Enblend

Combining Multiple Images
with Enblend version 4.1.4, 7 August 2015

Andrew Mihal

This manual is for Enblend (version 4.1.4, 7 August 2015), a tool for compositing images in such a way that the seam between the images is invisible, or at least very difficult to see.

Table of Contents

List of Tables

List of Figures

1 Overview

Enblend overlays multiple images using the Burt-Adelson multiresolution spline algorithm.[1] This technique tries to make the seams between the input images invisible. The basic idea is that image features should be blended across a transition zone proportional in size to the spatial frequency of the features. For example, objects like trees and windowpanes have rapid changes in color. By blending these features in a narrow zone, you will not be able to see the seam because the eye already expects to see color changes at the edge of these features. Clouds and sky are the opposite. These features have to be blended across a wide transition zone because any sudden change in color will be immediately noticeable.

Enblend expects each input file to have an alpha channel. The alpha channel should indicate the region of the file that has valid image data. Enblend compares the alpha regions in the input files to find the areas where images overlap. Alpha channels can be used to indicate to Enblend that certain portions of an input image should not contribute to the final image.

Enblend does *not* align images. Use a tool such as `hugin` or PanoTools to do this. The TIFF files produced by these programs are exactly what Enblend is designed to work with. Sometimes these GUIs allow you to select feathering for the edges of your images. This treatment is detrimental to Enblend. Turn off feathering by deselecting it or setting the feather width to zero.

Enblend blends the images in the order they are specified on the command line. You should order your images according to the way that they overlap, for example from left-to-right across the panorama. If you are making a multi-row panorama, we recommend blending each horizontal row individually, and then running Enblend a last time to blend all of the rows together vertically.

Enblend reads all layers of multi-layer images, like, for example, multi-directory TIFF images[2]. The input images are processed in the order they appear on the command line. Multi-layer images are processed from the first layer to the last before Enblend considers the next image on the command line.

Find out more about Enblend on its SourceForge web page.

[1] Peter J. Burt and Edward H. Adelson, “A Multiresolution Spline With Application to Image Mosaics”, ACM Transactions on Graphics, Vol. 2, No. 4, October 1983, pages 217–236.

[2] Use utilities like, e.g., `tiffcopy` and `tiffsplit` of LibTIFF to manipulate multi-directory TIFF images. See Chapter 8 [Helpful Programs], page 32.

2 Workflow

Enblend is a part of a chain of tools to assemble images. It combines a series of pictures taken at the same location but in different directions.

2.1 Standard Workflow

Figure 2.1 shows where Enblend and Enfuse sit in the tool chain of the standard workflow.

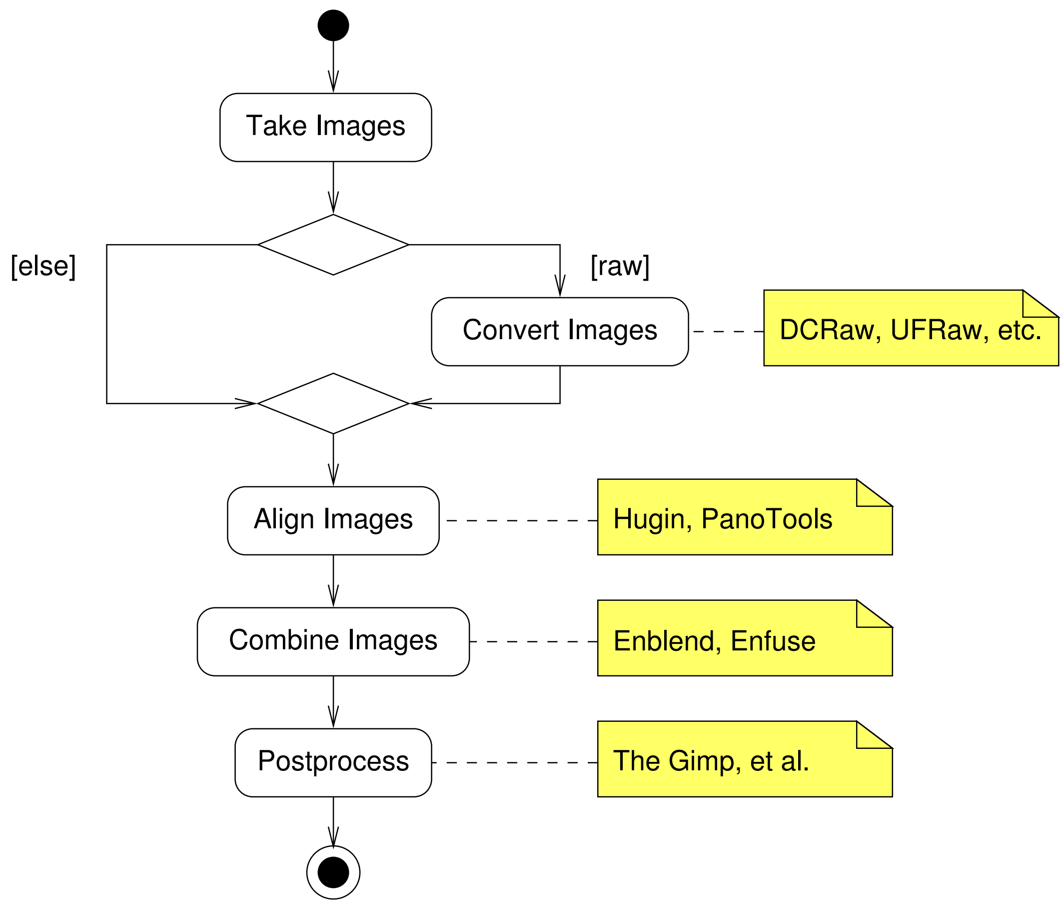

Figure 2.1: Photographic workflow with Enblend and Enfuse.

Take Images

Take *multiple* images to form a panorama, an exposure series, a focus stack, etc.

There is one exception with Enfuse when a single raw image is converted multiple times to get several – typically differently “exposed” – images.

Exemplary Benefits

- Many pictures taken from the same vantage point but showing different viewing directions. – Panorama

- Pictures of the same subject exposed with different shutter speeds. – Exposure series
- Images of the same subject focussed at differing distances. – Focus stack

Remaining Problem: The "overlayed" images may not fit together, that is the overlay regions may not match exactly.

Convert Images

Convert the raw data exploiting the full dynamic range of the camera and capitalize on a high-quality conversion.

Align Images

Align the images so as to make them match as well as possible.

Again there is one exception and this is when images naturally align. For example, a series of images taken from a rock solid tripod with a cable release without touching the camera, or images taken with a shift lens, can align without further user intervention.

This step submits the images to affine transformations. If necessary, it rectifies the lens' distortions (e.g. barrel or pincushion), too. Sometimes even luminance or color differences between pairs of overlaying images are corrected ("photometric alignment").

Benefit: The overlay areas of images match as closely as possible given the quality if the input images and the lens model used in the transformation.

Remaining Problem: The images may still not align perfectly, for example, because of parallax errors, or blur produced by camera shake.

Combine Images

Enblend and Enfuse combine the aligned images into one.

Benefit: The overlay areas become imperceptible for all but the most mal-aligned images.

Remaining Problem: Enblend and Enfuse write images with an alpha channel. (For more information on alpha channels see Chapter 6 [Understanding Masks], page 28.) Furthermore, the final image rarely is rectangular.

Postprocess

Postprocess the combined image with your favorite tool. Often the user will want to crop the image and simultaneously throw away the alpha channel.

View

Print

Enjoy

2.2 External Mask Manipulation

In the usual workflow Enblend and Enfuse generate the blending and fusing masks according to the command-line options and the input images and then they immediately use these masks for blending or fusing the output image.

Sometimes more control over the masks is needed or wanted. To this end, both applications provide the option pair `--load-masks` and `--save-masks`. See Chapter 3 [Invocation], page 5, for detailed explanations of both options. With the help of these options the processing can be broken up into two steps:

Save masks with `--save-masks`.
: Generate masks and save them into image files.

  Avoid option `--output` unless the blended or fused image at this point is necessary.

Load masks with `--load-masks`.
: Load masks from files and then blend or fuse the final image with the help of the loaded masks.

In between these two steps the user may apply whatever transformation to the mask files, as long as their geometries and offsets remain the same. Thus the “Combine Images” box of Figure 2.1 becomes three activities as is depicted in Figure 2.2.

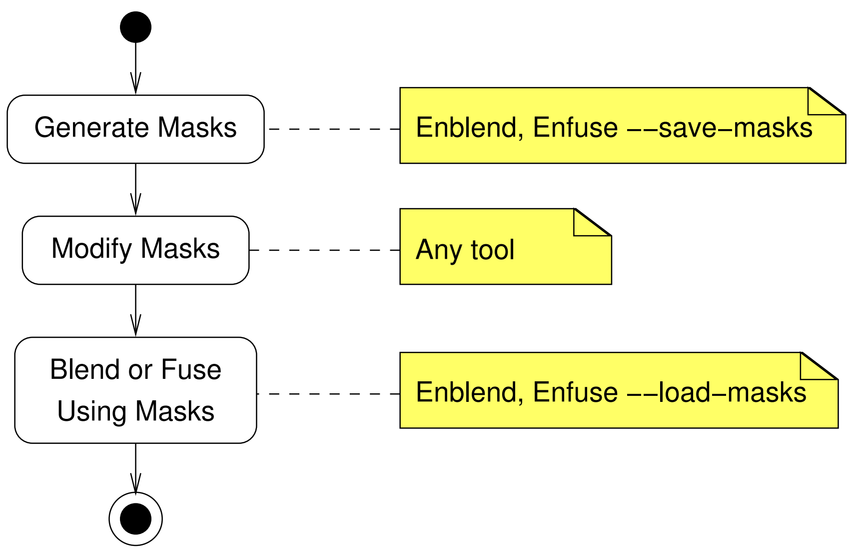

Figure 2.2: Workflow for externally modified masks.

To further optimize this kind of workflow, both Enblend and Enfuse stop after mask generation if option `--save-masks` is given, but *no output file* is specified with the `--output` option. This way the time for pyramid generation, blending, fusing, and writing the final image to disk is saved, as well as no output image gets generated.

Note that options `--save-masks` and `--load-masks` cannot be used simultaneously.

3 Invocation

`enblend` [*OPTIONS*] [`--output=`*IMAGE*] *INPUT*...

Assemble the sequence of images *INPUT*... into a single *IMAGE*.

Input images are either specified literally or via so-called response files (see below). The latter are an alternative to specifying image filenames on the command line.

3.1 Image Requirements

All input images must comply with the following requirements.

- Parts of the images overlap.
- Each image has an alpha channel also called "mask".
- The images agree on their number of channels:
  - one plus alpha or
  - three plus alpha.

  This is, either all images are black-and-white (one channel and alpha channel) or all are RGB-color images (three channels and alpha channel).
- The images agree on their number of bits-per-channel, this is, their "depth":
  - `UINT8`,
  - `UINT16`,
  - `FLOAT`,
  - etc.

  See option `--depth` below for an explanation of different (output) depths.
- Enblend understands the images' filename extensions as well as their file formats.

  You can check the supported extensions and formats by calling Enblend with the option pair `--version --verbose` and scan the output for '`Supported image formats`' or '`Supported file extensions`'.

Moreover, there are some "good practices", which are not enforced by the application, but almost certainly deliver superior results.

- Either all files lack an ICC profile, or all images are supplied with the *same* ICC profile.
- If the images' meta-data contains resolution information ("DPI"), it is the same for all pictures.

3.2 Response Files

A response file contains names of images or other response filenames. Introduce response file names with an at-character ('`@`').

Enblend and Enfuse process the list *INPUT* strictly from left to right, expanding response files in depth-first order. (Multi-layer files are processed from first layer to the last.) The following examples only show Enblend, but Enfuse works exactly the same.

Solely image filenames.

Example:

```
enblend image-1.tif image-2.tif image-3.tif
```

The ultimate order in which the images are processed is: `image-1.tif`, `image-2.tif`, `image-3.tif`.

Single response file.

Example:

```
enblend @list
```

where file `list` contains

```
img1.exr
img2.exr
img3.exr
img4.exr
```

Ultimate order: `img1.exr`, `img2.exr`, `img3.exr`, `img4.exr`.

Mixed literal names and response files.

Example:

```
enblend @master.list image-09.png image-10.png
```

where file `master.list` comprises of

```
image-01.png
@first.list
image-04.png
@second.list
image-08.png
```

`first.list` is

```
image-02.png
image-03.png
```

and `second.list` contains

```
image-05.png
image-06.png
image-07.png
```

Ultimate order: `image-01.png`, `image-02.png`, `image-03.png`, `image-04.png`, `image-05.png`, `image-06.png`, `image-07.png`, `image-08.png`, `image-09.png`, `image-10.png`,

3.2.1 Response File Format

Response files contain one filename per line. Blank lines or lines beginning with a sharp sign (‘`#`’) are ignored; the latter can serve as comments. Filenames that begin with an at-character (‘`@`’) denote other response files. Table 3.1 states a formal grammar of response files in EBNF.

response-file	::=	*line**
line	::=	(*comment* | *file-spec*) ['\r'] '\n'
comment	::=	*space** '#' *text*
file-spec	::=	*space** '@' *filename space**
space	::=	' ' | '\t'

where *text* is an arbitrary string and *filename* is any filename.

Table 3.1: EBNF definition of the grammar of response files.

In a response file relative filenames are used relative the response file itself, not relative to the current-working directory of the application.

The above grammar might unpleasantly surprise the user in the some ways.

Whitespace trimmed at both line ends
: For convenience, whitespace at the beginning and at the end of each line is ignored. However, this implies that response files cannot represent filenames that start or end with whitespace, as there is no quoting syntax. Filenames with embedded whitespace cause no problems, though.

Only whole-line comments
: Comments in response files always occupy a complete line. There are no "line-ending comments". Thus, in

```
# exposure series
img-0.33ev.tif # "middle" EV
img-1.33ev.tif
img+0.67ev.tif
```

only the first line contains a comment, whereas the second line includes none. Rather, it refers to a file called '`img-0.33ev.tif # "middle" EV`'.

Image filenames cannot start with '@'
: An at-sign invariably introduces a response file, even if the filename's extension hints towards an image.

If Enblend or Enfuse do not recognize a response file, they will skip the file and issue a warning. To force a file being recognized as a response file add one of the following syntactic comments to the *first* line of the file.

```
response-file: true
enblend-response-file: true
enfuse-response-file: true
```

Finally, here is an example of a valid response file.

```
# 4\pi panorama!

# These pictures were taken with the panorama head.
@round-shots.list

# Freehand sky shot.
zenith.tif

# "Legs, will you go away?" images.
nadir-2.tif
nadir-5.tif
nadir.tif
```

3.2.2 Syntactic Comments

Comments that follow the format described in Table 3.2 are treated as instructions how to interpret the rest of the response file. A syntactic comment is effective immediately and its effect persists to the end of the response file, unless another syntactic comment undoes it.

syntactic-comment	::=	*space** '`#`' *space** *key* *space** '`:`' *space** *value*
key	::=	('`A`' .. '`Z`' `|` '`a`' .. '`z`' `|` '`-`')+

where *value* is an arbitrary string.

Table 3.2: EBNF definition of the grammar of syntactic comments in response files.

Unknown syntactic comments are silently ignored.

3.2.3 Globbing Algorithms

The three equivalent syntactic keys

- `glob`,
- `globbing`, or
- `filename-globbing`

control the algorithm that Enblend or Enfuse use to glob filenames in response files.

All versions of Enblend and Enfuse support at least two algorithms: `literal`, which is the default, and `wildcard`. See Table 3.3 for a list of all possible globbing algorithms. To find out about the algorithms in your version of Enblend or Enfuse team up the options `--version` and `--verbose`.

`literal` Do not glob. Interpret all filenames in response files as literals. This is the default.

Please keep in mind that whitespace at both ends of a line in a response file *always* gets discarded.

`wildcard` Glob using the wildcard characters '`?`', '`*`', '`[`', and '`]`'.

The W*N32 implementation only globs the filename part of a path, whereas all other implementations perform wildcard expansion in *all* path components. Also see glob(7).

`none` Alias for `literal`.

`shell` The `shell` globbing algorithm works as `literal` does. In addition, it interprets the wildcard characters '`{`', '`}`', and '`~`'. This makes the expansion process behave more like common UN*X shells.

`sh` Alias for `shell`.

Table 3.3: Globbing algorithms for the use in response files

Example:

```
# Horizontal panorama
# 15 images

# filename-globbing: wildcard

image_000[0-9].tif
image_001[0-4].tif
```

3.2.4 Default Layer Selection

The key `layer-selector` provides the same functionality as does the command-line option `--layer-selector`, but on a per response-file basis. See Section 3.3 [Common Options], page 9.

This syntactic comment affects the layer selection of all images listed after it including those in included response files until another `layer-selector` overrides it.

3.3 Common Options

Common options control some overall features of Enblend.

Enblend accepts arguments to any option in uppercase as well as in lowercase letters. For example, '`deflate`', '`Deflate`' and '`DEFLATE`' as arguments to the `--compression` option described below, all instruct Enblend to use the Deflate compression scheme. This manual denotes all arguments in lowercase for consistency.

`-a` Pre-assemble non-overlapping images before each blending iteration.

This overrides the default behavior which is to blend the images sequentially in the order given on the command line. Enblend will use fewer blending iterations, but it will do more work in each iteration.

`--compression=`*`COMPRESSION`*
: Write a compressed output file.

Depending on the output file format, Enblend accepts different values for *COMPRESSION*.

JPEG format.
: The compression either is a literal integer or a keyword-option combination.

`LEVEL`
: Set JPEG quality *LEVEL*, where *LEVEL* is an integer that ranges from 0–100.

`jpeg[:`*`LEVEL`*`]`
: Same as above; without the optional argument just switch on (standard) JPEG compression.

`jpeg-arith[:`*`LEVEL`*`]`
: Switch on arithmetic JPEG compression. With optional argument set the arithmetic compression *LEVEL*, where *LEVEL* is an integer that ranges from 0–100.

TIF format.
: Here, *COMPRESSION* is one of the keywords:

`none`
: Do not compress. This is the default.

`deflate`
: Use the Deflate compression scheme also called ZIP-in-TIFF. Deflate is a lossless data compression algorithm that uses a combination of the LZ77 algorithm and Huffman coding.

`jpeg[:`*`LEVEL`*`]`
: Use JPEG compression. With optional argument set the compression *LEVEL*, where *LEVEL* is an integer that ranges from 0–100.

`lzw`
: Use Lempel-Ziv-Welch (LZW) adaptive compression scheme. LZW compression is lossless.

`packbits`
: Use PackBits compression scheme. PackBits is a particular variant of run-length compression; it is lossless.

Any other format.
: Other formats do not accept a *COMPRESSION* setting.

However, VIGRA automatically compresses `png`-files with the Deflate method.

`--layer-selector=`*`ALGORITHM`*
: Override the standard layer selector algorithm, which is '`all-layers`'.

This version of Enblend offers the following algorithms:

`all-layers`
Select all layers in all images.

`first-layer`
Select only first layer in each multi-layer image. For single-layer images this is the same as '`all-layers`'.

`largest-layer`
Select largest layer in each multi-layer image, where the "largeness", this is the size is defined by the product of the layer width and its height. The channel width of the layer is ignored. For single-layer images this is the same as '`all-layers`'.

`no-layer` Do not select any layer in any image.

This algorithm is useful to temporarily exclude some images in response files.

`-h`
`--help` Print information on the available options and exit.

`-l` *LEVELS*
`--levels=`*LEVELS*

Use at most this many *LEVELS* for pyramid[1] blending if *LEVELS* is positive, or reduce the maximum number of levels used by −*LEVELS* if *LEVELS* is negative; '`auto`' or '`automatic`' restore the default, which is to use the maximum possible number of levels for each overlapping region.

The number of levels used in a pyramid controls the balance between local and global image features (contrast, saturation, . . .) in the blended region. Fewer levels emphasize local features and suppress global ones. The more levels a pyramid has, the more global features will be taken into account.

As a guideline, remember that each new level works on a linear scale twice as large as the previous one. So, the zeroth layer, the original image, obviously defines the image at single-pixel scale, the first level works at two-pixel scale, and generally, the n-th level contains image data at 2^n-pixel scale. This is the reason why an image of *widthxheight* pixels cannot be deconstructed into a pyramid of more than $\lfloor \log_2(\min(width, height)) \rfloor$ levels.

If too few levels are used, "halos" around regions of strong local feature variation can show up. On the other hand, if too many levels are used, the image might contain too much global features. Usually, the latter is not a problem, but is highly desired. This is the reason, why the default is to use as many levels as is possible given the size of the overlap regions. Enblend may still use a smaller number of levels if the geometry of the overlap region demands.

Positive values of *LEVELS* limit the maximum number of pyramid levels. Depending on the size and geometry of the overlap regions this may or may not influence any pyramid. Negative values of *LEVELS* reduce the number of pyramid levels below the maximum no matter what the actual maximum is and thus

[1] As Dr. Daniel Jackson correctly noted, actually, it is not a pyramid: "Ziggaurat, it's a Ziggaurat."

always influence all pyramids. Use '`auto`' or '`automatic`' as *LEVELS* to restore the automatic calculation of the maximum number of levels.

The valid range of the absolute value of *LEVELS* is 1 to 29.

`-o`
`--output=`*FILE*

Place output in *FILE*.

If `--output` is not specified, the default is to put the resulting image in `a.tif`.

`--parameter=`*KEY*`[=`*VALUE*`]:...`

Set a *KEY-VALUE* pair, where *VALUE* is optional. This option is cumulative. Separate multiple pairs with the usual numeric delimiters.

This option has the negated form `--no-parameter`, which takes one or more *KEY*s and removes them from the list of defined parameters. The special key '`*`' deletes all parameters at once.

Parameters allow the developers to change the internal workings of Enblend without the need to recompile.

`-v`
`--verbose[=`*LEVEL*`]`

Without an argument, increase the verbosity of progress reporting. Giving more `--verbose` options will make Enblend more verbose. Directly set a verbosity level with a non-negative integral *LEVEL*.

Each level includes all messages of the lower levels.

Level	Messages
0	only warnings and errors
1	reading and writing of images
2	mask generation, pyramid, and blending
3	reading of response files, color conversions
4	image sizes, bounding boxes and intersection sizes
5	detailed information on the optimizer runs (Enblend only)
6	estimations of required memory in selected processing steps

The default verbosity level of Enblend is 1.

`-V`
`--version`

Output information on the Enblend version.

Team this option with `--verbose` to show configuration details, like the extra features that have been compiled in.

`-w`
`--wrap=`*MODE*

Blend around the boundaries of the panorama.

As this option significantly increases memory usage and computation time only use it, if the panorama will be

- consulted for any kind measurement, this is, all boundaries must match as accurately as possible, or
- printed out and the boundaries glued together, or
- fed into a virtual reality (VR) generator, which creates a seamless environment.

Otherwise, always avoid this option!

With this option Enblend treats the panorama of width w and height h as an infinite data structure, where each pixel $P(x, y)$ of the input images represents the set of pixels $S_P(x, y)$[2].

MODE takes the following values:

'none'
'open' This is a "no-op"; it has the same effect as not giving `--wrap` at all. The set of input images is considered open at its boundaries.

'horizontal'
Wrap around horizontally:

$$S_P(x, y) = \{P(x + mw, y) : m \in Z\}.$$

This is useful for 360° horizontal panoramas as it eliminates the left and right borders.

'vertical'
Wrap around vertically:

$$S_P(x, y) = \{P(x, y + nh) : n \in Z\}.$$

This is useful for 360° vertical panoramas as it eliminates the top and bottom borders.

'both'
'horizontal+vertical'
'vertical+horizontal'
Wrap around both horizontally and vertically:

$$S_P(x, y) = \{P(x + mw, y + nh) : m, n \in Z\}.$$

In this mode, both left and right borders, as well as top and bottom borders, are eliminated.

Specifying `--wrap` without *MODE* selects horizontal wrapping.

`-x` Checkpoint partial results to the output file after each blending step.

[2] Solid-state physicists will be reminded of the Born-von Kármán boundary condition.

3.4 Extended Options

Extended options control the image cache, the color model, and the cropping of the output image.

`-b` *BLOCKSIZE*

Set the *BLOCKSIZE* in kilobytes (KB) of Enblend's image cache.

This is the amount of data that Enblend will move to and from the disk at one time. The default is 2048 KB, which should be ok for most systems. See Chapter 7 [Tuning Memory Usage], page 30 for details.

Note that Enblend must have been compiled with the image-cache feature for this option to be effective. Find out about extra features with `enblend --version --verbose`.

`--blend-colorspace=`*COLORSPACE*

Force blending in selected *COLORSPACE*. For well matched images this option should not change the output image much. However, if Enblend must blend vastly different colors (as e.g. anti-colors) the result image heavily depends on the *COLORSPACE*.

Usually, Enblend chooses defaults depending on the input images:

- For input images *with* ICC profiles the default is to use CIECAM colorspace.
- Images *without* color profiles and floating-point images are blended in the RGB-color cube by default.

Enblend supports two blend colorspaces:

'`IDENTITY`', '`ID`', '`UNIT`'

Naively compute blended colors in the luminance interval (grayscale images) or RGB-cube (RGB images) spanned by the input ICC profile or sRGB if no profiles are present. Consider option `--fallback-profile` to force a different profile than sRGB on all input images.

'`CIECAM`', '`CIECAM02`', '`JCH`'

Blend pixels in the CIECAM02 colorspace.

Please keep in mind that by using different blend colorspaces, blending may not only change the colors in the output image, but Enblend may choose different seam line routes as some seam-line optimizers are guided by image differences, which may be different when viewed in different colorspaces.

`-c`
`--ciecam` Use '`--blend-colorspace=CIECAM`' instead. To mimic the negated option `--no-ciecam` use '`--blend-colorspace=IDENTITY`'.

`-d`
`--depth=`*DEPTH*

Force the number of bits per channel and the numeric format of the output image.

Enblend always uses a smart way to change the channel depth to assure highest image quality (at the expense of memory), whether requantization is implicit because of the output format or explicit with option `--depth`.

- If the output-channel width is larger than the input-channel width of the input images, the input images' channels are widened to the output channel width immediately after loading, that is, as soon as possible. Enblend then performs all blending operations at the output-channel width, thereby preserving minute color details which can appear in the blending areas.
- If the output-channel width is smaller than the input-channel width of the input images, the output image's channels are narrowed only right before it is written to disk, that is, as late as possible. Thus the data benefits from the wider input channels for the longest time.

All *DEPTH* specifications are valid in lowercase as well as uppercase letters. For integer format, use

`8`, `uint8` Unsigned 8 bit; range: 0..255

`int16` Signed 16 bit; range: −32768..32767

`16`, `uint16`
Unsigned 16 bit; range: 0..65535

`int32` Signed 32 bit; range: −2147483648..2147483647

`32`, `uint32`
Unsigned 32 bit; range: 0..4294967295

For floating-point format, use

`r32`, `real32`, `float`
IEEE754 single precision floating-point, 32 bit wide, 24 bit significant

- Minimum normalized value: 1.2e-38
- Epsilon: 1.2e-7
- Maximum finite value: 3.4e38

`r64`, `real64`, `double`
IEEE754 double precision floating-point, 64 bit wide, 53 bit significant

- Minimum normalized value: 2.2e-308
- Epsilon: 2.2e-16
- Maximum finite value: 1.8e308

If the requested *DEPTH* is not supported by the output file format, Enblend warns and chooses the *DEPTH* that matches best.

The OpenEXR data format is treated as IEEE754 float internally. Externally, on disk, OpenEXR data is represented by "half" precision floating-point numbers.

OpenEXR half precision floating-point, 16 bit wide, 10 bit significant

- Minimum normalized value: 9.3e-10
- Epsilon: 2.0e-3
- Maximum finite value: 4.3e9

`-f` *WIDTH*x*HEIGHT*

`-f` *WIDTH*x*HEIGHT*+x*XOFFSET*+y*YOFFSET*

Ensure that the minimum "canvas" size of the output image is at least *WIDTH*x *HEIGHT*. Optionally specify the *XOFFSET* and *YOFFSET*, too.

This option only is useful when the input images are cropped TIFF files, such as those produced by `nona`[3].

Note that option `-f` neither rescales the output image, nor shrinks the canvas size below the minimum size occupied by the union of all input images.

`--fallback-profile=`*PROFILE-FILENAME*

Use the ICC profile in *PROFILE-FILENAME* instead of the default sRGB. See option `--blend-colorspace` and Chapter 5 [Color Profiles], page 27.

This option only is effective if the input images come without color profiles and blending is performed in CIECAM02 color appearance model.

`-g` Save alpha channel as "associated". See the TIFF documentation for an explanation.

Gimp (before version 2.0) and CinePaint (see Chapter 8 [Helpful Programs], page 32) exhibit unusual behavior when loading images with unassociated alpha channels. Use option `-g` to work around this problem. With this flag Enblend will create the output image with the associated alpha tag set, even though the image is really unassociated alpha.

`--gpu` Use the graphics card – in fact the graphics processing unit (GPU) – to accelerate some computations.

This is an experimental feature that may not work on all systems. In this version of Enblend, 4.1.4, only mask optimization by Simulated Annealing benefits from this option.

Note that GPU-support must have been compiled into Enblend for this option to be available. Find out about this feature with `enblend --version --verbose`.

`-m` *CACHESIZE*

Set the *CACHESIZE* in megabytes (MB) of Enblend's image cache.

This is the amount of memory Enblend will use for storing image data before swapping to disk. The default is 1024 MB, which is good for systems with 3–4 gigabytes (GB) of RAM. See Chapter 7 [Tuning Memory Usage], page 30 for details.

Note that Enblend must have been compiled with the image-cache feature for this option to be effective. Find out about extra features with `enblend --version --verbose`.

`--no-ciecam`

Use '`--blend-colorspace=IDENTITY`' instead.

See option `--blend-colorspace` for details. Also see Chapter 5 [Color Profiles], page 27.

[3] The stitcher `nona` is part of Hugin. See Chapter 8 [Helpful Programs], page 32.

3.5 Mask Generation Options

These options control the generation and the usage of masks.

`--anneal=`*`TAU`*`[:`*`DELTA-E-MAX`*`[:`*`DELTA-E-MIN`*`[:`*`K-MAX`*`]]]`

Set the parameters of the Simulated Annealing optimizer (see Table 3.5).

TAU — *TAU* is the temperature reduction factor in the Simulated Annealing; it also can be thought of as "cooling factor". The closer *TAU* is to one, the more accurate the annealing run will be, and the longer it will take.

Append a percent sign ('%') to specify *TAU* as a percentage.

Valid range: 0 < *TAU* < 1.

The default is 0.75; values around 0.95 are reasonable. Usually, slower cooling results in more converged points.

DELTA-E-MAX
DELTA-E-MIN

DELTA-E-MAX and *DELTA-E-MIN* are the maximum and minimum cost change possible by any single annealing move.

Valid range: 0 < *DELTA-E-MIN* < *DELTA-E-MAX*.

In particular they determine the initial and final annealing temperatures according to:

$$T_{\mathrm{initial}} = \frac{\mathit{DELTA\text{-}E\text{-}MAX}}{\log(\mathit{K\text{-}MAX}/(\mathit{K\text{-}MAX}-2))}$$

$$T_{\mathrm{final}} = \frac{\mathit{DELTA\text{-}E\text{-}MIN}}{\log(\mathit{K\text{-}MAX}^2 - \mathit{K\text{-}MAX} - 1)}$$

The defaults are: *DELTA-E-MAX*: 7000.0 and *DELTA-E-MIN*: 5.0.

K-MAX — *K-MAX* is the maximum number of "moves" the optimizer will make for each line segment. Higher values more accurately sample the state space, at the expense of a higher computation cost.

Valid range: $\mathit{K\text{-}MAX} \geq 3$.

The default is 32. Values around 100 seem reasonable.

`--coarse-mask[=`*`FACTOR`*`]`

Use a scaled-down version of the input images to create the seam line. This option reduces the number of computations necessary to compute the seam line and the amount of memory necessary to do so. It is the default.

If omitted *FACTOR* defaults to 8, this means, option `--coarse-mask` shrinks the overlapping *areas* by a factor of $8x8$. With $\mathit{FACTOR} = 8$ the total memory allocated during a run of Enblend shrinks approximately by 80% and the maximum amount of memory in use at a time is decreased to 60% (Enblend compiled with image cache) or 40% (Enblend compiled without image cache).

Valid range: $\mathit{FACTOR} = 1, 2, 3, \ldots$.

Also see Table 3.4.

	`--no-optimize`	`--optimize`
`--fine-mask`	Use NFT mask.	Vectorize NFT mask, optimize vertices with simulated annealing and Dijkstra's shortest path algorithm, fill vector contours.
`--coarse-mask`	Scale down overlap region, compute NFT mask and vectorize it, fill vector contours.	Scale down overlap region, vectorize NFT mask, optimize vertices with simulated annealing and Dijkstra's shortest path algorithm, fill vector contours.

Table 3.4: Various options that control the generation of masks. All mask computations are based on the Nearest-Feature Transformation (NFT) of the overlap region.

`--dijkstra=`*`RADIUS`*

Set the search *RADIUS* of the Dijkstra Shortest Path algorithm used in Dijkstra Optimization (see Table 3.5).

A small value prefers straight line segments and thus shorter seam lines. Larger values instruct the optimizer to let the seam line take more detours when searching for the best seam line.

Valid range: *RADIUS* ≥ 1.

Default: 25 pixels.

`--fine-mask`

Instruct Enblend to employ the full-size images to create the seam line, which can be slow. Use this option, for example, if you have very narrow overlap regions.

Also see Table 3.4.

`--image-difference=`*`ALGORITHM`*`[:`*`LUMINANCE-WEIGHT`*`[:`*`CHROMINANCE-WEIGHT`*`]]`

Enblend calculates the difference of a pair of overlapping color images when it generates the primary seam with a Graph-Cut or before it optimizes a seam. It employs a user-selectable *ALGORITHM* that itself is controlled by the weights for luminance differences *LUMINANCE-WEIGHT*, $w_{\text{luminance}}$ and color differences *CHROMINANCE-WEIGHT*, $w_{\text{chrominance}}$.

For black-and-white images the difference is simple the absolute difference of each pair of pixels.

`maximum-hue-luminance`
`maximum-hue-lum`
`max-hue-luminance`
`max-hue-lum`
`max` Calculate the difference d as the maximum of the differences of the luminances l and hues h of each pair of pixels P_1 and P_2:

$$d = \max(w_{\text{luminance}} \times |l(P_1) - l(P_2)|, \\ w_{\text{chrominance}} \times |h(P_1) - h(P_2)|)$$

This algorithm was the default for Enblend up to version 4.0.

`delta-e`
`de` Calulate the difference d as the EUCLIDEAN distance of the pixels in L*a*b* space:

$$d = (w_{\text{luminance}} \times (L(P_1) - L(P_2))^2 + \\ w_{\text{chrominance}} \times (a(P_1) - a(P_2))^2 + \\ w_{\text{chrominance}} \times (b(P_1) - b(P_2))^2)^{1/2}$$

This is the default in Enblend version 4.1 and later.

Note that the “delta-E” mentioned here has nothing to do with *DELTA-E-MAX* and *DELTA-E-MIN* of option `--anneal`.

Both *LUMINANCE-WEIGHT* and *CHROMINANCE-WEIGHT* must be non-negative, their sum must be positive. Enblend automatically normalizes the sum of *LUMINANCE-WEIGHT* and *CHROMINANCE-WEIGHT* to one. Thus ‘`--image-difference=delta-e:2:1`’ and ‘`--image-difference=delta-e:0.6667:0.3333`’ define the same weighting function.

The default *LUMINANCE-WEIGHT* is 1.0 and the default *CHROMINANCE-WEIGHT* is 1.0.

At higher verbosity levels Enblend computes the true size of the overlap area in pixels and it calculates the average and standard deviation of the difference per pixel in the normalized luminance interval $[0 \ldots 1]$. These statistical measures are based on *ALGORITHM*, therefore they should only be compared for identical *ALGORITHM*s. The average difference is a rough measure of quality with lower values meaning better matches.

`--load-masks[=`*`IMAGE-TEMPLATE`*`]`

Instead of generating masks, use those in *IMAGE-TEMPLATE*. The default is `mask-%n.tif`. The mask images have to be a 8-bit grayscale images.

See `--save-masks` below for details.

`--mask-vectorize=`*`DISTANCE`*

Set the mask vectorization *DISTANCE* Enblend uses to partition each seam. Thus, break down the seam to segments of length *DISTANCE* each.

If Enblend uses a coarse mask (`--coarse-mask`) or Enblend optimizes (`--optimize`) a mask it vectorizes the initial seam line before performing further operations. See Table 3.4 for the precise conditions. *DISTANCE* tells Enblend how long to make each of the line segments called vectors here.

The unit of *DISTANCE* is pixels unless it is a percentage as explained in the next paragraph. In fine masks one mask pixel corresponds to one pixel in the input image, whereas in coarse masks one pixel represents for example 8 pixels in the input image.

Append a percentage sign ('`%`') to *DISTANCE* to specify the segment length as a fraction of the diagonal of the rectangle including the overlap region. Relative measures do not depend on coarse or fine masks, they are recomputed for each mask. Values around 5%–10% are a good starting point.

This option massively influences the mask generation process! Large *DISTANCE* values lead to shorter, straighter, less wiggly, less baroque seams that are on the other hand less optimal, because they run through regions of larger image mismatch instead of avoiding them. Small *DISTANCE* values give the optimizers more possibilities to run the seam around high mismatch areas.

What should *never* happen though, are loops in the seam line. Counter loops with higher weights of *DISTANCE-WEIGHT* (in option `--optimizer-weights`), larger vectorization *DISTANCE*s, *TAU*s (in option `--anneal`) that are closer to one, and blurring of the difference image with option `--smooth-difference`. Use option `--visualize` to check the results.

Valid range: $DISTANCE \geq 4$.

Enblend limits *DISTANCE* so that it never gets below 4 even if it has been given as a percentage. The user will be warned in such cases.

Default: 4 pixels for coarse masks and 20 pixels for fine masks.

`--no-optimize`

Turn off seam line optimization. Combined with option `--fine-mask` this will produce the same type of mask as Enblend version 2.5, namely the result of a Nearest-Feature Transform (NFT).[4]

Also see Table 3.4.

`--optimize`

Use a multi-strategy approach to route the seam line around mismatches in the overlap region. This is the default. Table 3.5 explains these strategies; also see Table 3.4.

[4] Muhammad H. Alsuwaiyel and Marina Gavrilova, "On the Distance Transform of Binary Images", Proceedings of the International Conference on Imaging Science, Systems, and Technology, June 2000, Vols. I and II, pages 83–86.

Simulated Annealing
: Tune with option `--anneal` = *TAU* : *DELTA-E-MAX* : *DELTA-E-MIN* : *K-MAX*.

  Simulated-Annealing

Dijkstra Shortest Path
: Tune with option `--dijkstra` = *RADIUS*.

  Dijkstra algorithm

Table 3.5: Enblend's strategies to optimize the seam lines between images.

`--optimizer-weights=`*DISTANCE-WEIGHT*`[:`*MISMATCH-WEIGHT*`]`

Set the weights of the seam-line optimizer. If omitted, *MISMATCH-WEIGHT* defaults to 1.

The seam-line optimizer considers two qualities of the seam line:

- The distance of the seam line from its initial position, which has been determined by NFT (see option `--no-optimize`).
- The total "mismatch" accumulated along it.

The optimizer weights *DISTANCE-WEIGHT* and *MISMATCH-WEIGHT* define how to weight these two criteria. Enblend up to version 3.2 used 1:1. This version of Enblend (4.1.4) uses 8.0:1.0.

A large *DISTANCE-WEIGHT* pulls the optimized seam line closer to the initial postion. A large *MISMATCH-WEIGHT* makes the seam line go on detours to find a path along which the mismatch between the images is small. If the optimized seam line shows cusps or loops (see option `--visualize`), reduce *MISMATCH-WEIGHT* or increase *DISTANCE-WEIGHT*.

Both weights must be non-negative. They cannot be both zero at the same time. Otherwise, their absolute values are not important as Enblend normalizes their sum.

`--primary-seam-generator=`*ALGORITHM*

Select the algorithm responsible for generating the general seam route.

This is the *ALGORITHM* that produces an initial seam line, which is the basis for later, optional optimizations (see `--optimize`). Nearest Feature Transform (NFT) is the only algorithm up to and including Enblend version 4.0. Version 4.1 adds a Graph-Cut (GC) algorithm. In this version of Enblend NFT is the default.

Valid *ALGORITHM* names are:

nearest-feature-transform
nft
: Nearest Feature Transform

graph-cut
gc
: Graph-Cut

See Chapter 4 [Primary Seam Generators], page 26 for details.

`--save-masks`
`--save-masks=`*`IMAGE-TEMPLATE`*

Save the generated masks to *IMAGE-TEMPLATE*. The default is `mask-%n.tif`. Enblend saves masks as 8 bit grayscale (single channel) images. For accuracy we recommend to choose a lossless format.

Use this option if you wish to edit the location of the seam line by hand. This will give you images of the right sizes that you can edit to make your changes. Later, use option **`--load-masks`** to blend the project with your custom seam lines.

Enblend will stop after saving all masks unless option **`--output`** is given, too. With both options given, this is, **`--save-masks`** and **`--output`**, Enblend saves all masks and then proceeds to blend the output image.

IMAGE-TEMPLATE defines a template that is expanded for each input file. In a template a percent sign ('`%`') introduces a variable part. All other characters are copied literally. Lowercase letters refer to the name of the respective input file, whereas uppercase ones refer to the name of the output file (see Section 3.3 [Common Options], page 9). Table 3.7 lists all variables.

A fancy mask filename template could look like this:

```
%D/mask-%02n-%f.tif
```

It puts the mask files into the same directory as the output file ('`%D`'), generates a two-digit index ('`%02n`') to keep the mask files nicely sorted, and decorates the mask filename with the name of the associated input file ('`%f`') for easy recognition.

`--smooth-difference=`*`RADIUS`*

This option has been deprecated.

Smooth the difference image prior to seam-line optimization to get a shorter and – on the length scale of *RADIUS* – also a straighter seam-line. The default is not to smooth.

If *RADIUS* is larger than zero Enblend blurs the difference images of the overlap regions with a GAUSSIAN filter having a radius of *RADIUS* pixels. Values of 0.5 to 1.5 pixels for *RADIUS* are good starting points; use option **`--visualize`** to directly judge the effect.

When using this option in conjunction with option **`--coarse-mask`**=*FACTOR*, keep in mind that the smoothing occurs *after* the overlap regions have been shrunken. Thus, blurring affects a *FACTOR*x*FACTOR* times larger area in the original images.

Valid range: $RADIUS \geq 0.0$.

`--visualize[=`*`VISUALIZE-TEMPLATE`*`]`

Create an image according to *VISUALIZE-TEMPLATE* that visualizes the unoptimized mask and the applied optimizations (if any). The default is `vis-%n.tif`.

The image shows Enblend's view of the overlap region and how it decided to route the seam line. If you are experiencing artifacts or unexpected output, it may be useful to include this visualization image in your bug report. See Appendix A [Bug Reports], page 34.

VISUALIZE-TEMPLATE defines a template that is expanded for each input file. In a template, a percent sign ('`%`') introduces a variable part; all other characters are copied literally. Lowercase letters refer to the name of the respective input file, whereas uppercase ones refer to the name of the output file (see Section 3.3 [Common Options], page 9). Table 3.7 lists all variables.

Visualization Image.

The visualization image shows the symmetric difference of the pixels in the rectangular region where two images overlap. The larger the difference the lighter shade of gray it appears in the visualization image. Enblend paints the non-overlapping parts of the image pair – these are the regions where *no* blending occurs – in dark red. Table 3.6 shows the meanings of all the colors that are used in seam-line visualization images.

dark red
: Non-overlapping parts of image pair.

various shades of gray
: Difference of the pixel values in the overlap region.

dark blue
: Location of an optimizer sample.

medium green
: First sample of a line segment.

light green
: Any other but first sample of a line segment.

bright cyan
: State space sample inside the Dijkstra radius.

bright magenta
: Non-converged point.

dark yellow
: Initial seam line as generated by the primary seam generator.

: Enblend marks a non-movable ("frozen") endpoint of a seam-line segment with a bright white cross, whereas it uses a light orange diamond to denote an endpoint that the optimizer is allowed to move around.

bright yellow
: Final seam line.

Table 3.6: Colors used in seam-line visualization images.

Figure 3.1 shows an example of a seam-line visualization. It was produced with an Enblend run at all defaults, but `--fine-mask` and `--visualize` enabled.

The large dark red border is "off-limits" for Enblend, for the images do not overlap there. The dark wedge inside the dark red frame is where the images share a common region.

The initial seam-line (dark yellow) is almost straight with the exception of a single bend on the left side of the image and the final seam-line (bright yellow) meanders around it.

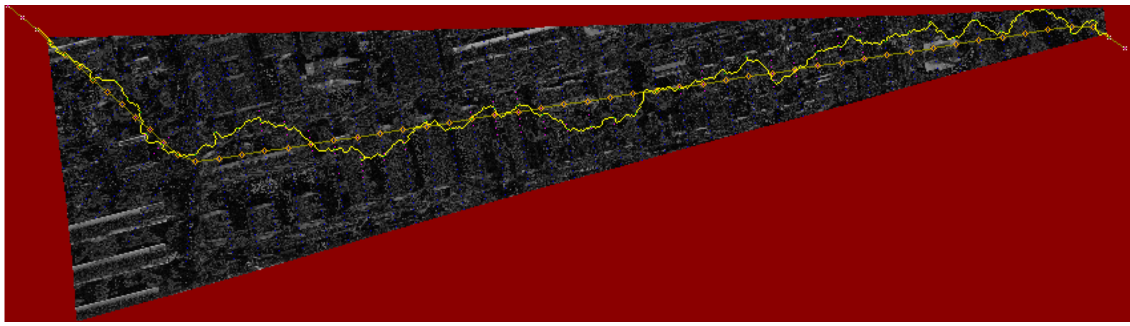

Figure 3.1: Seam-line visualization of a simple overlap. The 853x238 pixel image has been rescaled to a width of approximately 15 cm.

`%%`	Produces a literal '`%`'-sign.
`%i`	Expands to the index of the mask file starting at zero. '`%i`' supports setting a pad character or a width specification: `%` *PAD WIDTH* `i` *PAD* is either '`0`' or any punctuation character; the default pad character is '`0`'. *WIDTH* is an integer specifying the minimum width of the number. The default is the smallest width given the number of input images, this is 1 for 2–9 images, 2 for 10–99 images, 3 for 100–999 images, and so on. Examples: '`%i`', '`%02i`', or '`%_4i`'.
`%n`	Expands to the number of the mask file starting at one. Otherwise it behaves identically to '`%i`', including pad character and width specification.
`%p`	This is the full name (path, filename, and extension) of the input file associated with the mask. Example: If the input file is called `/home/luser/snap/img.jpg`, '`%p`' expands to `/home/luser/snap/img.jpg`, or shorter: '`%p`' ⇒ `/home/luser/snap/img.jpg`.
`%P`	This is the full name of the output file.
`%d`	Is replaced with the directory part of the associated input file. See Info file `coreutils.info`, node '`dirname invocation`'. Example (cont.): '`%d`' ⇒ `/home/luser/snap`.
`%D`	Is replaced with the directory part of the output file.
`%b`	Is replaced with the non-directory part (often called "basename") of the associated input file. See Info file `coreutils.info`, node '`basename invocation`'. Example (cont.): '`%b`' ⇒ `img.jpg`.
`%B`	Is replaced with the non-directory part of the output file.
`%f`	Is replaced with the filename without path and extension of the associated input file. Example (cont.): '`%f`' ⇒ `img`.
`%F`	Is replaced with the filename without path and extension of the output file.
`%e`	Is replaced with the extension (including the leading dot) of the associated input file. Example (cont.): '`%e`' ⇒ `.jpg`.
`%E`	Is replaced with the extension of the output file.

Table 3.7: Special characters to control the generation of mask filenames.

4 Primary Seam Generators

This version (4.1.4) of Enblend supports two main algorithms to generate seam lines. Use option `--primary-seam-generator` to select one of the generators.

Nearest Feature Transform (NFT)

: The NFT, also known as Distance Transform, is a fast and efficient technique to produce a seam line route given the geometries of multiple overlapping images.

: NFT as implemented in this version of Enblend only takes into account the *shape of the overlap area*. It completely ignore the images' contents.

Graph-Cut (GC)

: GC is a region-oriented way of segmenting images.

: The generator is based on the idea of finding a minimum cost "cut" of a graph created from a given image pair. A "cut" is where the seam line appears. GC determines the cost from the overlapping images' contents.

The most significant difference between the two algorithms is the output mask gradation. NFT produces a coarse approximation of the seam, running as far away from the overlap-region borders as possible. The resulting mask could then be blended as-is, however, Enblend by default runs image-content dependent optimizers to increase the mask gradation and for example omits the regions where the images differ. The result is a finer seam line, which only loosely follows the shape of NFT's primary seam.

Graph-Cut, on the other hand, is capable of producing the final mask in one pass without the need of further optimizers. It looks for a seam line that is *globally* optimal, taking into account

- feature frequency, as well as
- image dissimilarity.

This means, the seam is less likely to cross lines like for example fences, lampposts, or road markings, where they would be visible.

The optimizers which run after NFT can also be run after GC. Nevertheless, GC works best just with a fine mask (option `--fine-mask`); optimizers are then automatically *turned off* to take full advantage of the detailed seam GC produces.

GC requires more memory and computation time to complete than NFT. Thus, it is best to prefer NFT where the images used are large and execution time is crucial. If quality is the priority, using GC and fine mask usually produces visually more pleasing results.

GC is currently limited to seams that begin and end on the images' borders. This means that the algorithm cannot run in cases where, for example, one image is contained in another, resulting in a loop-like seam. In such cases, though, Enblend automatically falls back to a NFT-generated seam, making its application transparent to the user.

5 Color Profiles

Enblend and Enfuse expect that either

1. no input image has a color profile or
2. all come with the *same* ICC profile.

In case 1 the applications blend or fuse in the RGB-cube, whereas in case 2 the images first are transformed to CIECAM02 color space – respecting the input color profile – then they are blended or fused, and finally the data transformed back to RGB color space. Moreover, in case 2, Enblend and Enfuse assign the input color profile to the output image.

Mixing different ICC profiles or alternating between images with profiles and without them generates warnings as it generally leads to unpredictable results.

The options `--ciecam` (see Section 3.4 [Extended Options], page 14) and its opposite `--no-ciecam` (see Section 3.4 [Extended Options], page 14) overrule the default profile selection procedure described above. Use option `--ciecam` on a set of input images *without* color profiles to assign a profile to them and perform the blending or fusing process in CIECAM02 color space.

The default profile is sRGB. Override this setting with option `--fallback-profile` (see Section 3.4 [Extended Options], page 14).

On the other hand, suppress the utilization of CIECAM02 blending or fusing of a set of input images *with* color profiles with option `--no-ciecam`. The only reason for the latter is to shorten the blending- or fusing-time, because transforming to and back from the CIECAM02 color space are computationally expensive operations.

Option `--ciecam` as well as `--fallback-profile` have no effect on images with attached color profiles, just as option `--no-ciecam` has no effect on images without profiles.

The impact of blending in CIECAM02 color space as opposed to the RGB cube vary with the contents of the input images. Generally colors lying close together in RGB space experience less change when switching the blending spaces. However, colors close the border of any color space can see marked changes.

For color geeks: The transformations to CIECAM02 color space and back use

- perceptual rendering intent,
- the D50 white point,
- 500 lumen surrounding light (“average” in CIECAM02 parlance), and
- assume complete adaption.

6 Understanding Masks

A *binary mask* indicates for every pixel of an image if this pixel must be considered in further processing, or ignored. For a *weight mask*, the value of the mask determines how much the pixel contributes, zero again meaning “no contribution”.

Masks arise in two places: as part of the input files and as separate files, showing the actual pixel weights prior to image blendung or fusion. We shall explore both occurrences in the next sections.

6.1 Masks in Input Files

Each of the input files for Enfuse and Enblend can contain its own mask. Both applications interpret them as binary masks no matter how many bits per image pixel they contain.

Use ImageMagick’s `identify` or, for TIFF files, `tiffinfo` to inquire quickly whether a file contains a mask. Chapter 8 [Helpful Programs], page 32 shows where to find these programs on the web.

```
$ identify -format "%f %m %wx%h %r %q-bit" remapped-0000.tif
remapped-0000.tif TIFF 800x533 DirectClassRGBMatte 8-bit
                                             ^^^^^ mask

$ tiffinfo remapped-0000.tif
TIFF Directory at offset 0x1a398a (1718666)
  Subfile Type: (0 = 0x0)
  Image Width: 800 Image Length: 533
  Resolution: 150, 150 pixels/inch
  Position: 0, 0
  Bits/Sample: 8
  Sample Format: unsigned integer
  Compression Scheme: PackBits
  Photometric Interpretation: RGB color
  Extra Samples: 1<unassoc-alpha>            <<<<< mask
  Orientation: row 0 top, col 0 lhs
  Samples/Pixel: 4                           <<<<< R, G, B, and mask
  Rows/Strip: 327
  Planar Configuration: single image plane
```

The “Matte” part of the image class and the “Extra Samples” line tell us that the file features a mask. Also, many interactive image manipulation programs show the mask as a separate channel, sometimes called “Alpha”. There, the white (high mask value) parts of the mask enable pixels and black (low mask value) parts suppress them.

The multitude of terms all describing the concept of a mask is confusing.

Mask
: A mask defines a selection of pixels. A value of zero represents an unselected pixel. The maximum value (“white”) represents a selected pixel and the values between zero and the maximum are partially selected pixels. See Gimp-Savy.

Alpha Channel
: The alpha channel stores the transpacency value for each pixel, typically in the range from zero to one. A value of zero means the pixel is completely

transparent, thus does not contribute to the image. A value of one on the other hand means the pixel is completely opaque.

Matte
: The notion "matte" as used by ImageMagick refers to an inverted alpha channel, more precisely: 1 - alpha. See ImageMagick for further explanations.

Enblend and Enfuse only consider pixels that have an associated mask value other than zero. If an input image does not have an alpha channel, Enblend warns and assumes a mask of all non-zero values, that is, it will use every pixel of the input image for fusion.

Stitchers like `nona` add a mask to their output images.

Sometimes it is helpful to manually modify a mask before fusion. For example to suppress unwanted objects (insects and cars come into mind) that moved across the scene during the exposures. If the masks of all input images are black at a certain position, the output image will have a hole in that position.

6.2 Weight Mask Files

...

7 Tuning Memory Usage

The default configuration of Enblend and Enfuse assumes a system with 3–4 GB of RAM.

If Enblend and Enfuse have been compiled with the "image-cache" feature, they do not rely on the operating system's memory management, but use their own image cache in the file system. To find out whether your version uses the image cache say

```
enblend --verbose --version
```

or

```
enfuse --verbose --version
```

Enblend and Enfuse put the file that holds the image cache either in the directory pointed to by the environment variable `TMPDIR`, or, if the variable is not set, in directory `/tmp`. It is prudent to ensure write permissions and enough of free space on the volume with the cache file.

The size of the image cache is user configurable with the option '`-m` *CACHE-SIZE*' (see Section 3.4 [Extended Options], page 14). Furthermore, option '`-b` *BUFFER-SIZE*' (see Section 3.4 [Extended Options], page 14) allows for fine-tuning the size of a single buffer inside the image cache. Note that *CACHE-SIZE* is given in megabytes, whereas the unit of *BUFFER-SIZE* is kilobytes.

Usually the user lets the operating system take care of the memory management of all processes. However, users of Enblend or Enfuse might want to control the balance between the operating systems' Virtual Memory system and the image cache for several reasons.

- Paging in or out parts of a process' image runs at kernel level and thus can make user processes appear unresponsive or "jumpy". The caching mechanism of Enblend and Enfuse of course runs as a user process, which is why it has less detrimental effects on the system's overall responsiveness.
- The image cache has been optimized for accesses to image data. All algorithms in Enblend and Enfuse have been carefully arranged to play nice with the image cache. An operating system's cache has no knowledge of these particular memory access patterns.
- The disk access of the operating system to the swap device has been highly optimized. Enblend and Enfuse on the other hand use the standard IO-layer, which is a much slower interface.
- Limiting the amount of image cache prevents Enblend and Enfuse from eating up most or all RAM, thereby forcing all user applications into the swap.

The *CACHE-SIZE* should be set in such a way as to reconcile all of the above aspects even for the biggest data sets, that is, projects with many large images.

Table 7.1 suggests some cache- and buffer-sizes for different amounts of available RAM.

RAM MB	**CACHE-SIZE MB**	**BUFFER-SIZE KB**	**Comment**
4096	1024	2048	default
2048	512–1024	1024	
1024	256–512	256–512	

Table 7.1: Suggested cache-size settings

On systems with considerably more than 4 GB of RAM it is recommended to run Enblend or Enfuse versions without image cache.

8 Helpful Additional Programs

Several programs and libraries have proven helpful when working with Enfuse and Enblend.

Raw Image Conversion

- DCRaw is a universal raw-converter written by David Coffin.
- UFRaw, a raw converter written by Udi Fuchs and based on DCRaw, adds a GUI (`ufraw`), versatile batch processing (`ufraw-batch`), and some additional features such as cropping, noise reduction with wavelets, and automatic lens error correction.

Image Alignment and Rendering

- ALE, David Hilvert's anti-lamenessing engine for the real die-hard command-line users aligns, filters, and renders images.
- Hugin is a GUI that aligns and stitches images.

  It comes with several command line tools, like `nona` to stitch panorama images, `align_image_stack` to align overlapping images for HDR or create focus stacks, and `fulla` to correct lens errors.
- PanoTools the successor of Helmut Dersch's original PanoTools offers a set of command-line driven applications to create panoramas. Most notable are `PTOptimizer` and `PTmender`.

Image Manipulation

- CinePaint is a branch of an early Gimp forked off at version 1.0.4. It sports much less features than the current Gimp, but offers 8 bit, 16 bit and 32 bit color channels, HDR (for example floating-point TIFF, and OpenEXR), and a tightly integrated color management system.
- The Gimp is a general purpose image manipulation program. At the time of this writing it is still limited to images with only 8 bits per channel.
- ImageMagick and its clone GraphicsMagick are general purpose command-line driven image manipulation programs, for example, `convert`, `display`, `identify`, and `montage`.

High Dynamic Range

- OpenEXR offers libraries and some programs to work with the EXR HDR format.
- PFSTools create, modify, and tonemap high-dynamic range images.

Libraries

- LibJPEG is a library for handling the JPEG (JFIF) image format.
- LibPNG is a library that handles the Portable Network Graphics (PNG) image format.
- LibTIFF offers a library and utility programs to manipulate the ubiquitous Tagged Image File Format, TIFF.

  The nifty `tiffinfo` command quickly inquires the properties of TIFF files.

Meta-Data Handling

- EXIFTool reads and writes EXIF meta data. In particular it copies meta data from one image to another.

- LittleCMS is the color management library used by Hugin, DCRaw, UFRaw, Enblend, and Enfuse. It supplies some binaries, too. `tifficc`, an ICC color profile applier, is of particular interest.

Appendix A Bug Reports

> Most of this appendix was taken from the
> Octave documentation.

Bug reports play an important role in making Enblend and Enfuse reliable and enjoyable.

When you encounter a problem, the first thing to do is to see if it is already known. To this end visit the package's LaunchPad bug database. Search it for your particular problem. If it is not known, please report it.

In order for a bug report to serve its purpose, you must include the information that makes it possible to fix the bug.

A.1 Have You Really Found a Bug?

If you are not sure whether you have found a bug, here are some guidelines:

- If Enblend or Enfuse get a fatal signal, for any options or input images, that is a bug.
- If Enblend or Enfuse produce incorrect results, for any input whatever, that is a bug.
- If Enblend or Enfuse produce an error message for valid input, that is a bug.
- If Enblend or Enfuse do not produce an error message for invalid input, that is a bug.

A.2 How to Report Bugs

The fundamental principle of reporting bugs usefully is this: report all the facts. If you are not sure whether to state a fact or leave it out, state it. Often people omit facts because they think they know what causes the problem and they conclude that some details do not matter. Play it safe and give a specific, complete example.

Keep in mind that the purpose of a bug report is to enable someone to fix the bug if it is not known. Always write your bug reports on the assumption that the bug is not known.

Try to make your bug report self-contained. If we have to ask you for more information, it is best if you include all the previous information in your response, as well as the information that was missing.

To enable someone to investigate the bug, you should include all these things:

- The exact version and configuration of Enblend or Enfuse. You can get this by running it with the options `--version` and `--verbose`.
- A complete set of input images that will reproduce the bug. Strive for a minimal set of *small*[1] images.
- The type of machine you are using, and the operating system name and its version number.
- A complete list of any modifications you have made to the source. Be precise about these changes. Show a `diff` for them.
- Details of any other deviations from the standard procedure for installing Enblend and Enfuse.
- The *exact command line* you use to call Enblend or Enfuse, which then triggers the bug.

  Examples:

---
[1] Images of a size less than 1500x1000 pixels qualify as small.

```
~/local/bin/enblend -v \
    --fine-mask \
    --optimizer-weights=3:2 --mask-vectorize=12.5% \
    image-1.png image-2.png
```

or:

```
/local/bin/enfuse \
    --verbose \
    --exposure-weight=0 --saturation-weight=0 --entropy-weight=1 \
    --gray-projector=l-star \
    --entropy-cutoff=1.667% \
    layer-01.ppm layer-02.ppm layer-03.ppm
```

If you call Enblend or Enfuse from within a GUI like, for example, Hugin or KImageFuser by Harry van der Wolf, copy&paste or write down the command line that launches Enblend or Enfuse.

- A description of what behavior you observe that you believe is incorrect. For example, "The application gets a fatal signal," or, "The output image contains black holes."

  Of course, if the bug is that the application gets a fatal signal, then one cannot miss it. But if the bug is incorrect output, we might not notice unless it is glaringly wrong.

A.3 Sending Patches for Enblend or Enfuse

If you would like to write bug fixes or improvements for Enblend or Enfuse, that is very helpful. When you send your changes, please follow these guidelines to avoid causing extra work for us in studying the patches. If you do not follow these guidelines, your information might still be useful, but using it will take extra work.

- Send an explanation with your changes of what problem they fix or what improvement they bring about. For a bug fix, just include a copy of the bug report, and explain why the change fixes the bug.
- Always include a proper bug report for the problem you think you have fixed. We need to convince ourselves that the change is right before installing it. Even if it is right, we might have trouble judging it if we do not have a way to reproduce the problem.
- Include all the comments that are appropriate to help people reading the source in the future understand why this change was needed.
- Do not mix together changes made for different reasons. Send them individually.

  If you make two changes for separate reasons, then we might not want to install them both. We might want to install just one.
- Use the version control system to make your diffs. Prefer the unified diff format: `hg diff --unified 4`.
- You can increase the probability that your patch gets applied by basing it on a recent revision of the sources.

Appendix B Authors

Andrew Mihal (acmihal@users.sourceforge.net) has written Enblend and Enfuse.

Contributors

- Pablo d'Angelo (dangelo@users.sourceforge.net) added the contrast criteria.
- Joe Beda: Win32 porting up to version 3.2.
- Kornel Benko, kornelbenko@users.sourceforge.net: CMake support for version 4.0.
- Roger Goodman: Proofreading of the manuals.
- Max Lyons.
- Mark aka mjz: Win32 porting up to version 3.2.
- Thomas Modes, tmodes@users.sourceforge.net: Win32 porting of version 4.0.
- Ryan Sleevi, ryansleevi@users.sourceforge.net: Win32 porting of version 4.0.
- Christoph Spiel (cspiel@users.sourceforge.net) added the gray projectors, the LoG-based edge detection, an O(n)-algorithm for the calculation of local contrast, entropy weighting, and various other features.
- Brent Townshend, btownshend@users.sourceforge.net: HDR support.

Thanks to Simon Andriot and Pablo Joubert for suggesting the Mertens-Kautz-Van Reeth technique and the name "Enfuse".

Appendix C GNU Free Documentation License

Version 1.2, November 2002

0. PREAMBLE

   The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

   This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

   We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

   This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

   A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

   A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

   The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

   You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

   You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

   The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

   In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

   You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

   You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

   A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

   Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

   If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

   You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

    The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See `http://www.gnu.org/copyleft/`.

    Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

Program Index

Syntactic-Comment Index

Option Index

General Index

I

J

K

L

M

N

O

P