# Perceptual Effects in Real-time Tone Mapping

Grzegorz Krawczyk* Karol Myszkowski† Hans-Peter Seidel‡
MPI Informatik, Saarbrücken, Germany

## Abstract

Tremendous progress in the development and accessibility of high dynamic range (HDR) technology that has happened just recently results in fast proliferation of HDR synthetic image sequences and captured HDR video. When properly processed, such HDR data can lead to very convincing and realistic results even when presented on traditional low dynamic range (LDR) display devices. This requires real-time local contrast compression (tone mapping) with simultaneous modeling of important in HDR image perception effects such as visual acuity, glare, day and night vision. We propose a unified model to include all those effects into a common computational framework, which enables an efficient implementation on currently available graphics hardware. We develop a post processing module which can be added as the final stage of any real-time rendering system, game engine, or digital video player, which enhances the realism and believability of displayed image streams.

**CR Categories:** I.3.3 [Computer Graphics]: Display algorithms

**Keywords:** tone mapping, luminance adaptation, visual acuity, scotopic vision, glare, graphics hardware

## 1 Introduction

Computer modeling of the real world appearance involves reproducing high dynamic range (HDR) luminance values in resulting images. Traditionally, such HDR images have been the domain of physics-based global illumination computation, which is very costly. Recent progress towards improving the realism of images that was achieved through the use of captured HDR environment maps [Debevec and Malik 1997] and precomputed radiance transfer (PRT) techniques [Sloan et al. 2002; Ng et al. 2004] enables to generate HDR image sequences in real time. Furthermore, HDR video of real world scenes can easily be captured using multi-exposure techniques [Kang et al. 2003] or advanced video sensors [Nayar and Branzoi 2003] and compressed for an efficient storage and later playback [Mantiuk et al. 2004]. However, contrast in such rendered image sequences and HDR video streams often exceeds capabilities of typical displays and their direct rendering to the screen is not possible. Fortunately, the importance of tone mapping of HDR data is widely understood and many algorithms have already been developed [Devlin et al. 2002] with several implementations achieving real-time performance [Drago et al. 2003; Goodnight et al. 2003], which is required especially in real-time rendering systems and for HDR video playback.

Current real-time approaches to tone mapping are however far from perfect because the perceptual effects typical in everyday observation are generally neglected. The fact that a daylight scene appears very bright and colorful, but during the night everything looks dark and grayish, is obvious for an average observer. Although the lack of sharp vision and color perception in night scenes have already been accounted for in tone mapping [Ferwerda et al. 1996; Durand and Dorsey 2000], these effects were applied to the whole image with a uniform intensity (globally). This could lead to unrealistic exaggerations if a wide range of luminance was shown in an image. Very appealing results are obtained using the so-called local tone mapping operators [Devlin et al. 2002], which are very good at preserving fine details in a scene. However, in dimly illuminated scenes such fine details would not be perceivable, because the acuity of human vision is degraded. Therefore, so well preserved details give an unrealistic impression in such cases. On the other hand, certain perceptual effects, like glare, cannot be evoked because the maximum luminance of typical displays is not high enough. However, we are so used to the presence of such a phenomena, that adding glare to an image can increase subjective brightness of the tone mapped image [Spencer et al. 1995]. Clearly, it appears crucial to properly predict and simulate these perceptual effects during the tone mapping process in order to convey a realistic impression of HDR data over a wide range of luminance, when such data are displayed on typical display devices.

Although the account for perceptual effects during tone mapping have already been included in several algorithms for static images [Ferwerda et al. 1996; Ward et al. 1997; Durand and Dorsey 2000], these effects were discussed only in the context of global operators. Furthermore, the proposed solutions were composed of multiple stages each involving complex processing such as convolutions. Although the implementation of individual perceptual effects on graphics hardware is intuitive, a naive combination to include all of them does not even allow for an interactive performance on the best graphics hardware currently available.

In this paper, we present an efficient way to combine the most significant perceptual effects in the context of perception of HDR images including the local tone mapping to adjust contrast to LDR devices. Previously, the perceptual effects have been approached separately, however they have much in common in terms of spatial analysis. In this work, we show that making use of such similarities have a tremendous impact on the performance. With only one costly step of constructing a Gaussian pyramid, we are able to correctly realize all of the effects and tone mapping with no additional overhead. We implement our approach in graphics hardware as a stand-alone HDR image/video processing module and achieve the real-time performance. Such a module can be used in the final rendering stage of any real-time renderer or HDR video player.

The remainder of the paper is organized as follows. In Section 2 we explain the tone mapping method and give background information to each perceptual effect that we simulate. We then present our method of combining the tone mapping with the effects along with the implementation details in Section 3. Finally, we discuss our results in comparison with other approaches in Section 4 and conclude the paper in Section 5.

*e-mail: krawczyk@mpi-sb.mpg.de
†e-mail: karol@mpi-sb.mpg.de
‡e-mail: hpseidel@mpi-sb.mpg.de

## 2 Background

With many tone mapping algorithms available, we want to use a method that provides good, widely acknowledged results for static images. At the same time we want that the spatial analysis involved in tone mapping bare some similarities to processes taking place in the perception. We have found that photographic tone reproduction [Reinhard et al. 2002] satisfies our requirements. In the following sections, we briefly explain the tone mapping algorithm and each of the perceptual effects that we include, and we show the apparent similarities in the spatial analysis of perceived images.

Throughout the tone mapping pipeline, we assume the RGB color model where each channel is described by a positive floating point number. For the proper estimation of the simulated perceptual effects, the pixel intensity values should be calibrated to $\frac{cd}{m^2}$. In our implementation, we consider the values to be in range from $10^{-4}$ to $10^{8}$, which is sufficient to describe the luminance intensities perceivable by the human vision. The algorithm produces tone mapped RGB floating point values in the range $[0:1]$ which are then discretized to 8-bit values by an OpenGL driver.

### 2.1 Tone Mapping

The algorithm proposed by Reinhard et al. [2002] operates solely on the luminance values which can be extracted from RGB intensities using the standard CIE XYZ transform (D65 white point). The method is composed of a global scaling function and a local dodging & burning technique, which allows to preserve fine details. The results are driven by two parameters: the adapting luminance for the HDR scene and the key value. The adapting luminance ensures that the global scaling function provides the most efficient mapping of luminance to the display intensities for given illumination conditions in the HDR scene. The key value controls whether the tone mapped image appears relatively bright or relatively dark.

In this algorithm, the source luminance values $Y$ are first mapped to the relative luminace $Y_r$:

$$Y_r = \frac{\alpha \cdot Y}{\bar{Y}} \tag{1}$$

where $\bar{Y}$ is the logarithmic average of the luminance in the scene, which is an approximation of the *adapting luminance*, and $\alpha$ is the key value. The relative luminance values are then mapped to the displayable pixel intensities $L$ using the following function:

$$L = \frac{Y_r}{1+Y_r} \tag{2}$$

The above formula maps all luminance values to the $[0:1]$ range in such way that the relative luminance $Y_r = 1$ is mapped to the pixel intensity $L = 0.5$. This property is used to map a desired luminance level of the scene to the middle intensity on the display. Mapping higher luminance level to middle gray results in a subjectively dark image (low key) whereas mapping lower luminance to middle gray will give a bright result (high key) (see Figure 1). Obviously, images which we perceive at night appear relatively dark compared to what we see during a day. We can simulate this impression by modulating the key value in equation (1) with respect to the adapting luminance in the scene. We explain our solution in Section 3.1.

Unfortunately, the tone mapping function in equation (2) may lead to the loss of fine details in the scene due to the extensive contrast compression. Reinhard et al. [2002] propose a solution to preserve

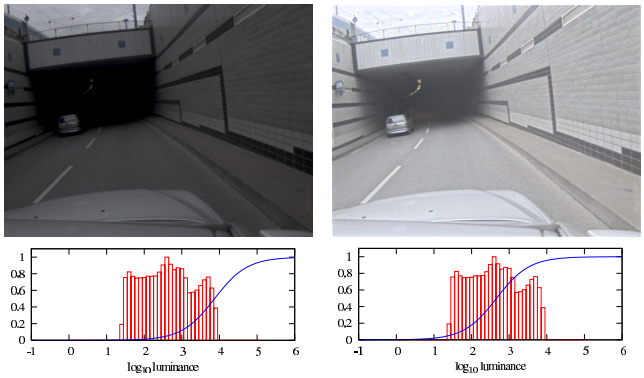


Figure 1: Tone mapping of an HDR image with a low key (left) and a high key (right). The curve on the histograms illustrates how the luminance is mapped to normalized pixel intensities. Refer to Section 2.1 for details.

local details by employing a spatially variant local adaptation value $V$ in equation (2):

$$L(x,y) = \frac{Y_r(x,y)}{1+V(x,y)} \tag{3}$$

The local adaptation $V$ equals to an average luminance in a surround of the pixel. The problem lies however in the estimation of how large the surround of the pixel should be. The goal is to have as wide surround as possible, however too large area may lead to well known inverse gradient artifacts, *halos*. The solution is to successively increase the size of a surround on each *scale* of the pyramid, checking each time if no artifacts are introduced. For this purpose a Gaussian pyramid is constructed with successively increasing kernel:

$$g(x,y,s) = \frac{1}{\pi s^2} \cdot e^{-\frac{x^2+y^2}{s^2}} \tag{4}$$

The Gaussian for the first scale is one pixel wide, setting kernel size to $s = (2\sqrt{2})^{-1}$, on each following scale $s$ is 1.6 times larger. The Gaussian functions used to construct seven scales of the pyramid are plotted in Figure 2. As we later show, such a pyramid is very useful in introducing the perceptual effects to tone mapping.

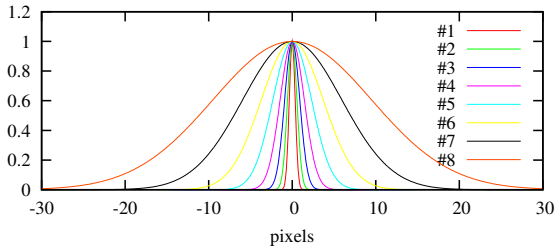


Figure 2: Plot of the Gaussian profiles used to construct the scales of the pyramid used for local dodging & burning in the tone mapping algorithm. The smallest scale is #1 and the largest is #8.

### 2.2 Temporal Luminance Adaptation

While tone mapping the sequence of HDR frames, it is important to note that the luminance conditions can change drastically from frame to frame. The human vision reacts to such changes through

the temporal adaptation processes. The time course of adaptation differs depending on whether we adapt to light or to darkness, and whether we perceive mainly using rods (during night) or cones (during a day). Many intricate models have been introduced to computer graphics, however it is not as important to faithfully model the process as to account for it at all [Goodnight et al. 2003].

In the tone mapping algorithm chosen by us, the luminance adaptation can be modelled using the adapting luminance term in equation (1). Instead of using the actual adapting luminance $\bar{Y}$, a filtered value $\bar{Y}_a$ can be used whose value changes according to the adaptation processes in human vision, eventually reaching the actual value if the adapting luminance is stable for some time. The process of adaptation can be modelled using an exponential decay function [Durand and Dorsey 2000]:

$$\bar{Y}_a^{new} = \bar{Y}_a + (\bar{Y} - \bar{Y}_a) \cdot (1 - e^{-\frac{T}{\tau}}) \tag{5}$$

where $T$ is the discrete time step between the display of two frames, and $\tau$ is the time constant describing the speed of the adaptation process. The time constant is different for rods and for cones:

$$\tau_{rods} = 0.4\text{sec} \quad \tau_{cones} = 0.1\text{sec} \tag{6}$$

Therefore, the speed of the adaptation depends on the level of the illumination in the scene. The time required to reach the fully adapted state depends also whether the observer is adapting to light or dark conditions. The values in equation (6) describe the adaptation to light. For practical reasons the adaptation to dark is not simulated because the full process takes up to tens of minutes. Therefore, it is acceptable to perform the adaptation symmetrically, neglecting the case of a longer adaptation to dark conditions.

## 2.3 Scotopic Vision

The human vision operates in three distinct adaptation conditions: scotopic, mesopic, and photopic. The photopic and mesopic vision provide color vision, however in scotopic range, where only rods are active, color discrimination is not possible. The cones start to loose their sensitivity at $3.4\frac{cd}{m^2}$ and become completely insensitive at $0.03\frac{cd}{m^2}$ where the rods are dominant. We model the sensitivity of rods $\sigma$ after [Hunt 1995] with the following function:

$$\sigma(Y) = \frac{0.04}{0.04 + Y} \tag{7}$$

where $Y$ denotes the luminance. The sensitivity value $\sigma = 1$ describes the perception using rods only (monochromatic vision) and $\sigma = 0$ perception using cones only (full color discrimination). The plot of equation (7) is shown in Figure 3.

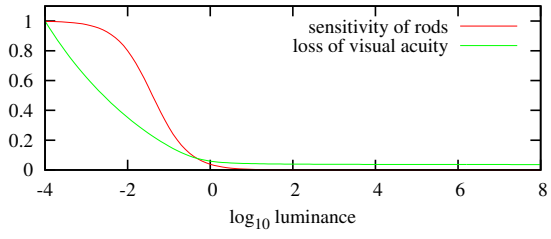


Figure 3: The influence of perceptual effects on vision depending on the luminance level. For details on rods sensitivity and visual acuity refer to Sections 2.3 and 2.4 respectively.

## 2.4 Visual Acuity

Perception of spatial details in the human vision is not perfect and becomes limited with a decreasing illumination level. The performance of visual acuity is defined by the highest resolvable spatial frequency and has been investigated by Shaler in [1937]. Ward et al. [1997] offer the following function fit to the data provided by Shaler:

$$RF(Y) = 17.25 \cdot \arctan(1.4 \log_{10} Y + 0.35) + 25.72 \tag{8}$$

where $Y$ denotes the luminance and $RF$ is the highest resolvable spatial frequency in cycles per degree of the visual angle. The plot of this function is shown in Figure 4.

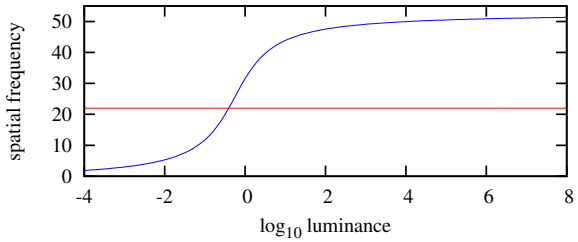


Figure 4: Plot of the highest resolvable spatial frequency for a given luminance level which illustrates the effect of loss of the visual acuity. Spatial frequency is given in cycles per degree of visual angle. The horizontal line marks the maximum displayable spatial frequency on a 15 inch LCD in typical viewing conditions.

To simulate the loss of visual acuity on a display device we need to map the visual degrees to pixels. Such a mapping depends on the size of the display, the resolution, and the viewing distance. For a typical observation of a 15 inch screen from half a meter at $1024 \times 768$ resolution we assume 45 pixels per 1 degree of the visual angle. It is important to note that the highest frequency possible to visualize in such conditions is 22 cycles per visual degree. Therefore, technically we can simulate the loss of visual acuity only for luminance below $0.5\frac{cd}{m^2}$. The irresolvable details can be removed from an image by the convolution with the Gaussian kernel from equation (4) where $s$ is calculated as follows [Ward et al. 1997]:

$$s_{acuity}(Y) = \frac{width}{fov} \cdot \frac{1}{1.86 \cdot RF(Y)} \tag{9}$$

where $width$ denotes width in pixels and $fov$ is the horizontal field of view in visual degrees. For typical observation the $width$ to $fov$ relation equals 45 pixels. We plot the profile of the kernel, according to equation (4), for several luminance values in Figure 5.

In Figure 3 we show the intensiveness of the loss of visual acuity with respect to the luminance level. Apparently the loss of the visual acuity correlates with the increasing sensitivity of rods, and is therefore present in the monochromatic vision.

## 2.5 Veiling Luminance

Due to the scattering of light in the optical system of the eye, sources of relatively strong light cause the decrease of contrast in their vicinity – glare. Such an effect cannot be naturally evoked while perceiving an image on a display due to different viewing conditions and limited maximum luminance of such devices. It is therefore important to account for it while tone mapping.

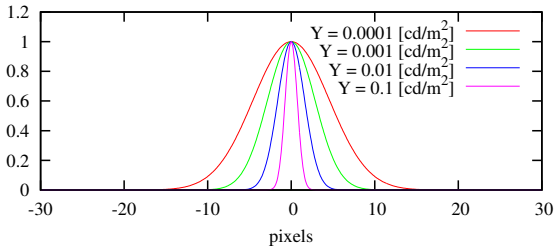


Figure 5: Plot of the profiles of the Gaussian kernels which can be used to simulate the loss of the visual acuity at different luminance levels.

The amount of scattering for a given spatial frequency $\rho$ under a given pupil aperture $d$ is modelled by an ocular transfer function [Deeley et al. 1991]:

$$OTF(\rho,d) = \exp\left(-\frac{\rho}{20.9-2.1\cdot d}^{1.3-0.07\cdot d}\right)$$
$$d(\bar{Y}) = 4.9 - 3\tanh(0.4\log_{10}\bar{Y}+1) \quad (10)$$

In a more practical manner the scattering can be represented in the spatial domain as a point spread function. In Figure 6 we show point spread functions for several adapting luminance levels, which were numerically found by applying the inverse Fourier transform to equation (10).

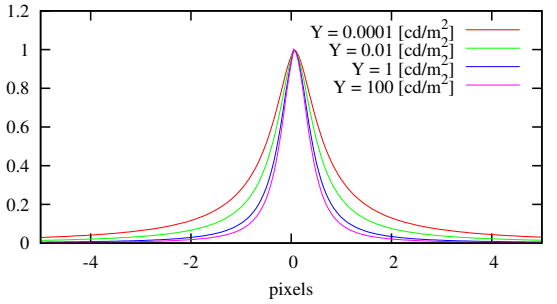


Figure 6: The point spread function illustrating scattering of light in the optical system of the eye for several adapting luminance levels.

Another model of the glare effect was introduced in computer graphics by Spencer et al. [1995]. They describe this phenomenon with four point spread functions linearly combined with three sets of coefficients for different adaptation conditions (scotopic, mesopic and photopic). Since their model is complex, and it is not obvious how to apply it in continuously changing luminance conditions, we decided to employ the model developed by Deeley at al. [1991], which describes the effect with one function, continuously for all adaptation levels, and provides equally good results.

### 2.6 Similarities in Spatial Analysis

Apparently, the local tone mapping, the visual acuity and the veiling luminance are based on the spatial analysis of an image modeled using point spread functions. At the same time, a Gaussian pyramid is required to perform local tone mapping. Interestingly convolution on particular scales corresponds to the convolution required to simulate visual acuity and glare at various luminance levels. This is an important observation which allows to model these effects without additional impact on the performance. The relation of which scale from the tone mapping (Figure 2) corresponds to which convolution for visual acuity and veiling luminance is plotted in Figure 7.

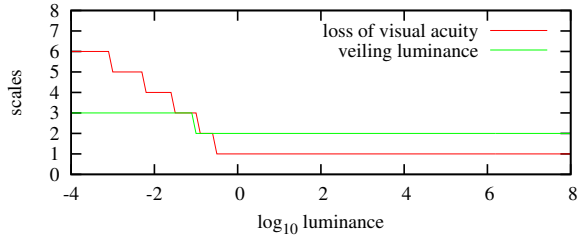


Figure 7: The relation of which scale from the tone mapping (Figure 2) corresponds to which convolution for visual acuity (Figure 5) and veiling luminance (Figure 6).

The Gaussian pyramid constructed for the purpose of tone mapping contains only luminance values. Clearly this is sufficient to simulate the light scattering in the eye, but at the first glance, visual acuity requires to perform the convolution on all three RGB channels. However, it is visible in Figure 3 that the noticeable loss of visual acuity is present only in the scotopic vision where colors are not perceived. Since we simulate the loss of visual acuity combined with scotopic vision, we can simulate it using luminance channel only.

## 3 Algorithm

We present a method that successfully combines tone mapping with the effects mentioned in the previous section, which we implement in graphics hardware for real-time performance. We first show some of our improvements to the tone mapping method in terms of perceived brightness and luminance adaptation process and then explain technical details of our hardware implementation.

### 3.1 Key value

The key value, explained in Section 2.1, determines whether the tone mapped image appears relatively bright or dark, and in the original paper [Reinhard et al. 2002] is left as a user choice. In his follow-up paper, Reinhard [2002] proposes a method of automatic estimation of the key value that is based on the relations between minimum, maximum and average luminance in the scene. Although the results are appealing, we feel this solution does not necessary correspond to the impressions of everyday perception. The critical changes in the absolute luminance values may not always affect the relation between the three values. This may lead to dark night scenes appearing too bright and very light too dark.

The key value, $\alpha$ in equation (1), takes values from $[0:1]$ range where 0.05 is the low key, 0.18 is a typical choice for moderate illumination, and 0.8 is the high key. We propose to calculate the key value based on the absolute luminance. Since the key value has been introduced in photography, there is no scientifically based experimental data which would provide an appropriate relation between the key value and the luminance, so the proper choice is a matter of experience. We therefore empirically specify key values for several illumination conditions and interpolate the rest using the following formula:

$$\alpha(\bar{Y}) = 1.03 - \frac{2}{2+\log_{10}(\bar{Y}+1)} \quad (11)$$

where $\alpha$ is the key value and $\bar{Y}$ is an approximation of the adapting luminance. The plot of this estimation is shown in Figure 8.

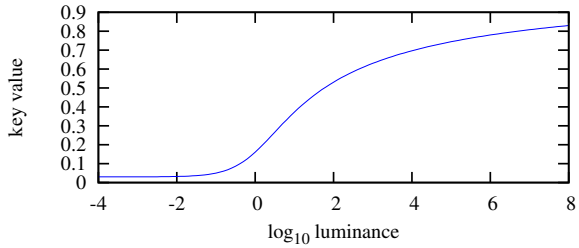


Figure 8: Key value related to adapting luminance in the scene. Refer to Section 2.1 for details.

### 3.2 Temporal Luminance Adaptation

We model the temporal luminance adaptation based on equation (5). However, in our algorithm we do not perform separate computations for rods and cones, what makes it hard to properly estimate the adaptation speed having two time constants $\tau_{rod}$ and $\tau_{cone}$ instead of one. To account for this, and still be able to correctly reproduce the speed of the adaptation, we interpolate the actual value of the time constant based on the sensitivity of rods (equation 7):

$$\tau(\bar{Y}) = \sigma(\bar{Y}) \cdot \tau_{rod} + (1 - \sigma(\bar{Y})) \cdot \tau_{cone} \qquad (12)$$

which we then use to process the adaptation value using equation (5).

### 3.3 Hardware Implementation

In order to perform tone mapping with perceptual effects, we need to compose three maps: a local adaptation map for the tone mapping, a map of visible spatial details to simulate visual acuity, and a map of light scattering in the eye for the glare effect. We will refer to these maps as *perceptual data*. Because different areas of these maps require different spatial processing, they cannot be constructed in one rendering pass. Instead, we render successive scales of the Gaussian pyramid and update the maps by filling in the areas for which the current scale has appropriate spatial processing. In the last step we use these three maps to compose the final tone mapped result.

Technically, we implement our tone mapping method as a standalone module, which can be added at the final rendering stage to any real-time HDR renderer or HDR video player. The only requirement is that the HDR frame is supplied to our module as a floating point texture, what can be efficiently realized using for instance *pixel buffers*. In addition to a texture which holds the HDR frame, our module requires to allocate five textures for processing: two textures for storing adjacent scale levels, two for holding the previous and the current set of perceptual data (due to the updating process), and one intermediate texture for the convolutions. Since the three maps contain only luminance data, we can store them in a single texture in separate color channels.

The process of rendering the perceptual data is illustrated in Figure 9. We start with calculating the luminance from the HDR frame and mapping it to the relative luminance according to equation (1). We calculate the logarithmic average of the luminance $\bar{Y}$ in the frame using the down sampling approach described in [Goodnight et al. 2003], and apply the temporal adaptation process (equation

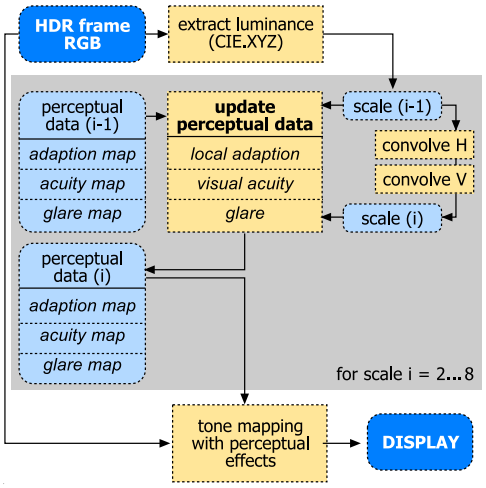


Figure 9: Illustration of the rendering process required to compute the local adaptation, visual acuity and glare. The blue boxes represent the texture data and yellow boxes represent rendering steps. The rendering steps marked by a gray rectangle are repeated for each scale to successively create the coarser scales of the Gaussian pyramid. After the rendering of each scale, the textures representing the perceptual data and the adjacent scales are swapped.

5). The map of relative luminance values constitutes the first scale of the Gaussian pyramid. At each scale of the Gaussian pyramid, we render the successive scale by convolving the previous scale with the appropriate Gaussian (equation 4). We perform the convolution in two rendering passes: one for the horizontal and one for the vertical convolution. To increase the performance we employ down-sampling, where the factor of down sampling is carefully chosen to approximate the kernel. Refer to Figure 10 for our choice of the scaling factors and the corresponding approximations of the Gaussian kernels from Figure 2. Having the current and the previous scales, we update the perceptual data on a per pixel basis in a separate rendering pass. The local adaptation is computed using the measure of the difference between the previous and the current scale as described in [Reinhard et al. 2002]. For the acuity map, we first estimate the proper scale for the luminance of the current pixel. If it falls between the previous and current scales, we interpolate the final value and update the map. In the other case the previous value is copied without change. The mapping from luminance to scale for visual acuity (Figure 7) is cached in a look-up texture to skip redundant computations. We update the glare map in the same manner, with one difference: the appropriate scale for glare depends on the adapting luminance and is uniform for the whole frame so we supply it as a parameter to the fragment program. Before descending to the next scale of the Gaussian pyramid, the texture containing the current scale becomes the previous scale, and the texture with the current set of the perceptual data becomes the previous set.

After descending to the lowest scale of the Gaussian pyramid, the perceptual data texture is complete. In the final rendering step, we tone map the HDR frame and apply the perceptual effects. For this, we use equation (3) from Section 2.1 in a slightly modified form to account for the loss of the visual acuity and the glare:

$$L(x,y) = \frac{Y_{acuity}(x,y) + Y_{glare}(x,y)}{1 + V(x,y)} \qquad (13)$$

where $L$ is the final pixel intensity value, $Y_{acuity}$ is the spatially processed luminance map that represents the visual acuity, $Y_{glare}$ is the amount of additional light scattering in the eye, and $V$ is the local

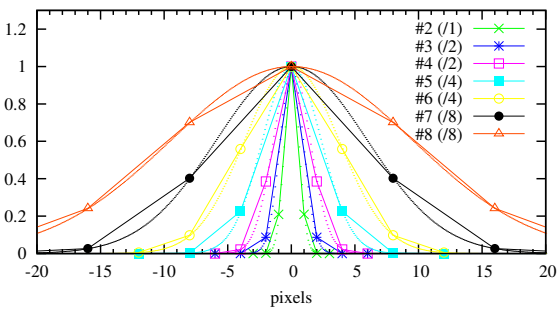


Figure 10: The effective approximation (solid lines) of the Gaussian kernels (dotted lines) from Figure 2 due to the down sampling. The values in parenthesis show the down sampling factor for each scale. For scale #1 we use the original image.

adaptation map. Because the glare map in fact contains the relative luminance from the appropriate scale of the Gaussian pyramid, we estimate the additional amount of scattering in the following way to include only the contribution of the highest luminance:

$$Y_{glare} = Y_{gmap} \cdot \left(1 - \frac{0.9}{0.9 + Y_{gmap}}\right) \tag{14}$$

where $Y_{gmap}$ denotes the glare map from the perceptual data.

We account for the last perceptual effect, the scotopic vision, while applying the final pixel intensity value to the RGB channels in the original HDR frame. Using the following formula, we calculate the tone mapped RGB values as a combination of the color information and the monochromatic intensity proportionally to the scotopic sensitivity:

$$\begin{bmatrix} R_L \\ G_L \\ B_L \end{bmatrix} = \begin{bmatrix} R \\ G \\ B \end{bmatrix} \cdot \frac{L \cdot (1 - \sigma(Y))}{Y} + \begin{bmatrix} 1.05 \\ 0.97 \\ 1.27 \end{bmatrix} \cdot L \cdot \sigma(Y) \tag{15}$$

where $\{R_L, G_L, B_L\}$ denotes the tone mapped intensities, $\{R, G, B\}$ are the original HDR values, $Y$ is the luminance, $L$ is the tone mapped luminance, and $\sigma$ is the scotopic sensitivity from equation (7). The constant coefficients in the monochromatic part account for the blue shift of the subjective hue of colors for the night scenes [Hunt 1995].

An alternative implementation of this tone mapping method, although without perceptual effects, was previously introduced in [Goodnight et al. 2003]. They propose a method to vectorize luminance which allows for efficient convolutions with large support kernels. However, we resigned from their approach to convolutions mainly due to performance reasons – the real-time performance of this algorithm for a $512 \times 512$ frame is reached only when the computations are limited to two scales, which is not sufficient to introduce the perceptual effects. On the other hand, the down-sampling approach provides higher performance with sufficient accuracy of computations.

## 4 Results

We demonstrate our method in combination with an HDR video player. The player renders the compressed HDR video stream [Mantiuk et al. 2004] to a floating point texture, which is then processed as described in Section 3.3. The sample results of our method including the perceptual effects are shown in Figure 11. The top image depicts a computer generated scene in moderate lighting conditions with strong illumination coming from behind the trees in the background. Such a setup would evoke a glare effect in the real-world perception, which is not visible when pure local tone mapping is applied (left part). However, the account for this perceptual phenomena not only contributes to the realism of the rendered image but also increases a subjective impression of the dynamic range (right part). The bottom image shows a car driving scene in day light and at night. Two perceptual phenomena are typical to night illumination: the scotopic vision and the loss of visual acuity. Clearly, in the perceptual tone mapping of the night scene (right part), it is hard to distinguish the colors and the overall brightness is low, what suggests the low illumination of the scene. The inset shows a close-up of the car with increased brightness for the night scene to illustrate the simulated loss of visual acuity.

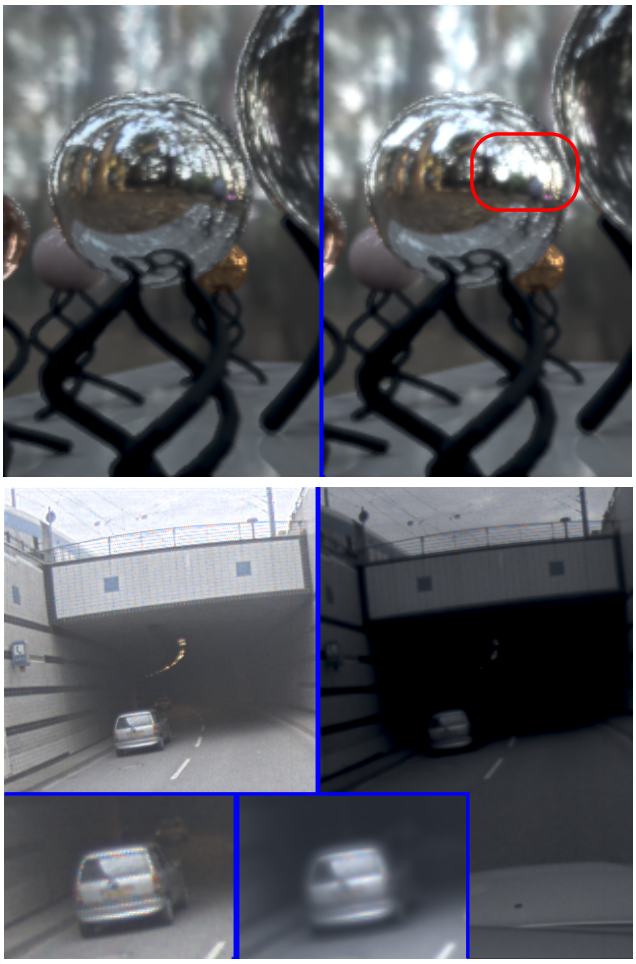

Figure 11: The sample results of our method showing the simulated perceptual effects: glare (top image) and scotopic vision with loss of visual acuity (bottom image). The close-up in the bottom image inset shows the areas around the car in such way that their brightness match to illustrate the loss of visual acuity. The HDR animation "Rendering with Natural Light" (top image) courtesy of Paul Debevec.

We measured the performance of our method on a desktop PC with a Pentium4 2GHz processor and a NVIDIA GeForce 6800GT graphics card. We give the time-slice required for the tone mapping with our method at several frame resolutions in Table 1. In a

| | 320x240 | 640x480 | 1024x768 |
|---|---|---|---|
| 8 scales | 8ms (58Hz) | 25ms (27Hz) | 80ms (10Hz) |
| 6 scales | 7ms (62Hz) | 21ms (30Hz) | 66ms (12Hz) |
| 4 scales | 6ms (62Hz) | 16ms (30Hz) | 51ms (14Hz) |

Table 1: Time-slice required for the display of an HDR frame using our method at several frame resolutions and several sizes of the Gaussian pyramid. In the parenthesis, we give the playback frame rate which we obtained with our method plugged to an HDR video player (note that the resolution also affects the frame decompression speed).

configuration with an HDR video player, where additional time is required for the decompression of the HDR video stream, we were able to obtain the playback at 27Hz. It is important to note that the performance of our solution is scalable. If the time-slice required for our method is too long for a certain application, the number of rendered scales can be limited at the cost of local performance of the tone mapping and the accuracy of the visual acuity processing for very low illumination conditions.

The main bottleneck in the performance is caused by the amount of context switching required for the multi-pass rendering using the *pixel buffers* extension. The currently developed *frame buffer object* extension to OpenGL may provide an improvement, because it eliminates the need for such context switching between the rendering passes thus reducing the delays. Also, current OpenGL drivers do not implement linear interpolation of floating point textures during the up-sampling. Such an interpolation is crucial for the quality of the results and currently is implemented in the fragment program as an additional operation.

In relation to the previous tone mapping techniques which accounted for perceptual effects, our method has the following advantages: we employ a local tone mapping technique, the perceptual effects are applied locally depending on the luminance in a given area, and we make use of the apparent similarities in spatial analysis between the effects to provide a very efficient implementation. The importance of simulating the scotopic vision and the loss of visual acuity was noticed by Ferwerda et al. [1996]. However, they applied these effects only in the context of global tone mapping with uniform intensity over the whole image. This may lead to visible inaccuracies when a dark scene with an area of considerably brighter illumination is processed. In such an area, the loss of color would be unrealistic, and too low spatial frequencies would be removed there. This fact was noticed by Ward et al. [1997] who proposed to apply the perceptual effects locally, still in combination with a global tone mapping method. Yet, in their work each of the effects has been treated separately and involved complex processing what made it inapplicable to real-time processing. In the attempt to provide an interactive tone mapping solution, Durand et al. [2000] reverted to global application of the perceptual effects, what in fact had the same drawbacks as the [Ferwerda et al. 1996] model.

Our real-time implementation leads as well to several constraints. For instance only the tone mapping algorithms, which make use of the Gaussian pyramid, can be implemented in such an efficient combination with the perceptual effects. Therefore, our framework is not appropriate for several different approaches to tone mapping like decomposition into intrinsic images [Durand and Dorsey 2002] or contrast domain algorithms [Fattal et al. 2002]. Also, more complex functions for glare effect simulation may not benefit from our framework, if for instance their point spread functions cannot be approximated with the supplied Gaussian kernels.

# 5 Conclusions

In view of the increasing application of the HDR images and video, we showed how to process such data in order to be able to render them on typical display devices with the substantial dose of realism. We stress out that it is not only necessary to reduce the contrast in such data, but it is also equally important to account for the perceptual effects which would appear in the real-world observation conditions. Owing to the observation that the perceptual effects share similarities in the spatial analysis of the perceived image with the tone mapping algorithm, we were able to efficiently combine them into a stand-alone rendering module and reached the real-time performance. The implementation of our method can be built upon any real-time rendering system which outputs HDR frames or any HDR video player. To demonstrate the importance of the account for perceptual effects, we plugged our method to an HDR video player, what lead to an enhanced realism of the displayed video. We improved the standard methods of simulation of the perceptual effects by applying them locally depending on the illumination in an area, and by providing smooth transition between different adaptation conditions. We envisage that in future the use of such a module will be standard in every real-time HDR renderer and HDR video player.

# Acknowledgements

This work was partially supported by the European Union within the scope of project IST-2001-34744, “Realtime Visualization of Complex Reflectance Behaviour in Virtual Prototyping” (RealReflect). Further, we would like to thank Rafał Mantiuk and Philipp Jenke for their valuable comments on the paper.

# References

DEBEVEC, P., AND MALIK, J. 1997. Recovering high dynamic range radiance maps from photographs. In *Proceedings of SIGGRAPH 97*, Computer Graphics Proceedings, Annual Conference Series, 369–378.

DEELEY, R., DRASDO, N., AND CHARMAN, W. N. 1991. A simple parametric model of the human ocular modulation transfer function. *Ophthalmology and Physiological Optics 11*, 91–93.

DEVLIN, K., CHALMERS, A., WILKIE, A., AND PURGATHOFER, W. 2002. Tone Reproduction and Physically Based Spectral Rendering. In *Eurographics 2002: State of the Art Reports*, Eurographics, 101–123.

DRAGO, F., MYSZKOWSKI, K., ANNEN, T., AND CHIBA, N. 2003. Adaptive logarithmic mapping for displaying high contrast scenes. *Computer Graphics Forum, proceedings of Eurographics 2003 22*, 3, 419–426.

DURAND, F., AND DORSEY, J. 2000. Interactive tone mapping. In *Rendering Techniques 2000: 11th Eurographics Workshop on Rendering*, 219–230.

DURAND, F., AND DORSEY, J. 2002. Fast bilateral filtering for the display of high-dynamic-range images. *ACM Transactions on Graphics 21*, 3 (July), 257–266.

FATTAL, R., LISCHINSKI, D., AND WERMAN, M. 2002. Gradient domain high dynamic range compression. *ACM Transactions on Graphics 21*, 3 (July), 249–256.

FERWERDA, J., PATTANAIK, S., SHIRLEY, P., AND GREENBERG, D. 1996. A model of visual adaptation for realistic image synthesis. In *Proceedings of SIGGRAPH 96*, Computer Graphics Proceedings, Annual Conference Series, 249–258.

GOODNIGHT, N., WANG, R., WOOLLEY, C., AND HUMPHREYS, G. 2003. Interactive time-dependent tone mapping using programmable graphics hardware. In *Proceedings of the 14th Eurographics workshop on Rendering*, Eurographics Association, 26–37.

HUNT, R. 1995. *The Reproduction of Colour in Photography, Printing and Television: 5th Edition*. Fountain Press.

KANG, S. B., UYTTENDAELE, M., WINDER, S., AND SZELISKI, R. 2003. High dynamic range video. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2003) 22(3)*, 319–325.

MANTIUK, R., KRAWCZYK, G., MYSZKOWSKI, K., AND SEIDEL, H.-P. 2004. Perception-motivated high dynamic range video encoding. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2004) 23*, 3, 733–741.

NAYAR, S., AND BRANZOI, V. 2003. Adaptive dynamic range imaging: Optical control of pixel exposures over space and time. In *Proc. of IEEE International Conference on Computer Vision (ICCV 2003)*, 1168–1175.

NG, R., RAMAMOORTHI, R., AND HANRAHAN, P. 2004. Triple product wavelet integrals for all-frequency relighting. *ACM Transactions on Graphics 23*, 3, 477–487.

REINHARD, E., STARK, M., SHIRLEY, P., AND FERWERDA, J. 2002. Photographic tone reproduction for digital images. *ACM Transactions on Graphics 21*, 3, 267–276.

REINHARD, E. 2002. Parameter estimation for photographic tone reproduction. *Journal of Graphics Tools: JGT 7*, 1, 45–52.

SHALER. 1937. The relation between visual acuity and illimination. *Journal of General Psychology 21*, 165–188.

SLOAN, P.-P., KAUTZ, J., AND SNYDER, J. 2002. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Transactions on Graphics 21*, 3, 527–536.

SPENCER, G., SHIRLEY, P., ZIMMERMAN, K., AND GREENBERG, D. 1995. Physically-based glare effects for digital images. In *Proceedings of ACM SIGGRAPH 95*, 325–334.

WARD, G., RUSHMEIER, H., AND PIATKO, C. 1997. A visibility matching tone reproduction operator for high dynamic range scenes. *IEEE Transactions on Visualization and Computer Graphics 3*, 4, 291–306.