CG-TMO: A LOCAL TONE MAPPING FOR COMPUTER GRAPHICS GENERATED CONTENT

Francesco Banterle[1]*, Alessandro Artusi*[2]*, Paolo Francesco Banterle*[3]*, Roberto Scopigno*[1]

[1] Visual Computing Laboratory, ISTI-CNR, Pisa, Italy
[2] Departament d'Informatica I Matemática Aplicada Universitat de Girona, Spain
3 Università di Siena, Italy

ABSTRACT

Physically based renderers produce high quality images with high dynamic range (HDR) values. Therefore, these images need to be tone mapped in order to be displayed on low dynamic range (LDR) displays. A typical approach is to blindly apply tone mapping operators without taking advantage of the extra information that comes for free from the modeling process for creating a 3D scene. In this paper, we propose a novel pipeline for tone mapping high dynamic range (HDR) images which are generated using physically based renderers. Our work exploits information of a 3D scene, such as geometry, materials, luminaries, etc. This allows to limit the assumptions that are typically made during the tone mapping step. As consequence of this, we will show improvements in term of quality while keeping the entire process straightforward.

***Index Terms*—** High Dynamic Range Imaging, Tone Mapping, Tone Mapping Operator, Computer Graphics, Physically Based Rendering, Virtual Environments, Interactive and Real-Time applications.

1. INTRODUCTION

HDR content, typically encoded as 32-bit floating point format per color channel, captures more information than 8-bit or low dynamic range (LDR) images and videos. This information can be used to reproduce specific visual effects such as adding veiling glare, accurate motion blur, etc. Furthermore, HDR content conveys to better details and contrast reproduction when displayed on traditional LDR displays. This last step is typically called tone mapping (TM). An intensive research has been carried out on this field which has provided several solutions to better preserve characteristics of the input HDR content exploiting visual perception and image processing aspects. In the rendering field [1, 2], i.e. generation of images from a 3D description of a scene (geometry, lighting, materials, etc.), tone mapping has been mostly applied at the end of the pipeline as a independent step. The exploitation of data, which is available in the 3D scene, helps to compute more accurately and efficiently statistics required by tone mapping operators (TMOs). This is because we can extract information from the entire scene instead of a single image.

In this work, we present modifications to the rendering pipeline and we propose a real-time tone mapping operator, CG-TMO, that can exploit information of a 3D scene. Our operator works in real-time, so it can be applied to images generated by physically based renderers, without time constraints, and for real-time renderers for virtual environments (VEs), computer simulations, and video-games which have typically real-time constraints. Our contributions are:

- a real-time local version of the Drago et al.'s operator [3] based on the bilateral filter [4];
- a cross bilateral filter for tone mapping based on information of the 3D scene, e.g. geometry and materials;
- a technique for computing statistics for the proposed operator using light sources in the 3D scene;
- an evaluation of the proposed technique compared to previous work.

2. RELATED WORK

A large number of TMOs have been proposed in the last 20 years and a detailed description can be found in reference books [5, 6]. Approaches that may relate to our work can be classified as real-time TMOs, and TMOs for virtual environments (VEs).

Real-Time Tone Mapping Operators. Hardware implementation of TMOs, tightly coupled with the current graphics hardware or FPGAs, have been presented in the past [7–9]. Recently, novel approaches, that exploit the capability of recent graphics hardware, have started to appear. For example, efficient uses of bilateral filtering have been presented [10, 11]. Other solutions leverage on aspects of the human visual system to limit the computational time and memory for computing the local luminance adaptation (local methods for high frequencies regions, and global ones for low frequencies regions) in a more efficient way [12].

Virtual Environments. Little work has been carried out to exploit information embedded in VEs for improving quality and/or increasing computational speed during tone mapping. For example, Durand and Dorsey [13] described a novel tone mapping pipeline taking into account the 3D scene and in particular light sources. However, this use is limited to only post-processing effects such as accurate lens-flares and veiling glare simulation. In the field of virtual reality environments tracking information (i.e. user's view) has been used to optimize the computations of luminance adaptation and to reduce computationally costly image operations which are restricted to the central field of view [14]. Josselin et al. [15] analyzed the dynamic behavior of TMOs designing two psychophysical experiments to assess the relevance of different TMOs to explore an outdoor virtual world at night. The major finding of these experiments was the need of modeling the temporal visual adaptation for night-time applications. The re-use of VE information has been also exploited by Josselin et al. [16] improving the realism of the rendered scene at lower computational costs when compared with the traditional rendering pipeline where the TMO is at its end and the information of the VE is not used.

3. A GENERAL TONE MAPPING PIPELINE

The main idea of our work is to exploit 3D scene information (e.g. light sources) and 3D information generated by the renderer for the current frame such as XYZ positions, normals, and albedo in order to improve the local filtering for the tone mapping. An overview of the proposed modified pipeline for rendering and tone mapping is shown in Figure 1.

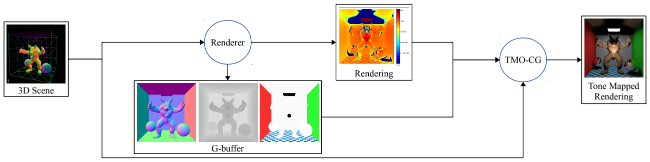


Fig. 1. The proposed modified rendering and tone mapping pipeline: when a renderer generates an HDR image (rendering) from a 3D scene, it saves the G-buffer as well (this is composed by geometrical information of the rendering view such as the positions (depth), normals, albedo, etc.). At this point, the rendering is tone mapped using CG-TMO which takes as input extra information such as the G-buffer and the 3D scene.

3.1. CG-TMO

Our novel TMO, CG-TMO, is an operator based on a local version of the Drago et al.'s operator [3], see Section 3.2, based on a non-linear filter which takes as input information generated by the renderer, see Section 3.3, with statistics which are compute form the 3D scene, see Section 3.4

3.2. Local Drago et al.'s Operator

Drago et al. [3] presented a simple and effective TMO which produces convincing appearance results. This operator is basically an adaptive logarithm mapping based on global image statistics. Given these good properties of the operator, we decided to extend it in order to preserve local details. We recall the operator definition:

$$L_d(\mathbf{x}) = \frac{\frac{L_{d,Max}}{100}\log(L_w(\mathbf{x})+1)}{\log_{10}(L_{w,Max}+1)\log_a(L_w(\mathbf{x}),L_{w,Max})} \quad (1)$$

$$\log_a(L_w(\mathbf{x}),L_{w,Max}) = \log 2 + 8\left(\frac{L_w(\mathbf{x})}{L_{w,Max}}\right)^{\log_{0.5} b}$$

where $L_w(\mathbf{x})$ is the HDR luminance at pixel $\mathbf{x}$, $L_{w,Max}$ is the maximum HDR luminance value, $L_d(\mathbf{x})$ is the tone mapped luminance, $L_{d,Max}$ is the maximum luminance of the output LDR display in cd/m^2, $b \in [0.7, 0.9]$ is a parameter for shaping the TM curve. Note that HDR luminance values are scaled by $L_{w,a}$ before applying Equation 2, which is the world luminance adaptation. This is approximated as the harmonic mean of the image's luminance [3].

In our work, we decided to create a local version of this operator by applying an approach used by Durand and Dorsey [17]. In their work, the Tumblin et al.'s operator [18] was extended into a local one by leveraging on the bilateral filter [4]. This approach can be formally defined as:

$$\begin{aligned} L_{w,\text{base}} &= B(L_w, \sigma_s, \sigma_r) \\ L_d &= TC(L_{w,\text{base}}) \times \left(\frac{L_w}{L_{w,\text{base}}}\right) \end{aligned} \quad (2)$$

B is a bilateral filter, σ_s is the spatial standard deviation of B, σ_r is the range standard deviation of B, and TC is a global tone mapping curve; in our case TC is Equation 2. This approach may suffer of artifacts such as blur, due to the fact that linear ramp signals are smoothed and not preserved. To avoid

Fig. 2. An example of our local Drago et al.'s operator [3]: on the left side an HDR rendering tone mapped with the classic global operator. On the right side the same rendering tone mapped using the approach in Equation 2. Note that fine details (e.g. architectural details such as bricks, bas-reliefs, etc.) are better preserved than the image on the left side.

this excessive smoothing, more sophisticated non-linear filters have been designed (e.g. the WLS filter [19]) but they price of having such filters are high computational costs or complex implementations to integrate.

3.3. Geometry-Buffer Cross Bilateral Filtering

In our approach, we decided to extend the bilateral filter to embed information generated by the renderer in order to guide the filter and to obtain similar results to the ones reached with more advanced non-linear filters.

An option for achieving such goal is to use the cross-bilateral filter [20, 21]. This is an extension of the bilateral filter where guiding edges are not from the input image, I, but from an edge image, E, which guides the filtering. The filter is defined as:

$$C(I,E)(\mathbf{x}) = \frac{\sum_{\mathbf{y}\in\Omega} I(\mathbf{y})g(\|\mathbf{x}-\mathbf{y}\|)f(\|E(\mathbf{x})-E(\mathbf{y})\|)}{K(\mathbf{x},\mathbf{y})} \quad (3)$$

where $\mathbf{x}$ and $\mathbf{y}$ are pixel coordinates, g and f are attenuation functions (typically Gaussian functions), K is the normalization term, and Ω is a neighborhood centered around $\mathbf{x}$. The cross bilateral filter can be extended to have an arbitrary number of edge images E. The general form is defined as:

$$C^n(I,E)(\mathbf{x}) = \\ = \frac{\sum_{\mathbf{y}\in\Omega} I(\mathbf{y})g(\|\mathbf{x}-\mathbf{y}\|)\prod_{i=1}^{n} f_i(\|E_i(\mathbf{x})-E_i(\mathbf{y})\|)}{K(\mathbf{x},\mathbf{y})} \quad (4)$$

In our case, I is the HDR rendered image generated by the render while the edge images $E_1 \ldots E_n$ are buffers that contain 3D information generated by the renderer. In particular, 3D information consists of: 3D world positions, P-buffer, normals, N-buffer, and albedos, A-buffer. These three components are typically referred in literature as the Geometry-buffer or G-buffer [2]. Therefore, our filter has $n = 3$, $E_1 = P$, $E_2 = N$ and $E_3 = A$. In our tests, g and f^i are Gaussian functions and their σ_i values were found through an experimental design: $\sigma_g = 8$ creates a medium-large neighborhood for capturing fine details, σ_1 is set to the pixel footprint size in world space [1] multiplied by σ_g, $\sigma_2 = 0.25$ discriminates well between two normals, $\sigma_3 = 0.25$ is a reasonable value for albedo colors which are typically flat. In Figure 3, we show how Equation 4 produces better results compared to the classic bilateral approach.

In order to speed up computations and to achieve real-time performances, we employed an efficient but approximated strategy for evaluating cross bilateral filters [22]. This strategy evaluates the cross bilateral filter for a subset pixels in the evaluation neighborhood using Poisson-disk distributed points [23].

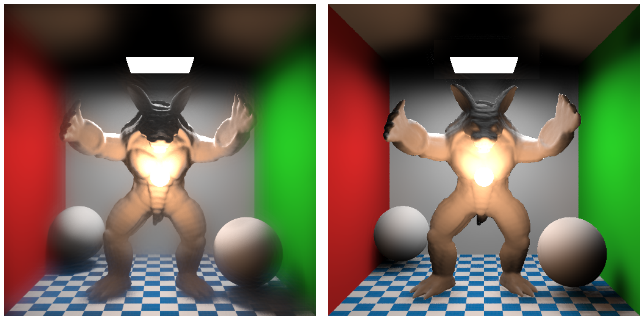

(a) (b)

Fig. 3. An example highlighting the differences between classic bilateral filtering and cross-bilateral filtering based on the g-buffer: (a) The classic bilateral filtering applied to a rendered image. Note that bleeding between spheres and walls and the one between the armadillo and the floor are quite evident. (b) The same image was filtered using the proposed cross-bilateral filtering using the g-buffer; note that no bleeding occurs.

3.4. Computing Statistics from the 3D Scene

$L_{w,max}$ and $L_{w,a}$, Equation 2, are typically computed from the HDR image obtaining approximated values. This can obviously affects the output quality performance. By using 3D scene description, we can accurately compute closer approximations or even exact values.

$L_{w,max}$ can be exactly calculated by computing the maximum value of 3D light sources luminance values from the formal description of the 3D scene. Moreover, we need to take into account only lighting information contributing to the current view otherwise we may still have an approximated computation of $L_{w,max}$. To achieve this, we have to compute a subset of light sources which are visible or indirectly visible from the viewing camera. This can be solved in a straightforward way by storing in a separate buffer, M, at each pixel the index of the light source with maximum luminance power that contributed to light the same pixel coordinates in the final rendering. Then, $L_{w,max}$ is calculated as the maximum of light sources luminance power in M. Although, this seems more complex than testing visibility using ray-tracing operations, we want to stress out that testing visibility may be an issue in the case of indirect lighting. For example, a scene with indirect lighting, two rooms with an open door and light sources only in the room which is not visible in the viewing camera, would be difficult to test or it would take an extreme amount of time to compute the correct value.

$L_{w,a}$ is typically computed as harmonic mean of the image luminance without taking into account the surrounding environment. To improve this, we opted to render at low resolution (we found that 64×32 was enough in our experiments) the 3D scene using a spherical camera centered in the position of the rendering camera. This allows to include lighting information from all environment in a similar manner as proposed by Gruson et al. [24].

4. RESULTS

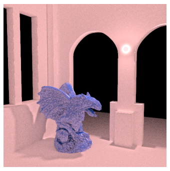
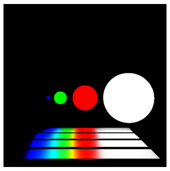
Scene	CG-TMO	WLS-TMO
	0.913	0.909
	0.906	0.905
	0.921	0.946
	0.971	0.976
	0.773	0.773

Table 1. Comparing our CG-TMO againt a local version of the Drago et al.'s operator with WLS filtering (WLS-TMO) using the TMQI [25]; the closer to 1.0 the better.

Quality. We compared our CG-TMO against a local version of Drago et al.'s operator using WLS filtering [19], we refer to this as WLS-TMO. The main goal of this comparison is to show that our proposed algorithm produces similar results to WLS which is more complex to implement and computationally slower due to the fact that a sparse and large linear system has to be solved (e.g. the linear system is $262,144 \times 262,144$ for a 512×512 image). In our experiment, we generated 5 3D scene and rendered them using a monte-carlo renderer. Then, we tone mapped the rendered scenes with the two operators. The WLS filter parameters, α and λ, were respectively set to 1.2 and 0.125. We compared each image against the original HDR rendered image using the tone mapped image quality index [25]. This index is an extension of the well known SSIM [26] to the HDR domain, for comparing tone mapped and HDR images. From the results which are shown in Figure 1, we can point out that both our CG-TMO and WLS-TMO provide similar results as we were expecting.

Timing. We evaluated time performances of CG-TMO by varying the size of images to be tone mapped. In order to obtain real-time performances, we wrote a GPU implementation of CG-TMO in C++ and OpenGL. In our experiments, the testing machine was an Intel Core i7 2.8Ghz equipped with 4GB of main memory and a NVIDIA GeForce GTX 480 with 2.5GB of memory with Windows 7 Professional at 64-bit as OS. The results of these experiments are shown in Figure 4. From this graph, we can note that our algorithm works in many cases in real-time (more than 24 fps). When the image size reaches 2 Megapixels the frame rate is only 15 fps. Note that the current implementation does not optimize the G-buffer, which could be greatly compressed gaining extra frames per second. From Figure 4 we can also deduce that the timing scales nearly linearly with image size.

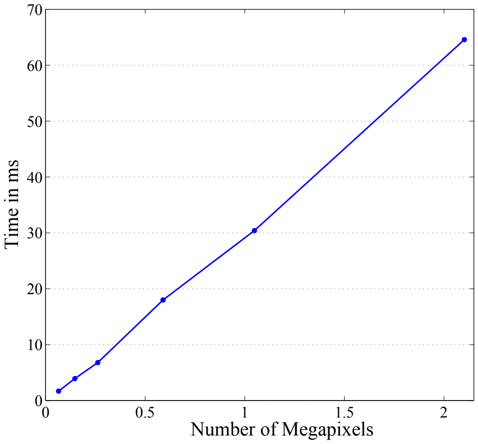


Fig. 4. Timing results of CG-TMO at different image sizes. Note that the timing scales nearly linearly with image size in Megapixels.

5. CONCLUSIONS

In this work, we presented a novel pipeline for tone mapping HDR images that were generated by a physically based renderer. Our work provides a tight integration between the rendering and tone mapping pipelines which were de-coupled before. The proposed approach allows to improve the quality of the tone mapping while keeping a straightforward and fast implementation. Finally, we showed that CG-TMO works in real-time in many cases, so it is suitable for VE, simulations, and video-games.

In future work, we would like to extend our technique to real-world content by using multi-view imaging techniques [27] for extracting depth, normal and albedo information.

Acknowledgment
We thank the anonymous reviewers for their comments which improved the final version of the paper. Alessandro Artusi is

partially supported by the Spanish Ministry of Science and Innovation Subprogramme Ramon y Cajal RYC-2011- 09372, TIN2013-47276-C6-1-R and from the Catalan Government 2014 SGR 1232. This work was partially supported by the EU FP7 ICT under the V-Must NoE (Grant Agreement 270404) and the European COST Action IC-1005.

REFERENCES

[1] Matt Pharr and Greg Humphreys, *Physically Based Rendering, Second Edition: From Theory To Implementation*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2nd edition, 2010.

[2] Tomas Akenine-Möller, Eric Haines, and Natty Hoffman, *Real-Time Rendering 3rd Edition*, A. K. Peters, Ltd., Natick, MA, USA, 2008.

[3] Frédéric Drago, Karol Myszkowski, Thomas Annen, and Norishige Chiba, "Adaptive logarithmic mapping for displaying high contrast scenes," *Computer Graphics Forum (Proceedings of Eurographics '03)*, vol. 22, no. 3, pp. 419–426, September 2003.

[4] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 839–846, 1998.

[5] Francesco Banterle, Alessandro Artusi, Kurt Debattista, and Alan Chalmers, *Advanced High Dynamic Range Imaging: Theory and Practice*, AK Peters (CRC Press), Natick, MA, USA, 2011.

[6] Erik Reinhard, Wolfgang Heidrich, Paul Debevec, Sumanta N. Pattanaik, Gregory Ward, and Karol Myszkowski, *High Dynamic Range Imaging: Acquisition, Display and Image-Based Lighting*, Morgan Kaufmann, second edition edition, May 2010.

[7] Alessandro Artusi, Jiří Bittner, Michael Wimmer, and Alexander Wilkie, "Delivering interactivity to complex tone mapping operators," in *Rendering Techniques 2003 (Proceedings Eurographics Symposium on Rendering)*, Per Christensen and Daniel Cohen-Or, Eds. Eurographics, June 2003, pp. 38–44, Eurographics Association.

[8] Marco Slomp and Manuel Oliveira, "Real-time photographic local tone reproduction using summed-area tables," in *CGI '08: Computer Graphics International 2008*, 2008.

[9] Chris Kiser, Erik Reinhard, Mike Tocci, and Nora Tocci, "Real time automated tone mapping system for HDR video," in *Proceedings of the IEEE International Conference on Image Processing*, Orlando, Florida, September 2012, IEEE, pp. 2749–2752, IEEE.

[10] Jiawen Chen, Sylvain Paris, and Frédo Durand, "Real-time edge-aware image processing with the bilateral grid," *ACM Trans. Graph.*, vol. 26, July 2007.

[11] Francesco Banterle, Massimiliano Corsini, Paolo Cignoni, and Roberto Scopigno, "A low-memory, straightforward and fast bilateral filter through subsampling in spatial domain," *Computer Graphics Forum*, vol. 31, no. 1, pp. 19–32, February 2012.

[12] Alessandro Artusi, Ahmet Oğuz Akyüz, Benjamin Roch, Despina Michael, Yiorgos Chrysanthou, and Alan Chalmers, "Selective local tone mapping," in *IEEE International Conference on Image Processing*, September 2013.

[13] Frédo Durand and Julie Dorsey, "Interactive tone mapping," in *Proceedings of the Eurographics Workshop on Rendering*. June 2000, Springer Verlag, Held in Brno, Czech Republic.

[14] Alexander Hausner and Marc Stamminger, "Tone mapping in vr environments," in *Eurographics 2004 Short Papers*, 2004.

[15] Josselin Petit, Roland Brmond, and Ariane Tom, "Evaluation of tone mapping operators in night-time virtual worlds," *Virtual Reality*, vol. 17, no. 4, pp. 253–262, 2013.

[16] Josselin Petit and Roland Brmond, "A high dynamic range rendering pipeline for interactive applications," *The Visual Computer*, vol. 26, no. 6-8, pp. 533–542, 2010.

[17] Frédo Durand and Julie Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," in *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, 2002, SIGGRAPH '02, pp. 257–266, ACM.

[18] Jack Tumblin, Jessica K. Hodgins, and Brian K. Guenter, "Two methods for display of high contrast images," *ACM Transaction on Graphics*, vol. 18, no. 1, pp. 56–94, 1999.

[19] Zeev Farbman, Raanan Fattal, Dani Lischinski, and Richard Szeliski, "Edge-preserving decompositions for multi-scale tone and detail manipulation," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 67:1–67:10, August 2008.

[20] Georg Petschnigg, Richard Szeliski, Maneesh Agrawala, Michael Cohen, Hugues Hoppe, and Kentaro Toyama, "Digital photography with flash and no-flash image pairs," *ACM Transaction on Graphics*, vol. 23, no. 3, pp. 664–672, 2004.

[21] Elmar Eisemann and Frédo Durand, "Flash photography enhancement via intrinsic relighting," *ACM Transaction on Graphics*, vol. 23, no. 3, pp. 673–678, 2004.

[22] Francesco Banterle, Alessandro Artusi, Elena Sikudova, Thomas Bashford-Rogers, Patrick Ledda, Marina Bloj, and Alan Chalmers, "Dynamic range compression by differential zone mapping based on psychophysical experiments," in *Proceedings of the ACM Symposium on Applied Perception*, New York, NY, USA, 2012, SAP '12, pp. 39–46, ACM.

[23] Robert L. Cook, "Stochastic sampling in computer graphics," *ACM Trans. Graph.*, vol. 5, no. 1, pp. 51–72, January 1986.

[24] R. Cozot A. Gruson, M. Ribardiére, "Eye-Centered Color Adaptation in Global Illumination," *Computer Graphics Forum*, vol. 32, no. 7, pp. 111–120, October 2013.

[25] Hojatollah Yeganeh and Zhou Wang, "Objective quality assessment of tone mapped images," *IEEE Transactions on Image Processing*, vol. 22, no. 2, pp. 657–667, February 2013.

[26] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *Trans. Img. Proc.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[27] Richard Szeliski, *Computer Vision: Algorithms and Applications*, Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010.