INTERACTIVE GLOBAL ILLUMINATION WITH VIRTUAL LIGHT SOURCES

Thesis points of PhD dissertation

LÁSZLÓ SZÉCSI

Supervisor:
LÁSZLÓ SZIRMAY-KALOS, PhD

Budapest University of Technology And Economics
Faculty of Electrical Engineering and Informatics
Department of Control Engineering and Information Technology

Budapest

June 18, 2009

Synopsis

Research area and research objectives

The basic problem of the image synthesis branch within computer graphics is to compute an image of a virtual world that delivers the impression of viewing the real world. In order to produce the visual stimuli identical to what light reflected from real world objects would induce, light power arriving at the eye from solid angles corresponding to image elements has to be computed. Global illumination algorithms ambition to do this by evaluating light transport for arbitrary surface models and multiple reflections.

The virtual light sources method, or indirect photon mapping, is a global illumination algorithm which combines the advantages and capabilities of shooting and gathering type random walk algorithms. It also traces the illumination problem back to lighting by point-like, abstract light sources, which is a local illumination task strongly supported by hardware.

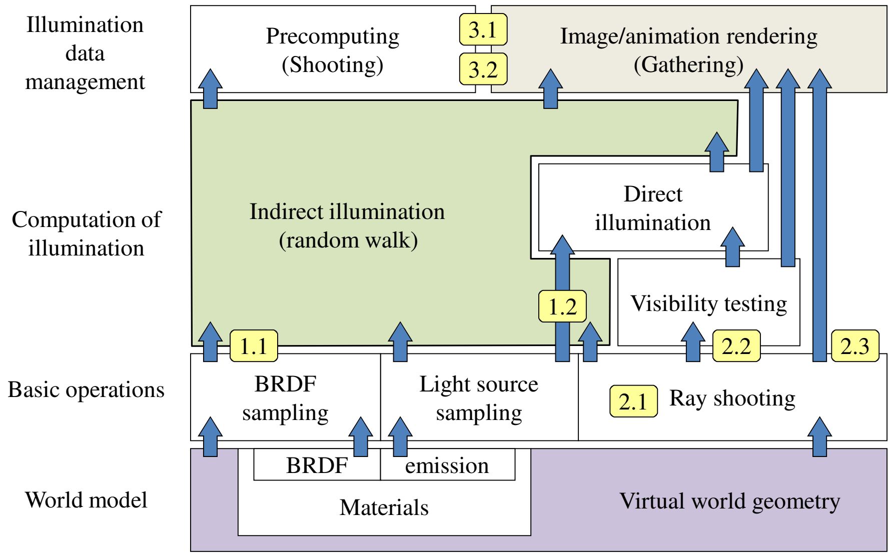

Figure 1: Modular model of image synthesis with virtual light sources and the contributions of the theses.

The algorithm has two phases. First, a Monte Carlo random walk photon shooting phase

[J5] L. Szécsi, K. Ralovich. Loose kd-trees on the GPU. IV. Hungarian Conference on Computer Graphics and Geometry, Budapest, Hungary, pp 94-101. 2007.

[J6] L. Szécsi, L. Szirmay-Kalos, P. Anton. Environment mapping with halftoning 7th Conference of the Hungarian Association for Image Processing and Pattern Recognition. Budapest, Hungary, pp. 1-9. 2009.

[H1] L. Szirmay-Kalos, L. Szécsi, M. Sbert. GPUGI: Global Illumination Effects on the GPU. Eurographics Tutorial, 2006. Hivatkozsok: 4.

[F6] **L. Szirmay-Kalos, T. Umenhoffer, B. Tóth, L. Szécsi, M. Sbert. Volumetric Ambient Occlusion. IEEE Computer Graphics and Applications, pp. 1-13. 2009.** IF: 1.398

[F7] L. Szirmay-Kalos, T. Umenhoffer, G. Patow, L. Szécsi, M. Sbert. Specular Effects on the GPU: State of the Art. Computer Graphics Forum, 26:1 pp. 1-24. 2009. IF: 1.107

[F8] L. Szirmay-Kalos and L. Szécsi. Deterministic Importance Sampling with Error Diffusion. Computer Graphics Forum, 28:4 pp. 1-11. 2009. IF: 0.801

[I1] **L. Szécsi, L. Szirmay-Kalos, M. Sbert. Light Animation with Precomputed Light Paths on the GPU. Graphics Interface 2006, Quebeck, Canada. pp 187-194. 2006. Hivatkozsok: 4.**

[I2] **L. Szécsi. The hierarchical ray engine. Proceedings of WSCG (Full papers), 2006.**

[I3] Sz. Czuczor, L. Szirmay-Kalos, L. Szécsi. Photon map gathering on the GPU. Proceedings of Eurographics (short papers), pp 117-120. 2005. Hivatkozsok: 1.

[I4] A. Barsi, L. Szécsi, L. Szirmay-Kalos. Real-time Image Based Lighting with Shadows. HACIPPR Conference, Veszprm, 2005.

[I5] L. Szécsi, L. Szirmay-Kalos. Improved Indirect Photon Mapping with Weighted Importance Sampling. Proceedings of Eurographics (short paper), pp 45-52. 2003.

[I6] L. Szécsi, B. Benedek. Accelerating Animation Through Verification of Shooting Walks. Spring Conference on Computer Graphics, Budmerice, pp 255-261. 2003. Hivatkozsok: 3.

[I7] **L. Szirmay-Kalos, V. Havran, B. Benedek, L. Szécsi. On the Efficiency of Ray-shooting Acceleration Schemes. Spring Conference on Computer Graphics, Budmerice, pp 255-261. 2003. Hivatkozsok: 21.**

[I8] **T. Umenhoffer, L. Szirmay-Kalos, L. Szécsi, B. Tóth, M. Sbert. Partial, Multi-scale Precomputed Radiance Transfer. Spring Conference on Computer Graphics, Budmerice, pp. 87-94. 2008.**

[I9] L. Szirmay-Kalos, L. Szécsi, A. Penzov. Importance Sampling with Floyd-Steinberg Halftoning. Proceedings of Eurographics (short papers), pp 69-72. 2008.

[J1] **L. Szécsi. Conservative rasterization of texture atlases. III. Hungarian Conference on Computer Graphics and Geometry, Budapest, Hungary, pp 79-85. 2005.**

[J2] **L. Szécsi, B. Benedek. Improvements on the kd-tree. I. Hungarian Conference on Computer Graphics and Geometry, Budapest, Hungary, pp 165-172. 2002.**

[J3] B. Benedek, L. Szécsi. Performance Improvements of Rendering Caustics using Photon Maps in Interactive Ray Tracing. I. Hungarian Conference on Computer Graphics and Geometry, Budapest, Hungary, pp 207-211. 2002.

[J4] L. Szécsi. Procedural Ocean Waves. IV. Hungarian Conference on Computer Graphics and Geometry, Budapest, Hungary, pp 80-87. 2007.

generates the virtual light source representation of scene radiance, then the virtual light sources are used to illuminate the rendered scene.

A random walk is a sample of the recursive integrand in the rendering equation:

$$L(\vec{x},\omega) = \int_{\Omega} f_r(\omega,\vec{x},\omega') \cdot \cos\theta' \cdot L\left(h(\vec{x},-\omega'),\omega'\right)\ d\omega',$$

where L is the radiance function to be computed, Ω is the set of all directions, $h(\vec{x},-\omega'),\omega')$ is the surface point visible from $\vec{x}$ at direction $-\omega'$, and θ' is the angle between the ω' direction and the surface normal. $f_r(\omega,\vec{x},\omega')\cdot\cos\theta'$ is the probability distribution function of a photon arriving at $\vec{x}$ from ω being reflected to ω'. f_r is the bidirectional reflection distribution function, or BRDF, which describes surface reflection properties.

In order to avoid the dimensional explosion of classical quadrature rules, Monte Carlo integration can be applied for the evaluation of the integral. This turns the integral into an expected value, which can be approximated as the average of a finite number of samples:

$$I = \int_{\mathcal{U}} f(\mathbf{u})\ \mathrm{d}\mathbf{u} = \int_{\mathcal{U}} \frac{f(\mathbf{u})}{p(\mathbf{u})} \cdot p(\mathbf{u})\ \mathrm{d}\mathbf{u} = E\left[\frac{f(\mathbf{u})}{p(\mathbf{u})}\right] \approx \frac{1}{M}\sum_{j=1}^{M}\frac{f(\mathbf{u}_j)}{p(\mathbf{u}_j)},$$

wherel $\mathbf{u} = [u^{(1)},\ldots,u^{(d)}]$ is the integration variable, $p(\mathbf{u})$ is a probability density function in the d-dimensional $\mathcal{U}$ integration domain, and $\mathbf{u}_1,\ldots,\mathbf{u}_M$ points are chosen randomly with this distribution. As importance sampling theory states, the probability distribution of random samples has to mimic the integrand in order to get a low-variance estimator. **Thesis group 1** addresses how the variance of relevant Monte Carlo estimators can be decreased.

The basis of importance sampling of the directional domain is the BRDF, the part of the integrand that can be locally evaluated. Practical BRDFs are composed as a sum of several functions corresponding to diffuse, specular, reflective, refractive or more complex light transport effects. Russian roulette is applied to make random walks finite, attributing some probability to not continuing the walk from a surface point. However, it introduces additional variance by sampling the zero-contribution domain.

In order to generate random walks, rays along sampled directions have to be cast, finding the nearest intersection with scene objects. Furthermore, during the second, gathering phase, when shading a surface point we need to determine the visibility of all virtual light sources, which is done by casting shadow rays. **Thesis group 2** offers solutions for both ray shooting problems.

In order to have a light path of non-zero contribution, the eye and a light source must be connected. Random walks do not guarantee this. A deterministic step to connect a shooting path to the eye, or a gathering path to light sources is needed. Area light sources need to be sampled to compute this direct illumination of surface points. If light sources are huge, which is evident in image-based lighting and sky lighting, the variance of the estimator is also going to be large. However, there is an analytical formula for the illumination of polygonal light sources, disregarding occlusions.

The performance bottleneck of the virtual light sources method is the final gathering part, where surface elements in the final image have to be shaded as illuminated by a large number of abstract light sources. A visibility test equivalent to casting a shadow ray has to be performed to see whether a light source illuminates a surface point. With millions of pixels and hundreds of light sources this means billions of shadow rays. Tracing them against the complete scene geometry is not affordable, but given the low frequency of indirect illumination, an approximate solution is appropriate.

Ray shooting is accelerated by recursively subdividing space into cells, identifying cells a ray passes through, and performing the ray-primitive intersection with primitive objects within those cells. The most popular subdivision schemes are the uniform grid, the octree, the kd-tree and the bounding volume hierarchy.

Graphics hardware has been built for incremental image synthesis. Ray shooting, and especially random walk algorithms are better suited to pointer-chasing CPUs, where incoherent random access to all scene elements is possible. However, the evolution of programmable graphics processing units has made it possible to implement ray-tracing on the hardware. The different architecture and some limitations mean that CPU algorithms can not simply be ported. The ray engine solution maps rays to texels and renders ray-casting primitives as full-screen quadliraterals to intersect every ray with every primitive in pixel shaders. The CPU is required to delegate rays and read back results.

No matter how many rays we are able to trace in a second, it will never be comparable to the number of actual photons in nature, and will always remain a limiting factor. However, there is a huge amount of coherence between light paths in all settings and problems, exploiting which is the key factor to get to interactive rendering times. Thus, we have to store and reuse information of previously traced rays. These methods tend to introduce a visually welcome feeling of smoothness instead of random noise, by transferring error from the high-frequency domain to the less disturbing low-frequency one. Reusing rays is always accomplished by storing random walks and recombining them into complete light paths, preferrably still trying to retain importance sampling properties. Virtual light sources themselves are nothing else but stored shooting light paths.

In **Thesis group 3**, I describe two algorithms utilizing ray reuse, which use the virtual light sources method to precompute radiance data and store it in textures. One of them allows light animation, where information from previous frames is re-weighted and re-used to get a global illumination solution of the current frame in a fraction of the time required to render a still image. Finally, the precomputed light paths algorithm allows moving both the camera and the lights offering real-time indirect illumination.

Applied methods

My work is based on extensive literature in Monte Carlo sampling theory, global illumination, ray reuse algorithms and stream processing. These fields are vertically connected to form real-time global illumination solutions. I have achieved new scientific results in all of these areas, partly by arriving at new theoretical finding, partly by adapting existing methods to the virtual light sources context. All of the results were verified by measurements. In the context of computer graphics, this means I have implemented new algorithms, using the graphics hardware where required and applicable. I have compared results with those predicted by theory, given by literature or delivered by implementations of previous algorithms.

List of publications

[B1] L. Szirmay-Kalos, L. Szécsi, M. Sbert. GPU-based Techniques for Global Illumination Effects. Morgan & Claypool, San Francisco, USA, 252 pps. 2008. Hivatkozsok: 2.

[D1] L. Szécsi. An Effective Implementation of the K-D Tree. in Graphics Programming Methods (editor: Jeff Lander), Charles River Media, Hingham, Massachusetts, pp 315-325, 2003. Hivatkozsok: 5.

[D2] L. Szécsi. Alias-free Hard Shadows with Geometry Maps. in ShaderX^5: Advanced Rendering Techniques (editor: Wolfgang Engel), Charles River Media, Hingham, Massachusetts, pp 219-237, 2007.

[D3] L. Szécsi, K. Arman. Procedural Ocean Effects. in ShaderX^6: Advanced Rendering Techniques (editor: Wolfgang Engel), Charles River Media, Hingham, Massachusetts, pp 331-350, 2008.

[D4] L. Szécsi, L. Szirmay-Kalos, M. Sbert. Interactive Global Illumination with Precomputed Radiance Maps. in ShaderX^6: Advanced Rendering Techniques (editor: Wolfgang Engel), Charles River Media, Hingham, Massachusetts, pp 401-410, 2008.

[D5] L. Szécsi. Instant Radiosity with GPU Photon Tracing and Approximate Indirect Shadows. in ShaderX^7: Advanced Rendering Techniques (editor: Wolfgang Engel), Charles River Media, Hingham, Massachusetts, pp 479-494, 2009.

[F1] A. Barsi, L. Szirmay-Kalos, L. Szécsi. Image-based Illumination on the GPU. Machine Graphics and Vision, Vol 14., No 2., pp 159-169, 2006. Hivatkozsok: 3.

[F2] M. Sbert, L. Szécsi, L. Szirmay-Kalos. Real-time Light Animation. Computer Graphics Forum, Vol 23., No 3., pp 291-299, 2004. Hivatkozsok: 8. IF: 0.801

[F3] L. Szécsi, M. Sbert, L. Szirmay-Kalos. Combined Correlated and Importance Sampling in Direct Light Source Computation and Environment Mapping. Computer Graphics Forum, Vol 23., No 3., pp 585-593, 2004. Hivatkozsok: 12. IF: 0.801

[F4] L. Szécsi, L. Szirmay-Kalos. Efficient Approximate Visibility Testing Using Occluding Spheres. Journal of WSCG, Vol 12., No 3., pp 435-442, 2004.

[F5] L. Szécsi, L. Szirmay-Kalos, Cs. Kelemen. Variance Reduction for Russian Roulette. Journal of WSCG, Vol 11. No 3., pp 456-463, 2003.

Thesis 3.2 *Dynamic indirect lighting with precomputed light paths*

I have introduced the concept of light path maps, where pre-generated light paths connect entry points to reference points, and the color appearing at a reference points due to unit irradiance arriving at an entry point is stored in a Precomputed Radiance Map. The computation of the LPM uses the virtual light sources method itself. When rendering the scene, LPM entries can be combined according the to actual lighting, yielding indirect illumination results at reference points. For surface points between reference points, the results are interpolated. The algorithm works on the GPU, offering real-time indirect illumination. [I1, J1, I8, D4, B1]

New scientific results

Thesis Group 1. *Sampling in random walk algorithms*

Thesis 1.1 *Variance reduction and spectral optimization for Russian roulette and combined BRDF sampling*

The Russian roulette technique introduces additional variance into the Monte-Carlo radiance estimator as it violates importance sampling by assigning a non-zero probability to walk termination, which equals to sampling the integrand where it has zero contribution. I have shown that this effect can be reduced by a modified estimator, which is still unbiased, but assigns a non-zero contribution to termination.

I have composed and analyzed different random estimators according to how we use the individual probability densities of BRDFS and the contributions they provide for a given sample.

A random walk light path is assumed to transport light on all representative wavelengths. I have proved that both Russian roulette and BRDF selection can be improved by evaluating selection probabilities separately for all wavelengths, and averaging the results. When colored light would be reflected according to a disjunctly colored BRDF, traversing light rays of little contribution can be avoided. [F5]

Thesis 1.2 *Combination of correlated and importance sampling*

For unoccluded area light sources and for environment lighting, illumination can be expressed using an analytic formula. Using correlated sampling, only the difference of the actual lighting and that predicted by the formula has to be integrated. This eliminates all noise not due to occlusions, but introduces more noise in occluded areas. I have proposed a sampling scheme that combines correlated and importance sampling using a weight computed from the samples, resulting in an estimator with lower variance in the whole domain. [F3]

Thesis Group 2. *Ray shooting acceleration*

Thesis 2.1 *Data structures for fast ray shooting*

I have investigated the classical, space subdivision ray shooting solutions, and proposed improvements that take the memory and cache characteristics of available hardware into account. I have proposed a new representation for triangle meshes, which allows processing units to read a minimal amount of data, with intersection computation outperforming previous methods. I have also proposed representations for the building and traversing of space

Figure 2: Reduced variance for environment mapping using correlated sampling.

subdivision schemes, in particular for kd-trees and bounding interval hierarchies. These fit CPU cache lines and GPU textures. [I7, D1, J2, J5, H1, D5, B1]

Thesis 2.2 *Approximate visibility testing with occluding spheres*

I have proposed an approximate visibility testing algorithm most effective for a large number of point-like light sources. Triangle mesh geometry is converted into a set of spheres using Delaunay tetrahedralization. Then a novel merging algorithm is applied to decrease the number of spheres while keeping general shadowing capacity, for which a metric based on ray hit probabilities is introduced. Interpreting the spheres as occluding discs when shading surface point, visibility of indirect light sources can be tested extremely efficiently. [F4]

Figure 3: Sets of spherical occluders

Thesis 2.3 *Ray tracing with a ray hierarchy*

I have proposed a specialized, accelerated approach based on the ray engine. As opposed to the original scheme, which intersects every primitive with every ray, the new algorithm builds an acceleration hierarchy on the ray domain, making use of coherence. [I2]

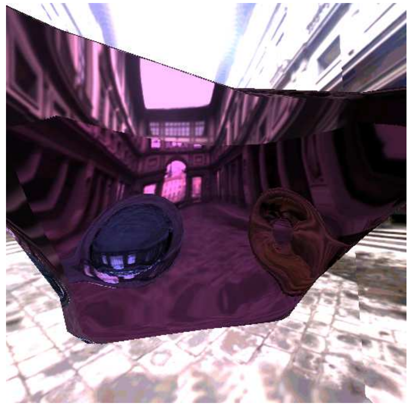

Figure 4: Refractive objects rendered real-time with the hierarchical ray engine.

Thesis Group 3. *Ray reuse with virtual light sources*

Thesis 3.1 *Real-time light animation with path reuse*

I have proposed a solution for an important special case of animation, when the light sources are small and moving but other objects and the camera remain still. In this case the unchanged parts of light paths can be reused in different frames. In order to store light paths and compute the image from them, virtual light sources are applied. When rendering a specific frame, contributions in other frames are re-weighted according to the actual light position. As a result, a single frame is rendered in a fraction of time required to render an individual still image, making fast animation possible. [F2]

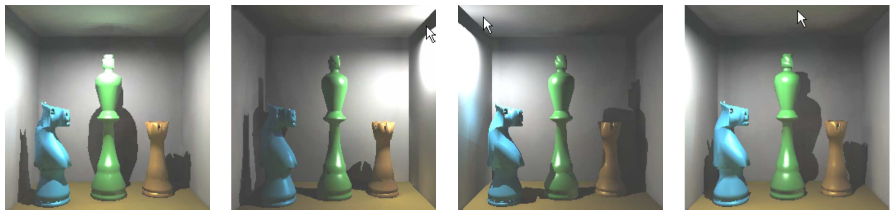

Figure 5: Images captured during interactive positioning of a spotlight.