Spatially Nonuniform Scaling Functions for High Contrast Images

Proceedings of Graphics Interface 93, pp. 245-244, May 1993

K Chiu, M Herf, P Shirley, S Swamy, C Wang, K Zimmerman
Computer Science Department,
Lindley Hall, Indiana University, Bloomington, IN 47405
812 - 855 - 6486
e-mail address : [chiuk | mherf | shirley | shankar | wangc | kuzimmer]@cs.indiana.edu

An algorithm is presented that scales the pixel intensities of a computer generated greyscale image so that they are all displayable on a standard CRT. This scaling is spatially nonuniform over the image in that different pixels with the same intensity in the original image may have different intensities in the resulting image. The goal of this scaling transformation is to produce an image on the CRT that perceptually mimics the calculated image, while staying within the physical limitations of the CRT.

CR Categories and Subject Descriptors: I.3.0 [Computer Graphics]: General; I.3.6 [Computer Graphics]: Methodology and Techniques.

Additional Key Words and Phrases: brightness, contrast, dynamic range, filter, glare, luminance, radiance, vision.

1 Introduction

In recent years much attention has been paid to the chromatic limitations of CRTs and printers, and their resulting shortcomings in accurately reproducing highly saturated colors[5, 14, 17, 4]. This is often referred to as a gamut mapping problem because the gamut (set of distinct colors) in the image we wish to display is not a subset of the CRT gamut. Spatial limitations of CRTs have also been explored, usually in the language of signal processing (e.g. [6, 11, 3]).

Perhaps the greatest limitation of the CRT, its small dynamic range, has received relatively little attention. This limitation particularly manifests itself when we wish to display images with contain both high and low radiance values. In this paper we examine the difficulties in displaying a particular test image of this type. To simplify our task, we have chosen a greyscale image as our test image. We view this as a feasibility study into whether such "difficult" images can be adequately displayed on a CRT. We conclude that our test image can be displayed adequately, and the display method we use gives some insight into what type of general algorithms should be developed for the display of arbitrary images.

The work presented in this paper is meant to be exploratory, and is not meant to be an alternative to the approach of Tumblin and Rushmeier[18]. This paper presents spatially varying image mappings without considering adaptation issues. Tumblin and Rushmeier take an orthogonal approach and concentrate on adaptation issues (making dark scenes look dark and bright scenes look bright). Both Tumblin and Rushmeier and this paper ignore issues involving color. Ultimately, the issues of overall dark adaptation, spatial dark adaptation, and chromatic adaptation must be addressed in a single context.

In Section 2 we describe our test image, a lamp with an exposed bulb lighting a room, and why the properties of this image imply that a spatially nonuniform transform is needed to display the image on a CRT. In Section 3 we formally state the problem of transforming a computer generated greyscale image to a displayable image, and discuss our basic strategy for finding a "good" transformation. In Sections 4 and 5, we discuss both spatially uniform and spatially nonuniform techniques for scaling a greyscale image to an image to be displayed. Our particular ad hoc image scaling algorithm, along with a method for introducing glare effects into the image, is presented in Section 6. Finally, we discuss our current thinking and speculate on future solutions in Section 7.

2 Test Image

Our test image is of a room lit by a single incandescent bulb. This image was generated using distribution ray tracing[2] for the direct lighting, and a physically plausible ambient component to approximate the indirect lighting. We compared the luminances in this image to the luminances in a real room[1], and found the range and location of luminance values to be fairly consistent.

The test image is shown in Figure 1, where the radiance varies from 0.0167 (arbitrary units) in the table leg, to 500 at the light-bulb, for a ratio of approximately 1:30000. This image is displayed by mapping radiance 0 to black, and mapping 1.0, 10.0, and 500.0 to white (top to bottom). Because of the wide range of radiances in the image, none of mappings is satisfactory. The top image (white = 1.0) displays the floor in a way that is qualitatively similar to the way the floor looks in the real scene (the woodgrain is clearly visible), but pixels near the light bulb are washed out. The middle image (white = 10) shows both the light and the floor, but the detail in the floor is

[1] Luminance in the real room was measured using a spot photometer. Using this device helped our qualitative understanding of the relation between luminance and brightness. We recommend that anyone working in this area attempt to acquire such a device, which are available in good photography supply stores for under $500.

lost, and the pixels on the edge of the bulb all are mapped to white, so the bulb edge appears jaggy. The bottom image (white = 500) correctly antialiases the light bulb edges, but the rest of the image pixels go to black or near black. None of these three mappings will do.

The test image characteristics make us believe that the same mapping cannot be used at each pixel. The first characteristic is that a pixel that is completely filled by the light bulb will have radiance 500, and pixels that is half filled by the light bulb will have radiance 250. The pixel with value 500 will be mapped to white (1.0). To correctly antialias the bulb, and thus prevent jaggies, the pixel with value 250 must be mapped to a medium grey (about 0.5). A pixel partially filled by the bulb edge with value 5 should be mapped to a dark grey (around 0.01). This mapping for the actual pixels near the bulb is shown in Figure 2. On the other hand, the wood grain in the floor, which is definitely visible in the real scene, varies in radiance from approximately 0.02 to 0.08. For this difference to be visible as light and dark features, the 0.08 radiance must be mapped to some non-black radiance, which means a pixel value greater than 0.01. This means a pixel covering the floor with original value 0.08 should be mapped to a higher value than a pixel near the bulb edge with original value 5.0. This basic observation implies that the way a radiance is mapped to a pixel value on the CRT should depend on spatial position in the image. This is the basic observation that lead us to begin this work.

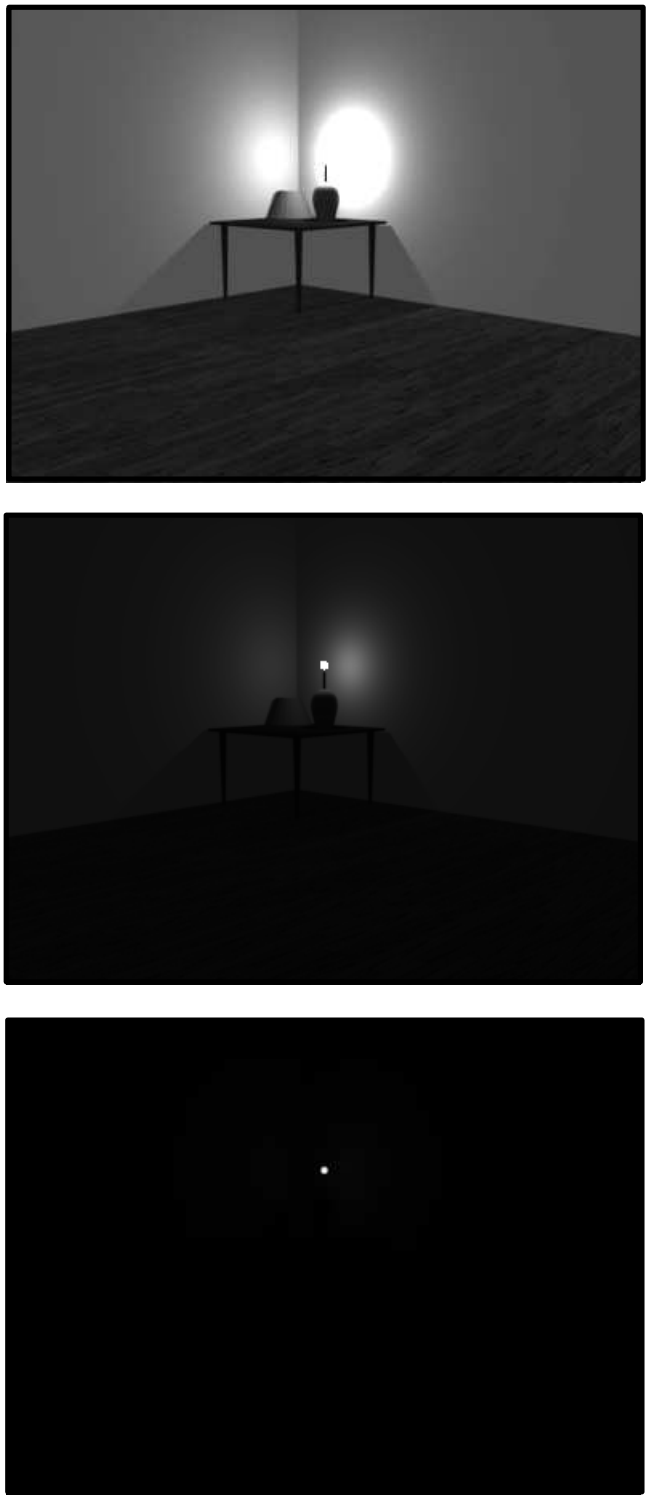

Figure 1: (top) black = 0.0, white = 1.0+. (middle) black = 0.0, white = 10.0+. (bottom) black = 0.0, white = 500.0.

3 Mathematical Formulation

In this section we attempt to formally define what a display algorithm does, and outline our basic strategy for finding a good display algorithm. We do this by viewing the display process as mapping a particular image to some approximate image. A good display algorithm is one that finds an image that is approximate in some perceptual sense.

A "real-world" greyscale image r is a member of the set of all such *real* images R. If we parameterize the radiances that define r over some image plane, we can represent any real image as a mapping from $[0,1]^2$ to a radiance. If we view radiance as a positive scalar, we can write down the set of all real images as:

$$R = \{r | r : [0,1]^2 \rightarrow [0,\infty)\}$$

For display purposes, an image in R is mapped to a discrete-domained *raster* image[2] defined on a n_x by n_y pixel grid of pixels $P = \{0,\cdots,n_x-1\} \times \{0,\cdots,n_y-1\}$. The set F of all such images is defined as:

$$F = \{f | f : P \rightarrow [0,\infty)\}$$

Our test image is a member of F with $(n_x, n_y) = (400, 300)$.

[2]The raster image is normally computed from the real image through a process known as *filtering*. In computer graphics, the raster image is typically calculated directly because the filtering process is tightly coupled to the sampling process. This is not surprising because of the vastly reduced information content in a raster image relative to a real image.

1.2	1.3	1.4	1.6	1.7	1.8	2.0
1.3	11	83	154	83	1.8	2.0
1.3	73	439	490	419	83	2.0
1.3	225	490	500	500	235	2.0
1.3	184	470	500	470	195	2.0
1.3	11	235	337	246	12	2.0
1.3	1.3	1.5	12	1.7	1.8	2.0

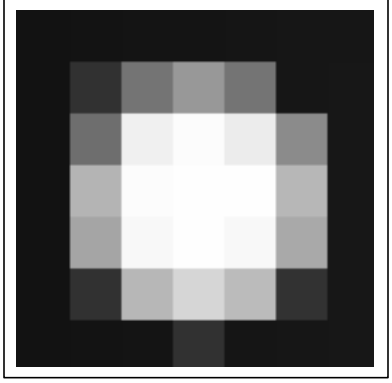

Figure 2: Pixel luminance values near the bulb (above) and the resulting image if 500 is defined to be white

The finite dynamic-range of the display hardware requires a further normalization step. We define the set of all *normalized raster* images as:

$$\hat{F} = \{\hat{f} | \hat{f} : P \rightarrow [0,1]\}$$

Finally, the digital nature of the display devices requires that an image be a discrete-valued function. A *discrete raster* image is then a member of the set:

$$D = \{d | d : P \rightarrow \{l_0, l_1, l_2, \cdots, l_n\}\}$$

Though this last step is reasonably straightforward, to get a "correct-looking" discrete raster image careful attention must still be paid to gamma correction, and dithering (even with twenty-four-bit color images) is recommended[5]. Of course, the final image on the screen is a member of some subset of $\hat{F}$ (having the same cardinality as D), because at each pixel a continuous phosphor kernel will be generated, and the overlapping kernels of all pixels form an image continuous in space. The spatial limitations of CRTs is examined in detail in Glassner's book[3].

Of these steps, the transformation from raster image $f \in F$, to normalized raster image, $\hat{f} \in \hat{F}$ is the one we will concentrate on. Therefore, we will assume that we are given a raster image f and only need to find a normalized raster image $\hat{f}$. If we had a perceptual distance metric, $\hat{f}$ would simply be the image "closest" to f. This error minimization strategy has been applied by Glassner et al. for gamut-mapping[4] and for brightness mapping by Tumblin and Rushmeier[18].

In determining $\hat{f}$, the characteristics of the human visual system should be considered. We can extensively change image characteristics that the visual system is particularly *in*sensitive to, such as overall luminance (e.g. an image that is scaled everywhere by a factor of two looks the same as the unscaled image). However, we must be very careful to preserve the qualities of the image that the visual system is particularly sensitive to, such as local contrast.

Physical limitations of the eye create a number of artifacts in the real-world image. Two examples might be bloom (glare) and dark adaptation effects. Other examples can be seen in various optical illusions. The causes range from scattering in the lens to chemical equilibria. Artifacts induced in the visual system by the real image may not be induced by the displayed image, however, so introduction of these artifacts directly into the displayed image can increase realism. Care must be used, though, because many of these artifacts are normally compensated for by brain functions. Such artifacts can then look "wrong" if present in the displayed image. For example, simultaneous contrast effects will occur in a displayed image as long as the contrast value is achievable on the CRT, and hence these effects should not be exaggerated in the displayed image. On the other hand, the glare we see in real scenes cannot be produced with the relatively small maximum luminance displayable on a CRT, so we must simulate the glare in the image itself, as is done by Nakamae et al.[12].

Understanding what characteristics of the image must be preserved, and what can be changed (and more importantly how much change is allowed) is difficult, and unfortunately the vision literature does not yet provide enough detailed, quantitative data to develop such a metric from purely non-empirical means. Furthermore, this situation is not likely to improve in the near future. Tumblin and Rushmeier summarize the issue nicely:

> The eye's response to light is still not completely understood. It is difficult to accurately quantify because vision blends smoothly with higher brain functions, and because the eye's behavior is strongly dependent on the content of the viewed image. Brightness response is usually described by several processes, including at least adaptation, simultaneous contrast, brightness, color-constancy, memory, and cognitive processes. Many of these are interdependent, self-adjusting, and difficult to measure separately; each tends to obscure the other, so that brightness rules inferred from simple tests often fail when applied to more complex images.

4 Spatially Uniform Mappings

The simplest mapping from F to $\hat{F}$ is to uniformly scale at each pixel and to then clamp intensities above one:

$$\hat{f}(i,j) = \begin{cases} \frac{f(i,j)}{I_{\text{clamp}}} & \text{if } f(i,j) < I_{\text{clamp}} \\ 1.0 & \text{otherwise} \end{cases}$$

For high dynamic range images, the image either looks too dark if I_{clamp} is set high, or clamping artifacts occur if I_{clamp} is set low (e.g. Figure 1). These artifacts can be reduced in severity if a more

continuous transform is used[5], but either the image will still appear too dark, or the antialiasing near luminaires will be compromised, as will be discussed in Section 5.

Tumblin and Rushmeier noted that the adaptation level of the observer should be considered in designing the uniform mapping. Using the model for adaptation given by Stevens and Stevens[16, 15], they arrived at a uniform transform to reproduce the *tone* of the image being rendered. In other words, an image of a room lit by a one watt bulb should be displayed on a CRT in a way that makes it look dark, and an image of a room lit with a 1000 watt bulb should appear bright. This hands-off method nonlinearly transforms and scales f to its corresponding adaptation level on a display and works very well with relatively low contrast scenes. To our knowledge, Tumblin and Rushmeier have produced the first algorithm that displays a computer generated image based on a brightness perception model. As they point out, the brightness model they base their algorithm on is too simple to completely succeed in a spatially complex image. However, their basic idea of using a computational brightness model to calculate a perceptual distance between two images is likely to be part of most future image display algorithms.

5 Spatially Nonuniform Mappings

The transformations discussed in the last section are of the form $\hat{f}(i,j) = g(f, f(i,j))$, where the function g is independent of the position of the pixel, and is solely based on the properties of f. So a particular pixel value will be set the same regardless of its position in the image. As argued in Section 2, such transforms can never adequately display our test image because either the light bulb will be washed out, or the woodgrain on the floor will become too dark to be seen.

Two considerations, however, lead us to believe that spatially nonuniform mappings can yield better results. First, the visual system is itself spatially non-uniform. Second, and more convincingly, artists and photographers have established the usefulness of spatially non-uniform mappings through a history of experience and observation. An artist working with her eyes and brain may produce a normalized raster image that is perceptually closer to the real image than any image that a spatially uniform filtering process can produce. Photographers also use spatially non-uniform techniques when they selectively expose different areas of the photographic paper.

In Figure 1, an artist would use a combination of all three pictures: she would dim the pixels near the light bulb, and increase the intensity of the pixels far from the bulb. Effectively, this is a spatially nonuniform mapping, because the mapping depends not only on pixel value, but on i and j as well.

In general we can describe spatially nonuniform mappings as $\hat{f}(i,j) = g(i,j,f,f(i,j))$. We have found it more convenient to represent the transformation as a product of the original image with a scaling function:

$$\hat{f}(i,j) = S(i,j)f(i,j)$$

In this scheme, a spatially uniform transform would have $S(i,j) =$

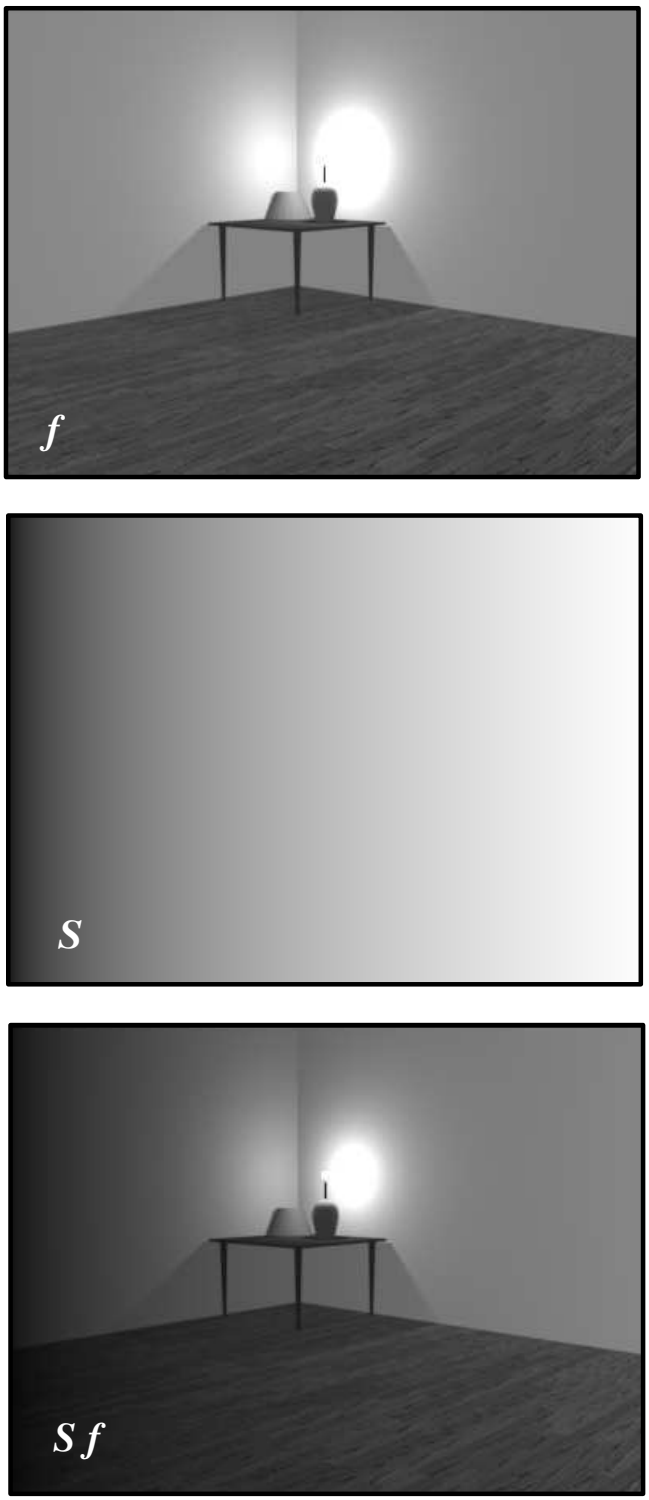

Figure 3: (top) Input image f, black = 0.0, white = 1.0+. (middle) Scaling function S, black = 0.0, white = 1.0. (bottom) Image where each pixel is the product of the corresponding pixel in S and f.

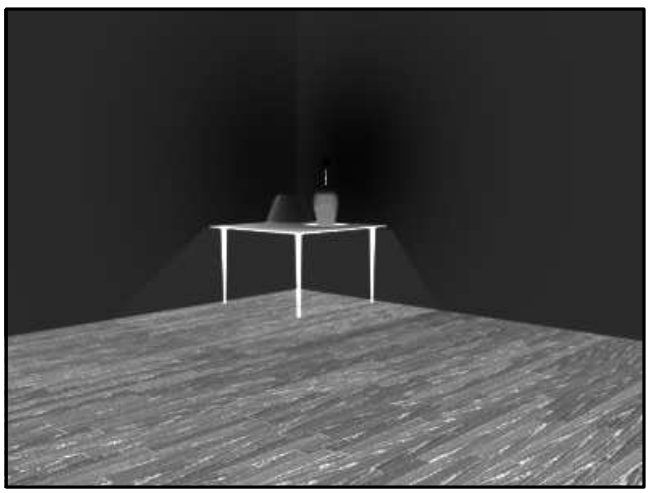

Figure 4: The inverse of f. Using this image as a scaling function will yield a white final image.

$g(i,j,f,f(i,j))/f(i,j)$. An example of a nonuniform transform is shown in Figure 3, where the scaling function S is simply a ramp from 0.0 to 1.0, left to right. This darkens the left side of f. This scaling function is not actually valid, because Sf is not a member of $\hat{f}$. There are still some pixels near the bulb with values greater than one. From observations like this, we can immediately establish some constraints on S. Because $0 \leq \hat{f}(i,j) \leq 1$,

$$0 \leq S(i,j) \leq \frac{1}{f(i,j)}$$

This immediately points out two valid scaling functions (i.e functions that map all pixels in f to a value between 0 and 1): $S_0(i,j) = 0.0$ and $S_{max}(i,j) = 1.0/f(i,j)$. A valid scaling function must have $S_0(i,j) \leq S(i,j) \leq S_{max}(i,j)$ for all (i,j). S_{max} is just the inverse of f and is shown in Figure 4 where zero maps to black, and 50.0 maps to white.

We should try to discover other simple properties S should have. In the same way stochastic sampling relies on the eye's insensitivity to noise[2], we should design S with human visual properties in mind. Land[8] and Marr[10] have argued that the eye is not sensitive to luminance so much as reflectance. As a result of this, slow spatial variation in luminance is to some extent ignored by the eye. Land demonstrated this by illuminating a Mondrian print with different incident intensity ramps across the image and demonstrating that the images look very similar even though luminance values for particular locations vary by as much as a factor of ten from image to image. This implies that we can display a pictures with wider range than the dynamic range of the display, and that, as long as S has a low magnitude gradient, we shouldn't see the scaled image as "looking" very different than the unscaled image. We can use these constraints to investigate scaling strategies.

6 An Ad Hoc Scaling Function

Tumblin and Rushmeier state that some vision researchers have speculated that the adaptation level in a part of the image is related to a low pass filter of the image[18]. It seems natural to make $S(i,j)$ proportional to the reciprocal of a filtered (blurred) f. This will accent dark areas and dim bright areas, much the same way darkroom workers preferentially expose dark regions of a print[3].

Two questions must be answered to apply that technique here. First, what filter should be used to blur f? Second, what should the constant of proportionality be? We attacked this question empirically and found that noticeable artifacts were introduced if the filter was not extremely wide (the kernel covering the whole image). We found that the largest variation that could be used in the kernel before the nonuniform scaling was objectionable was approximately a factor of four. For the proportionality constant, we started with $k = 2$ in the equation:

$$S(i,j) = \frac{1}{k f_{blur}(i,j)}$$

where f_{blur} is the filtered f. Of course, k is rather arbitrary as we've discussed. A k of 2 will bring colors near the average to $\hat{f}(i,j) = 0.5$. This doesn't leave a lot of room for the brighter pixels where the transformed image still has pixels out of range. This led us to think we wanted k as large as we could make it without losing detail in the dark regions. We found this to be at approximately $k = 8$.

The low pass filter used to create the images in this paper is $e^{-0.01r}$ before normalization, where r is the distance (in one pixel width equals one) from the center of the kernel. In a more general algorithm, some angular distance related to field-of-view would probably be more appropriate. We tried several other filters, such as the cone filter, the Gaussian filter, and the Perlin[13] filter. However, as long as the filter was wide, we saw no obvious differences in the results. To avoid darkening near the edge, we normalized the weights of the kernel pixels within the image boundaries so that they summed to one. In other words, pixels in the blurred image are a weighted average of pixels in the original image.

Unfortunately, a full blown filtering operation with full screen filters requires approximately $2N^2$ operations, where N is the number of pixels in the image. This is over one trillion (10^{12}) operations for a full-screen image and over ten billion operations for the 400 by 300 test image in this paper. Because the blurred image is quite smooth, we calculated a selected set of pixels (every tenth row and column) in the blurred image and then interpolated between them. This makes a more manageable number if operations in the blurring calculation. In order to avoid Mach bands, the interpolation function must maintain a continuous derivative in the blurred image.

[3] In fact, the technique of using a blurred f as a filter has been used before in photography and in image processing. In the so called unsharp masking technique, [9], the image filtered with a Gaussian filter, or some other blurring filter, is subtracted from the original image to get an image enhanced in sharpness. This technique may cause negative pixels, so the subtracted image is often scaled down, or the absolute value of the difference is used. This technique is really meant to enhance features in the image rather than to make the image perceptually correct.

Figure 5: The blurred image. Black = 0.0, white = 1.0.

For ease of calculation we used Perlin's interpolation function[13], which has this property in its derivative. In one dimension, Perlin's function interpolates between to values c_0 and c_1 as $c = (-2t^3 + 3t^2)c_0 + (2t^3 - 3t^2 + 1)c_1$, where t varies from zero to one between c_0 and c_1. The blurred image produced using this method is shown in Figure 5.

6.1 Corrected Transformation

Unfortunately, this transformation based on blurring still leaves some transformed pixels above 1 if the dynamic range is high. This can be seen because $S(i,j) > 1/f(i,j)$ for some pixels on or near the light. A poor but valid transformation can be obtained by clamping the intensities above $1/f(i,j)$:

$$\hat{S}(i,j) = \begin{cases} S(i,j) & \text{if } S(i,j) < \frac{1}{f(i,j)} \\ \frac{1}{f(i,j)} & \text{otherwise} \end{cases}$$

However the resulting image will have the same problems near the luminaire as described in Section 2, i.e. the antialaising near the bulb will not behave as shown in Figure 2. Another way that we can see this clamped S is inadequate is to observe that S will have a very steep gradient where the clamping occurs (the bulb edge). To avoid the artifacts because of this steep gradient, we can smooth them. We can be fairly sure that we want the luminaire to map to 1.0. So we can attempt to smooth $\hat{S}$ with the constraint that the pixels where S is out of range stay fixed. This smoothing can be accomplished with repeated filtering on $\hat{S}$ using a small width kernel, with the constraint that previously clamped pixels in S will not change. The actual filter used is a three by three filter shown in Figure 7. The weights in this filter sum to one, and their relative sizes are $1, 0.5,$ and $0.25\sqrt{2}$.

This process is illustrated in Figure 6 where in the top the original S is shown, in the middle the clamped $\hat{S}$ is shown, and in the bottom right the clamped and smoothed $\hat{S}$ is shown. The smoothing was accomplished by running the filter in Figure 7 over the middle image

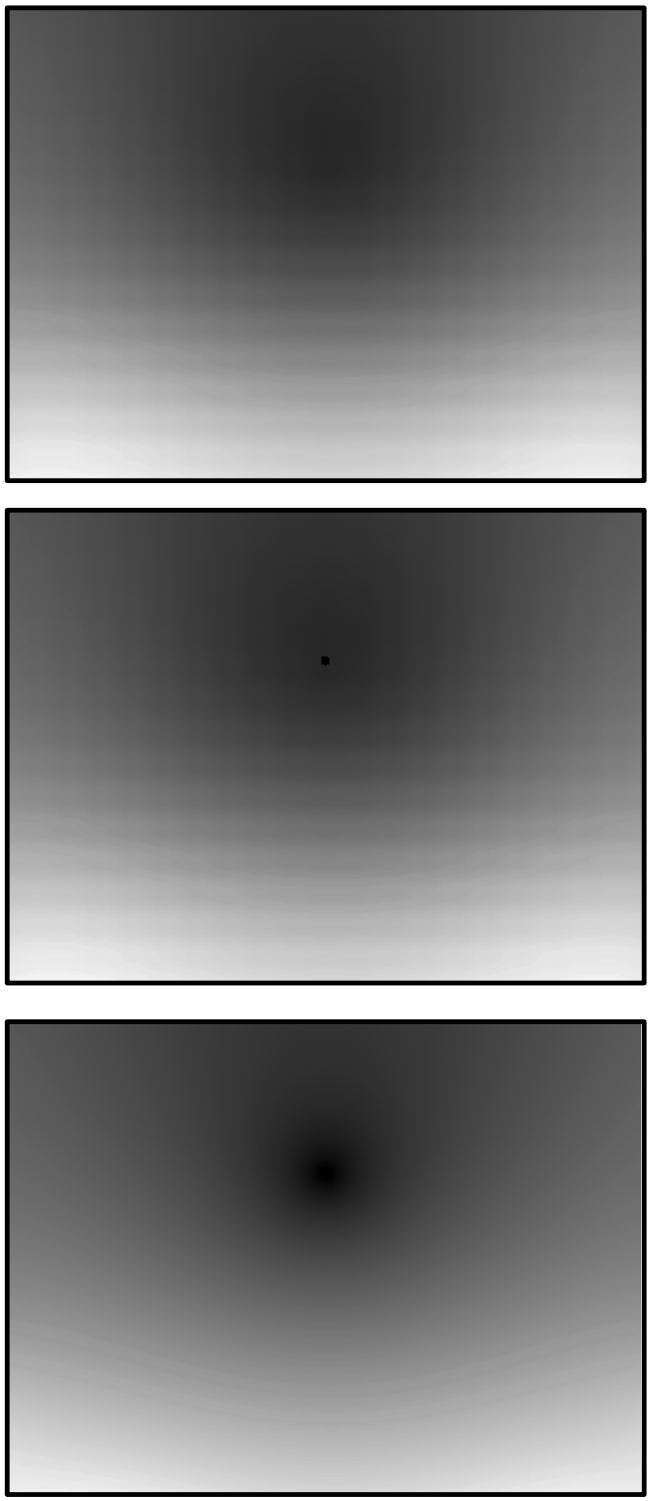

Figure 6: (top) $S = 1.0/(8 * f_{blur})$. (middle) clamped S. (bottom) smoothed and clamped S.

0.080	0.113	0.080
0.113	0.227	0.113
0.080	0.113	0.080

Figure 7: Filter used to to smooth $\hat{S}$.

ten thousand times. Figure 8 shows the application of the scaling function to out test image. The bottom of this figure shows the scaled image to be displayed. The dark ring around the bulb looks strange. One reason for this is that the blooming effect (glare) we see around real luminaires is not present.

6.2 Glare from High Intensity Objects

Better results can be achieved if glare is modeled. This reduces the maximum contrast in the image and creates an illusion of brightness. Nakamae et al.[12] modeled two types of glare. The first was blurring or "blooming", where a hazy fog is visible around the luminaire (see Figure 9). The second was "streaking", where star-like smears emanate from the luminaires. This streaking is also implemented in Ward's *Radiance* program[19]. We have only implemented the blooming effect.

In a real scene a glare or blooming effect can be observed near objects of high intensity. This is the result of light scattering within the eye. This effect has to be placed in the displayed image because the CRT cannot produce a luminance large enough to trigger noticeable glare in the eye. The amount of scattering which occurs is different for each viewer. Factors which determine the scatter include the age of the viewer, the viewer's distance from the viewed object and the intensity of the object [7].[4]

Since the blooming effect is highly viewer dependent, the development of our model for introducing glare into the image is empirical. Intuitively a pixel in a filtered image $\hat{f}(i,j)$ should retain some constant factor k of the original value of $f(i,j)$ where $k < 1.0$. The remaining $1-k$ should be based on a weighted average of surrounding pixels with adjacent pixels contributing more. A simple filter with a large peak is appropriate. A filter $\hat{\mathcal{F}}$ was created with

$$\hat{\mathcal{F}}(i,j) = \begin{cases} k & \text{if } i=j=0 \\ k'\mathcal{F}(i,j) & \text{if } \sqrt{i^2+j^2} \le w/2 \\ 0 & otherwise \end{cases}$$

where

[4]The increase in scattering that occurs with age is almost entirely due to changes in the lens.

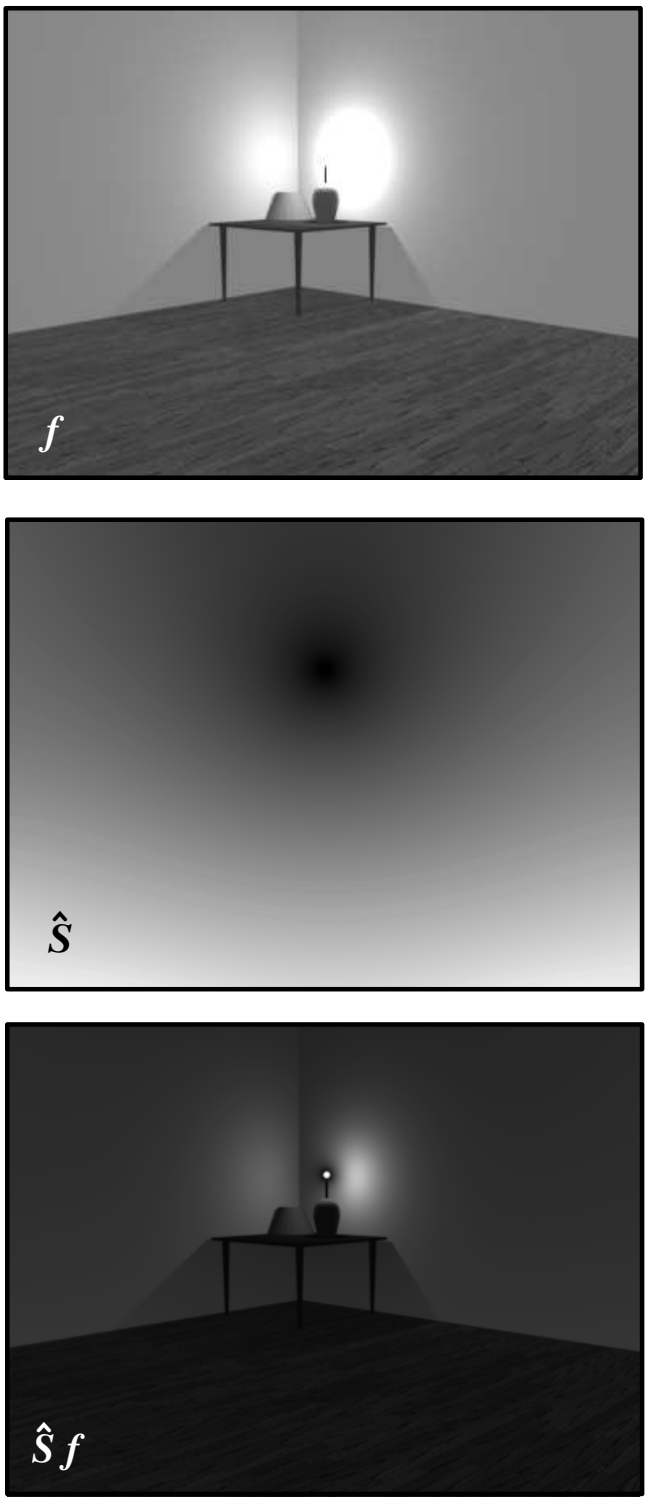

Figure 8: (top) Input image f, black = 0.0, white = 1.0+. (middle) Scaling function $\hat{S}$, black = 0.0, white = 1.0. (bottom) Image where each pixel is the product of the corresponding pixel in $\hat{S}$ and f.

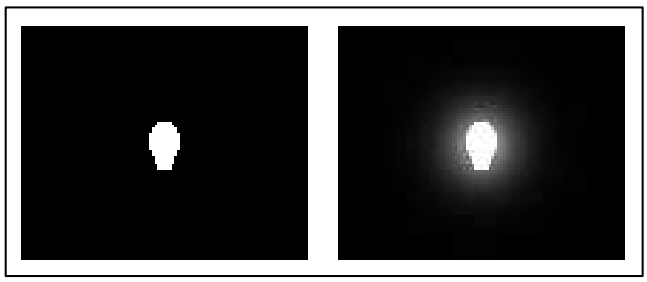

Figure 9: (left) A bare light bulb. (right) A light bulb with blooming effect.

w is the width of the filter and

$k' = (1 - k)/I$

$\mathcal{F}(i,j) = \left| \sqrt{i^2 + j^2} - \frac{w}{2} \right|^n$

$I = -\mathcal{F}(i,j) + \sum_{j=w/2}^{w/2} \sum_{i=w/2}^{w/2} \mathcal{F}(i,j)$ is the normalization constant

n is a controllable exponent greater than one.

Note that the (i,j) in $\mathcal{F}(i,j)$ refers to the coordinates relative to the center of the filter. The particular form for F was chosen based on three considerations: that it is a simple function whose shape is controllable by n, that the sum of the filter coefficients is one, and that it has a zero first derivative at its boundary, which is needed to avoid the introduction of Mach bands into the image (for example, see [1]).

The result of passing this filter with $w = 121$, $n = 8$ and $k = 0.8$ is shown in Figure 10. This is the same basic approach used by Nakamae et al. to generate their very impressive driving simulation animations, although they only run their filter over luminaires. While this saves time, it does not take into account the fact that scattering in the eye takes place over the whole image. It is just more perceivable near patches of high intensity. This can create errors in images containing high intensity non-luminaires (e.g. specular reflections of luminaires).

We do not model any kind of streaking from the luminaire, although this was done by Nakamae et al. and in Ward's *Radiance* program, both with good results. We chose not to do this because we do not always see streaks when viewing lights, and we still do not understand the mechanism behind these streaks when they do occur. Nakamae et al. suggest that the streaks are caused by diffraction through the lashes of the viewer. While this does cause some streaking effects, some streaks can be seen even when the eye is wide open. One possibility is that the streaks are caused by imperfections or scratches on the surface of the lens, but more information is needed to be certain. We are fairly sure that blooming is caused by imperfections in the lens, and that diffraction is not an important mechanism in blooming. According to the *IES Lighting Handbook*[7]:

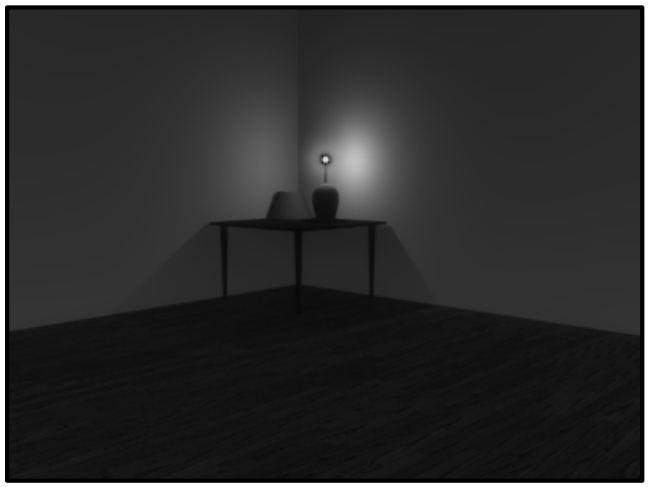

Figure 10: Image after glare and scaling transforms

> Diffraction—regardless of whether rays are in focus or not, there is always a certain amount of blur due to the diffraction of light. This determines the ultimate resolving power when the eye is in best focus, but the blur is not large enough to be perceived.

This implies any blooming we do model should probably not have any significant color divergence.

7 Conclusion

We have presented a method to map a particular high dynamic range image to a normalized image that is easy to display on a CRT. Our method is not meant to be optimal, but a demonstration that spatially-varying scaling is essential for the display of high dynamic range images, and that this type of transform will be of great utility to computer graphics practitioners. Our method assumed a medium intensity scene (not on the extremes of our ability to dark adapt) with a properly adapted person.

Our solution is based purely on experimental results. Currently we are working on finding a theoretical basis for these results. One way to interpret the final scaling function we used is that we have filtered the image using multiple scales of the form proposed by Witkins in his scaled space filtering work[20]. We think that this is a good area to investigate and we hope it will yield a more hands-off approach.

Future work should also account for dark adaptation, chromatic adaptation, time varying adaptation for walk-throughs, and optical illusions. Other empirical techniques to map f to $\hat{f}$ should also be investigated. It is our feeling that instead of filtering we might be better off applying the normalization constraint along with contrast preservation constraints to f and follow a relaxation procedure. This is a similar strategy to the device-directed rendering technique used by Glassner et al. to force images into a device color gamut. The brightness and color work could certainly be combined into a system that fits an image to a three dimensional gamut.

Perhaps the most interesting implication of this work is that the error metrics employed by rendering algorithms should pay less attention to low frequency errors (indirect lighting), because the eye is not sensitive to those frequencies. This may help explain why it is often not visually obvious that a radiosity algorithm is simulating color bleeding effects, but a good geometric model never goes unnoticed.

8 Acknowledgements

Thanks to Greg Ward for several helpful discussions and his willingness to share code and pictures, to Gary Meyer for initial suggestions about this work, to Andy Hanson for suggesting the use of the scaling function as a formalism, to Tim McBride for initial work on this project, to Greg Spencer for discussions of glare and extended precision file formats, to Andrew Glassner for making his *Device Direct Rendering* paper available before it was finished, to the audience at a rather muddled talk at the Cornell Program of Computer Graphics who provided several valuable suggestions, and to Holly Rushmeier and Jack Tumblin for their great assistance in getting us to understand the brightness mapping problem and their algorithm. This work was partially funded by NSF grant *True Virtual Reality Systems*, NSF-CCR-92-09457.

References

[1] Edward C. Carterette and Morton P. Friedman, editors. *Seeing*, volume 5 of *Handbook of Perception*. Academic Press, 1975.

[2] Robert L. Cook. Stochastic sampling in computer graphics. *ACM Transactions on Graphics*, 5(1):51 – 72, Januaary 1986.

[3] Andrew Glassner. *Principles of Digital Image Synthesis (in preparation)*. Morgan-Kaufmann, New York, N.Y., 1993.

[4] Andrew S. Glassner, Kenneith PP. Fishkin, David H. Marimont, and Maureen C Stone. Devicd-directed rendering. Xerox Technical Report (to appear), February 1993.

[5] Roy Hall. *Illumination and Color in Computer Generated Imagery*. Springer-Verlag, New York, N.Y., 1988.

[6] J. Kajiya and M. Ullner. Filtering high quality text for display on raster scan devices. *ACM Siggraph '81 Conference Proceedings*, 15(3):7 – 15, August 1981.

[7] John E. Kaufman, editor. *The Illumination Engineering Society Lighting Handbook*. Waverly Press, Baltimore, MD, 1984.

[8] Edwin H. Land. The retinex theory of color vision. *(The) Scientific American*, 12:108–128, December 1977.

[9] Leo Levi. Survey : Unsharp masking and related image enhancement techniques. *Computer Graphics and Image Processing*, 3:163 – 167, 1974.

[10] David Marr. *Vision*. Wiley, New York, NY, 1982.

[11] Don P. Mitchell and Arun N. Netravali. Reconstruction filters in computer graphics. *ACM Siggraph '88 Conference Proceedings*, 22(4):221 – 228, August 1988.

[12] Eihachiro Nakamae, Kazufumi Kaneda, Takashi Okamoto, and Tomoyuki Nishita. A lighting model aiming at drive simulators. *ACM Siggraph '90 Conference Proceedings*, 24(3):395 – 404, June 1990.

[13] Ken Perlin and Eric M. Hoffert. Hypertexture. *Computer Graphics*, 23(3):253–262, July 1989. ACM Siggraph '89 Conference Proceedings.

[14] Brian E. Smits and Gary W. Meyer. Newton's colors: Simulating interference phenomena in realistic image synthesis. *Proceedings of the Eurographics Workshop on Photosimulation, Realism and Physics in Computer Graphics*, pages 185 – 194, 1990.

[15] J. C. Stevens and S. S. Stevens. Brightness function: Effects of adaptation. *Journal of the Optical Society of America*, 53(3):375–385, March 1963.

[16] S.S. Stevens and J. C. Stevens. Brightness function: Parametric effects of adaptation and contrast. *Journal of the Optical Society of America*, 53(11):1139, November 1960.

[17] Maureen C. Stone and William E. Wallace. Gamut mapping computer generated imagery. *Graphics Interface '91*, pages 32 – 39, June 1991.

[18] Jack Tumblin and Holly Rushmeier. Tone reproduction for realistic computer generated images. Technical Report GIT-GVU-91-13, Graphics, Visualization, and Usuability Center, Georgia Institute of Technology, 1991.

[19] Gregory J. Ward. Visualization. *Lighting Design and Applications*, pages 4–20, August 1990.

[20] Andrew P. Witkin. Scale-space filtering. In Alan Bundy, editor, *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, pages 1019–1021. IJCAI, William Kaufmann, Inc., August 1983.