Real-time Shading with Filtered Importance Sampling

Mark Colbert[*]
University of Central Florida

Jaroslav Křivánek[†]
Czech Technical University in Prague

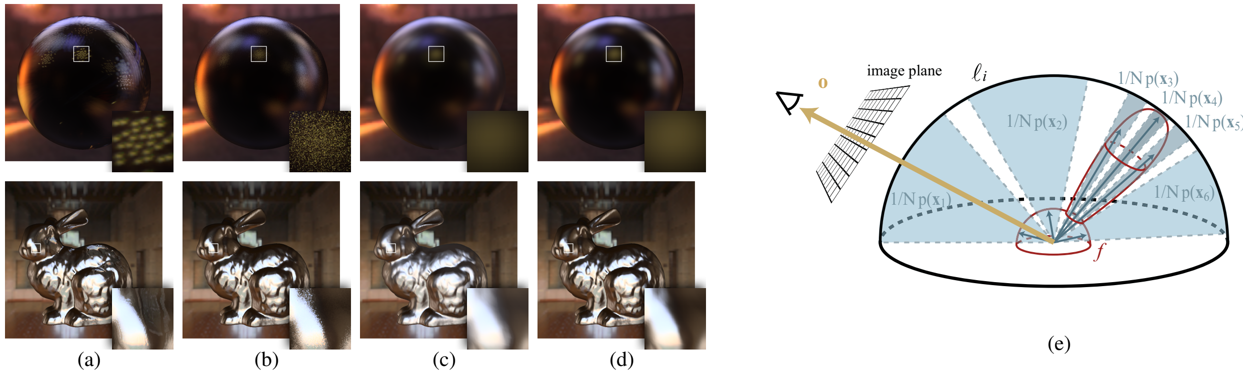

Figure 1: *(a) Rendering using deterministic importance sampling without filtering produces aliasing in the estimate of the illumination integral, which translates into image artifacts. (b) Randomizing the directions for each image pixel trades alias for visible noise. (c) Our filtered importance sampling suppresses the alias at the cost of slightly blurred highlights. (d) Reference solution. Images (a), (b), and (c) are rendered with 40 importance samples per pixel. (e) Illustration of the size of the filter used on the environment map* ℓ_i.

1 Introduction

Correct perception of materials requires complex, natural illumination [Fleming et al. 2003]. Thus for material or lighting design applications, realistic, interactive rendering of objects with arbitrary materials under natural illumination is essential. We present a simple and efficient technique for real-time, image-based lighting of objects with spatially-varying, glossy materials. The key to our algorithm is combining BRDF-proportional *importance sampling* with *environment map filtering* to attain computationally efficient rendering amenable to the GPU.

Environment map pre-filtering [Kautz et al. 2000] and some frequency space solutions provide real-time visualizations, but their use may be too cumbersome since it requires a hefty amount of precomputation and/or a sizable code base. Our filtered importance sampling requires minimal pre-computation and operates within a GPU shader, thereby fitting into almost any production pipeline needing real-time dynamically changing materials or lighting.

2 Filtered Importance Sampling

The color of one pixel is given by the illumination integral, i.e. the integral of the lighting and BRDF product. Our approach to evaluating this integral on the GPU is motivated by BRDF-proportional Monte Carlo importance sampling. For each pixel, we take uniformly distributed numbers, transform them in the GPU shader into important sample directions, and evaluate the samples. Here, the uniformly distributed numbers are a precomputed deterministic set used for all pixels. This results in aliasing in our estimate of the illumination integral causing image artifacts (Figure 1a). Monte Carlo would use random numbers to trade aliasing for more visually acceptable noise (Figure 1b). Since generating random numbers on the GPU is expensive, we reduce the artifacts via a filtering operation (Figure 1c).

Our main contribution is determining the appropriate filter size for each importance sample. Intuitively, the filter size should be small for densely sampled directions around the BRDF lobe and larger elsewhere (Figure 1e). This suggests that filter size is inversely proportional to the *probability density function* (PDF) associated with the sample direction. Performing a formal analysis of numerical integration in the frequency domain supports this intuition. We use hardware accelerated mipmaps for fast environment map filtering with a mipmap level determined from the PDF-proportionally sized filter:

$$l = \max\left\{\log_4\left(\frac{w\cdot h}{N}\frac{1}{p(\mathbf{x}_i)}\right), 0\right\},$$

where, N is the number of samples, w and h are the width and height the environment map, and p is the PDF. A formal analysis of filtered importance sampling exposes the approximations used in the described implementation. Notably, the isotropic filter, with its size inversely proportional to the PDF, is a crude approximation of the close-to-ideal, spatially variant, anisotropic filter. However, anisotropic filtering with mipmapping is not possible due to current hardware limitations. By performing the mipmap look-ups via the aforementioned technique, we achieve a good performance at the cost of slight blurring.

3 Conclusion

Our novel shading technique, filtered importance sampling, provides real-time rendering for dynamically changing illumination and spatially-varying BRDFs. The algorithm runs at about 350 frames per second on an NVIDIA 8800GTX, when rendering a sphere at a resolution of 512x512 with 40 samples per pixel. However, the true strength of the algorithm is its simplicity and flexibility–it fits in a single GPU shader, it handles any BRDF model that can be importance sampled, and has modest precomputation and memory requirements (a mipmap of the environment).

To support diffuse surfaces, we use low-order spherical harmonics [Ramamoorthi and Hanrahan 2001]. Our work currently ignores self-occlusion, and we leave this for future work. However, we provide the necessary adaptability for interactive material and lighting design within any production pipeline.

References

FLEMING, R. W., DROR, R. O., AND ADELSON, E. H. 2003. Real-world illumination and the perception of surface reflectance properties. *Journal of Vision 3* (July), 347–368.

KAUTZ, J., VÁZQUEZ, P.-P., HEIDRICH, W., AND SEIDEL, H.-P. 2000. A unified approach to prefiltered environment maps. In *11th Eurographics Workshop on Rendering*, Eurographics Association, 185–196.

RAMAMOORTHI, R., AND HANRAHAN, P. 2001. An efficient representation for irradiance environment maps. In *Proc. of SIGGRAPH 2001*, ACM Press.

[*]colbert@cs.ucf.edu
[†]xkrivanj@fel.cvut.cz