Rendering high dynamic range images

Jeffrey M. DiCarlo[*a] and Brian A. Wandell[a,b]

[a]Department of Electrical Engineering, Stanford University, CA 94305

[b]Department of Psychology, Stanford University, CA 94305

ABSTRACT

Within the image reproduction pipeline, display devices limit the maximum dynamic range. The dynamic range of natural scenes can exceed three orders of magnitude. Many films and newer camera sensors can capture this range, and people can discriminate intensities spanning this range. Display devices, however, such as CRTs, LCDs, and print media, are limited to a dynamic range of roughly one to two orders of magnitude.

In this paper, we review several algorithms that have been proposed to transform a high dynamic range image into a reduced dynamic range image that matches the general appearance of the original. We organize these algorithms into two categories: tone reproduction curves (TRCs) and tone reproduction operators (TROs). TRCs operate pointwise on the image data, making the algorithms simple and efficient. TROs use the spatial structure of the image data and attempt to preserve local image contrast.

The basic properties of algorithms from each class are described using both real and synthetic monochrome images. We find that TRCs have difficulty preserving local image contrast when the intensity distributions of the bright and dark regions of the image overlap. TROs, which are traditionally based on multiresolution decomposition algorithms such as Gaussian decomposition, better measure and preserve local image contrast. However, the decomposition process can lead to unwanted spatial artifacts. We describe an approach for reducing these artifacts using robust operators. Coupled with further analyses of the illumination distribution in natural scenes, we believe robust operators show promise for reducing unwanted artifacts in dynamic range compression algorithms.

Keywords: human vision, tone reproduction, adaptation, spatial vision, image processing, high dynamic range

1. INTRODUCTION

In this paper, we discuss principles and algorithms for transforming monochrome images that span a wide intensity range into images that span a much narrower intensity range. There are at least two types of applications that call for such algorithms. First, for a number of years it has been possible to create synthetic images that accurately represent scenes comprising very wide intensity ranges.[1,2] The synthetic image intensity range can significantly exceed the range that can be displayed on conventional devices, creating interest in algorithms that transform the synthetic image while retaining its visual impact. Second, advances in digital imaging technology have produced imaging systems that can acquire a very wide intensity range.[3,4] This provides an additional source of large intensity range data that need to be rendered on devices with a relatively narrow intensity range.

Figure 1 shows the location in the imaging pipeline for algorithms that reduce the intensity range. The pipeline is separated into five stages: original scene, acquisition device, algorithm, display device and observer. The objective of the imaging pipeline is to transform the original scene into a displayed image that appears similar. In imaging applications, the image rendered on a display is rarely a precise physical match to the original scene: display intensity ranges fall between 1 and 150 cd/m^2 while the image data usually fall well outside this range. Hence, the figure emphasizes that the rendering algorithm must maximize perceived similarity. There is one general principle that dominates how to match rendered and original images: the relative intensity ratios (image contrast) of the displayed and original images should match.[5] The theme of manipulating the relative intensity ratios will be central to the developments in this paper.

[*] Correspondence: Email: dicarlo@white.stanford.edu; Telephone: (650) 723-0993; Fax: (650) 723-0993.

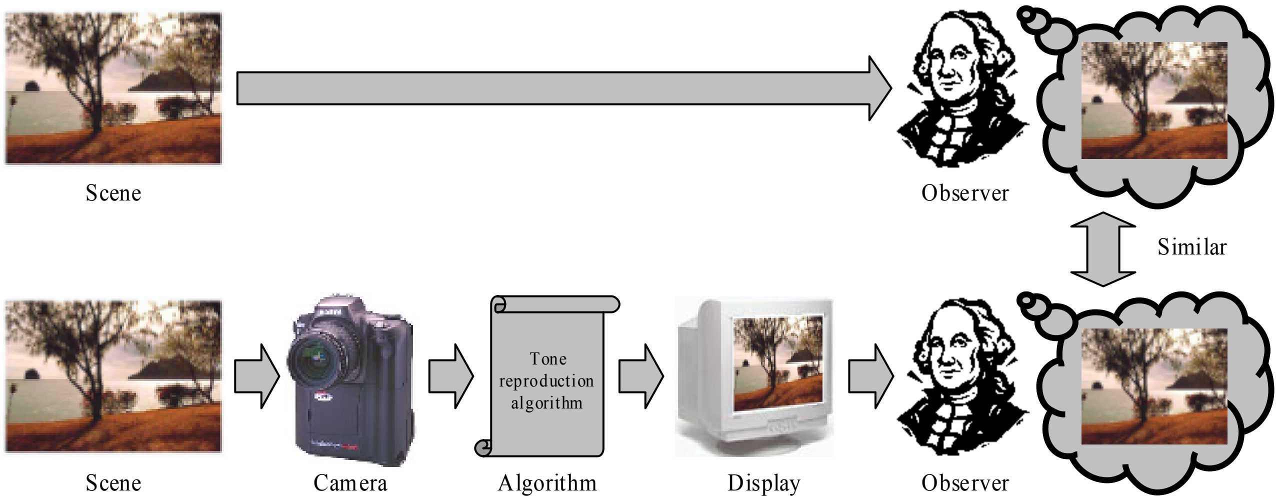

Figure 1. The location of tone reproduction algorithms that incorporate dynamic range compression in the image reproduction pipeline. The quality of the reproduction is evaluated using human visual metrics that compare the appearance of the reproduction to the appearance of the original.

2. DYNAMIC RANGE

Because of the importance of intensity ratios for image reproduction, it is natural to summarize the range of an image using a single extreme ratio between the maximum and minimum image intensities. This ratio is usually called the image dynamic range. A rendering of the image can preserve the original intensity ratios only if the dynamic range of every device within the image reproduction pipeline matches or exceeds that of the original scene. Comparing the dynamic range of the individual elements within the pipeline measures the source and scope of reproduction problems. The dynamic range of natural scenes exceeds three log units. Many films and newer cameras can capture this range, and people can discriminate intensities spanning this range. Display devices, however, such as CRTs, LCDs, and print media, are limited to a smaller dynamic range of roughly one to two log units.

In the description given above, the term dynamic range is a dimensionless quantity that is used to refer to several different physical measures. In the case of a natural image, the dynamic range is the ratio of the lightest to darkest point in the image. In the case of a camera and display, the dynamic range is a stimulus-referred quantity. For a camera, the dynamic range is the ratio of the intensities that just saturate the camera and that just lift the camera response one standard deviation above the camera noise. For a display, the dynamic range is the ratio of the maximum and minimum intensities emitted from the screen.

The concept of dynamic range is useful, but it is imperfect for summarizing several important aspects of image reproduction. Before analyzing dynamic range compression algorithms, two limitations of dynamic range as applied to image rendering should be emphasized.

2.1. Absolute intensity level

Dynamic range, by definition, eliminates any specification of the absolute intensity level. This is too extreme a summary for image rendering applications; the absolute level matters because imaging devices and observers have upper and lower bounds on their operating range. For example, suppose that an observer sees a scene through a very dark neutral density filter. The filter preserves the scene dynamic range, yet the low intensity level changes the sensitivity as well as the spatial and temporal sensitivity of the human visual system.[6] If the absolute level is scaled too much, the entire image can move from photopic (cone) to scotopic (rod) vision.

Because absolute level matters, the dynamic range of the reproduction system may not match the dynamic range of any individual device. Specifically, one device in the imaging pipeline can set the system's lower bound, while a different device sets the upper bound. For example, while viewing an image on a monitor, the observer may not be able to perceive differences among the darkest levels of the display.[7] The display, however, cannot reach the upper intensity levels that are

Figure 2. Image dynamic range reduction. (A) The original image. When rendered on a calibrated monitor, the image has a dynamic range of 1.08 log units. (B) The same image is shown, but now rendered on a monitor with a dynamic range of 0.78. Low intensity image points were clipped to compress the dynamic range. (C) The original image is rendered again with a dynamic range of 0.78, but this time the high intensity image points were clipped. See text for details.

within the observer's capabilities. In this typical reproduction pipeline, the effective dynamic range of the monitor and observer together, is lower than either one alone. Hence, we must consider the dynamic range of the entire imaging pipeline, a concept that we call the *system dynamic range*.

2.2. Visual sensitivity

As we already observed, it is widely agreed that image reproduction algorithms should aim to preserve intensity ratios. When preserving all ratios is impossible, certain ones are more visually significant than others and these should be preserved first. For example, it is difficult to perceive the difference between an edge whose two sides differ by 3.8 log units from an edge whose sides differ by 4 log units. On the other hand, it is very easy to discriminate between a pair of edges whose sides have intensity ratios of 0.1 log units and 0.3 log units. Hence, preserving the latter intensity ratio is much more important than the former. Visual insensitivity to image differences is always an important factor in designing algorithms, and we should rely on such insensitivity in dynamic range compression design as well.

Another significant visual factor related to dynamic range is illustrated in Figure 2. An original image is rendered in Panel A. The same image is rendered at half the original dynamic range in Panels B and C. In Panel B, the dynamic range is halved by increasing the intensity of the darkest pixels to twice the minimum image intensity. In Panel C, the dynamic range is halved by reducing the intensity of the brightest pixels to half of the maximum intensity. Panel A and Panel B, although rendered with a different dynamic range, look very similar. Panels B and C were rendered with the same dynamic range, but they look very different. Describing the image transformations simply in terms of dynamic range does not capture the visual impact because the intensity ratios that are preserved in Panel B are visually more important than the intensity ratios in Panel C.

With these caveats in mind, it remains roughly true that preserving intensity ratios is a good first principle for image rendering. Using system dynamic range as a measure, we can determine in advance whether preserving intensity ratios is possible. If the original scene has a dynamic range that is less than or equal to the system dynamic range, we can preserve intensity ratios. Rendering the image in this scenario is a simple process. However if the original scene's dynamic range is greater than the system dynamic range, intensity ratios can not be preserved and dynamic range compression algorithms are needed.

2.3. Compression methods

There are two different types of computational methods used to reduce the dynamic range of an image. A simple computational method is to apply a tone reproduction curve (TRC) to the image data; that is, each pixel is transformed from its current intensity to a new intensity within the display range of the output device. This type of transformation adjusts the intensity of each pixel using a function that is independent of the local spatial context. The TRC method is efficient because the operation is applied to pixels independently and thus can be performed in parallel using a look-up table.

Given the significance of edge intensity ratios, the spatial context of an image pixel may contain important information that should influence its displayed value. Various context-sensitive algorithms have been proposed, as well. Such algorithms

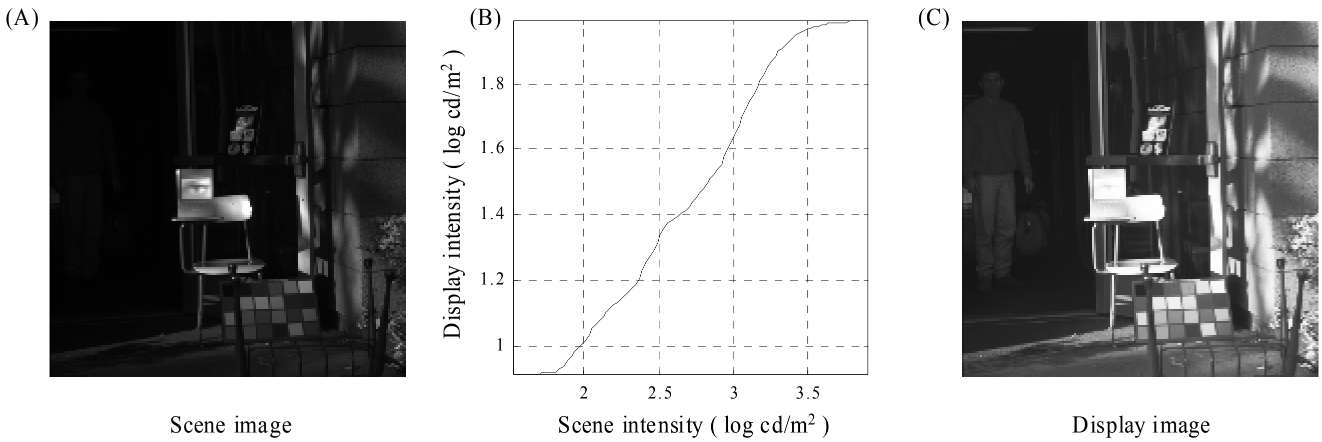

Figure 3. A tone reproduction curve is used to reduce dynamic range. (A) The original image. (B) The tone reproduction curve. TRCs are one-to-one and monotonic mappings. (C) The resulting image after the dynamic range has been compressed.

cannot be summarized by a single function that maps input intensity to output intensity. We call these more general context-sensitive methods tone reproduction operators (TROs). In the following, we will discuss and compare TRCs and TROs.

3. TONE REPRODUCTION CURVES

Tone reproduction curves compress the dynamic range by defining a function that maps the original input intensities into a narrower range of display intensities. If the image input intensities are I, and the display intensities are D, then the tone reproduction curve is a function, D = T(I), where T() is one-to-one and monotonic. An example of a tone reproduction curve is illustrated in Figure 3.

To preserve all image intensity ratios, the TRC must be a linear scaling, T(I) = s*I, analogous to looking at the scene through a neutral density filter. This linear scaling, however, can not be implemented when the display device has a smaller dynamic range than the original image. Suppose that the original image spans an intensity range of 1 to 1000 cd/m^2, and the display device spans an intensity range of 1 to 100 cd/m^2. If we chose a scalar such that the maximum image intensity matches the maximum display intensity, the rendered image would span the intensity range of 0.1 to 100 cd/m^2. Since the display device can not produce intensities below 1 cd/m^2, values between 0.1 and 1 cannot be displayed accurately.

This procedure is undesirable when the intensity range from 0.1 to 1 cd/m^2 contains visually significant spatial information. Alternative TRCs that spread the distortion smoothly across the entire intensity range are preferred. The next level of complexity then is to choose a traditional TRC that is nearly image independent. Two commonly used TRCs are power functions ($T(I) = s*I^p$, where $p<1$) or logarithmic mappings (T(I) = s*log(I)), where s scales the peak image intensity to the peak display intensity. These functions smoothly compress the dynamic range but do not preserve intensity ratios. Intensity ratios in bright regions are compressed, while intensity ratios in dark regions are preserved. While these compression schemes can be helpful for some images or for a general view of the data, often they are not very satisfactory.

Traditional TRC curves are nearly image independent: they are applied to an image by choosing only a single parameter (*e.g.* the exponent). More sophisticated algorithms have been designed to select the TRC using information about the image contents. For example, Larson[8] has described an algorithm that calculates image-dependent TRCs. These TRCs compress the intensity ratios of underrepresented pixel intensities and preserve the ratios of highly represented pixel intensities. The algorithm has two significant features. First, the algorithm takes into account some spatial properties of the image, such as the visual angle of the image. Second, the algorithm compresses image data more or less depending on how close a fit it is to the dynamic range of the display device. Since the algorithm takes into account the properties of the image display, the algorithm has the very useful feature of being idempotent (T(I) = T(T(I))). A compressed output image will not be further compressed if processed through the algorithm a second time. Surprisingly, not all proposed algorithms have this useful feature.

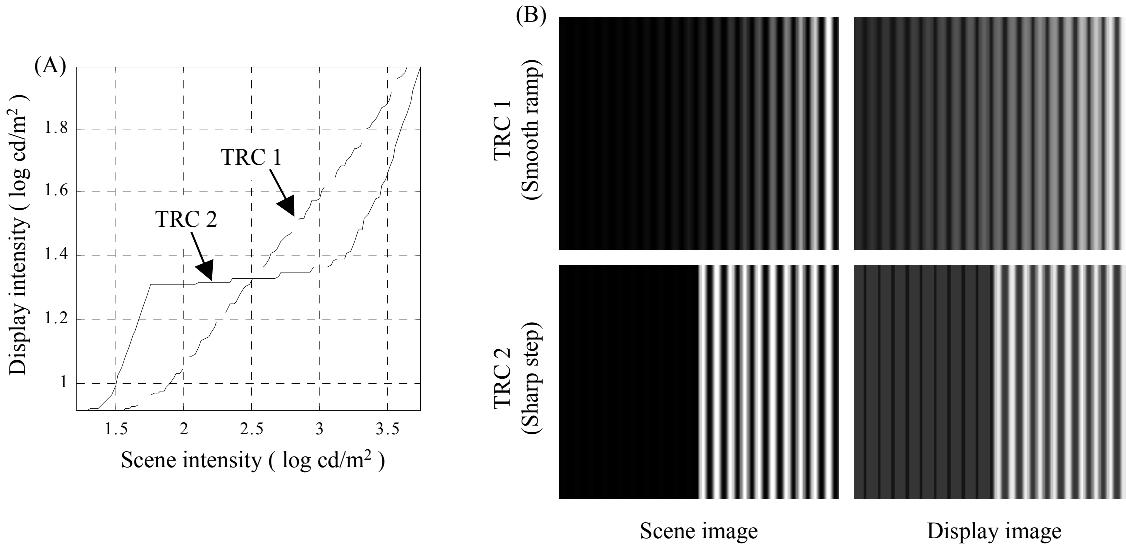

Figure 4. Larson's algorithm is applied to two scene images shown on the right. (A) The TRCs computed for the two images. (B) The scene and display images corresponding to the TRCs. See text for details.

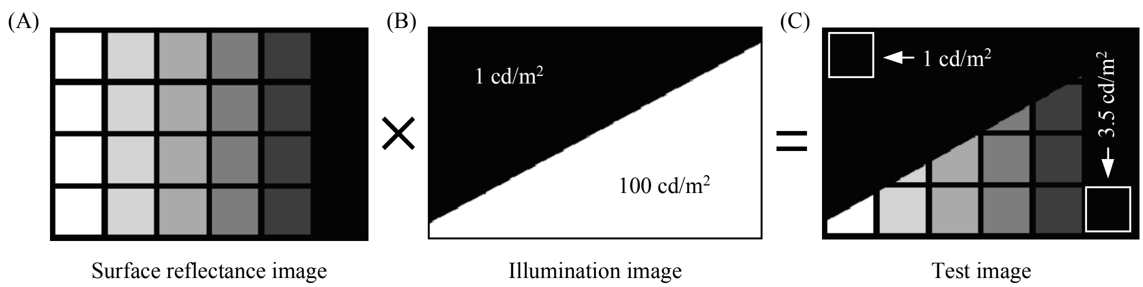

Figure 5. A test image with two spatially distinct illumination regions. (A) The surface reflectance image. (B) The illumination image. (C) The test image. The luminance of the white surface in the upper left corner is 1 cd/m^2. The luminance of the black surface in the lower right corner is 3.5 cd/m^2.

Figure 4 shows the TRC computed using Larson's algorithm on two different test images. The TRCs differ quite significantly depending on the image content. In the first image, pixel intensities are uniformly distributed across the intensity range. The algorithm creates a TRC that is roughly a power function (linear in the log-log plot). In the second image, pixel intensities fall within distinct intensity bands. The algorithm produces a TRC that is nearly linear within each range (slope near one in the log-log plot) and that is nearly constant over the region between the two intensity ranges. It has been our experience that Larson's algorithm performs efficiently and reasonably well.

To motivate why one might wish to have a more sophisticated dynamic range reduction algorithm, consider the image in Figure 5. This synthetic image comprises four rows of squares, each row containing the same gray series. The upper left and lower right image regions are illuminated at two different levels. Even though the illumination boundary is sharp, the image intensity values span the full range roughly uniformly. Hence, Larson's method produces a TRC that is close to a power function. The image consists of two discrete parts, but using only the pixel intensity distribution, the algorithm cannot discover that there is a spatial separation between the two illumination regions. This is one motivation for considering rendering algorithms that account for the spatial structure of the image.

A second motivation for considering more complex transformations comes from considering the relative intensity of the white surface in the upper left corner and the black surface in the lower right corner. Under these two illuminations, the luminance of the black surface exceeds the luminance of the white surface. When we render these images on a display with a limited dynamic range (equivalent to that of typical surface reflectance variations), it is desirable to render the white surface lighter than the black one. Permitting the range of the two illumination regions to overlap increases the available intensity

Figure 6. Tone reproduction operators are context-sensitive. Depending on its position in the image, a given intensity level may map into a different output level. (A) The distribution of input and output levels. (B) The output image. (C) The image to display intensity mapping of the TRO is illustrated for three different spatial regions of the image.

range within each region. Because TRCs need to be monotonic, or else we risk introducing reversals in local edge contrast, they can not perform this operation. Context-sensitive algorithms (TROs) are necessary.

4. TONE REPRODUCTION OPERATORS

We refer to algorithms that adjust pixel intensity using spatial context as tone reproduction operators. These operators are permitted to transform the same pixel intensity to different display values, or different pixel intensities to the same display value. Figure 6 illustrates a tone reproduction operator applied to the scene image shown in Figure 3A. The resulting mapping (Panel A) is clearly not one-to-one. Within local regions, however, TROs usually behave like a TRC. Panel C on the right shows the mapping of scene and display intensities for different image regions. Each region is mapped differently. This particular algorithm chooses a different power function within each local region.

The guiding principle of TRO design is to preserve local intensity ratios, which may correspond to surface reflectance changes, and reduce large global differences, which may be associated with illumination differences. By following this principle, TROs are designed to match the light adaptation of the visual pathways, which discounts illuminant variation but recognizes surface reflectance variations.

Multiresolution representations are a natural computational vehicle for preserving local intensity ratios, and there have been several published multiresolution algorithms for dynamic range compression.[9-11] While these authors use somewhat different methods, their multiresolution implementations do share several common features (Figure 7). First, by applying multiple lowpass filters, usually Gaussian filters, the scene image is decomposed into a set of images that represent the mean of the image at different spatial resolutions. Next, each mean image in the set is divided pointwise by its lower resolution image, producing a set of images that measure the local image contrast. The very lowest spatial resolution image, the lowpass image, is also kept for reconstruction purposes. Finally, applying a compressive function to each of the contrast images and the lowpass image reduces the image dynamic range. Power functions, like those used in TRCs, are typically used as compression operators. The display image is reconstructed by multiplying the compressed images together.

Figure 8C illustrates a significant difficulty in using multiresolution methods for dynamic range reduction. The figure shows the compression of a single scan line from the sinusoid-step test image (Figure 4). Panel A shows the Gaussian decomposition calculated for the test image at two spatial resolutions. Panel B shows the resulting contrast images calculated from the Gaussian decomposition and the original image. Because an edge includes many spatial frequencies, its representation is spread across both resolution bands. When the contrast images are compressed independently, the reconstructed edge becomes distorted. Panel C shows the resulting distortion or "halo" artifact.[9,11] This artifact is fundamental to multiresolution methods because each resolution band must be compressed differently. Indeed, if the compression does not differ across resolution bands there is no point in performing the decomposition.

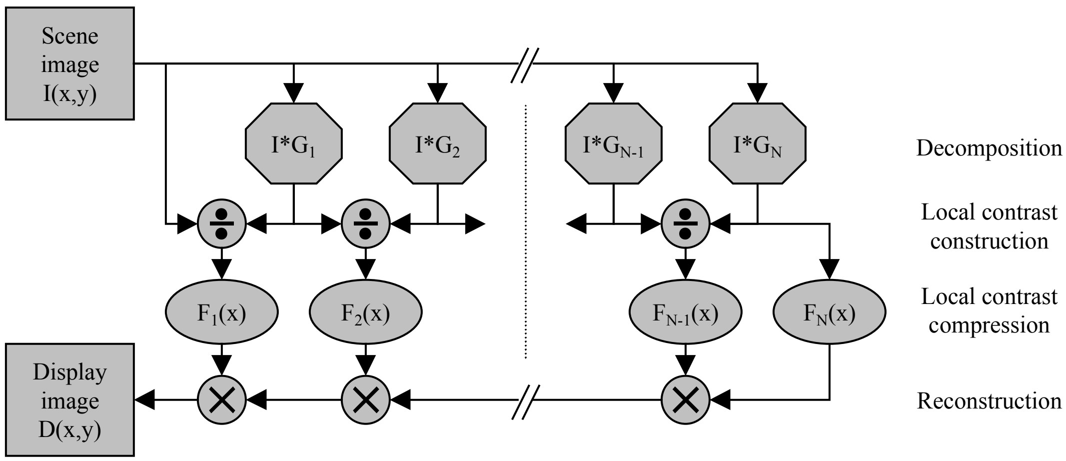

Figure 7. The basic multiresolution decomposition used by TRO algorithms. I is the scene image. G_1, G_2, G_{N-1}, G_N are Gaussian filters of increasing spatial size. The "*" denotes the convolution operator. $F_1(x)$, $F_2(x)$, $F_{N-1}(x)$, $F_N(x)$ are the compression functions.

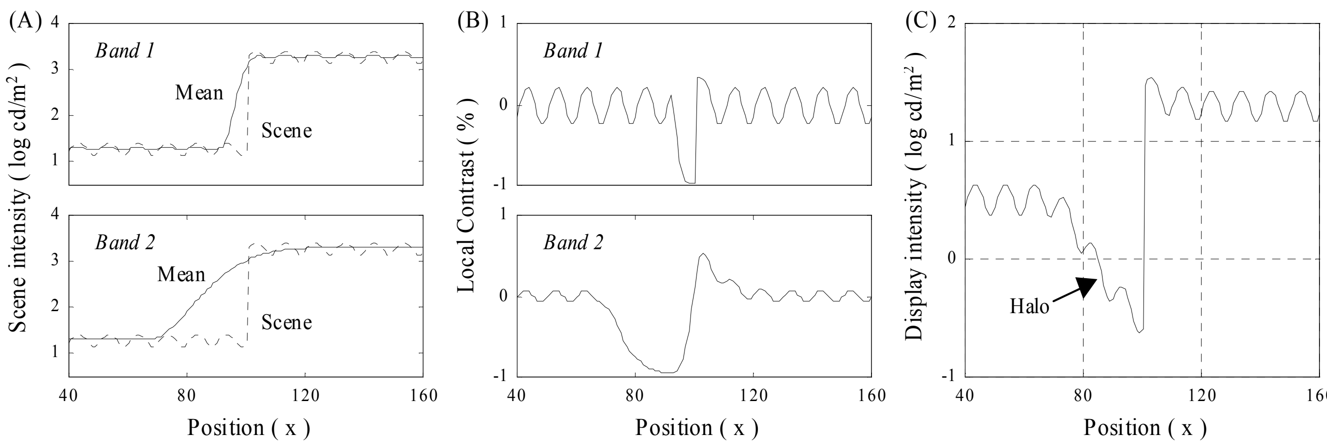

Figure 8. A multiresolution analysis of the sinusoid-step image. (A) Image mean at two different spatial resolutions using a Gaussian filter (Gaussian decomposition). (B) Local contrast at two different spatial resolutions (Contrast decomposition). (C) The reconstructed display image. The reconstructed image contains a halo because the compression operation scales each contrast image differently.

Two methods have been used for reducing the halo artifact. One method is to strive for a simpler decomposition that contains all large edges within one band. For example, it would be desirable to decompose the image into a representation of the illumination and a second representation of the surface reflectances. Then, one could compress only the illumination image (which will contain the high dynamic range variation) and recombine with the original surface reflectance image. This is the objective of algorithms such as Retinex.[12] A second method of reducing halos is to link the compression functions in a smooth continuous way across the different bands. While this method can not eliminate halo artifacts completely, it can reduce them.

Tumblin combines portions of the two methods in his LCIS algorithm.[11] The LCIS (low-curvature image simplifier) algorithm uses a form of anisotropic diffusion in an attempt to group edges of similar spatial properties into one resolution image during the decomposition process. To the extent that an edge is completely contained within one resolution image, it can be compressed independently of the other images. Since this cannot fully be achieved in practice, smooth compression across the different resolution bands is still necessary. Tumblin achieves smooth compression by adjusting the compression coefficients by hand for each image. Overall, Tumblin's algorithm reduces the effects of halos, but it is computationally expensive.

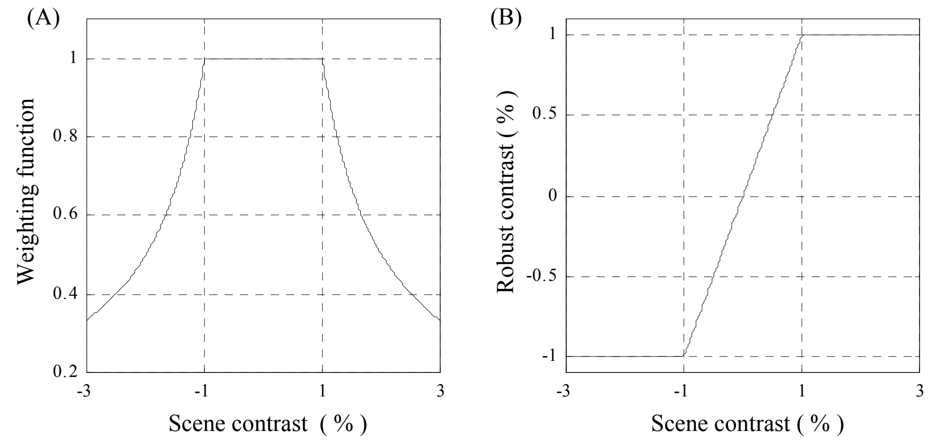

Figure 9. Huber's minimax robust estimator applied to image contrast. (A) The robust weights applied based on the contrast deviation from the mean. (B) The resulting effect on the local contrast.

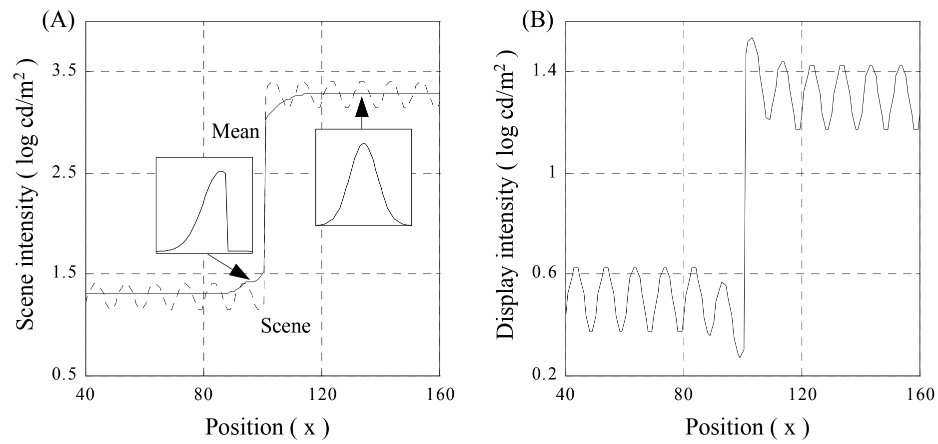

Figure 10. A robust Gaussian filter preserves edges better than a Gaussian filter. (A) A robust Gaussian filter is applied to one of the test images. The panel insets show the effective filter at two different points across the large shadow edge. The filter changes shape and preserves the high contrast edge. (B) The reconstruction of the test image. Robust filtering has smaller halo artifacts than conventional filtering. The precise size of the artifacts depends on a combination of the image and the filter parameters.

Given the computational expense of anisotropic diffusion, it is worth exploring the related but simpler methods of robust filtering. A robust filter is similar to a conventional filter, but the influence of outliers is reduced. In normal filtering, say Gaussian, the local mean is calculated as the weighted sum of nearby intensities. The weight depends only on the spatial position of the nearby pixel. A robust filter, say a robust Gaussian, includes a second weight that depends on the intensity difference between the current pixel and its spatial neighbor. A nearby pixel whose intensity is very different from the current pixel intensity is considered an outlier. Its contribution to the local mean is reduced by reducing its weight. If the pixel intensity is similar to the current pixel's intensity, its weight is not adjusted. A typical weighting function, expressed in the contrast domain, is shown in Figure 9, though many variants are possible. Robust operators are closely connected to the various types of anisotropic diffusion calculations.[13]

Like anisotropic diffusion, robust operators tend to preserve the sharpness at large transitions. Figure 10 illustrates an example of a robust filter applied to the sinusoidal-step test image (Figure 4). Panel A shows the robust mean of the test image calculated at a coarse spatial resolution. The insets in the panel illustrate how the convolution kernel changes as the operator passes near the sharp shadow edge. The change in the kernel reduces blurring across the edge. Panel B shows the resulting reconstruction after dynamic range compression. For the choice of filter parameters in this figure, the use of the robust operator, like the anisotropic diffusion operator used by Tumblin, reduces but does not eliminate the halos. The choice of filter parameters for dynamic range compression is an interesting research question that might best be decided by examining the statistics of natural scenes.

Finally, we observe that the TRO decomposition methods proposed to date do not explicitly include information about the display dynamic range. Algorithms that do not explicitly include this information are not guaranteed to be idempotent. Reprocessing an output image may change the result even though the dynamic range of the image was compressed on the first pass. TROs can (and should) be constructed to be idempotent. Recall that dynamic range compression is achieved by applying different power functions to the original image. If the dynamic range of the image is within that of the display, the exponent of each power function should be set to one so that the algorithm does not alter the original image. The exponent can vary gracefully from one to smaller values as the image dynamic range begins to exceed that of the display.

5. DISCUSSION AND CONCLUSIONS

For many years there has been conceptual agreement on a principal for achieving dynamic range reduction. First, decompose the image into two parts: one that captures the ambient lighting variation, and a second that captures the surface reflectance variation. Then, compress the range of the lighting image and reconstruct. For example, Horn describes this principle in his early and influential analysis of the Retinex theory, and various authors have applied variants of this idea in the domain of dynamic range reduction.[9,12,14] Horn's approach, along with the TRO algorithms that use linear filters, assume local contrast variations (high spatial frequencies) correspond to surface reflectance variations, while large global variations (low spatial frequencies) correspond to ambient lighting variations. This assumption fails for sharp shadows, which are fairly common, and produces artifacts in the compressed images.

Anisotropic diffusion operators such as LCIS and decomposition methods based on robust calculations are extensions of the linear TRO algorithms that try to overcome these difficulties. The operators relax the strong assumption that the spatial variation of the illuminant is entirely within low spatial frequencies. By providing a wider range of possible spatial distributions of the illuminant, they offer the hope of finding an automated procedure for the alternative decomposition. When coupled with methods that make the operator idempotent, these calculations may be useful for fully automated dynamic range reduction applications.

In reviewing this material and evaluating algorithm performance, we have noted that an important limitation is that we have only a modest understanding of the illumination and surface reflectance properties in natural scenes. What is the real dynamic range of a scene over the range that a human can discriminate? How does the dynamic range in a scene vary spatially? What are the properties of typical shadow edges? What about surface edges? Obtaining a more precise understanding of scene properties should be a priority for developing good test samples as well as evaluating practical algorithms. This information should serve as an important guide in developing the new generation of algorithms based on more robust methods.

ACKNOWLEDGMENTS

This work is supported by the Programmable Digital Camera project at Stanford, sponsored by Agilent, Canon, Hewlett-Packard, Interval Research, and Kodak. Jeffrey DiCarlo thanks Kodak for their support through the Kodak Fellowship program. We thank Abbas El Gamal, Peter Catrysse, Feng Xiao, Ting Chen, SukHwan Lim, Xinqiao Liu and Julie Sabataitis for their help and comments on the paper. We also thank David Brainard for the high dynamic range image shown in Figure 3, and John DiCarlo for the image shown in Figure 2.

REFERENCES

1. G. W. Larson and R. Shakespeare, *Rendering with Radiance: The Art and Science of Lighting Visualization,* Morgan Kaufmann Publishers, San Francisco, 1998.
2. P. E. Debevec and J. Malik, "Recovering high dynamic range radiance maps from photographs," *24th Annual Conference on Computer Graphics & Interactive Techniques,* 369-378, Los Angeles, 1997.
3. D. Yang, B. Fowler, and A. El Gamal, "A nyquist rate pixel level ADC for CMOS image sensors," IEEE Journal of Solid State Circuits*,* **34***,* 348-556, 1999.

4. D. Yang, A. El Gamal, B. Fowler, and H. Tian, "A 640x512 CMOS image sensor with ultra wide dynamic range floating-point pixel-level ADC," *IEEE International Solid-State Circuits Conference,* San Francisco, 1999.
5. R. W. G. Hunt, *The Reproduction of Colour,* Fountain Press, London, 1995.
6. B. A. Wandell, *Foundations of Vision,* Sinauer Associates, Sunderland, 1995.
7. B. E. Rogowitz, A. D. Kalvin, A. Pelah, and A. Cohen, "Which trajectories through which perceptually uniform color spaces produce appropriate colors scales for interval data?," *Seventh Color Imaging Conference: Color Science, Systems, and Applications,* 321-326, Scottsdale, 1999.
8. G. W. Larson, H. Rushmeier, and C. Piatko, "A visibility matching tone reproduction operator for high dynamic range scenes," IEEE Transactions on Visualization and Computer Graphics, **3**, 291-306, 1997.
9. Z. Rahman, D. J. Jobson, and G. A. Woodell, "A multiscale retinex for color rendition and dynamic range compression," *SPIE International Symposium on Optical Science, Engineering, and Instrumentation,* 1996.
10. S. N. Pattanaik, J. A. Ferwerda, M. D. Fairchild, and D. P. Greenberg, "A multiscale model of adaptation and spatial vision for realistic image display," *25th Annual Conference on Computer Graphics,* 287-298, Orlando, 1998.
11. J. Tumblin and G. Turk, "LCIS: A boundary hierarchy for detail-preserving contrast reduction," *SIGGRAPH '99 Annual Conference on Computer Graphics,* 83-90, Los Angeles, 1999.
12. B. K. P. Horn, "Determining lightness from an image," Computer Graphics and Image Processing, **3**, 277-299, 1974.
13. M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger, "Robust anisotropic diffusion," IEEE Transactions on Image Processing, **7**, 421-432, 1998.
14. J. McCann, "Lessons learned from mondrians applied to real images and color gamuts," *Seventh Color Imaging Conference: Color Science, Systems and Applications,* 1-8, Scottsdale, 1999.