Vis Comput
DOI 10.1007/s00371-010-0430-5

ORIGINAL ARTICLE

A high dynamic range rendering pipeline for interactive applications

In search for perceptual realism

Josselin Petit · Roland Brémond

Abstract Realistic images can be computed at interactive frame rates for Computer Graphics applications. Meanwhile, High Dynamic Range (HDR) rendering has a growing success in video games and virtual reality applications, as it improves the image quality and the player's immersion feeling. In this paper, we propose a new method, based on a physical lighting model, to compute in real time a HDR illumination in virtual environments. Our method allows to re-use existing virtual environments as input, and computes HDR images in photometric units. Then, from these HDR images, displayable 8-bit images are rendered with a tone mapping operator and displayed on a standard display device. The HDR computation and the tone mapping are implemented in *OpenSceneGraph* with pixel shaders. The lighting model, together with a perceptual tone mapping, improves the perceptual realism of the rendered images at low cost. The method is illustrated with a practical application where the dynamic range of the virtual environment is a key rendering issue: night-time driving simulation.

Keywords Rendering · High dynamic range · Perceptual realism · Interactivity · Pixel shader

J. Petit · R. Brémond (✉)
Université Paris-Est, LEPSiS, 75015 Paris, France
e-mail: roland.bremond@lcpc.fr

J. Petit
e-mail: josselin.petit@lcpc.fr

J. Petit
Université Lyon 1, LIRIS, UMR5205, 69622, Lyon, France

1 Introduction

Immersion is a major goal for virtual reality applications. Among the factors which contribute to this objective, the perceptual realism of the displayed images is paramount [8]. The most straightforward way to achieve perceptual realism is to get physical realism. However, reaching physical realism raises tough problems, both in terms of real-time computation and physical modeling. First, true photorealism implies very accurate 3D models. Next, one must collect the photometric and colorimetric inputs (light source description, spectral representation, material texture, BRDF, etc.) needed for the illumination computation. Then, the full simulation of the scene illumination (either with radiosity or ray tracing methods) in large virtual environments is far from real time. And finally, one cannot avoid the technical limits of current display devices, which are not designed to display physical High Dynamic Range (HDR) luminance values but Low Dynamic Range (LDR) 8-bit images (however, see Seetzen et al. [35]).

Due to the limited luminance gamut of typical LDR display devices, one cannot avoid to lose physical realism somewhere along the rendering pipeline. For instance, usual LCD display devices cannot render light sources (they are limited around 300 cd/m^2), and cannot render very dark areas (the minimum luminance in an office may be around 1 cd/m^2). In contrast, the sensitivity of the human photoreceptors ranges between 10^{-6} cd/m^2 for starlight and 10^8 cd/m^2 for sunlight [9]. Thus, the input range of the physical visual signal must be mapped to the dynamic range of the display device (2 orders of magnitude only). The purpose of tone mapping operators is to transform the HDR images into LDR images which can be displayed on conventional display devices, still keeping perceptual realism.

Published online: 08 April 2010

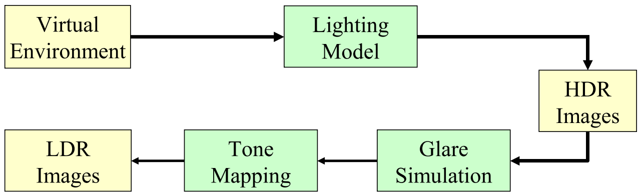

Fig. 1 Proposed approach: a physically-based HDR rendering is followed by a perceptual tone mapping

This perceptual realism is related to photometric indexes: for instance, the human visual system is sensitive to luminance contrast rather than to its absolute values [39]. Thus, one may modify the physical luminance values in the images, and still keep the perceptual levels of contrast perception [27, 40].

To address the issue of perceptual realism for interactive applications, we propose a two-step approach, summarized in Fig. 1.

1. A real-time model of illumination leads to an intermediate HDR physical representation of the images to be displayed, in photometric units, owing to a photometric description of the light sources and of the surfaces.
2. A glare simulation [16] followed by a tone mapping operator [14] transform the physically realistic HDR images into perceptually realistic LDR images on the fly.

The graphic pipeline is implemented in *OpenSceneGraph* with pixel shaders, and runs in real time (30 frames per second) in full HD. To our sense, our main contribution is that any existing virtual environment may be rendered with this pipeline, with only a few modifications of the environment and a true improvement of the perceptual realism.

2 Background

2.1 HDR rendering

There are many ways to perform the lighting of a virtual environment. For real-time applications, ray-tracer engines such as *OptiX* [25] are emerging; however, these tools are still limited to specific GPUs, and do not provide any interface between the virtual environment and the rendering engine. In video games, one popular method is to precompute light-maps with a global illumination algorithm, and to load them in the application which avoids computing the lighting in real time. For instance, interactive applications were proposed with this approach, including a tone mapping at interactive frame rates [7, 33]. The problem with pre-computed global illumination is that it excludes moving light sources and moving objects passing under the light sources. Thus, most interactive applications need alternative rendering strategies.

Recent advances in hardware enable to program *OpenGL* and *DirectX* applications to compute environment mapping (an efficient method to simulate complex mirroring surfaces) using HDR image-based lighting [6]. These applications simulate light reflection, refraction, Fresnel effect, chromatic dispersion, blooming and glare. However, the eight light sources provided by *OpenGL* do not suffice for most applications.

Some libraries for HDR rendering have emerged, notably in the open-source rendering engines *Ogre3D* [20] and *OpenSceneGraph* [26, 37]. HDR rendering is also becoming a standard feature in video games such as *Half Life* and *Crysis* [5, 23], where it makes participative media and bright light sources more realistic. HDR rendering is used with advanced lighting and post-processing effects, such as glare, which improves the apparent brightness of the light sources.

2.2 Glare rendering

Usual display devices are not bright enough to produce a glare experience. Thus, post-processing effects have been proposed in order to simulate glare in Computer Graphics applications [24]. Spencer et al. [36] used a point-spread-function as a convolution filter to simulate the physiological veiling luminance responsible for the glare, however not in real time. Some tone mapping operators included a glare simulation based on Spencer's paper [7, 41].

Instead of this physiological model, Gaussian functions may be used around light sources to simulate the glare experience [16]. This technique is fast and very popular in video games, and was implemented in our pipeline. Recently, two models of glare based on wave optics also gave convincing results; however, these techniques need more computing power [15, 32].

2.3 Tone mapping operators

The interest for HDR capture and imaging may be noticed from the beginning of photographic art, and was already addressed by painters. But there was an increasing interest in the image community since the development of tone mapping operators (see Reinhard et al. for a review [31]).

Some of these operators can process HDR videos [14, 28], create glare and blooming effects [17, 41] and simulate the visual acuity and adaptation to light and darkness, using criteria related to psychophysical data [9, 28]. These techniques are of great interest for the rendering of virtual environments because they make it possible to use an HDR representation of the scene illumination far away in the rendering pipeline, thus improving the final visual quality.

In the past 15 years, a number of operators were proposed, and the focus of attention moved towards assessing the visual quality of the resulting images [3, 18, 19, 43].

The key issue is the difference, depending on the operator, in terms of the resulting visual appearance. To select an operator for a given application, the main image quality criteria deal with immersion (e.g. visual appearance, see [22]) and fidelity (e.g. visual performance, see [11]).

Note that the tone mapping operators used in video games are generally very basic (e.g. a power function, a logarithm, or a sigmoid), due to the need for a fast rendering.

3 The HDR rendering pipeline

3.1 Framework

Perceptual realism is not always a major objective in video games and interactive applications, where esthetics and expressivity may be more important. However, when available, perceptual realism improves the immersion feeling. Our proposal is that an HDR rendering can take advantage of a physical description of the virtual environment (light intensity in candela, luminance in candela per square meter (cd/m^2), reflectance, etc.) leading to physically-based HDR images. Thus, an HDR rendering may strongly improve the perceptual realism of virtual environments.

In the following, we propose a pipeline for the rendering of HDR virtual environments on conventional display devices, summarized in Fig. 2 [29]. From an HDR photometric description of the light sources, together with an LDR virtual environment, a physically based rendering model computes HDR images of the environment, with physical values as close as possible to the simulated scene (Fig. 2, top of the pipeline). Then, a tone mapping operator computes LDR displayable images with perceptual attributes as close as possible to those in the HDR image (Fig. 2, bottom). As stated by Tumblin and Rushmeier, perceptual realism aims at "*a close match between real-world and display* (...) *sensations*" [38].

Of course, real-time applications lead to a trade-off between physical accuracy and computing speed. The pipeline computes an illumination model on the GPU, as well as a tone mapping operator, with pixel shaders in cascade (see Figs. 2 and 4). It also addresses the practical issue of reusing existing virtual environments, with minimal effort. The proposed framework includes 3 steps (the first one is off-line):

- Preprocessing of an existing LDR virtual environment, providing an HDR photometric description to the light sources.
- HDR rendering of the virtual environment (see top of Fig. 2 and Sect. 4.1).
- Post-processing of the HDR images (glare and tone mapping), to compute displayable LDR images (bottom of Fig. 2 and Sect. 4.3).

3.2 HDR light sources

To render virtual environments with luminance and color values related to real physical data, we propose a photometric interpretation of modified virtual environments. Depending on the available virtual environment, the light sources must be either modified or added.

Two descriptions are used for each light source: one for the illumination model, another one for the rendering of the direct illumination when the light source is in the field of view.

- For the illumination model, the intensity and the shape of the light sources are given in photometric and goniometric units, taken from lighting designers and lighting databases.
- For the direct illumination, the light source texture is converted from an LDR to an HDR texture image, according to photometric data. When such an emissive surface is found in the rendering process, then HDR texture is taken into account instead of the illumination computation (the `Emissive` component of the Material may also be used here, see the `OR` box in Fig. 2).

3.3 The HDR lighting model

To render virtual environments with luminance and color values related to real physical data, we propose a physical interpretation of virtual environments. These environments include texture information and materials description; however, these data are seldom linked with the physical properties of light emission and reflection.

As the sensor settings of the texture photographs are usually lost, we made an assumption about the visual tuning performed by the Computer Graphics artist when creating the virtual environment. Our hypothesis is that the final tuning (texture intensity, `Diffuse` component of the materials, etc.) was set in order to get the right visual appearance of any surface, with an `Ambient` light set to 1.0. Thus, given that these tunings were chosen on a screen with a given γ ($\gamma = 2.2$ in the following), we converted the texture images T into dimensionless reflectance maps ρ, allowing further illumination computations on the three color channels with:

$$\rho = T^{\gamma}. \tag{1}$$

This trick enables to re-use existing LDR virtual environments with a physical lighting model, using the HDR light sources described in photometric units.

Then, a physical luminance is computed for the three color channels on every pixel in real time. As in Phong's model, the luminance L is split into three components: $L = L_A + L_D + L_S$ (`Ambient`, `Diffuse` and `Specular` [23, 30]).

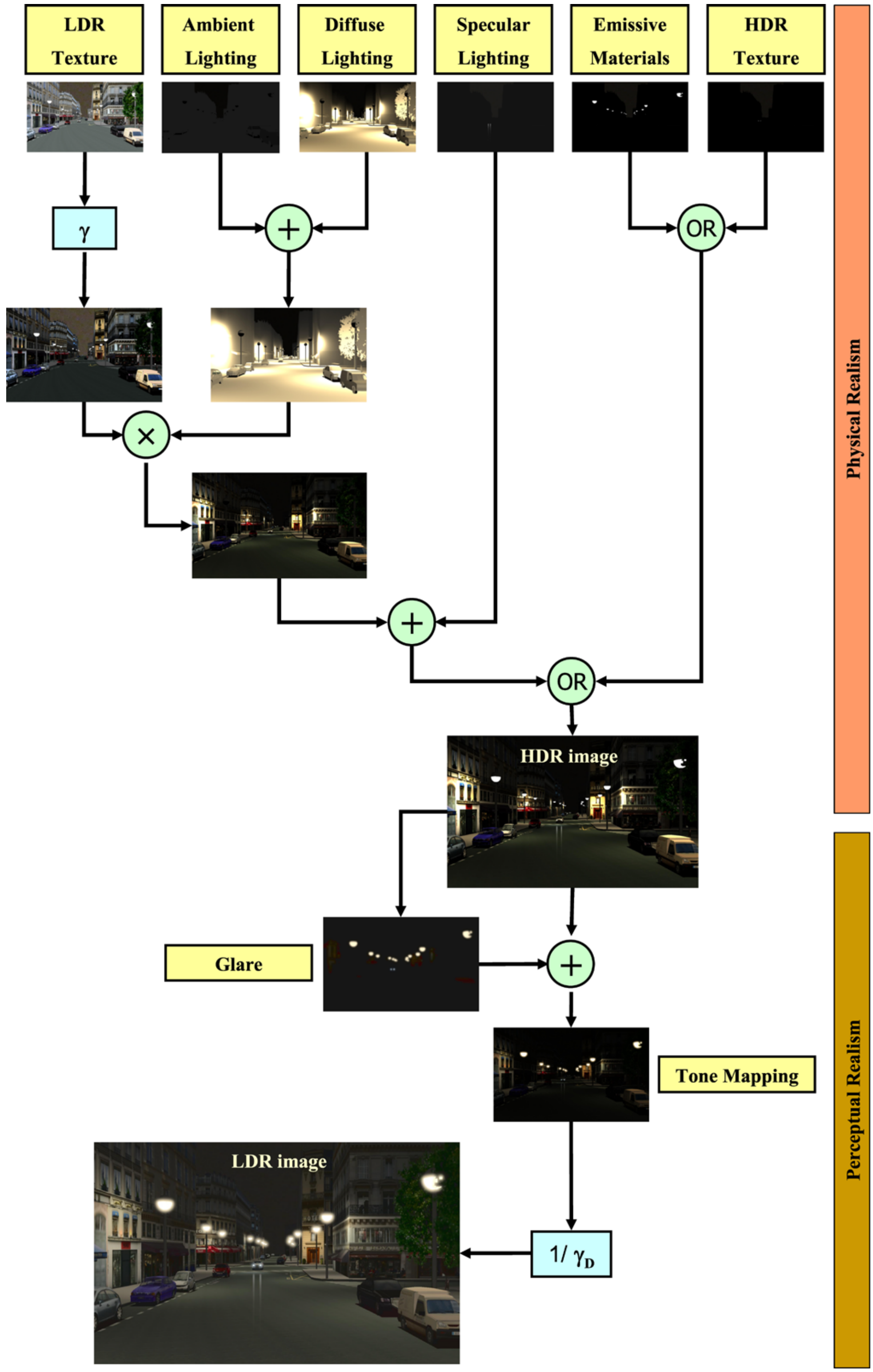

Fig. 2 HDR rendering pipeline for virtual environments based on a physical lighting model and a perceptual tone mapping. *Top*: HDR computation of the scene illumination (physical realism). *Bottom*: glare simulation, tone mapping and display (perceptual realism)

The HDR `Ambient` term L_A provides a base level of illumination to everything in the scene:

$$L_A = \rho I_A M_A, \tag{2}$$

where I_A is the `Ambient` intensity of the light source, M_A is the `Ambient` material, and ρ is the reflectance texture map from (1). Due to the fact that in *OpenGL*, the `Diffuse` and `Specular` components only take into account

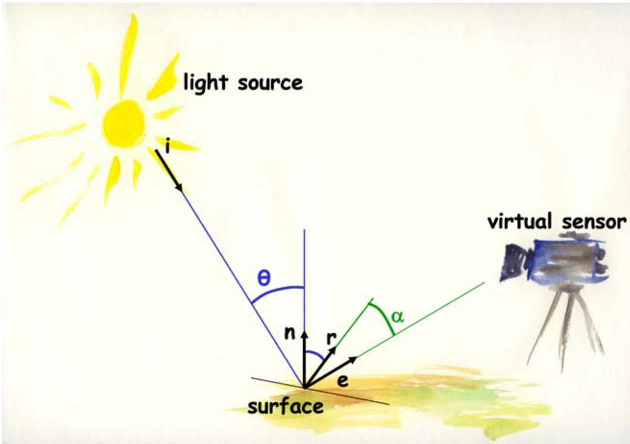

Fig. 3 Normalized vectors for the lighting computation

one light reflection, the `Ambient` term may be understood as a shortcut for:

- complex inter-reflections;
- uniform illumination by the sky dome.

If the `Ambient` luminance of a surface, due to the sky illumination, is $L_A = \rho E/\pi$ (with E the surface illuminance due to the sky), one may set $I_A = E/\pi$ for the `Ambient` sky contribution, while $M_A = 1$. Thus, the `Ambient` component is related to the physical illuminance level E due to the sky. Then, the inter-reflection component of L_A needs more information about the inter-reflections (it is not considered in the following).

The HDR `Diffuse` (L_D) and `Specular` (L_S) components are directional and depend on each light source and material photometric properties. The `Diffuse` luminance models light reflections equally scattered in all directions. The light reflected towards the virtual camera is computed as:

$$L_D = \frac{\rho I_D M_D \cos\theta}{\pi d^2}, \tag{3}$$

where I_D is the `Diffuse` intensity of the light source towards the reflecting surface (in candela), M_D the `Diffuse` material, $\cos\theta = -\mathbf{i}.\mathbf{n}$ is the illumination angle (see Fig. 3 for the vector notations), and d is the distance between the light source and the surface. The `Diffuse` luminance L_D is the result of a physical interaction between the `Diffuse` light component (set to the light source intensity I) and the `Diffuse` (Lambertian) material reflection, understood as the diffuse reflection coefficient ρM_D: for instance, for snow $\rho M_D \simeq 1.0$.

The specular reflection models the highlights caused by reflections on mirror-like surfaces. The `Specular` luminance L_S is computed with the same kind of physical interpretation of Phong's model as for L_D:

$$L_S = \frac{\rho I_S M_S \cos\alpha^{sh}}{\pi d^2}, \tag{4}$$

where $\cos\alpha = \mathbf{r}.\mathbf{e}$ (see Fig. 3): $\mathbf{r}$ is the reflection vector and $\mathbf{e}$ the "eye" vector (from the surface to the eyes). I_S is the `Specular` light intensity, M_S is the `Specular` material, and *sh* the material shininess. In our implementation, the default value was $sh = 64$.

3.4 Tone mapping

A number of tone mapping operators have been proposed in order to compress the dynamic range of HDR images into the 8 bits of usual digital images. For virtual environments, some specific issues arise, such as real-time computation and temporal adaptation, which few operators address.

In order to fulfill the requirements of our two-step approach (Fig. 1), we needed a tone mapping operator which secures key perceptual attributes in the images. We were driven towards operators which use psychophysical models of the human visual system. For interactive applications where the gaze direction is not known, we did not consider local adaptation models such as [7, 33]. Pattanaik et al.'s model was not selected either [28], as it focuses on a color appearance model [13] rather than on visual performance.

In the following, we adapted Irawan et al.'s algorithm [14], which is derived from the *Histogram Adjustment* method proposed by Ward et al. [41], and also takes into account the temporal adaptation of the human visual system with a psychophysical model derived from Pattanaik et al. [28]. This operator was designed to keep perceptual realism in terms of visual performance and temporal adaptation. Thus, we could take advantage of the physical illumination model in the virtual environment.

The first part of the operator is based on a generalized visibility threshold model which extends a conventional Threshold-versus-Intensity (TVI) function [9] to account for the viewer's visual adaptation state. A temporal adaptation model includes fast and slow neural mechanisms as well as photopigment bleaching. The second part of the operator uses the generalized threshold model (TVIA) as a constraint to compute the *Histogram Adjustment*, mapping "world" luminance to display luminance. With this constraint, "world" luminance values which are not visibly different will map to "display" luminance values which are not visibly different either.

4 Real-time implementation

The rendering pipeline was implemented with the pixel shader technology, which is versatile and supports an easy implementation of the physical equations. It is also efficient in computational speed, and allowed to use `float` values all along the physical lighting computations. The render-to-texture of the HDR images used a 32-bit texture buffer.

4.1 A pixel shader for lighting

A realistic real-time lighting simulation could not be reached, to date, with a large number of light sources. Moreover, the *OpenGL* pipeline is limited to 8 light sources, which is not sufficient, for instance, for a night-time driving simulator. We have implemented a potentially much higher number of HDR light sources in a single pixel shader, computing the HDR rendering in one pass.

The critical feature which limits the pipeline's performance is the number of light sources which contribute to a pixel's illumination. In order to allow a larger number of light sources, those which do not contribute in a significant way to a pixel illumination are not taken into account at this pixel for the `Diffuse` component. A threshold for the illuminance level was set to $E_{\min}$ (in Lux), so that a threshold distance d_t is computed for each light source:

$$d_t = \sqrt{\frac{I_{D\max}}{E_{\min}}}, \tag{5}$$

where $I_{D\max}$ is the maximum output intensity of the light source. For practical applications, $E_{\min}$ should be tuned to a value consistent with the illumination level in the scene.

4.2 Pixel shaders for the glare simulation

The tone mapping applies to an image which is the addition of the physical HDR image and a glare image (see Fig. 2). The glare image simulates the veiling glare due to the light sources, which cannot be experienced on conventional display devices.

The glare simulation is computed on the HDR image with an approach close to Kawase's [16], using a set of Gaussian functions around light source pixels, on a sub-sampled image (for computational efficiency). The sub-sampling is computed for each pixel at level i as the maximum over four pixels at level $i-1$, because the usual arithmetic mean would lead to counter-intuitive visual results for approaching lights. The filter F is computed as:

$$F = k\big[\lambda_1 G(\sigma) + \lambda_2 G(2\sigma) + \lambda_3 G(4\sigma)\big], \tag{6}$$

where $G(\sigma)$ is a normalized Gaussian function with variance σ, and λ_i are weighting coefficients set to $8/14$, $2/14$ and $4/14$ (so that $\lambda_1 + \lambda_2 + \lambda_3 = 1$). In contrast with Kawase, we normalized the Gaussian functions in order to control that the amount of light which is spread onto the image is proportional to the light source energy. The energy rate which contributes to the glare effect is k, which is set to 0.06 in the following. The Gaussian filters are implemented with 2 pixel shaders each, for the convolutions in the horizontal and vertical dimensions.

4.3 Pixel shaders for tone mapping

The tone mapping is a post-processing step in the rendering pipeline. Irawan et al.'s algorithm [14] was implemented with 9 pixel shaders in cascade, each one using the results from the previous one. The algorithm is based on Ward's *Histogram Adjustment* method as far as static tone mapping is concerned citeWard1997, and uses a physiological model of light and dark adaptation as far as temporal adaptation is concerned [28]. The shaders are described in Fig. 4. In the proposed implementation, the operator was restricted to the contribution of the cone photoreceptors, as even in the mesopic domain, the contribution of the rod photoreceptors has little effect on the luminance beyond 0.1 cd/m^2 [1], which is lower than even the darker areas of most display devices.

1. The first shader converts the HDR image into a luminance channel L and two color channels, from the RGB to the XYZ color space. Then, $\log_{10}(L)$ is computed in order to anticipate the next shader.
2. The adaptation luminance L_a is the overall light level our eyes perceive. It is computed as the $\log_{10}$-mean of the luminance, using a mip-mapping process from the previous shader. The higher level of the mip-map is the $\log_{10}$-mean of the entire image.
3. The luminance level where our eyes are fully adapted not only depends on the adaptation luminance $L_a(t)$, but also on the adaptation levels at previous time steps. Moreover, as the eyes adapt locally, a local adaptation state is computed for each pixel in the image (see Irawan et al. [14]). These mechanisms are implemented with a buffer containing the past adaptation values (see Fig. 4), updated at each time step.
4. The TVIA (Threshold versus Intensity and Adaptation) provides the minimum visible luminance difference for a given adaptation state and a given luminance value. The TVIA computation needs two shaders: the first one (shader 4) computes the amount of additional response ΔR needed in order to perceive a luminance offset ΔL. The minimum of ΔR is taken as the minimum response variation for a visible variation of luminance.
5. The second stage (shader 5) computes the TVIA function itself (see (6) in Irawan et al.'s paper).
6. The luminance histogram of the HDR image is computed on a low-resolution image obtained with the mip-mapping. In the same shader, the thresholds for each bin of the histogram are computed using the TVIA function, along with the number of pixels above threshold for each bin.

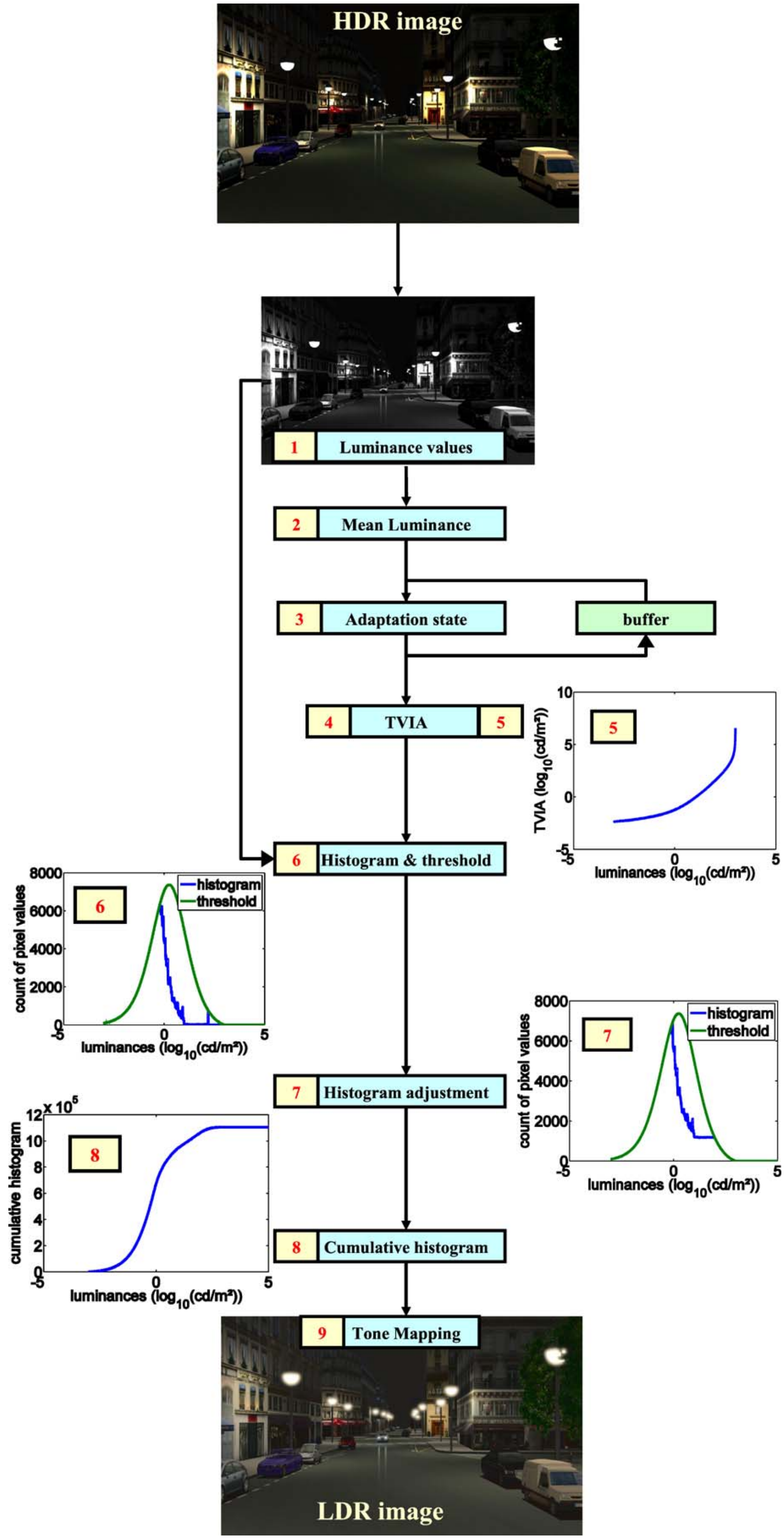

Fig. 4 The proposed pipeline implements Irawan et al.'s tone mapping operator in 9 pixel shaders in cascade, to convert an HDR image into a displayable LDR image. The numbers stand for the pixel shaders (the TVIA computation needs two shaders, 4 and 5). Shader 3 uses a memory buffer which stores the past values of the visual adaptation state

Fig. 5 Screenshots of the HDR night-time rendering of the *Rivoli* virtual environment. *Top*: original LDR environment used as input

7. The *Histogram Adjustment* is computed, spanning the trimmings in the histogram bins where the threshold is not reached.
8. The cumulated histogram is computed and normalized, leading to a tone mapping function between the HDR and LDR luminance values.
9. The last step is the tone mapping itself. The HDR luminance values are mapped to LDR values using the tone mapping function. In order to save computing time while keeping the overall image quality, the luminance values are interpolated, which lowers the number of bins in the histogram. Finally, the color information from the first stage is combined with the LDR luminance value to compute a displayable image in the RGB color space.

The tone-mapped image is finally displayed on a display device with a given γ_D value (see Fig. 2). We used $\gamma_D = 2.2$ for an LCD display device, and $\gamma_D = 1.8$ for a video projector.

5 Results

To illustrate the proposed rendering pipeline, videos were computed using the *Rivoli* virtual environment which was previously developed for a driving simulator application [10]. Figure 5 shows screenshots from these videos, as well as a view of the original LDR environment (a video is available online as supplementary material of the present paper).

One practical advantage of the proposed technique is that one can easily re-use such an existing LDR environment, where each surface includes an *OpenGL* material and/or texture. Virtual environments developed for video games and driving simulators are thus ready-to-use for HDR rendering with the proposed pipeline, providing that a limited list of changes is performed. In the *Rivoli* example, the changes in the virtual environment were the following:

– The texture images of the headlights of motor vehicles were replaced by HDR textures. These HDR textures were the previous LDR textures tuned to the physical luminance of the headlamps (photometric data from the University of Michigan were used [34]).
– Street lighting luminaries were added, with `Emissive` values set to photometric intensities, taken from actual street luminaries.

Both kinds of light source were shaped in order to get closer to actual goniometric data, through attenuation functions. For the street lighting, we used

$$I_D = I_{D\max}\left[0.5 + \frac{\sin(2\theta)}{2}\right] \tag{7}$$

with a maximum for $\theta = 45°$ down the street. The car low beams used 25°-width lobes directed 12° down the road.

The Specular term was modified for the low beam reflections on the road surface, as the Phong model is known to be bad at simulating the specular reflections under raking light.

The *Rivoli* environment, including 75 light sources, runs at 30 frames per second (fps), with 1680 × 1050 images on an Intel Core2 Duo, with a nVidia GeForce GTX 280 GPU. This performance was obtained with 70 bins in the luminance histogram (shaders 5–7 in Fig. 4), using the fifth mip-map level in the tone mapping pipeline. The minimum illuminance contribution was set to $E_{\min} = 1$ Lux.

6 Conclusion

We propose an HDR rendering pipeline for interactive applications, focusing on the perceptual realism of virtual environments. The pipeline was demonstrated in a city street environment at night. The main contribution is to improve the perceptual realism of existing virtual environments, with only small changes in the environment description. The only mandatory changes concern the light sources intensity and emissive shape. The HDR environment is then rendered using a photometric model of illumination allowing to include dozens of light sources. The resulting HDR images are then processed by a glare simulator and a tone mapping operator, which simulate eye vision.

Future work should address two main issues: image quality and image evaluation. In order to improve the pipeline performance in terms of image quality, several issues have to be addressed. An accurate rendering of glare effects may be implemented following Kakimoto et al. [15] or Ritschel et al. [32], however with a computational cost compared to the current implementation. Moreover, the key issue in glare rendering may not be the computation itself, but the light source intensity value in the virtual sensor's direction. Thus, instead of splitting the light source representation into an emissive part (for the scene illumination) and an HDR texture (for the direct sensor illumination), one promising approach may be to gather them into a single representation of the light source.

Of course Phong's model is far from perfect, but in our application it produced good results, due to the fact that the photometric inputs were well selected.

The main limitation of the physical model is that, using *OpenGL*, the light transport is limited to one reflection from the light sources towards the virtual camera (the Ambient component is only a rough approximation of the inter-reflections complexity). An improvement would be to pre-compute off-line the Ambient parameter L_A for each object with a global illumination limited to the static light sources, adjusting L_A to the inter-reflection contribution for each surface.

Another improvement concerns the photometric values of the Specular parameters of the surface materials, M_S and *sh*, which were both set to default values in our example, due to the lack of available photometric database. Future work includes the measurement of road surfaces' specularity and BRDF with a gonio-photometer developed at the LCPC [12]. Then, the "BRDF-shop" proposed by Colbert et al. may help to choose the BRDF parameters once the target shape is known [4], thus discounting Phong's model.

As the aim of the proposed rendering pipeline was perceptual realism, an important issue is to assess the visual quality of the displayed images. This may be done by comparing the displayed images with various state-of-the-art tone mapping operators, as most of them (at least the "global" ones) may be implemented at interactive frame rates [2]. The comparison method may either use subjective assessment as in video coding [42], or objective criteria, such as visible differences [21].

Acknowledgements This study was supported by a grant from the Laboratoire Central des Ponts et Chaussées. We wish to thank E. Dumont, V. Ledoux, V. Biri and B. Péroche for fruitful discussions.

References

1. Alferdink, J.W.A.M.: Target detection and driving behaviour measurements in a driving simulator at mesopic light levels. Ophthalmic Physiolog. Opt. **26**, 264–280 (2006)
2. Artusi, A., Bittner, J., Wimmer, M., Wilkie, A.: Delivering interactivity to complex tone mapping operators. In: Proceedings of Eurographics Symposium on Rendering, pp. 38–45 (2003)
3. Ashikhmin, M., Goyal, J.: A reality check for tone mapping operators. ACM Trans. Appl. Percept. **3**(4), 399–411 (2006)
4. Colbert, M., Pattanaik, S.N., Krivanek, J.: BRDF-Shop: Creating physically correct bidirectional reflectance distribution functions. IEEE Comput. Graph. Appl. **26**(1), 30–36 (2006)
5. Crytek: The Crytek web page. www.crytek.com/technology/cryengine-2/specifications/ (2009)
6. Debevec, P.: Tutorial: Image-based lighting. IEEE Comput. Graph. Appl. **22**(2), 26–34 (2002)
7. Durand, F., Dorsey, J.: Interactive tone mapping. In: Proceedings of Eurographics Symposium on Rendering, pp. 219–230 (2000)
8. Ferwerda, J.A.: Three varieties of realism in computer graphics. In: Proceedings SPIE Human Vision and Electronic Imaging, pp. 290–297 (2003)
9. Ferwerda, J.A., Pattanaik, S.N., Shirley, P., Greenberg, D.P.: A model of visual adaptation for realistic image synthesis. In: Proceedings of SIGGRAPH, pp. 249–258 (1996)
10. Giannopulu, I., Bertin, R., Brémond, R., Espié, S., Kapoula, Z.: Visuomotor strategies using driving simulators in virtual and prerecorded environment. Adv. Transp. Stud. **14**, 49–56 (2008)
11. Grave, J., Brémond, R.: A tone mapping algorithm for road visibility experiments. ACM Trans. Appl. Percept. **5**(2), art. 12 (2008)
12. Hubert, R., Canestrelli, M., Richard, J.P.: Le laboratoire de photométrie pour les études de visibilité et de la qualité des équipements de la route. Bull. Liaison Lab. Ponts Chaussées **176**, 19–27 (1993) (A photometry laboratory dedicated to road equipement measurements)

13. Hunt, R.W.G.: The Reproduction of Color, 5th ed. Fountain Press (1995)
14. Irawan, P., Ferwerda, J.A., Marschner, S.R.: Perceptually based tone mapping of high dynamic range image streams. In: Proceedings of Eurographics Symposium on Rendering, pp. 231–242 (2005)
15. Kakimoto, M., Matsuoka, K., Nishita, T., Naemura, T., Harashima, H.: Glare generation based on wave optics. Comput. Graph. Forum **24**(2), 185–193 (2005)
16. Kawase, M.: Practical implementation of high dynamic range rendering. In: Game Developers Conference (2004)
17. Krawczyk, G., Myszkowski, K., Seidel, H.P.: Perceptual effects in real-time tone mapping. In: Proceedings of SCCG, pp. 195–202. ACM, New York (2005)
18. Kuang, J., Yamaguchi, H., Liu, C., Johnson, G.M., Fairchild, M.D.: Evaluating HDR rendering algorithms. ACM Trans. Appl. Percept. **3**(3), 286–308 (2007)
19. Ledda, P., Chalmers, A., Troscianko, T., Seetzen, H.: Evaluation of tone mapping operators using a high dynamic range display. In: Proceedings of SIGGRAPH, pp. 640–648. ACM, New York (2005)
20. Luksh, C.: Realtime HDR rendering. Tech. Rep., Institute of Computer Graphics and Algorithms, TU Vienna (2007)
21. Mantiuk, R., Daly, S., Myszkowski, K., Seidel, H.P.: Predicting visible differences in high dynamic range images. In: Model and its calibration. In: Proceedings of SPIE Human Vision and Electronic Imaging X, pp. 204–214. SPIE (2005)
22. McNamara, A.: Exploring visual and automatic measures of perceptual fidelity in real and simulated imagery. ACM Trans. Appl. Percept. **3**(3), 217–238 (2006)
23. Mitchell, J., McTaggart, G., Green, C.: Shading in valve's source engine. In: Proceedings of SIGGRAPH, pp. 129–142. ACM, New York (2006)
24. Nakamae, E., Kaneda, K., Okamoto, T., Nishita, T.: A lighting model aiming at drive simulators. Comput. Graph. **24**(4), 395–404 (1990)
25. nVidia: The OptiX web page. www.nvidia.com/object/optix (2009)
26. OpenSceneGraph: The OpenSceneGraph website. www.openscenegraph.org/projects/osg (2009)
27. Pattanaik, S.N., Ferwerda, J.A., Fairchild, M.D., Greenberg, D.P.: A multiscale model of adaptation and spatial vision for realistic image display. In: Proceedings of SIGGRAPH, pp. 287–298. ACM, New York (1998)
28. Pattanaik, S.N., Tumblin, J., Yee, H., Greenberg, D.P.: Time-dependent visual adaptation for fast realistic image display. In: Proceedings of SIGGRAPH, pp. 47–54. ACM, New York (2000)
29. Petit, J., Brémond, R.: Interactive high dynamic range rendering for virtual reality applications. In: Proceedings of Virtual Reality and Software Technology, Kyoto, Japan, pp. 251–252. ACM, New York (2009) (poster abstract)
30. Phong, T.B.: Illumination for computer generated pictures. Commun. ACM **18**(6), 311–317 (1975)
31. Reinhard, E., Ward, G., Pattanaik, S.N., Debevec, P.: High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting. Morgan Kaufmann, New York (2005)
32. Ritschel, T., Ihrke, M., Frisvad, J.R., Coppens, J., Myszkowski, K., Seidel, H.P.: Temporal glare: Real-time dynamic simulation of the scattering in the human eye. Comput. Graph. Forum **28**(2), 183–192 (2009)
33. Scheel, A., Stamminger, M., Seidel, H.P.: Tone reproduction for interactive walkthroughs. Comput. Graph. Forum **19**(3), 301–312 (2000)
34. Schoettle, B., Sivak, M., Flannagan, M.J., Kosmatka, W.J.: A market-weighted description of low-beam headlighting patterns in Europe. Tech. Rep., University of Michigan Transportation Research Institute (2003)
35. Seetzen, H., Heidrich, W., Stuerzlinger, W., Ward, G., Whitehead, L., Trentacoste, M., Ghosh, A., Vorozcovs, A.: High dynamic range display systems. ACM Trans. Graph. **23**(3), 760–768 (2004)
36. Spencer, G., Shirley, P., Zimmerman, K., Greenberg, D.P.: Physically based glare effect for digital images. In: Proceedings of SIGGRAPH, pp. 325–334. ACM, New York (1995)
37. Tevs, A.: The osgPPU web page. projects.tevs.eu/osgppu (2009)
38. Tumblin, J., Rushmeier, H.: Tone reproduction for realistic images. IEEE Comput. Graph. Appl. **13**(6), 42–48 (1993)
39. Wandell, B.: Foundations of vision. Sinauer associates, Sunderland, MA, USA (1995)
40. Ward, G.: A contrast based scale-factor for image display. In: Graphic Gems IV, pp. 415–421. Academic Press Professional, San Diego (1994)
41. Ward, G., Rushmeier, H., Piatko, C.: A visibility matching tone reproduction operator for high dynamic range scenes. IEEE Trans. Vis. Comput. Graph. **3**(4), 291–306 (1997)
42. Winkler, S.: Digital Video Quality: Vision Models and Metrics. Wiley, New York (2005)
43. Yoshida, A., Blanz, V., Myszkowski, K., Seidel, H.P.: Perceptual evaluation of tone mapping operators with real world scenes. In: Proceedings of the SPIE, vol. 5666, pp. 192–203 (2005)

Josselin Petit is a Ph.D. student in Computer Graphics. He works at the Laboratoire Central des Ponts et Chaussées (Paris, France) in the LEPSiS team. He also belongs to the LIRIS lab, Lyon University (France). His main topics are high dynamic range imaging and rendering, and real-time rendering.

Roland Brémond is researcher at the Laboratoire Central des Ponts et Chaussées (Paris, France) since 1996, and since 2009 in the LEPSiS team. He had his Engineer Degree from the ENPC in Paris in 1990 and his Ph.D. from the ENSMP in 1993 in the field of Image Processing. His current researches focus on human vision for road safety applications, including computer graphics, driving simulation, photometry and visual perception.