Finding the Hole: Efficient area light importance lights with light portals

J. Romeu Huidobro

Supervisors: M. van de Ruit, E. Eismann

EEMCS, Delft University of Technology, The Netherlands

Abstract. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Write abstract last

1 Introduction

Physically based Monte Carlo light transport algorithms used to create photorealistic imagery with applications ranging from videogames, to architecture visualization. This being said, the most significant limitation is their very high computational demands.

In most scenes, the direct light is the most significant component, so most rendering algorithms compute direct and indirect lighting independently from one another, where direct light is computed by sending shadow rays from the point to the light sources. This being said, the probability distribution from which the shadow ray directions are can have a high impact on the convergence rate of the scene. In particular consider a scene where a lamp is occluded by a lampshade, shadow rays which are blocked by the lampshade will have no contribution to the image and are thus wasted sampling effort.

One widely used technique for computing direct lighting from environment lights, is *light portals*. These are artist specified regions in space which indicate an opening to the environment light. This information is then used in rendering to focus the sampled directions to those that go through the window. Our contribution consists in extending the notion of light portals to accelerate the sampling of area lights. In particular, we repurpose shadow volume algorithms to precompute the regions from which the light is visible through the portal.

After reviewing monte carlo techniques for direct lighting calculation, and providing an overview of existing use cases and sampling strategies for light portals (section 2), we expand on the details of our method (section 3). Next we discuss our implementation into an existing path tracer (section 4), evaluate it, in comparison to existing strategies for a diverse set of scenes (section 5), and discuss the results (section 6) before concluding (section 7).

2 Background and related work

The Light Transport Equation Physically based renderers are generally concerned with evaluating the recursive *light transport equation* (LTE) [1].

$$L_o(p,\omega_o) = L_e(p,\omega_o) + \int_\Omega f(p,\omega_o,\omega_i) L_o(p,\omega_o,\omega_i) |\cos\theta_i| d\omega_i, \tag{1}$$

which says that the exitant radiance at a point p in direction ω_o is equal to the emitted radiance at p towards ω_o, plus the portion of the incident light from the hemisphere of directions Ω which gets reflected towards ω_o.

The direct lighting integral By repeated unfolding of the recursive call, the LTE can be reformulated into the path-integral formulation [2] which separates paths of different lengths into separate terms, this allows us to calculate direct and indirect independently. The direct lighting term is the integral

$$\int_\Omega f(p,\omega_o,\omega_i) L_d(p,\omega_i) |\cos\theta_i| d\omega_i, \tag{2}$$

where L_d is the *direct* radiance from direction ω_i, f is the Bidirectional Scattering Function (BSDF) and the cosine term accounts for lamberts law.

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 ω_i and apply the Monte Carlo estimator:

$$\frac{1}{N} \sum_{i=1}^{N} \frac{f(p,\omega_o,\omega_i) L_d(p,\omega_r) |\cos\theta_i|}{p(\omega_i)} \tag{3}$$

To reduce variance, we can use *importance sampling* to choose the directions ω_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 approach does not take the visibility of the light source into account when drawing the samples, which can lead to high variance when rendering scenes where the light is partially occluded.

Light portals *Light portals* are artist specified regions in the scene which indicate an opening to an environment light (such as a window). The portal is then used to focus the environment light sampling to only those directions that are visible through the portal. They are highly effective for accelerating the convergence of path tracing for scenes using an environment light that is largely occluded, such as interior scenes with a window. Light portals are implemented in a number of production renderers such as Cycles [3] and Renderman [4], and are a widely used technique.

Several portal-sampling strategies exist, uniform portal sampling is easy to implement but can give poor convergence for scenes where the environment map has a highly non-uniform brightness distribution. To handle such cases, the product of the environment light's importance map and the portal visibility can be sampled directly [5], or alternatively, samples can be drawn according to the portal and the importance map and combined with multiple importance sampling.

This being said, light portals are only used for sampling *environment lights*, and a number of different challenges and opportunities are presented when applying the same technique to area lights.

Environment lights extend across the whole scene, thus any direction sampled towards the portal will also reach the light source, this is not the case for area lights. Furthermore, in most practical scenes the solid angle subtended by the light is smaller than the one subtended by the portal, making uniform portal sampling a very poor strategy for area lights.

3 Method

Sampling light through a portal

An interesting way to think about light portals is to consider the shadow that would be cast by the portal if we were to replace it with an occluder. With this in mind, we can partition the space in front of the portal into 3 distinct regions, the umbra, penumbra, and the region out of shadow. These are defined by (elmar citation) as:

- The **umbra** (hard shadow) represents the points where all paths to the light source are occluded by the portal
- The **penumbra** (soft shadow) is the remainder of the shadow, and represents the points where some, but not all paths to the light are occluded
- The **region outside of shadow** is the area where none of the paths to the light are occluded

However, the portal, of course is not an occluder, it is precisely the opposite, as it indicates an opening to the portal. Thus, we can reason about the portal-equivalent regions:

- **anti-umbra** the region where all paths to the light go *through* the portal
- **anti-penumbra** the region where some of the paths to the light go through the portal, but others don't
- region **outside of anti-shadow** is the area where none of the paths to the light go through the portal

The advantage of reasoning in terms of shadows is that we can use existing shadow volume algorithms to compute these regions, and to test for being inside of them (citations). Knowing which region we are in can be used to highly accelerate the sampling process. For points in the antiumbra, visibility through the portal is guaranteed, so we can use standard light sampling. For points in the antipenumbra we compute the projection of the light onto the portal-plane and sample their intersection, this, of course is more expensive, but guarantees visibility through the portal. Lastly, for points outside of the anti-shadow, there is no need to do any sampling at all, as the light is not visible through the portal, so we can just return black.

4 Implementation

We implement our method within the PBRT-V3 renderer [6]. To support planar lights, the shape base class is extended by the `AAPlane` which implements axis-aligned plane intersection. Next, the area light base class is extended by the `PortalLight` class, which behaves like a standard area light but also stores the geometry of the portal and implements the sampling strategy explained in section 3. No modification is done to the integrators themselves, only the direct lighting calculation changes, thus the portal sampling strategy can be used in any integrator which separates direct lighting from indirect lighting. For this report, the path integrator is used, which implements standard unidirectional path tracing with next event estimation.

5 Results

We evaluate the methods presented in a variety of scenes and compare their mean squared errors (MSE) with respect to time. The reference images were produced using a large number of samples with an unbiased unidirectional path tracer using light and BSDF MIS sampling to compute direct lighting. To ensure a fair comparison, the runtimes given include the maximum visible frustum pre-computation step.

Experiment 1

The concrete experiments and comparisons to do are not finalized yet, so I cannot write about them in detail. subsections of this structure will be repeated for each of the experiment setups

We first compare each method in x scene(s) containing description of the properties of the scene(s) (Figure 1). As demonstrated discussion of the results

6 Discussion

Incomplete due to missing results.

Things to potentially include (depending on if I implement the feature or not):

- **Limitation:** Requires additional artist input so not great, but mention that since standard light portals are already included in most pipelines it should be feasible to implement it.
- **Limitation:** Method presented is limited to certain combinations of portal shape and light shape, give general guidance on how to generalize
- **Future work:** Multiple portals, decide on which portal to sample based on frustum
- **Future work:** Multiple deciding which portal to sample based on frustum
- **Future work:** Chain portals against eachother and project ith portal to $i + 1$th portal, good for very precise visibility constraint

7 Conclusion

Incomplete since still on results

We have developed a strategy for efficiently importance sampling area lights according to the visibility of the light through an artist-specified portal. We demonstrated that the method accelerates convergence for a wide variety of scenes where the light source is partially occluded.

References

[1] James T Kajiya. "THE RENDERING EQUATION". In: 20.4 (1986), p. 8.

[2] Eric Veach. "ROBUST MONTE CARLO METHODS FOR LIGHT TRANSPORT SIMULATION". In: (), p. 432.

[3] *Light Settings — Blender Manual*. URL: https://docs.blender.org/manual/en/latest/render/cycles/light_settings.html?highlight=light%20portals (visited on 05/06/2022).

[4] *PxrPortalLight*. Renderman Documentation. URL: https://rmanwiki.pixar.com/display/REN24/PxrPortalLight (visited on 05/06/2022).

[5] 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. URL: https://onlinelibrary.wiley.com/doi/10.1111/cgf.12674 (visited on 05/06/2022).

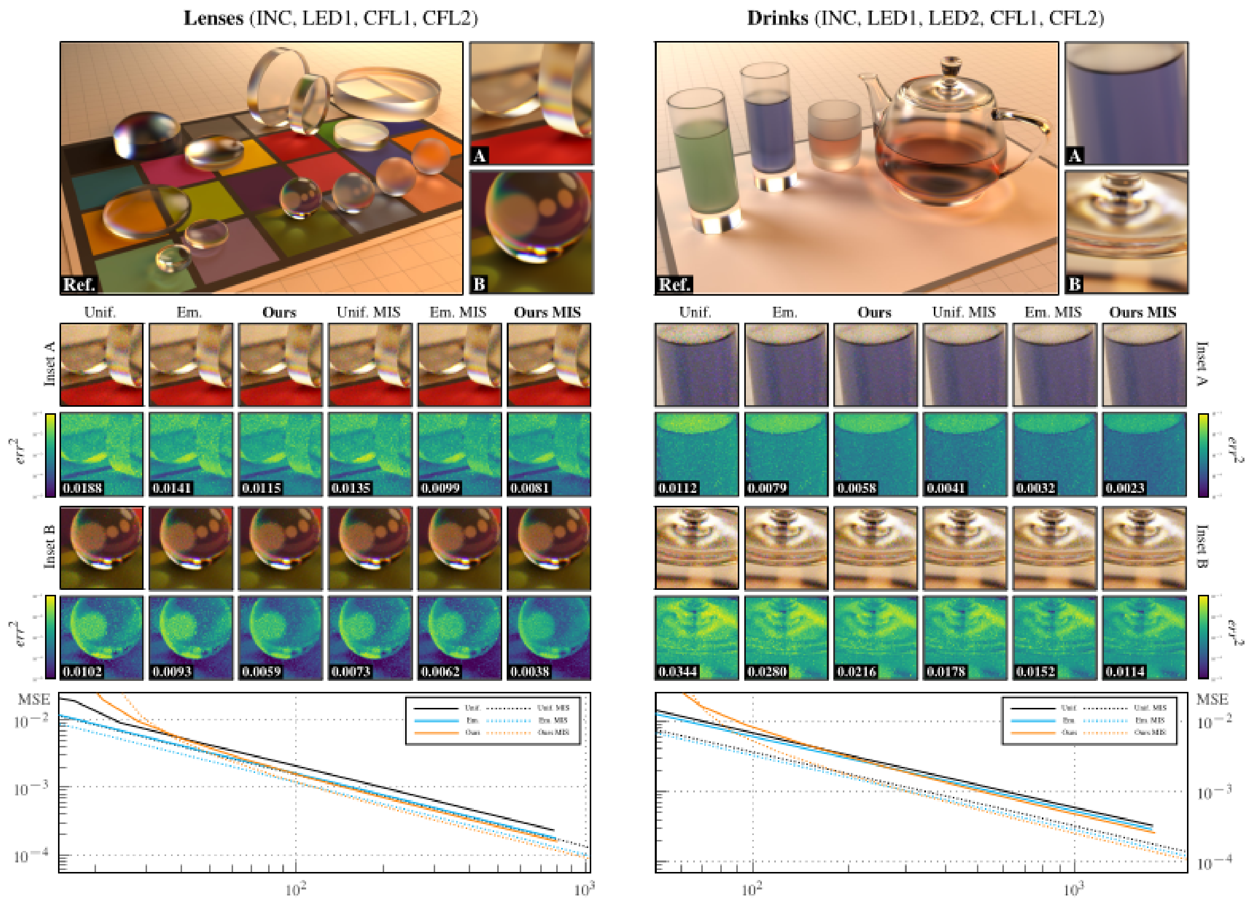

Figure 1: *(The results here are not mine!, Just a screenshot from one of Mark's papers to illustrate the general layout of the image) We compare the convergence rates for light sampling (Light), light-portal-projection (Proj) sampling, and light-portal MIS sampling (MIS) each method with and without frustum culling for a variety of scenes. The listed errors are for the highlighted image insets. As demonstrated, projection sampling generally is the most robust method, but converges slightly slower for scenes where the light is comparatively small in comparison to the portal*

[6] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically Based Rendering: From Theory to Implementation (3rd Ed)*. Morgan Kaufmann Publishers Inc, 2016. 1266 pp.