

# Importance sampling over the visibility of area lights with *light portals*

J. Romeu Huidobro

Supervisors: M. van de Ruit, E. Eismann

EEMCS, Delft University of Technology, The Netherlands

## Abstract

Abstract goes here

## 1 Introduction

Sampling light sources as a means for estimating direct illumination at a given point is an important sampling operation in rendering [1]. Traditional techniques importance sample according to the light position, solid angle, and BSDF, but do not take into account visibility. This can lead to very high variance when rendering scenes with large light sources that are mostly occluded.

In this paper, we investigate the effectiveness of artist specified *light portals* importance sampling according to visibility. We first use multiple importance sampling [2] to combine visibility sampling with traditional light sampling. Next, we present a method to directly sample the product of the visibility and light position for certain scenes. Next we present an implementation extending the PBRT-v3 renderer described in [1]. Lastly we evaluate and discuss the effectiveness of the methods for a variety of scenes before concluding

## 2 Background and related work

**The direct lighting integral** Physically based renderers are generally concerned with evaluating the *light transport equation* (LTE) [3]. Most light transport algorithms reformulate the LTE into the path-integral formulation [2] which separates paths of different lengths into separate terms.

The second term in the path-integral formulation gives the outgoing radiance at a point  $p$  and direction  $\omega_o$  due to lighting from paths of length 1. The integral is

$$\int_{S^2} f(p, \omega_o, \omega_i) L_d(p, \omega_i) |\cos \theta_i| d\omega_i, \quad (1)$$

where  $S^2$  is the hemisphere of directions,  $f$  is the bidirectional scattering distribution function (BSDF) at point  $p$ , and  $L_d$  is the incident radiance coming directly from a light source.

**Monte Carlo Integration** This integral cannot be solved analytically in general, thus we use *Monte Carlo integration* as a means to numerically estimate it. To compute this estimate, we need to sample  $N$  directions  $\omega_i$  and apply the Monte Carlo estimator:

$$\frac{1}{N} \sum_{i=1}^N \frac{f(p, \omega_o, \omega_i) L_d(p, \omega_i) |\cos \theta_i|}{p(\omega_i)} \quad (2)$$

To reduce variance, we can use *importance sampling* to choose the directions  $\omega_i$ . The best practice is to use multiple importance sampling [2] to draw samples according to the BSDF term as well as the lighting term (by only sampling directions towards the light source), but this doesn't take into account the visibility of the light source.

**Light portals** *Light portals* are artist specified regions in the scene which indicate opening to an environment light (such as a window), which is useful to focus sampling of area lights to only the paths that are visible. They are highly effective for accelerating the convergence of path tracing in scenes using an environment light that is largely occluded, such as an interior scene with a window. Light portals are implemented in a number of production renderers such as Cycles [4] and Renderman [5]. Relevant papers into light portals include [6], which developed a portal-rectified reparameterization of the environment map which allows the environment light to be sampled according to its light intensity, giving better convergence than uniform sampling. [7] developed a generalization of light portals that allows for the use of arbitrary polygon meshes as portals, rather than the traditional planar rectangles.

This being said, traditional light portals only support sampling *environment* lights and currently do not support accelerating the sampling of area lights.

**Adaptive sampling** [8] developed a two step algorithm for efficiently sampling environment lights without requiring the use of artist specified portals. It works by firstly caching visibility information in the camera space and then adapting the sampling strategy according to the cached visibility information. Once again, this method works only for environment lights.

**Tridirectional path tracing** [9] developed a specialized tridirectional path tracing algorithm for rendering scenes resembling a camera obscura, that is, the observer and the light source are separated by a wall with a small opening. Their algorithm works by tracing the light paths that follow the trajectory of this opening rather than random sampling, leading to much faster convergence for specific scenes. The difference is that our method addresses the general situation of an area light source being partially occluded, not just the very specific, camera obscura scene.

## References

- [1] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically Based Rendering: From Theory to Implementation (3rd Ed)*. Morgan Kaufmann Publishers Inc, 2016. 1266 pp.
- [2] Eric Veach. “ROBUST MONTE CARLO METHODS FOR LIGHT TRANSPORT SIMULATION”. In: (), p. 432.
- [3] James T Kajiya. “THE RENDERING EQUATION”. In: 20.4 (1986), p. 8.
- [4] *Light Settings — Blender Manual*. URL: [https://docs.blender.org/manual/en/latest/render/cycles/light\\_settings.html?highlight=light%20portals](https://docs.blender.org/manual/en/latest/render/cycles/light_settings.html?highlight=light%20portals) (visited on 05/06/2022).
- [5] *PxrPortalLight*. Renderman Documentation. URL: <https://rmanwiki.pixar.com/display/REN24/PxrPortalLight> (visited on 05/06/2022).
- [6] Benedikt Bitterli, Jan Novák, and Wojciech Jarosz. “Portal-Masked Environment Map Sampling”. In: *Computer Graphics Forum* 34.4 (July 2015), pp. 13–19. ISSN: 0167-7055, 1467-8659. DOI: [10 . 1111 / cgf . 12674](https://doi.org/10.1111/cgf.12674). URL: <https://onlinelibrary.wiley.com/doi/10.1111/cgf.12674> (visited on 05/06/2022).
- [7] Shinji Ogaki. “Generalized Light Portals”. In: *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 3.2 (Aug. 26, 2020), pp. 1–19. ISSN: 2577-6193. DOI: [10 . 1145 / 3406176](https://doi.org/10.1145/3406176). URL: <https://dl.acm.org/doi/10.1145/3406176> (visited on 04/23/2022).
- [8] Asen Atanasov et al. “Adaptive Environment Sampling on CPU and GPU”. In: *ACM SIGGRAPH 2018 Talks*. SIGGRAPH ’18: Special Interest Group on Computer Graphics and Interactive Techniques Conference. Vancouver British Columbia Canada: ACM, Aug. 12, 2018, pp. 1–2. ISBN: 978-1-4503-5820-0. DOI: [10 . 1145 / 3214745 . 3214808](https://doi.org/10.1145/3214745.3214808). URL: <https://dl.acm.org/doi/10.1145/3214745.3214808> (visited on 05/06/2022).
- [9] Luke Anderson et al. “Aether: An Embedded Domain Specific Sampling Language for Monte Carlo Rendering”. In: *ACM Transactions on Graphics* 36.4 (July 20, 2017), pp. 1–16. ISSN: 0730-0301, 1557-7368. DOI: [10 . 1145 / 3072959 . 3073704](https://doi.org/10.1145/3072959.3073704). URL: <https://dl.acm.org/doi/10.1145/3072959.3073704> (visited on 05/06/2022).