Interactive Tone Mapping

Frédo Durand and Julie Dorsey

Laboratory for Computer Science
Massachusetts Institute of Technology
fredo@graphics.lcs.mit.edu, dorsey@lcs.mit.edu
http://www.graphics.lcs.mit.edu

Abstract. Tone mapping and visual adaptation are crucial for the generation of static, photorealistic images. A largely unexplored problem is the simulation of adaptation and its changes over time on the visual appearance of a scene. These changes are important in interactive applications, including walkthroughs or games, where effects such as dazzling, slow dark-adaptation, or more subtle effects of visual adaptation can greatly enhance the immersive impression. In applications such as driving simulators, these changes must be modeled in order to reproduce the visibility conditions of real-world situations. In this paper, we address the practical issues of interactive tone mapping and propose a simple model of visual adaptation. We describe a multi-pass interactive rendering method that computes the average luminance in a first pass and renders the scene with a tone mapping operator in the second pass. We also propose several extensions to the tone mapping operator of Ferwerda et al. [FPSG96]. We demonstrate our model for the display of global illumination solutions and for interactive walkthroughs.

1 Introduction

The human visual system performs effectively over a vast range of luminous intensities, ranging from below starlight, at $10^{-6} cd/m^2$, to sunlight, at $10^6 cd/m^2$. The visual system copes with the high dynamic range present in real scenes by varying its sensitivity through a process known as *visual adaptation*. Visual adaptation does not occur instantaneously. Hence, the time course of such variations in luminance is also important. The recovery of sensitivity from dramatic light to dark changes, which can take tens of minutes, is known as *dark adaptation*; the faster recovery from dark to light changes or small light decrements, which can take seconds, is known as *light adaptation*. Moreover, our visual system does not function equally at all illumination levels. In dark scenes we are unable to distinguish colors and our acuity is low, while as the luminance is increased, colors become increasingly vivid and acuity increases. Another fascinating feature of vision is *chromatic adaptation*, which allows us to discount the illuminant: a given object seen under sunlight (bluish) or under a tungsten illuminant (yellowish) will be perceived as having a constant color, although the physical stimuli arriving at the eye have objectively different hues. This explains why a variety of photographic films are necessary to obtain a correct color balance.

The ability to simulate the time course of visual adaptation is important for a variety of visual simulation applications. In architectural walkthroughs, for example, the perceptual effects due to lighting variations between different rooms or between interior and exterior spaces must be displayed with good subjective fidelity in order to convey the ambiance of these environments. For driving simulators, the ability to model adaptation is essential not only for providing a sense of immersion but also in accurately reproducing the visibility conditions actually encountered in real scenes. A particular modeling challenge is the slow dark adaptation and dazzling during light adaptation that a driver would face when entering and then leaving a tunnel. In addition, when driving at night, the visibility of traffic signs and traffic lights is a major concern. A similar need for modeling accuracy exists in the case of immersive VR safety training, for example for hazardous factories. The realism of games can also benefit from visual adaptation simulation. Because these applications also rely on 8 bits displays, the dynamic range cannot exceed two orders of magnitude. The use of Digital Mirror Devices or direct projection on the retina to obtain a higher dynamic range has however great potential.

In order to address the problem of mapping image luminances into the displayable range of a particular output device, a variety of *tone reproduction* operators have been developed [TR93, War94, CHS$^+$93, Sch95, FPSG96, WLRP97, GTS$^+$97, PFFG98, THG99, Tum99, TT99]. However, most existing tone mapping operators are based on steady-state viewing conditions and have been developed as post-processes for static images. They therefore do not model the time course of visual adaptation discussed above, with the notable exception of the work by Ferwerda et al. [FPSG96], which introduces a specific case of transient adaptation, and the forthcoming work by Scheel et al. [SSS00] and Pattanaik et al. [PTYG00]. As a result, although lighting simulation methods are able to simulate a variety of lighting effects, the display of the results still lacks an important part of the perceptual experience. In order to address this problem, the time component must be integrated into the tone mapping process to provide a faithful subjective impression for interactive visualization. This involves first understanding and modeling the mechanisms of human perception, and second being able to perform such a tone mapping interactively. The current paper focuses on the latter, while the former will be the subject of a companion paper [DD00].

This paper describes a system that performs tone mapping interactively. We take as input a 3D model with high dynamic range colors at vertices, and tone mapping is performed on-the-fly while the user walks through the scene. We present a simple model of light adaptation, and use flares to increase the subjective brightness of visible light sources, and simulate the loss of acuity in dark conditions.

1.1 Visual adaptation

In this section we briefly review the mechanisms of visual adaptation. More detailed descriptions can be found in [Fer98, FPSG96, HF86, WECaS90].

The human eye has two different types of photoreceptors. *Cones* are responsible for sharp chromatic vision in luminous conditions, or the *photopic* range. *Rods* provide less precise vision but are extremely sensitive to light and allow us to see in dark conditions, or the *scotopic* range. Both rods and cones are active in moderately luminous conditions, known as the *mesopic* range.

Light adaptation, or simply *adaptation*, is the (fast) recovery of visual sensitivity after an increase or a small decrease in light intensity. Otherwise, the limited range of neurons results in *response compression* for (relatively) high luminances. This is why everything appears white during the dazzling observed when leaving a tunnel. To cope with this and to always make the best use of the small dynamic range of neurons (typically 1 to 40 [WECaS90]), sensitivity is controlled through *multiplicative* (gain-control) and *subtractive* mechanisms. In previous work on tone reproduction operators, all of these mechanisms are modeled as multiplicative; we make this general assumption in our work as well. It is reasonable for the steady state, however since subtractive and multiplicative mechanisms have different time-constants and different effects, they should be differentiated for a more accurate simulation of adaptation [DD00].

Dark adaptation is the (slow) recovery of sensitivity after a dramatic reduction in light. It can take up to tens of minutes. The classic example of this is the adaptation one experiences on a sunny day upon entering a theater for a matinée. Initially, everything inside appears too dark, and visual acuity is, at best, poor.

Note that all of these mechanisms are local, i.e. they occur independently for single receptors or for small "pools" of receptors. In this paper, we will however make the assumption of a global adaptation state for the entire image. This is motivated by efficiency considerations, but also because considering local adaptation states is extremely challenging. The interaction between local adaptation and eye-gaze movements is very complex and is left as a subject of future research. However, choosing a single *adaptation level* (or "average" light intensity) is not trivial, as we will discuss in Section 4.

Chromatic adaptation allows us to perceive objects as having a constant color, even if they are observed under illuminants with varying hues. This type of adaptation involves mechanisms in the retina as well as higher level mechanisms [Fai98]. It can be reasonably modeled as a different gain-control for each cone type, which is called *Von Kries adaptation*. Chromatic adaptation can be complete or partial and takes approximately one minute to completely occur [FR95]. One of the fascinating aspects of this type of adaptation is that it is not driven by the average color of the stimulus, but by the color of the illuminant: our visual system is somehow able to distinguish the color of the illuminant and discount it, similar to the function of the white balance feature of camcoders [DL86, Fai98, Bre87].

1.2 Previous work

A comprehensive review and discussion of the issues in tone mapping can be found in the thesis by Tumblin [Tum99]; we review issues pertinent to our operator below.

Tone mapping operators can be classified into two categories: *global* and *local*. Global operators use the same function for each pixel to map physical radiances into image colors, while local operators vary the mapping across the image. The simplicity of global methods makes them more suitable for interactive applications than local methods. The classic tone reproduction process used in photography, film, and printing is global, using an *S*-shaped response [Hun95].

Tumblin and Rushmeier [TR93] brought the issue of tone reproduction to the attention of the graphics community. Their global tone mapping operator preserves the impression of *brightness*. Ward's *visibility* preserving operator [War94] ensures that the smallest perceptible intensity difference ("Just Noticeable Difference") in the real scene will correspond to the smallest perceptible difference in the image.

Ferwerda et al.'s global visibility preserving operator [FPSG96] accounts for the transition between achromatic night vision and chromatic daylight vision. In addition, this model introduces the time course of dark and light adaptation for two specific conditions: 1) the sudden transition from a steady illuminated scene to a dark scene and 2) the transition from a completely dark scene to a steady illuminated scene. This model also simulates the loss of acuity for low-light vision. Our work builds on this operator in several respects. In particular, we introduce a more psycho-physically based formula for mesopic conditions, and add a blue-shift to improve the appearance of dark scenes. We also introduce models for transient light adaptation and chromatic adaptation. Finally, we improve their formula for the loss of acuity.

Ward-Larson et al. [WLRP97] propose a very efficient and elegant tone mapping operator based on histogram adjustment. Tumblin et al. [THG99] present two original methods; in the first one they decompose an image into *layers*, such as reflectance and illumination. They apply a compression of the high dynamic range only to the illumination layer. In the second method, the adaptation level is specified by the area of interest (*fovea*) which the user selects with a mouse.

On the other hand, local operators [CHS$^+$93, Sch95, PFFG98, TT99] use a different mapping for each pixel, typically depending on the intensity of its surrounding pixels. They are more computationally demanding and are currently not well suited for interactive implementations. The method by Pattanaik et al. [PFFG98] is nonetheless of interest for our work, as it is the only operator that takes chromatic adaptation into account.

The modeling of flares and glares [NKON90, SSZG95, WLRP97, MH99] is an additional method to increase the subjective brightness of parts of the image, such as those containing light sources. In addition, they simulate the loss in visual sensitivity in the direction of strong sources.

All the methods presented so far are applied in a post-processing phase, which takes high dynamic range images as input. Recently, Scheel et al. [SSS00] have proposed a system that performs tone mapping on-the-fly for radiosity solutions with high dynamic range colors at the vertices. They compute the adaptation level using samples obtained through ray-tracing and perform interactive tone mapping using the texturing hardware: the tone mapping function is computed for each frame and stored as a texture. We present an interactive system based on an alternative approach. Our system provides a more general tone mapping operator, albeit at the expense of speed. Pattanaik et al. also recently proposed a simplified model of dynamic adaptation [PTYG00]. However, they do not address interactive applications.

2 System architecture

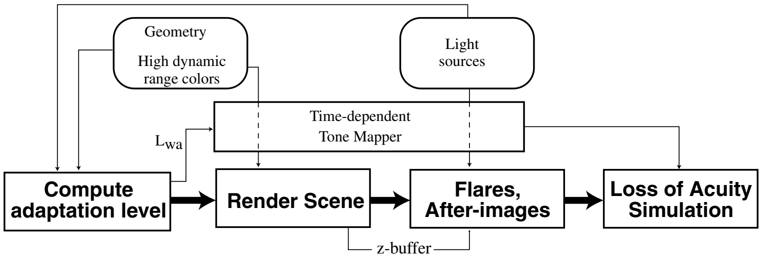


Fig. 1. Architecture of the interactive tone mapping process.

Fig. 1 presents an overview of our interactive system. We take as input a 3D scene with high dynamic range colors at the vertices (typically the output of a lighting simulation, such as radiosity) and a set of point-light sources. We employ a multipass scheme in which we compute the adaptation level, then map this to the displayed colors for each frame, and finally, add flares to improve the appearance of the light sources (Fig. 1). A filtering step is then applied to simulate the loss of acuity due to dim conditions.

As discussed earlier, we consider a single world adaptation level L_{wa} for each frame. This level is computed by first displaying the scene with the log of the colors for each vertex, as will be described in Section 4. This adaptation level is used to drive our time-dependent tone mapping approach that we describe in Section 3. Although we describe our system with this particular tone mapping operator, we have also experimented with other global tone mapping operators [TR93, Tum99, War94, FPSG96, DD00]. Next, we render the actual frame. The scene geometry is sent to the graphics pipeline, and the color of each vertex is mapped on-the-fly (Section 5).

The special case of light sources is crucial to convey a faithful lighting experience. In Section 6 we present our interactive implementation of flares [SSZG95], with a careful treatment of their visibility. Finally, we perform a post-process on the image to simulate the loss of sensitivity due to dim conditions [FPSG96] (Section 7).

2.1 Quantities and units

We briefly summarize the quantities, units, and notation that we will use throughout the remainder of the paper. Quantities corresponding to the scene (world) will have the subscript w, while displayed quantities will have the subscript d. The R and C subscripts will correspond to rods and cones respectively.

For efficiency reasons, our implementation makes the severe assumption that the three RGB components correspond to the three cone types. It is well known that this assumption is not valid for applications where the accuracy of color reproduction is crucial [Hun95, Fai98]. To compensate for this approximation for applications requiring more accuracy, a matrix color transform can be performed at the end of the tone mapping process. The colors in the scene are denoted by r_w, g_w, b_w and are expressed in cd/m^2, the same units as all other quantities. See e.g. [PS86] for conversions. The displayed color is r_d, g_d, b_d.

We use a rod input component s_w for the scotopic range, which in practice is computed using the linear combination given by Pattanaik et al. [PFFG98]. If spectral data are available, a more accurate value could be used (see e.g. [WS82]).

L denotes a luminosity, either the standard formula $L_C = 0.33*r+0.71*g+0.08*b$ for the cone system, or $L_R = s$ for the rod system. Adaptation levels will be expressed as L_a, with various subscripts as described above. We use t to represent time, and ε to represent thresholds.

3 Time-dependent tone mapping

The tone mapping operator that we employ builds on the one developed by Ferwerda et al. [FPSG96]. We propose several improvements, including a blue-shift for viewing night scenes and chromatic adaptation. In addition, we extend this operator to time-dependent tone mapping by incorporating a simple model of visual adaptation.

3.1 Review of Ferwerda et al.'s operator

As is the case with Ward's operator [War94], this method uses a global scale-factor: $L_d = m\, L_w$. It is based on *threshold mapping*: the threshold in the real scene $\varepsilon(L_{wa})$ is mapped onto the threshold of the display $\varepsilon(L_{da})$. The scaling factor used is thus $m = \frac{\varepsilon(L_{da})}{\varepsilon(L_{wa})}$. The thresholds for rods and cones respectively are:

$$\log \varepsilon_R(L_{aR}) = \begin{cases} -2.86 & if\ \log L_{aR} \le -3.94, \\ \log L_{aR} - 0.395 & if\ \log L_{aR} \ge -1.44, \\ (0.405 \log L_{aR} + 1.6)^{2.18} - 2.86 & otherwise. \end{cases} \quad (1)$$

$$\log \varepsilon_C(L_{aC}) = \begin{cases} -0.72 & if\ \log L_{aC} \le -2.6, \\ \log L_{wa} - 1.255 & if\ \log L_{aC} \ge 1.9, \\ (0.249 \log L_{aC} + 0.65)^{2.7} - 0.72 & otherwise. \end{cases} \quad (2)$$

Since the observer viewing the display is assumed to be in the photopic state, the scale factor computed for rods uses the threshold of the cones in the display state $m_R = \frac{\varepsilon_C(L_{daC})}{\varepsilon_R(L_{waR})}$.

Two responses are computed: one chromatic for the cone system and one achromatic for the rod system. These responses are then added, and a factor k is used to simulate the loss of rod sensitivity in the mesopic range:

$$\begin{aligned} r_d &= m_C\, r_w + k\, m_R\, s_w \\ g_d &= m_C\, g_w + k\, m_R\, s_w \\ b_d &= m_C\, b_w + k\, m_R\, s_w \end{aligned} \tag{3}$$

The original Ferwerda et al. model uses the following formula for k [FP00]: $k = \left(1 - \frac{L_{wa}/2 - 0.01}{10 - 0.01}\right)^2$ clamped to between 0 and 1. However, this formula is an approximation and has no perceptual basis.

3.2 Improvements

We replace their formula for the mesopic factor k by the saturation function used by Walraven and Valeton to fit the rod threshold for higher luminances [WV84]:

$$k = \frac{\sigma - 0.25\, L_{waR}}{\sigma + L_{waR}}, \tag{4}$$

clamped to 0, where $\sigma = 100\ cd/m^2$. (The whole formula (18) of [WV84] could also be used in place of Eq. 1. We have chosen to keep Eq. 1 to minimize the changes to Ferwerda et al.'s operator [FPSG96].)

Night scenes are often represented with a blue hue in paintings and cinema [Mil91]. Indeed, Hunt notes that the subjective hue of colors undergoes a blue shift for dark scenes [Hun52]. Hunt's data show that the subjective color of a white sample in very dark conditions has a CIE hue of $x = 0.3$, $y = 0.3$ — that is, a normalized $RGB = (1.05, 0.97, 1.27)$ on a typical NTSC display. This is further discussed by Trezona [Tre70] who suspects that rod signals are interpreted as slightly bluish. Indeed, rods share many neural pathways with short wavelength cones. Based on these conclusions, we use the following formula to model the blue shift:

$$\begin{aligned} r_d &= m_C\, r_w + 1.05\, k\, m_R\, s_w \\ g_d &= m_C\, g_w + 0.97\, k\, m_R\, s_w \\ b_d &= m_C\, b_w + 1.27\, k\, m_R\, s_w \end{aligned} \tag{5}$$

Note that a bluer color can be used for a more dramatic effect.

3.3 A simple model of light adaptation

In this paper we present only a model for light adaptation, since the time constants involved in dark adaptation (up to tens of minutes) make it less important for walkthroughs, unless accurate visibility is required. Moreover, the complex mechanisms required to simulate dark adaptation are beyond the scope of this article.

For simplicity and like previous authors [FPSG96, War94], we simulate both multiplicative and subtractive light adaptation as the multiplicative gain-control m described in the previous section.

The transient recovery of sensitivity is simulated using exponential filters on the values of m: $\frac{dm_R}{dt} = \frac{m_{R\,steady} - m_R(t)}{\tau_R}$ and $\frac{dm_C}{dt} = \frac{m_{C\,steady} - m_C(t)}{\tau_C}$, where m_{steady} is the scaling factor in the steady state (Eq. 2 and 1). Using data from [Ade82, HBH87] we use $\tau_R = 0.4\ sec$ for rods and $\tau_C = 0.1\ sec$ for cones.

3.4 A simple model of chromatic adaptation

The achromatic cone scaling factor m_C obtained from the light adaptation must be adapted for each cone type in order to account for chromatic adaptation. We first describe the adaptation correction in the steady state before describing its time-course.

Chromatic adaptation is driven mainly by the chromaticity of the illuminant, or equivalently, by the normalized chromaticity of a white object, (r_{wa}, g_{wa}, b_{wa}). The effect of the reflectances present in the scene is weak [Bre87, Bäu99]. In this paper, we assume a complete chromatic adaptation:

$$\begin{aligned} r_d &= \tfrac{m_C}{r_{wa}}\, r_w + 1.05\, k\, m_R\, s_w \\ g_d &= \tfrac{m_C}{g_{wa}}\, g_w + 0.97\, k\, m_R\, s_w \\ b_d &= \tfrac{m_C}{b_{wa}}\, b_w + 1.27\, k\, m_R\, s_w \end{aligned} \tag{6}$$

Chromatic adaptation takes more than one minute to complete, but most of it is complete in a matter of seconds [FR95]. We use an exponential filter on the values (r_{wa}, g_{wa}, b_{wa}) with time constant $\tau_{chroma} = 5\ sec$.

4 Adaptation level computation

4.1 Interactive implementation

To compute the adaptation level, we first render a version of the scene in which each vertex is assigned a color corresponding to the log of its high-dynamic range color $(\log r_w, \log g_w, \log b_w)$. The log of the rod intensity is mapped to the alpha channel. A lower resolution image can be used for this pass; in practice, we use 64×64 or 128×128. If LODs are available, coarser versions are used for acceleration.

As aforementioned, chromatic adaptation depends on the color of the illuminant, not on the color of the objects of the scene. Brown has moreover shown that the pixels of an image usually do not average to grey [Bro94]. We thus do not exactly use the log of the colors of the vertices. Instead, the chromaticity $(r_{ill}, g_{ill}, b_{ill})$ of the incoming light (irradiance) is instead used together with the luminance of the outgoing light, in a way similar to Neuman et al. [NMNP98]. The color used for each vertex is then:
$\log\left(\frac{L_w}{L_{ill}} r_{ill}\right), \log\left(\frac{L_w}{L_{ill}} g_{ill}\right), \log\left(\frac{L_w}{L_{ill}} b_{ill}\right)$

These values are linearly mapped from $[-6, 6]$ to $[0, 1]$. If, following Ward [War94], we use a non-weighted average, we simply compute the mean value of the pixels. Similarly, the histogram of scene values can be computed.

Building on Tumblin et al. [THG99] and Scheel et al. [SSS00], we also can take advantage of the observer's gaze and use a weighted average. For this, we alpha-blend the log-rendered scene with a texture containing the weights. See the following section for a discussion of the metering strategies. The texture is then rendered alone to compute the total of the weights (i.e. the normalization factor). See Fig. 2(a) in the Appendix for an example.

4.2 Exposure metering

As discussed by Scheel et al. [SSS00], computing the adaptation level is very similar to exposure metering for photography. This analogy also shows the complexity of the problem, since metering is one of the hardest tasks a photographer confronts. To illustrate this difficulty, today's state-of-the-art automatic metering system, Nikon's widely acclaimed 3D matrix system [Nik00], collects metering data from eight zones of the picture, computes contrast between these zones, and then uses distance, color, focus information, and a database of 30,000 images to choose the best exposure!

More typical measurements usually involve a centered weighted average. Nikon's weighted centered metering uses 75% of the weight in the 12mm center of a 24x36 film [Nik00]. This is the solution that we generally use, however, spot-centered metering is also available using Gaussian weights.

5 On-the-fly tone mapping

Once the adaptation levels have been used to update the tone mapper, the scene is rendered as usual, and the displayed colors of the vertices are computed on-the-fly with our tone mapping operator.

We accelerate tone mapping by caching the function in Look–Up-Tables. This is not crucial for simple tone mapping operators such as the one described in this paper, but it is more important if gamma-correction or more complex operators are used, such as those described in [TR93, DD00]. The tables are re-computed once for each frame and are indexed by the logarithm of r_w, g_w, b_w and s_w. We use 2000 values for the whole 10^{12} range, which provides enough resolution [SFB92]. We thus do not have to use interpolation between the two closest entries in the table.

6 Light sources

6.1 Flares

Flares have been shown to be of great importance in increasing the subjective dynamic range of images [NKON90, SSZG95]. We have implemented an interactive version of the Spencer et al. method [SSZG95]. This method convolves the scene's high-dynamic range pixels with a filter based on psycho-physical models consisting of the following basis functions, where *lambda* is the wavelength in *nm* and θ the angle in *radians*:

$$\begin{array}{lclcrcl} f_0(\theta) & = & 2.61*10^6 e^{-\left(\frac{\theta}{0.02}\right)^2} & & f_1(\theta) & = & \frac{20.91}{(\theta+0.02)^3} \\ f_2(\theta) & = & \frac{72.37}{(\theta+0.02)^2} & & f_3(\theta,\lambda) & = & 436.9\frac{568}{\lambda}e^{-(\theta-3\frac{\lambda}{568})^2} \end{array} \qquad (7)$$

The filter used to convolve the scene is then a weighted average of these basis functions, each f_i having weight w_i. Spencer et al. propose three different filters (i.e. three sets of weights) for photopic, mesopic, and scotopic conditions. Based on their conclusions, we assume that the weights of the different sorts of flares depend on the pupil diameter D (in *mm*). We propose the following formulas, which are linear interpolations between their three sets of weights:

$$\begin{array}{lcllcl} w_1 & = & 0.478 & w_2 & = & 0.138 + 0.08(D(L_{wa}) - 2)/5 \\ w_3 & = & 0.033(D(L_{wa}) - 2)/5 & w_0 & = & 1 - w_1 - w_2 - w_3, \end{array} \qquad (8)$$

where $D(L_{wa}) = 4.9 - 3\tanh(0.4\log(0.5(L_{waR} + L_{waC}) + 1))$ [SSZG95].

6.2 Interactive Flares

When applied to the whole image, flares are costly and moreover require floating point values for pixels. We thus restrict the addition of flares to light sources. The implementation of interactive flares usually uses textured polygons facing the viewpoint [MH99]. Unfortunately, this causes artifacts since the flares are then tested for occlusion as well, although their visibility should depend only on the visibility of the point light source.

To cope with this, we first render all the geometry, then read the z-buffer and use it in software to query the visibility of the light sources. The hardware z-buffer is disabled, and a flare is rendered only if the pixel corresponding to the point light source has a depth lower or equal to the depth of the light source. Alternatively, ray-casting or, even better, hardware occlusion-culling flags, could also be used if available.

$L_{wa}\ (cd/m^2)$	-3.5	-3	-2.5	-2	-1.5	-1	-0.5	0	0.5	1	1.5	2	2.5	3
$\omega\ (cyc/deg)$	2.1	2.9	4.1	5.5	9	16.3	23.8	32.5	38.5	43.1	46	48	48.8	50

Table 1. Tabulated data from Shaler's acuity experiments [Sha37], expressing maximum visible spatial frequency ω vs. adaptation luminance L_{wa}.

7 Loss of visual acuity

Ferwerda et al. simulate the loss of acuity in low light by applying a 2D Gaussian blur filter g to the image [FPSG96]. They base the size of the filter on data from Shaler [Sha37] (see Table 1). They choose the filter at the given adaptation level L_{waR} such that the highest perceptible frequency $\omega(L_{waR})$ for a grating with high contrast as used by Shaler (black and L_{waR}) is displayed at the scotopic threshold $\varepsilon_R(L_{waR})$. Using capitals for the Fourier transform, this gives:

$$G(\omega(Lwa)) = \frac{\varepsilon_R(L_{waR})}{L_{waR}} \tag{9}$$

Unfortunately, with this formula, blurring does not always increase when the luminance decreases. At very low light levels, $\frac{\varepsilon_R(L_{waR})}{L_{waR}}$ increases with light decrements: $\varepsilon_R(L_{waR})$ becomes constant as absolute sensitivity is reached, and the ratio increases. This means that the cutoff frequency is reduced to a lesser degree, which results in images that are less blurry for darker scenes. To cope with this, we use the constant ratio $10^{-0.395}$ computed from the linear portion of the rod threshold function (Eq. 1).

If $a(L_{waR})$ defines the width of our Gaussian and r is the radius, leaving the parameter L_{waR} for clarity, we have the unit volume 2D Gaussian $g(r) = a^2 e^{-\pi(a\,r)^2}$, and we want: $G(\omega) = 10^{-0.395}$. Since the Fourier transform of a Gaussian is also a Gaussian[1], we obtain

$$\begin{aligned} \frac{1}{a^2} a^2 e^{-\pi\frac{\omega^2}{a^2}} &= 10^{-0.395} \\ \frac{\omega}{a} &= \sqrt{\frac{-\ln 10^{-0.395}}{\pi}} \\ a &= 1.86\,\omega. \end{aligned} \tag{10}$$

We use hardware convolution, available in OpenGL 1.2 and on SGI machines, to perform this Gaussian blur. We use the tabulated version of ω given in Table 1 with linear interpolation. The values of a and ω must be converted into pixels using the perspective field of view fov and display resolution $width_{pixel}$.

$$a_{pixel^{-1}}(L_{waR}) = 1.86\,\omega(L_{waR}) * fov/width_{pixel} \tag{11}$$

For efficiency purposes, we apply convolution only when necessary (in practice when $a_{pixel^{-1}}$ is smaller than 1 $pixel^{-1}$), and we interpolate between 1 and 1.1 $pixels^{-1}$ to obtain a smooth transition. The additional cost of convolution could then be compensated for by the use of coarser levels of detail. We plan to implement this solution in the near future.

Like Ferwerda et al., our implementation assumes a fixed distance between the viewer and the display. If a tracking of the observer's position is available (e.g. as in CAVE systems), it can be used to change Eq. 11 for each frame.

[1] Note that we use the definition of the Fourier transform given by Bracewell [Bra95] pp. 140-154, which is consistent with our definition of ω.

8 Results

We have tested our system on the output of a radiosity global illumination program. All of the results presented are rendered directly from 3D geometry on a Silicon Graphics Onyx2 Infinite Reality using one R10K processor, with the method described in the paper.

Our examples include a street at night with a car passing by (Fig. 2(a), see Appendix), a walk from the outside to the inside of a house with both yellowish incandescent and neon lighting, an indoor scene with light turned on (Fig. 4) and off. We typically obtain a framerate of 30Hz for the 11,000 triangle scenes, and 6 Hz for a 80,000 triangle model.

Fig. 3 illustrates the importance of using the chromaticity of the light source and not the average chromaticity of the image. The log of the color used in the first pass to compute the adaptation level of Fig. 3(c) is computed with our method.

Our experiments suggest that maintaining a high frame rate (at least 30 Hz) is important. Since light adaptation has small time-constants, a smooth frame rate greatly improves the transient effects. Simple optimizations should be implemented in our system. For example, radiosity meshes are usually too finely subdivided, we use no triangle strips and the levels of detail we use for the adaptation level computation only decrease the polygon count by a factor of 2.

As far as metering is concerned, weighted centered metering is better when the user explores the environment, but uniform average provides a smoother variation of the adaptation level and is more suitable for architectural walkthroughs. Coupling the adaptation level metering with a gaze tracking system would certainly permit optimal results.

The loss of acuity convolution causes the frame rate to drop from 30Hz to 7.5Hz on the gallery scene. It is effective only for dim scenes (below 1.5 $\log cd/m^2$, with a 50° field of view at video resolution). It is most apparent for high resolution images, as the display resolution is otherwise the major frequency limitation.

9 Conclusions and future work

In this paper we have introduced an interactive tone mapping system, simple extensions to Ferwerda et al.'s operator [FPSG96], and a simplified model of light and chromatic adaptation. Our viewer performs tone mapping for each frame on the high dynamic range colors of the vertices of the 3D geometry. Flares are added to increase the subjective dynamic range of the images, and loss of acuity is simulated.

Our system is obviously slower than the solution proposed by Scheel et al. [SSS00], since they avoid the CPU-consuming on-the-fly mapping that we perform and can instead take advantage of OpenGL display lists. However, their method is unable to use a sophisticated tone mapping such as [FPSG96] or the one described in this paper. Note that some components of the two systems are interchangeable, e.g. their ray-tracing-based adaptation level computation could be plugged into our system and vice-versa.

Our system would greatly benefit from a multiprocessing API using pipelined deferred rendering such as *IRIS Performer* [RH94]. On-the-fly tone mapping could be performed on a separate processor, without impeding the frame rate.

Few restrictions prevent a better hardware integration of tone mapping. Using 12 bits per pixel and a 4th component for rods could be an interesting alternative for specialized hardware, such as driving simulators for which accurate simulation of visual adaptation is important. Textures are currently hard to handle with interactive tone mapping because the hardware performs clamping on vertex colors before texture mapping, rather than on a per pixel basis. A pixel of the texture with value $v < 1.0$ thus cannot be mapped to a color greater than v, preventing the depiction of the white impression due to response compression during dazzling.

Many interesting problems in the area of tone mapping remain. A forthcoming paper will present a more elaborate model of visual adaptation [DD00] for interactive applications. A great deal of work remains to be done, especially for local methods, where the conjugate effect of gaze saccades and local adaptation is quite complicated. Tone mapping in mesopic conditions is another challenging area, as the complex interaction between rods and cones and their effect on the appearance of colors must be simulated.

Acknowledgments

We thank Stephen Duck, Max Chen and Gernot Schaufler for the models that they built for this project, and Jim Ferwerda and Sumant Pattanaik, who kindly answered our questions about their adaptation model. Thanks to the reviewers and Steven Gortler for their advice. This work was supported by an NSF Postdoctoral Research Associates award (EIA-9806139), an NSF CISE Research Infrastructure award (EIA-9892229), an NSF-INRIA Cooperative Research award (INT-9724005), and a grant from Intel.

References

Ade82. Adelson. Saturation and adaptation in the rod system. *Vision Research*, 22:1299, 1982.

Bäu99. Bäuml. Color constancy: the role of image surface in illuminant adjustment. *JOSA A*, 16(7):1521, 1999.

Bra95. R. Bracewell. *Two-Dimensional Imaging*. Prentice Hall, 1995.

Bre87. Breneman. Corresponding chromaticities for different states of adaptation to complex visual fields. *JOSA A*, 4(6):115, 1987.

Bro94. Brown. The world is not grey. *Investigative Ophtalmology and Visual Sc*,, 35(Suppl.), 1994.

CHS^{+}93. Chiu, Herf, Shirley, Swamy, Wang, and Zimmerman. Spatially nonuniform scaling functions for high contrast images. In *Proc. Graphics Interface*, 1993.

DD00. Durand and Dorsey. A computational model of visual adaptation for time-dependent tone-mapping. *submitted for publication*, 2000. http://graphics.lcs.mit.edu/˜fredo.

DL86. D'Zmura and Lennie. Mechanisms of color constancy. *JOSA A*, 10:1662, 1986.

Fai98. Fairchild. *Color Appearance Models*. Addison-Wesley, 1998.

Fer98. Ferwerda. Fundamentals of spatial vision. In *Applications of visual perception in computer graphics*, 1998. Siggraph '98 Course Notes.

FP00. Jim Ferwerda and Sumant Pattanaik. personal communication, 2000.

FPSG96. Ferwerda, Pattanaik, Shirley, and Greenberg. A model of visual adaptation for realistic image synthesis. In *Computer Graphics (Proc. Siggraph)*, 1996.

FR95. Fairchild and Reniff. Time course of chromatic adaptation for color-appearance judgments. *JOSA A*, 12(5):824, 1995.

GTS^{+}97. Greenberg, Torrance, Shirley, Arvo, Ferwerda, Pattanaik, Lafortune, Walter, Foo, and Trumbore. A framework for realistic image synthesis. In *Computer Graphics (Proc. Siggraph)*, 1997.

HBH87. Hayhoe, Benimoff, and Hood. The time course of multiplicative and subtractive adaptation process. *Vision Research*, 27:1981, 1987.

HF86. Hood and Finkelstein. Sensitivity to light. In Boff, Kaufman, and Thomas, editors, *Handbook of Perception and Human Performance*. Wiley and Sons, 1986.

Hun52. Hunt. Light and dark adaptation and the perception of color. *JOSA A*, 42(3):190, 1952.

Hun95. Hunt. *The reproduction of Color (5th ed.)*. Kings Langley:Fountain Press, 1995.

MH99. Möller and Haines. *Real-Time Rendering*. A.K. Peters Ltd., 1999.

Mil91. Millerson. *Lighting for Television and Film, 3rd ed.* Focal Press, 1991.

Nik00. Nikon. http://www.nikon.ca, 2000.

NKON90. Nakamae, Kaneda, Okamoto, and Nishita. A lighting model aiming at drive simulators. In *Computer Graphics (Proc. Siggraph)*, 1990.

NMNP98. Neumann, Matkovic, Neumann, and Purgathofer. Incident light metering in computer graphics. *Computer Graphics Forum*, Dec. 1998.

PFFG98. Pattanaik, Ferwerda, Fairchild, and Greenberg. A multiscale model of adaptation and spatial vision for realistic image display. In *Computer Graphics (Proc. Siggraph)*, 1998.

PS86. Pokorny and Smith. Colorimetry and color discrimination. In Boff, Kaufman, and Thomas, editors, *Handbook of Perception and Human Performance*. Wiley and Sons, 1986.

PTYG00. Pattanaik, Tumblin, Yee, and Greenberg. Time-dependent visual adaptation for realistic image display. In *Computer Graphics (Proc. Siggraph)*, 2000. to appear.

RH94. Rohlf and Helman. IRIS performer: A high performance multiprocessing toolkit for real–Time 3D graphics. In *Computer Graphics (Proc. Siggraph)*, 1994.

Sch95. Schlick. Quantization techniques for visualization of high dynamic range pictures. *Eurographics Workshop on Rendering*, 1995.

SFB92. Stokes, Fairchild, and Berns. Precision requirement for digital color reproduction. *ACM Trans. on Graphics*, 11:406, 1992.

Sha37. Shaler. The relation between visual acuity and illumination. *J. of General Physiology*, 21:165, 1937.

SSS00. Scheel, Stamminger, and Seidel. Tone reproduction for interactive walkthroughs. In *Proc. of Eurographics*, 2000. to appear.

SSZG95. Spencer, Shirley, Zimmerman, and Greenberg. Physically-based glare effects for digital images. In *Computer Graphics (Proc. Siggraph)*, 1995.

THG99. Tumblin, Hodgins, and Guenter. Two methods for display of high contrast images. *ACM Trans. on Graphics*, 18(1), January 1999.

TR93. Tumblin and Rushmeier. Tone reproduction for realistic images. *IEEE Computer Graphics and Applications*, 13(6), 1993.

Tre70. P. W. Trezona. Rod participation in the "blue" mechanism and its effect on colour matching. *Vision Research*, 10:317, 1970.

TT99. Tumblin and Turk. LCIS: a boundary hierarchy for detail-preserving contrast reduction. In *Computer Graphics (Proc. Siggraph)*, 1999.

Tum99. Tumblin. *Three methods of detail-preserving contrast reduction for displayed images*. PhD thesis, College of Computing Georgia Inst. of Technology, Sep. 1999.

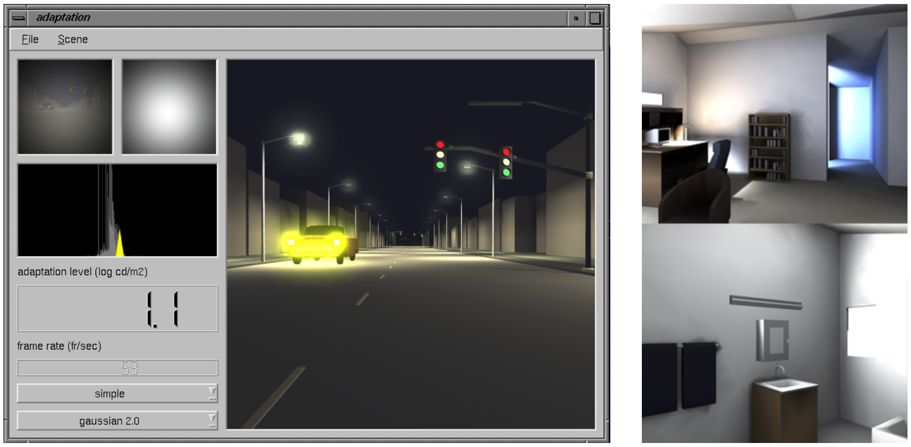
War94. Ward. A contrast-based scalefactor for luminance display. In Heckbert, editor, *Graphics Gems IV*, page 415. Academic Press, 1994.

WECaS90. Walraven, Enroth-Cugell, Hood andMacLeod, and Schnapf. The control of visual sensitivity. In Spillmann and Werner, editors, *Visual Perception, The neurophysiological foundation*. Academic Press, 1990.

WLRP97. Ward-Larson, Rushmeier, and Piatko. A Visibility Matching Tone Reproduction Operator for High Dynamic Range Scenes. *IEEE Trans. on Visualization and Computer Graphics*, 3(4), 1997.

WS82. Wyszecki and Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulae*. Wiley, 1982.

WV84. Walraven and Valeton. Visual adaptation and response saturation. In A. J. van Doorn, W. A. Van de Grind, and J. J. Koenderink, editors, *Limits in Perception*. VNU Science Press, 1984.



adaptation level (log cd/m2)

1.1

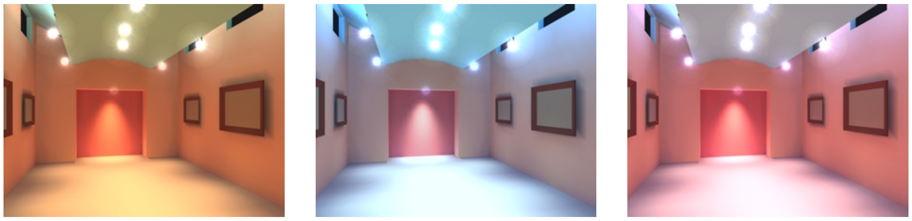
frame rate (fr/sec)

simple

gaussian 2.0

(a) (b)

Fig. 2. (a) Our interactive tone mapper for a street scene (70 ktri, 6Hz). The upper left window displays the scene with log colors. The window on the right is used to compute the normalization factor for the weighted average.Below is the histogram of scene luminance. (b) House scene (80 ktri, 6.6Hz). Top: Living room (1.4 $\log cd/m^2$). Note the bluish lighting of the adjacent bathroom. Bottom: Bathroom after chromatic adaptation (1.8 $\log cd/m^2$).



(a) (b) (c)

Fig. 3. Chromatic adaptation for a scene with yellowish light sources (10 ktri, 30Hz). (a) No chromatic adaptation. (b) Adaptation to the image chromaticity; since the objects are reddish, the image looks too blue. (c) Adaptation to the source chromaticity.

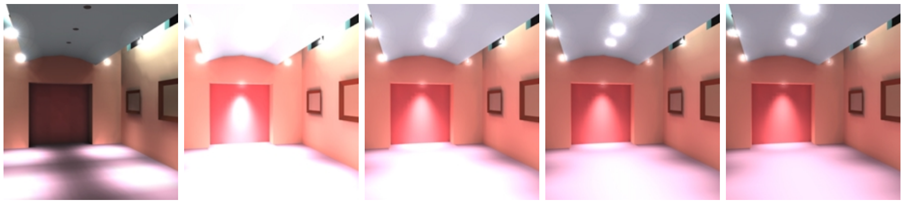


Fig. 4. Light adaptation when primary lights are turned on (from 1.4 $\log cd/m^2$ to 1.9 $\log cd/m^2$, 10 ktri, 30Hz). Frames are 30ms apart.