Image-based Illumination on the GPU

Attila Barsi, László Szirmay-Kalos, and László Szécsi
Department of Control Engineering and Information Technology
Budapest University of Technology and Economics
Budapest, Magyar Tudósok krt. 2, H-1117, HUNGARY
szirmay@iit.bme.hu

Abstract. In many computer graphics applications it is desirable to augment the virtual objects with high dynamic range images representing a real environment. In order to provide the illusion that the virtual objects are parts of the real scene, the illumination of the environment should be taken into account when the virtual objects are rendered. Proper calculation of the illumination from an environment map may be extremely expensive if we wish to account for occlusion, self shadowing and specular materials. In this paper we present a method that does all the calculations on a per-frame basis, and is already real-time on non-cutting-edge hardware.

Key words: HDRI, shadow, GPU, reflection.

1. Introduction

When real and virtual objects have to be shown together, we have to illuminate the virtual objects by the light present in the real environment. The information of the environment lighting is usually available in form of high dynamic range images taken of the real world, called *environment maps* [1, 4, 5, 7]. This paper proposes a method to compute accurate environment lighting with shadows for dynamic scenes. Unlike previous image based lighting and shadow algorithms, our aim is to compute these methods in real-time without preprocessing of the scene, exploiting the features of the graphics hardware.

From the point of view of rendering, the environment map is a large, heterogeneous area light source. Alternatively, the environment map can also be imagined as a large sky dome or a sky cube (figure 1). The illumination of this source on the virtual objects can be obtained by tracing rays from the points of the virtual object in the directions of the environment map, and checking whether or not occlusions occur [6, 8, 11]. However, the software ray-casting approach fails to deliver real-time frame rates, and a technique that allows for an effective hardware implementation should take a different, texture-based approach [9].

The key idea of our approach is to decompose the environment map to a finite number of directional domains. While visibility is checked with one sample in each domain with a depth map, the reflected radiance is computed accurately. The number of directional

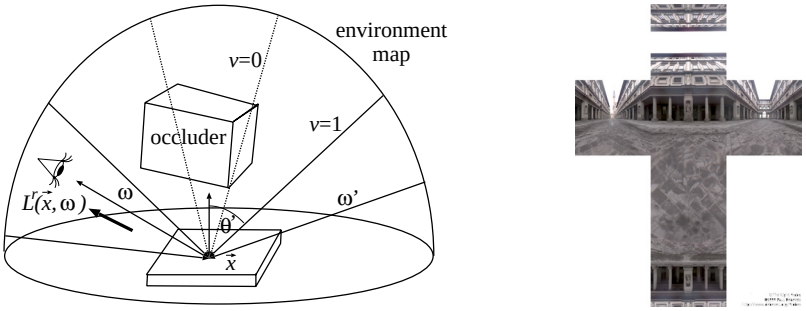


Fig. 1. The concept of environment mapping and an environment map stored in a cube map.

domains can be considered fairly constant, so the corresponding depth maps can be calculated in run-time, allowing for dynamic scenes.

2. The proposed method

In order to compute the image of a virtual scene under environment illumination, we should evaluate the reflected radiance L^r of every visible point $\vec{x}$ at view direction $\vec{\omega}$:

$$L^r(\vec{x}, \vec{\omega}) = \int_{\Omega'} L^{env}(\vec{\omega}') \cdot f_r(\vec{\omega}', \vec{x}, \vec{\omega}) \cdot \cos\theta' \cdot v(\vec{x}, \vec{\omega}')\ d\omega',$$

where Ω' is the set of all directions, $L^{env}(\vec{\omega}')$ is the radiance of the environment map at direction $\vec{\omega}'$, f_r is the Bi-directional Reflectance Distribution Function (BRDF), θ' is the angle between the surface normal and illumination direction $\vec{\omega}'$, and $v(\vec{x}, \vec{\omega}')$ is the *visibility function* checking whether no virtual object is seen from $\vec{x}$ at direction $\vec{\omega}'$ (that is, the environment map can illuminate this point from the given direction). In order to estimate this integral, directional domain Ω' is decomposed to solid angles $\Delta\omega'_i, i = 1, \ldots, N$ meeting the following criteria:

- the radiance is roughly uniform in each domain,
- the solid angles are small and are inversely proportional to the average radiance, thus it is enough to test the visibility of the environment map with a single sample in each solid angle:

$$L^r(\vec{x}, \vec{\omega}) \approx \sum_{i=1}^{N} v(\vec{x}, \vec{\omega}'_i) \cdot \tilde{L}_i^{env} \cdot a(\Delta\omega'_i, \vec{\omega}),$$

where

$$\tilde{L}_i^{env} = \int\limits_{\Delta\omega_i'} L^{env}(\vec{\omega}')\ d\omega'$$

is the *total incoming radiance* from solid angle $\Delta\omega_i'$, and

$$a(\Delta\omega_i', \vec{\omega}) = \frac{1}{\Delta\omega_i'} \cdot \int\limits_{\Delta\omega_i'} f_r(\vec{\omega}', \vec{x}, \vec{\omega}) \cdot \cos\theta'\ d\omega'.$$

is the *average reflectivity* from solid angle $\Delta\omega_i'$ to viewing direction $\vec{\omega}$. In order to use this approximation, the following tasks need to be solved:

1. The directional domain should be decomposed meeting the prescribed requirements, and the total radiance should be obtained in them. Since these computations are independent of the objects and of the viewing direction, we can execute them in the preprocessing phase.
2. The visibility of the environment map in the directions of the centers of the solid angles needs to be determined. Since objects are moving, this calculation is done on the fly. Note that this step is equivalent to shadow computation assuming directional light sources.
3. The average reflectivity values need to be computed and multiplied with the total radiance and the visibility for each solid angle. Since the average reflectivity also depends on the normal vector and viewing direction, we should execute this step as well on the fly.

2.1. Decomposition of the environment map

For a static environment map, the generation of sample directions should be performed at loading time, making it a non time-critical task. The goal is to make the integral quadrature accurate while keeping the number of samples low. This requirement is met if solid angles $\Delta\omega_i'$ are selected in a way that the environment map radiance is roughly homogeneous in them, and their size is inversely proportional to this radiance. The task is usually completed by generating random samples with a probability proportional to the power of environment map texels, and applying *Lloyd's relaxation* [3] to spread the samples more evenly. A weighted version of the relaxation method may be used to preserve the density distribution. As the basic idea of Lloyd's relaxation is to move the sample points to the center of their respective Voronoi areas, the relatively expensive computation of the Voronoi mesh is necessary.

Recently an extremely fast algorithm using *Penrose tiling* has been published to solve this problem [10]. This method should be considered if non-static environment maps are to be used. However, if we wish to assign the most accurate light power values to the sampled directions, we have to add up the contributions of those texels, which fall into

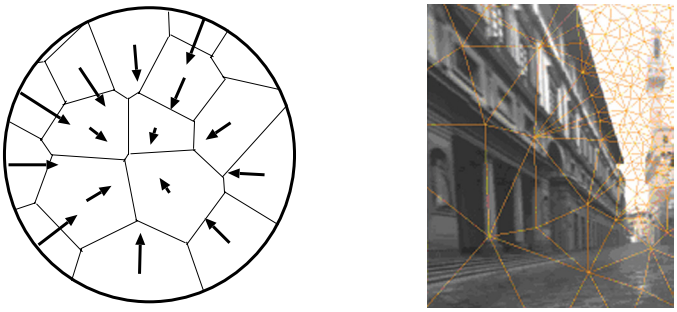


Fig. 2. Radiance values of texels within the Voronoi areas are summed to compute their total radiance and the final Delaunay grid on high dynamic range image. The centers of Voronoi cells are the sample points.

the Voronoi region of the direction. In this case, Lloyd's relaxation means no significant overhead, as rigorously summing the texel contributions takes more time (figure 2).

Centers of these cells define sample directions $\vec{\omega}_i'$ for which the visibility will be tested. Having identified the solid angles, the radiance is summed in their texels to obtain total incoming radiance $\tilde{L}_i^{env}$. In order to prepare for the fast calculation of the reflectivity, the approximate radius r_i of these cells are also obtained, which corresponds to the angle of view in which this subdomain is visible from the center of the environment map.

2.2. Visibility determination

Having decomposed the environment map to solid angles and deciding that the visibility is tested with a single sample in each subdomain, the problem is traced back to calculation of the shadow caused by directional light sources. In order to render shadows effectively, a hardware-supported shadow technique has to be applied. *Depth map shadows* [2] generated for every discrete direction are well suited for the purpose (figure 3).

When the resolution of the depth map is 256×256 and each element stores 16 bit depth values, one depth map requires 0.13 Mb storage space. For a typical scenario of 100 sampling points in the environment map, we use 13 Mb of memory and we need to perform one hundred texture reads in the fragment shader.

2.3. Computation of the average reflectivities

When average reflectivity $a(\Delta\omega_i', \vec{\omega})$ from solid angle $\Delta\omega_i'$ to viewing direction $\vec{\omega}$ is computed, we have to make simplifications to make the method real time. We consider a standard diffuse + specular BRDF, where the specular part is defined by the Phong

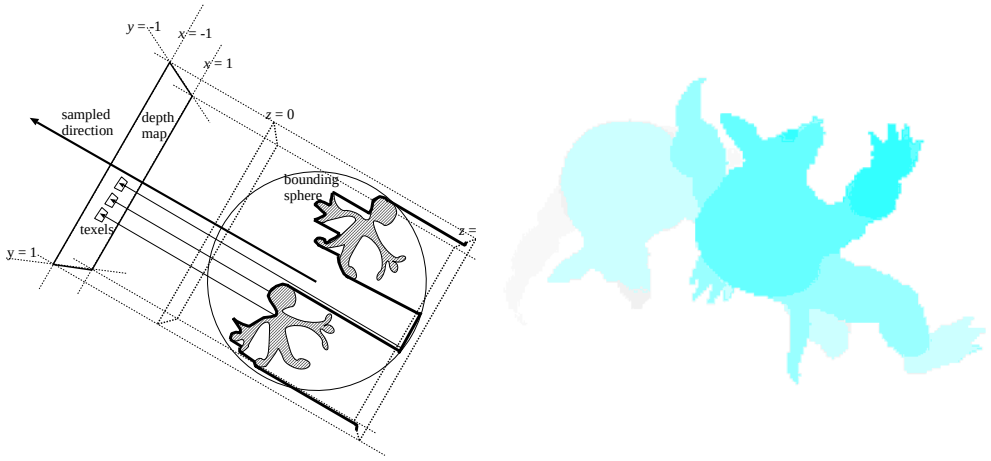


Fig. 3. The concept of depth mapped shadows for the directional lights corresponding to Voronoi cells and a particular depth image.

model:

$$f_r(\vec{\omega}', \vec{x}, \vec{\omega}) \cdot \cos\theta' = k_d \cdot \cos\theta' + k_s \cdot \cos^n \psi,$$

where k_d is the diffuse reflection parameter, k_s is the specular reflectivity, n is the shininess of the surface, and ψ is the angle between the ideal reflection direction of view vector $\vec{\omega}$ and the illumination direction $\vec{\omega}'$. Let us first examine the diffuse reflection. Since in this case the integrand is a cosine function, which has low variation, the integral is estimated from a single sample associated with the center of the Voronoi region:

$$a_d(\Delta\omega_i', \vec{\omega}) \approx k_d \cdot \cos\theta_c$$

where θ_c is the angle between the surface normal and direction $\vec{\omega}_i'$ corresponding to the center of this Voronoi region.

The case of specular reflection is more complicated, since it may have high variation when the Voronoi region contains the ideal reflection direction. Suppose that Voronoi cell $\Delta\omega_i'$ can be approximated by a spherical circle of angle r_i, and let us denote the angle between the ideal reflection direction and the center of the Voronoi cell by ψ_c. If the ideal reflection direction is not contained by $\Delta\omega_i'$, i.e. $\psi_c > r_i$, then a single sample is usually enough:

$$a_s(\Delta\omega_i', \vec{\omega}) \approx k_s \cdot \cos^n \psi_c.$$

However, when the ideal reflection direction is contained by the Voronoi cell, the high variation of the function would result in a large error if only a single sample were considered. Let us first examine a special case when the ideal reflection direction is equal to the center of the Voronoi cell, that is $\psi_c = 0$. In this special case, the average

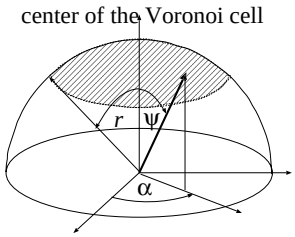


Fig. 4. Average reflectance computation when $\psi_c = 0$

reflectivity can be analytically determined. We work with angle ψ between the ideal reflection direction and angle α that is between the projection of $\vec{\omega}'$, and a reference direction on the plane perpendicular to the the ideal reflection direction (figure 4). When $\psi_c = 0$ the average specular reflectivity $a_s(\Delta\omega_i', \vec{\omega})$ is:

$$\frac{\int_{\alpha=0}^{2\pi}\int_{\psi=0}^{r_i} k_s \cdot \cos^n\psi \cdot \sin\psi \; d\psi d\alpha}{\int_{\alpha=0}^{2\pi}\int_{\psi=0}^{r_i} \sin\psi \; d\psi d\alpha} = k_s \cdot \frac{(1-\cos^{n+1} r_i)}{(n+1)\cdot(1-\cos r_i)}.$$

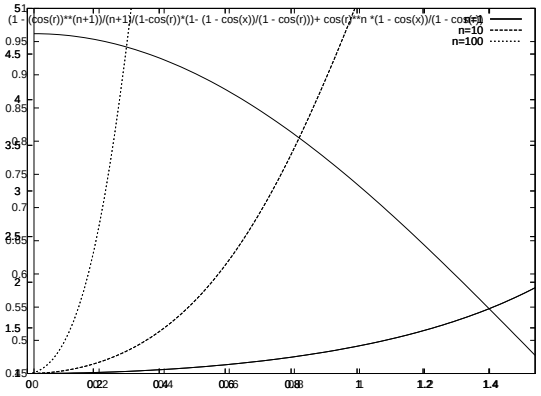


Fig. 5. Ratio of the approximated and exact average reflectivities when $\psi_c = 0$ as a function of $r_i = 0 \ldots \pi/2$ for different shininess parameters n. Note that the error is larger than 100%, i.e. the ratio is greater than 2, if the Voronoi cell is larger than $r_i = 0.6$ for shininess $n = 10$ or if the Voronoi cell is larger than $r_i = 0.18$ for shininess $n = 100$.

Note that if we took just a single sample selected by $\psi_c = 0$, then the approximation

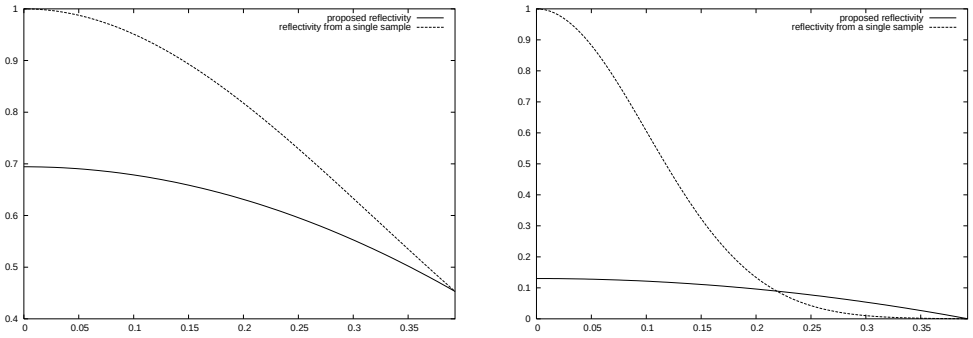
of the average albedo would be k_s, thus the ratio of the approximated and the real radiance would be

$$\frac{(n+1)\cdot(1-\cos r_i)}{(1-\cos^{n+1} r_i)}.$$

Figure 5 shows this ratio as a function of Voronoi cell size r_i for different shininess values. Note that the ratio can be very far from 1, which means a large error. This error is also visible in the left image of figure 7 that is rendered with using a single sample in each cell.

Note that we obtained an exact formula for the case of $\psi_c = 0$ and an approximation for $\psi_c > r_i$. In order to propose a good approximation for angles $0 < \psi_c < r_i$, we blend these results with weights $(1-\cos\psi_c)/(1-\cos r_i)$ and $1-(1-\cos\psi_c)/(1-\cos r_i)$, respectively, thus we obtain the following approximation:

$$a_s(\Delta\omega_i', \vec{\omega}) \approx k_s \cdot \frac{(1-\cos^{n+1} r_i)\cdot(\cos\psi_c - \cos r_i)}{(n+1)\cdot(1-\cos r_i)^2} + k_s \cdot \cos^n \psi_c \cdot \frac{1-\cos\psi_c}{1-\cos r_i}.$$



$n = 10$ $n = 100$

Fig. 6. Comparison of the proposed average reflectance approximation with that of obtained with a single sample as functions of $\psi_c \in [0, r]$ for materials of different shininess. The size of the Voronoi cell is $r = \pi/8$.

The average reflectance provided by this formula is compared to the average reflectance obtained with a single sample in figure 6. Note that the incorrect high peaks of highly specular materials at $\psi_c = 0$ are eliminated by the new reflectance formula.

Note that this formula is quite cheap computationally. During preprocessing we associate a spherical circle with each Voronoi cell and store the cosine of its angle ($\cos r_i$) together with center direction $\vec{\omega}_i$. When the given cell is processed for mirror direction $\vec{\omega}_R$, the scalar product $\cos\psi_c = \vec{\omega}_i \cdot \vec{\omega}_R$ is obtained and $\cos r_i$ is taken from the stored parameters.

3. Results

The algorithm has been implemented on an NV6800GT graphics card. With Shader Model 3.0 multiple directional samples could be handled in a single pass. This also implies more texture reads, but with tiling depth maps into bigger textures, the limitation on active samplers can also be avoided. The rendering times and the rendered images are shown in table 1 and in figure 8, respectively. Note that we could achieve interactive rates with a high number of samples, delivering artifact-free animation.

Scene	Sample points	Frames / sec
Armadillo	64	33 Fps
Armadillo	128	16.8 Fps
Two Armadillos	64	17.8 Fps
Two Armadillos	128	7.5 Fps

Tab. 1. Rendering times for the scene of Armadillos visiting Florence. A single Armadillo character consists of 15000 triangles.

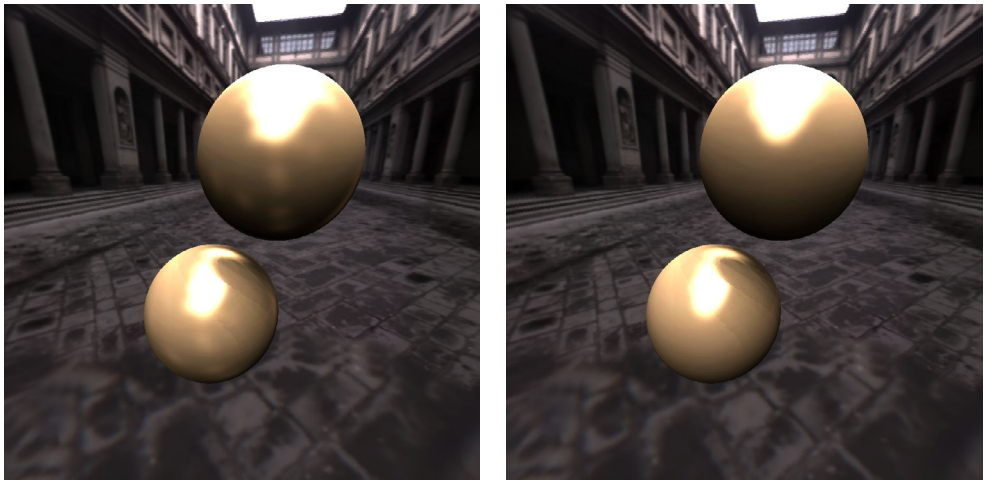


Fig. 7. Spheres in Florence rendered with taking a single sample in each Voronoi cell, assuming directional light sources (left) and using our new formula for average reflectance over Voronoi cells (right). The shininess of the spheres is 100.

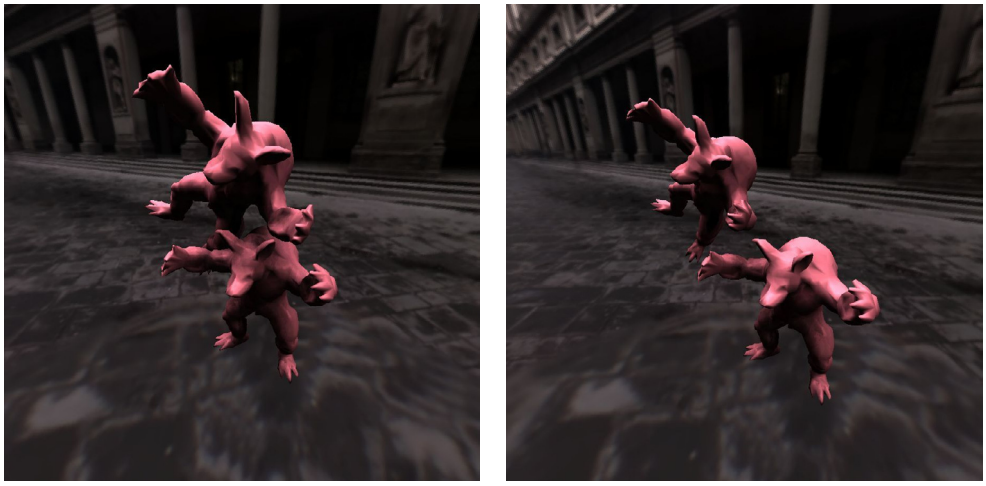


Fig. 8. Two images from a video rendered on 18 FPS. Observe how the illumination of the standing Armadillo changes when the other Armadillo flies over.

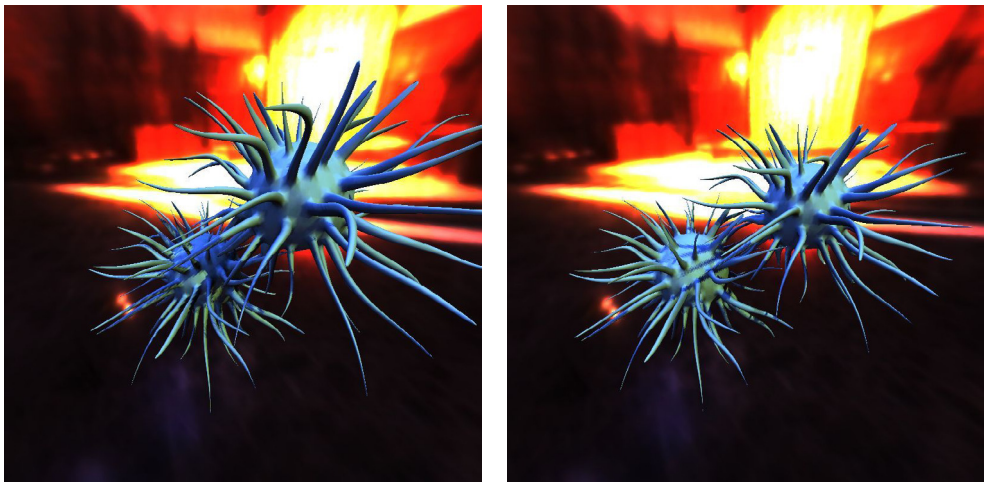


Fig. 9. Viruses in the Irish pub scene (30 FPS)

4. Conclusions

This paper proposed a real-time environment lighting algorithm that generates shadows and works well even for specular surfaces. With a low number of directional samples (32), slight shadow patterns were observable when the objects were moving, but the frame rates were more than enough for interactive rendering. A higher number of samples, producing fine results for animations, are more demanding, but still can provide real-time frame rates.

5. Acknowledgements

This work has been supported by OTKA (T042735) and GameTools FP6 (IST-2-004363) project.

References

1976
[1] J. F. Blinn and M. E. Newell. Texture and reflection in computer generated images. Communications of the ACM, 19(10):542–547.

1978
[2] L. Williams. Casting curved shadows on curved surfaces. In Computer Graphics (SIGGRAPH '78 Proceedings), pages 270–274.

1982
[3] S. Lloyd. Least square quantization in PCM. IEEE Transactions on Information Theory, 28:129–137.

1984
[4] N. Greene. Environment mapping and other applications of world projections. IEEE Computer Graphics and Applications, 6(11):21–29.

1994
[5] E. Reinhard, L. U. Tijssen, and W. Jansen. Environment mapping for efficient sampling of the diffuse interreflection. In Photorealistic Rendering Techniques, pages 410–422. Springer.

1998
[6] P. Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In SIGGRAPH '98, pages 189–198.

2001
[7] A. Wilkie. Photon Tracing for Complex Environments. PhD thesis, Institute of Computer Graphics, Vienna University of Technology.

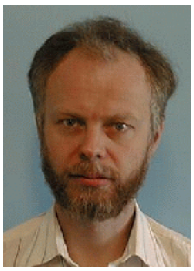
2003
[8] T. Kollig and A. Keller. Efficient illumination by high dynamic range images. In Eurographics Symposium on Rendering, pages 45–51.

2004
[9] Ch. Mei, J. Shi, and F. Wu. Rendering with spherical radiance transport maps. Computer Graphics Forum (Eurographics 04), 23(3):281–290.
[10] V. Ostromoukhov, C. Donohue, and P-M. Jodoin. Fast hierarchical importance sampling with blue noise properties. In Proc. SIGGRAPH 2004.

[11] L. Szécsi, M. Sbert, and L. Szirmay-Kalos. Combined correlated and importance sampling in direct light source computation and environment mapping. Computer Graphics Forum (Eurographics 04), 23(3):585–604.

Attila Barsi is the Ph.D. student of the Computer Graphics Lab at the Budapest University of Technology and Economics. His research interests include real-time graphics, the application of GPU for global illumination computations, and Monte Carlo methods.

Lászlo Szécsi is an assistant professor of the Department of Control Engineering and Information Technology of the Budapest University of Technology and Economics. He works on sampling and rendering algorithms to solve global illumination and realistic lighting problems.



László Szirmay-Kalos received Ph.D. and Doctor of Science degrees from the Hungarian Academy of Sciences in 1992 and in 2000, respectively. Currently he works as a full professor at the Department of Control Engineering and Information Technology, where he heads the Computer Graphics Lab. His research interests include real-time global illumination rendering, Monte Carlo techniques, and GPGPU methods. He is the author of more than a hundred papers and five books.