

Interactive High Dynamic Range Rendering for Virtual Reality Applications

Josselin Petit*

Lyon I University, LIRIS (France) and Paris Est University, LEPSIS (France)

Roland Brémond

Paris Est University, LEPSIS (France)†



Figure 1: Snapshots from the Rivoli virtual database, using the proposed HDR rendering pipeline under night-time conditions with the TMO from [Irawan et al. 2005]. The virtual environment includes static and moving light sources.

1 Overview

Realistic images can now be computed at interactive frame rates for Virtual Reality applications. Meanwhile, High Dynamic Range (HDR) rendering has a growing success in video games and virtual reality, as it improves the image quality and the player’s immersion feeling. We propose a new method, based on a simplified physical model of light propagation, to compute in real time a HDR illumination in a Virtual Environment (VE). Our method allows to re-use existing Low Dynamic Range (LDR) Virtual Databases as input in order to compute the HDR images. Then, from these HDR images, displayable 8-bit images are rendered with a Tone Mapping Operator (TMO) and displayed on a standard display device. The HDR computation and the TMO are implemented in *OpenSceneGraph* (OSG), working in real time with pixel shaders. The method is illustrated with a practical application where the Dynamic Range of the VE is a key rendering issue: driving at night on a Driving Simulator. The VE includes light sources such as road lighting and car headlights. The visual impact of these light sources is twofold: first, the light sources themselves may be included in the field of view; second, the objects luminance is due to the physical interaction between these light source and the photometric properties of the surface (reflectance, BRDF).

2 Previous Work

Tone Mapping Operators (TMO) convert HDR images (with a large gamut of luminance) into displayable LDR images suited for 8-bit display devices (with a limited gamut of luminance), however keeping the main visual features in the images. TMO have been proposed to process HDR video, to create glare and blooming effects on bright areas, and to simulate visual acuity and adaptation to light and darkness using psychophysical models from vision science. These techniques are of great interest for the rendering of VE because they make possible to use a HDR representation of the Environment far away in the rendering pipeline.

One popular method in videogames is to pre-compute lightmaps with a global illumination algorithm, and to load them in the application, which avoids computing the lighting in real time. The problem with these viewers is that the radiosity stage must be pre-computed (radiosity is far from real time), and thus cannot include moving light sources or moving objects passing under the light sources. Thus, most VR applications need alternative rendering strategies. Recent advances in hardware have enabled to program

OpenGL and *DirectX* applications to compute environment mapping, which is an efficient method to simulate complex mirroring surfaces, using HDR Image-Based Lighting (IBL). These applications simulate light reflection, refraction, Fresnel effect, chromatic dispersion, blooming and glare. Moreover, HDR Rendering is becoming a standard feature in many games, such as *Half Life* and *Crysis*, which use HDR rendering to make participative media and bright light sources more realistic. However the TMO used in videogames are generally very basic, due to the need for a fast rendering. These operators create nice images, but they don’t take advantage of a physical description of the VE to preserve the displayed contrasts and brightness of the simulated environment. Some libraries have emerged for HDR rendering, notably the open-source rendering engines *Ogre3D* and *OpenSceneGraph* (OSG) [OpenSceneGraph 2009]. The OSG library allows the users to load an object textured with an HDR image, and displays LDR tone mapped images in real time.

3 Real Time HDR Rendering

We propose a rendering pipeline for the computation and rendering of HDR virtual environments. Our framework is split in three steps: (1) Re-configuration of existing VE databases in order to allow HDR rendering based on a physical model; (2) HDR computation in real-time with pixel shaders in *OpenGL*; and (3) Post-processing of the HDR images (tone mapping and glare), in order to compute displayable LDR images.

This pipeline was designed with two main goals. The first one was to compute HDR images with luminance and colorimetric values related to physical values. We propose then a physical interpretation of modified LDR databases, including light sources and reflection properties. The light source intensities are described in photometric units (candela). The second goal was to use, when possible, existing LDR VE, and to modify these environments only when the LDR description is irrelevant for HDR rendering (e.g. for light sources). Our approach was to modify VE from modeling software such as *3DSMax* in order to take into account the light sources, first for the direct lighting, then for the illumination, using a simplified real-time physical model.

The light sources are described twice in the modified VE. For the direct rendering of the light sources, one may use a HDR texture created from a set of LDR photographs taken with various exposures (bracketing), or the Emissive coefficient of the *OpenGL* Material. For the lighting computation, we used a per-pixel lighting in the pixel shader, where the photometric intensity I is given in candela for each light source.

*e-mail:josselin.petit@lcp.fr

†e-mail:roland.bremond@lcp.fr

A key idea in our rendering pipeline is to convert the LDR texture images into dimensionless reflectance maps, allowing further illumination computations. Namely, the texture of the LDR surfaces are converted into spatial distributions of light reflection. Although 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 Database, assuming that the final tuning in the LDR Database (texture intensity and Diffuse Material) are set in order to get the right visual appearance of the surface, lighted with an Ambient light. Thus, given that these tuning are performed on a screen with a given γ (we set $\gamma = 2.2$ in the following), we convert the textures images into reflectance maps, on the 3 channels Red, Green and Blue, with $R_{i,j} = T_{i,j}^\gamma \times D$, where R is for Reflectance, T for Texture and D is the Diffuse component of the surface OpenGL Material. From this modified VE, it is possible to compute a physical luminance and color at each pixel in real time, with a simplified lighting model. We demonstrated this idea, re-using a daytime virtual database from the LEPSIS Lab. previously used in [Giannopulu et al. 2008], the *Rivoli* database (see Fig. 1). Following the physical laws of illumination, the luminance (the light directed towards the camera) is computed as the product (mode MODULATE in OpenGL) of illuminance E and reflectance $R_{i,j}$. The illuminance (the light send to the surface), in turn, is computed for each light source k as:

$$E_k = \frac{I_k \cos\theta}{d^2} \quad (1)$$

where I_k is the light intensity of source k in the pixel direction, θ is the illumination angle with the lit surface, and d the distance between the light source and the surface.

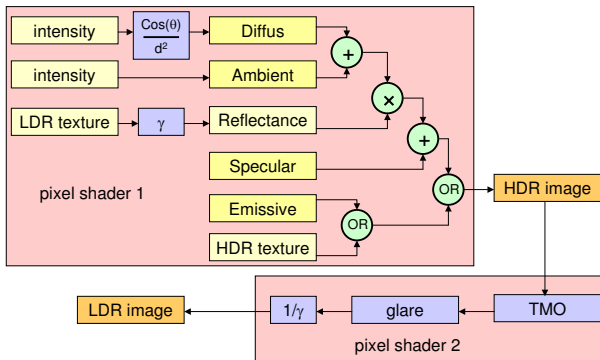


Figure 2: HDR Rendering pipeline based on a simplified physical model of illumination (shader 1), and LDR rendering (shader 2).

In addition to this computation of the diffuse lighting, more terms are computed for the Ambient, Specular and for a possible HDR texture. Fig. 2 shows the proposed graphic pipeline, which is implemented in *OpenGL* using two pixel shaders, one for the HDR rendering, and one for the LDR rendering. The Ambient term stands, for instance, for the sky illumination which cannot be neglected, even at night. The Specular component is taken from the existing *OpenGL* Material, however the reflected light may be over 1.0. Moreover, some textures are modified from LDR images to HDR images: this trick allowed to take into account the direct illumination of the virtual camera by the light sources in the field of view. The texture of a light source is set at a level far above 1.0, that is, to its physical luminance in cd/m^2 . When such an emissive surface is found in the rendering process, this HDR texture is taken into account instead of the previous light reflection computation (the Emissive Material may be used if available, see the OR boxes in Fig. 2).

A number of TMO have been proposed in order to compress the dynamic range of HDR images into the 8 bits of usual digital images. However, depending on the operator, the rendered images may be visually very different. The more relevant TMO use psychophysical models of the Human Visual System (HVS). At this stage, the rendering pipeline is fully implemented with pixel shaders with the global version of Reinhard's TMO [Reinhard et al. 2002], while [Irawan et al. 2005] is not implemented yet in a pixel shader, so that the images in Fig. 1 were computed offline under Matlab.

4 Results

In order to demonstrate the rendering pipeline, a video was computed in the *Rivoli* Virtual Environment at night (see Fig. 1). One advantage of the proposed technique is that its design makes it easy to re-use an existing LDR database, where each surface is described using an OpenGL Material and/or a texture image. In our example, a limited list of changes were needed in order to render the daylight scene by night in HDR. The car light Texture images were replaced by HDR Texture images: these HDR images are the previous LDR Texture, normalized with a maximum value equal to the light source intensity. The road lighting light sources were modified, using Emissive values above 1.0 in the OpenGL Material (They are set to photometric values). At this stage, the rendering pipeline is fully implemented with pixel shaders with the global version of Reinhard *et al.*'s TMO [Reinhard et al. 2002], allowing to use up to 20 light sources in real time. Irawan *et al.*'s TMO [Irawan et al. 2005] gives better visual results, however it is not implemented yet in a pixel shader, so that the images Fig. 1 were computed offline with Matlab.

5 Conclusion

We have proposed a real-time HDR rendering framework for interactive VE, using a modified version of a standard LDR database and a new rendering pipeline, with two pixel shaders in cascade. The HDR images are rendered using a simplified photometric model of light propagation, and then processed by a TMO, which simulates eye vision, including temporal adaptation. We hope that this strategy may help game developers to improve existing databases at low cost, which is a good starting point if one wishes to extend the proposed framework to commercial applications. Our framework is innovative because we propose a physical (photometric) interpretation of the image rendering pipeline. The gain is a better visual fidelity, especially when there are bright and dark areas at the same time on the screen. Future work first includes the implementation of [Irawan et al. 2005] in a pixel shader.

References

- GIANNOPULU, I., BERTIN, R., BRÉMOND, R., ESPIÉ, S., AND KAPOULA, Z. 2008. Visuomotor strategies using driving simulators in virtual and pre-recorded environment. *Advances in Transportation Studies* 14, 49–56.
- IRAWAN, P., FERWERDA, J. A., AND MARSCHNER, S. R. 2005. Perceptually based tone mapping of high dynamic range image streams. In *Proceedings of Eurographics Symposium on Rendering*, 231–242.
- OPENSCENEGRAPH, 2009. The OpenSceneGraph website. www.openscenegraph.org/projects/osg.
- REINHARD, E., STARK, M., SHIRLEY, P., AND FERWERDA, J. 2002. Photographic tone reproduction for digital images. In *Proceedings of SIGGRAPH*.