

A Deferred Shading Algorithm for Real-Time Indirect Illumination

Cyril Soler, Olivier Hoel, Frank Rochet

▶ **To cite this version:**

Cyril Soler, Olivier Hoel, Frank Rochet. A Deferred Shading Algorithm for Real-Time Indirect Illumination. ACM SIGGRAPH 2010 Full Conference DVD Rom, 2010, Siggraph Talks. inria-00480869v1

HAL Id: inria-00480869

https://inria.hal.science/inria-00480869v1

Submitted on 5 May 2010 (v1), last revised 15 Nov 2010 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Deferred Shading Pipeline for Real-Time Indirect Illumination

Cyril Soler
INRIA, Grenoble university*

Olivier Hoel
INRIA, Grenoble university

Frank Rochet
Eden Games

1 Introduction

Computing indirect lighting in video games simultaneously improves gameplay and scene realism. However, the context of 3D video games brings very restrictive constraints: (1) the computation should be very fast (less than 10 ms) and –most importantly– with a constant cost, *i.e.* independent of both the geometric and lighting complexity of the scene; (2) the indirect illumination algorithm should work seamlessly on dynamic scenes, with any source of direct illumination –not only point light sources– and easily take place into a game engine pipeline; and (3) the computed result may be approximate but must be artifact-free and temporally coherent.

We present a deferred-shading algorithm for computing indirect illumination that exactly suites these constraints. The method is approximate as the indirect visibility is not accounted for, but it naturally allows multiple bounces and the last bounce of indirect illumination can reflect on non-diffuse materials.

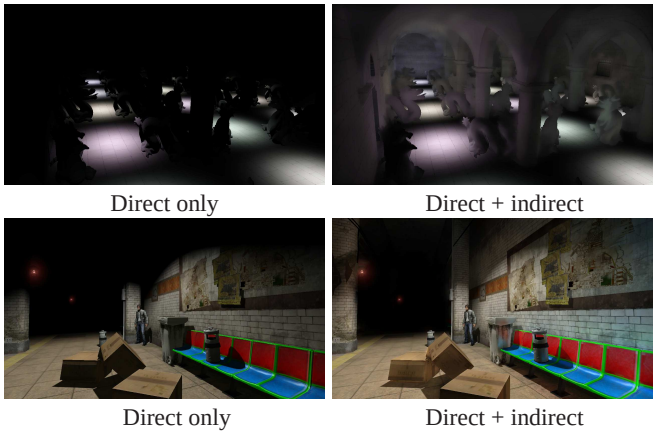


Figure 1: *One-bounce indirect illumination computed by our technique (*top row *in a scene with 3M polygons), and* bottom row *in the the game engine "*Alone in the Dark, Near Death Investigation*". In the bottom the indirect illumination is importance sampled and reflected on the glossy metallic bin. Computation time is 10* ms *for both on a GTX260 NVidia card, at resolution* 1280*x*1024*.*

2 Deferred shading indirect illumination

Our algorithm works entirely in screen-space, hierarchically. It takes G-Buffers as input: normals, reflectance parameters, depth, and diffuse direct illumination. These buffers are mipmap-ed, and we compute indirect illumination at each level using a common shader. This shader gathers light from points randomly sampled in a fixed-size circular region of the image. This eventually gathers light from arbitrary distances. To do this, we reformulate the Monte-Carlo sampling of the bouncing light equation to fit uniform screen-space samples. The contributions from all mipmap levels are then up-sampled (with bilateral up-sampling) and added together. The resulting image contains indirect lighting between all visible parts of the scene. To remove the noise due to sampling, we average indirect lighting across successive UN-warped frames (at now extra cost), and we combine multiple indirect illumination components from several cameras. The indirect illumination is combined with the reflectance buffer. It is added to the direct illumination, and possibly further used as input to compute the next bounce.

As a consequence, our technique computes light interaction at arbitrary distance in the image plane, at a constant cost. Furthermore, it allows a computation time / accuracy trade-off by selecting starting from which screen resolution the light gathering shader is applied. This approximation results in low-pass filtering the indirect illumination rather than loss of illumination. It works with arbitrary sources of direct illumination (most games commonly use *e.g.* a huge number of omni-directional point light sources), and the cost is independent on visual complexity and illumination. Finally, our algorithm is real-time (less than 10 *ms* per frame) on recent graphics hardware as shown in the table below:

Resolution/samples	Quarter/64		Quarter/256		Half/256		Half/64	
Architectures	(*a*)	(*b*)	(*a*)	(*b*)	(*a*)	(*b*)	(*a*)	(*b*)
Total time (ms)	13.9	**4.6**	33.6	**11.2**	110.6	32.6	34.8	**10.5**

Figure 2: *Computation times in* ms *for different parameter sets, measured using* (a) *NVidia 8800GTS and* (b) *NVidia GTX260 cards, for a display resolution of* $1280 \times 720p$ *for the images on Figure 1.*

3 Related work

We extend to arbitrary distances the local indirect illumination computation proposed by Ritschel [2009]. Multiresolution Splatting of Indirect Illumination [Nichols and Wyman 2009] and Hierarchical Image-Space Radiosity [Nichols et al. 2009] also compute long-distance indirect illumination. They use virtual point lights (VPL) to represent the direct illumination, which they collect with Reflective Shadow Maps (RSM). While the former adaptively splats illumination for all VPLs, the later further organizes the VPLs into a tree structure and performs adaptive gathering.

RSMs may need a large number of render passes to approximate low frequency sources of direct illumination (*e.g.* envmaps, or indirect lighting for 2nd bounce). Methods based on RSMs thus depend on the complexity of the direct illumination and therefore are less robust to varying lighting conditions and less suited to more than one bounce of indirect illumination. Because we use and compute screen-space illumination, we don't have this limitation. The cost of RSMs also depends on the scene complexity, whereas our technique has a constant cost w.r.t to geometric and illumination complexity. Splatting indirect illumination from VPLs does not allow to importance sample non diffuse BRDFs, because all pixels share the same VPLs. With our method, each pixel may have it's own sets of lights samples in screen-space.

Imperfect shadow maps [Ritschel et al. 2008] approximately simulates indirect shadows, for large enough objects. In addition to using VPLs, this method requires preprocessing of the geometry, which makes it incompatible with fully dynamic scenes where the geometry content may not be known in advance, such as in games.

References

NICHOLS, G., AND WYMAN, C. 2009. Multiresolution splatting for indirect illumination. In *Proc. ACM Symposium on Interactive 3D Graphics and Games 2009*.

NICHOLS, G., SHOPF, J., AND WYMAN, C. 2009. Hierarchical image-space radiosity for interactive global illumination. In *Proceedings of Eurographics Symposium on Rendering 2009*, H. Lensch and P.-P. Sloan, Eds.

RITSCHEL, T., GROSCH, T., KIM, M. H., SEIDEL, H.-P., DACHSBACHER, C., AND KAUTZ, J. 2008. Imperfect shadow maps for efficient computation of indirect illumination. *ACM Transactions on Graphics 27*, 5 (December).

RITSCHEL, T., GROSCH, T., AND SEIDEL, H.-P. 2009. Approximating dynamic global illumination in image space. In *Proc. ACM Symposium on Interactive 3D Graphics and Games 2009 (I3D '09)*.