

---

# **LuminanceHDR Documentation**

***Release 2.5.0***

**LuminanceHDR**

**May 03, 2017**



<b>1</b>	<b>Getting Started</b>	<b>3</b>
1.1	Getting Started . . . . .	3
<b>2</b>	<b>Contents</b>	<b>11</b>
2.1	Creating HDR images . . . . .	11
2.2	Editing . . . . .	16
2.3	Using Editing Tools . . . . .	18
2.4	Example . . . . .	20
2.5	Tools . . . . .	20
2.6	Tone Mapping . . . . .	23
2.7	Color Management . . . . .	29
2.8	Preferences . . . . .	29
2.9	Glossary . . . . .	38
2.10	About LuminanceHDR . . . . .	40
<b>3</b>	<b>Indices and Tables</b>	<b>47</b>



Luminance HDR is an open source workflow tool for HDR imaging.

Its highlights are:

- streamlined user interface in more than a dozen languages
- support for all major HDR file formats
- batch hdr creation
- batch tonemapping
- resize/crop/transform HDRs
- color managed workflow
- works on Linux, Windows and Mac OS X
- free to distribute among family, friends and enemies



## Getting Started

### Introduction

Luminance HDR is an open source graphical user interface application that provides a workflow for HDR imaging.

### Summary of features

Current supported features include:

1. Create an HDR from a set of files.
2. Tone map an HDR image to get a LDR image.
3. Save and load HDR images.
4. Rotate and resize HDR images.
5. Apply projective transformations to HDR images.
6. Copy exif data between sets of images.

The **first feature** is accessible via the “File -> New Hdr...” wizard: in order to create an HDR the user can either load a set of JPEG files, a set of RAW files, or a set of TIFF files (8bit or 16bit). Raw files are processed with LibRaw in order to obtain a (8 or 16 bit) tiff file. For more information read this page. The pictures must have been taken at the same scene, with different exposure settings (change the exposure time and/or aperture, and use a tripod if you have one). The newly created HDR will be available in the workspace as soon as the HDR creation wizard has ended. The input files can be aligned via two alignment engines: `align_image_stack` and MTB. The set of images can contain moving objects. This can result in an (unwanted) effect called ghosting. Luminance HDR provides an interactive anti-ghosting tool that can help avoid such artifacts. Read the chapter about the creation of an hdr for more information about the alignment engines and the interactive anti-ghosting tool. To tone map an HDR file to get an LDR image (**second feature**) you can press the “Tonemap the HDR” button. Via the “File -> Open Hdr...” wizard you can choose to load in the workspace an HDR image file, and the “File->Save Hdr as...” item lets you save the currently selected hdr image to a HDR image file format (**third feature**). Users can also rotate and resize (**fourth**

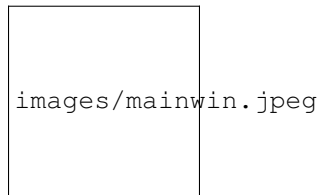
**feature**) the currently selected hdr image via the “Image” menu item, see below. It is also possible to apply panoramic (projective) transformation to a Hdr image via the “Image” menu item (**fifth feature**). In order to create an HDR image Luminance HDR requires to have a set of images with exif data in it. Luminance HDR requires this information to get the exposure settings for an image in the set. When Luminance HDR doesn’t find this information in an image it warns the user and aborts the hdr creation process. To cope with this requirement Luminance HDR provides a panel that performs a one-to-one copy of the exif data between two sets of files (**sixth feature**).

## Using Luminance HDR

This chapter describes the most important elements of Luminance HDR: the Main window, the “New Hdr...” wizard procedure, the Resize tool, the interactive tone mapping window, the batch tone mapping, the copy exif data tool and the Preferences panel.

### The main window

Here’s the main window that you can see once the program has launched and an image has been loaded.



### The menubar

At the very top you can see the menubar which, as its name implies, contains the various menus. When an item in a menu is “grayed out” it means that you cannot use (because it doesn’t make sense) that particular function at that time. For example you cannot tone map an HDR unless you have at least one HDR image loaded in the workspace. In case the text describing an item in a menu is not clear enough, below you can find a complete reference of all the items contained in all the menus.

### The toolbar

Below the menubar you can see the toolbar. It simply contains some of the most frequently used functions listed in the menus: “New HDR...”, “Open HDR...”, “Save as...”, “Save All” and “Exit”. Again, if an item is “grayed out” it means that you cannot use (because it doesn’t make sense) that particular function at that time.

### The workspace

The main gray area is the workspace. Here you can see all the HDR images which you can work on. As soon as the program is launched the workspace is empty. To have an HDR in the workspace you can either load an existing one (File->Open Hdr...) or create a new one (File->New Hdr...).



## Visualization of an HDR

All of the visualization options do not modify the current HDR, they are only a visualization tool. In the picture above you can see what an HDR image looks like once it is loaded in the workspace. From left to right, in an HDR image titlebar you can see:

1. A gamma combobox, which changes the visualization brightness.
2. The green histogram with its blue “histogram selection” (you can use the mouse to drag it and/or move its boundaries).

In the View menu you can find the visualization options, which fall in 2 categories:

1. Zoom options: (These are: View->Fit to window, Normal size, Zoom in, Zoom out).
2. Histogram options: (all under: View->HDR Histogram->...)

The zoom options are self explanatory, they deal with the fitting of the HDR in its containing window.

The histogram options require more explanation: we somehow have to visualize an HDR image on a CRT or LCD, even if they can only show a normal LDR with 8 bit per color channel. So a simple “luminosity compression” algorithm is performed. The problem is that when an HDR has a “wide” histogram, (a high gamut of dynamic range) its not possible, even with this (simple) “luminosity compression” algorithm, to show correctly at the same time all the regions of different luminosity in the image (this indeed would be the tone mapping’s job). So you may ask: “Why do we need this tool?” The answer is that, for example, you may want to visualize correctly all the regions of luminosity of your HDR image by:

1. narrowing down the range of the visible histogram (View->HDR Histogram->Low Dynamic Range)
2. dragging repeatedly the blue rectangle over the different areas of the green histogram.

## Operations on an HDR

As soon as at least one HDR image has been loaded in the workspace you can:

- Save it (File->Save Hdr as...): This is useful when you have just created an HDR from a set of JPEGs (or RAWs or TIFFs) via the “File->New Hdr...” wizard.
- Tone map it (Image->Tonemap the Hdr...): Tone mapping an HDR involves showing another window, called “Interactive Tone Mapping window” which uses the HDR as a source to create an LDR.

## The Menu reference

In this section you can find a complete reference describing what all the items in the menu do.

**File -> New Hdr...** launches a wizard that enables you to create an HDR starting from either a set of JPEGs or a set of RAWs, or a set of 8 or 16 bit TIFF files.

**File -> Open Hdr...** launches a window that lets you load in the workspace either an existing HDR image file format (OpenEXR, Radiance RGBE, PFS stream) or a RAW file or also a TIFF file.

**File -> Save Hdr as...** launches a window that lets you save the HDR image currently selected in the workspace to a HDR image file format (OpenEXR, Radiance RGBE, PFS stream, or 32bit or LogLuv TIFF).

**File -> Exit** Exits the program

**Image -> Rotate CounterClockWise** modifies the HDR image currently selected in the workspace by rotating it counterclockwise.

**Image -> Rotate ClockWise** modifies the HDR image currently selected in the workspace by rotating it clockwise.

**Image -> Projective Transformation...** launches a window that lets you apply a projective (aka panoramic) transformation to the HDR image currently selected in the workspace.

**Image -> Resize the Hdr...** launches a window that lets you resize the HDR image currently selected in the workspace.

**Image -> Tonemap the Hdr...** launches a window that lets you tone map the HDR image currently selected in the workspace.

**View -> HDR Histogram -> Fit to dynamic range** sets the boundaries of the blue “histogram selection” rectangle to the leftmost and rightmost values of the histogram.

**View -> HDR Histogram -> Low dynamic range** sets the boundaries of the blue “histogram selection” rectangle to values which enable a correct representation of the image on a LCD/CRT. The “histogram selection” rectangle can later be dragged.

**View -> HDR Histogram -> Shrink dynamic range** sets the boundaries of the blue “histogram selection” rectangle closer to each other.

**View -> HDR Histogram -> Extend dynamic range** sets the boundaries of the blue “histogram selection” rectangle away to each other.

**View -> HDR Histogram -> Decrease Expos** moves the boundaries of the blue “histogram selection” rectangle left

**View -> HDR Histogram -> Increase Exposure** moves the boundaries of the blue “histogram selection” rectangle right.

**Tools -> Preferences...** launches a window that lets you configure the global behaviour of Luminance HDR.

**Tools -> Copy Exif Data...** launches a window that lets you copy (a one-to-one copy) the exif tags in a set of images into another set.

**Help -> Documentation...** Launches the help window containing this documentation.

## Features Overview

### Streamlined Localized User Interface

Creating HDR files is very easy: you just drop your bracketed shots from file or photos manager to the main window and in few more clicks you have an HDR file which you can then tonemap.

Luminance HDR is available in the following languages:

- Chinese
- Czech (outdated)
- Danish
- English
- Finnish
- French (outdated)
- German
- Hungarian (outdated)
- Indonesian (outdated)
- Italian
- Polish (outdated)

- Portuguese (Brazilian)
- Romanian
- Russian
- Spanish
- Turkish (outdated)

If your language is not supported or localization is incomplete or outdated and you are willing to take care of that, please learn how to do it.

## **Supported HDR File Formats**

Luminance HDR reads and writes:

- OpenEXR (.exr)
- Radiance RGBE (.hdr)
- Logluv TIFF (.tif)
- 16/32bpc TIFF (.tif)
- PFS stream (.pfs)
- FITS (.fit)

Additionally Raw images can be opened (using LibRaw) as both HDR and LDR images.

## **Supported LDR File Formats**

Luminance HDR supports the following low dynamic range file formats:

- JPEG
- TIFF (16bpc and 8bpc)
- PNG (only saving)
- PBM (only saving)

Luminance HDR can also write Exif metadata to JPEG, PNG and TIFF files using Exiv2.

## **Extensive Help System**

We don't really pretend that HDR imaging is easy. It surely can be easy if lousy results is what you want. For the rest of you we bundle a solid help system to help you getting the most of your pictures.

The help system is, again, localizable. Right now only English version is available, but you can contribute a translation into your native language, if you really want to.

## **Free to Distribute**

Luminance HDR is a free-as-in-speech application licensed under terms of GNU GPL. Which in fact means:

- you don't have to pay developers of Luminance HDR for a license;
- you may install Luminance HDR on as many computers as you can reach;

- you may for god's sake modify the source code, but you will have to share it with us, if you distribute the modified version.

## FAQ for Luminance HDR

**Q: What is the meaning of the name Luminance HDR?** A: The name can be decomposed in 3 parts: Qt-pfs-gui.

**Qt** the program uses Qt4 ([www.trolltech.com](http://www.trolltech.com)) to show its graphical widgets.

**pfs** the main backend library and original sourcecode base.

**gui** this stands simply for graphical user interface.

**Q: What is the meaning of the various settings for tone mapping operator X?** A: To answer precisely this question one would have to explain the inner workings of the tone mapping operator X, in terms of the original research paper. At the end of the day all that matters (to some people, at least) is to fiddle with the settings until you obtain a nice result.

**Q: Why can't Luminance HDR transfer the exif tags to TIFF files?** A: Because the library Luminance HDR uses to perform this task (exiv2) doesn't support writing to tiff files yet. It's in the working, though.

**Q: Should I store JPEG or RAW files for HDR?** A: Both give the same result, provided you capture all the dynamic range in the scene. This means that with RAW files you may need to capture less files than with JPEGs. On the other hand creating an HDR with JPEG files is more "lightweight" process (reduced memory footprint).

**Q: Where can I get information about all the various HDR formats?** A: An overview (rather technical) can be found at [http://www.anyhere.com/gward/hdrenc/hdr\\_encodings.html](http://www.anyhere.com/gward/hdrenc/hdr_encodings.html).

- OpenEXR: Industrial Light and Magic format, widespread use, best compression ratios. [www.openexr.com](http://www.openexr.com)
- LogLuv TIFF: see [http://en.wikipedia.org/wiki/Logluv\\_TIFF](http://en.wikipedia.org/wiki/Logluv_TIFF) and <http://www.anyhere.com/gward/pixformat/tiffluv.html>
- Radiance RGBE: see [http://en.wikipedia.org/wiki/Radiance\\_%28software%29#HDR\\_image\\_format](http://en.wikipedia.org/wiki/Radiance_%28software%29#HDR_image_format)
- PFS: This format stores the binary internal (float) representation of images. It is very size demanding, and it supported only by pfstools/pfscalibration/pfstmo, Luminance HDR and a few other applications. On the other hand it is a lossless format and supports metadata (tags).
- Float TIFF (aka 32 bit TIFF): very size demanding (similar to pfs above).

**Q: Can we have the tonemapping dialog apply its settings as soon as the user changes a value? In other words, can we avoid hav**

A: Given how the tone mapping panel is implemented right now, this possibility has been discarded.

## Hints

### Contents

- *Hints*
  - *Single Raw file workflow*
  - *Post-processing of the LDR*
  - *Align\_image\_stack*

## Single Raw file workflow

You can load directly a raw file in LuminanceHDR via the “File->Load Hdr...” menu item. Doing so you will be able to tone map directly this single raw file. This means that there is no need to create different exposures in Ufraw from your raw file.

When the user wants to load a single raw file in the main workspace a RAW->TIFF conversion takes place by calling the dcrw executable. For more information please read the page about Raw Conversion.

If you still don't like what LuminanceHDR does with your raw file and you want to process your raw file directly before loading it in LuminanceHDR (white balance, color profile, and so on), you can use Ufraw to tweak the color settings of your raw file, and then save the result as a 16-bits tiff file. Just remember to save the result as a 16-bits file, or you'll lose some dynamic range in the process.

## Post-processing of the LDR

1. If you don't like the result of a specific tone mapping operator, please keep in mind that after the tone mapping step you can still use tools like GIMP to post process the resulting image. For example, you can still fix the brightness, change the gamma or the levels, and so on.
2. Some users have reported [1] pleasant results combining in GIMP 2 LDRs: one obtained with Fattal and the other one with Drago. Drago for the first layer and Fattal for the second in overlay mode (70% can be a good starting point). This is not a silver bullet technique, these values can be thought as a starting point, you can then go on tweaking the opacity value and the tone mapping parameters, your mileage may vary.
3. You can also put the Fattal image down as the master and then layer the Drago image on top in overlay mode with c.50% opacity [1]. [1] <http://www.flickr.com/groups/luminance/discuss/72157600715644855/>

## Align\_image\_stack

align\_image\_stack is a tool that can be found in the hugin project (<http://hugin.sf.net>). It is a standalone tool (an exe in windows) that can perform various functions, we use it in LuminanceHDR to do what it says, i.e. to align a stack of (LDR) images.

**Linux** As of today (23 Nov 2007) the hugin project has not published a release with align\_image\_stack in it yet (they are working hard for their next release). Linux users have to checkout the project's subversion repository and compile the sources. On this page ( [http://luminance.wiki.sourceforge.net/align\\_image\\_stack](http://luminance.wiki.sourceforge.net/align_image_stack) ) you can find how to do that.

**Mac OS X** The align\_image\_stack executable has already been included in the dmg file you downloaded.

**Windows** In windows the align\_image\_stack.exe file has to stay in the same directory of the luminance.exe file (or, as an alternative, in one of the directories listed in the PATH environment variable).

## Workflow

Luminance HDR imposes a particular workflow for HDR imaging. Basically it separates the workflow into two stages: creation of an HDR image and its conversion to an LDR image. Both creation and tone mapping of HDR images can be interactive (using GUI) and non-interactive (using console).

or those photographers who tend to shoot in similar lighting conditions the workflow can be greatly enhanced by using tonemapping settings files. This allows loading previously used tonemapping settings when doing interactive tonemapping and Batch Tone Mapping already available HDR images.



## Creating HDR images

This section covers both ways of creating HDR images in Luminance HDR:

### Interactive HDR Creation

You can access the wizard that will guide you through the process of creating a new HDR image via the File > New HDR image... menu item.

On the first page the wizard will ask you to select the set of images (of the same scene, but taken at different exposures) that are going to contribute to the final hdr (supported input: jpeg, raw and tiff -8 and 16 bit-).

Even if Luminance HDR doesn't find the required Exif data (Shutter Speed & Aperture) in your image set you can still proceed creating an HDR. To do so you have to insert **manually** the EV (exposure values) or stop difference values for the images in your the set.

The first page of the wizard enables the user to apply an automatic alignment step to the images in the set. It is possible to use one of two alignment options (or "engines"): *align\_image\_stack* and *MTB*.

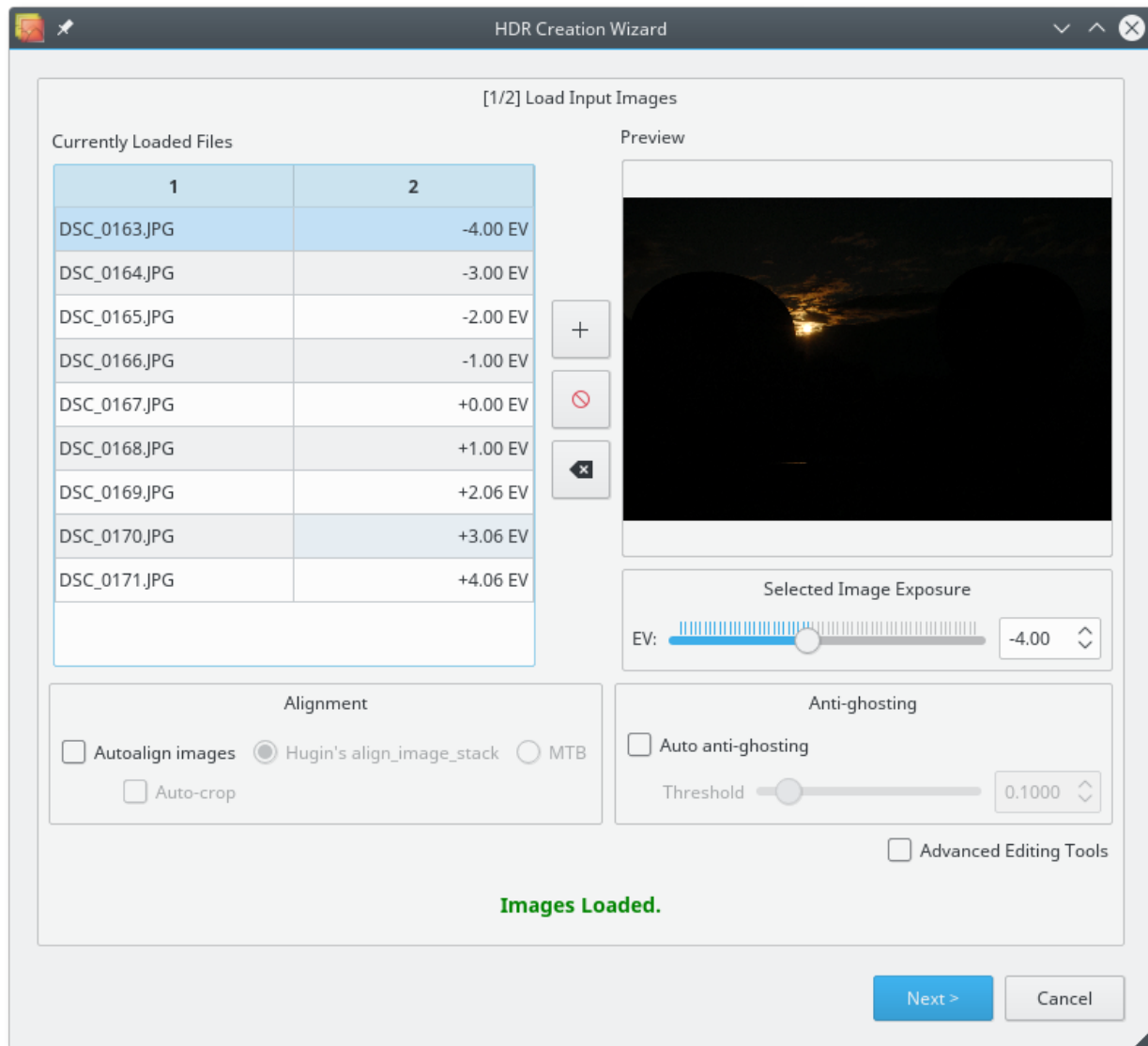
The first option is usually a good choice (MTB works only on LDR images and has a simpler model that does not take into account rotation).

If your image set consists of LDR images (JPEG, 8 bit TIFF, or RAW files) you can optionally open the Editing Tools Dialog clicking on the *Advanced Editing Tools* check box and clicking Next. A dialog will then show up which can be used to perform some "pre merging" editing activities as well manual anti ghosting.

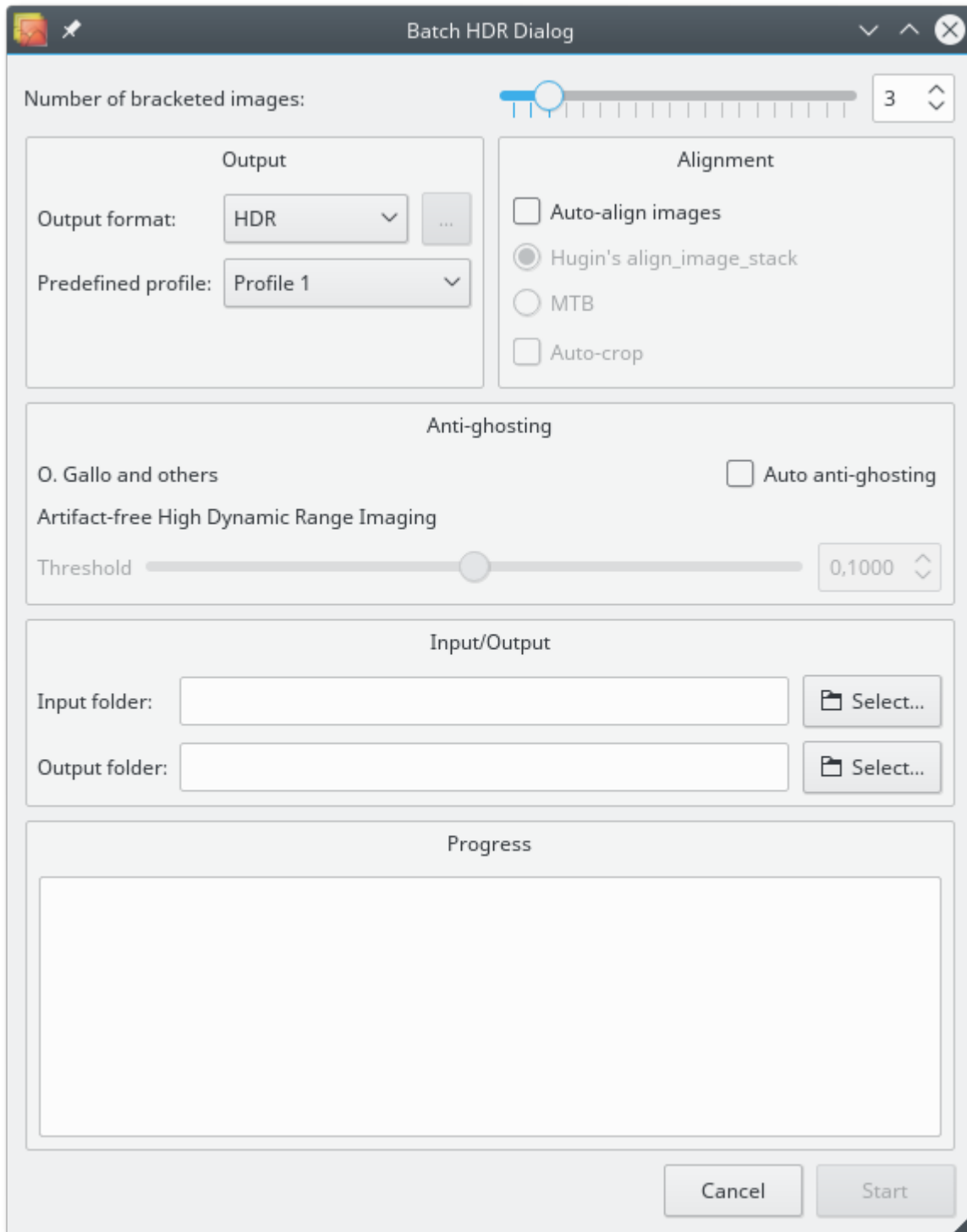
### Batch HDR Creation

This tool lets you creart HDR images starting from a set of bracketed photos ordered alphabetically.

You have to specify the number of bracketed photos, the HDR creation predefined profile, optionally the auto alignment engine to use for aligning the photos, the output hdr file format, the input directory where the bracketed photos are located, the output directory where the resulting HDRs will be saved to.







You can also always see what's happening in the Progress panel at the bottom.

## Command Line HDR Creation

Below are shown the command line switches used to create an HDR image and tonemap it. The relevant ones to just create an HDR are: -a to select the automatic alignment engine, -e to enter the EV values (one per image) if the images do not contain Exif data, -hdrModel to select the creation model and -s to save the resulting HDR

Usage: ./luminance-hdr-cli [OPTIONS]... [INPUTFILES]...:

-h [ -help ] Display this help. -V [ -version ] Display program version. -v [ -verbose ] Print more messages during execution. -c [ -cameras ] Print a list of all supported cameras. -a [ -align ] arg [AIS|MTB] Align Engine to use during HDR creation (default: no alignment). -e [ -ev ] arg EV1,EV2,... Specify numerical EV values (as many as INPUTFILES). -d [ -savealigned ] arg prefix Save aligned images to files which names start with prefix -l [ -load ] arg HDR\_FILE Load an HDR instead of creating a new one. -s [ -save ] arg HDR\_FILE Save to a HDR file format. (default: don't save) -g [ -gamma ] arg VALUE Gamma value to use during tone mapping. (default: 1) -r [ -resize ] arg VALUE Width you want to resize your HDR to (resized before gamma and tone mapping) -o [ -output ] arg LDR\_FILE File name you want to save your tone mapped LDR to. -t [ -autoag ] arg THRESHOLD Enable auto anti-ghosting with given threshold. (0.0-1.0) -b [ -autolevels ] Apply autolevels correction after tonemapping. -w [ -createwebsite ] Enable generation of a webpage with embedded HDR viewer.

HDR creation parameters - you must either load an existing HDR file (via the -l option) or specify INPUTFILES to create a new HDR:

**--hdrWeight arg** weight = triangular|gaussian|plateau|flat (Default is triangular)  
**--hdrResponseCurve arg** response curve = from\_file|linear|gamma|log|srgb (Default is linear)  
**--hdrModel arg** model: robertson|robertsonauto|debevec (Default is debevec)  
**--hdrCurveFilename arg** curve filename = your\_file\_here.m

LDR output parameters:

-q [ -ldrQuality ] arg VALUE Quality of the saved tone mapped file (1-100). -ldrTiff arg Tiff format. Legal values are [8b|16b|32b|logluv] (Default is 8b) -ldrTiffDeflate arg Tiff deflate compression. true|false (Default is true)

HTML output parameters:

**-k [ -htmlQuality ] arg VALUE Quality of the interpolated exposures, from the worst (1) to the best(4).**  
Higher quality will introduce less distortions in the brightest and the darkest tones, but will also generate more images. More images means that there is more data that needs to be transferred to the web-browser, making HDR viewer less responsive. (Default is 2, which is sufficient for most applications)  
**--pageName arg** Specifies the file name, of the web page to be generated. If <page\_name> is missing, the file name of the first image with .html extension will be used. (Default is first image name)  
**--imagesDir arg** Specify where to store the resulting image files Links to images in HTML will be updated accordingly. This must be a relative path and the directory must exist.  
Useful to avoid clutter in the current directory. (Default is current working directory)

Tone mapping parameters - no tonemapping is performed unless -o is specified:

**--tmo arg** Tone mapping operator. Legal values are: [ashikhmin|dragoldurand|fattall|ferradans|pat-tanaik|reinhard02|reinhard05|mailmantiuk06|mantiuk08] (Default is mantiuk06)

**--tmoFile arg** SETTING\_FILE Load an existing setting file containing pre-gamma and all TMO settings

Fattal:

**--tmoFatAlpha arg** alpha FLOAT

**--tmoFatBeta arg** beta FLOAT

**--tmoFatColor arg** color FLOAT

**--tmoFatNoise arg** noise FLOAT

**--tmoFatNew arg** new true/false

Ferradans:

**--tmoFerRho arg** rho FLOAT

**--tmoFerInvAlpha arg** inv\_alpha FLOAT

Mantiuk 06:

**--tmoM06Contrast arg** contrast FLOAT

**--tmoM06Saturation arg** saturation FLOAT

**--tmoM06Detail arg** detail FLOAT

**--tmoM06ContrastEqual arg** equalization true/false

Mantiuk 08:

**--tmoM08ColorSaturation arg** color saturation FLOAT

**--tmoM08ContrastEnh arg** contrast enhancement FLOAT

**--tmoM08LuminanceLvl arg** luminance level FLOAT

**--tmoM08SetLuminance arg** enable luminance level true/false

Durand:

**--tmoDurSigmaS arg** spatial kernel sigma FLOAT

**--tmoDurSigmaR arg** range kernel sigma FLOAT

**--tmoDurBase arg** base contrast FLOAT

Drago:

**--tmoDrgBias arg** bias FLOAT

Reinhard 02:

**--tmoR02Key arg** key value FLOAT

**--tmoR02Phi arg** phi FLOAT

**--tmoR02Scales arg** use scales true/false

**--tmoR02Num arg** range FLOAT

**--tmoR02Low arg** lower scale FLOAT

**--tmoR02High arg** upper scale FLOAT

Reinhard 05:

**--tmoR05Brightness arg** Brightness FLOAT

**--tmoR05Chroma arg** Chroma adaption FLOAT

**--tmoR05Lightness arg** Light adaption FLOAT

Ashikmin:

**--tmoAshEq2 arg** Equation number 2 truelfalse

**--tmoAshSimple arg** Simple truelfalse

**--tmoAshLocal arg** Local threshold FLOAT

Pattanaik:

**--tmoPatMultiplier arg** multiplier FLOAT

**--tmoPatLocal arg** Local tone mapping truelfalse

**--tmoPatAutoLum arg** Auto luminance truelfalse

**--tmoPatCone arg** cone level FLOAT

**--tmoPatRod arg** rod level FLOAT

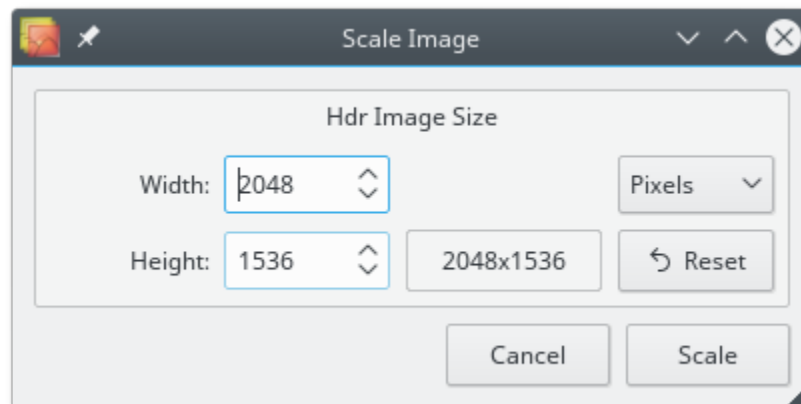
## Editing

After creating an new HDR file or opening an existing one you can do several things to them except tonemapping.

## Resizing

You can access this feature via the Edit > Resize... menu item.

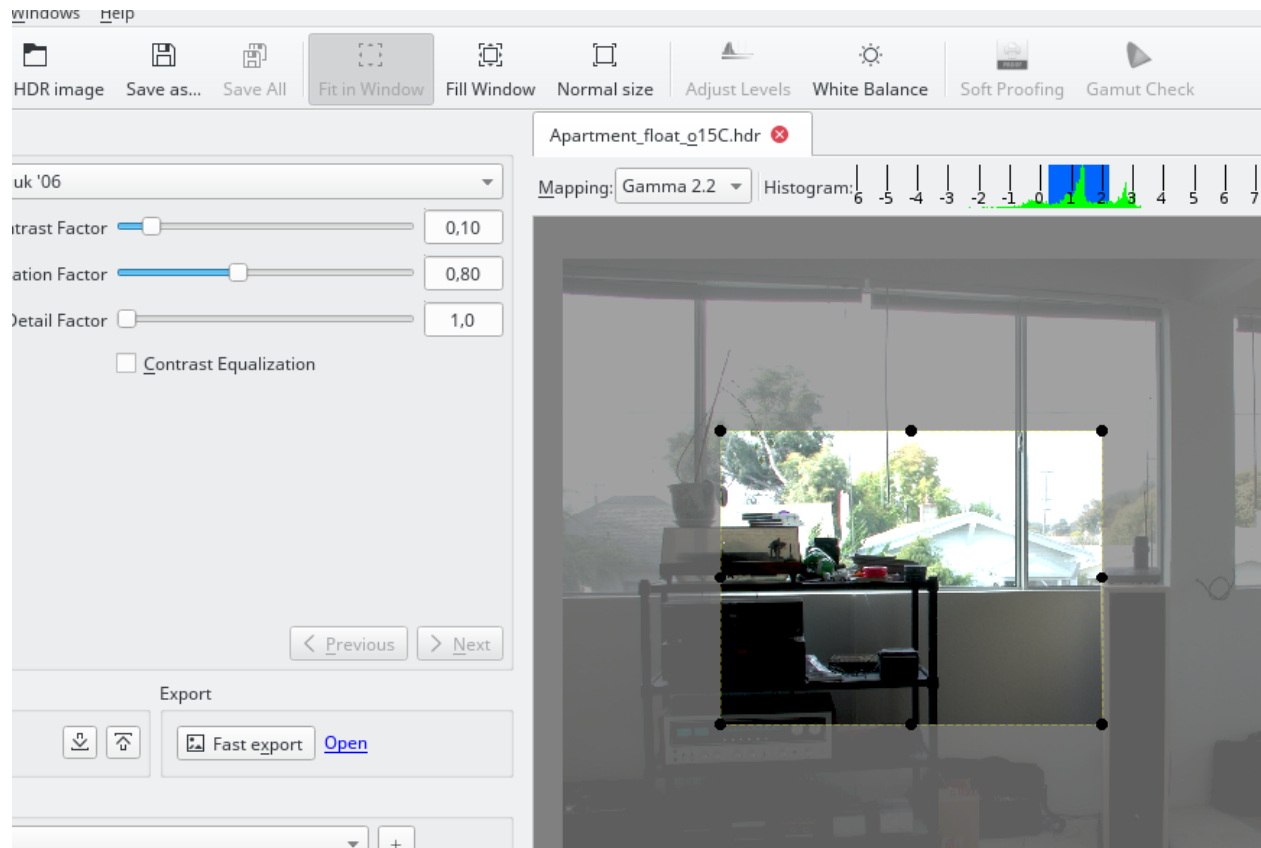
Luminance HDR can resize an HDR image to a given pixel size or percentage value counting from the original. If you use percentage, the final size in pixels will be calculated and displayed to the right from *Height* entry field.



Clicking **Scale** button will resize the HDR image.

## Cropping

To crop an HDR file to some area first you need to select this area. Click somewhere on an image, drag the mouse pointer to a side and release it. You will see something like this:



You can further edit the frame you created by dragging its edges or corners. You can also move the frame around by clicking inside it and dragging mouse pointer (that will change from an arrow to a hand icon).

When the frame is placed correctly, choose **Edit > Crop to Selection** in menu or use the relevant button in the toolbar. Luminance HDR will create a new unsaved HDR image that contains cropped version of the original image.

To get rid of the selection frame simply single-click anywhere outside the frame or use the **Edit > Remove Selection** menu item.

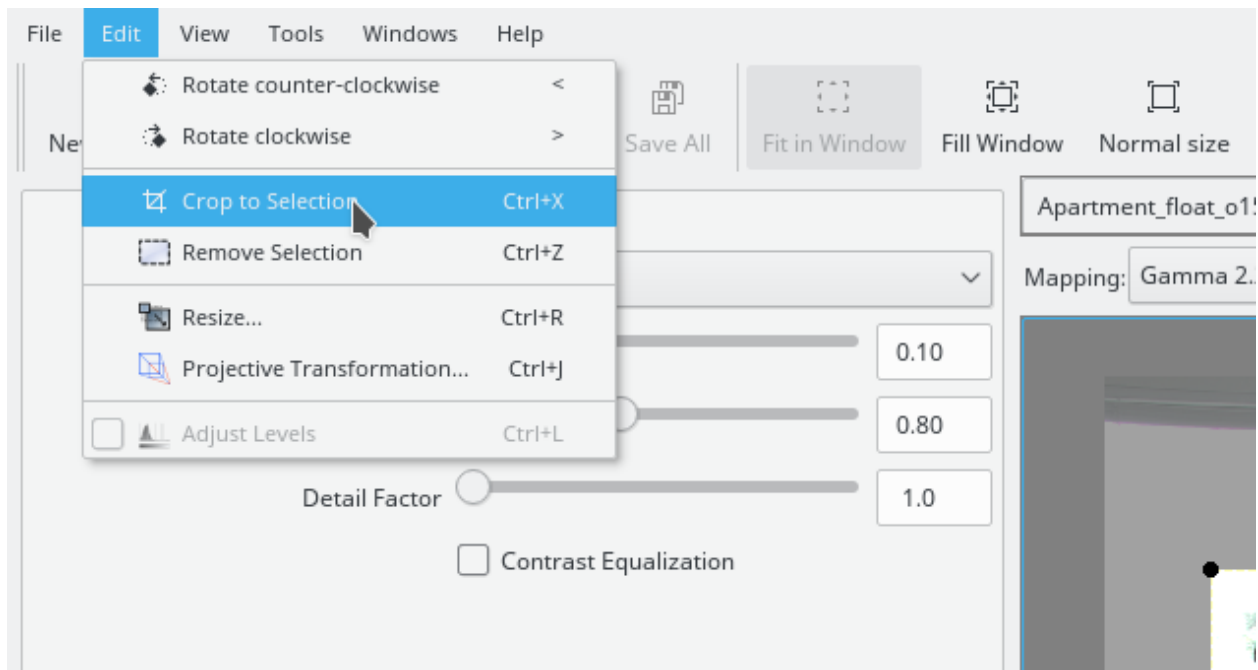
## Rotating

You can rotate an HDR image to 90 degrees a step, using **Edit > Rotate Counter-Clockwise** and **Edit > Rotate Clockwise** commands or **<** and **>** shortcuts respectively. Unlike setting an Exif orientation tag this will physically modify the HDR image.

## Projective Transformation

You can access this feature via the **Edit > Projective Transformation...** menu item.

With this tool the user is able to apply projective transformations to an HDR image. This is useful if you shoot mirrorball Hdrs and you want to unwrap them, for example.



The Angular projection accepts an Angle parameter which defines how many degrees from the viewing direction the projection should cover.

## Adjust Levels

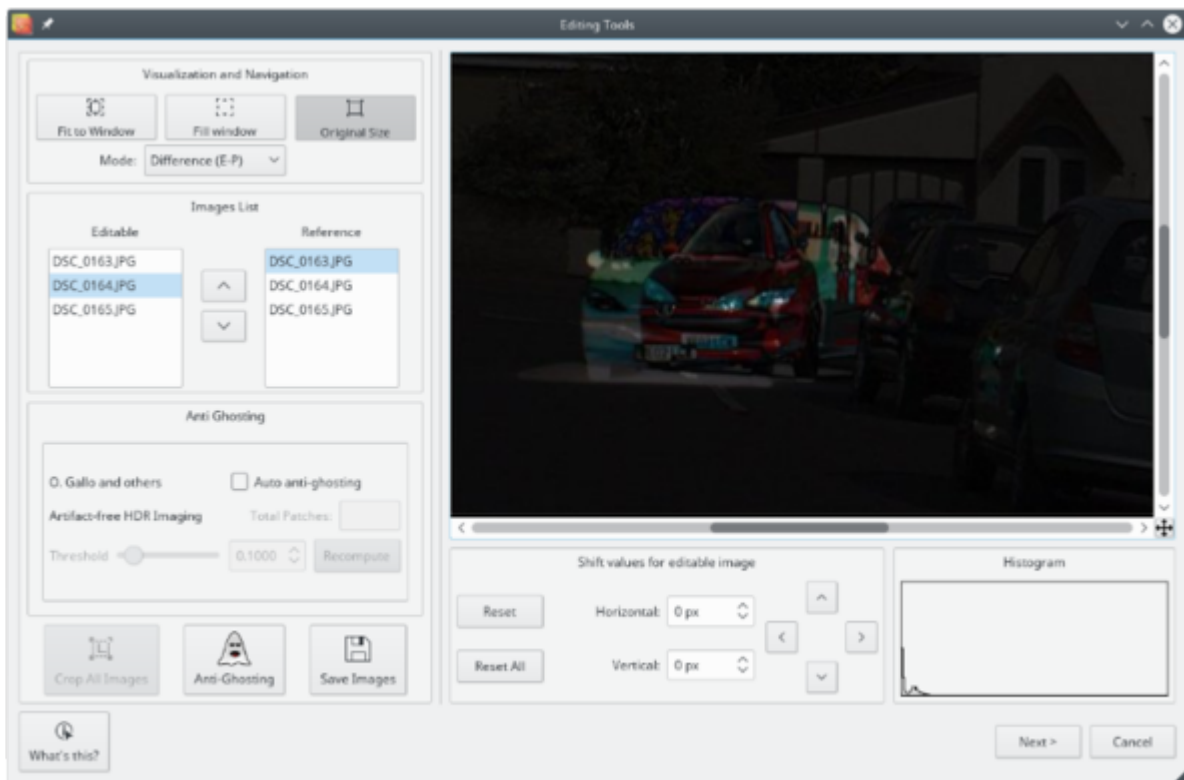
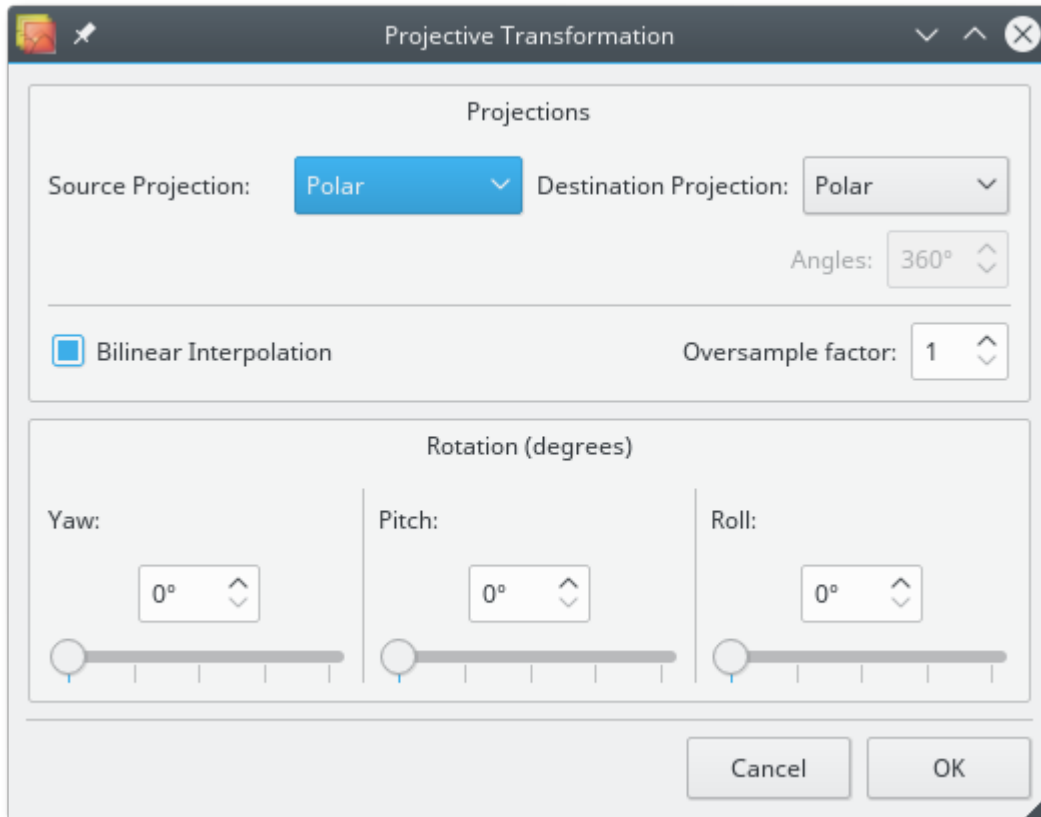
TODO.

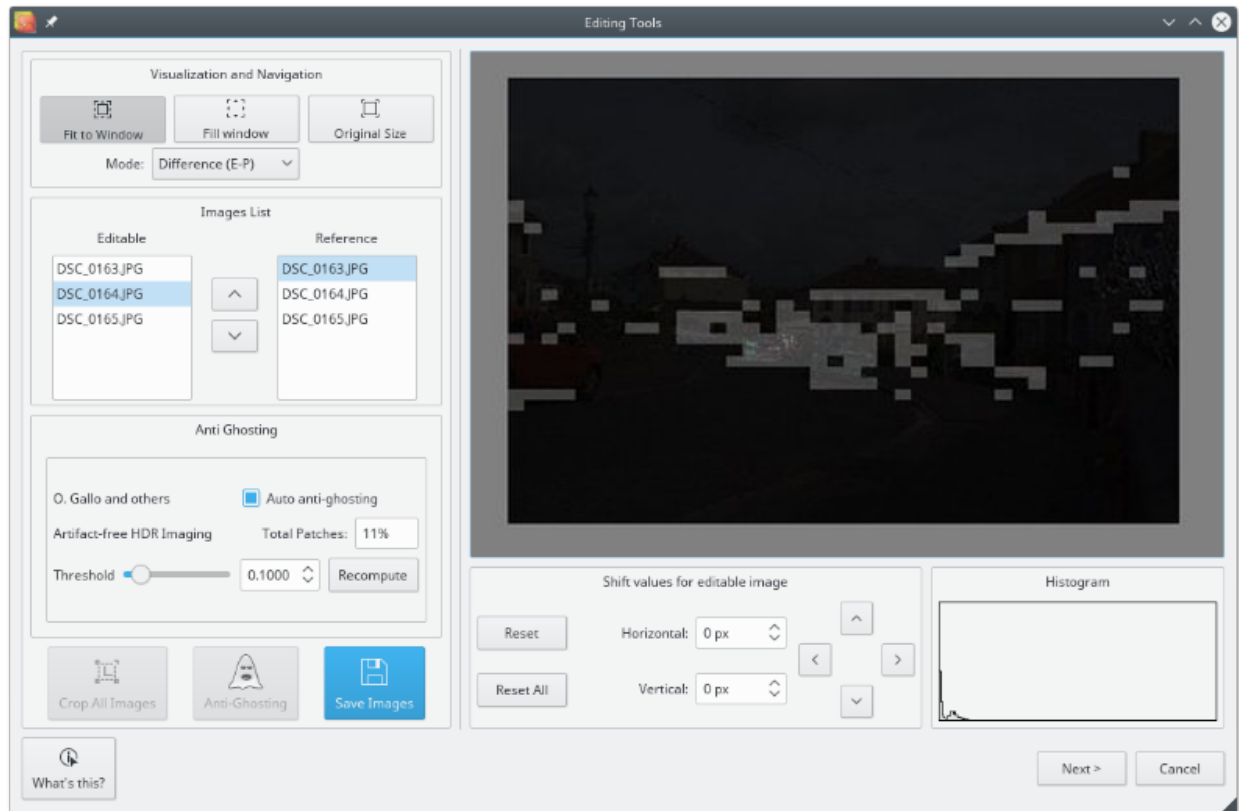
## Using Editing Tools

Within Editing Tools it is possible to do some manual editing activities before performing the hdr merge.

Using the arrow buttons shown in the bottom right part of the window you can shift the editable image to better align it. Press shift to shift the image by 10 pixels, control to shift by 50 pixels and shift-control to shift by 100 pixels. Use the visualization mode combo box on the upper left of the window to help you with the alignment. The best mode for aligning the images is Difference (E-P) where the difference between the editable and reference images is shown. After aligning the images you can crop them drawing a selection box and clicking on the crop button. Then you can save them for future use, all the relevant EXIF data will be saved as well. Now, if needed, you can proceed removing the ghosting artefacts resulting from moving objects in the scene. You have two options: manual and auto anti-ghosting. The auto anti-ghosting is pretty straightforward, just click on the Auto anti-ghosting check box and select a threshold value. Clicking on the Recompute button a series of patches is computed and visualized so that you can actually see how well the “ghost” it will be removed. The patches must cover all the affected “ghosting area” but, especially with misaligned images, it is possible that some patches will cover even unaffected areas. If this happens you can increase the threshold and recompute the patches again or you can manually add or remove patches just clicking on the image on the position you want a patch to be removed or added (clicking on a patch will remove it). Once you are done you can click next to close Editing Tools and continue with HDR merging.

If you prefer it is possible to manually draw a mask over the “ghost” using a lasso or a brush. To access the manual anti-ghosting feature click on the Anti-ghosting button. New options to work with the mask will then appear, you can chose the size of the brush, the mask strength (draw the mask semi-transparent), if to add or remove the mask while drawing. You can also save a previously created mask and apply it again to another image. You now must select the





good image from the list, the good image is the one that will be left unchanged after ghost removal. Keep in mind that you must draw a mask even on the good image. Basically all the pixels that change in all images must be covered by a mask. The Difference (E-P) visualization mode will help you discovering the ghost affected areas of the scene. The image below shows an example of a mask covering just a part of the ghost.

## Example

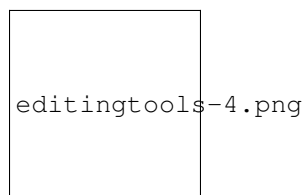


Fig. 2.1: Example of anti-ghosting mask

## Tools

### Copying Exif metadata

It is typical to lose Exif metadata of original images when constructing a new HDR image from bracketed exposures. You can fix this by doing a mass-copy of Exif metadata to tonemapped images.



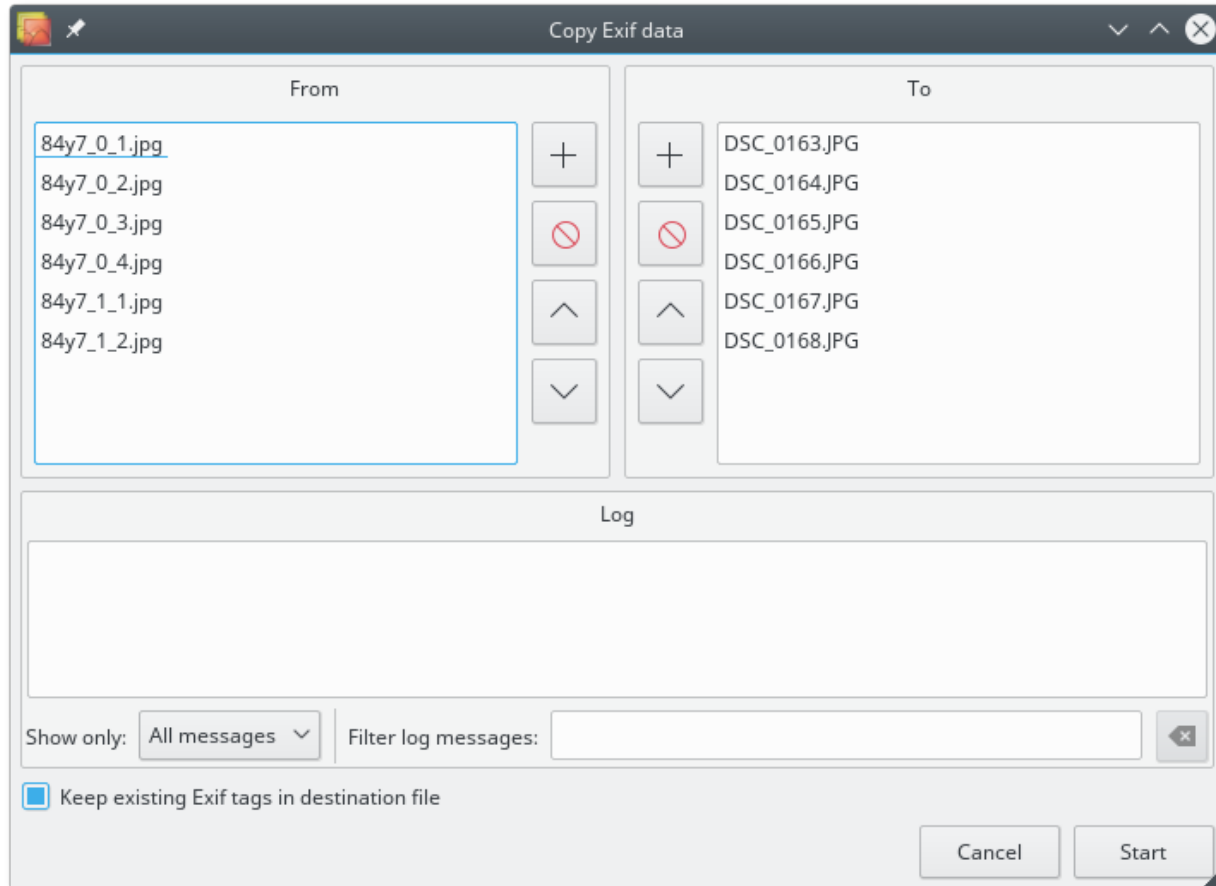


Fig. 2.2: Final HDR without ghost removal



Fig. 2.3: Final HDR after performing manual anti-ghosting

To do so, choose Tools > Copy Exif Data... menu item in the main window.



With this dialog you can copy the Exif data contained in a set of files (the sources) into another set of files (the destinations). This is a pairwise, one-to-one data copy, i.e. the first file in the destination list gets the Exif data from the first file in the sources list and so on.

Because Luminance HDR saves tonemapping settings description to Exif comment tag, “Keep existing Exif tags in destination file” checkbox is enabled by default.

## Tone Mapping

This section covers all three ways of performing tone mapping in Luminance HDR:

### Interactive Tone Mapping

To start tonemapping an HDR file click on Update preview button in the tone mapping panel or use `Ctrl+T`.

The tone mapping step first applies to the HDR the gamma value specified by the pregamma value, then the tone mapping operator is applied.

Adjusting pregamma is an optional step, so you can safely start with picking an operator. The available operators are listed in a combobox on top of Tone Mapping Panel. Most useful operators are listed first. The last two operators, Ashikhmin and Pattanaik, have less practical use for photography, but can be of interest for robotic vision specialists.

Tonemap

Operator

Mantiuk '06

Contrast Factor

0.10

Saturation Factor

0.80

Detail Factor

1.0


☐ Contrast Equalization


↶ Restore


< Previous


> Next

Tone Mapping Settings










Export

 Fast export

[Open](#)

Process

Result Size

256x170

+

Pre-gamma

↶

1.00

Update preview

Update current LDR☒ Auto Levels☒

Having chosen the operator, tweak its settings, choose resulting size and click *Update preview* button in the bottom to tonemap an HDR image to an LDR image. Please see the “Tonemapping operators reference” chapter for details on using a particular operator.

To tonemap only a part of the HDR for a quick preview you can select a rectangle of the image using the mouse and click the *Update preview* button. Alternatively, you can click on the small preview images on the right panel. Because three of nine available operators (both Mantiuks and Fattal) are gradient domain operators that cannot provide consistent results on all result sizes by design, the preview function is not very accurate for those operators, so the selection method is preferred.

If you wish to compare several results with different settings, choose the View > Lock Viewers menu entry and use the navigation widget in the lower right corner of an image to pan around. Select another image and note that contents of every image tab will be synced.

Sometimes the tone mapped LDR image is either too bright or too dark. You can adjust brightness either setting the new *Auto Levels* feature available in the Tone Mapping Panel through a check box or using *Levels* dialog (*Adjust Levels* button in the toolbar) that works just like in GIMP, Krita, Photoshop or any similar raster graphics editor with the only exception that it works on luminance channel only.

If you have a lot of pictures shot in same lighting conditions or just wish to save current tone mapping settings for future use, you can save them to a text file or in a centralized database. Use the button on the left in the *Tone Mapping Panel* under *Tone Mapping Settings* for saving to a file or the small button on the right for saving in the database. You can then add a comment that describe your settings. Use the “Load” buttons to reload the previously saved settings.

The text file contains:

- the currently selected tone mapping operator and its related settings.
- the values of the pre-gamma settings.

---

**Note:** Please note that since version 1.8.4 the application (at the time named Qtpfsgui) uses a new file format which doesn't contain the postgamma setting.

---

## Batch Tone Mapping

This window lets you tonemap a batch of M HDR images using N tonemapping settings files. After the process you'll get M\*N LDR images.

The top-left list contains the input HDRs, the top-right list contains the input tonemapping settings files.

You have to specify an output directory where the output LDR images will be saved to.

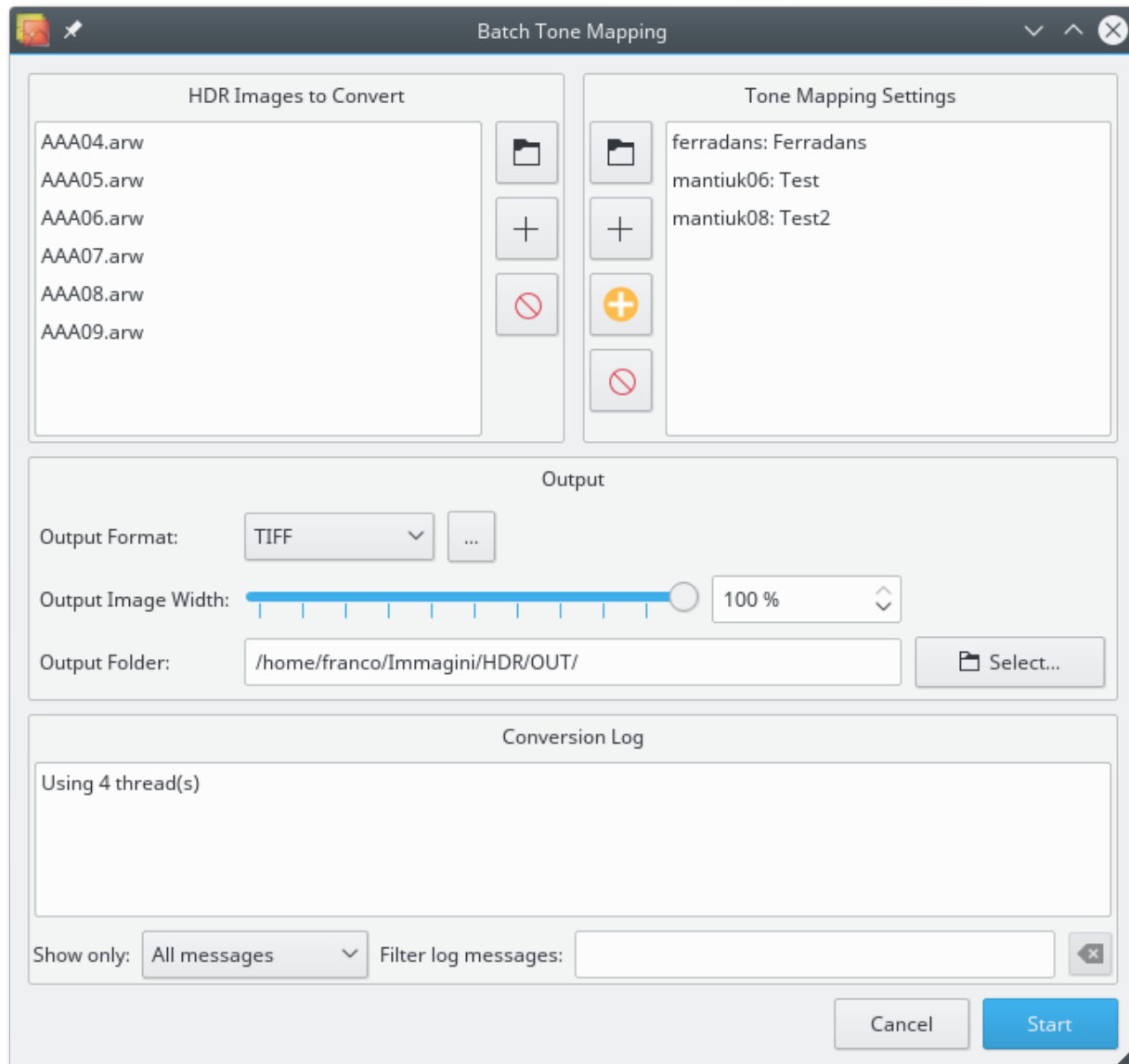
You can also always see what's happening in the Log panel at the bottom. You can filter the visualization of the log messages using the drop-down menu on the bottom-left and the text-field at the bottom-right of the window.

## Console Tone Mapping

Below are listed the command line switches used to create an HDR image and tonemap it. The relevant parameters for tonemap an image are: `-g` to define a gamma value to be applied before tonemapping, `-r` to resize the image to a given value before tonemapping, `--tmo` to select a tonemapping operator, `-o` to specify the resulting output file.

Usage: `./luminance-hdr-cli [OPTIONS]... [INPUTFILES]...`:

- `-h [ -help ]` Display this help.
- `-V [ -version ]` Display program version.
- `-v [ -verbose ]` Print more messages during execution.



**-c [ -cameras ]** Print a list of all supported cameras.  
**-a [ -align ] arg [AIS|MTB]** Align Engine to use during HDR creation (default: no alignment).  
**-e [ -ev ] arg EV1,EV2,...** Specify numerical EV values (as many as INPUTFILES).  
**-d [ -savealigned ] arg prefix** Save aligned images to files which names start with prefix  
**-l [ -load ] arg HDR\_FILE** Load an HDR instead of creating a new one.  
**-s [ -save ] arg HDR\_FILE** Save to a HDR file format. (default: don't save)  
**-g [ -gamma ] arg VALUE** Gamma value to use during tone mapping. (default: 1)  
**-r [ -resize ] arg VALUE** Width you want to resize your HDR to (resized before gamma and tone mapping)  
**-o [ -output ] arg LDR\_FILE** File name you want to save your tone mapped LDR to.  
**-t [ -autoag ] arg THRESHOLD** Enable auto anti-ghosting with given threshold. (0.0-1.0)  
**-b [ -autolevels ]** Apply autolevels correction after tonemapping.  
**-w [ -createwebpage ]** Enable generation of a webpage with embedded HDR viewer.

HDR creation parameters – you must either load an existing HDR file (via the `-l` option) or specify INPUTFILES to create a new HDR:

**--hdrWeight arg** weight = triangular|gaussian|plateau|flat (Default is triangular)  
**--hdrResponseCurve arg** response curve = from\_file|linear|gammallog|srgb (Default is linear)  
**--hdrModel arg** model: robertson|robertsonauto|debevec (Default is debevec)  
**--hdrCurveFilename arg** curve filename = your\_file\_here.m

LDR output parameters:

**-q [ -ldrQuality ] arg VALUE** Quality of the saved tone mapped file (1-100).  
**--ldrTiff arg** Tiff format. Legal values are [8b|16b|32b|logluv] (Default is 8b)  
**--ldrTiffDeflate arg** Tiff deflate compression. true/false (Default is true)

HTML output parameters:

**-k [ -htmlQuality ] arg VALUE** Quality of the interpolated exposures, from the worst (1) to the best (4). Higher quality will introduce less distortions in the brightest and the darkest tones, but will also generate more images. More images means that there is more data that needs to be transferred to the web-browser, making HDR viewer less responsive. (Default is 2, which is sufficient for most applications)  
**--pageName arg** Specifies the file name, of the web page to be generated. If <page\_name> is missing, the file name of the first image with .html extension will be used. (Default is first image name)  
**--imagesDir arg** Specify where to store the resulting image files. Links to images in HTML will be updated accordingly. This must be a relative path and the directory must exist. Useful to avoid clutter in the current directory. (Default is current working directory)

Tone mapping parameters - no tonemapping is performed unless -o is specified:

**--tmo arg** Tone mapping operator. Legal values are:  
 [ashikhmin|dragoldurand|fattall|ferradans|pattanaik|reinhard02|reinhard05|mailmantiuk06|mantiuk08] (Default is mantiuk06)  
**--tmoFile arg** SETTING\_FILE Load an existing setting file containing pre-gamma and all TMO settings

Fattal:

- tmoFatAlpha arg** alpha FLOAT
- tmoFatBeta arg** beta FLOAT
- tmoFatColor arg** color FLOAT
- tmoFatNoise arg** noise FLOAT
- tmoFatNew arg** new true/false

Ferradans:

- tmoFerRho arg** rho FLOAT
- tmoFerInvAlpha arg** inv\_alpha FLOAT

Mantiuk 06:

- tmoM06Contrast arg** contrast FLOAT
- tmoM06Saturation arg** saturation FLOAT
- tmoM06Detail arg** detail FLOAT
- tmoM06ContrastEqual arg** equalization true/false

Mantiuk 08:

- tmoM08ColorSaturation arg** color saturation FLOAT
- tmoM08ContrastEnh arg** contrast enhancement FLOAT
- tmoM08LuminanceLvl arg** luminance level FLOAT
- tmoM08SetLuminance arg** enable luminance level true/false

Durand:

- tmoDurSigmaS arg** spatial kernel sigma FLOAT
- tmoDurSigmaR arg** range kernel sigma FLOAT
- tmoDurBase arg** base contrast FLOAT

Drago:

- tmoDrgBias arg** bias FLOAT

Reinhard 02:

- tmoR02Key arg** key value FLOAT
- tmoR02Phi arg** phi FLOAT
- tmoR02Scales arg** use scales true/false
- tmoR02Num arg** range FLOAT
- tmoR02Low arg** lower scale FLOAT
- tmoR02High arg** upper scale FLOAT

Reinhard 05:

- tmoR05Brightness arg** Brightness FLOAT
- tmoR05Chroma arg** Chroma adaption FLOAT
- tmoR05Lightness arg** Light adaption FLOAT



Ashikmin:

- tmoAshEq2 arg** Equation number 2 true/false
- tmoAshSimple arg** Simple true/false
- tmoAshLocal arg** Local threshold FLOAT

Pattanaik:

- tmoPatMultiplier arg** multiplier FLOAT
- tmoPatLocal arg** Local tone mapping true/false
- tmoPatAutoLum arg** Auto luminance true/false
- tmoPatCone arg** cone level FLOAT
- tmoPatRod arg** rod level FLOAT

## Color Management

Luminance HDR has the ability to read the color profile embedded in your RAW or JPEG files, convert the colors to the internal sRGB color space for processing with the program, use the provided monitor profile to show the result in your monitor, do soft proof and gamut check using the printer profile provided. To begin with the color management support in Luminance HDR click on Tools->Preferences then select Color Management. Select a profile for the monitor, printer and camera (or set the option to Built-in to use the embedded profile). After tonemapping an HDR, the resulting image is displayed on the monitor with the proper colors, you can then click on the Soft Proofing button on the toolbar to see how the image will appear once printed on your printer. Clicking on the Gamut Check button will show in green the colors that are out of gamut, that is cannot be reproduced by your printer. If you choose to save the LDR in TIFF or JPEG format the standard sRGB profile is automatically embedded to the file for further processing with another color managed application.

## Preferences

You can configure behaviour of Luminance HDR using Preferences dialog available via Tools > Preferences... menu item.

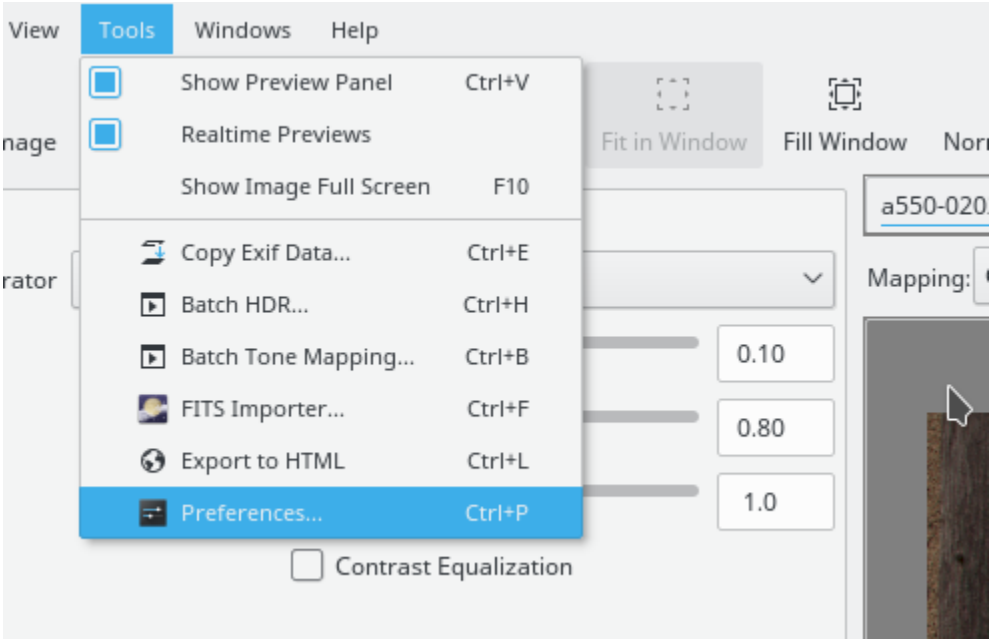
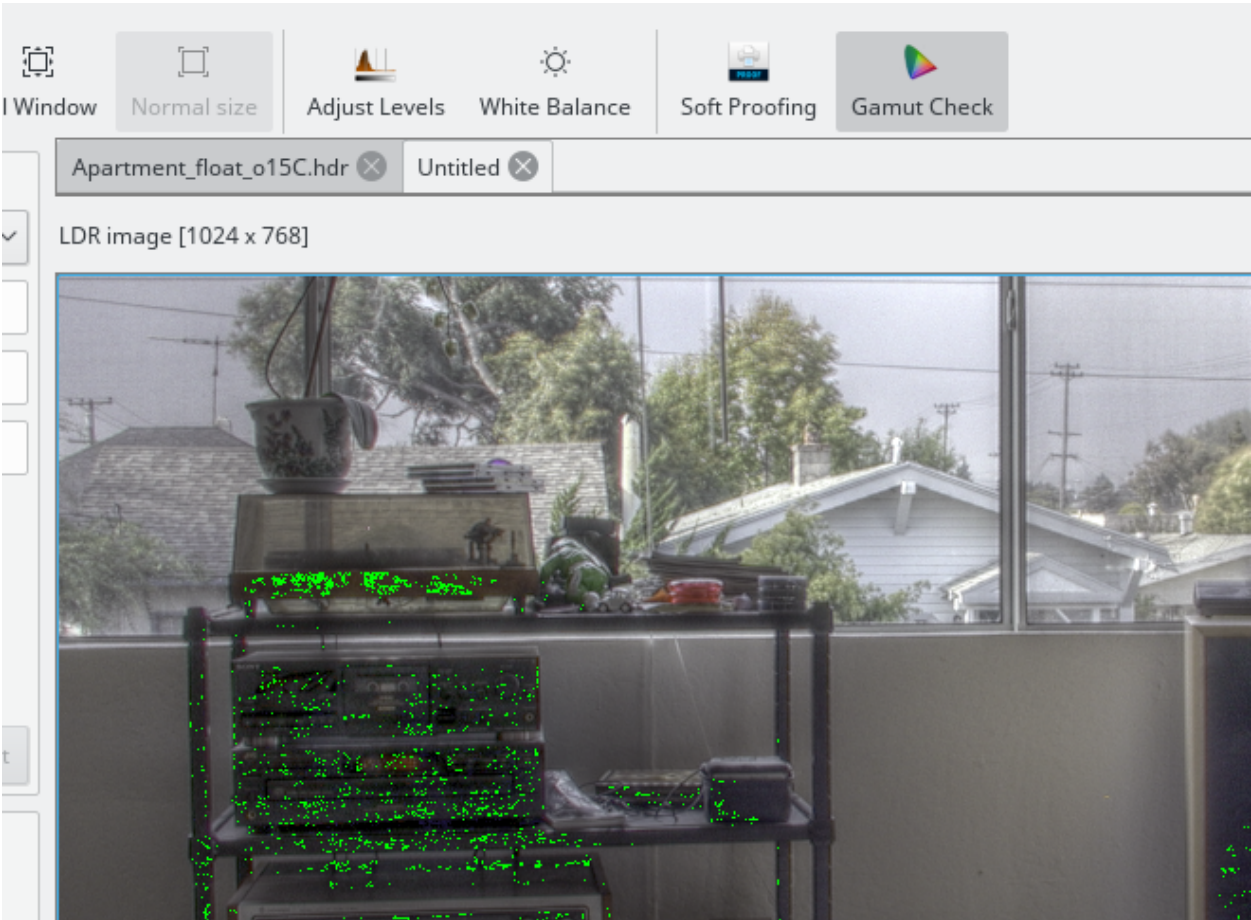
## Interface

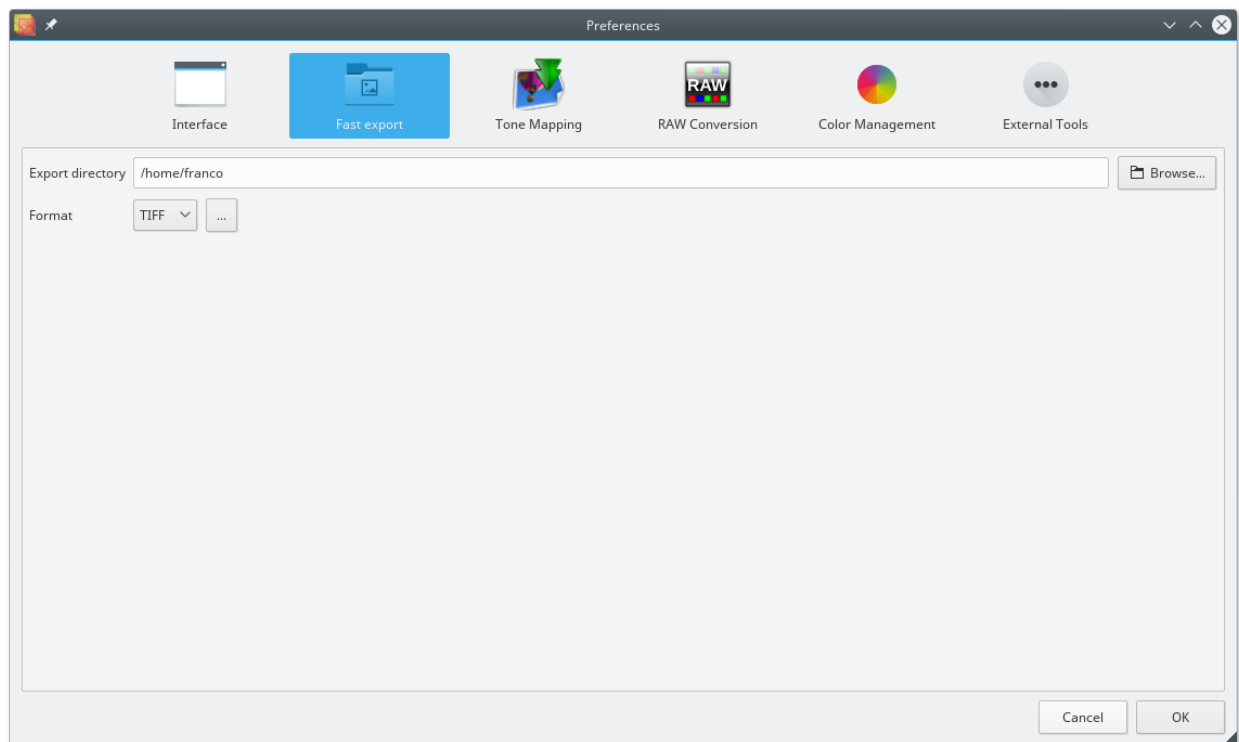
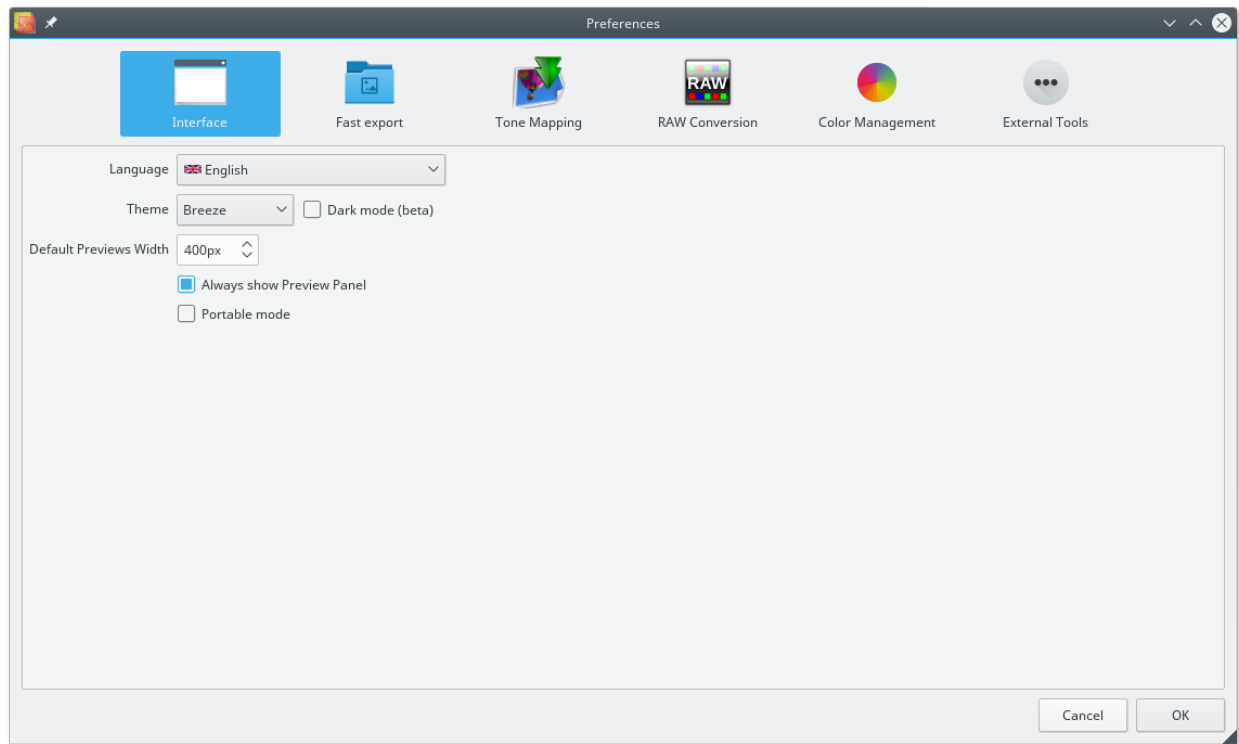
Here you can choose the language for the user interface and the help system. If no help system for your native language is available, the English version will be displayed. You can choose the size of a tonemapped preview image created when clicking on a picture in the preview panel. It is possible to choose an interface theme among the ones available on your platform. You can also set if you want to always show the preview panel in the main window. Enabling the Portable mode will allow you to put the application on a USB stick with all its settings.

## Fast Export

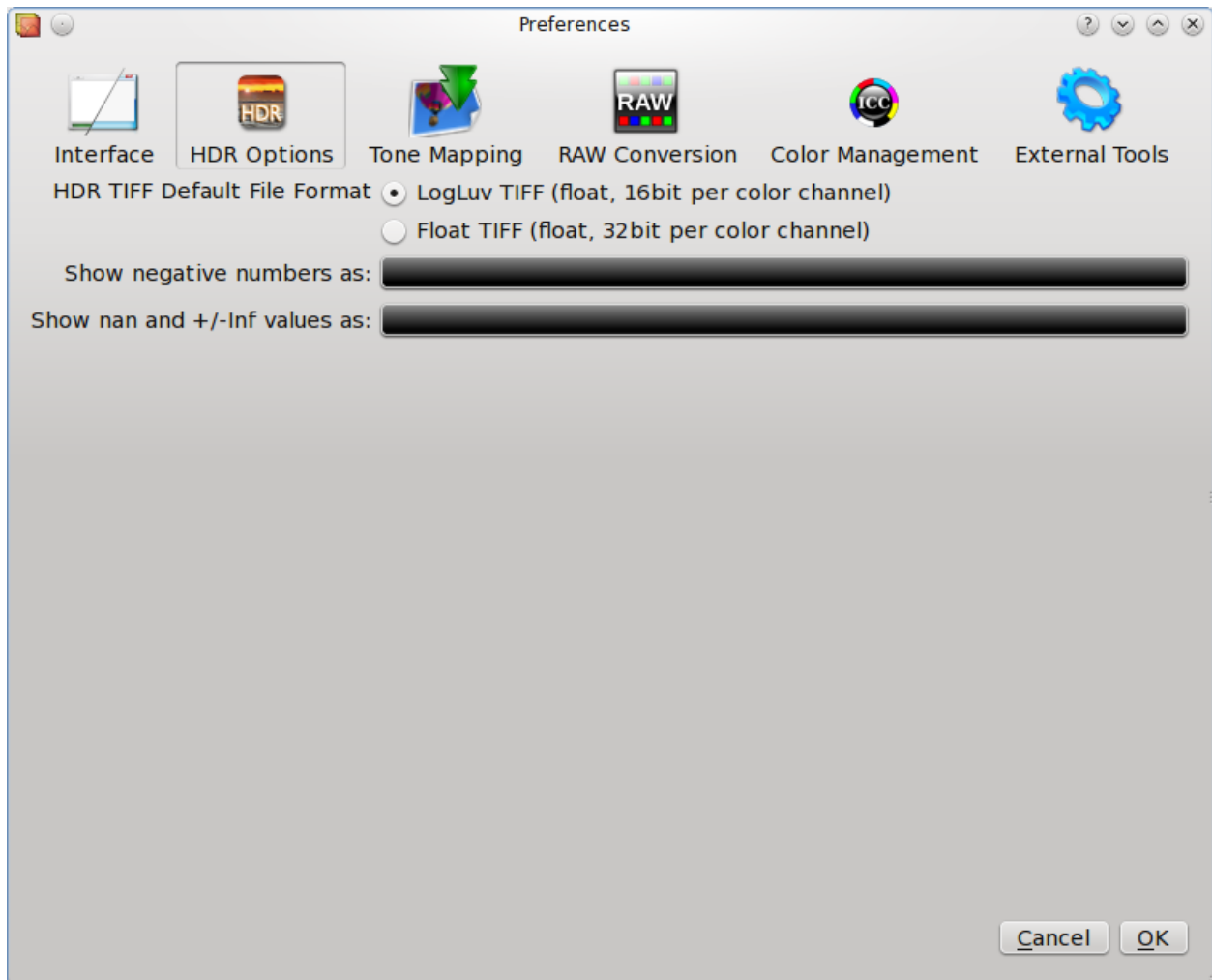
**Export Directory** The folder for the output of the *fast export* feature.

**Format** The default output image format for the *fast export* feature.





## HDR Options



Here you can choose which TIFF-HDR format Luminance HDR will use to save an HDR image.

### HDR TIFF Default File Format

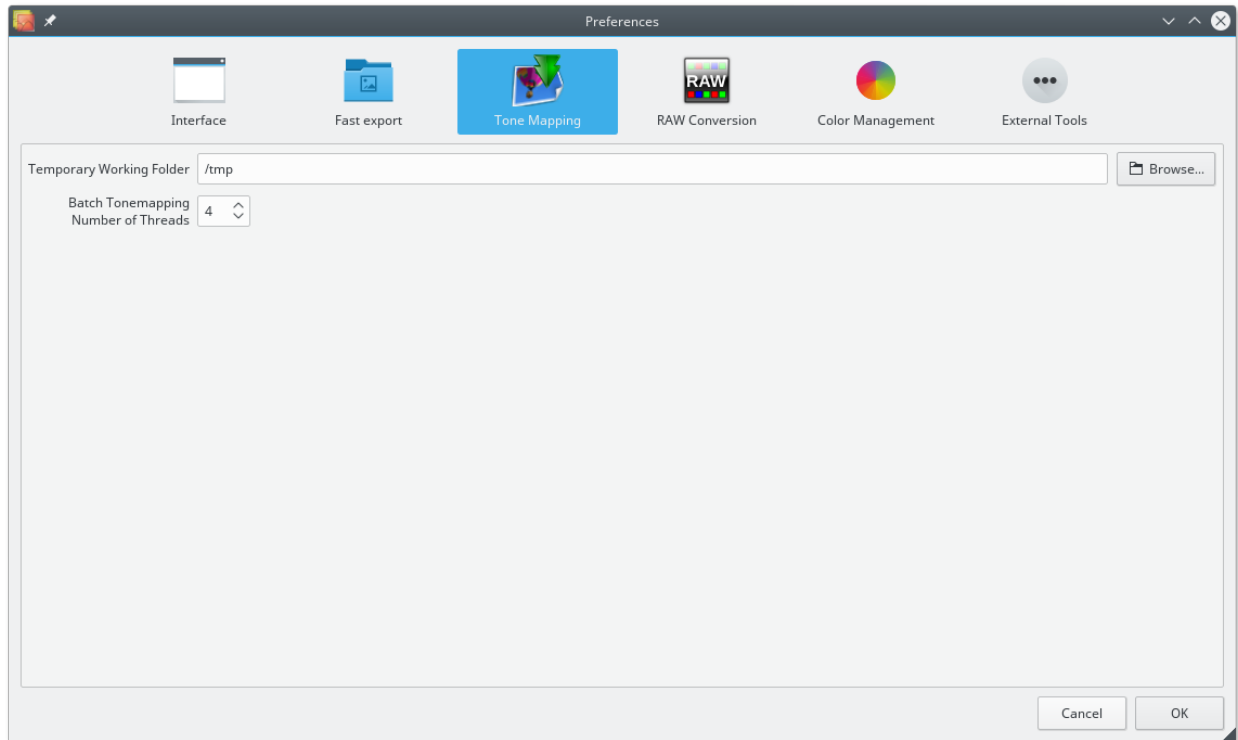
**LogLuv TIFF** The appropriate format for most cases because it stores the hdr data in a reasonable amount of space (i.e. file size). Even if it is a 16 bit format, it can store floating point data.

**Float TIFF** This format has to be used only if you want to load your tiff in another application that cannot open the LogLuv format.

### HDR Visualization

**Show negative numbers as** Color to show negative color values.

**Show nan and +/- inf values as** Color to show nan/inf color values.



## Tone Mapping

**Temporary Working Folder** Luminance HDR needs a directory to save temporary files to. By default home directory is used, but you can specify some other directory.

**Batch Tonemapping Number of threads** How many threads you want to use. Each thread gets allocated to a different CPU or a core of a multicore CPU.

## Raw Conversion

### General

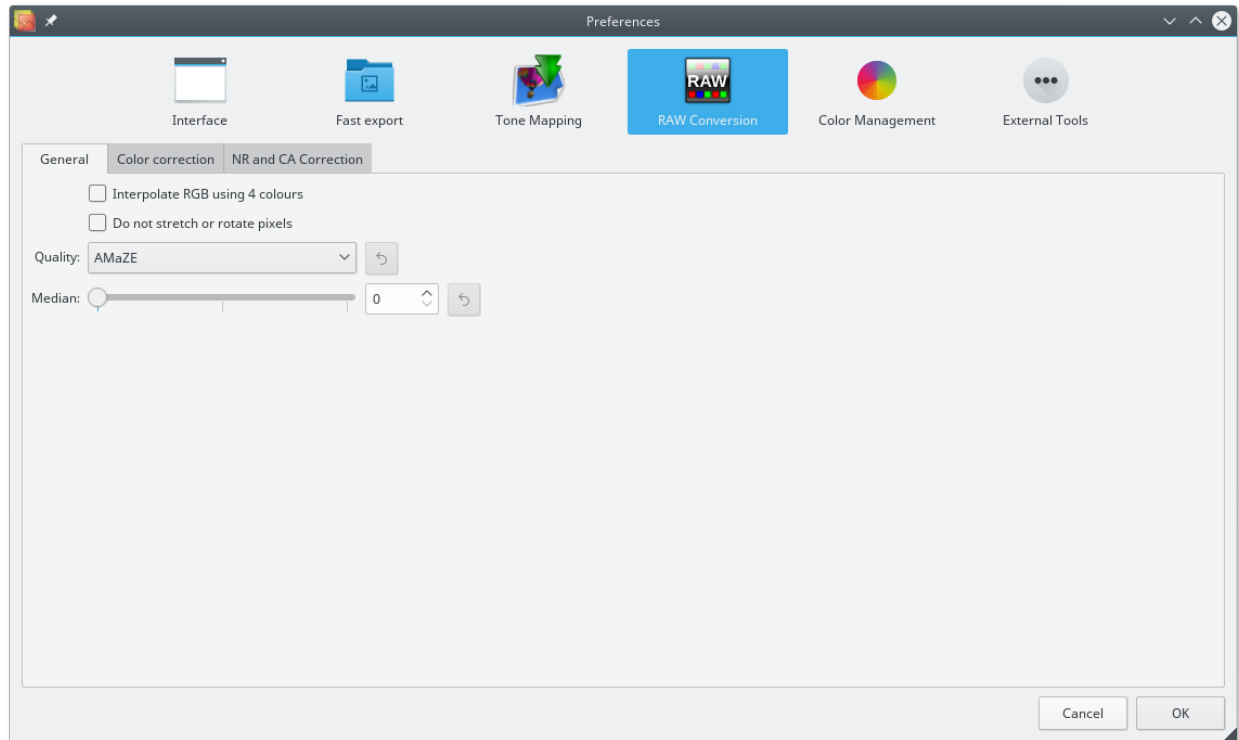
### Demosaicing

A demosaicing algorithm is a digital image process used to reconstruct a full color image from the incomplete color samples output from an image sensor overlaid with a color filter array (CFA).

**Interpolate RGB as four colors** The default is to assume that all green pixels are the same. If even-row green pixels are more sensitive to ultraviolet light than odd-row this difference causes a mesh pattern in the output; using this option solves this problem with minimal loss of detail. To resume, this option blurs the image a little, but it eliminates false 2x2 mesh patterns with VNG quality method or mazes with AHD quality method.

**Do not stretch or rotate pixels** For Fuji Super CCD cameras, show the image tilted 45 degrees. For cameras with non-square pixels, do not stretch the image to its correct aspect ratio. In any case, this option guarantees that each output pixel corresponds to one RAW pixel.

**Quality (interpolation)** Select here the demosaicing RAW images decoding interpolation method. There are 4 methods to demosaicing RAW images:



**Bilinear** use high-speed but low-quality bilinear interpolation (default - for slow computer). In this method, the red value of a non-red pixel is computed as the average of the adjacent red pixels, and similar for blue and green.

**VNG** use Variable Number of Gradients interpolation. This method computes gradients near the pixel of interest and uses the lower gradients (representing smoother and more similar parts of the image) to make an estimate.

**PPG** use Patterned Pixel Grouping interpolation. Pixel Grouping uses assumptions about natural scenery in making estimates. It has fewer color artifacts on natural images than the Variable Number of Gradients method.

**AHD** Adaptive Homogeneity-Directed interpolation. This method selects the direction of interpolation so as to maximize a homogeneity metric, thus typically minimizing color artifacts.

**Median Filter** Set here the passes used by median filter applied after interpolation to Red-Green and Blue-Green channels.

## White Balance

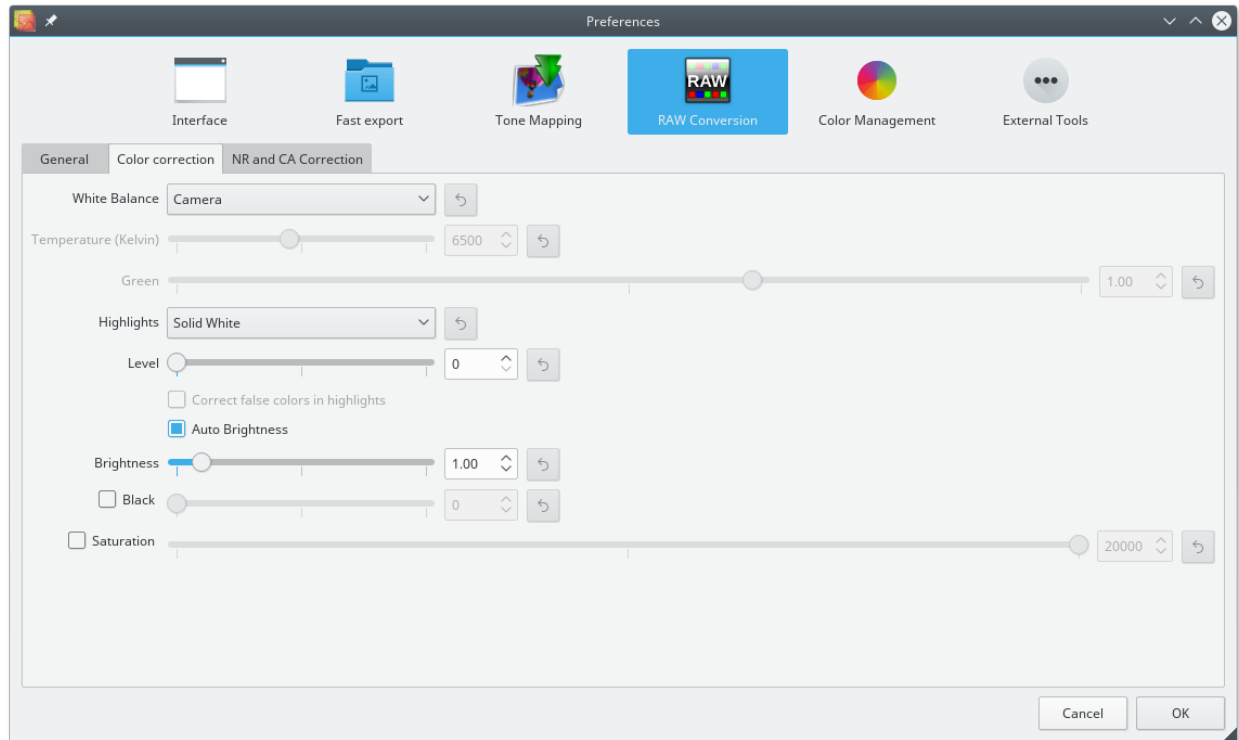
In photography and image processing, color balance is the global adjustment of the intensities of the colors (typically red, green, and blue primary colors). An important goal of this adjustment is to render specific colors — particularly neutral colors — correctly; hence, the general method is sometimes called gray balance, neutral balance, or white balance.

**White Balance Method** Configure the raw white balance.

**Default D65** Use a standard daylight D65 white balance (dcraw defaults)

**Camera** Use the white balance specified by the camera. If not available, reverts to default neutral white balance

**Automatic** Calculates an automatic white balance averaging the entire image



**Manual** Set a custom temperature and green level values

**Temperature** Set here the color temperature in Kelvin.

**Green** Set here the green component to set magenta color cast removal level

**Highlights** Select here the highlight clipping method.

**Solid white** clip all highlights to solid white

**Unclip** leave highlights unclipped in various shades of pink

**Blend** Blend clipped and unclipped values together for a gradual fade to white

**Rebuild** reconstruct highlights using a level value

**Brightness** Specify the brightness level of output image. The default value is 1.0

**Black point value** Specify specific black point value of the output image.

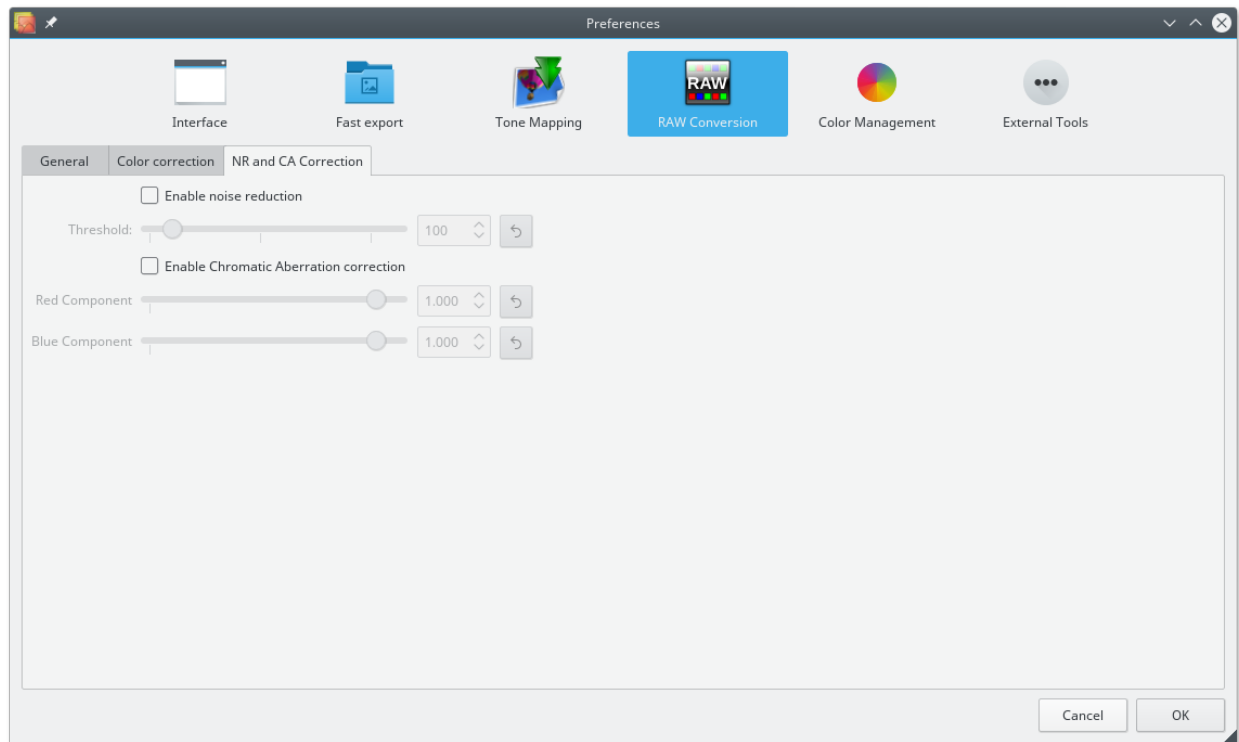
**White point value** Specify specific white point value of the output image.

## Corrections

**Noise reduction** Images taken with both digital cameras and conventional film cameras will pick up noise from a variety of sources. Many further uses of these images require that the noise will be (partially) removed – for aesthetic purposes as in artistic work or marketing, or for practical purposes such as computer vision.

**Threshold** Set here the noise reduction threshold value to use.

**Chromatic Aberration correction** In optics, chromatic aberration (CA, also called achromatism or chromatic distortion) is a type of distortion in which there is a failure of a lens to focus all colors to the same convergence point. It occurs because lenses have a different refractive index for different wavelengths of light (the dispersion of the lens). The refractive index decreases with increasing wavelength. Chromatic aberration manifests itself



as “fringes” of color along boundaries that separate dark and bright parts of the image, because each color in the optical spectrum cannot be focused at a single common point. Since the focal length  $f$  of a lens is dependent on the refractive index  $n$ , different wavelengths of light will be focused on different positions.

**Red multiplier** Set here the magnification factor of the red layer

**Blue multiplier** Set here the magnification factor of the blue layer

## Color Management

**Monitor Profile** Select a color profile associated with your monitor from file. You can find the icc profile in the CD supplied with the monitor or you can download it from the vendor website.

**Camera Profile** Select a color profile for your camera make and model. Leave it empty if your camera does not have a profile (sRGB is used as default).

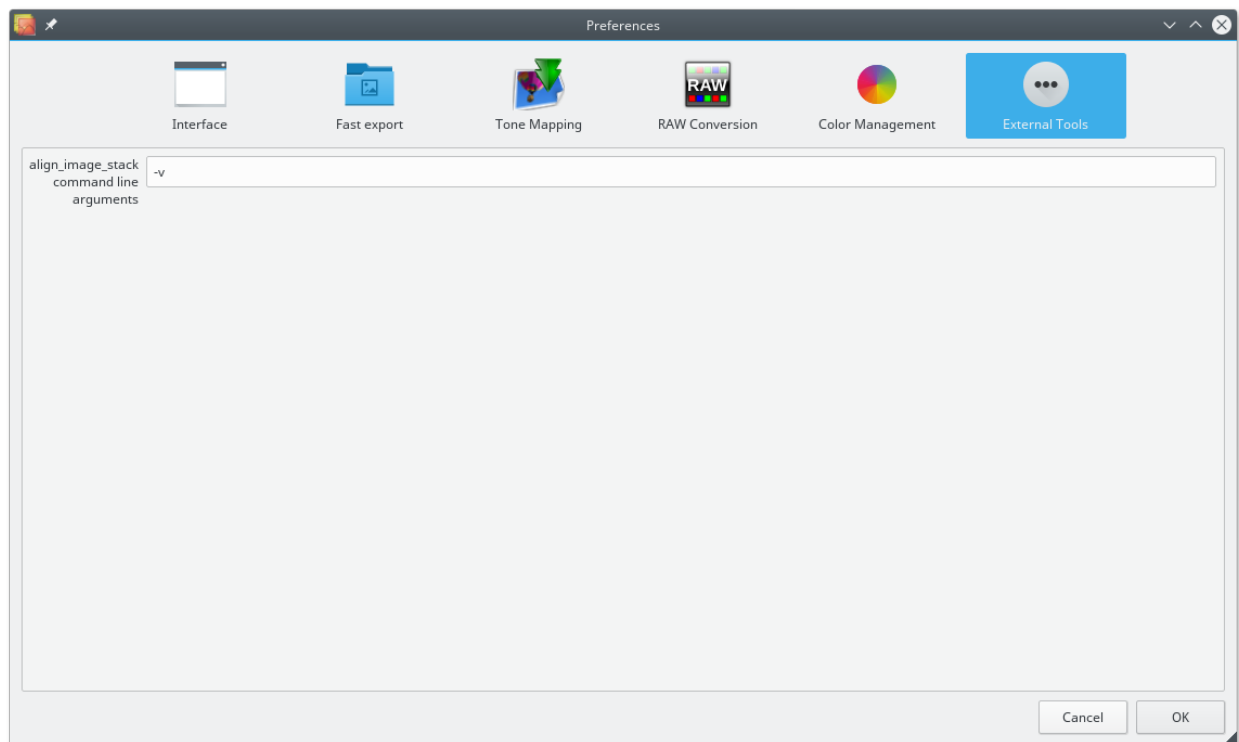
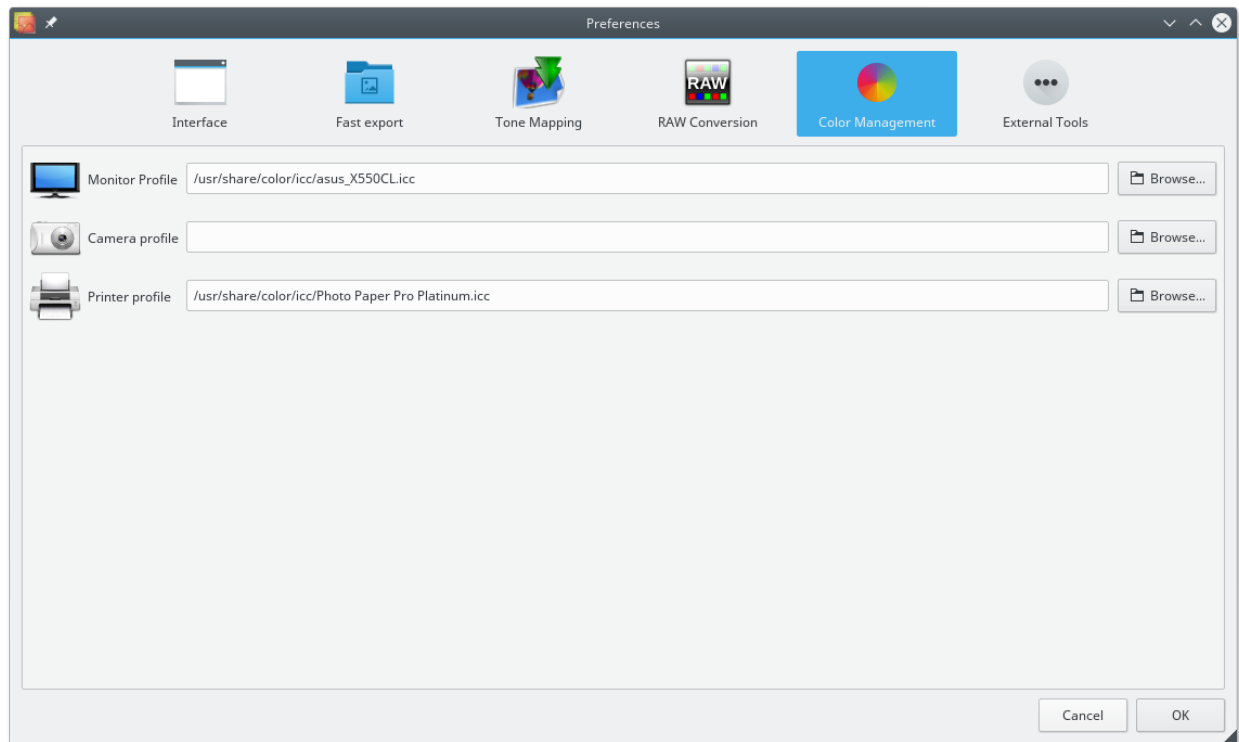
**Printer Profile** Select a color profile for your printer from file. As for the monitor you should be able to find the icc profile in the CD provided or you can download it from the vendor website.

## External Tools

### Alignment options

Luminance HDR can optionally use *align\_image\_stack* command line tool bundled with Hugin, a panorama stitcher, for the alignment of slightly misaligned images. When several differently exposed images are taken in the process of creating a High Dynamic Range image, the images are not perfectly aligned, and the quality of the HDR image created will suffer from this. *align\_image\_stack* uses the tools available to Hugin to optimize the roll, pitch and yaw, as well as some parameters of the lens geometry, to align the images.





*align\_image\_stack* doesn't have a lot of relevant options, so we list them here:

### Options

<b>-a prefix</b>	Output aligned images as prefix_XXXX.tif
<b>-e</b>	Assume input images are full frame fish eye (default: rectilinear)
<b>-t num</b>	Remove all control points with an error higher than num pixels (default: 3)
<b>-f HFOV</b>	Approximate horizontal field of view of input images, use if EXIF info not complete
<b>-m</b>	Optimize field of view for all images, except for first. Useful for aligning focus stacks with slightly different magnification.
<b>-c num</b>	Number of control points (per grid) to create between adjacent images (default: 8)
<b>-l</b>	Assume linear input files
<b>-s scale</b>	Scale down image by 2 <sup>scale</sup> (default: 1 [2x down-sampling])
<b>-g gsize</b>	Break image into a rectangular grid (gsize x gsize) and attempt to find num control points in each section (default: 5 [5x5 grid])
<b>-v</b>	Verbose, print progress messages. Repeat for higher verbosity

By default Luminance HDR just defines the option for verbose printing **-v**.

## Glossary

## References

### DCRaw Reference

Here's an excerpt from [DCRaw's man page](#) (Retrieved: October 30, 2007).

<b>-v</b>	Print verbose messages, not just warnings and errors.
<b>-c</b>	Write decoded images or thumbnails to standard output.
<b>-e</b>	Extract the camera-generated thumbnail, not the raw image. You'll get either a JPEG or a PPM file, depending on the camera.
<b>-z</b>	Change the access and modification times of an AVI, JPEG, TIFF, or raw file to when the photo was taken, assuming that the camera clock was set to Universal Time.
<b>-i</b>	Identify files but don't decode them. Exit status is 0 if dcrw can decode the last file, 1 if it can't. <b>-i -v</b> shows metadata. dcrw cannot decode JPEG files!!
<b>-d</b>	Show the raw data as a grayscale image with no interpolation. Good for photographing black-and-white documents.
<b>-D</b>	Same as <b>-d</b> , but totally raw (no color scaling).
<b>-h</b>	Output a half-size color image. Twice as fast as <b>-q 0</b> .

**-q 0** Use high-speed, low-quality bilinear interpolation.

- q 1** Use Variable Number of Gradients (VNG) interpolation.
- q 2** Use Patterned Pixel Grouping (PPG) interpolation.
- q 3** Use Adaptive Homogeneity-Directed (AHD) interpolation.
  - f** Interpolate RGB as four colors. Use this if the output shows false 2x2 meshes with VNG or mazes with AHD.
  - m number\_of\_passes** After interpolation, clean up color artifacts by repeatedly applying a 3x3 median filter to the R-G and B-G channels.
  - n noise\_threshold** Use wavelets to erase noise while preserving real detail. The best threshold should be somewhere between 100 and 1000.
  - b brightness** By default, dcraw writes 8-bit PGM/PPM/PAM with a BT.709 gamma curve and a 99th-percentile white point. If the result is too light or too dark, -b lets you adjust it. Default is 1.0.
  - 4** Write 16-bit linear pseudo-PGM/PPM/PAM with no gamma curve, no white point, and no -b option.
  - T** Write TIFF output (with metadata) instead of PGM/PPM/PAM.
  - k black** Set the black point. Default depends on the camera.
- K darkframe.pgm** Subtract a dark frame from the raw data. To generate a dark frame, shoot a raw photo with no light and do dcraw -D -4 -j -t 0.
- w** Use the white balance specified by the camera. If this is not found, print a warning and use another method.
- a** Calculate the white balance by averaging the entire image.
- A left top width height** Calculate the white balance by averaging a rectangular area. First do dcraw -j -t 0 and select an area of neutral grey color.
- r mul0 mul1 mul2 mul3** Specify your own raw white balance. These multipliers can be cut and pasted from the output of dcraw -v.
- no white balance option** Use a fixed white balance based on a color chart illuminated with a standard D65 lamp.
- +M or -M** Use (or don't use) any color matrix from the camera metadata. The default is +M if -w is set, -M otherwise. This option only affects Olympus, Leaf, and Phase One cameras.
- C red\_mag blue\_mag** Enlarge the raw red and blue layers by the given factors, typically 0.999 to 1.001, to correct chromatic aberration.
- H 0** Clip all highlights to solid white (default).
- H 1** Leave highlights unclipped in various shades of pink.
- H 2** Blend clipped and unclipped values together for a gradual fade to white.
- H 3-9** Reconstruct highlights. Low numbers favor whites; high numbers favor colors. Try -H 5 as a compromise. If that's not good enough, do -H 9, cut out the non-white highlights, and paste them into an image generated with -H 3.
- o [0-5]** Select the output colorspace when the -p option is not used:
  - 0 Raw color (unique to each camera) 1 sRGB D65 (default) 2 Adobe RGB (1998) D65 3 Wide Gamut RGB D65 4 Kodak ProPhoto RGB D65 5 XYZ
- p camera.icm [ -o output.icm ]** Use ICC profiles to define the camera's raw colorspace and the desired output colorspace (sRGB by default).

- p embed** Use the ICC profile embedded in the raw photo.
- t [0-7,90,180,270]** Flip the output image. By default, dcraw applies the flip specified by the camera. -t 0 disables all flipping.
- s [0..N-1] or -s all** If a file contains N raw images, choose one or “all” to decode. For example, Fuji Super CCD SR cameras generate a second image underexposed four stops to show detail in the highlights.
- j** For Fuji Super CCD cameras, show the image tilted 45 degrees. For cameras with non-square pixels, do not stretch the image to its correct aspect ratio. In any case, this option guarantees that each output pixel corresponds to one raw pixel.

## Terms

**Anti-ghosting** The HDR creation process involves merging a stack of images. An object changing position in the image set creates a strange effect in which the object is partially visible (like a ghost) in the final HDR image. This problem can be corrected by automatic or manual procedures.

**HDR** Stands for “High Dynamic Range”. An HDR image is an image which presents more than 8 bit per color channel. Most CRTs, LCDs and printers only have a limited dynamic range, and can display only LDR images (see below). Thus various methods of “converting” HDR images into a viewable format have been developed, generally called “tone mapping”.

**LDR** Stands for “Low Dynamic Range”. The most common image formats, such as JPEG, PNG, GIF, ... have 8 bits per color channel, LDR is just another umbrella definition.

**Raw** Another umbrella definition for several (minimally processed) image formats. Raw files can have 12 or 14 bits per color channel, but noise usually cuts down the available dynamic range to something like 1000:1, roughly 10 bits. For all intents and purposes they are HDR files.

**TMO** Shorthand for “Tone Mapping Operator”.

**Tone mapping** A method of converting an HDR image into a LDR image. Various algorithms exist for this purpose, and in this context they are also known as “tone mapping operators”, or in this manual simply as “operators”. You can choose an operator from a list in the top of tone mapping options sidebar.

## About LuminanceHDR

### News

#### What’s New

The version you are looking at contains a number of improvements over the previous one, such as:

- Two brand new tonemapping operators: ferradans and mai
- Optional automatic adjustment of LDRs levels
- Greater EV values range in HDR Creation Wizard
- Restore load/save curves in HDR Creation Wizard
- Fix various crashes
- Export to HTML (Create a webpage with embedded HDR viewer)
- Better printing support and print preview in Help Browser
- A new “Dark Theme” (beta) and native platform icons support

- Switch UI full screen (F11), show LDRs and HDRs images full screen (F10)
- Portuguese (Brazilian) translation
- Other small improvements and bugfixing as usual

Previous release features and fixes:

- Color Management Support (read/write ICC profiles, soft proofing, gamut check)
- Selection of output image size and quality in Batch TM
- Fixed memory leak in Batch HDR
- Fattal and Reinhard02 now thread safe (speed improvement with multicore processors)
- Windows: Better OS Integration

Due to lack of active developers please don't expect much other than cleanups and documentation updates in the coming releases.

## Resources

TODO

## Contributing

There are several ways to help us making this project better:

### Testing and reporting

#### Testing

Testing is what helps making applications rock stable. Since we are a community project, we rely on you, yes — you, our dear users.

We don't really encourage you to compile the most unstable cutting edge source code, but in case you find some bugs in the latest released version, do not hesitate to point them out to us.

## Crashes

So you found a reproducible way to crash Luminance HDR. If you are on Linux, please use application called `gdb` to create a report which in programmers lingo is called a *backtrace*. Here is how you do it:

1. Install `gdb` via package manager
2. Open terminal
3. `$ gdb luminance`
4. `gdb`'s console appears
5. `> run`
6. Luminance HDR starts, a little slower than usually
7. Reproduce the crash
8. Go back to terminal

9. > bt
10. Copy the output using mouse and paste it somewhere
11. > quit

If you are on Windows or Mac or simply do not have time to fiddle with gdb, at least own up and tell us exactly what you did.

### Reporting bugs and requesting features

You can submit bugs reports to our [bugtracker](#).

A good, useful bug report contains:

1. List of actions that led to a bug
2. Backtrace, if the application crashed (see above)
3. Information on your system

Typical information on your system we would appreciate:

- Linux: name and version of distribution, version of Qt, version of Luminance HDR.
- Mac OS X: version of Mac OS X, version of Luminance HDR.
- Windows: version of Windows, version of Luminance HDR.

If you want to request a new feature, please also use the feature request tracker.

### Documentation

TODO

### Translations

The documentation is inside help directory and consists of menu.xml file that defines table of contents, HTML files with text and illustrations in PNG or JPEG files.

Every translation is kept in its own directory named with two-letter language code like ru for Russian or es for Spanish. So download source code, unpack it and create a copy of help/en directory in help directory.

Start translating. It's best to translate table of contents first and proceed with actual content later. To translate table of contents open menu.xml file in your editor of choice and translate values of every text attribute. E.g. for <area text="Setting up Luminance HDR" file="prefs.html"> translatable text will be "Setting up Luminance HDR".

To test your translation open a terminal window, go to the top level directory with source code, and run `sudo make install` to reinstall Luminance HDR. All available translations will be automatically copied to the right place, and you will have to restart Luminance HDR to let it pick the added translation. However, as you progress with your translation, you only need to restart help browser to see changes in table of contents and you don't even need to restart the help browser to see changes in separate chapters — just click on some other chapter and go back again.

If user interface is not localized, you might want to do it before translating docs. Some users might complain and tell you that user interface in English is a de-facto standard and thus localized documentation should refer only to English UI. But this is just because they have grown up to use unlocalized software, so don't you worry.

The English (and Russian) translations have screenshots with Dust theme for both GTK+ and Metacity (and GTK+ engine for Qt), and Droid Sans 9pt font. You don't have to try to reproduce that, but please be visually consistent across your translation.

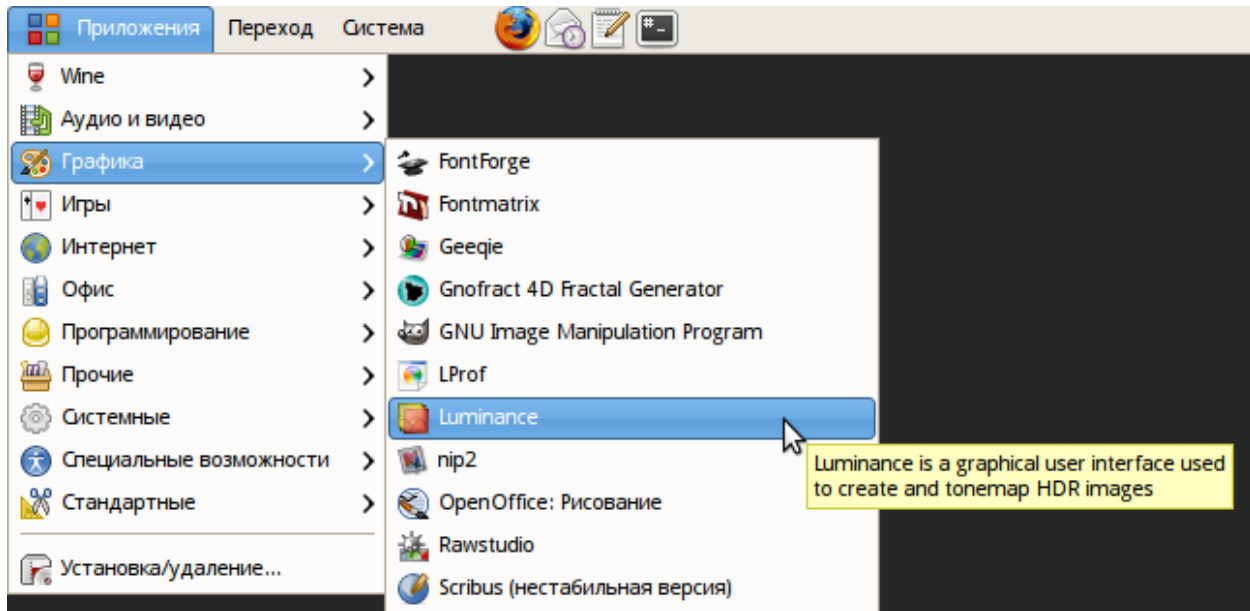
Please keep all of your illustrations below 800 pixels on the longer side. The reason is: when an image doesn't fit help browser's window, a nasty horizontal scrollbar appears. To get rid of it you need to grow width of the window, and that means that text will reflow and there will be too long barely readable lines of text.

When you are done, archive help/LANGUAGE directory with your translation and send it to us.

## Translating

### Translating desktop entry

On Linux systems .desktop files are used to build system menus that list applications available to users. Here is what it looks like in GNOME desktop environment:



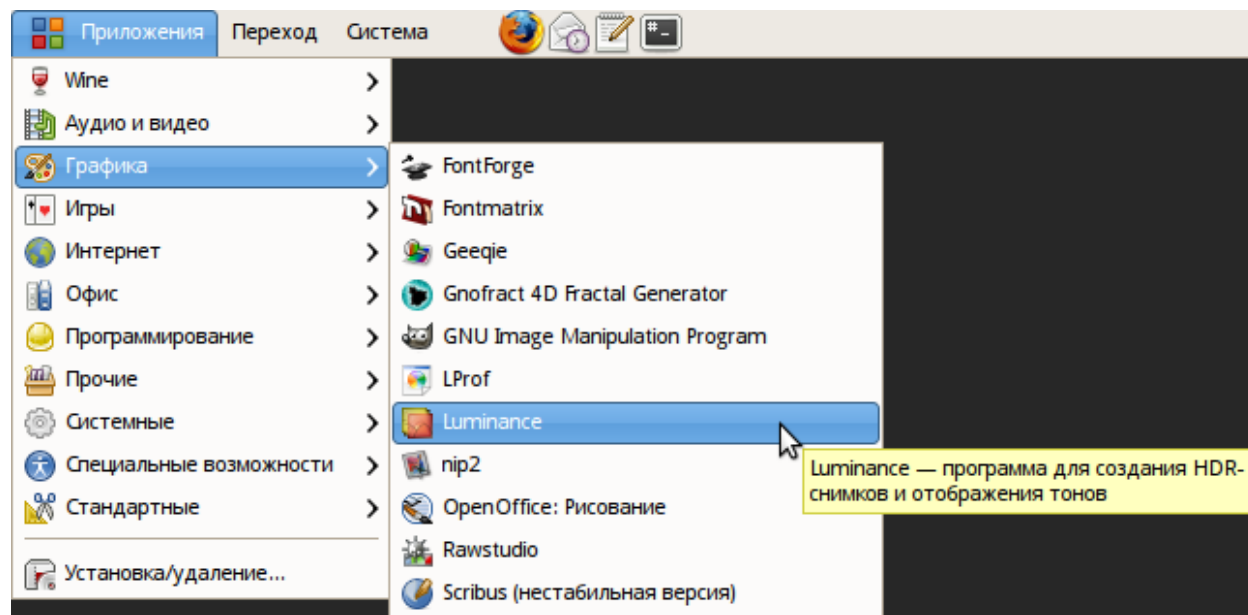
Luminance HDR ships with such a file as well. It is located in root directory with source code and gets installed to `/usr/share/applications` or `/usr/local/share/applications`, depending on your preferences.

To get a localized menu entry you need to do a very simple thing:

1. Open this file in your preferred text editor and make sure you opened it as a UTF-8 encoded text file.
2. Create a new entry which looks like “Comment[LANG]=Create and tonemap HDR images”, where LANG is a two-letter code for your language (as referenced by *ISO-639-1* <[https://en.wikipedia.org/wiki/List\\_of\\_ISO\\_639-1\\_codes](https://en.wikipedia.org/wiki/List_of_ISO_639-1_codes)>) and everything after “=” is translated.
3. Create a new entry which looks like “GenericName[LANG]=HDR imaging”.
4. Save.
5. Test by running ‘`sudo make install`’ (if Luminance HDR is already installed or just ‘`sudo cp luminance.desktop /usr/share/applications/`’ and look in the menu.

You should see something like this:

Send the updated file to us.



## Translating user interface

The very first thing you need to translate Luminance HDR into your native language is to get source code from the current development branch. To do this, you need a Subversion client (svn being the regular command line client). Then type:

```
svn co https://qtpfsgui.svn.sourceforge.net/svnroot/qtpfsgui/trunk/qtpfsgui qtpfsgui
```

somewhere in your home directory to fetch source code, so that you always have access to it.

Then you will need to install Qt development package that contains Linguist – the application to assist you with translating. On Linux start your package manager and look for a package named something like qt4-dev, install it.

The next steps are as follows:

1. Go to the top level directory of Luminance HDR's source code.
2. Open the file called project.pro.
3. Find a section that starts with "TRANSLATIONS =" and add a new line that looks like "i18n/lang\_LOCALE.ts", where LOCALE is a two-letter code for your language (as referenced by ISO-639-1), lowercase, e.g. pt for Portuguese.
4. Save the project.pro file.
5. Open console in that directory and run "\$ lupdate-qt4 project.pro" command. This will create a new translation file (and update existing translation files).
6. Open your lang\_LOCALE.ts file in Linguist and start translating.
7. To test your translation, use File > Release menu item in Linguist to create a binary version of translation and then run sudo make install from console to quickly install created translation in the right place. When your work is done, compress this lang\_LOCALE.ts file to a ZIP, GZ or BZ2 archive and submit it via patches tracker.

Here are some tips to help you make translation better.

Translating Luminance HDR takes a while, so it's best to translate those parts of user interface that you use most of the time. This will give you a false, but useful feeling of accomplishment and motivation to finish the whole work.



Test your translation as frequently as possible. This is especially important for dialogs that you rarely use.

Make sure you find a good balance between short and easy to understand phrases and words. English language is known to have relatively shorter words, so in most cases your translation will make user interface a bit larger. But if you start using abbreviations or shorter synonyms that don't quite fit the context, users won't appreciate that either.

Some translatable messages use variables like %1. Those are substituted by some values. For example, in "Using %1 thread(s)" (Batch Tonemapping dialog) this variable is substituted with amount of threads used to process HDR images into LDR images. When you type these variables manually, you can make a mistake and the trick with a variable won't work. So it's better to paste original text to translation entry field by pressing Ctrl+B in Linguist and then replace this original text with translation, leaving all present variables intact.

## **Programming**

Documentation for potential programmers isn't ready at this time. Please come back later or ask your questions.

Pfstmo which we use as a backend will become a library eventually, so we shall have to introduce some changes. The other plan is to start using some solid graphics lib like GEGL or VIPS to deal with images larger than RAM in a sane way. If you want to help the Luminance HDR team with that, do tell us.

## **Donating**

Luminance HDR is open-source software and its development required hundreds of hours of work.

If you like it, you're using it in your work and you would like to see it constantly improved, please support its authors by making a donation.

Would you like to make a donation for Luminance HDR now? [Donate](#)



## CHAPTER 3

---

### Indices and Tables

---

- `genindex`
- `search`



## A

Anti-ghosting, [40](#)

## H

HDR, [40](#)

## L

LDR, [40](#)

## R

Raw, [40](#)

## T

TMO, [40](#)

Tone mapping, [40](#)