High Dynamic Range Imaging - Documentation for Tone Reproduction Operators

Erik Reinhard

May 13, 2005

1 Introduction

This documentation pertains to the source code and binaries available on the DVD-ROM that accompanies the book "High Dynamic Range Imaging" [20]. Before compiling or using this code, please refer to the licence information contained in the `licence.txt` file.

In the following section, the procedure for compiling and installing the source is explained. The general use of all operators is detailed in Section 3, while operator-specific command-line flags are explained for each operator in the remaining sections.

2 Installation

For convenience, binaries for Windows, Linux and Mac OS X are provided in the `binaries` directory on this DVD. The source code is available in the `sources` directory.

To compile the source code, gunzip and untar the distribution using

```
tar -zxvf tm_source.tar.gz
```

Then in the `tm_source/` directory, edit the `Makefile` to suit your system. Follow the instruction in the `Makefile`. Before compiling the sources, please check that all the libraries that this distribution depends on, are installed. If not, install those libraries first. In particular, the availability of `libjpeg`, `libtiff`, `libfftw`, `libm`, `libz` and the OpenEXR libraries need to be present. Each of these libraries is available for free on the web or comes pre-installed with your system.

In addition, this distribution comes with a precompiled library to read and write HDR images in JPEG format. This library is courtesy Sunnybrook Technologies, and may not be redistributed without permission. See `tm_source/jpghdr/include/jpeghdr.h` for further information. If this library is not available for your platform, please comment out the relevant parts of the `Makefile` in the `tm_source` directory by following the instructions given in that file.

Once all libraries are available, in the `tm_source` directory, type

```
make
make install
```

To see all operators in action, and to verify that everything has compiled correctly, type

```
make test
```

Note that several operators take a significant amount of time to complete, so that the above command may take several minutes to execute. This command also assumes the presence of the xv display program. If this program is not available on your system, replace xv with your favorite jpeg viewer in the Makefile.

3 Operation

All operators included in this distribution require at least an input file and an output file to be specified. For example, the photographic operator [19] may be called with:

```
tm_photographic -i <infile> -o <outfile>
```

All operators recognize the -i <infile> and -o <outfile> command line options. The input file has to be a high dynamic range image, given in either Radiance .hdr/.pic/.rgbe, TIFF .tif, JPG HDR .jpg or OpenExr .exr format. The output can be either of these formats, or can additionally be specified as a low dynamic range PPM .ppm or JPEG .jpg format. The file type for both input and output is recognized by the file extension. Thus, an input given in Radiance format and an output in JPEG format can be specified as:

```
tm_photographic -i hdr_image.hdr -o ldr_image.jpg
```

Although a standard gamma correction value is compiled in (and can be adjusted for your system in the top level Makefile in the tm_source directory), unless otherwise noted, each operator also parses the command line for the -gamma <val> flag. To tonemap an OpenEXR into a PPM file suitable for display on a monitor with a gamma value of 2.8, the following command may be given:

```
tm_photographic -i hdr_image.exr -o ldr_image.ppm -gamma 2.8
```

To disable gamma correction, use a value of 1.

Some operators include a -saturation <val> command line option to allow adjustment of the saturation. This option is provided for convenience and is not part of the original operators. The mechanism employed is described by Fattal et al [8]. It may be disabled by specifying -saturation <1>.

Some operators also include a -factor <val> which allows the input to be scaled before applying the specific operator. This facility is not part of the original operators and is also provided for convenience. The intention of this parameter is to bring the input to real-world units for images that are arbitrarily scaled. To disable this option, specify -factor <1>.

All other command line options are specific to each individual operator, and are discussed in the following sections. For a detailed understanding of the use and meaning of each of these operators, please see the individual papers referenced in each section.

4 Photographic operator, `tm_photographic`

The command line options for the photographic tone reproduction operator [19, 17] are:

-auto <> Estimate all user parameters using a parameter estimation technique [17]. This option is recommended and usually produces reasonable output.

-white <**val**> White value. This option should be set to a very large number ($1e20$ is default) for high dynamic range images, and should only be set to a small value if the input is already low dynamic range. All input luminances larger than `val` are mapped to white.

-key <**val**> The estimated key of the scene. The log average luminance of the input is mapped to this value. A reasonable initial value is 0.18. If this produces an image that is too dark, try doubling this value. If the image is too light, try halving this value.

-scale <> By default, this operator uses a fast and efficient global mapping. For extra fidelity (and computation time), the `-scale` option may be given, which invokes the multi-level scale selection mechanism. For some high dynamic range images, this produces better contrast.

5 Photoreceptor-based operator, `tm_photoreceptor`

This operator mimics to some extend the behavior of photoreceptors and includes a rudimentary color correction scheme [18]. The options are:

-a <**val**> Light adaptation parameter which controls the amount of contrast. The range of this parameters is between 0 and 1.

-c <**val**> Chromatic adaptation parameter, controlling the amount of color correction according to the grey-world assumption. The range of this parameter is between 0 and 1.

-f <**val**> Scale factor for the semi-saturation constant. This parameter affects how overall light or dark the image is.

-m <**val**> Exponent for the semi-saturation constant. This parameter affects contrast.

For many images, it is advised to adjust the parameters `-a` and `-c` only. If no satisfactory result can be obtained with these two parameters, additional adjustments to `-f` and `-m` may further improve the results.

6 Histogram adjustment, `tm_histadj`

This operator implements the histogram adjustment technique [25]. This is a partial implementation which does not include simulations of human contrast sensitivity, mesopic color sensitivity, acuity or veiling glare. To access those features, it is recommended to download the Radiance lighting simulation package and use `pcond` instead.

The operator implemented here uses a fast encoding scheme when .hdr or .tif input is given. For other file formats, it uses the same internal data structure as all the other operators in this distribution, and is therefore slower for these formats.

There are no operator-specific flags for histogram adjustment.

7 Bilateral filtering, `tm_bilateral`

This operator implements Durand and Dorsey's tone reproduction operator which is based on bilateral filtering [5]. It includes the speed-up mechanism outlined in their paper. The options for this operator are:

-s <**val**> Standard variation for the spatial Gaussian.

-r <**val**> Standard variation for the intensity domain Gaussian.

-contrast <**val**> Desired contrast of the base layer (in log units).

-shift <**val**> Constant added to the base layer (in log units).

-down <**val**> Down sampling factor. Higher numbers give faster results, but if this value is chosen too high, artifacts may start to appear.

-saturation <**val**> Saturation may be adjusted with this parameter. Set to 1 to keep saturation unchanged.

8 Trilateral filtering, `tm_trilateral`

Trilateral filtering is an extension to the bilateral filter [3]. This implementation does not include the optimizations proposed by Durand and Dorsey [5]. It carries out the convolution in the spatial domain and is therefore much slower than the bilateral filter. The user parameters are:

-sigma <**val**> Main user parameter controlling compression.

-contrast <**val**> Desired contrast of the base layer (in log units).

-shift <**val**> Constant added to the base layer (in log units).

-saturation <**val**> Saturation may be adjusted with this parameter. Set to 1 to keep saturation unchanged.

9 Ashikhmin's operator, `tm_ashikhmin`

This operator is similar to the photographic operator, although rooted in human visual perception [1]. The parameters are:

-threshold <**val**> The threshold used in the scale selection mechanism

-factor <**val**> Prescale factor. The input is scaled by this factor before applying the operator. This may be used to create input in real-world units if the input is given in arbitrary units.

-saturation <**val**> Saturation may be adjusted with this parameter. Set to 1 to keep saturation unchanged.

10 Tumblin-Rushmeier operator, `tm_tr`

This implementation is for the revised operator [22] of the classic operator [23]. This operator includes its own method for correcting for display gamma, and the default `-gamma` <`val`> is therefore not supported on the command line. The command line options are:

-factor <**val**> Prescale factor. The input is scaled by this factor before applying the operator. This may be used to create input in real-world units if the input is given in arbitrary units.

-saturation <**val**> Saturation may be adjusted with this parameter. Set to 1 to keep saturation unchanged.

11 Ward's contrast-based scale factor, `tm_ward`

This operator provides a linear scaling of the input to fit the limitations of the display device [24]. The command line options are:

-factor <**val**> Prescale factor. The input is scaled by this factor before applying the operator. This may be used to create input in real-world units if the input is given in arbitrary units.

-ldmax <**val**> Maximum luminance of the target display device. The default value is 100 cd/m^2.

12 Ferwerda's scale factor, `tm_ferwerda`

This linear scale factor also includes a simulation of scotopic lighting conditions [10]. The implementation given here is partial, because the loss of visual acuity modeled in the original operator, is omitted here. The command line options are:

-factor <**val**> Prescale factor. The input is scaled by this factor before applying the operator. This may be used to create input in real-world units if the input is given in arbitrary units.

-ldmax <**val**> Maximum luminance of the target display device. The default value is 100 cd/m^2.

13 Ferschin's exponential mapping, `tm_ferschin`

The exponential mapping implemented here [9], does not have user controls. By default this operator is anchored to the average of the scene. This may be overridden, and the maximum luminance in the scene may be used instead by specifying:

-max <> Use maximum scene luminance to anchor the computation.

14 Pattanaik's multi-scale observer model, `tm_mom`

This is a partial implementation of the multi-scale observer model [14]. Achromatic response is not computed. User parameters are:

-k <**val**> Kernel size of Gaussian filter

-factor <**val**> Prescale factor. The input is scaled by this factor before applying the operator. This may be used to create input in real-world units if the input is given in arbitrary units.

15 iCAM, `tm_icam`

The iCAM color appearance model [7, 6] takes the following user parameters:

-variance <**val**> Variance of the low-pass filter used in the first step of the model.

-variance2 <**val**> Variance of the low-pass filter used in the second step of the model.

-D <**val**> Degree of adaptation.

-factor <**val**> Prescale factor. The input is scaled by this factor before applying the operator. This may be used to create input in real-world units if the input is given in arbitrary units.

-percentile <**val**> Clamp a percentage of <`val`> luminance values after mapping.

16 Chiu's spatially variant operator, `tm_chiu`

This spatially variant operator divides the input by a Gaussian blurred version of the input [2]. Available parameters are:

-variance <**val**> Variance of the Gaussian filter kernel.

-iter <**val**> Number of smoothing iterations to apply after running the tone reproduction operator.

-factor <**val**> Prescale factor. The input is scaled by this factor before applying the operator. This may be used to create input in real-world units if the input is given in arbitrary units.

-saturation <**val**> Saturation may be adjusted with this parameter. Set to 1 to keep saturation unchanged.

17 Logarithmic mapping, `tm_log`

Logarithmic mapping with the following user parameters:

-contrast <**val**> In the logarithmic domain, the image may be scaled to have <`val`> units of dynamic range.

-shift <**val**> Constant added in logarithmic domain.

-saturation <**val**> Saturation may be adjusted with this parameter. Set to 1 to keep saturation unchanged.

18 Drago's logarithmic mapping, `tm_drago`

Drago's mapping is logarithmic, but the base of the logarithm is adjusted based on the intensity of the pixel [4]. User parameters are:

-b <**val**> Parameter controlling the amount of compression. By default this is set to 0.85.

-saturation <**val**> Saturation may be adjusted with this parameter. Set to 1 to keep saturation unchanged.

19 Schlick's uniform rational quantization, `tm_schlick`

Schlick's uniform rational quantization requires two user parameters [21]:

-M <**val**> Just noticeable difference in quantized display luminance steps.

-ck <**val**> User parameter k, should be between 0 and 1.

20 Retinex, `tm_retinex`

An interpretation of Rahman and Jobson's retinex simulation [16]. Note that this implementation may be incomplete due to a lack of sufficient details in the original papers. The parameters provided for experimentation are:

-w <**val**> Weight factor.

-s <**val**> Scale factor.

-low <**val**> Smallest kernel size in pixels (default 2 pixels).

-high <**val**> Largest kernel size in pixels (default 64 pixels).

-scale <**val**> Number of levels in the Gaussian pyramid (default 6 levels).

-saturation <**val**> Saturation may be adjusted with this parameter. Set to 1 to keep saturation unchanged.

21 Horn's lightness computation, `tm_horn`

A rendition of Horn's lightness computations [11]. User parameters are as follows:

-threshold <**val**> Gradient threshold

-lightness <> Normally, gradients larger than the threshold value are clamped to 0. The `-lightness` flag reverses this so that gradients smaller than the threshold are clamped to 0.

22 Gradient domain compression, `tm_gdc`

Gradient domain compression [8] with the following user parameters:

-alpha <**val**> The parameter α is used to steer which gradient magnitudes are attenuated.

-beta <**val**> The parameter β controls by how much gradients are attenuated.

-cmin <**val**> Percentile of pixels being clamped at the dark end.

-cmax <**val**> Percentile of pixels being clamped at the light end.

-saturation <**val**> Saturation may be adjusted with this parameter. Set to 1 to keep saturation unchanged.

23 Oppenheim's frequency domain operator, `tm_oppenheim`

Attenuation for Oppenheim's operator is applied in the Fourier domain [13]. User parameters are:

-k <**val**> The shape of the compressive function is sigmoidal, and controlled by k.

-comp <**val**> Interpolation factor controlling the weight of the sigmoidal compression function relative to linear scaling. This parameter should be between 0 and 1.

-saturation <**val**> Saturation may be adjusted with this parameter. Set to 1 to keep saturation unchanged.

24 Yee's segmentation-based technique, `tm_yee`

This is a partial implementation of Yee's segmentation-based operator [26] with the following user parameters:

-f <**val**> Factor by which the adaptation is multiplied.

-exp <**val**> Exponent for semi-saturation constant.

-global <> By default, a local operator is invoked. A global version is called if this flag is specified.

-layers <**val**> Number of layers to use.

25 Pattanaik's visual adaptation for time-dependent display, `tm_pattanaik`

This is a partial implementation of the visual adaptation operator for time-dependent display [15]. The time-dependent component is not implemented, making this operator suitable for single images. The user parameters are:

-ambient <**val**> Estimation of ambient component.

-scale <**val**> Scale factor for scaling the input.

26 Miller's operator, `tm_miller`

Miller's operator is suitable for indoor scenes [12]. The user parameters are:

-max <**val**> Clamp all values larger than <`val`>.

-factor <**val**> After clamping, scale the input according this this factor.

27 Linear scaling, `tm_linear`

Linear scaling is included for comparison. It can also be used for simple file conversions. The user parameters are:

-factor $<$**val**$>$ Scale factor.

-low $<$**val**$>$ After scaling, clamp pixels below this value to this value.

-high $<$**val**$>$ After scaling, clamp pixels above this value to this value.

-noop $<>$ Do not apply scaling. Use this to read an image and write it in a different file format.

References

[1] Michael Ashikhmin. A tone mapping algorithm for high contrast images. In *Proceedings of 13th Eurographics Workshop on Rendering*, pages 145–155, 2002.

[2] K. Chiu, M. Herf, P. Shirley, S. Swamy, C. Wang, and K. Zimmerman. Spatially nonuniform scaling functions for high contrast images. In *Proceedings of Graphics Interface '93*, pages 245–253, May 1993.

[3] Prasun Choudhury and Jack Tumblin. The trilateral filter for high contrast images and meshes. In *Proceedings of the Eurographics Symposium on Rendering*, pages 186–196, 2003.

[4] Frédéric Drago, Karol Myszkowski, Thomas Annen, and Norishige Chiba. Adaptive logarithmic mapping for displaying high contrast scenes. *Computer Graphics Forum*, 22(3), 2003.

[5] Frédo Durand and Julie Dorsey. Fast bilateral filtering for the display of high-dynamic-range images. *ACM Transactions on Graphics*, 21(3):257–266, 2002.

[6] M D Fairchild and G M Johnson. The iCAM framework for image appearance, image differences, and image quality. *Journal of Electronic Imaging*, 2004.

[7] Mark D Fairchild and G M Johnson. Meet iCAM: an image color appearance model. In *IS&T/SID* 10^{th} *Color Imaging Conference*, pages 33–38, Scottsdale, 2002.

[8] Raanan Fattal, Dani Lischinski, and Michael Werman. Gradient domain high dynamic range compression. *ACM Transactions on Graphics*, 21(3):249–256, 2002.

[9] P Ferschin, I Tastl, and W Purgathofer. A comparison of techniques for the transformation of radiosity values to monitor colors. In *First IEEE International Conference on Image Processing*, pages 13–16, 1994.

[10] James A. Ferwerda, Sumant Pattanaik, Peter Shirley, and Donald P. Greenberg. A model of visual adaptation for realistic image synthesis. In *SIGGRAPH 96 Conference Proceedings*, pages 249–258, August 1996.

[11] Berthold K P Horn. Determining lightness from an image. *CVGIP*, 3:277–299, 1974.

[12] Naomi Johnson Miller, Peter Y Ngai, and David D Miller. The application of computer graphics in lighting design. *Journal of the IES*, 14:6–26, October 1984.

[13] Alan V. Oppenheim, Ronald Schafer, and Thomas Stockham. Nonlinear filtering of multiplied and convolved signals. *Proceedings of the IEEE*, 56(8):1264–1291, 1968.

[14] Sumanta N. Pattanaik, James A. Ferwerda, Mark D. Fairchild, and Donald P. Greenberg. A multiscale model of adaptation and spatial vision for realistic image display. In *SIGGRAPH 98 Conference Proceedings*, pages 287–298, July 1998.

[15] Sumanta N. Pattanaik, Jack Tumblin, Hector Yee, and Donald P. Greenberg. Time-dependent visual adaptation for fast realistic display. In *SIGGRAPH 2000 Conference Proceedings*, pages 47–54, July 2000.

[16] Zia-ur Rahman, Daniel J Jobson, and Glenn A Woodell. A multiscale retinex for color rendition and dynamic range compression. In *SPIE Proceedings: Applications of Digital Image Processing XIX*, volume 2847, 1996.

[17] Erik Reinhard. Parameter estimation for photographic tone reproduction. *Journal of Graphics Tools*, 7(1):45–51, 2003.

[18] Erik Reinhard and Kate Devlin. Dynamic range reduction inspired by photoreceptor physiology. *IEEE Transactions on Visualization and Computer Graphics*, 11(1):13–24, January/February 2005.

[19] Erik Reinhard, Michael Stark, Peter Shirley, and Jim Ferwerda. Photographic tone reproduction for digital images. *ACM Transactions on Graphics*, 21(3):267–276, 2002.

[20] Erik Reinhard, Greg Ward, Sumanta Pattanaik, and Paul Debevec. *High Dynamic Range Imaging*. Morgan Kaufmann Publishers, San Francisco, 2005.

[21] Christophe Schlick. Quantization techniques for the visualization of high dynamic range pictures. In Peter Shirley, Georgios Sakas, and Stefan Müller, editors, *Photorealistic Rendering Techniques*, pages 7–20. Springer-Verlag Berlin Heidelberg New York, 1994.

[22] Jack Tumblin, Jessica K. Hodgins, and Brian K. Guenter. Two methods for display of high contrast images. *ACM Transactions on Graphics*, 18 (1):56–94, 1999.

[23] Jack Tumblin and Holly Rushmeier. Tone reproduction for computer generated images. *IEEE Computer Graphics and Applications*, 13(6):42–48, November 1993.

[24] Greg Ward. A contrast-based scalefactor for luminance display. In Paul Heckbert, editor, *Graphics Gems IV*, pages 415–421. Academic Press, Boston, 1994.

[25] Greg Ward, Holly Rushmeier, and Christine Piatko. A visibility matching tone reproduction operator for high dynamic range scenes. *IEEE Transactions on Visualization and Computer Graphics*, 3(4), 1997.

[26] H Yee and S N Pattanaik. Segmentation and adaptive assimilation for detail-preserving display of high-dynamic range images. *The Visual Computer*, 19(7–8), 2003.