

Interactive Ray-Traced Area Lighting with Adaptive Polynomial Filtering

Jose A. Iglesias-Guitian
jalley@disneyresearch.com

Bochang Moon
bochang.moon@disneyresearch.com

Kenny Mitchell
kenny.mitchell@disneyresearch.com

Disney Research,
The Walt Disney Company

Area lighting computation is a key component for synthesizing photo-realistic rendered images, and it simulates plausible soft shadows by considering geometric relationships between area lights and three-dimensional scenes, in some cases even accounting for physically-based material properties [2]. In rendering, computing visibility on a surface point can be naturally performed by a ray casting process, e.g., processing shadow rays. However, this process typically requires a large number of shadow rays to accurately simulate penumbra regions. Otherwise, the rendered shadows can be often corrupted by a high variance, i.e., noise. This makes fully ray traced approaches non-practical under real-time constraint scenarios [1].

To tackle this problem, while achieving high-quality shadows interactively, we propose a filtering based visibility computation framework. Our method employs ray casting to accurately compute the visibility value of each sample, but we use only small sample counts to lower the sampling time required. This visibility image, generated by a small number of samples, is corrupted by high-frequency noise especially in penumbra areas. Instead of increasing the sample count, we apply our new filtering to reduce the noise and enable a high-quality rendering with soft shadows.

In this short paper we present an efficient hybrid area light computation framework that uses ray casting with a small number of shadow samples to get smooth visibility maps using an adaptive post-processing filter. In the end the result of direct lighting computed using the rasterization pipeline is combined with the filtered ray-traced visibility to create the final images. Next, we provide the overview of our rendering framework devised to accelerate the visibility computations, and describe a new post-processing filter for producing high-quality soft shadows. In addition, we demonstrate that our method generates high-quality rendering results interactively, guided by our area lighting computation that runs in real-time.

Hybrid Rendering Framework

We have implemented interactive area lighting effects using a hybrid GPU rendering framework. Our proposed rendering system is built on top of an OpenGL deferred shading rasterization pipeline, encompassing a visibility pass that leverage on a GPU ray tracing engine (i.e. NVidia OptiX) to gather visibility samples. Since the number of samples required to get a noise free visibility map easily exceed the available computation time for interactive scenarios, we propose to employ a lower number of visibility samples, and filter this approximate result to get a nearly noise free visibility.

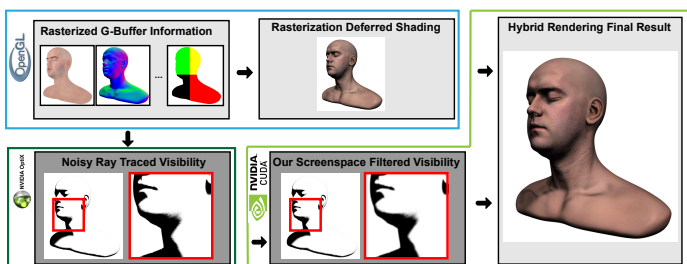


Figure 1: Our GPU Hybrid Rendering Framework Overview. A classic GPU deferred shading rasterization pipeline (OpenGL) cooperates with a GPU ray tracer program (i.e. using NVidia OptiX), in charge of gathering in real-time a cheap amount of visibility samples. Our visibility filter, implemented in CUDA, is capable of reducing the noise in the visibility buffer and makes possible to generate high-quality rendered results.

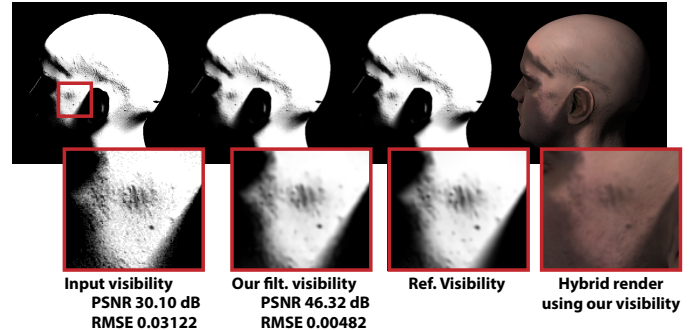


Figure 2: Filtering results for soft shadows. Our filtering output significantly reduces the Monte Carlo noise by just using pixel positions as a denoising feature buffer.

Visibility Filtering Algorithm

Our approach employs real-time filtering to reduce high-frequency noise in the visibility buffer. The goal of this filter is to approximate the ground truth visibility function that can be only computed with an infinite number of samples. In our real-time rendering scenario, allocated samples per pixel are typically a very small number (e.g., 1 to 4), and thus generated visibility functions are often very noisy especially when the shadow functions are generated from large area lights.

We adapt a recent filtering method that adaptively controls polynomial order to optimally fit our visibility function to the unknown ground truth visibility [4]. The previous method demonstrated high-quality filtering results for images rendered by Monte Carlo ray tracing. The main idea was to approximate the unknown image locally with adaptively chosen polynomial functions. Especially, their underlying functions were formed by using geometric features such as normals, textures, and depths, as well as pixel positions. However, geometric features are not good indicators for our target functions, i.e., visibility functions, since the correlation between visibility and geometric buffers is not high. Hence, we choose to utilize pixel positions as an indicator for our approximations. This choice allow our filter to compute filtered visibility in real-time. As result, our computational overhead (e.g., 15 ms) is much lower than previously developed methods for general rendered images (e.g., [3]) is typically one order of magnitude slower than ours).

Our conclusion is that high-quality visibility on hybrid rendering systems is promising thanks to the successful combination of accurate ray-traced shadows that can be filtered, for low-sampling rates, using a real-time version of adaptive image-based denoising methods.

- [1] Koa Ming Di and Henry Johan. Efficient screenspace rendering for area lights. In *Proceedings of the 33rd Computer Graphics International, CGI '16*, pages 29–32, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4123-3.
- [2] Eric Heitz, Jonathan Dupuy, Stephen Hill, and David Neubelt. Real-time polygonal-light shading with linearly transformed cosines. *ACM Transactions on Graphics (TOG)*, 35(4):41, 2016.
- [3] Bochang Moon, Jose A Iglesias-Guitian, Sung-Eui Yoon, and Kenny Mitchell. Adaptive rendering with linear predictions. *ACM Transactions on Graphics (TOG)*, 34(4):121, 2015.
- [4] Bochang Moon, Steven McDonagh, Kenny Mitchell, and Markus Gross. Adaptive polynomial rendering. *ACM Transactions on Graphics (TOG)*, 35(4):40, 2016.