

NAME

movimenc – MovIm encoder

SYNOPSIS

movimenc [input_options] **-i** *input_file* [output_options] **-o** *output_file*

movimenc **-h** | **-v**

DESCRIPTION

MovIm is a video codec specifically designed for both conservation and restoration of moving images.

libmovim is a C library implementing **MovIm**. Its associated utility **movimenc** is a **MovIm** encoder.

The **openMovIm** package includes the **libmovim** library and its associated **movimenc**, **movimdec** and **movimplay** utilities, as well as the **openmovim** Bash command–line interface.

OPTIONS**GENERAL OPTIONS**

-i *input_file*, **--input=***input_file*

All container formats and video codecs supported by FFmpeg should work.

-o *output_file*, **--output=***output_file*

The uncompressed or lossless compressed MovIm data can be used directly as a file (.movim).

In addition, the container formats NUT (.nut), MP4 (.mp4), QuickTime (.mov), AVI (.avi) and Matroska (.mkv) have been tested as wrappers for the MovIm video codec.

ENCODING OPTIONS

The following list is not exhaustive.

--bit-depth=*bit_depth*

bit_depth can be any positive integer

We have tested mainly with 10, 12, 16 (default) or 24 per channel. We suggest to digitise at 16-bit per channel and to use this bit–depth for actual restoration work.

Currently, 24-bit per channel is primarily meant for research purposes on file formats for the future, because it can hardly be transcoded into current formats.

--xyz-matrix=(*x_0 y_0 z_0 . . . x_n y_n z_n*)

defines how the *input_file* should be read and/or how the *output_file* should be written

A channel can be not only one of the current R, G, B, Y, Cb, Cr, Co, Cg or alpha, but also a Bayer-filtered channel or any band of a multispectral scan. Any number of channels is supported.

The format of the XYZ matrix is still evolving. An example of the current matrix format (CIE RGB) is:

```
0.7355  0.2645  0.0000
0.2658  0.7243  0.0099
0.1669  0.0085  0.8246
```

--illuminant=(*x y z*)

defines the illuminant

The default value is D65, i.e.

```
0.31271  0.32902  0.35827
```

--endian={*big|little*}

endianness can be *big* or *little* (default)

--compression={*no|yes*}

compression can be *no* (default) or *yes*

A lossless compression can be applied for conservation purposes, in order to reduce the needed storage, typically between one and two thirds, depending on the image content. The wavelet

compression used is possibly a wee bit better than HuffYUV, FFV1 or JPEG2000 in terms of both speed and compression rate. Remember that in general a higher resolution means a less good compression, because of the increased noise.

The uncompressed format is always faster for restoration, because any compression would slow down significantly the image processing.

--lut=[channel=]path

path to an 1D LUT or to a 3D LUT to apply (default is no LUT)

LUTs can be applied to the input file and/or the output file. Moreover a LUT can be applied to the whole file or only to a single *channel*.

This option may be repeated.

For 1D LUT, which transforms e.g. from floating-point scene linear into camera log or a display-referred space, the maximum allowed size is currently 16'777'216, i.e. 24-bit precision.)

--bayer2rgb

transform a Bayer-encoded *input_file* into a half-resolution RGB *output_file*

This allows to generate a full RGB file with a Bayer sensor. This new option is a proof of concept that will probably evolve in future.

OTHER OPTIONS

-h, --help

display a help message

-v, --version

display the running version

SEE ALSO

movimdec(1) and **movimplay(1)**; **libmovim(1)** and **movim(1)**; **openmovim(1)**.

COPYRIGHT

Copyright (c) 2014–2019 by Reto Kromer

LICENSE

The **openMovIm** package is released under a 3-Clause BSD License.

DISCLAIMER

The **openMovIm** package is provided "as is" without warranty or support of any kind.