21.2: Invited Paper: An Overview of Dynamic Range Reduction

Erik Reinhard
University of Bristol and University of Central Florida

Abstract

With a surge of interest in high dynamic range imaging, techniques for displaying such images on conventional display devices are gaining in importance. Although the last review of dynamic range reduction algorithms appeared about a year ago, the advances in this field are rapid. In this paper, the current state-of-the-art in dynamic range reduction is reviewed, with an emphasis on sigmoidal compression.

1 Introduction

Real-world environments typically contain a range of illumination much larger than can be represented by conventional 8-bit images. For instance, sunlight at noon may be as much as 100 million times brighter than starlight [8, 21]. The human visual system is able to detect 4 or 5 log units of illumination simultaneously, and can adapt to a range of around 10 orders of magnitude over time [8].

On the other hand, conventional 8-bit images with values between 0 and 255, have a useful dynamic range of around 2 orders of magnitude. Such images are represented typically by one byte per pixel for each of the red, green and blue channels. The limited dynamic range afforded by 8-bit images is well-matched to the display capabilities of CRTs. Their range, while being larger than 2 orders of magnitude, lies partially in the dark end of the range where the human vision has trouble discerning very small differences under normal viewing circumstances. Hence, CRTs have a useful dynamic range of 2 log units of magnitude.

The contrast of LCD devices is normally not much greater, although they can be made much brighter. As a result, both the black level and the white level are much higher, yielding once more a dynamic range of around 2 log units. Currently, very few display devices have a dynamic range that significantly exceeds this range, the notable exception being LCD displays with an LED back-panel where each of the LEDs is separately addressable [20].

Capturing the full dynamic range of a scene implies that in many instances the resulting high dynamic range (HDR) image cannot be directly displayed, as its range is likely to exceed the 2 orders of magnitude range afforded by conventional display devices.

There are two strategies to display high dynamic range images. First, we may develop display devices which can directly accommodate high dynamic range imagery [20]. Second, high dynamic range images may be prepared for display on low dynamic range display devices by applying a *tone reproduction operator* [18]. The purpose of tone reproduction is therefore to reduce the dynamic range of an image such that it may be displayed on a conventional display device, which includes printers. The key issue in tone reproduction is to compress an image while at the same time preserving one or more attributes of the image, such as contrast, visible detail, brightness, or appearance.

2 Spatial operators

In the following sections we discuss tone reproduction operators which apply compression directly on pixels. Often global and local operators are distinguished. Tone reproduction operators in the former class change each pixel's luminance values according to a compressive function which is the same for each pixel [19, 22, 24]. The term global stems from the fact that many such functions need

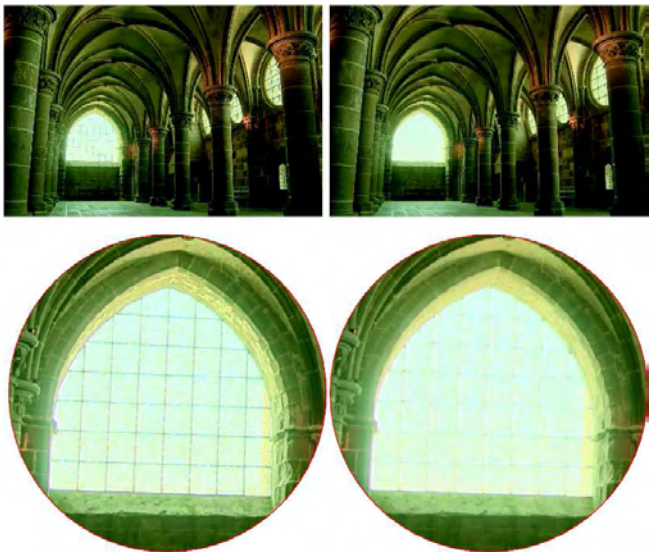

Fig 1. A local tone reproduction operator (left) and a global tone reproduction operator (right) [17]. The local operator shows more detail, as for instance seen in the insets.

to be anchored to some values that are computed by analyzing the full image. In practice most operators use the geometric average $\bar{L}_v$ to steer the compression:

$$\bar{L}_v = \exp\left(\frac{1}{N}\sum_{x,y}\log(\delta + L_v(x,y)\right) \tag{1}$$

The small constant δ is introduced to prevent the average to become zero in the presence of black pixels. The geometric average is normally mapped to a predefined display value. The main challenge faced in the design of a global operator lies in the choice of compressive function.

On the other hand, local operators compress each pixel according to a specific compression function which is modulated by information derived from a selection of neighboring pixels, rather than the full image [1, 2, 6, 7, 15, 17]. The rationale is that the brightness of a pixel in a light neighborhood is different than the brightness of a pixel in a dark neighborhood. Design challenges for local operators involve choosing the compressive function, the size of the local neighborhood for each pixel, and the manner in which local pixel values are used. In general, local operators are able to achieve better compression than global operators (Figure 1), albeit at a higher computational cost.

Most operators employ one of two distinct compressive functions. Display values $L_d(x,y)$ are most commonly derived from image luminances $L_v(x,y)$ by the following two functional forms:

$$L_d(x,y) = \frac{L_v(x,y)}{f(x,y)} \tag{2a}$$

$$L_d(x,y) = \frac{L_v^n(x,y)}{L_v^n(x,y) + g^n(x,y)} \tag{2b}$$

Fig 2. Halos are artifacts commonly associated with local tone reproduction operators. Chiu's operator is used here without smoothing iterations to demonstrate the effect of division (left).

In these equations, $f(x, y)$ and $g(x, y)$ may either be constant, or a function which varies per pixel. In the former case, we have a global operator, whereas a spatially varying function results in a local operator. The exponent n is a constant which is either fixed, or set differently per image.

Equation 2a divides each pixel's luminance by a value derived from either the full image or a local neighborhood. As an example, the substitution $f(x, y) = L^{\max}/255$ in (2a) yields a linear scaling such that values may be directly quantized into a byte, and can therefore be displayed. A different approach would be to substitute $f(x, y) = L^{\text{blur}}(x, y)$, i.e. divide each pixel by a weighted local average, perhaps obtained by applying a Gaussian filter to the image [2]. While this local operator yields a displayable image, it produces halos, as demonstrated in Figure 2.

The cause of halos stems from the fact that Gaussian filters blur across sharp contrast edges in the same way that they blur small details. If there is a large contrast gradient in the neighborhood of the pixel under consideration, this causes the Gaussian blurred pixel to be significantly different from the pixel itself. By using a large filter kernel in a division-based approach such large contrasts are averaged out, and the occurrence of halos can be minimized. However, very large filter kernels tend to compute a local average that is not substantially different from the global average. The size of the filter kernel in division-based operators presents a trade-off between the ability to reduce the dynamic range, and the visibility of artifacts.

3 Sigmoidal Compression

Equation 2b has an S-shaped curve on a log-linear plot, and is called a sigmoid for that reason. This functional form fits data obtained from measuring the electrical response of photoreceptors to flashes of light in various species [14]. It has also provided a good fit to other electro-physiological and psychophysical measurements of human visual function [9–11].

Sigmoids have several desirable properties. For very small luminance values the mapping is approximately linear, so that contrast is preserved in dark areas of the image. The function has an asymptote at one, which means that the output mapping is always bounded between 0 and 1. A further advantage of this function is that for intermediate values, the function affords an approximately logarithmic compression. This can be seen for instance in Figure 3, where the middle section of the curve is approximately linear on a log-linear plot. To illustrate, both $L_d = L_w/(L_w + 1)$ and $L_d = 0.25 \log(L_w) + 0.5$ are plotted in this figure [1], showing that

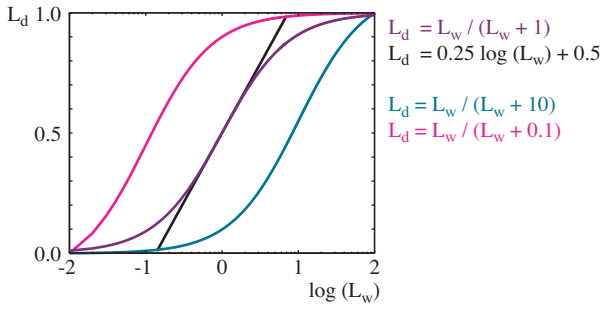

Fig 3. Over the middle range values, sigmoidal compression is approximately logarithmic. The choice of semi-saturation constant determines how input values are mapped to display values.

these functions are very similar over a range centered around 1.

In Equation 2b, the function $g(x, y)$ may be computed as a global constant, or as a spatially varying function. Following common practice in electro-physiology, we call $g(x, y)$ the "semi-saturation" constant. Its value determines which values in the input image are optimally visible after tone mapping, as shown in Figure 3. The effect of choosing different semi-saturation constants is also shown in this figure.

In its simplest form, $g(x, y)$ is set to $\bar{L}_v/k$, so that the geometric average is mapped to user parameter k [17]. In this case, a good initial value for k is 0.18, which conforms to the photographic equivalent of middle gray.

The exponent n in Equation 2b determines how pronounced the S-shape of the sigmoid is. Steeper curves map a smaller useful range of scene values to the display range, whereas shallower curves map a larger range of input values to the display range.

The several different variants of sigmoidal compression shown above are all global in nature. This has the advantage that they are fast to compute, while they are also amenable to implementation on graphics hardware. For very high dynamic range images, however, it may be necessary to resort to a local operator since this may give better compression.

A straightforward method to extend sigmoidal compression replaces the global semi-saturation constant by a spatially varying function. Thus, $g(x, y)$ then becomes a function of a spatially localized average, for instance by using a Gaussian blurred image. Each pixel in a blurred image represents a locally averaged value.

In local operators, halo artifacts occur when the local average is computed over a region that contains sharp contrasts with respect to the pixel under consideration. It is therefore important that the local average is computed over pixel values that are not significantly different from the pixel that is being filtered.

This suggests a filtering strategy whereby no blurring over such edges occurs. A simple, but computationally expensive way is to compute a stack of Gaussian blurred images with different kernel sizes, i.e. an *image pyramid*. For each pixel, we may choose the largest Gaussian which does not overlap with a significant gradient.

Once for each pixel the appropriate neighborhood is known, the Gaussian blurred average L_{blur} for this neighborhood may be used to steer the semi-saturation constant, such as for instance employed by the photographic tone reproduction operator [17]:

$$L_{\text{d}} = \frac{L_{\text{w}}}{1 + L_{\text{blur}}} \tag{3}$$

It is instructive to compare the result of this operator with its global

[1] The constants 0.25 and 0.5 were determined by equating the values as well as the derivatives of both functions for $x = 1$.

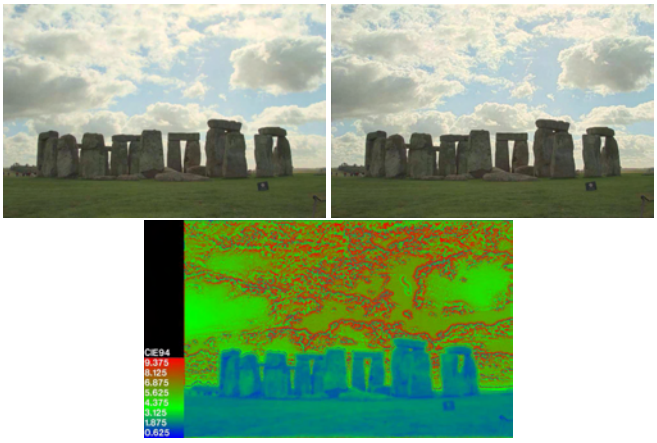

Fig 4. This image was tonemapped with both global and local versions of the photographic tone reproduction operator (top left and right). Below, the CIE94 color difference is shown.

equivalent, which is defined as:

$$L_{\mathrm{d}} = \frac{L_{\mathrm{w}}}{1 + L_{\mathrm{w}}} \tag{4}$$

Images tonemapped with both forms are shown in Figure 4. The CIE94 color difference metric shown in this figure shows that the main differences occur near (but not precisely at) high-frequency high-contrast edges, predominantly seen in the clouds. These are the regions where more detail is produced by the local operator.

An alternative approach includes the use of edge preserving smoothing operators, which are designed specifically for removing small details while keeping sharp contrasts in tact. Such filters have the advantage that sharp discontinuities in the filtered result coincide with the same sharp discontinuities in the input image, and may therefore help to prevent halos [5]. Several such filters, such as the bilateral filter, trilateral filter, Susan filter, the LCIS algorithm and the mean shift algorithm are suitable [3, 4, 6, 16, 23], although some of them are expensive to compute. Image pyramids can be used directly for the purpose of tone reproduction, provided the filter bank is designed carefully [12].

4 Mappings for HDR Displays

The techniques for tone reproduction discussed so far are all geared toward the display of images on low dynamic range display devices. Specific algorithms to prepare images for display on HDR devices have not yet appeared. However, we can speculate about the features that such algorithms are likely to exhibit.

As opposed to conventional displays, HDR display devices emit enough light to be sufficiently different from the average room lighting conditions. An important, yet poorly understood issue is that the human visual system (HVS) will adapt in part to the device and in part to the room environment. Such partial adaptation is notoriously difficult to model, and is certainly not a feature of current tone reproduction algorithms. Nonetheless, a good tone reproduction algorithm for HDR display devices would have to account for partial adaptation.

There is, unfortunately, a second problem with most tone reproduction operators. Many operators are modeled after some aspects of human vision. The computed display values therefore essentially represent perceived quantities, for instance brightness if the tone reproduction operator is based on a model of brightness perception. If we assume that the model is an accurate representation of some aspect of the HVS, then displaying the image and observing it will cause the HVS to interpret these perceived values as luminance values.

The HVS thus applies a second perceptual transform on top of the one applied by the algorithm. This is formally incorrect. A good tone reproduction operator should follow the same common practice as employed in color appearance modeling, and apply both a forward and a backward transform. Some algorithms do takes this approach [13, 15, 23], although most do not.

The forward transform can be any algorithm thought to be effective at compressing luminance values. The backward transform will then apply the algorithm in reverse, but with display parameters inserted. This approach compresses luminance values into perceived values, while the reverse algorithm will convert the perceived values back into luminance values. As an example, Equation (2b) should be rewritten as:

$$V(x,y) = \frac{L_v^n(x,y)^n}{L_v^n(x,y) + g^n(x,y)} \tag{5}$$

where V is a perceived value. The function g continues to return a globally or locally computed adaptation value, which is based on the image values. This equation then needs to be inverted, whereby g is replaced with a display adaptation value. For instance, we could try to replace g with the mean display luminance $L_{d,\mathrm{mean}}$:

$$L_d(x,y) = \exp\left(\frac{1}{m}\log\left(\frac{-V(x,y)\,L_{d,\mathrm{mean}}^m}{V(x,y)-1}\right)\right) \tag{6}$$

For a conventional display, we would set $L_{d,\mathrm{mean}}$ to 128. The exponent m is also a display related parameter and determines how display values are spread around the mean display luminance. For low dynamic range display devices, this value can be set to 1, thereby simplifying the above equation to:

$$L_d(x,y) = \frac{-V(x,y)\,L_{d,\mathrm{mean}}}{V(x,y)-1} \tag{7}$$

The computation of display values is now driven entirely by the mean luminance of the image, as well as the exponent n which specifies how large a range of values around the mean luminance will be visualized. As a result, the inverse transform may create display values that are outside the display range. These will have to be clamped.

For comparison, example images computed with and without the backward transform are shown in Figure 5. For the images computed without the backward transform, we compute display values using:

$$L'_d = (1+f)\,L_{d,\max} \tag{8}$$

The f term allows some of the light values in the image to go over the maximum display value $L_{d,\max}$. These values will subsequently be clamped. The purpose of this step is to clamp pixels such that the visual appearance of the with and without images matches as much as possible.

The images in Figure 5 show a trend in that application of the backward transform allows a better impression of bright objects. For instance, the reflection in the sea and the sun in the bottom image are better reproduced.

It is desirable that a tone reproduction operator does not alter an image that is already within the display range [5]. In the model proposed here this is implicitly achieved, as for $n = m$ and $g = L_{d,\mathrm{mean}}$, the backward transform is the true inverse of the forward transform. This is borne out in the CIE94 color difference metric, which is uniformly 0 for all pixels after running the algorithm twice.

For HDR display devices, we expect to be able to use the same image-dependent parameters n and g, but substitute new parameters $L_{d,\mathrm{mean}}$ and m that characterize the target display. Hence, the utility of this approach will increase when the dynamic range of display devices becomes more variable.

Fig 5. The images on the left are computed by applying the forward transform. The backward transform is also applied for the right images. Parameter values: $L_{d,\text{mean}} = 128$, $L_{d,\max} = 256$, $m = 1$, $f = 0.5$. Top images: $g = 0.18/L_{\text{av}}$ and $n = 0.8$. Bottom images: $g = 0.05/L_{\text{av}}$ and $n = 0.7$.

5 Discussion

Tone reproduction for low dynamic range display devices is nowadays a reasonably well understood problem. The majority of images can be compressed well enough for applications in photography and entertainment, and any other applications that do not critically depend on accuracy. Recent validation studies show that some algorithms perform well over a range of different tasks and displayed material.

When dealing with different displays, each having their own dynamic range, it becomes more important to consider tone reproduction operators that can be parameterized for both different types of images and different types of display. Following common practice in color appearance modeling, we have argued that both a forward and a backward transform are necessary. This is formally the correct approach to tone reproduction, but also appears to have direct advantages in terms of visual appearance.

Acknowledgments

Thanks to Edward Adelson, Ahmet Oğuz Akyüz, Michael Ashikhmin, Mark Colbert, Paul Debevec, Ranaan Fattal, Charles Hughes, Garrett Johnson, Erum Arif Khan, Grzegorz Krawczyk, Timo Kunkel, Yuanzhen Li, Dani Lischinski, Rafal Mantiuk, Karol Myszkowski, Helge Seetzen, Lavanya Sharan, Peter Shirley, and Greg Ward. for their help, discussions, and other direct or indirect contributions.

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, 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] Dorin Comaniciu and Peter Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[5] Jeffrey M DiCarlo and Brian A Wandell. Rendering high dynamic range images. In *Proceedings of the SPIE Electronic Imaging 2000 conference*, volume 3965, pages 392–401, 2000.

[6] 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.

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

[8] James A Ferwerda. Elements of early vision for computer graphics. *IEEE Computer Graphics and Applications*, 21(5):22–33, 2001.

[9] Donald C Hood and Marcia A Finkelstein. Comparison of changes in sensitivity and sensation: implications for the response-intensity function of the human photopic system. *Journal of Experimental Psychology: Human Perceptual Performance*, 5(3):391–405, 1979.

[10] Donald C Hood, Marcia A Finkelstein, and Eugene Buckingham. Psychophysical tests of models of the response function. *Vision Research*, 19(4):401–406, 1979.

[11] Jochen Kleinschmidt and John E Dowling. Intracellular recordings from gecko photoreceptors during light and dark adaptation. *Journal of General Physiology*, 66(5):617–648, 1975.

[12] Yuanzhen Li, Lavanya Sharan, and Edward H Adelson. Compressing and companding high dynamic range images with subband architectures. *ACM Transactions on Graphics*, 24(3):836–844, 2005.

[13] 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, 1984.

[14] K I Naka and W A H Rushton. S-potentials from luminosity units in the retina of fish (cyprinidae). *Journal of Physiology (London)*, 185(3):587–599, 1966.

[15] 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, 1998.

[16] Sumanta N Pattanaik and Hector Yee. Adaptive gain control for high dynamic range image display. In *Proceedings of Spring Conference in Computer Graphics (SCCG2002)*, pages 24–27, Budmerice, Slovak Republic, 2002.

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

[18] Erik Reinhard, Greg Ward, Sumanta Pattanaik, and Paul Debevec. *High Dynamic Range Imaging: Acquisition, Display and Image-Based Lighting*. Morgan Kaufmann Publishers, San Francisco, 2005.

[19] 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, Berlin, 1994. Springer-Verlag.

[20] Helge Seetzen, Wolfgang Heidrich, Wolfgang Stuerzlinger, Greg Ward, Lorne Whitehead, Matthew Trentacoste, Abhijeet Ghosh, and Andrejs Vorozcovs. High dynamic range display systems. *ACM Transactions on Graphics*, 23(3):760–768, 2004.

[21] L Spillmann and J S Werner, editors. *Visual perception: the neurological foundations*. Academic Press, San Diego, 1990.

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

[23] Jack Tumblin and Greg Turk. LCIS: A boundary hierarchy for detail-preserving contrast reduction. In Alyn Rockwood, editor, *SIGGRAPH 1999, Computer Graphics Proceedings*, pages 83–90, 1999.

[24] 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.