Adaptive Temporal Tone Mapping

Shaun David Ramsey, J. Thomas Johnson III, Charles Hansen *
University of Utah, Salt Lake City, Utah, USA

Abstract

Monitor intensity ranges are much lower than the range of intensities in the real world or even in high quality renderings. Rendering in high dynamic range (HDR) is becoming more common in computer graphics. HDR video cameras are also available. The process of compressing a single frame of HDR data (real or synthetic) into a range displayable by monitors is called tone mapping. Videos (a real or synthetic sequence of images) require this technique as well. Tone mapping video introduces a temporal constraint to maintain consistent intensities between frames. We present a novel method, called adaptive temporal tone mapping, which provides smooth intensity transitions in tone mapped video, while allowing for discontinuous dynamic lighting changes (such as turning on a light or exiting a tunnel).

KEY WORDS
Temporal, Time, Tone Reproduction, Tone Mapping

1 Introduction

High dynamic range (HDR) images and video (a real or synthetic sequence of images) are becoming more common and important in computer graphics. The dynamic range of most display devices (such as monitors, printers, and projectors) is much lower than the dynamic range found in real-world scenes and in high quality renderings. The ability to display these HDR images and sequences on low dynamic range devices is desirable. The process of mapping high dynamic range images to be displayed on low dynamic range devices is known as tone mapping.

While tone mapping has been a research focus in recent years [1, 2, 3], work in this field is not new [4, 5]. HDR images have shown to be useful in a variety of applications [6, 7, 8, 9, 10, 11]. Tone mapping has been accomplished through explicit models, human visual system models, and luminance mapping.

In computer graphics, two temporal tone mapping methods have been developed. In work presented by Kang et al. [12], video with alternating exposure time is converted into tone mapped HDR video. In their method, two seconds of video (at fifteen frames per second) is used to compute a log average luminance. This method assumes a slowly changing scene intensity and is dependent upon the method presented by Reinhard et al. [1]. In Pattanaik et al. [13], temporal tone mapping is based on the human visual system (HVS). When an abrupt lighting change occurs in an image sequence, the tone mapping operator requires multiple frames to adjust to the overall luminance change. This temporal constraint is modeled after the hysteresis of the HVS. Our model quickly adapts to luminance changes to allow for the greatest perceptual clarity in each frame while maintaining temporal coherence.

We also have chosen to extend the method presented in [1] due to its simplicity and speed. However, our temporal method extends to any tone mapping operator which uses a luminance mapping operator. Given a sequence of HDR images, the luminance between tone mapped frames without temporal coherence may vary enough to cause flickering. In an HDR video, adaptivity of the key map [14, 15, 16] may be required to allow for the highest quality low dynamic range video. Our method allows for quick luminance changes where expected (e.g., turning on a light) while maintaining smooth transitions.

2 Background

The method of Reinhard et al. [1] uses the luminance of pixel values to compute a global log average luminance value. Using RGB values, the pixel luminance $L_p(x,y)$ is computed by:

$$L_p(x,y) = 0.27\,R + 0.67\,G + 0.06\,B \tag{1}$$

This method was developed for tone mapping strictly single frames through the use of log average luminance L_f. Each frame is considered to have N pixels. This frame luminance is computed by:

$$L_f = exp\left(\frac{1}{N}\sum_{x,y} log(\delta + L_p(x,y))\right) \tag{2}$$

$L(x,y)$ defines the scaled pixel luminance. We obtain $L(x,y)$ by a user-defined key value a and the frame luminance.

$$L(x,y) = \frac{a}{L_f} L_p(x,y) \tag{3}$$

The key value is generally set in the range of 0 to 1 although in very dark scenes, higher values may bring desirable contrast to a scene. To perform tone mapping, a value L_{white} is required. L_{white} defines the maximum luminance that is set to white in the scene. By setting L_{white} lower than infinity, burn out may occur, but contrast in the scene is often

*[ramsey@cs | hansen@cs | tjohnson@sci].utah.edu

improved. The tone mapping operator is then:

$$L_t(x,y) = \frac{L(x,y)\left(1+\frac{L(x,y)}{L^2_{white}}\right)}{1+L(x,y)} \qquad (4)$$

This is the tone mapping operator (without dodging and burning) presented in [1]. To acquire final RGB values, one can simply multiply the original pixel high dynamic range [HDR] values by L_t of the same pixel.

3 Algorithm

Temporal sequences of video requires several changes to the tone mapping algorithm. The first major change creates a luminance L_i dependent on a number of frames. L_i replaces L_f in Equation 3. The approach presented in [12] forces L_f to depend on a static number of frames n. $\overline{N}$ now represents the number of pixels in n frames. Their method is implemented by:

$$L_i = exp\left(\frac{1}{N}\sum_{x,y,i} log(\delta + L(x,y))\right) \qquad (5)$$

Our method allows the number of frames n to adapt according to the scene luminance. This adaptation allows swift changes in luminance when necessary (e.g., a light is switched on). To ensure that all transitions are smooth, we force a small number of frames (5) to be averaged. We limit the maximum number of frames (60) averaged to prevent over-computation and other temporal artifacts. We store the average log luminance per frame (L_{f_i}), as computed in Equation 2. Our technique produces a new value L_a to replace the value L_i as presented in Algorithm 3.1.

Algorithm 3.1, describes in detail how to obtain L_a. Simply, we average the luminance of the last few frames as long as the frame's luminance is within a tolerance limit of the current frame luminance. When we find a frame outside of our temporal window or is outside of our luminance window, we do not average this frame or any further frames.

The value L_a is now used in the place of L_f in equation 3. For scenes with changing luminance, changes in the key value a may also be necessary. Setting the key value in still frames or photographs requires some knowledge of the desired brightness in the final images. In a video, some knowledge of the change in key value is also required. We acquire frame specific key values with the following formula:

$$a = -\alpha\ \arctan(\beta(L_a - \gamma)) + \alpha\frac{\pi}{2} \qquad (6)$$

The constants in Equation 6 are designed according to taste and preference. The effects of changing these constants can be seen in Figure 1.

To determine the key value for a frame in an image sequence, we use a low-pass filter algorithm. First, we store the key values from Equation 6 as a_i. Then, using the number of frames ($NumFrames$) from Algorithm 3.1, we perform the low-pass filter to acquire the new key value a_n as shown in Algorithm 3.2.

Algorithm 3.1: $L_a(L_{f_i})$

$range = 0.1 * L_{f_i}$
$minL = L_{f_i} - range$
$maxL = L_{f_i} + range$
$j = i$
while $(j > 0$ **and** $i - j < 60)$
  (
    if $(L_{f_j} > minL$ **and** $L_{f_j} < maxL)$
      $j = j - 1$
    else if $(i - j) < 5$
      $j = j - 1$
    else
      break
)
$NumFrames = i - j + 1$
$total_{log} = 0$
for $(k = i;\ k \geq j;\ k = k - 1)$
  $total_{log} = total_{log} + L_{f_k}$
$L_a = exp(total_{log}/NumFrames)$

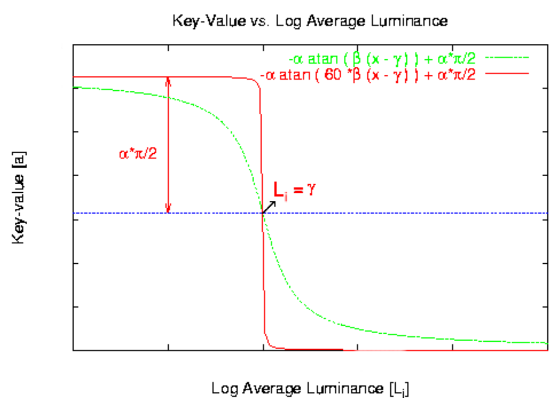

Figure 1. The results of different constants in Equation 6. As α increases, the amplitude of the curve increases (changing affective key-values). As β increases, the slope of the curve is affected. Changing β allows for slower or faster key-value changes based on log average luminance.

Algorithm 3.2: KEY-VALUE(a_i)

$a_n = 0;$
for $(j = i;\ i - j < NumFrames;\ j = j - 1)$
  $a_n = a_n + a_j$
$a_n = a_n/NumFrames$

We combine algorithms 3.1 and 3.2 into one function with optimizations for greater efficiency. We now complete the tone mapping operation as before.

$$
\begin{aligned}
L_p(x,y) &= 0.27\,R + 0.67\,G + 0.06\,B \\
L'(x,y) &= \frac{a_n}{L_a} L_p(x,y) \\
L'_t(x,y) &= \frac{L'(x,y)\left(1 + \frac{L'(x,y)}{L^2_{white}}\right)}{1 + L'(x,y)}
\end{aligned}
$$

4 Results

Our method runs at approximately 70 frames per second on a 2.00 GHz Pentium 4 processor, without file I/O. Our adaptive method uses a varying number of frames to compute the low-pass filter based on key values and log average luminances. This is shown in Figure 2. The key value changes according to Algorithm 3.2. The frame-specific and actual key values used in our bright living room scene are shown in Figure 3.

In a scene with discontinuous luminance changes our method is superior to the method of Kang et al., since their method does not allow the key value to adapt. A fixed key value for an image sequence may cause undesirable results. A scene in which a light is turned on demonstrates this. With a high key value, a dark room appears in high contrast before the light is turned on. This same room appears burned out after the light is turned on. With a low key value, the dark room will appear completely black, whereas the lit room will appear in desirable contrast. Also, when the luminance in a scene changes, our method adapts to the luminance with the greater perceptual clarity more quickly while maintaining temporal coherence. This is true because our adaptive averaging requires a minimum of 5 frames instead of the constant 30 frames of Kang et al.

In Figures 4 and 5, we show the difference between an HDR image which is clamped, scaled, and tone mapped with our method. In Figure 6, we show various techniques applied to a living room scene. In the first row, the temporal sequence is shown clamped and much of the visible area is burned out. The second row shows the sequence when using a constant key value of 0.05. The third row shows one possible result while using $\alpha = 1000, \beta = 550, \gamma = 4$ for the values in Equation 6. The fourth row is another rendering using $\alpha = 999.95, \beta = 400, \gamma = 3$ for the constants in Equation 6. The preferred quality in the last two rows vary by artist, but both show an improvement.

5 Conclusion

In summary we have developed a temporal tone mapping method (without artifacts) which allows for discontinuous luminance changes while enabling the user to visualize the most interesting portions of a scene. Our method is quick and efficient, allowing for interactive tone mapping rates

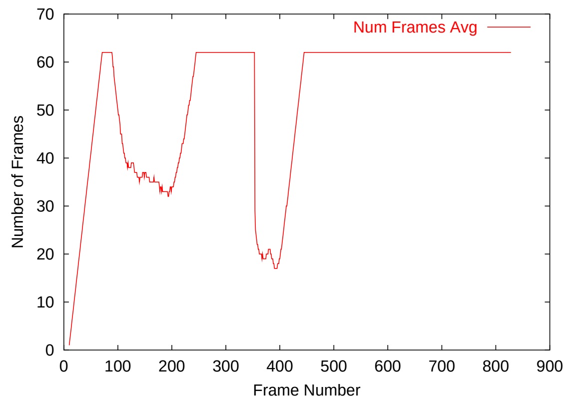

Figure 2. The number of frames used to average key values and log average luminances for the living room scene.

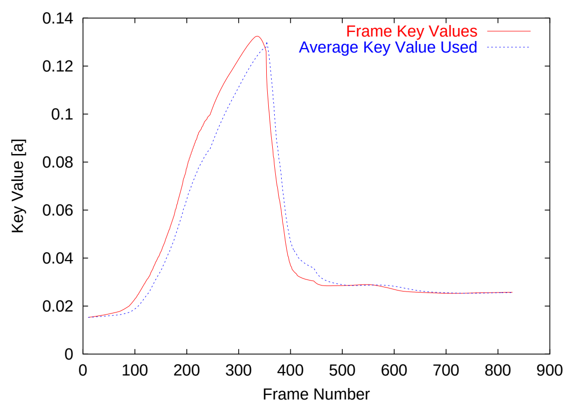

Figure 3. Key values for the living room scene.

of video that are in main memory (70 frames per second) and near interactive tone mapping rates of video streaming from disk (3.5 frames per second).

Our method is faster, more flexible and more robust than previous temporal tone mapping methods. Kang et al. [12] reported two seconds per frame for their method on a 2.00 GHz Pentium 4. Their method is also unable to deal with luminance discontinuities.

As future work, we plan to develop a more sophisticated low-pass filter to obtain better control over the changing parameters in the scene. The key values and *artistic* parameters of tone mapping should also be automated as much as possible for novice users. Better visualization tools for detected problem areas could allow users to tailor the desired results more efficiently.

6 Acknowledgments

We would like to thank the entire Utah Real-Time Ray-Tracing development crew for creating such beautiful scenes for our renderings. Special thanks to R. Keith Morley for the use of galileo and the sponza data set.

References

[1] Erik Reinhard, Michael Stark, Peter Shirley, and James Ferwerda. Photographic tone reproduction for digital images. In *ACM Transactions on Graphics*, volume 21, pages 267–276, July 2002.

[2] Frédo Durand and Julie Dorsey. Fast bilateral filtering for the display of high-dynamic-range images. In *ACM Transactions on Graphics*, volume 21, pages 257–266, July 2002.

[3] Raanan Fattal, Dani Lischinski, and Michael Werman. Gradient domain high dynamic range compression. In *ACM Transactions on Graphics*, volume 21, pages 249–256, July 2002.

[4] Jack Tumblin and Holly E. Rushmeier. Tone reproduction for realistic images. *IEEE Computer Graphics & Applications*, 13(6):42–48, November 1993.

[5] James A. Ferwerda, Sumant Pattanaik, Peter S. Shirley, and Donald P. Greenberg. A model of visual adaptation for realistic image synthesis. In *Proceedings of SIGGRAPH 96*, Computer Graphics Proceedings, Annual Conference Series, pages 249–258, August 1996.

[6] Paul E. Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *Proceedings of SIGGRAPH 97*, Computer Graphics Proceedings, Annual Conference Series, pages 369–378, August 1997.

[7] Paul Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings of SIGGRAPH 98*, Computer Graphics Proceedings, Annual Conference Series, pages 189–198, July 1998.

[8] Jack Tumblin, Jessica K. Hodgins, and Brian K. Guenter. Two methods for display of high contrast images. *ACM Transactions on Graphics*, 18(1):56–94, January 1999.

[9] Fredo Durand and Julie Dorsey. Interactive tone mapping. In *Rendering Techniques 2000: 11th Eurographics Workshop on Rendering*, pages 219–230, June 2000.

[10] A. Scheel, M. Stamminger, and Hans-Peter Seidel. Tone reproduction for interactive walkthroughs. *Computer Graphics Forum*, 19(3):301–312, August 2000.

[11] Jonathan Cohen, Chris Tchou, Tim Hawkins, and Paul Debevec. Real-Time high dynamic range texture mapping. In *Rendering techniques*, pages 313–320, 2001.

[12] Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High dynamic range video. In *ACM Transactions on Graphics*, volume 22, pages 319–325, July 2003.

[13] Sumanta N. Pattanaik, Jack Tumblin, Hector Yee, and Donald P. Greenberg. Time-dependent visual adaptation for fast realistic image display. In *Proceedings of ACM SIGGRAPH 2000*, Computer Graphics Proceedings, Annual Conference Series, pages 47–54. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.

[14] Ansel Adams. The camera. *The Ansel Adams Photography series*, 1980.

[15] Ansel Adams. The negative. *The Ansel Adams Photography series*, 1981.

[16] Ansel Adams. The print. *The Ansel Adams Photography series*, 1983.

Figure 4. The sponza scene clamped, scaled and tone mapped ($a = 0.7$) from left to right.

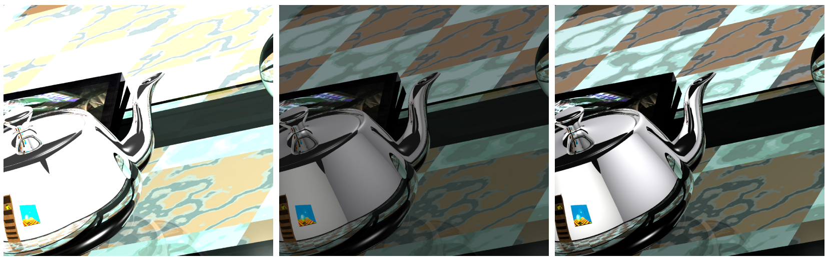

Figure 5. The teapot scene clamped, scaled and tone mapped ($a = 0.5$) from left to right.

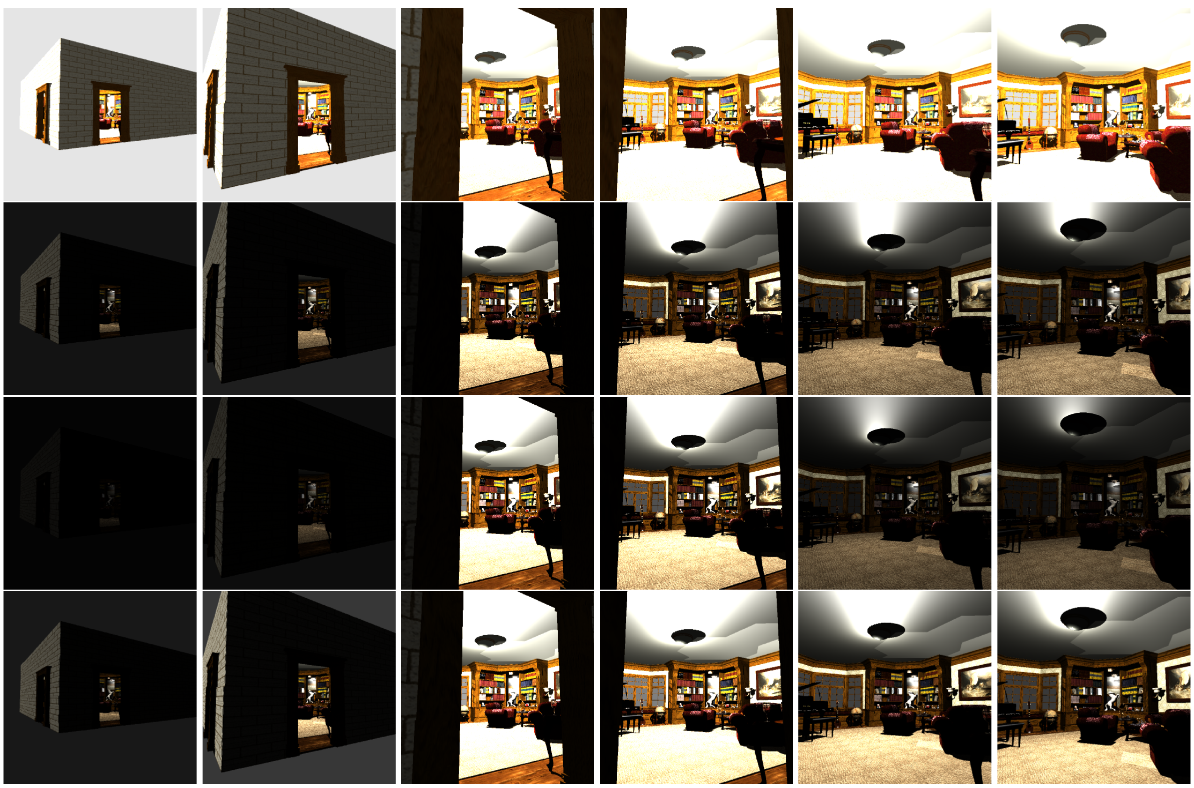

Figure 6. The living room scene clamped and tone mapped with various strategies. Discoloration along the ceiling and walls of this scene are part of the scene itself and not an artifact of our algorithm.