Interactive Monte-Carlo Ray-Tracing Upsampling

Malik Boughida   Thibault Groueix   Tamy Boubekeur

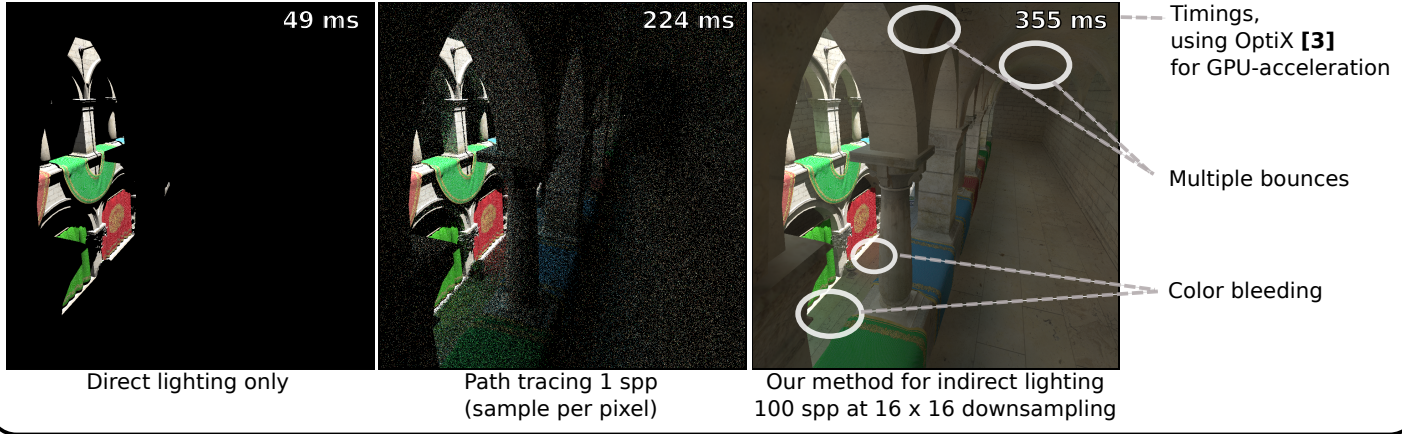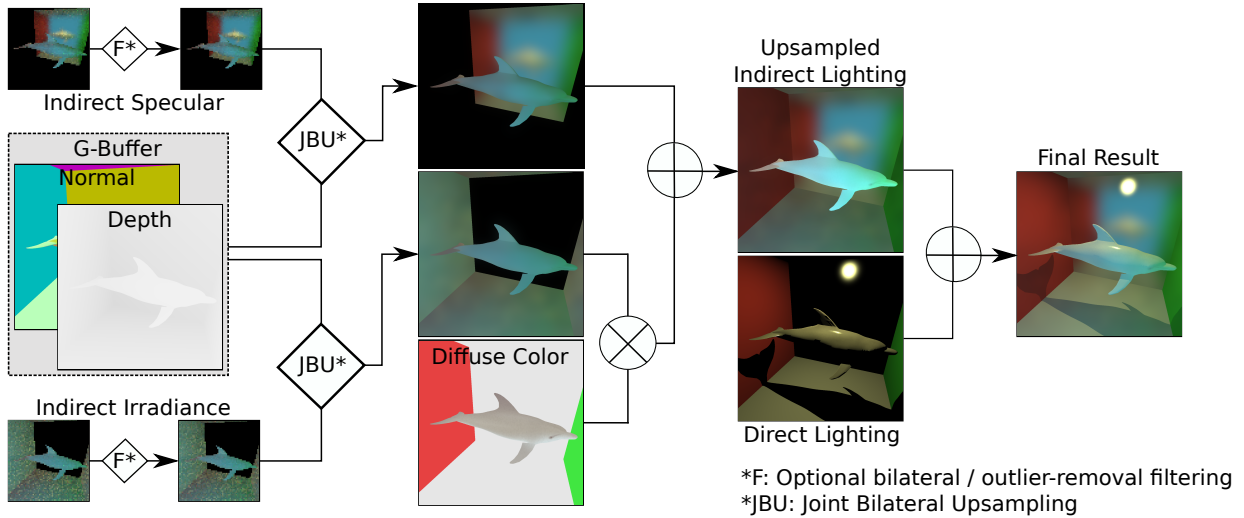*LTCI, CNRS, Telecom ParisTech, Paris-Saclay University*

Abstract

We propose a practical method to approximate global illumination at interactive framerates for dynamic scenes. We address multi-bounce, visibility-aware indirect lighting, for diffuse to moderately glossy materials, relying on GPU-accelerated ray-tracing for this purpose. While Monte-Carlo ray-tracing algorithms offer unbiased results, they produce images which are, under interactive constraints, extremely noisy, even with GPU acceleration. Unfortunately, filtering them to reach visual appeal induces a large kernel, which is not compatible with interactive framerate. We address this problem using a simple downsampling approach. First, we trace indirect paths on a uniformly distributed subset of pixels, decorrelating diffuse and specular components of lighting. Then, we perform a joint bilateral upsampling on both components, taking inspiration from deferred shading by driving this upsampling with a full-resolution G-Buffer. Our solution provides smooth results, does not require any pre-computations, and is both easy to implement and flexible, as it can be used with any generation strategy for indirect rays.
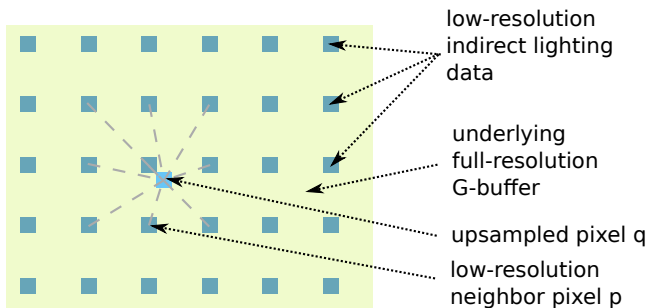
Overview

Timings, using OptiX **[3]** for GPU-acceleration

Multiple bounces

Color bleeding

Direct lighting only

Path tracing 1 spp (sample per pixel)

Our method for indirect lighting 100 spp at 16 x 16 downsampling

Pipeline

Indirect Specular

G-Buffer

Normal

Depth

JBU*

JBU*

Indirect Irradiance

Diffuse Color

Upsampled Indirect Lighting

Final Result

Direct Lighting

*F: Optional bilateral / outlier-removal filtering
*JBU: Joint Bilateral Upsampling

Joint Bilateral Upsampling [2]

low-resolution indirect lighting data

underlying full-resolution G-buffer

upsampled pixel q

low-resolution neighbor pixel p

$$jbu[q] = \frac{\sum_p w_{p,q} input[p]}{\sum_p w_{p,q}}$$

$$w_{p,q} = exp[-(\lambda^{pix} d^{pix}_{p,q} + \lambda^z d^z_{p,q} + \lambda^{normal} d^{normal}_{p,q})]$$

$$d^{pix}_{p,q} = ||p - q||^2$$

$$d^z_{p,q} = (\frac{z[p]}{z[q]} - 1)^2$$

$$d^{normal}_{p,q} = ||normal[p] - normal[q]||^2$$

Separating Irradiance

Rendering Equation **[1]**: $L_o(\omega_o) = \int_\Omega L_i(\omega_i) f(\omega_i, \omega_o)(n \cdot \omega_i) d\omega_i$

$$L_i = L_i^{dir} + L_i^{ind} \longrightarrow L_o = L_o^{dir} + L_o^{ind}$$

$$f = \frac{k_d}{\pi} + f^{spec} \longrightarrow L_o^{ind} = \frac{k_d}{\pi} E^{ind} + L_o^{ind,spec}$$

lambertian + specular

diffuse albedo (**high** freq.)

indirect irradiance (**low** freq.)

indirect spec. (**medium** freq.)

References

[1] The rendering equation. *J. T. Kajiya.* Proc. SIGGRAPH 20, 4 (1986), 143–150. 2

[2] Joint bilateral upsampling. *J. Kopf, M. F. Cohen, D. Lischinski, M. Uyttendaele.* ACM Trans. Graph. 26, 3 (July 2007). 1

[3] Optix: A general purpose ray tracing engine. *S. G. Parker, J. Bigler, A. Dietrich, H. Friedrich, J. Hoberock, D. Luebke, D. McAllister, M. McGuire, K. Morley, A. Robison, M. Stich.* ACM Trans. Graph. 29, 4 (July 2010), 66:1–66:13. 1