide any speed gain versus the compiler-optimized CPU code. Therefore darktable simply discards these drivers / devices by default.

12.2.6. scheduling profile

darktable can use the CPU and one or several OpenCL capable GPUs. Depending on the relative performance of these devices, users can choose among certain scheduling profiles to optimize performance. This is achieved by setting the configuration parameter preferences > processing > OpenCL > OpenCL scheduling profile , which offers the following choices:

default

If an OpenCL-capable GPU is found darktable uses it for processing the center image view while the navigation preview window is processed on the CPU in parallel. This is the preferred setting for systems with a reasonably fast CPU and a moderately fast GPU. The exact allocation of devices to the various pixelpipe types can be finetuned with the "opencl_device_priority" configuration parameter (see multiple devices).

very fast GPU

With this scheduling profile darktable processes the center image view and the preview window on the GPU sequentially. This is the preferred setting for systems with a GPU that strongly outperforms the CPU.

multiple GPUs

This setting addresses systems with multiple GPUs whose relative performance do **not** differ significantly. Whenever a processing job is started darktable uses any currently idle GPU but not the CPU. Users of systems with a variety of GPUs will need better control on their relative priority. They would be better off selecting the “default” profile and fine-tuning their system with the “opencl_device_priority” configuration parameter (see multiple devices).

On first start-up or after any detected change in the GPU configuration of your system darktable tries to identify the best suited profile for you. You can change it at any time in preferences > processing > OpenCL .

12.2.7. multiple devices

The scheduling of OpenCL devices can be optimized on most systems using the “OpenCL scheduling profile” settings. However, if your system is equipped with more than one GPU, you might want to set the relative device priority manually. To do this you need to select the “default” scheduling profile and change the settings in the “opencl_device_priority” configuration parameter.

It is important to understand how darktable uses OpenCL devices. Each processing sequence of an image – to convert an input to the final output using a history stack – is run in a pixelpipe. There are five different types of pixelpipe in darktable. One type is responsible for processing the center image view (or full view) in darkroom mode, another pixelpipe processes the preview image (navigation window, also required for histograms and more internal stuff relevant for correct output of the full view). Another preview pixelpipe would be required for showing a second darkroom window. There can be one of each of these three pixelpipes running at any time, with the full and preview pixelpipes running in parallel. In addition there can be multiple parallel pixelpipes performing file exports as well as multiple parallel pixelpipes generating thumbnails. If an OpenCL device is available, darktable dynamically allocates it to one specific pixelpipe for one run and releases it afterwards.

The computational demand varies significantly depending on the type of pixelpipe being executed. The preview image and thumbnails are low resolution and can be processed quickly, whereas processing the center image view or the second window is more demanding. A full export pixelpipe is more demanding still.

The configuration parameter “opencl_device_priority” holds a string with the following structure:

```
a,b,c.../d,e,f.../g,h,i.../j,k,l.../m,n,o...
```

Each letter represents one specific OpenCL device. There are five fields in the parameter string separated by a slash, each representing one type of pixelpipe. `a,b,c...` defines the devices that are allowed to process the center image (full) pixelpipe. Likewise devices `d,e,f...` can process the preview pixelpipe, devices `g,h,i...` the export pixelpipes, devices `j,k,l...` the thumbnail pixelpipes and finally devices `m,n,o...` preview pixelpipe for the second window. An empty field means that no OpenCL device may serve this type of pixelpipe.

darktable has an internal numbering system, whereby the first available OpenCL device receives the number 0. All further devices are numbered consecutively. This number, together with the device name, is displayed when you start darktable with `darktable -d opencl`. You can specify a device either by number or by its canonical name (upper/lower case and whitespace do not matter). If you have more than one device with the same name you need to use the device numbers in order to differentiate them.

A device specifier can be prefixed with an exclamation mark `!`, in which case the device is excluded from processing a given pixelpipe. You can also use an asterisk * as a wildcard, representing all devices not previously explicitly mentioned in that group.

Sequence order within a group matters – darktable will read the list from left to right and whenever it tries to allocate an OpenCL device to a pixelpipe it will scan the devices in that order, taking the first free device it finds.

If a pixelpipe process is about to be started and all GPUs in the corresponding group are busy, darktable automatically processes the image on the CPU by default. You can enforce GPU processing by prefixing the list of allowed GPUs with a plus sign +. In this case darktable will not use the CPU but rather suspend processing until the next permitted OpenCL device is available.

darktable’s default setting for “opencl_device_priority” is `*/!0,*/*/*/!0,*`.

Any detected OpenCL device is permitted to process the center view image. The first OpenCL device (0) is not permitted to process both preview pixelpipes. As a consequence, if there is only one GPU available on your system, the preview pixelpipes will always be processed on the CPU, keeping your single GPU exclusively for the more demanding center image view. This is a reasonable setting for most systems. No such restrictions apply to export and thumbnail pixelpipes.

The default is a good choice if you have only one device. If you have several devices it forms a reasonable starting point. However, as your devices might have quite different levels of processing power, it makes sense to invest some time optimizing your priority list.

Here is an example. Let’s assume we have a system with two devices, a fast Nvidia Quadro RTX 4000 and a slower GeForce GTX 1050. darktable (started with `darktable -d opencl`) will report the following devices:

```
[opencl_init] successfully initialized.
[opencl_init] here are the internal numbers and names of
                         OpenCL devices available to darktable:
[opencl_init]           0       'NVIDIA GeForce GTX 1050'
[opencl_init]           1       'NVIDIA CUDA Quadro RTX 4000'
[opencl_init] FINALLY: opencl is AVAILABLE on this system.
```

with the canonical names shown above as `nvidiagforcegtx1050` and `nvidiacudaquadrortx4000`

Here, the GeForce GTX 1050 is detected as the first device and the Quadro RTX 4000 as the second. This order will normally not change unless the hardware or driver configuration is modified, but it's better to use device names rather than numbers to be on the safe side.

As the GTX 1050 is slower than the RTX 4000, an optimized "opencl_device_priority" could look like:

`!nvidiagforcegtx1050,*/!nvidiacudaquadrortx4000,*/nvidiacudaquadrortx4000,*/nvidiacudaquadrortx4000,*/!nvidiacudaquadrortx4000`.

The GTX 1050 is explicitly excluded from processing the center image pixelpipe; this is reserved to "all" other devices (i.e. the RTX 4000). Conversely – for the preview pixelpipes – the RTX 4000, so that only the GTX 1050 is permitted to do the work.

For file export and thumbnail generation we want all hands on deck. However, darktable should first check whether the RTX 4000 device is free, because it's faster. If it is not free, then all other devices – in fact only the GTX 1050 – are checked.

12.2.8. OpenCL still does not run for me

As has been mentioned, OpenCL systems come with a huge variety of setups: different GPU manufacturers and models, varying amounts of GPU memory, different drivers, different distributions etc..

Many of the potential problems will only appear with very specific combinations of these factors. As the darktable developers only have access to a small fraction of these variations, please understand that we might not be able to fix your specific problem. There is not much we can do if we are unable to reproduce your issue.

If you are having issues with a specific device (and you have other OpenCL devices available) the first thing to do would be to try disabling just that device – see memory & performance tuning for more information.

If nothing else helps, the best option is probably to start darktable with

```
darktable --disable-opencl
```

In the end there is nothing in darktable that only runs on GPU. Don't let the lack of OpenCL discourage you – darktable's CPU code is also highly optimized for performance!

12.3. AI features

12.3.1. overview

darktable uses machine learning models for a handful of image-processing tasks: interactive object masking, denoising, upscaling, etc. This section explains how those features work end-to-end – what runs where, which models exist, how inference is accelerated, and what the user's choices in the AI preferences actually control.

AI features are opt-in. They are off by default in a fresh install and can be toggled with a single preference at any time.

AI features also have to be present in the darktable binary. The official builds (Linux AppImage, Windows installer, macOS dmg) all include them. Third-party packagers may ship a build without AI support – in that case the *AI* tab is absent from preferences entirely. Ask your distribution's package maintainer to enable AI support, or use an official build.

what the AI features do

tasks

A *task* is a capability slot inside darktable – “produce a mask around the clicked subject”, “denoise an RGB image”, and so on. Tasks are decoupled from the model that implements them: multiple competing models can live on disk for the same task (the NIND denoiser, the NAFNet denoiser, an experimental third...), but only one of them is *active* for that task at any given time, and that’s the one darktable’s modules consume. Switching active model is a one-click toggle in AI preferences and requires no changes to any module – the consumer just asks for “the active denoise model” and gets whichever you chose.

This setup allows for flexibility so the model repository can publish alternative or improved models without darktable needing to know about them ahead of time: pick the model in preferences and every consumer of that task picks it up automatically.

darktable currently exposes four tasks:

mask
: Interactive object segmentation. Click on a subject in the darkroom and the model produces a mask around it, vectorised into a group of path shapes that can be edited like any other drawn shape. Used by the AI object drawn-mask shape to select regions without drawing by hand. Powered by models such as SAM 2.1 and SegNext.

denoise
: Machine-learning RGB denoise as an alternative to the classical denoise modules. Operates on already demosaiced RGB image data, trained on pairs of noisy and clean images. Removes sensor noise without the blocky or smoothed look that can come from traditional filters. Used by the neural restore module’s *denoise* tab.

rawdenoise
: Machine-learning denoise applied directly to the raw CFA mosaic before demosaic, with denoise and demosaic combined into a single inference pass. Saved out as a LinearRaw DNG and re-imported, so downstream darkroom processing is unchanged. Useful when sensor noise needs to be addressed before the demosaic algorithm runs. Used by the neural restore module’s *raw denoise* tab.

upscale
: 2x and 4x super-resolution, producing plausible detail from low-resolution inputs. Useful for small or cropped images where classical interpolation would blur. Used by the neural restore module’s *upscale* tab.

Tasks run only when a user triggers them – there is no continuous AI activity in the background, no telemetry, and no cloud component. Everything runs locally on your machine.

a note on accuracy

Like any machine-learning system, the AI features in darktable are not perfect. Every model has edge cases where its output will look worse than the classical alternative – unusual subjects, extreme exposures, or content far from what the model was trained on. This is intrinsic to how ML works, not a bug: 100% accuracy across all possible inputs is not something any model – ours or anyone else’s – can offer.

What the models *can* offer is consistently good results in the common cases they were designed for. The right mindset is to treat AI features as a tool that’s usually better than the classical alternative, and to keep the classical modules available as a fallback on the few images where AI output is not. If you find a case where an AI model produces a noticeably poor result, please consider reporting it so the model can be improved over time.

what darktable does *not* do

darktable uses *narrow AI* – small, task-specific models that run on your laptop and do one well-defined job (denoise, mask, upscale). It does **not** use the large generative models that can rewrite photographs – no sky replacement, no inserted objects, no style transfer, no synthesised content of any kind. The full scope is set out in the project's AI Model Integration Policy ; the governing principle is **scene integrity**: AI may correct technical defects of the capture, but it may not change what was in front of the camera.

Beyond scope, the technical guarantees:

- **No cloud inference:** Every model runs locally on your CPU or GPU. Nothing about your image is sent off-device.
- **No autonomous edits:** Every AI feature is user-triggered; there's no background AI activity on your library.
- **No training:** darktable ships pre-trained models and runs inference only. The darktable-ai repository holds the conversion scripts and training notes for anyone who wants to retrain or export a new model.
- **No automatic updates:** Models are downloaded when you ask for them in preferences; nothing happens in the background.
- **No analytics:** darktable does not report model use, provider selection, or inference performance anywhere.

12.3.2. how AI features work

This page covers what goes on under the hood: the inference library darktable loads, the execution providers available to it, and how models are stored and activated.

the inference runtime

darktable loads models through ONNX Runtime – a production-grade inference library originally developed at Microsoft and now governed by the Linux Foundation – that accepts models in the ONNX open format. ONNX is the de-facto standard for interchanging machine-learning models across frameworks (PyTorch, TensorFlow, etc.) so we can take a model trained in any of them, export it to ONNX, and run it from darktable without carrying its training framework along.

Only one ONNX Runtime library is loaded per darktable process. It is picked in this order:

1. If the *ONNX Runtime library* field in AI preferences points at a file (or the `DT_ORT_LIBRARY` environment variable does), that library is loaded instead.
2. Otherwise, the bundled runtime that ships with darktable is loaded from the installation directory.

The bundled runtime is CPU-only on Linux, DirectML-enabled on Windows, and CoreML-enabled on macOS. That makes every install functional out of the box. If you want NVIDIA, AMD or Intel GPU acceleration on Linux or Windows, you replace the library with a GPU-enabled build – see GPU acceleration .

execution providers

An *execution provider* (EP) is an ONNX Runtime back-end for a specific hardware accelerator – darktable's *AI acceleration* preference is this concept. A single ONNX Runtime library can ship with multiple providers; which ones actually work depends on which are compiled into the library and which are available on your system. The preference combo lists only those the currently-loaded library supports:

- **auto** – let ONNX Runtime pick the best provider at inference time, with CPU as the ultimate fallback. Recommended unless you have a reason to override.
- **CPU** – always available. Fine for object-mask and occasional denoise/upscale. Slow for interactive work and for upscale at 4x.
- **NVIDIA CUDA** – NVIDIA GPUs on Linux/Windows. Requires a CUDA-enabled ONNX Runtime and the matching CUDA toolkit + cuDNN 9.x on your system.
- **AMD MIGraphX** – AMD GPUs on Linux. Requires a MIGraphX-enabled ONNX Runtime and a ROCm 6.3+ install. Not supported on Windows.
- **Intel OpenVINO** – Intel integrated and discrete GPUs on Linux/Windows. The OpenVINO runtime is bundled with the installed package – no separate install needed beyond an up-to-date Intel GPU driver.
- **Windows DirectML** – any DirectX-12-capable GPU on Windows. Good fallback when a vendor-specific provider is impractical. Bundled.
- **Apple CoreML** – Apple Silicon (Neural Engine and GPU) and recent Intel Macs. Bundled.

The first time a GPU provider runs a given model, it compiles an optimised version of the model graph for your specific hardware. How long this takes varies a lot by provider: CUDA and DirectML are typically a few seconds; AMD MIGraphX can take **5 to 30 minutes** on the first run of each model, depending on the GPU and the model size. Subsequent uses reuse the compiled graph and run at full speed.

models

Each AI feature is backed by one or more models per task . A model is a directory containing an ONNX file and a small `config.json` with metadata (name, task, description, licensing information). Models live under:

- Linux: `~/.local/share/darktable/models/`
- macOS: `~/Library/Application Support/darktable/models/`
- Windows: `%APPDATA%\darktable\models\`

darktable scans this directory at startup (and whenever you change models in preferences) to discover what is available. Multiple models can be installed for the same task – for denoise, for example, you might have both the NIND-trained and NAFNet-trained options on disk at once. Only one can be *active* per task at any time; ticking the enabled checkbox on a second model of the same task automatically un-ticks the previously active one. Modules that consume that task load the model currently marked active.

Models are not bundled with darktable. They are distributed as `.dtmodel` packages from the darktable-ai companion repository. The AI preferences tab includes buttons to download the default set directly; for additional per-task models (alternative denoisers, segmentation backbones, etc.), download the `.dtmodel` from the repository's releases page and install it via the *install model* button in the same tab.

12.3.3. GPU acceleration

Enabling GPU acceleration for darktable's AI features is a two-step process:

1. **Install a GPU-capable ONNX Runtime** for your hardware (this page).
2. **Point darktable at the new library** via AI preferences – click **detect** to auto-scan standard install locations, or browse to the file by hand.

On Linux, the bundled runtime is CPU-only; on Windows, it uses DirectML; and on macOS, it uses CoreML. So, step 1 only applies if you want NVIDIA, AMD, or Intel GPU acceleration on Linux or Windows. Step 1 has three paths, covered below.

requirements

The bundled providers on Windows (DirectML, any DirectX 12 GPU) and macOS (CoreML, Apple Silicon or recent Intel Mac) have no extra requirements. The Linux/Windows GPU providers below need supported hardware and a few system libraries already in place.

NVIDIA (CUDA)

Pascal-or-newer GPU (compute capability 6.0+). NVIDIA driver 525 or later. Plus **one of** CUDA Toolkit 12.x or CUDA Toolkit 13.x and cuDNN 9.x installed on the system. The two CUDA branches have separate ONNX Runtime builds and you need the one that matches the toolkit version installed on your machine.

To check what CUDA toolkit you have:

- run `nvcc --version` in a terminal – the `release` line tells you the installed toolkit (e.g. `release 12.6` means CUDA 12 is installed).
- `nvidia-smi` reports a `CUDA Version:` field at the top, but that is the **maximum** version the driver is capable of running, not what's installed. A machine with driver 580+ might show `CUDA Version: 13.2` while actually having only CUDA 12 installed via `nvcc`.

If you have CUDA 12 installed and want to switch to 13, install the CUDA 13 toolkit (in parallel with 12 is fine) and then re-run the ONNX Runtime install script – it picks the build matching the toolkit it finds.

AMD (MIGraphX) *(Linux only)*

ROCm-supported GPU (Radeon RX 7700-series or newer, Instinct MI100 or newer). ROCm 6.x or newer with MIGraphX installed on the system.

Note: the first inference of each model on MIGraphX takes minutes because the model graph is compiled for your specific GPU and then cached on disk – see the performance notes for details. Once cached, subsequent runs are fast and survive across darktable restarts.

Intel (OpenVINO)

Intel iGPU (HD / UHD / Iris Xe) or Arc discrete GPU. Intel GPU driver with OpenCL (`intel-opencl-icd`) and/or Level Zero . OpenVINO itself ships in the install package – nothing extra to install.

If a required system library is missing, the install script will tell you which one and point you at the vendor's install docs. The package still installs; acceleration only kicks in once the missing library is in place.

use the install scripts (recommended)

The simplest path: paste one line into a terminal. The script figures out which GPU you have, downloads the matching ONNX Runtime build, verifies it, and installs it under your user account. No root, no source checkout.

Linux:

curl -fsSL https://raw.githubusercontent.com/darktable-org/darktable/HEAD/tools/ai/install-ort-gpu.sh | bash

Windows (PowerShell):

irm https://raw.githubusercontent.com/***darktable-org***/darktable/HEAD/tools/ai/***install-ort***-gpu.ps1 | ***iex***

The Linux script needs `bash`, `curl`, `jq`, `tar`, and `unzip` available on `PATH` – if any are missing it tells you which ones up front so the install doesn't fail halfway through. The Windows script needs PowerShell 5.1 or newer (preinstalled on Windows 10+).

If you have more than one GPU, the script asks which one to target. If your GPU needs system libraries that aren't installed (cuDNN on NVIDIA, MIGraphX on AMD), the script tells you what's missing and where to get it – the install still completes, but acceleration only kicks in once those system libraries are in place.

To force a specific execution provider instead of letting the script auto-detect, pass `--ep` (bash) or `-Ep` (PowerShell). Values: `cuda12`, `cuda13`, `rocm`, `migraphx`, `openvino`. Useful when both NVIDIA and Intel GPUs are present and you want a specific one, or when `nvcc` is not installed and the script can't tell whether you have CUDA 12 or CUDA 13 on the system.

When the script finishes, restart darktable, open AI preferences , and click **detect** so darktable picks up the newly-installed runtime. Then choose your provider (CUDA, MIGraphX, OpenVINO) under *AI acceleration*.

detect a system-installed ONNX Runtime

If your distribution already ships a GPU-capable ONNX Runtime (common on Arch, occasionally on Ubuntu/Debian via third-party repos), just click **detect** in the AI preferences tab – darktable looks in standard system locations and configures itself.

install manually

When the automatic options don't fit (locked-down environment, custom audit, GitHub API rate-limited – see below) you can fetch and place the files yourself. The goal is a `libonnxruntime.so.*` / `onnxruntime.dll` on disk that darktable can be pointed at via the *ONNX Runtime library* field in AI preferences.

where to download

Vendor	Source
NVIDIA CUDA 12	GitHub Releases – `onnxruntime-{linux|win}-x64-gpu-X.Y.Z.{tgz|zip}`
NVIDIA CUDA 13	GitHub Releases – `onnxruntime-{linux|win}-x64-gpu_cuda13-X.Y.Z.{tgz|zip}`
AMD ROCm 6.x	PyPI `onnxruntime-rocm` – `cp312 manylinux` wheel
AMD ROCm 7.x	PyPI `onnxruntime-migraphx` – `cp312 manylinux` wheel
Intel OpenVINO (Linux)	PyPI `onnxruntime-openvino` – `cp312 manylinux_2_28_x86_64` wheel
Intel OpenVINO (Windows)	PyPI `onnxruntime-openvino` + PyPI `openvino` – both `cp312 win_amd64`

NVIDIA tarballs/zips are simplest to extract: the library sits under `lib/`. PyPI wheels (`.whl` files are renamed zips) carry the library under `<package>/capi/` along with the `LICENSE` and `ThirdPartyNotices.txt` you should keep next to it. AMD wheels may additionally ship a sibling `<package>.libs/` directory of auditwheel-bundled ROCm libs – when present, that directory must stay alongside `libonnxruntime.so.*` after copying, because the providers' RPATH resolves through a relative `$ORIGIN` reference. Intel on Windows needs DLLs from a second wheel (`openvino`) copied next to `onnxruntime.dll`.

By convention, place the extracted files under `~/.local/lib/onnxruntime-<ep>/` on Linux or `%LOCALAPPDATA%\onnxruntime-<ep>\` on Windows (`<ep>` = `cuda`, `rocm`, `migraphx`, or `openvino`). The **detect** button in AI preferences scans for any `onnxruntime-*` directory under those roots.

AMD: alternative source when PyPI's GPU coverage isn't enough

PyPI's `onnxruntime-rocm` / `onnxruntime-migraphx` wheels ship HIP kernel binaries for a fixed set of GPU architectures. If your card's `gfx*` ID isn't covered, ONNX Runtime loads but model compilation fails at runtime:

```
migraphx_program_compile: Error: ... no kernel image is available
for execution on the device
```

This commonly hits some RDNA 3 iGPUs (e.g. Radeon 780M / gfx1103). Two cheaper workarounds before reinstalling:

- **hide the unsupported GPU** from MIGraphX so it binds to a supported card (find indices with `rocminfo | grep -E "Marketing Name|gfx"`):

  HIP_VISIBLE_DEVICES=0 darktable

- **override the gfx version** – RDNA 3 is mostly binary-compatible with gfx1100, so the bundled kernels often work:

  HSA_OVERRIDE_GFX_VERSION=11.0.0 darktable

If neither helps, AMD's own ROCm repository ships builds with broader per-architecture coverage. Browse https://repo.radeon.com/rocm/manylinux/ , find the `rocm-rel-X.Y.Z/` directory matching your installed ROCm version, download the `cp312 onnxruntime_rocm-*.whl`, and extract it the same way as the PyPI wheel.

building AMD from source

When even the AMD repository binaries don't match your system (custom-built ROCm, an unusual distro), build ONNX Runtime against your installed ROCm. From a darktable source checkout:

./tools/ai/install-ort-amd-build.sh

Needs `cmake` 3.26+, a C++ compiler, `python3`, and `git`. Takes 10–20 minutes.

GitHub API rate limit

For CUDA 13, the install script asks `api.github.com` which release is latest. GitHub limits unauthenticated callers to **60 requests per hour per source IP**. On shared NAT (corporate networks, VPNs, cloud VMs) that limit gets exhausted quickly by other users on the same IP, and the script fails with a clear message pointing here.

Two ways out:

- **set `GITHUB_TOKEN`** to any personal access token (no scopes required for public repos) before re-running – raises the limit to 5000 requests/hour authenticated.
- **install manually** – for the CUDA 13 tarball this is `tar xzf` plus a copy. The install script prints the same instructions on failure.

PyPI sources (AMD, Intel, NVIDIA CUDA 12) have no published rate limit and are unaffected.

after installing a new ONNX Runtime

When darktable starts up with a new ONNX Runtime library, the *AI acceleration* dropdown in AI preferences re-populates with only the providers that library actually advertises. Older selections that are no longer supported are switched to another available GPU provider automatically, or to *auto* if no GPU is left.

Between the install and the restart, the dropdown shows a *restart to apply* hint next to the provider name – a reminder that the currently-loaded runtime is stale.

12.3.4. performance & troubleshooting

performance notes

- CPU inference is acceptable for object masks (small one-shot evaluations) but much slower for denoise and upscale. A CUDA / MIGraphX / OpenVINO / DirectML / CoreML runtime is strongly recommended if you use those regularly.
- Upscale at 4x on a large image is memory-hungry regardless of provider. Tiling is enabled automatically to keep memory in check, at a cost of some extra processing time.
- Loading ONNX Runtime lazily on first use means the very first AI operation after startup takes an extra moment. This is normal.
- When switching between execution providers (e.g. auto → CUDA), the new provider's first inference incurs a graph-compilation delay. After that, runtime is steady for the rest of the session.
- **AMD (MIGraphX) has a multi-minute first-run cost per model.** ROCm/MIGraphX compiles the model graph for your specific GPU the first time it sees it, which can take 1–3 minutes for large models (denoise, upscale); mask models are smaller and compile faster. The compiled graph is cached on disk in `ai_v*_migraphx_*` in the darktable cache directory and reused across darktable restarts, so subsequent runs of the same model are fast.
- Model discovery is cached at startup. If you drop a new `.dtmodel` file into the models directory manually while darktable is running, open AI preferences and toggle any control to force a rescan.

troubleshooting

Most AI problems fall into one of three buckets.

the feature doesn't appear at all

Check that AI features are enabled in AI preferences – the master toggle is off by default.

If the AI tab itself is missing from preferences, darktable was built without AI support. The official darktable builds (Linux AppImage, Windows installer, macOS dmg) all include it, but third-party packagers may ship a build without it. Ask your distribution's package maintainer to enable AI support, or switch to an official build.

If enabled and the feature is still missing, the required model may not be installed or active. Open AI preferences and confirm a model is ticked in the *enabled* column for the relevant task. Only one model per task can be active at a time.

inference fails or falls back to CPU unexpectedly

Run darktable from a terminal with `darktable -d ai` to see ONNX Runtime load messages and provider probe results. Typical causes:

- **`ort_library_path` points at a missing or corrupt file.** Reset the path in preferences (double-click the label) to fall back to the bundled runtime, or re-run *detect* after re-installing via the GPU install scripts .
- **The selected execution provider is missing a system dependency.** cuDNN for CUDA; MIGraphX and rocm-smi-lib for AMD; matching ROCm SONAMEs on Fedora-style distros that rebuild ROCm with their own sonames. The install scripts flag these before download; the `darktable -d ai` log will show the exact `dlopen` error at runtime.
- **The provider was initialised but failed to compile the model graph.** This sometimes happens on first-generation GPUs with outdated drivers. Update your driver, or fall back to a different provider (e.g. DirectML instead of CUDA on Windows).

results look off

AI denoise and upscale are restoration tasks – the model is inferring plausible output from a training prior. Slight color or contrast shifts are possible, especially for images outside the training distribution (extreme ISO, synthetic captures, very wide-gamut scenes).

If fidelity matters more than aggressive cleanup, in e.g. denoising an image, stick with the classical denoise (profiled) module.

further reading

- AI preferences – every control on the AI preferences tab.
- GPU acceleration – installing a GPU-capable ONNX Runtime (scripts, manual, per-vendor recipes).
- darktable-ai – the model repository, conversion scripts, and development notes.
- ONNX Runtime docs – upstream reference for execution providers, troubleshooting, and performance tuning.

12.4. memory & performance tuning

memory requirements

Processing a Raw image in darktable requires a great deal of system memory. A simple calculation makes this clear: For a 20 megapixel image, darktable requires a 4x32-bit floating point cell to store each pixel, meaning that each full image of this size will require approximately 300MB of memory just to store the image data. In order to actually process this image through a given module, darktable needs at least two buffers (input and output) of this size, with more complex modules potentially requiring several additional buffers for intermediate data. Without further optimization, anything between 600MB and 3GB of memory might be required to store and process image data as the pixelpipe executes. On top of this is darktable's code segment, the code and data of any dynamically-linked system libraries, as well as further buffers that darktable uses to store intermediate states (cache) for quick access during interactive work.

All in all, darktable requires *at least* 4GB of physical RAM plus 4 to 8GB of additional swap space to run but it will perform better the more memory you have.

As well as executing on your CPU, many darktable modules also have OpenCL implementations that can take full advantage of the parallel processing offered by your graphics card (GPU). Similarly, the more GPU memory you have, the better darktable will perform.

tiling

If darktable does not have sufficient memory to process the entire image in one go, modules may choose to use a “tiling strategy”, wherein the image is split into smaller parts (tiles) which are processed independently, and then stitched back together at the end. While this allows images to be processed with a much smaller memory footprint, it does also come with some down-sides:

- tiling is always slower – sometimes up to 10x slower, though for some modules the difference is negligible,
- tiling is not technically possible for some modules because of the nature of the underlying algorithms

For most systems, tiling will probably only be used for full-sized image exports, with interactive work in the darkroom being processed more efficiently. For best performance (and avoidance of tiling modes) you should run darktable alongside as few other applications as possible and configure darktable to use as much of your system and GPU memory as you can.

performance tuning

There are a number of configuration parameters that can help you to fine-tune your system’s performance. Some of these parameters are available in preferences > processing > CPU / memory and others need to be modified directly in darktable’s configuration file (`darktablerc` in the darktable configuration directory).

This section provides some guidance on how to adjust these settings.

how to test

In order to determine how much your modifications improve (or not) darktable’s performance, you will need one or more sample images to test with, and a method of assessing the speed of the pixelpipe.

For sample images, you are advised to use some of the more intensive modules, such as diffuse or sharpen or denoise (profiled) . Exports are likely to have more consistent and comparable timings between pipe runs than interactive work (and will also push your hardware more).

In order to obtain profiling information you need to start darktable from a terminal with `darktable -d opencl -d perf`. If you want more information about tiling you should use `darktable -d opencl -d tiling -d perf`.

Each time the pixelpipe is processed (when you change module parameters, zoom, pan, export etc.) you will see (in your terminal session) the total time spent in the pixelpipe and the time spent in each of the OpenCL kernels. The most reliable value is the total time spent in the pixelpipe and you should use this to assess your changes.

Note: The timings given for each individual module are unreliable when running the OpenCL pixelpipe asynchronously (see *asyncronous mode* below).

To allow for efficient processing with OpenCL it is essential that the GPU is kept busy. Any interrupts or a stalled data flow will add to the total processing time. This is especially important for the small image buffers used during interactive work, which can be processed quickly by a fast GPU. However, even short-term stalls of the pixelpipe can easily become a bottleneck.

On the other hand darktable’s performance during file exports is more or less only governed by the speed of the algorithms and the horse-power of your GPU. Short-term stalls will not have a noticeable effect on the total time of an export.

darktable resources

The “darktable resources” preference (in preferences > processing > CPU / memory) allows you to choose between four different approaches to allocating your system’s resources to darktable. Each of these options controls multiple individual parameters, which are defined independently in `darktablerc` in the darktable configuration directory . You can amend any of these directly within your darktablerc file to tweak values for your selected resource level, though you cannot add your own custom resource level to the preferences drop-down.

Each of the four “darktable resources” options are defined as follows:

```
resource_default=512 8 128 700
resource_large=700 16 128 900
resource_small=128 4 64 400
```

More generally, these can be represented as `resource_level=a b c d` where a - d are defined as follows:

a. system memory for module processing

The maximum amount of system memory made available for module processing. Lower values force memory-hungry modules to process images with an increasing number of tiles. This number is a fraction of the total amount of system memory, divided by 1024. For example, on a system with 16GB of total system memory the amount assigned by `resource_default` (in GB) is `16 * 512 / 1024`, or 8GB of system RAM.

b. minimum tiling buffer size

The minimum size of a single tiling buffer, similarly expressed as a fraction of total system memory. For example, on a system with 16GB of total system memory the amount assigned by `resource_default` (in GB) is `16 * 8 / 1024`, or 0.125GB of system RAM. Note that this setting is largely historic and is no longer of much practical use – you are advised to leave it at its default value.

c. thumbnail cache memory

The amount of memory to use for the thumbnail cache. Again, this is expressed as a fraction of total system memory and, on a 16GB system, the amount assigned by `resource_default` is `16 * 128 / 1024`, or 2GB of system RAM.

d. OpenCL (GPU) memory

The maximum amount of GPU memory made available for module processing. As with system memory, lower values will force memory-hungry modules to process images with an increasing number of tiles. Your GPU memory will likely also be used by other applications on your system. However, in contrast to system memory, your GPU is not able to take advantage of swap files and it is impossible for darktable to know how much memory is available at a given time. If this parameter is set too high, darktable could be forced to fall back to CPU processing (which will be significantly slower but stable and with correctly processed data) or darktable might crash and even make your system unusable. For this reason, the GPU memory parameter fraction also includes an extra 600MB of headroom in an attempt to avoid over-allocation of memory. For example, on a GPU with 6GB of memory, darktable will use approximately `(6 - 0.6) * 700 / 1024`, or 3.5GB of GPU RAM when using the `resource_default` level.

In addition to the resource levels presented in the UI the following options can be set via the command-line (e.g. `darktable --conf resourcelevel="notebook"`). These modes are designed for debugging tiling issues and testing performance of common systems on larger development machines. The following options are provided:

- “mini” (1GB ram, 2MB single buffer, 128MB thumbnail cache, 200MB OpenCL memory)
- “notebook” (4GB ram, 32MB single buffer, 512MB thumbnail cache, 1GB OpenCL memory)
- “reference” (8GB ram, 32MB single buffer, 512MB thumbnail cache, 2GB OpenCL memory)
- “common2026” (16GB ram, 64MB single buffer, 1GB thumbnail cache, 5.5GB OpenCL memory)

tuning GPU memory usage

If you want to make maximal use of your GPU memory for OpenCL, you have two options:

- Choose the “large” resource level. For a 6GB card, this will use approximately 5GB of GPU memory, leaving 1GB for the rest of your system. (recommended)
- Alter darktablerc to increase the last number (the OpenCL memory fraction) for your selected resource level. For example, increasing the OpenCL memory fraction to 950 would increase the available memory on a 6GB GPU to approximately 5.3GB. (absolutely not recommended)

balanced OpenCL versus CPU tiling

In most cases, running a processing module on a high-powered GPU (the OpenCL codepath) is significantly faster than running the same module using the CPU codepath. However, many users have fast multi-core CPUs with a large amount of system RAM, but a GPU with significantly lower capabilities (typically, integrated graphics with small amounts of dedicated memory). Use of OpenCL code in these cases can lead to excessive tiling, and it is often better to run a module without tiling using the CPU codepath than to attempt to use OpenCL with heavy tiling.

While processing the pipeline, darktable attempts to determine which mode will be best for a given module by estimating the expected workloads for both OpenCL and CPU codepaths. In most cases it will prefer the OpenCL codepath even if that would mean tiling the image, since OpenCL is typically much faster than running the CPU code (often as much as 10 times faster if it is a dedicated card).

If the ratio of estimated workloads (CPU vs GPU) is larger than the **advantage hint** (see below), darktable will use the CPU for processing that module, else it will use the GPU.

device-specific OpenCL configuration

The default darktable settings deliver a reasonable GPU performance on most systems. However, if you want to try to optimize things further, this section describes the relevant configuration parameters (all of which are set in your darktablerc file).

Most of the OpenCL-related options are managed with a “per device” strategy. The configuration parameter for each device looks like:

```
cldevice_v7_rusticlamdradeon8060sgraphics=events:on asyncmode:off device:on unifraction: 0.250
```

or, more generally

```
cldevice_version_canonicalname=events:on/off asyncmode:on/off device:on/off unifraction: float
```

An entry will be automatically created in darktablerc for each newly-detected device when you launch darktable for the first time, with the correct canonical device name and version number. The parameters are defined as follows and can be manually edited:

events

on = use OpenCL events when calling enqueue kernels (default); off = don’t use

This flag controls the use of OpenCL events when calling enqueue kernels.

asyncmode

on = use asynchronous mode; off = don’t use (default)

This flag controls how often darktable blocks the OpenCL pixelpipe to get a status on success/failure of the kernels that have been run. For optimum latency set this to *on*, so that darktable runs the pixelpipe asynchronously and tries to use as few interrupts/events as possible. If you experience OpenCL errors like failing kernels, reset the parameter to *off*. This will cause darktable to interrupt after each module so that you can more easily isolate any problems. Issues have been reported with some older AMD/ATI cards (like the HD57xx) which can produce garbled output if this parameter is set to *on*. If in doubt, leave it at its default of *off*.

device

on = enable device; off = disable device

If darktable detects a malfunctioning device it will automatically mark it as such by setting this parameter to *off*. If you have a device that reports a lot of errors you can manually disable it by setting this field to *off*. If darktable has disabled the device but you are sure it should be used you can re-enable it by setting this field to *on*.

unifraction

The unifraction is a float and can be chosen by the user in the 0.02-0.5 range (default: 0.25). Some OpenCL devices don’t have dedicated memory but share it with the CPU – Apple ARM silicon is one example but also onboard devices from Intel, AMD or ARM SOCs. As we want to keep system memory available for caching or CPU codepaths we restrict the amount of all memory used to the given fraction. So with a value of 0.5 and an Apple computer with 16GB of system RAM, OpenCL would be able to make use of 8GB.

Note: if darktable detects a “buggy” device configuration key it will be rewritten back to default values.

---

device-specific OpenCL disabling

It is possible to disable an OpenCL device for specific modules by adding a comma-separated list of modules to the *nocl* option in the *darktablerc* file, for example:

```
cldevice_v7_rusticlamdradeon8060sgraphics_nocl=colorbalancergb,denoiseprofile
```

or, more generally

```
cldevice_version_canonicalname_nocl=comma,separated,list,of,modules
```

Any module in this list will be executed on the CPU instead of OpenCL.

id-specific OpenCL configuration

Another device-specific configuration key is also provided, which takes into account both the device name **and** the device id (in case you have two identical devices). In this case, the usual key name `cldevice_version_canonicalname` is followed by `_idX` with X being the device id. For example, if the above example device was referred to as device 0, the configuration setting would (by default) be `cldevice_v7_rusticlamdradeon8060sgraphics_id0=headroom: 600`.

This configuration key currently only has a single parameter defined:

headroom (default 600)
: The amount of memory (in MB) that will **not** be used by darktable during OpenCL processing. This setting is valid if you have more than one hardware device and the tuned GPU memory switch has been switched on.

  If you are certain that no apps (or your OS) make use of the specific device you can set this parameter to 0 for the otherwise-unused device so that darktable will use all of that device's memory.

  The default of 600MB should be fine for most systems. If you find you run into performance problems due to darktable falling back to CPU, try changing it to 800 or larger.

other configuration keys

The following additional configuration keys are also available in darktablerc:

cldevice_version_canonicalname_building
: This option is used when compiling OpenCL kernels and may be provided for performance tuning or to work around bugs. You must remove any existing kernels in order to recompile them with the new options. Provide an empty string to recompile without any options. Remove the setting entirely to recompile with default options.

opencl_mandatory_timeout
: *default 400*

  If darktable wants to make use of any OpenCL device it has to reserve it for further usage. If that device is current used darktable will wait up to *opencl_mandatory_timeout* * *5ms* before it does a fallback to CPU. Increase this value if you would prefer to use OpenCL (because your card is really fast and your CPU is not).

12.5. using darktable-chart

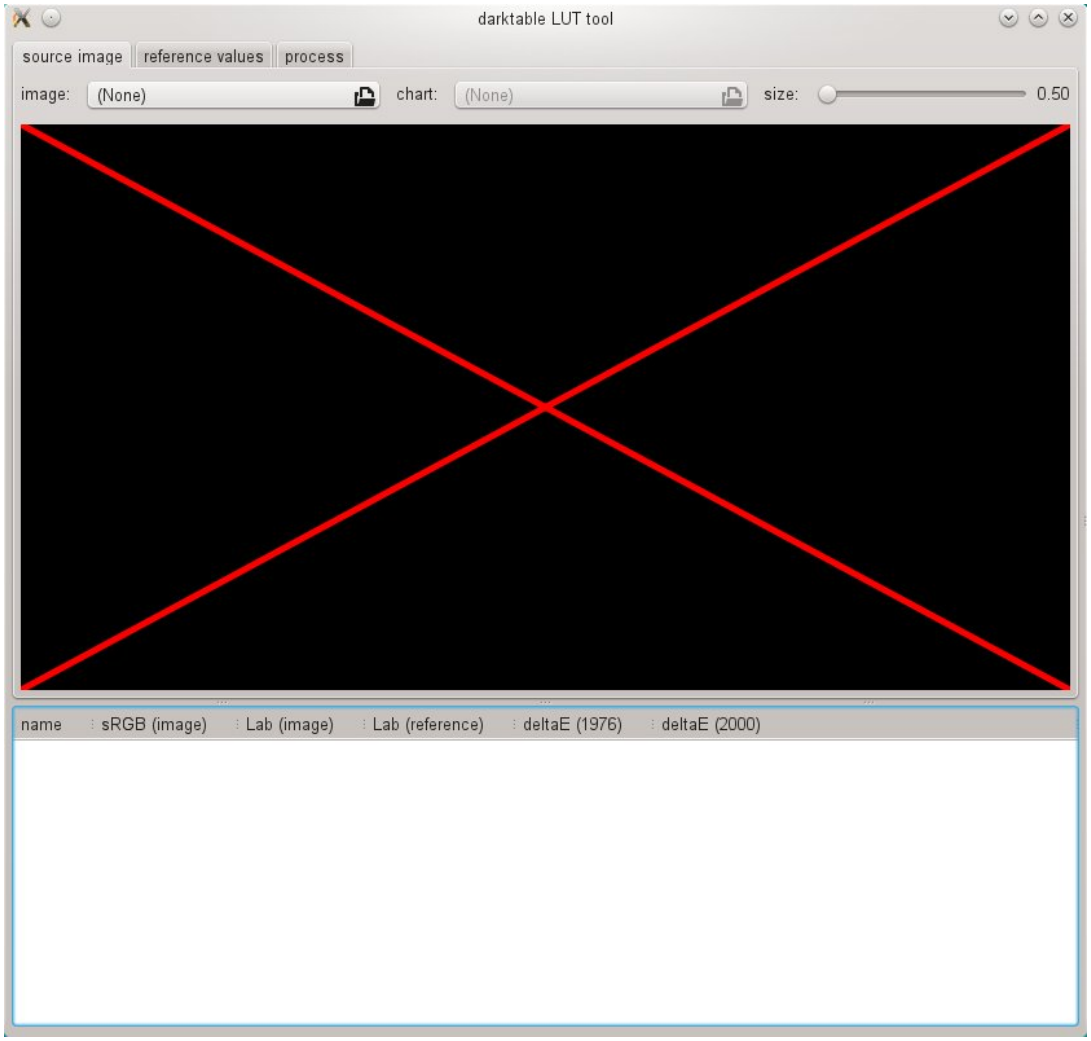
12.5.1. overview

`darktable-chart` is a tool for extracting luminance and color values from images of color reference cards such as IT8.7/1 charts. Its main purpose is to compare a source image (typically a largely unprocessed raw image) to a target image (typically a JPEG image created in-camera) and produce a darktable style that is able to use the luminance and color values of the source image to produce the target image. This style employs the *tone curve* module, the *input color profile* module, and the *color look up table* module for that purpose.

Some cameras offer various film simulation modes of your choice. With the help of `darktable-chart` and the underlying modules you can create styles that replicate these film simulations from within darktable.

12.5.2. usage

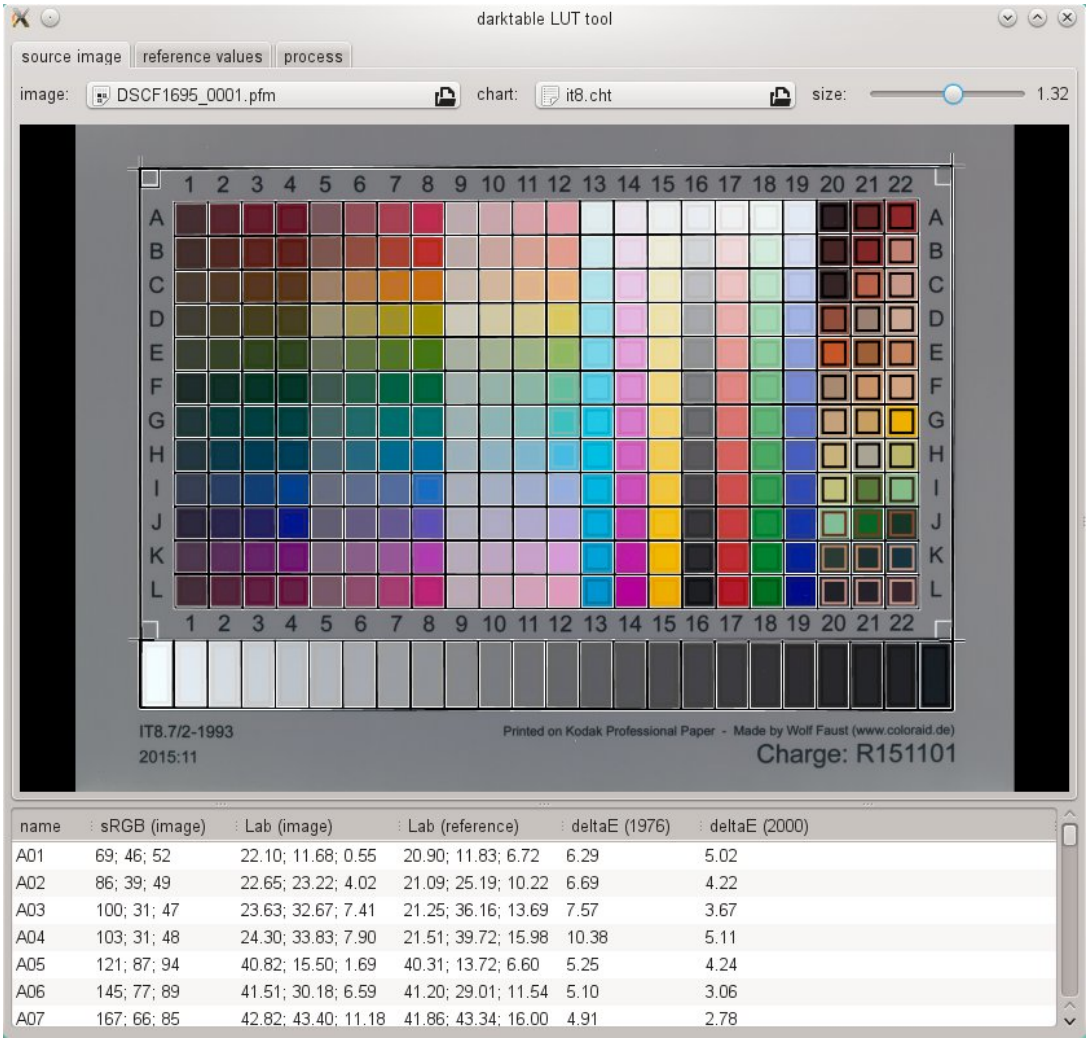
The tool is organized into three tabs in the upper part and a text output frame in the lower part.



The first tab is used to define the source image, the second tab defines the reference (target) and the third tab contains the controls to generate the resulting darktable style.

12.5.3. source image

In the “source image” tab you set your source image, which requires two elements. The first element is an input file in Lab Portable Float Map format (extension `.pfm`). The source file represents the largely unmodified data as the camera sees it. Information about how to take photos of a color reference card and produce a `.pfm` output file are described below. The second element is a chart file that contains a formal description of the underlying color reference card’s layout (extension `.cht`). Chart files are usually shipped with your color reference card or can be downloaded from the internet.



name	sRGB (image)	Lab (image)	Lab (reference)	deltaE (1976)	deltaE (2000)
A01	69; 46; 52	22.10; 11.68; 0.55	20.90; 11.83; 6.72	6.29	5.02
A02	86; 39; 49	22.65; 23.22; 4.02	21.09; 25.19; 10.22	6.69	4.22
A03	100; 31; 47	23.63; 32.67; 7.41	21.25; 36.16; 13.69	7.57	3.67
A04	103; 31; 48	24.30; 33.83; 7.90	21.51; 39.72; 15.98	10.38	5.11
A05	121; 87; 94	40.82; 15.50; 1.69	40.31; 13.72; 6.60	5.25	4.24
A06	145; 77; 89	41.51; 30.18; 6.59	41.20; 29.01; 11.54	5.10	3.06
A07	167; 66; 85	42.82; 43.40; 11.18	41.86; 43.34; 16.00	4.91	2.78

In real life the photo taken from the color reference card will show some perspective distortions relative to the layout defined in the chart file. For that reason the layout is displayed as a grid over the image and can be modified.

You can move the corners of the grid using the mouse to reach best alignment of grid and image.

A rectangular frame is displayed for each patch and defines the area from which `darktable-chart` will sample the required input data. You may need to modify the size of these rectangles so that the sampling area is big enough but does not overlap with neighboring patches. Use the “size” slider in the upper right part of the GUI. Higher values lead to smaller sizes.

12.5.4. reference values

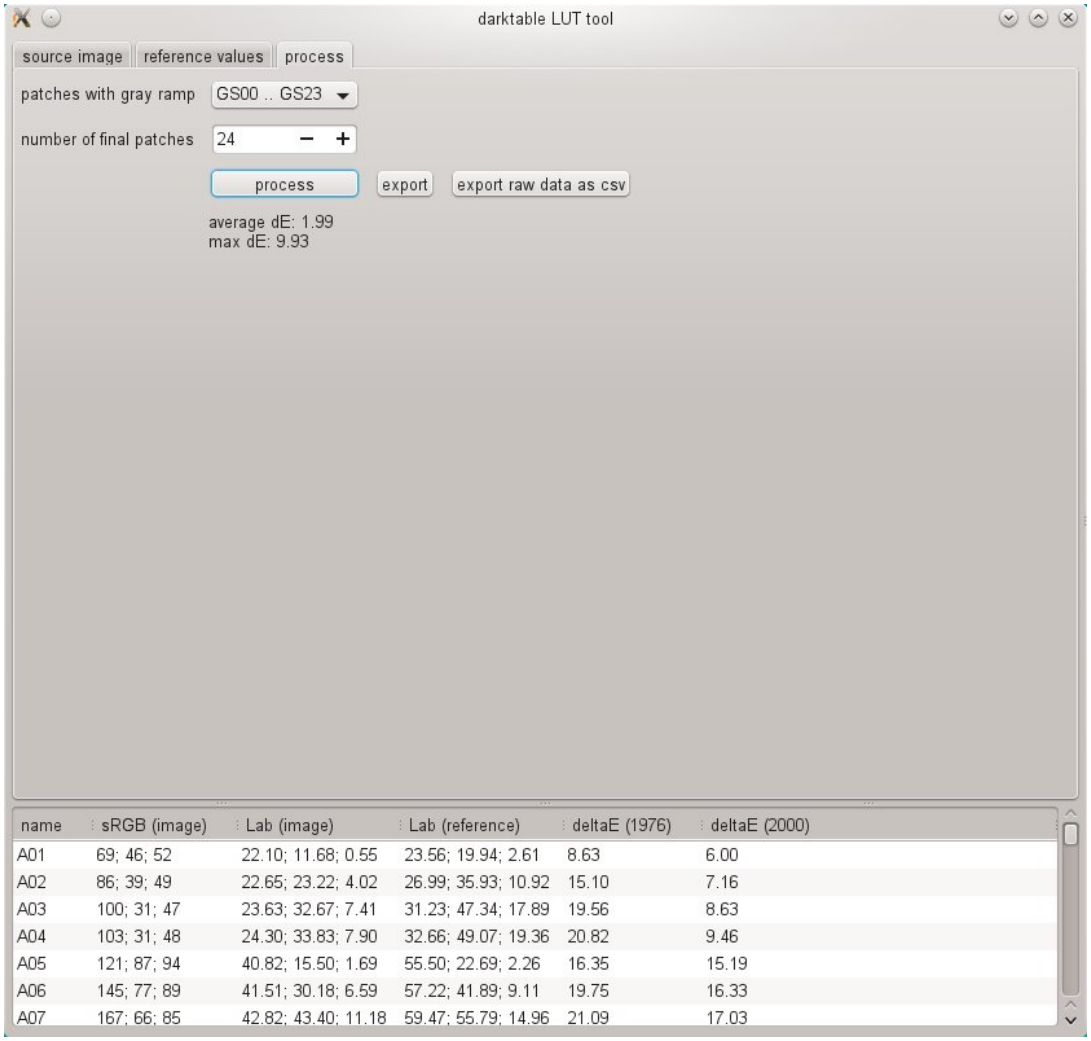
The “reference values” tab determines the target values to which the source image must to be modified by the resulting style. You can either supply reference values in the form of measured data of your color reference card (mode “cie/it8 file”), or you can supply a photographic image (mode “color chart image”) much in the same way as described above. This second image must also be supplied in Lab Portable Float Map format. There is no need to supply the chart file again as `darktable-chart` takes the same one as defined under “source image”. You only need to again align the layout grid and the image and potentially adjust the “size” slider.

In a typical use case the second image will be based on a JPEG file produced in-camera. This way you can create a style to simulate the in-camera processing within darktable.

In the lower text output frame you will see the color values extracted from the available data for each individual color patch. The first column gives the name of the patch, the second and third column show the corresponding color values of the source image in RGB and Lab format, respectively. The fourth column contains the Lab value coming from the reference (or from the chart file if no reference image has been given). Finally, the fifth and sixth columns display how strongly source and reference values deviate in terms of delta-E values.

12.5.5. process

If all of the required settings in the “source image” and “reference values” tabs are ready you can move on to the “process” tab.



First, you need to tell `darktable-chart` which of the patches represents the gray ramp. In the screenshot displayed above, the gray ramp is positioned in the lower part of the color reference chart, denoted as “GS00...GS23”.

The “number of final patches” input defines how many editable color patches the resulting style will use within the *color look up table* module.

Click the “process” button to start the calculation.

The quality of the result (in terms of average delta-E and maximum delta-E) is displayed below the button. This data shows how closely the resulting style (when applied to the source image) will be able to match the reference values – the lower the better.

Once you are happy with the result you can click on “export” to save the generated style.

Supply a style name and a style description under which the style will later appear in darktable. `darktable-chart` saves the style as a `.dtstyle` file which can be imported into darktable and shared with others. See styles .

The “export raw data as csv” button allows you to save the extracted raw data as a CSV file for debugging purposes or later use. `darktable-chart` offers a command line option to produce a style with the desired number of final patches from a supplied CSV file (see `darktable-chart`).

12.5.6. making input images for darktable-chart

To start with, you need a suitable photo of your color reference card in RAW+JPEG format. It goes beyond the scope of this manual to explain the details of how to take this photo, but in a nutshell you need to make the shot on a sunny day around midday with the light source (the sun) shining at an angle onto the card. You need to avoid any glare in the image. The neutral white color patch in the gray ramp (G00) should end up at the L value specified in the description of your card. Often this is L=92 and requires you to overexpose the shot by about 1/3 EV. Ideally you will make several shots with slightly different exposures and later select the right one in darktable. Make sure that the chart fills up most of the frame. Use a lens with a “normal” focal length (e.g. 50mm equivalent) and stop down a bit to avoid vignetting.

You then open the raw file in darktable and disable most modules, especially *base curve* . Select the standard input matrix in the *input color profile module* and disable gamut clipping. Select “camera white balance” in the *white balance* module.

There is a special situation if your camera automatically applies some lens corrections (namely vignetting correction) to the resulting JPEG file. In this case you need to activate the *lens correction* module in darktable so that raw processing matches the JPEG in this respect. However, since darktable’s vignetting correction may not exactly match the in-camera correction, it is better to disable this correction in-camera if possible.

To output your image go to the export module in the lighttable.

You will need to select “Lab” as the output color profile. This color space is not visible in the combobox by default. You first need to enable it by setting `allow_lab_output` to TRUE in darktable’s configuration file `darktablerc` in the configuration directory . Alternatively, you can start darktable with:

```
darktable --conf allow_lab_output=true
```

Then select “PFM (float)” as the output format and press “export” to generate the source image file.

You can produce the corresponding reference (target) image from the JPEG in a similar way. This time you will need to disable all modules and export with the “Lab” output color profile in “PFM (float)” format.

12.6. program invocation

12.6.1. invocation per OS

`darktable` and the other binaries (e.g. `darktable-cltest`) can be executed from the console as follows:

Linux

- Installed as package or snap: `darktable`
- Flatpak: `flatpak run org.darktable.Darktable`
  - To run e.g. `darktable-cltest`: `flatpak run --command=darktable-cltest org.darktable.Darktable -- arguments-go-here`.
- AppImage: Navigate to containing folder, then `./darktable.Appimage` (or however you named the file)
  - To run e.g. `darktable-cltest`: `./darktable.Appimage darktable-cltest`
  - An additional option to run the various binaries included in darktable (e.g. `darktable-cltest` or `darktable-cli`) is through a symbolic link:
    - Create a symbolic link to your AppImage and call it e.g. `darktable-cli`.
    - Then, when calling the AppImage through the symbolic link with `./darktable-cli`, the code inside the package checks if there is a binary file inside the AppImage with a name that matches the symlink name. If so, it will be run instead of the default one.

Windows

- Navigate to darktable’s program folder: `cd "\Program Files\darktable\bin"` then execute `.\darktable.exe`.
- The other binaries e.g. `darktable-cltest.exe` are also located in this folder and can be called accordingly.

macOS

- Execute `/Applications/darktable.app/Contents/MacOS/darktable`.
- The other binaries e.g. `darktable-cltest` are also located in this folder and can be called accordingly.

Command line options are in any case simply appended. (e.g. `-d common`).

12.6.2. darktable

The `darktable` binary starts darktable with its GUI and full functionality. This is the standard way to use darktable. The command to run the binary depends on your OS .

The following command line parameters are available:

```
darktable [-d {all,act_on,ai,cache,camctl,camsupport,common,control,
               dev,expose,fswatch,imageio,input,ioporder,lighttable,lua,
               masks,memory,nan,opencl,params,perf,pipe,print,pwstorage,
               signal,sql,tiling,undo,verbose}]
          [--d-signal <signal>]
          [--d-signal-act <all,raise,connect,disconnect,print-trace>]
          [--disable-pipecache]
          [<input file>|<image folder>]
          [--version]
          [--disable-opencl]
          [--configdir <user config directory>]
          [--library <library file>]
          [--datadir <data directory>]
          [--moduledir <module directory>]
          [--tmpdir <tmp directory>]
          [--cachedir <user cache directory>]
          [--localedir <locale directory>]
          [--print-paths]
          [--print-paths-as-flags]
          [--luacmd <lua command>]
          [--noiseprofiles <noiseprofiles json file>]
          [--conf <key>=<value>]
          [-t <num openmp threads>]
```

All parameters are optional. In most cases darktable should be started without any additional parameters, in which case darktable uses suitable defaults.

`-d {all,act_on,ai,cache,camctl,camsupport,common,control,dev,expose,fswatch,imageio,input,ioporder,lighttable,lu`

Enable debug output to the terminal. There are several subsystems of darktable and each of them can be debugged separately. You can use this option multiple times if you want to debug more than one subsystem (e.g. `darktable -d opencl -d camctl`) or debug all of them at once (with `-d all`). The `-d common` switch is provided to give information about most relevant subsystems while debugging darktable or if you want to provide a log for reporting a darktable issue. Some debug options (like `-d opencl`) can also provide more verbose output, which can be invoked with the additional option `-d verbose`. The verbose option must be explicitly provided, even when using `-d all`.

`--d-signal <signal>`

If `-d signal` or `-d all` is specified, specify the signal to debug using this option. Specify ALL to debug all signals or specify signal using its full name. Can be used multiple times.

`--d-signal-act <all,raise,connect,disconnect,print-trace>`

If `-d signal` or `-d all` is specified, specify the signal action to debug using this option.

`--disable-pipecache`

Disable the pixelpipe cache. This option allows only two cachelines per pipe, and should be used for debugging purposes only.

`<input file>|<image folder>`

Optionally supply the name of an image file or folder. If a filename is given darktable starts in darkroom view with that file opened. If a folder is given darktable starts in lighttable view with the content of that folder as the current collection.

`--version`

Print the darktable version number, a copyright notice, some other useful information, and then terminate.

--disable-opencl

Prevent darktable from initializing the OpenCL subsystem. Use this option if darktable crashes at startup due to a defective OpenCL implementation.

--configdir <config directory>

Define the directory where darktable stores user-specific configuration. The default paths are listed on the configuration directory page.

--library <library file>

darktable keeps image information in an sqlite database for fast access. The default location of that database file is file name `library.db` in the directory specified by `--configdir` or the default configuration directory . Use this option to provide an alternative location (e.g. if you want to do some experiments without compromising your original `library.db`). If the database file does not exist, darktable creates it for you. You may also provide `:memory:` as the library file, in which case the database is kept in system memory – all changes are discarded when darktable terminates.

Whenever darktable starts, it will lock the library to the current user. It does this by writing the current process identifier (PID) into a lock file `<library file>.lock` next to the library specified. If darktable finds an existing lock file for the library, it will terminate immediately.

--datadir <data directory>

Define the directory where darktable finds its runtime data. The default location depends on your installation. Typical locations are `/opt/darktable/share/darktable/` and `/usr/share/darktable/`.

--moduledir <module directory>

darktable has a modular structure and organizes its modules as shared libraries for loading at runtime. This option tells darktable where to look for its shared libraries. The default location depends on your installation. Typical locations are `/opt/darktable/lib64/darktable/` and `/usr/lib64/darktable/`.

--tmpdir <tmp directory>

Define where darktable should store its temporary files. If this option is not supplied darktable uses the system default.

--cachedir <cache directory>

darktable keeps a cache of image thumbnails for fast image preview and precompiled OpenCL binaries for fast startup. Multiple thumbnail caches may exist in parallel – one for each library file. The default cache path is listed on the configuration directory page.

--localedir <locale directory>

Define where darktable can find its language-specific text strings. The default locale path is listed on the configuration directory page.

--print-paths

Print the resolved configdir, cachedir, tmpdir, datadir, moduledir, localedir and library paths for this install, honoring any of the overrides, then exit.

--print-paths-as-flags

Like `--print-paths`, but printed as a single line of –flag value pairs ready to insert into another invocation's command line.

--luacmd <lua command>

A string containing lua commands to execute after lua initialization. These commands will be run after your "luarc" file.

If lua is not compiled-in, this option will be accepted but won't do anything.

--noiseprofiles <noiseprofiles json file>

Provide a json file that contains camera-specific noise profiles. The default location depends on your installation. Typical locations are `/opt/darktable/share/darktable/noiseprofile.json` and `/usr/share/darktable/noiseprofile.json`.

--conf <key>=<value>

darktable supports a rich set of configuration parameters defined by the user in file `darktablerc`, located in the directory specified by `--configdir` or the default configuration directory . You may temporarily overwrite individual settings on the command line with this option – these settings will not be stored in `darktablerc` on exit.

-t <num openmp threads>

limit number of openmp threads to use in openmp parallel sections.

--dump-pfm MODULE_A,MODULE_B, --dump-pipe MODULE_A,MODULE_B, --dumpdir DIR

These options are provided for debugging darktable's internal processing pipeline. For example, if called with `--dump-pfm demosaic` darktable will dump the input and output of the demosaic module as pfm files. By default the location of these files is defined by the operating system – some temporary folder reported in the log output – but you can also explicitly define it via the `--dumpdir` option.

12.6.3. darktable-cli

The `darktable-cli` binary starts the command line interface variant of darktable which allows images to be exported.

This variant does not open any display – it works in pure console mode without launching a GUI. This mode is particularly useful for servers running background jobs.

`darktable-cli` can be called with the following command line parameters:

```
darktable-cli [<input file or folder>]
              [<xmp file>]
              <output file or folder>
              [--width <max width>]
              [--height <max height>]
              [--hq <0|1|true|false>]
              [--upscale <0|1|true|false>]
              [--style <style name>]
              [--style-overwrite]
              [--apply-custom-presets <0|1|false|true>]
              [--out-ext <extension>]
              [--import <file or dir>]
              [--library <path>]
              [--icc-type <type>]
              [--icc-file <file>]
              [--icc-intent <intent>]
              [--pq-peak <nits>]
              [--print-paths]
              [--print-paths-as-flags]
              [--verbose]
              [--help [option]]
              [--core <darktable options>]
```

The user must supply an input filename and an output filename. All other parameters are optional.

`<input file or folder>`
: The name of the input file or folder (containing images) to be exported. If you wish to process multiple images or multiple folders use the `--import` option instead.

`<xmp file>`
: The optional name of an XMP sidecar file containing the history stack data to be applied during export. If this option is not provided darktable will search for an XMP file that belongs to the given input file(s).

`<output file or folder>`
: The name of the output file or destination folder. The parameter is mandatory if you use the program on an image folder containing multiple images, and it must point to an existing folder. If the specified path does not point to an existing folder, it is interpreted as a filename. If the path is interpreted as a filename, the export file format is derived from the file extension, unless the `--out-ext` option is present (see below). Note that the file extension is only used to determine the **export format**, which then enforces the actual output extension (e.g. `image.jpeg` sets the format to JPEG, which results in the filename `image.jpg`). You can also use a number of variables in the output filename. If you specify output folder it is recommended that you also specify the file format with `--out-ext`.

`--width <max width>`
: Limit the width of the exported image to the given number of pixels.

`--height <max height>`
: Limit the height of the exported image to the given number of pixels.

`--hq <0|1|true|false>`
: Define whether to use high quality resampling during export (see the export module reference for more details). Defaults to true.

`--upscale <0|1|true|false>`
: Define whether allow upscaling during export. Defaults to false.

`--style <style name>`
: Specify the name of a style to be applied during export. If a style is specified, the path to the darktable configuration directory must also be specified (i.e. `--core --configdir ~/.config/darktable`). By default no style is applied.

--style-overwrite
The specified style overwrites the history stack instead of being appended to it.

--apply-custom-presets <0|1|false|true>
Whether to load `data.db` which contains presets and styles. Disabling this option allows you to run multiple instances of `darktable-cli` at the cost of being unable to use the `--style` option. Defaults to true.

--out-ext <extension>
Defines the export format to use (e.g. `jpeg`, `tiff`, `openexr`) based on the provided argument string, which can be a common extension (e.g. `jpg`, `tif`, `exr`) or a format (e.g. `jpeg`, `tiff`). The name `--out-ext` is preserved for compatibility with existing scripts, but it is actually used to determine the **format**, not the file extension; the extension, as always, is determined by the format. If specified, this format takes precedence over the one derived from `<output file>`. If the output filename already contains an extension, it is removed **only if** it exactly matches the string passed to `--out-ext`. Otherwise, it is treated as part of the filename (e.g. `output.jpg` with `--out-ext jpeg` results in `output.jpg.jpg`, because `jpg` does not match `jpeg`). Defaults to `jpg` if `<output folder>` is specified.

--import <file or dir>
Specify input file or folder, can be used multiple times. This option cannot be combined with `<input file or folder>`.

--library <path>
Read the image processing history stack from the specified library database instead of XMP sidecar files.

--icc-type <type>
Specify the ICC profile type, which is the same as specifying the “output profile” in the *output color profile* module. Defaults to “image specified”. Use `--help icc-type` to obtain a list of the supported types. See the *output color profile* module reference for a more detailed description of the available options.

--icc-file <file>
Specify the ICC profile filename. Defaults to an empty filename.

--icc-intent <intent>
Specify the rendering intent. Defaults to “image specified”. Use `--help icc-intent` to obtain a list of the supported intents. See rendering intent for a more detailed description of the available options.

--pq-peak <nits>
Set the peak luminance of PQ (HDR) output, from 203 to 10000 nits. Tone mappers that follow the output, such as *AgX* and *sigmoid*, map their white to this peak. Defaults to the *PQ peak* setting of the *export* module.

--print-paths
Print the resolved configdir, cachedir, tmpdir, datadir, moduledir, localedir and library paths for this install, honoring any of the overrides, then exit.

--print-paths-as-flags
Like `--print-paths`, but printed as a single line of –flag value pairs ready to insert into another invocation’s command line (for `darktable-cli` they need to be *after* the `--core` flag).

--verbose
Enables verbose output.

--help [option]
Prints usage and exits. If `option` is specified, additionally prints usage for the given option.

--core <darktable options>
All command line parameters following `--core` are passed to the darktable core and handled as standard parameters. See the `darktable binary` section for a detailed description.

export options

Export options for darktable are defined as configuration items, set from within the export module . There are two ways to alter this configuration when using `darktable-cli`, as described below.

use the export module

The `darktable-cli` command will use the last format configuration used in the export module, when run in interactive (gui) mode. You may therefore manually set your desired format options in the darktable gui and then run `darktable-cli` to export your files.

pass options on the command-line

You can set any export format configuration option using the following syntax:

```
--core --conf plugins/imageio/format/<FORMAT>/<OPTION>=<VALUE>
```

where <FORMAT> is the name of the desired output format and <OPTION> is any configuration option for that format.

An option set in this way will not be permanently stored but will be used just for this run of `darktable-cli`.

The following sections describe the configuration options/values that are available for each export format:

jpeg

quality
The compression quality (5 - 100)

j2k (jpg2000)

format
The format of the output
- 0: J2K
- 1: jp2

quality
The compression quality (5 - 100)

preset
The DCP mode
- 0: Cinema2K, 24 FPS
- 1: Cinema2K, 48 FPS
- 2: Cinema4K, 24 FPS

exr (OpenEXR)

bpp
The bit depth (16 or 32)

compression
The compression type
- 0: uncompressed
- 1: RLE
- 2: ZIPS
- 3: ZIP
- 4: PIZ
- 5: PXR24
- 6: B44
- 7: DWAA
- 8: DWAB

pdf

title
The title of the pdf (any character)

size
The size of the pdf (a4, a3, `letter`, `legal`)

orientation
the paper orientation of the pdf
- 0: portrait
- 1: landscape

border
The empty space around the pdf; format: size (a number) + unit; examples: 10 mm, 1 inch

dpi
  The resolution in dots per inch inside the pdf (1 - 5000)
rotate
  Whether to rotate the pdf (0 or 1)
icc
  Whether to embed an icc profile (0 or 1)
bpp
  The bit depth (8 or 16)
compression
  Whether to compress the pdf (0 or 1)
mode
  The mode to put the images in the pdf
    - 0: normal: just put the images into the pdf
    - 1: draft: images are replaced with boxes
    - 2: debug: only show the outlines and bounding boxen

pfm

No options provided.

png

bpp
  The bit depth (8 or 16)
compression
  The compression level (0 - 9)

ppm

No options provided.

tiff

bpp
  The bit depth (8, 16, 32)
compress
  The compression type
    - 0: uncompressed
    - 1: deflate
    - 2: deflate with predictor
compresslevel
  The compression level (0 - 9)
shortfile
  B&W or color image
    - 0: write rgb colors
    - 1: write grayscale

webp

comp_type
  The compression type
    - 0: lossy
    - 1: lossless
quality
  the compression quality (5 - 100)

hint
The preferred way to manage the compression
- 0: default
- 1: picture: digital picture, like portrait, inner shot
- 2: photo: outdoor photograph, with natural lighting
- 3: graphic: discrete tone image (graph, map-tile, etc.)

copy

No options provided.

xcf

bpp
The bit depth (8, 16, 32)

JXL

bpp
The bit depth (8, 10, 12, 16, 32)
pixel_type
Boolean whether the (16 bit) pixel type is unsigned integer or floating point
- 0: unsigned integer
- 1: floating point

quality
Integer (4-100): the quality of the image, roughly corresponding to JPEG quality (100 is lossless)
original
Boolean whether to encode using the original color profile or the internal XYB one
- 0: internal
- 1: original

effort
Integer between 1-9. Effort with which to encode output; higher is slower (default is 7)
tier
Integer between 0-4. Higher value favors decoding speed vs quality (default is 0)

12.6.4. darktable-generate-cache

The `darktable-generate-cache` binary updates darktable's thumbnail cache. Invoke this program to generate all missing thumbnails in the background when your computer is idle.

`darktable-generate-cache` can be called with the following command line parameters:

```
darktable-generate-cache
              [-h, --help]
              [--version]
              [--min-mip <0-8>] [-m, --max-mip <0-8>]
              [--min-imgid <N>] [--max-imgid <N>]
              [--core <darktable options>]
```

All parameters are optional. If started without parameters `darktable-generate-cache` uses reasonable defaults.

-h, --help
Display usage information and terminate.
--version
Display copyright and version information and terminate.

--min-mip <0-8>, -m, --max-mip <0-8>
darktable can store thumbnails with up to eight different resolution steps for each image. These parameters define the maximum resolution to be generated (defaults to a range of 0-2). There is normally no need to generate all possible resolutions here – missing ones will be automatically generated by darktable the moment they are needed. When asked to generate multiple resolutions at once, the lower-resolution images are quickly downsampled from the highest-resolution image.

--min-imgid <N>, --max-imgid <N>
Specifies the range of internal image IDs from the database to work on. If no range is given, `darktable-generate-cache` will process all images.

--core <darktable options>
All command line parameters following `--core` are passed to the darktable core and handled as standard parameters. See the darktable binary section for a detailed description.

12.6.5. darktable-chart

The `darktable-chart` binary is a dedicated utility to create styles out of pairs of images such as RAW+JPEG with in-camera processing. Details about its usage can be found in the using darktable-chart section.

`darktable-chart` can either start a GUI or be used as a command-line program.

```
darktable-chart
              [--help]
              [<input Lab pfm file>]
              [<cht file>]
              [<reference cgats/it8 or Lab pfm file>]
```

All parameters are optional, however, if you want to supply the second file name you also need to supply the first one. Starting `darktable-chart` this way opens a special GUI (details can be found in the using darktable-chart section).

--help
Provide usage information and terminate.

<input Lab pfm file>
Open the utility with the given file as source image. The input file needs to be in Lab Portable Float Map format.

<cht file>
Specify a chart file describing the layout of the used color reference chart.

<reference cgats/it8 or Lab pfm file>
Specify the reference values, either as measured values according to the CGATS standard, or as a reference image in Lab Portable Float Map format.

Alternatively `darktable-chart` can be used as a command line program to generate darktable style files using previously saved CSV files.

```
darktable-chart
              --csv
              <csv file>
              <number patches>
              <output dtstyle file>
```

All parameters are mandatory.

<csv file>
A CSV file previously saved from within `darktable-chart`.

<number of patches>
The number of color patches to be used in the color look up table settings of the created style.

<output dtstyle file>
The name of the style file to be created.

12.6.6. darktable-cltest

The `darktable-cltest` binary checks if there is a usable OpenCL environment on your system that darktable can use. It emits some debug output that is equivalent to calling `darktable -d opencl` and then terminates.

`darktable-cltest` is called without command line parameters.

12.6.7. darktable-cmstest

The `darktable-cmstest` binary (Linux only) investigates whether the color management subsystem of your computer is correctly configured and displays some useful information about the installed monitor profile(s).

`darktable-cmstest` is called without command line parameters.

12.6.8. darktable-mcp

The `darktable-mcp` binary provides a Model Context Protocol (MCP) server interface for darktable.

`darktable-mcp` can be used to let an AI agent drive the raw pipeline over MCP, for example it can be used to:

- list modules and inspect their parameter schemas,
- develop an image through a given module stack and get back a preview or per-channel statistics,
- read history, styles and metadata from the library.

`darktable-mcp` is only built if darktable was compiled with the `USE_MCP` option, so the binary may not be present in all packages.

Module parameters are referred to by name, so a set of instructions written for one version of darktable keeps working when a module is updated. Images are developed on a temporary copy, so neither the original file nor its existing edits are changed.

It waits for commands on its stdin, and outputs responses to its stdout; stdout carries only JSON-RPC (darktable's own logging is redirected to stderr), so a client can speak the protocol cleanly.

darktable-mcp can be called with the following optional command line parameters:

```
darktable-mcp [--core <darktable options>]
```

`--core <darktable options>`

All command line parameters following `--core` are passed to the darktable core and handled as standard parameters. See the `darktable binary` section for a detailed description.

Unless you override them, two defaults are injected:

- `--library :memory:`
  - the database is kept in system memory - all changes are discarded when the application terminates. Images are rendered and inspected by file path, and the library tools see nothing else.
- `--conf write_sidecar_files=never`
  - never update XMP sidecars.

To work against a real catalog instead, so that existing images, their edits and saved styles are visible, pass `--core --library /path/to/library.db`. darktable locks `library.db` and `data.db` while it runs, so this only works when the darktable GUI is not open on that library – or against a copy of it. The default in-memory database takes no such lock and can be used while darktable is running.

available tools

The server exposes three groups of tools to the connected client.

Introspection tools describe darktable's modules: list the available processing modules, retrieve the parameters of a given module with their types, ranges and defaults, and convert between a module's stored parameter blob and named values. Each module also reports a link to its page in this manual.

Develop tools process images: develop an image through its history stack, optionally with additional modules applied on top, and return either a preview image or per-channel statistics. Rendering is performed on a throwaway duplicate, so the source image is never modified.

Library tools read and modify the catalog: list images, read an image's decoded history stack, list, apply, save and import styles, and export a developed image to a file. Note that several of these write to the library – applying or saving a style changes the catalog, just as it would in the GUI.

connecting a client

`darktable-mcp` is not run directly by the user – it is launched by an MCP client, which talks to it over stdin and stdout. Most clients are configured with a JSON entry naming the command and its arguments, for instance like this example (which uses paths typical for a Linux system):

```
{
 "mcpServers": {
  "darktable": {
    "command": "/usr/bin/darktable-mcp",
    "args": ["--core",
          "--configdir", "/home/user/.config/darktable-mcp",
          "--cachedir",  "/home/user/.config/darktable-mcp/cache"]
  }
 }
}
```

Give the server its own configuration and cache directories, as above. With the default in-memory library it can then be used at any time, including while darktable itself is open.

Consult the documentation of your MCP client for where this configuration is stored, and whether it provides a command to add a server rather than editing the file by hand.

12.6.9. purge_non_existing_images.sh

Find and remove entries from the library database referencing images that no longer exist in the filesystem. You must close darktable before running this script.

The script can be called with the following command line parameters:

```
purge_non_existing_images.sh [-c|--configdir <path>]
                             [-l|--library <path>]
                             [-p|--purge]
```

Run the script with no options to perform a "dry run", which generates a report of the missing files without committing any changes to the database.

The available options are:

-c|--configdir <path>
: Specify the path to the darktable configuration directory to be used by the script. If this option is not provided, the default configuration directory location will be used.

-l|--library <path>
: Specify the path to the library.db database file to be analysed by the script. If this option is not specified, the default library.db file location will be used.

-p|--purge
: Actually delete any entries in the database that refer to non-existent files. If the option is not provided, a report will be printed without committing any changes to the database.

Notes:

1. The script must be run in a unix shell, and the `sqlite3` client must be available in the command search path. For Linux systems, this will normally not be an issue.
2. For Windows systems, you will normally need the MSYS2 environment to be installed, as described in the instructions for building darktable in a Windows environment . If you installed darktable using the standard Windows installer package, the location of the script would normally be something like: `C:\Program Files\darktable\share\darktable\tools\purge_non_existing_images.sh`.
3. For macOS systems, the Terminal application provides a shell, and the `sqlite3` client is provided by the operating system by default. If darktable was installed using an application bundle from a dmg image, then the default location for the script would be `/Applications/darktable.app/Contents/Resources/share/darktable/tools/purge_non_existing_images.sh`
4. The delete operation can't be undone. It is therefore strongly recommended that you take a backup of the database before purging any entries.

12.6.10. gimp integration

GIMP can use darktable to open a Raw file if the darktable executable is on the PATH. If the GIMP user opens a RAW file, then GIMP invokes darktable with some special command line parameters to get a temporary file for GIMP to work with, as follows:

```
darktable --gimp [version]
                 [file <path>]
                 [thumb <path> <dim>]
```

In all cases, darktable will report its result to GIMP wrapped in tags like this (where <res> depends on the command given):

```
<<<gimp
<res>
gimp>>>
```

Only one of the commands should be present on the command line.

`version`
Note that <res> is the version of the GIMP API (*not* the darktable version).

`file <path>`
Starts darktable in darkroom mode with the image from the <path> loaded. The user is prevented from switching to another darktable mode. When the user closes the darkroom window, the result is exported as an EXR file to a temporary location. The returned <res> is the full path of the exported file. If the image is already in the darktable database, then any edits previously done in darktable will be part of the image exported. Any edits during the GIMP invocation of darktable will not, however, be saved to the database.

`thumb <path> <dim>`
darktable writes a thumbnail JPEG file to a temporary location – <path> is the RAW file path, <dim> (in pixels) is is the greater of width or height (in pixels) required for the thumbnail. The original ratio of dimensions is maintained. The returned <res> has the following format:

- The full path of the exported file on the first line.
- The width and height of the original file as space-separated integers on the second line. These dimensions are informational and may not be accurate. For Raw files these represent the sensor size.

12.7. variables

darktable supports variable substitution in a number of modules and preference settings. For example:

- Defining file names in the export module
- Displaying image information in the darkroom's image information line
- Displaying image information in the lighttable's overlays and tooltips (see preferences > lighttable)
- Placing text on an image in the *watermark* processing module

available variables

The following variables are available, though they may not all be applicable in every context:

$(ROLL.NAME[n])	film roll of the input image, where n specifies the folder level counting from the right, default n=1
$(FILE.FOLDER)	folder containing the input image
$(FILE.NAME)	basename of the input image
$(FILE.EXTENSION)	extension of the input image
$(ID)	the image id
$(VERSION)	the duplicate version number
$(VERSION.IF_MULTI)	same as $(VERSION) but null string if only one version exists
$(VERSION.NAME)	version name from metadata
$(DARKTABLE.VERSION)	the version of the running darktable instance
$(DARKTABLE.NAME)	name of darktable
$(SEQUENCE[n,m])	a sequence number within an export job with n digits and starting with m parameters are optional, default is [4,1]
$(WIDTH.SENSOR)	width of RAW data in pixels before RAW crop
$(HEIGHT.SENSOR)	height of RAW data in pixels before RAW crop
$(WIDTH.RAW)	width of RAW data in pixels after RAW crop
$(HEIGHT.RAW)	height of RAW data in pixels after RAW crop
$(WIDTH.CROP)	image width in pixels at the end of the pixelpipe, but before export resize
$(HEIGHT.CROP)	image height in pixels at the end of the pixelpipe, but before export resize
$(WIDTH.EXPORT)	image width in pixels at the end of the pixelpipe and after export resize
$(HEIGHT.EXPORT)	image height in pixels at the end of the pixelpipe and after export resize
$(WIDTH.MAX)	maximum width entered in export module
$(HEIGHT.MAX)	maximum height entered in export module
$(YEAR)	year at date of import/export
$(YEAR.SHORT)	two-digit year at date of import/export
$(MONTH)	numeric (1-12) month at date of import/export
$(MONTH.LONG)	full month name at date of import/export
$(MONTH.SHORT)	abbreviated month name at date of import/export
$(DAY)	day at date of import/export
$(HOUR)	hour at time of import/export
$(MINUTE)	minute at time of import/export
$(SECOND)	second at time of import/export
$(MSEC)	millisecond at time of import/export
$(EXIF.YEAR)	Exif year
$(EXIF.YEAR.SHORT)	Exif year, two-digit version
$(EXIF.MONTH)	Exif month, numeric
$(EXIF.MONTH.LONG)	Exif month, full name
$(EXIF.MONTH.SHORT)	Exif month, abbreviated name
$(EXIF.DAY)	Exif day
$(EXIF.HOUR)	Exif hour
$(EXIF.MINUTE)	Exif minute
$(EXIF.SECOND)	Exif second
$(EXIF.MSEC)	Exif millisecond
$(EXIF.DATE.REGIONAL)	Exif date using user's preferred regional date format
$(EXIF.TIME.REGIONAL)	Exif time using user's preferred regional date format
$(EXIF.ISO)	Exif ISO value
$(EXIF.EXPOSURE)	Exif exposure
$(EXIF.EXPOSURE.BIAS)	Exif exposure bias
$(EXIF.EXPOSURE.PROGRAM)	Exif exposure program set in camera
$(EXIF.APERTURE)	Exif aperture
$(EXIF.CROP_FACTOR)	Exif crop factor
$(EXIF.FLASH)	Exif flash setting
$(EXIF.FLASH.ICON)	flash symbol if Exif information says flash was fired, empty string if not
$(EXIF.FOCAL.LENGTH)	Exif focal length
$(EXIF.FOCAL.LENGTH.EQUIV)	Exif 35 mm equivalent focal length
$(EXIF.FOCUS.DISTANCE)	Exif focus distance
$(EXIF.LENS)	Exif lens name
$(EXIF.MAKER)	Exif camera maker
$(EXIF.METERING)	Exif metering mode

```
$(EXIF.MODEL)              Exif camera model
$(EXIF.WHITEBALANCE)       Exif white balance set in camera
$(IMAGE.EXIF)              basic exposure information from Exif data (aperture, exposure, ISO)
$(IMAGE.ID)                the image id (note that this will be 0 during copy&import)
$(IMAGE.ID[n])             the image id (note that this will be 0 during copy&import), zero-padded to n digit
$(IMAGE.ID.NEXT)           the next image id to be assigned (can be used during copy&import)
$(IMAGE.ID.NEXT[n])        the next image id to be assigned (can be used during copy&import), zero-padded to
$(IMAGE.TAGS)              tags list (Xmp.dc.Subject), with any hierarchy flattened
$(IMAGE.TAGS.HIERARCHY)    tags list (Xmp.dc.Subject), preserving hierarchy
$(LONGITUDE)               longitude
$(LATITUDE)                latitude
$(ELEVATION)               elevation
$(GPS.ELEVATION)           elevation
$(GPS.LATITUDE)            latitude
$(GPS.LONGITUDE)           longitude
$(GPS.LOCATION)            latitude, longitude, and elevation (omitting any values which are not set)
$(GPS.LOCATION.ICON)       symbol to indicate that geolocation information is present, empty string if not
$(STARS)                   star rating (text only)
$(RATING.ICONS)            star rating (using star characters)
$(LABELS)                  colorlabels (color labels as text)
$(LABELS.ICONS)            colorlabels (color labels as icons)
$(MAKER)                   camera maker
$(MODEL)                   camera model
$(LENS)                    lens
$(Xmp.dc.title)            title from metadata
$(Xmp.dc.description)      description from metadata
$(Xmp.dc.creator)          creator from metadata
$(Xmp.dc.publisher)        publisher from metadata
$(Xmp.dc.rights)           rights from metadata
$(TAGS)                    tags list (Xmp.dc.Subject)
$(CATEGORY[n,category])    tag name of level n [0,9] of selected category (or tag)
$(SIDECAR_TXT)             content of the text sidecar file (if any)
$(FOLDER.PICTURES)         pictures folder
$(FOLDER.HOME)             home folder
$(FOLDER.DESKTOP)          desktop folder
$(OPENCL.ACTIVATED)        whether OpenCL is activated
$(USERNAME)                user name defined by OS
$(NL)                      newline character
$(JOBCODE)                 internal jobcode of current job
```

metadata

Fields from the metadata editor can be used for variable substitution using the full case-sensitive Xmp tag name as the variable name (e.g.`Xmp.iptc.location`).

string substitution

All of the variables support basic string substitution inspired by bash though some of the details differ.

All patterns are treated as simple string comparisons. There is no regex support.

The following string replacement functions are provided, where `var` is one of the variables listed above:

```
$(var-default)                  If var is empty, return "default"
                                It is possible to use another variable as "default", e.g.
                                $(WIDTH.CROP-$(WIDTH.RAW))

$(var+alt_value)                If var is set, return "alt_value" else return empty string

$(var:offset)                   Return var starting from offset
                                If offset is negative count from the end of the string

$(var:offset:length)            Starting from offset, return at most length characters of var
```

	If offset is negative the length is counted from the end of var If length is negative this indicates the end of the result, counted from the end of var, and not an actual length
$(var#pattern)	Remove "pattern" from the start of var
$(var%pattern)	Remove "pattern" from the end of var
$(var/pattern/replacement)	Replace the first occurrence of "pattern" in var with "replacement" If "replacement" is empty then "pattern" will be removed
$(var//pattern/replacement)	Replace all occurrences of "pattern" in var with "replacement" If "replacement" is empty then "pattern" will be removed
$(var/#pattern/replacement)	If var starts with "pattern then "pattern" is replaced with "replacement"
$(var/%pattern/replacement)	If var ends with "pattern" then "pattern" is replaced with "replacement"
$(var^)	Make the first character of var uppercase
$(var^^)	Make all characters of var uppercase
$(var,)	Make the first character of var lowercase
$(var,,)	Make all characters of var lowercase

formatting

The image information patterns support markup . For example, adding the following will provide a clear warning (large, red, bold text) when OpenCL has failed to initialise:

```
<span alpha='1%'>$(OPENCL_ACTIVATED/no/<span foreground='red' weight='heavy' size='xx-large'
alpha='100%'>OPENCL ACTIVATION FAILED</span>$(NL))</span>
```

12.8. darktable's color pipeline

Most image processing applications come from the 1990s and/or inherit a 1990s workflow. These applications processed images encoded with 8 bit unsigned integers because it was more memory- and computationally-efficient. However, due to the use of an integer format (which implies rounding errors) they had to apply a “gamma” (essentially a transfer function applying a power 1/2.2 or 1/2.4 to encode the RGB values) and increase the bit-depth in the low-lights in order to reduce rounding errors there (humans are very sensitive to low-light details). The 8 bit integer formats are also technically limited to the 0-255 range. Anything outside of this range overflows and is clipped to the nearest bound.

These workflows, using bounded RGB representations and possibly non-linear transforms to encode RGB signals, are called “display-referred”. They rely on the assumption that the image has been prepared for display at an early stage in the processing pipeline, and embed hard-coded assumptions about the RGB values of black, middle-gray and white. Most of the image-processing algorithms used in these workflows have been tuned around these assumptions. For example, the overlay blending mode expects a middle-gray encoded at 50% (or 128 in integer encoding).

Unfortunately the non-linear scaling, which is mandatory to make the integer encoding work, breaks the natural relationships between pixel values. Hue and saturation change in unpredictable ways, and value relationships between neighboring pixels are dilated or compressed such that gradients are also altered unpredictably.

Display-referred pipelines therefore break optical filters (lens blurring or deblurring), alpha compositing (which relies on optical and geometrical definitions of occlusion), colors and gradients (local relationships between chrominance and luminance of pixels). They also don’t scale well to HDR images, which led to the development of many questionable local and global tonemapping methods and the infamous 2010s “HDR look”.

Modern computers are not tied to the same computational limitations as those from the 1990s, and can work on pixels whose values are completely unbounded (from 0 up to +infinity) and encoded as real numbers (using floating point formats). These possibilities enable what we call a “scene-referred” workflow, in which pixels can retain their original radiometric relationships along almost the entire processing pipe. In scene-referred workflow, pixels are prepared for display only at the last stage of the pipeline, in the display transform. This means that the RGB values of the pixels are kept proportional to the intensity of the light emission recorded by the camera on the scene, enabling accurate alpha compositing and optical filter emulations, while also scaling to any dynamic range through the same algorithm (SDR as well as HDR).

However, scene-referred pipelines lose the convenient fixed values of white, middle-gray and black which characterised display-referred pipelines, and setting these values, according to the scene and to the conditions of shooting, now becomes the responsibility of the user. This requires a more complex user interface.

Also, because scene-referred values are supposed to be physically-meaningful, pixels cannot have zero intensity. This would mean they have no light at all, and the existence of zero light breaks many physically-accurate algorithms. In fact, white and black mean nothing with regard to the original scene, which is only a collection of luminances at varying intensities. The scene-referred workflow simply aims at remapping some arbitrary scene luminances to what will appear white or black on the output medium.

Versions of darktable prior to 2.6 had a non-linear display-referred pipeline, assuming that a non-linear transform took place early in the pipe and middle-gray was thereafter encoded as 50%. However, not all modules and filters clipped pixel values above 100%, leaving open the possibility of recovering those values later in the pipe.

The *filmic* module’s view transform, introduced in darktable 2.6, was the first step toward a scene-referred pipeline, and deferred the mandatory, non-linear, display preparation to the end of the pipe, along with the ability to set custom black, gray and white values. The *color balance* module then introduced a way to deal with a variable definition of middle-gray.

Starting in darktable 3.2, users could choose between two workflows that defined consistent default settings, modules and pipeline order for both *display-referred* and *scene-referred* processing.

In darktable 3.4, a full scene-referred masking and blending option was introduced, allowing masks to be defined for pixel values above 100% and using only unbounded blending operators.

Switching to *scene-referred* is a cognitive leap for most experienced users, who are used thinking in display-referred ways. In a display-referred workflow, it is customary to anchor the white value and let tone adjustments revolve around that point, trying to maximize brightness while avoiding clipping. In a scene-referred workflow, white and black values are fluid and adapted to the output medium. It is advised that users anchor middle-gray (which will be preserved as-is for any output medium) and let the view transform (*filmic*) dilate or contract the dynamic range around that point. Because 10 bit HDR white is 4 times as bright as 8 bit SDR white, any rigid definition of “white” becomes irrelevant. But anchoring for middle-gray is actually more convenient, since it keeps the average brightness of the image unchanged through the view transform.

Some modules (*levels*, *rgb levels*, *tone curve*, *rgb curve*) are inherently incompatible with a scene-referred workflow, because their graphical interface implicitly suggests RGB values that are bounded within the 0-100% range. While the pixel operations they perform can be used in either scene-referred or display-referred workflows because they are unbounded internally, their controlling interface does not allow pixels to be selected outside of the 0-100% range.

Similarly, blending modes such as overlay, linear light, soft light, hard light, darken, brighten, etc. all have hard-coded thresholds that internally expect display-referred non-linear encoding.

In darktable 3.4 and above, hovering the cursor over a module header shows a tooltip detailing the color spaces, ranges and encodings that the module expects, uses and produces. Here are the definitions of the terms used:

linear
Pixel values are proportional to the scene radiometric emission, in a way that enables accurate emulation of physical filters.

non-linear
Pixel values are re-scaled such that low-lights are given a larger encoding range, usually to remap the 18.45% reference middle-gray to a value between 46 and 50%.

display-referred
Pixel values are expected to lie between 0 and 100% of the display range, where 100% is understood to be the luminance of a 20% reflective white surface (the white patch of a Color Checker) and 0% is understood to be the maximum density of the output medium (saturated black ink or minimum LED panel backlighting).

scene-referred
Pixel values are expected to be greater than zero up to +infinity. The meaning of specific pixel values needs to be defined at runtime by the user. Scene-referred values don’t automatically imply a linear, radiometrically scaled, encoding.

12.9. midi device support

introduction

MIDI is a communication protocol used by many electronic musical instruments (“pianos”), digital audio studio equipment (“control surfaces”) and even dedicated “photo editing keyboards” like the Loupedeck/Loupedeck+ (but not their later products). Such devices commonly feature sets of keys/buttons and sometimes encoders (knobs/rotors) and faders (sliders). Buttons sometimes feature lights, which makes them ideal for toggling features in darktable, because the lights can reflect the current (on/off) status. Encoders and faders are ideal for use with on-screen sliders in processing modules. They can be “endless” (without a start/end point and without any markings on the knob) allowing greater than 360 degrees rotation, and they may be motorized. This is helpful when switching between images or points in the editing history since the “physical” position of the encoders/faders always corresponds to the on-screen position. You may also find a ring of LEDs around an encoder that indicates its current value.

endless encoders; absolute / relative encoding

Besides their physical features, these devices are often highly configurable in how they send data. Encoders can send their “absolute” position when they are turned, as a value in the range 0-127. Darktable will send back the current position of the on-screen slider, which a ring of LEDs would show (if present) or a motorized fader would respond to (by moving the knob). If the encoder is endless, it can also be used at different speeds and higher resolution than just the 128 steps, with either many rotations or less than a full rotation required to move the on-screen slider between extremes.

Endless encoders without ring lights might send relative movements which can be accelerated when turning quickly. Different encoding modes can be used, and the device might come with software to configure these options. It is important that all encoders use the same settings and send on the same “channel” as the keys (ideally the first: 0 or 1). By default darktable tries to figure out which channel and encoding is used by listening to the first five messages. After starting a session you should slowly turn a knob left/down so that 5 identical one-step-down messages are sent. If this fails the input system can be reset (by pressing Ctrl+Alt+Shift+i) for another try. On success, a toast message will be shown in the middle of the screen displaying the discovered encoding mode – 127 (“2s Complement”) and 63 (“Relative Offset”) are common.

usage

If your device has been successfully configured and connected, you should see messages like “G#–1 (8) not assigned” when you press a button (the first is a musical notation code for the “tone”, the second a numerical representation) or “CC1,up not assigned” when adjusting an encoder. You are now ready to assign buttons and encoders to actions. The easiest way to do this for a bunch of them in one go is to press the visual shortcut mapping button (to the left of the preferences button) to enter shortcut visual mapping mode. In this mode, when you hover the mouse over a button or slider, if you then press a key or turn an encoder on the MIDI device, it gets assigned as a shortcut to that widget. You can immediately test it (while staying in mapping mode) by moving the mouse to the middle of the screen (so you don’t accidentally assign it to a different widget) and then repeating the action. Right-click to switch off mapping mode.

You might also want to assign buttons to duplicate the Ctrl and Shift key functions. For this, you need the full shortcuts dialog (right-click on the shortcuts button or open preferences and go to the shortcuts tab). In the top (“action”) list, double-click on “global/modifiers”. Now press the MIDI button you want to assign to Shift. Double click on “global/modifiers” again, but this time after assigning a button change the *element* for the newly created shortcut in the bottom list from “shift” to “ctrl”. Now you can use your MIDI device together with your mouse to access most of the functionality in darktable. After ticking “enable fallbacks” in the shortcuts dialog, holding Ctrl while turning an encoder will slow it down by a factor 10, whereas holding Shift speeds it up.

Several devices produced by Behringer have LEDs in a circle around their encoder knobs. These use different patterns depending on what slider or combobox they are assigned to:

- If the slider goes from -1 to +1, only the middle LED will be lit at zero. Moving negative will gradually light up the left side, moving positive will light the right side.
- If the slider goes from 0-100%, you’ll see more LEDs coming on when moving right, until they are all lit at 100%.
- Other numeric values will use 1 or 2 LEDs to indicate intermediate values.
- Dropdowns will use 1 LED for the options that fit on the first turn of the dial and 2 on the second turn.

troubleshooting and configuration

If you don’t see any “not assigned” toast at the top of your screen when pressing keys or turning knobs, you’ll want to test if any other MIDI applications can see the device. Under Linux you’ll need to have Alsa installed. After plugging in the device, dmesg should show lines like these:

```
usb 1-1: new full-speed USB device number 2 using xhci_hcd
...
mc: Linux media interface: v0.10
usbcore: registered new interface driver snd-usb-audio
```

Darktable uses the simple cross-platform layer PortMidi to access the underlying OS API (Alsa, Core Midi, WinMM). If you are self-compiling, make sure you have included this library.

Starting darktable with the debug parameters `-d input` will give additional information. It should try to open up to 10 MIDI devices in the order it finds them. On the command line you might see something like this:

```
[midi_open_devices] opened midi device 'Arturia BeatStep' via 'MMSystem' as midi0
[midi_open_devices] opened midi device 'BCR2000' via 'MMSystem' as midi1
[midi_open_devices] opened midi device 'X-TOUCH MINI' via 'MMSystem' as midi2
```

Two issues can arise:

- a device you don't want to use might be opened anyway (and potentially cause inappropriate behavior, like starting a fireworks show prematurely); or
- devices might appear in a different order at the next startup (for example because they are plugged into a different usb port). Since configurations are stored with the device number only, reordering would cause an incorrect layout to be picked up.

You can fix which devices to load in a specific location and which ones to skip using the configuration parameter preferences > miscellaneous > interface > order or exclude midi devices . To skip loading the BCR2000 in the above example and to fix the other two devices into the number 0 and 2 slots, you could set this config parameter to "`BeatStep,dontuse,X-TOUCH,-BCR2000`". This would leave the BeatStep as device midi0, always leave midi1 unused and would not load the BCR2000 at all, but if any other devices are connected, they would appear as midi3, midi4 and so on. Adding "`,-`" at the end would prevent any further devices loading.

If you just specify the configuration parameter as a single minus sign "`-`", no devices will be loaded at all.

For devices that use relative encoding as mentioned above, you'd have to perform the slow-turn-left procedure on every start-up, or add the found encoding to the configuration string. For example "`Loupedeck:127`".

Some MIDI controllers have keys with a light beneath them. These can be used to toggle settings and show the current position by having the light on or off. For this to work, darktable checks periodically (a few times a second) whether, say, the position of an on-screen toggle button has been changed, and sends messages to any linked MIDI device buttons to switch their light on or off. But if an unknown device has unintentionally been connected, this could be undesirable. So by default darktable waits until a "note" message is received *from* a MIDI button before sending any "note" light on/off messages *back* for that (and any lower numbered) button. That way no more buttons are addressed than exist on the device. If you immediately want all button lights to be used (rather than having to press the highest note once for each session) you can specify the number of buttons in the "order or exclude MIDI devices" preference, for example, "`BeatStep:63:16`".

devices reported to be working

The shortcut mapping system has been most extensively tested with the Behringer devices and contains custom code to deal with their specific features. All other devices are treated as "generic midi" and may or may not work (well). If you succeed in getting a MIDI device up and running that hasn't been mentioned below, it would be greatly appreciated if you would provide feedback in order to assist others, if any special steps are required. You could do this either by submitting a documentation pull request to amend this page or by filing an issue containing the necessary information.

Akai LPD8 mk2

The device is reported as plug and play on Windows (not yet tested on Mac and Linux). Shortcuts were easily assigned following the process described above. Editor software exists for Windows and Mac to customize the device controls (not yet tested).

The device has eight absolute encoders and eight pressure pads. The pads can be assigned as buttons, but require more force to activate than a typical button as they're designed for finger drumming.

Behringer X-touch Mini / Compact

These devices should be in Standard Mode (not MC). Layers A & B are somewhat supported, however, since the device does not send a notification when switching between layers, and since lights (both under buttons and the pattern used around the rotors) are set based on which layer darktable believes is active, everything will only be updated completely after you press or turn something in the "new" layer. Default settings are assumed; if any have been changed they can be restored using the X-Touch Editor

Behringer BCR2000 / BCF2000

These machines are highly configurable so there are many settings that could complicate the interaction with darktable's MIDI module. The BC Manager tool (available for Windows and MacOS) can be used to configure them. The easiest thing to do is to reset all encoders and buttons to their simplest settings, which can be done (for the BCR2000) using this file . You can send it to the machine with BC Manager or (under Linux) with amidi. There's also a global setting called "Deadtime" that determines how long the BCR ignores arriving messages after sending out updates. This is to avoid feedback loops, but for darktable it means that it blocks the adjustments sent back immediately after each rotor move. So Deadtime needs to be set to 0.

Arturia Beatstep

Individual rotors can be configured to send absolute (0-127) values or changes (+/- 1,2,3,... in different encodings). The recommended setting is Relative #1 for all knobs with Knob Acceleration set to Slow (Off) or Medium. Then put the string `BeatStep:63:16` in preferences > miscellaneous > interface > order or exclude midi devices . This can be configured with Midi Control Center , available for Windows or MacOS.

Loupedeck / Loupedeck+

Put the string `Loupedeck:127` in "preferences > miscellaneous > interface > order or exclude midi devices".

@jenshannoschwalm has provided a layout that can be imported in the shortcuts dialog/tab. It can be found, with documentation, here https://github.com/darktable-org/darktable/pull/12829#issuecomment-1320264833

Korg nanoKONTROL2

The device should be configured first using the Korg Kontrol Editor application to be in the CC mode and every button should be set to the Note type and Momentary button behavior. To control the lights in the buttons the LED mode should be set to External. It is important to note that the Track and the Marker buttons do not have leds in them.

There is a Kontrol Editor profile available here , which can be loaded using the Windows application to directly configure all these settings to correctly work with darktable.

12.10. using a test environment

To run multiple installations of darktable on your computer, for example to try out a nightly build , you can give each installation its own configuration without affecting the others.

Three steps are needed:

1. Install the additional version of darktable without overwriting the existing installation.
2. Give it its own configuration directory.
3. Make sure it doesn't touch edits created by other installations.

Note: It is possible to affect your main installation if you make mistakes in separating your testing setup. Before starting off, make sure to backup your configuration directory and consider setting up a regular backup schedule of the configuration directory.

---

1. Install the version you want to test

Linux

If an AppImage is available, this is the easiest option. Just download it and make it executable. Alternatively, you can build darktable from source and install it with a custom `--prefix` (e.g. `--prefix /opt/darktable-nightly`).

Windows

Install it into its own separate directory.

macOS

Rename the `.app` file before moving it to `/Applications` (e.g. `darktable-nightly.app`), instead of overwriting the existing one.

2. Set up a separate configuration directory

darktable stores its settings, presets, styles, and library database in its configuration directory . The default location depends on your operating system and installation method.

To prevent an installation from writing to the same configuration directory as another installation, launch it with the option `--configdir`, followed by the path to a new folder. If the folder does not exist yet, darktable creates and populates it automatically.

You can run an experiment without making persistent changes to the library by specifying `--library :memory:`, which keeps the library database in RAM. All changes to the database are discarded on exit: `darktable --configdir /path/to/test-config --library :memory:`

Tip: You can also copy an existing configuration directory into the new folder instead of starting from scratch. This allows the other installation to inherit your presets, styles, and keyboard shortcuts as a starting point, while both installations still run fully independently from then on. If you want to be more selective, `darktablerc`, `shortcutsrc`, `data.db`, styles folder, possibly lua folder and `luarc` (if present) should get you started with the setup you are used to. If you copy the entire directory, especially when including `library.db`, take special care in step 3 to avoid affecting your existing edits.

Linux

Run the following commands in a terminal:

- AppImage: `/path-to-appimage/darktable.AppImage --configdir /path-to-test-config-dir`
- Built binary: `/prefix-path-to-binary/darktable --configdir /path-to-test-config-dir`

For convenience, you can put the whole command in a shell script or a `.desktop` file (see darktable's `.desktop` file for an example).

Windows

Either create a new shortcut for `darktable.exe`:

1. Navigate to `C:\path-to-your-install\bin\`, right-click `darktable.exe` and choose *Show more options* and then *Create shortcut*.
2. Under the shortcut's "Properties", append the `--configdir` option to the "Target" field: `"C:\path-to-your-install\bin\darktable.exe" --configdir "C:\path-to-test-config"`
3. Set "Start in" to the matching `bin` folder.

Or launch darktable from the terminal:

```
cd "C:\path-to-your-install\bin\"
.\darktable.exe --configdir "C:\path-to-test-config"
```

macOS

An `.app` file can't be launched with extra parameters by double-clicking, so launch the executable inside the app bundle directly from a terminal:

```
/Applications/darktable-nightly.app/Contents/MacOS/darktable --configdir /path/to/new/config/dir
```

Replace `darktable-nightly.app` with the filename you gave the `.app` file.

3. Protect your existing edits

Edits are stored in two places: in the library database inside the configuration directory, and, if enabled, in XMP sidecar files next to your RAW files. Step 2 gives the other installation its own library database, but XMP sidecar files are stored alongside the RAW files and are therefore shared between installations.

To prevent the other installation from modifying existing XMP files, you have two options:

a) Work with separate copies of your RAW files

Safest option. Use copies of your RAW files with the other version in a separate folder. Your original RAW files and their XMP sidecars remain untouched.

b) Turn off XMP creation in the other installation's preferences

Launch darktable with the additional `--conf write_sidecar_files=never` option. Alternatively, you can set *preferences / storage / create XMP files* to "never" in the UI. This lets you open folders with existing edits without writing changes back to their XMP files. The setting only affects this installation because it has its own configuration.

Caution: If you cloned an existing installation's configuration directory for another installation, disable XMP file creation as described above in Step 3b. Alternatively, remove (not delete) all photos from the library in the new installation. This removes them only from the library database; it does not delete the files from disk. Otherwise, you could interfere with edits made by the original installation.

Summary

For an independent installation that should not write XMP sidecar files:

Linux/macOS

Options:

```
--configdir /path-to-test-config-dir --conf write_sidecar_files=never
```

Windows

Options:

```
--configdir "C:\path-to-test-config" --conf "write_sidecar_files=never"
```

Linux/macOS

Options:

```
--configdir /path-to-test-config-dir --conf write_sidecar_files=never --library :memory:
```

Windows

Options:

```
--configdir "C:\path-to-test-config" --conf "write_sidecar_files=never" --library ":memory:"
```

For the safest setup, use copies of the RAW files rather than relying on XMP creation being disabled.

12.11. using multiple computers

Editing the same photos from multiple computers requires some setup as darktable has no dedicated function for sharing its database across computers.

The local copies feature is the included solution for "editing on the road" and could be used if your files live on a NAS or external storage.

General considerations

Multiple machines are not natively supported. darktable assumes one local database per installation; there is no built-in functionality for resolving edit-conflicts.

- Make sure the same darktable version is used everywhere; using different versions can trigger incompatible edits.
- Always make sure synchronization is up to date; never work on both machines at the same time or synchronization conflicts might arise.

darktable stores its edits in two places:

- XMP sidecar files: `image.raw.xmp`; if enabled in the preferences
- `library.db` (unless using `--library :memory:`-mode)

Presets, settings etc. are stored in further files inside the configuration directory .

Approach A: Sync RAW-files + XMP-sidecars only

The simplest approach; syncs your edits leaving your configuration alone. Does allow for mixed setups with e.g. Windows and Linux.

Setup:

- Use a file-synchronization service of your choice (e.g. Syncthing, Dropbox) to sync the RAW-Files + XMP files between your devices.
- Have darktable read the XMP files upon startup: In the preferences set *create XMP files* to “on import” and enable *look for updated XMP files on startup*.

Caveats:

- Only updated XMP files will be read by darktable: New files added on one computer must be imported manually on the other. The same goes for deleted files.
- Settings, presets, styles etc. will not be synced and will have to be manually transferred.
- Take care to ensure synchronization is up to date before opening darktable. There is no built in function to resolve synchronization conflicts.

Can be combined with the `--library :memory:` mode to omit the usage of a local `library.db`; darktable will then always only work off the XMP files.

Approach B: Sync files and entire configuration directory

More complicated to setup, more error prone and requires some special considerations.

Setup:

- Use a file-synchronization service of your choice (e.g. Syncthing) to sync the RAW-Files + XMP files AND the contents of your configuration directory between your devices.
    - Note: Services like Dropbox might not work as well because they mostly don’t allow for syncing multiple folders in different locations.
- Since the `library.db` in the configuration directory references the RAW-files by their *absolute* path the location of the files has to be exactly matched between different machines.
- Somewhat more error prone than Approach A as synchronization conflicts could mess with your configuration.

Variation: Sync the files and a separate `library.db`, specify the location of the `library.db` in your launcher, not automatically syncing the rest of the configuration folder.

Caveats:

- Not cross platform compatible.
- Machine specific settings could get overridden, for instance OpenCL settings.
- Consider excluding the cache directory from sync. The darktable cache (thumbnails etc.) is large; syncing it just costs bandwidth and storage for no benefit.
- It is recommended to have a robust versioned backup strategy *including the configuration directory* to avoid breaking/losing your setup.

13. Contributing to the Docs

13.1. overview

This page defines the style guide for dtdocs and information about how to contribute to the project.

It is included in the user manual so that you can see how the page is rendered as well as how it is written. Please go to GitHub to see the source for this page.

The manual structure and content have been carefully considered based on the following criteria:

1. The manual should be comprehensive – it should describe all of the functionality available in darktable
2. It should have a consistent and logical structure and every piece of functionality should have its own logical place within that structure
3. It should be as long as necessary but as short as possible – brevity is a must
4. It should be objective
5. Functionality should be explained once and only once (with the exception of the basic workflow guidelines in the overview section)
6. Images should be included only where necessary to improve understanding of key principles and should not contain text unless it is unavoidable

We are generally **not** interested in:

1. Restructuring the manual
2. Switching markup languages
3. Detailed workflow tutorials (though we are interested in publishing those on the blogs of either darktable.org or pixls.us)

We **are** interested in

1. Spelling and grammar corrections
2. Clarification of text
3. Documentation for new features

We are always extremely interested in hearing about which sections of the manual did not make sense to you and *why*, so that we can improve the documentation.

In general, if you wish to make a major change, please open an issue and discuss it with the maintainers first. This is to avoid doing work that wouldn't be accepted.

modality

The preferred way is to create a pull request. See the workflow for a detailed step-by-step guide.

If you do not know what that is or don't feel comfortable doing it, feel free to suggest minor improvements to the docs by opening a new issue on github.

13.2. development & deployment

darktable's documentation is maintained in the dtdocs repository . Changes are introduced through pull requests (see workflow). Translations are maintained through weblate; see translations .

Building

The full set of outputs (HTML site, EPUB, and per-language PDFs) is built by `./tools/build-all.sh <base_url>`

This generates the translated content from the PO files, then builds the HTML site, the EPUB, and one PDF per enabled language. Outputs are written under `public/`, with PDFs at `public/<lang>/darktable_user_manual.pdf`.

Requirements: Hugo (extended), `weasyprint`, and po4a for the translation step.

GitHub Pages deployment (for development)

For development purposes, the docs are deployed to GitHub Pages on every push: A GitHub Action runs `tools/build-all.sh`, which generates HTML, EPUB and PDF outputs. These are committed to the repository's gh-pages branch and served via GitHub Pages at https://darktable-org.github.io/dtdocs/ . Generation of translated pages is disabled in the GitHub pages deployment.

Official deployment to darktable.org

The official deployment of the manual is on the pixls.us server at docs.darktable.org/usermanual/stable/ for the current release and docs.darktable.org/usermanual/development/ for darktable's current development snapshot.

The repositories are pulled regularly and the documentation is built with Hugo, including translations as well as EPUB and PDF outputs. The build log can be found here .

Versioning and archiving of the manual

With each major release of darktable a release-x.x branch is created, from which a versioned snapshot of the manual is maintained and deployed as described above.

Older versions of the docs are taken offline from time to time and are archived in EPUB and PDF format as releases on GitHub:

- https://github.com/darktable-org/dtdocs/releases/tag/v4.8
- https://github.com/darktable-org/dtdocs/releases/tag/v4.6
- https://github.com/darktable-org/dtdocs/releases/tag/v4.2
- https://github.com/darktable-org/dtdocs/releases/tag/v4.0
- https://github.com/darktable-org/dtdocs/releases/tag/v3.8
- https://github.com/darktable-org/dtdocs/releases/tag/v3.6

Between 4.8 and 5.6 no snapshots are available because versioning of the manual had been suspended in that period.

13.3. workflow

These instructions assume that you already have a github account and have installed **Git** as well. Make sure that git is in your PATH by executing the command `git` in any terminal or cmd and checking that it doesn't output an error such as "command not found". If you don't have git installed, follow these instructions .

Documenting undocumented features

We are particularly interested in documenting closed pull requests that require additions or changes to this manual. Pull requests that require documentation are cataloged with a GitHub label `documentation: pending`.

The high-level steps to do this are:

1. View the list of Pull Requests that need documentation .
2. Select a Pull Request where you understand what has changed.
3. Author the change as detailed in the workflow below.
4. When you make the Pull Request, include a link to the pull request from Step 1.

The documentation workflow

One-time setup

If you have not already configured git locally, use these two commands to configure your identity before committing:

1. `git config --global user.email "you@example.com"`
2. `git config --global user.name "Your Name"`

And if you have not yet authenticated your local git instance with github, follow the instructions for **GitHub CLI** here . This will ensure that you have your keys and credentials set up correctly.

Then you can clone a copy of dtdocs to your computer.

1. Go to `https://github.com/darktable-org/dtdocs`.
2. Press the `Fork` button and then the `Create fork` button.
3. You will be redirected to a page with your fork of the darktable documentation. Press the `Code` button and copy the link under **HTTPS**.
4. Open a terminal or cmd in your desired directory where the darktable documentation repository will reside. Something like 'C:\Users\YourUser\Documents' or '/home/youruser/Documents/builds'. Both on windows and macOS/Linux you can use the following command to navigate to your desired directory: `cd C:\Users\YourUser\Documents\` replacing the example directory for yours.
5. Clone the dtdocs repository by executing: `git clone https://github.com/<your username>/dtdocs.git` Change the URL to the one you copied in step 3.
6. Now you have a copy of the dtdocs repository on your PC. Some further setup will follow before you can start editing files.

Syncing changes

1. Go to your repositories on github and select your darktable fork.
2. Press the `Sync fork`.
3. Open a terminal or cmd in your darktable repository directory.
4. Ensure you have no local pending changes by executing `git status` and then execute `git checkout master` (if you followed this guide for the first time you will already be on master).
5. Execute `git pull` (this ensures your local copy is up to date with your fork on github).

Alternatively you can sync your fork and local copy via terminal:

1. Ensure you have no local pending changes by executing `git status` and then execute `git checkout master`.
2. Check your remote repositories: `git remote -v`. You should at least see `origin` (your fork). If `upstream` is missing, add it in the next step.
3. Add the dtdocs repository as upstream (only once): `git remote add upstream https://github.com/darktable-org/dtdocs.git`.
4. Pull the current state from the main repository to your local copy: `git pull upstream master`.
5. Push your now synced local copy to your fork: `git push origin master`.

Branch, edit and create a pull request

1. Open a terminal or cmd in your darktable documentation repository directory.
2. Execute `git pull origin master` to get the latest changes from GitHub.
3. Execute `git checkout -b branch-name` change `branch-name` for your desired branch name. This is often related to the work you're doing. For example, for this PR https://github.com/darktable-org/darktable/pull/17589 you could call it `dt-synch-edit-docs`.
4. Do the documentation work.
5. Verify your changed files with `git status`. If you have more files modified than the ones you intend to submit, you can do `git restore path/to/file/` to revert your changes.
6. Execute `git add -A`. This will stage all changed files.
7. Execute `git commit -m "Added documentation for yada yada"`. Inside the "" you will put a small message about the work you did.
8. Execute `git push --set-upstream origin branch-name` Replace branch-name with the one you picked earlier.
9. Create your pull request by following the instructions below.
10. If you need to make further corrections using the current branch, simply repeat 3-8. For the command in step 3 you can omit `-b` and in step 8 you can just do `git push`.
11. After you are done and your pull request is merged or you are finished with your work for this specific pull request, execute `git checkout master` to return to the master branch. I recommend working on only one branch/pr at a time if you don't know how git works.

Running `hugo` locally to see changes

hugo can also be executed locally to create a local live preview of the docs. This is useful to check for formatting, links etc.

Before you can preview the site with hugo you need to generate the stylesheets:

```
cd ./themes/hugo-darktable-docs-theme/assets/
yarn install
```

Instead of `yarn install` you can use `npm install`.

Execute `hugo server -D --disableFastRender` from the base directory of your local repository. When no errors are encountered, the terminal will report: `Web Server is available at http://localhost:1313/usermanual/development/ (bind address 127.0.0.1)` and the site should be available at the URL. Hugo will monitor the directory for changes and redeploy upon changing files automatically. When you have a specific page open and make changes to the corresponding .md file the browser should also reload the page after hugo has automatically redeployed.

Pull request

1. Go to https://github.com/darktable-org/dtdocs/compare
2. Press `compare across forks`. Select your repository, and then the branch you just worked on.
3. Press `Create Pull Request`

13.4. styles & conventions

The following details the writing style and documentation conventions used in this manual.

format

This website is authored in pure markdown, using some extensions. It is initially designed to work with the Hugo SSG but intended to be portable enough that it can be easily rendered with another application if required.

Files should be rendered in UTF-8 and should not include any column wrapping.

structure

The following shows the structure of an example main chapter with subsections in the dtdocs website.

```
example-chapter/
   _index.md
   section1-with-subsections/
      subsection1/
         image.png
      _index.md
      subsection1.md
      subsection2.md
   section2.md
   section3.md
```

A couple of notes on the above structure:

- `_index.md` files do not contain any content (they contain metadata only) and are used to render section headers and ToC entries. In the above example `example-chapter/_index.md` defines the title of the example chapter and the order in which it appears in the main table of contents. Similarly `example-chapter/section1-with-subsections/_index.md` defines metadata for the first section of the chapter.
- Media files should be contained in a directory with the same name as the page to which they relate. In this example, `example-chapter/section1-with-subsections/subsection1` contains media related to the `subsection1.md` page.

metadata

Metadata for the markdown files is presented at the head of the page using yaml. Any metadata may be defined – the module reference sections contain quite a lot of specific metadata – however the following defines some minimal metadata for the example page `example-chapter/section1-with-subsections/subsection1.md`.

```
---
title: Sub Section 1 Title
id: subsection1
weight: 10
include_toc: true
---
```

title

This should contain the rendered title of your page. To include a colon within a title, enclose the title in double-quotes.

id

This is the id used to identify the page by Hugo. It should usually be the same name as the file (for content files) or the parent directory (for `_index.md` files).

weight

This is an optional metadata field used to define the order in which sections are presented in the Table of Contents. If the *weight* field is not included, pages will be rendered in alphabetic order by default. For example, to define the sections and subsections of the above example in reverse order, the following metadata would need to be set:

```
example-chapter/
   section1-with-subsections/
      _index.md               # weight: 30 (place section1 page at the end of example-chapter)
      subsection1.md             # weight: 20 (place subsection1 page at the end of section1)
      subsection2.md             # weight: 10 (place subsection2 page at the start of section1)
   section2.md                # weight: 20 (place section2 in the middle of example-chapter)
   section3.md                # weight: 10 (place section3 at the start of example-chapter)
```

include_toc

Optional; enables or disables the collapsible table of contents. Should only be used on large, complex pages.

content

general style guidance

- All content should be authored in plain markdown without shortcodes and HTML should be kept to an absolute minimum, if used at all
- Minimalism is an absolute must. Fewer words are preferred
- Markdown files should be as short as possible
- Follow the naming and capitalization norms present in the GUI of the application – namely all headers and titles are in lower case, except for the very top-level chapter names. Similarly, use all lower case when referencing module names and controls.
- Headers in a file should not exceed level three (###)
- The primary authoring language is American English (which is what the application uses). Avoid idiomatic language where possible as the English version of the documentation may be read by people for whom English is not their native language
- Assume the reader has the application open while reading the user manual and only include images where they contribute to the explanation of complex functionality
- Use image callouts if you need to annotate an image (i.e. mark parts of the image with a letter or number and then explain the meaning in some text following the image). Do not place words directly into the image for annotations, as this makes localization difficult. See this page for an example.
- Changes to the content should be proposed via pull request or a similar mechanism
- Your submissions will be copy-edited – don't take it personally

technical information for processing modules

For each of the processing modules there is a special (collapsible) block of technical information at the top:

```
{{< details summary="Technical information" class="technical-info" >}}

description
: applies a tone mapping curve. inspired by Blender's AgX tone mapper.

purpose
: corrective and creative.

input
: linear, RGB, scene-referred.

processing
: non-linear, RGB.

output
: linear, RGB, display-referred

{{< /details >}}
```

This should mirror the information available as tooltip upon hovering the mouse over a processing module.

keyboard and mouse shortcuts

- Reference named keyboard keys using CamelCase (Ctrl, Shift, Alt, Esc, AltGr, CapsLock, PageUp, PageDown)
- Reference single letter keys in upper case. Quotation marks might help with clarification (press "H" to see a list of active shortcuts)
- Reference mouse actions using lower case, with multiple words joined by a hyphen (scroll, click, single-click, double-click, right-click)
- Connect combinations of keys/actions with a plus sign (Ctrl+Shift+H, Shift+double-click)

definition lists

The standard method of presenting information about darktable module controls is with the use of definition lists.

gui control name
A declaration of what the control does. For example “Set the exposure in EV units”.

You can include as many paragraphs as you like, but try to limit to 2 or 3 where possible.

 **a control accessed through a button with an icon**
When a control is activated using an icon, take a screenshot of the icon using the standard darktable theme (darktable-elegant-grey) and add it before the name of the control

gui combobox name
Comboboxes often have multiple options that all need to be displayed with separate definitions. Use bulleted lists with *italics* for the combobox values.

- *the first value*: What the first value means
- *the second value*: What the second value means

Definition lists are also used throughout the document, wherever a named piece of functionality needs to be defined. See, for example, darktable-cli .

notes

If you wish to present an important note to the user, use the following format:

Note: This is an important note.

---

fixed-width fonts and code blocks

Fixed width fonts (using the ` character) should normally only be used for code blocks and when referencing file names and command line parameters.

links

Internal links must be relative to the current file and must point to a valid markdown (.md) file. Start links with either `./` to represent the current directory or `../` to represent the parent directory.

- Links to a processing module should be in italics, e.g. *exposure*
- Links to a utility module should be in plaintext, e.g. history stack
- Link to a top level section by referencing the `_index.md` file, e.g. module reference
- Link to a tab in the preferences dialog: preferences > general
- Link to a specific preference setting: preferences > general > interface language
- Each header within a page can be linked to directly with an anchor link: contributing/notes

images

When taking screenshots from the darktable application itself, use the default darktable theme (darktable-elegant-grey).

Several filename suffixes can be used to control how an image is rendered.

icon
To insert an image as an icon, include `#icon` after the image name in the link. The markdown `![squirrel icon](./styles-conventions/squirrel.png#icon)` outputs the following: 

image width

You can set the image width to 25, 33, 50, 66, 75 or 100 per cent of the rendered page width by including #wxx after the image name in the link, where xx is the desired width. For example:

`![squirrel](./styles-conventions/squirrel.png#w25)` outputs



`![squirrel](./styles-conventions/squirrel.png#w75)` outputs



inline

With the exception of icons, images are included as block elements by default. You can override this by including `#inline` after the image name. This can be combined with the width setting as follows.

`![squirrel](./styles-conventions/squirrel.png#w25#inline)` outputs 

default

By default images are presented as block elements with 100% width. So `![squirrel](./styles-conventions/squirrel.png#w100)` and `![squirrel](./styles-conventions/squirrel.png)` are equivalent and both output the following:



13.5. translations setup

Translations of darktable’s documentation are maintained via Weblate .

Overview of the setup and flow of information for translations using Weblate

In short:

1. The docs are primarily maintained in English in the .md files present in `content/*.md`.
2. From there a POT file (`po/content.pot`) of all *translatable* strings is maintained.
3. Weblate maintains each language's individual PO files (`po/content.{lang}.po`) which contain the *translated* strings.
4. Upon deployment of the docs, language-specific .md-files are generated using `generate-translations.sh` which hugo then renders to .html files and the EPUB and PDF variants.

GitHub to Weblate: source string synchronisation

1. The original English files (`content/*.md`) are changed through a pull request on GitHub (see workflow).
2. A nightly GitHub action updates the POT file (`po/content.pot`) housing all translatable strings with `generate-translations.sh --no-translations`. Note: The script updates POT and PO files but we don't commit the updated PO files to the repository as in our setup the PO files are entirely managed by Weblate to avoid merge conflicts.
3. Weblate automatically pulls the updated POT file (triggered by the Weblate GitHub app) and internally populates the PO file of each translated language with new/updated strings
4. Weblate commits the changes to GitHub by opening a pull request.
5. The pull request gets merged after review.

Weblate to GitHub: translation synchronisation

1. Translations happen on Weblate. See translators guide for a guide on how to translate.
2. Weblate internally updates each language's PO file and commits them to GitHub via pull request. To keep the number of pull requests at bay this happens in batches.

Deployment of translations

Upon deployment at docs.darktable.org translated .md-files are generated from the PO files in `/po/` using `generate-translations.sh --no-update`. This step is disabled for the auto-build GitHub Pages .

Adding a new language

1. Add a new language to Weblate. See Weblate's documentation for how to do this. Make sure the newly created PO file is committed to the dtdocs repository.
2. In the files `config.yaml`, `config-pdf.yaml` and `config-epub.yaml`, locate the `languages:` line.
3. Add the language you wish to translate. For example, the English entry looks like this:

```
en-us:
  title: darktable user manual
  weight: 1
```

4. Add new language files in the Hugo theme directories (`themes/hugo-darktable-docs-epub-theme/i18n`, `themes/hugo-darktable-docs-pdf-theme/i18n`, and `themes/hugo-darktable-docs-theme/i18n`).

Enabling and disabling languages

Languages are controlled by two independent systems that must be manually kept in sync:

- **`config.yaml` and `config-pdf.yaml`** — hugo's configuration. Languages listed here get their own section of the built website or PDF.
- **`disable-languages`** — A space-separated list of language codes in the repository's base directory. `generate-translations.sh` reads this file and skips generating `.{lang}.md` files for any language listed here, regardless of whether a PO file exists for it.

The language switcher in the theme only lists a language when translated `.{lang}.md` files exist for the current page — it uses Hugo's `.IsTranslated` and `.Translations` variables, which reflect actual files on disk, not `config.yaml` entries. A language present in `config.yaml` but listed in `disable-languages` will therefore not appear in the switcher, as no translated files are generated for it. Its `config.yaml` entry is dormant until it is removed from `disable-languages`.

To enable a language, it should be present in `config.yaml`, `config-pdf.yaml` and `config-epub.yaml` and absent from `disable-languages`.

To disable a language, it should be listed in `disable-languages`. The `config.yaml`, `config-pdf.yaml` and `config-epub.yaml` entries can remain in place for when the language is re-enabled.

Translating theme strings for website, EPUB, and PDF

There are three themes for the darktable documentation: one for the HTML website, one for EPUB, and one for the PDF. Each has a small set of UI strings (e.g. "Table of Contents", "Copyright", "Search") that must be translated manually — these strings are **not** managed by Weblate.

1. Go to `themes/hugo-darktable-docs-theme/i18n`.
2. Copy the file `en-us.yaml` and name the new file after your language using a locale code with a dash (e.g. `de-de.yaml`, `fr-fr.yaml`).
3. Translate the strings in the new yaml file.
4. Check the translated yaml file into git, push it to GitHub, and open a pull request to have your changes accepted.
5. Repeat for `themes/hugo-darktable-docs-epub-theme/i18n` and `themes/hugo-darktable-docs-pdf-theme/i18n`. These themes use shorter locale codes, using only the language code (e.g. `de.yaml`) or an underscore for regional variants (e.g. `pt_br.yaml`).

generate-translations.sh

This script is used by the nightly GitHub action and can also be run locally by maintainers who need to update translations manually or debug the build.

The script is a wrapper around po4a to orchestrate the interplay between the original .md files and the translations. The script reads the `disable-languages` file in the repository's base directory, filters out disabled languages, builds a temporary po4a config file containing all `content/**/*.md` files, and then calls `po4a $1 --verbose $po4a_conf` — the flag is passed through directly to po4a. It requires one of the three arguments.

`--no-translations`
: Generates POT and PO files (`po/content.pot` and `po/content.{lang}.po`) from the source .md files (`content/*.md`). Does *not* produce any translated output files. New or changed strings from the source files are pulled into the POT/PO files, but no `*.{lang}.md` files are created. This is used after updates to the English source files to prepare the POT and PO files for Weblate.

`--no-update`
: Generates the translated Markdown files from the existing PO files, without updating the POT/PO files. This means the source files (`content/*.md`) are not re-read; existing translations are rendered directly into localized `.{lang}.md` files.

`--rm-translations`
: Removes the generated translated files. po4a deletes the localized `*.{lang}.md` output files previously created by `--no-update`. The PO/POT files are left untouched.

13.6. translators guide

You are welcome to help translate darktable's docs. Please read through this guide to learn the conventions and requirements for translation work.

Translations are maintained through Weblate , apart from a small number of Hugo-specific strings . Unregistered users can add suggestions for translations. To contribute to the translations you need to register with Weblate .

We do **not** accept automated AI translations without human review and quality control.

Recommended reading:

- Guide for styles and conventions : Formalities and conventions on how dtdocs should be done
- Translations setup contains information on how data flows between the dtdocs github repository and Weblate
- Documentation on how to use weblate

Visit the projects page to see which languages need work and select one you are fluent in. You can then dive into translation work by picking one of the categories (e.g. "Untranslated strings").

If you want to know where a translatable string is located in the docs, you can check the *string information* and then *details* on the right side.

When translating, always preserve the original Markdown formatting — bold, italics, links, and shortcodes must appear in the translated string exactly as they do in the source.

translating a specific section of the docs

If you want to translate a specific page, you need to use Weblate's search function as there is no direct way of selecting e.g. "translate filmic.md". But if you either open "all strings" or "untranslated strings" and use the filter-search box at the top to look for the strings *location*: E.g. by entering `location:"filmic" or` location:translators-guide`. Here you can also search for specific strings like` "You are welcome to help translate darktable's docs"`. Note: If the searched string contains spaces use quotation marks. If the searched string contains markdown-formatting this needs to be included in the searched string exactly as well.

After translating a string you can click "Save and continue" to go through the translatable strings in order.

It is helpful to open the page you are currently working on in a second window (e.g. using split view) to better see the context of the translatable strings.

technical information sections

At the top of each processing module's page there is a collapsible "Technical information" section that needs special formatting and is supposed to exactly mirror the descriptions available as tooltips in darktable. All languages were seeded with the exact translations from darktables program code/translations with "needs edit" to ensure human oversight.

darktable 5.8 user manual
November 16, 2026