COMBINING BIDIRECTIONAL PATH TRACING AND MULTIPASS RENDERING

Frank Suykens, Yves D. Willems

Department of Computer Science
Katholieke Universiteit Leuven
Celestijnenlaan 200A, B3001 Heverlee
Belgium
Frank.Suykens@cs.kuleuven.ac.be

ABSTRACT

A flexible multipass framework for global illumination is presented in which bidirectional path tracing is used as a final pass. This final pass is constructed so that it is ensured that a complete solution for the global illumination problem is obtained, no matter what kind of preprocessing steps are used. Results show significant improvement compared to both other multipass methods and standard bidirectional path tracing.

Keywords: global illumination, Monte Carlo, multipass rendering, bidirectional path tracing

1 INTRODUCTION

The goal of physically based rendering is to create photo-realistic images of virtual scenes. In order to compute such images, light transport must be simulated in this scene. In the absence of participating media this transport can be described by an integral equation, the rendering equation [Kaj86]. Over the years many global illumination algorithms have been proposed for solving this difficult problem, each having its strong points and weaknesses.

Object based methods store illumination information in the scene. They compute an approximate solution of the rendering equation for each surface in the scene. For instance the classic radiosity method [GTGB84, CW93] assumes that all surfaces in the scene are diffuse reflectors. The solution is stored on a finite element mesh and can be computed efficiently using hierarchical and clustering techniques [CW93]. Extensions to non diffuse environments are possible but they require a lot more storage for an accurate solution [SAWG91, CSS96].

Pixel based methods compute light transport in the scene for every pixel of the image. These methods are inherently view dependent. Well known examples are classical ray tracing [Whi80], Monte Carlo path tracing [Kaj86], particle tracing [DW95] and bidirectional path tracing [LW93, VG94]. The last three methods are capable of computing a full solution to the global illumination problem, the difference being that they respectively trace paths from the eye, the lights and both the eye *and* the lights. However computational demands are high in order to reduce the noise in the resulting images to an acceptable level. For specular-like surfaces these methods perform extremely well.

Two-pass and multipass methods attempt to combine different methods and use their respective advantages to create a better, more robust algorithm. Usually one or more object based methods are used as a preprocessing step for a final pixel based pass.

In this paper we present a flexible multipass framework, where bidirectional path tracing is used as a final pass. All the light transport paths not covered by the preprocessing steps are filled in by the bidirectional path tracing pass. As such the proposed method ensures that a full solution to the global illumination problem (with arbitrary reflection and refraction functions) is obtained.

In section 2 of the paper a more detailed explanation of multipass methods is described. Section 3 covers bidirectional path tracing (BPT)

in general. How to use BPT as a final pass is explained in section 4 and section 5 gives implementation details and results.

2 MULTIPASS RENDERING

In '87 Wallace et al. [WCG87] presented a two pass method that uses radiosity for computing the diffuse component of the light transport and a view dependent Z-buffer pass for the specular component. Many other multipass algorithms have been proposed over the years ([SP89, Jen96, Kel97], and for an overview [CW93, p. 319-325]). Each pass in a multipass algorithm computes a different part of the light transport, incorporating the results of previous passes. Care must be taken that all light transport paths are accounted for only once.

To compute only part of the light transport in multipass algorithms, the bidirectional scattering distribution function (BSDF) is split into separate components. We distinguish between diffuse, glossy and specular reflection and transmission. The diffuse component scatters light uniformly over the entire hemisphere. Glossy reflection has moderate preferential directions for scattering. Specular reflection means that light of an incoming ray is scattered in a very small solid angle around the perfectly reflected direction (in fact so small that storing a full radiance solution on a specular surface would not be feasible). BSDF's of real materials consist of a mixture of all three components.

Each pass in a multipass algorithm generates paths that include only some components of the BSDF. Each pass can be characterised by the type of light transport paths that are covered by the pass. Regular expressions are a convenient way to describe the covered transport paths [Hec90]. Some examples will be given next.

Radiosity algorithms are often used as a preprocess step to efficiently calculate diffuse interreflections [GTGB84, CW93]. Paths covered by classical radiosity are $L(D_r)^*$ paths [1], meaning all paths starting on a light and hitting a diffuse surface possibly after some diffuse reflections. Extended form factors [SP89, RT90] or glossy finite element methods [SAWG91, CSS96] can be used to extend the covered path types. These extensions however require a lot more memory to store an accurate solution.

Light ray tracing is frequently used to precalculate caustics on diffuse surfaces [CRMT91], since it is much easier to generate $L(S)^+D_r$ paths when starting from the light sources.

A final raytracing or path tracing pass completes most of the multipass algorithms. This pass computes glossy and specular highlights and reflections. A final gather pass can be used to compute high quality images from a lower quality radiosity solution [RPV93].

Although multipass algorithms have produced very realistic images, most of them do not include all possible transport paths. Shirley's extensive method [Shi90] for example, uses light raytracing for caustics, Monte Carlo radiosity as a second pass for diffuse interreflections and raytracing as the final pass. Paths covered are $L(S)^*(D_r)^*(S)^*E$, missing $D(S)^*D$ subpaths.

The method presented here, allows arbitrary (but covering disjunct transport paths) preprocessing passes. The regular expressions describing these passes, are used directly to guide the bidirectional path tracing step. All paths not covered are automatically filled in by this final pass.

Slusallek et al. [SSH$^+$98] recently presented Lighting Networks, a flexible system allowing many different multipass configurations. Our method could be nicely integrated into this system and could be used as a final pass in a lighting network.

3 BIDIRECTIONAL PATH TRACING

Bidirectional path tracing was introduced independently by Lafortune [LW93, LW94] and Veach [VG94]. It is an unbiased Monte Carlo algorithm that can handle arbitrary reflection or transmission functions. We will shortly review the idea of bidirectional path tracing here. For a detailed mathematical explanation the reader is referred to [Vea97].

Paths are traced from the eye and the light simultaneously as shown in figure 1. Each vertex of the light path is then connected to each vertex of the eye path. Each connection results in a full path that may carry light from a light source to the eye. Figure 1 shows a path of length five [2], generated by a light subpath of length three connected with an eye subpath of length two. The same path could also be generated by tracing an eye subpath of another length, say 4 and a light subpath of length 1. So in fact bidirectional path tracing offers a set of estimators for the same integral. In order to prevent counting paths more than once, Veach [VG95] shows how these different estimators can be weighted in a provable good way and still get an unbiased solution.

[1] The following notation is used for the regular expressions : 'D_r' stands for diffuse reflection, 'G_r' for glossy and 'S_r' for specular reflection. For transmission a subscript 't' is used instead of an 'r'. If 't' and 'r' are omitted, both reflection and transmission are included. All components together are denoted by 'X' and equal '$(D|G|S)$'. '$*$' means zero or more occurrences and '$+$' at least one.

[2] The length of a path is defined by the number of vertices in the path, including the eye and/or light vertex

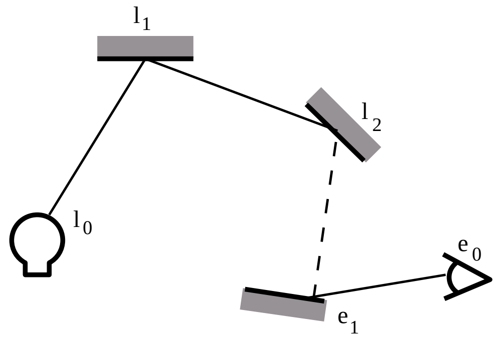

Figure 1: A bidirectional path of length 5 with an eye subpath of length 3 and an eye subpath of length 2.

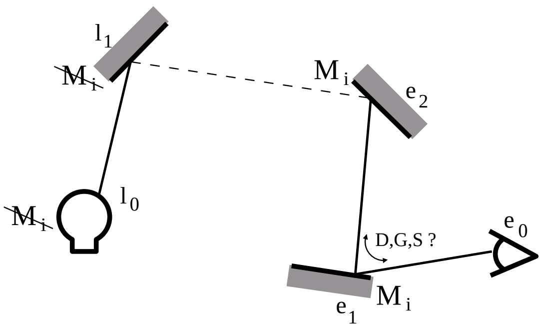

Figure 2: A bidirectional path. SPaR can be read at e_1 and e_2. BSDF components must be chosen carefully in e_1 to prevent counting the same paths multiple times

Bidirectional path tracing performs better than standard path tracing or particle tracing for a wide variety of scenes. In fact path tracing and particle tracing are special cases of BPT where the maximum light subpath length and respectively the eye subpath length are set to 1. Scenes with a large amount of diffuse and even glossy interreflection, still do require a lot of samples to reduce the noise in the resulting image. A multipass approach where diffuse interreflection is computed with a radiosity method may improve the results significantly, as will be demonstrated later.

4 COMBINATION

This section will explain the basis of our method, namely how bidirectional path tracing can be combined with multipass rendering.

4.1 Object based preprocessing steps

In a multipass algorithm a number of object based preprocessing steps are performed first. The proposed method allows a wide variety of preprocessing steps. The result of these steps are a number of *Stored Partial Radiance* solutions (SPaR). Each of these SPaR's stores radiance information in the scene for certain classes of light transport pathtypes.

The paths covered in each SPaR i are described by a corresponding regular expression M_i. Radiosity methods provide a SPaR with $M_{rad} = L(D_r)^*$. For a caustic map, $M_{caustic}$ equals $L(S)^*D_r$. How the radiance information is stored is encapsulated in the SPaR itself. Only a function must be provided that returns the stored radiance for a certain point on a surface and an outgoing direction.

The regular expressions M_i of the SPaR's can be used to check if no common transport paths are covered by different SPaR's (See [SSH+98]). If this were so the common paths would be accounted for multiple times in the final image, resulting in an incorrect image.

4.2 Readout strategy

The SPaR's store partial radiance information in world-space. In this section we will introduce the *Readout Strategy* of a SPaR, which determines how this world-space radiance information is used in a pixel based gather pass such as path tracing or even Z-buffer based algorithms. The result of applying the Readout Strategy is an image showing partial illumination. The next section will explain how the remaining illumination can be computed with bidirectional path tracing.

We will use path tracing as a pixel based gather pass in our Readout Strategies. Eye paths are traced into the scene and are scattered when hitting a surface. The Readout Strategy determines at which vertices v the SPaR will be read and which BSDF components must be evaluated for intermediate vertices.

Consider for instance the eye subpath $e_0e_1e_2$ in figure 2. A first thing to choose is whether M_i will be read out at vertex $e2$. If so one must also carefully choose which components of the BSDF will be used in vertex e_1 for evaluating the contribution of the path $e_0e_1e_2$.

Consider a radiosity SPaR ($M_i = L(D_r)^*$) and suppose we will use this SPaR at every vertex of an eye path. Readout will happen at depth 1 (e_1), resulting in a contribution to the image of M_iE paths (plain visualisation of the radiosity solution). Readout will also occur at vertex $e2$, resulting in the following contribution to the image : $M_i\ A\ E$, where A are the *admissible scattering components* (ASC). A must *not* include D_r since this would give $M_i(D_r)E = L(D_r)^*(D_r)E \subset L(D_r)^*E$ contributions to the image, which are already handled by readout in vertex e_1. A maximal contribution for this readout is obtained by setting $A = (D_t|G|S)$.

Generalised to paths of different lengths this yields regular expressions $M_i A_k E$ for eye path vertices at depth k. A_k is a regular expression of ASC's indicating the scattering components to use for vertices at depth $l < k$.

For the radiosity example one could choose the following readout strategy :

$$\begin{array}{llllll}
1 & & & & M_i & E \\
2 & & & M_i & (D_t|G|S) & E \\
3 & & M_i & (D_t|G|S) & (D|G|S) & E \\
4 & M_i & (D_t|G|S) & (D|G|S) & (D|G|S) & E \\
\ldots & & & & &
\end{array} \tag{1}$$

To ensure that a strategy does not count paths more than once, the following must hold [3] :

$$\forall m, n \geq 0 : M_i A_m E \cap M_i A_n E = \Phi$$

This is easily verified for the radiosity readout strategy (1).

The total contribution to the final image of a SPaR with its associated readout strategy can be written as a single regular expression :

$$T_i = M_i E \cup M_i A_1 E \cup M_i A_2 E \cup \ldots$$

If more than one SPaR is used in a multipass configuration, $T_i \cap T_j$ must be empty for all SPaR pairs i and j to avoid counting paths twice.

4.3 Bidirectional final pass

The bidirectional path tracing pass must fill in all paths not covered by the SPaR's and their Readout strategies. We ensure this by computing for each bidirectional path the full solution (using all components of the BSDF) and then subtracting exactly those contributions that are already covered by the SPaR's. Formally the following contributions are computed :

$$(L(D|G|S)^* E) \ \setminus \ (\bigcup_i T_i)$$

In order to subtract the contributions efficiently, operations on regular expressions for every traced path should be avoided. We propose the following method : A (user adjustable) maximum path length is chosen for the bidirectional paths. For every possible path length, a *RegExp list* of path types to subtract is created. Each element of a list consists of a single chain of BSDF components (one chain element per vertex in the path).

Take for example a caustic map. Such a map stores $M_c = L(S)^+ D$ paths. A readout strategy can be $A_j = (D|G|S)^j$, meaning that all BSDF components are evaluated for intermediate eye subpath vertices. The subtract list for paths of length 5 looks like :

$$\begin{array}{l}
L\,S\,D\,(D|G|S)\,E \\
L\,S\,S\,D\,E
\end{array}$$

The subtract list can be obtained by enumerating the regular expression T_i for every SPaR and path length using standard regular expression algorithms. For simple T_i expressions specific routines can be used in order to reduce the size of the RegExp list.

When a bidirectional path of a certain length needs to be evaluated, the subtract list for that length is consulted. For each element in the list the path must be evaluated with BSDF components defined by that element. Our implementation stores the separate BSDF components in the path vertices, so that this evaluation can be done efficiently. Subtracting the contributions introduces only a small overhead compared to the tracing of the bidirectional paths.

Interesting to note is that :

- The contributions to be subtracted from bidirectional paths only depend on the total length of the path, and not on individual lengths of the eye or light subpath. These lengths only play a role in the probability density function that was used to generate the path and not in the evaluation of the path function itself.
- The eye subpaths generated for bidirectional path tracing can also be used to perform readout of the SPaR's. This reduces computation time. However no readout may occur on vertices of a light subpath, because the probability density functions used to generate the light vertices are unsuitable for gathering stored radiance (See figure 2).

5 IMPLEMENTATION AND RESULTS

The proposed method was implemented in RenderPark [4]

The bidirectional path tracing implementation is based on the work of Veach. When tracing subpaths, BSDF sampling is used to choose a scattered direction after a surface hit. Russian roulette is used to terminate the paths together with a maximum path length constraint. Light importance sampling [SWZ96] is also used when generating points on light sources for paths with a light subpath of length 1.

[3] Φ denotes the empty set

[4] RenderPark provides a testbed for realistic image synthesis and is freely available from `http://www.cs.kuleuven.ac.be/~graphics/`

5.1 Radiosity preprocess

The method was tested with a higher order Galerkin radiosity preprocess [Zat93]. The corresponding SPaR covers LD_r^* paths. The method was tested on a specific scene containing several important transport path types. The flexibility of the system will be demonstrated by how readout strategies can be chosen and adapted to get better images.

Figure 3 shows the scene computed with the radiosity method. The panel above the left light is painted diffuse white, the black sphere is in fact nearly perfect transparent. It is clear from this image that important transport paths are missing.

Figure 4 shows the same scene but now a traditional path tracing second pass is performed as well. At each vertex of a path, the radiosity solution is read and a scattered ray is spawn for specular and glossy reflection. Shadow rays are sent to the light sources for glossy and specular highlights. This can be fitted into our framework by choosing the readout strategy : $(A_k = (D_t|G|S)^{k-1})$. This method covers paths $L\ D_r^*\ (D_t|G|S)^*\ E$. As can be seen from this expression and the image, caustic paths $(L(S)^+DE)$ are not included in the solution. Specular and glossy reflection and refraction are, so the sphere at least looks transparent.

Now we will try to find a good readout strategy, so that both the strengths of radiosity and bidirectional path tracing are combined. An important conclusion will be that it is not always best to handle as many transport paths as possible with the readout strategy.

In our readout strategies we allow arbitrary expressions for each A_k. Recall the previously mentioned strategy for radiosity (1) :

1				M_i	E
2			M_i	$(D_t\|G\|S)$	E
3		M_i	$((D_t\|G\|S)$	$(D\|G\|S)$	E
4	M_i	$((D_t\|G\|S)$	$(D\|G\|S)$	$(D\|G\|S)$	E
...					

Diffuse scattering is also considered except for vertex $k-1$ in a path of length k (readout in vertex k). Note that different scattering components are used in the same vertex ($(D_t|G|S)$ or $(D|G|S)$). With this strategy the following paths contribute to the final image ($T_{rad} = L(D_r)^*(D|G|S)^*E = L(D_r)^*(X)^*E = L(X)^*E$). So *all* possible paths would be handled by eye path tracing and readout and light subpaths would not contribute. This is highly undesirable, because caustics are much easier generated by tracing light paths.

Therefore we will modify the radiosity preprocess so that it does *not include emitted light* in the readout. This results in $M_{rad'} = L(D_r)^+$ and $T_{rad'} = L(D_r)^+(X)^*E$. With this setup caustic paths (LS^+DE) will automatically be handled by bidirectional path tracing.

Figure 5 shows the bidirectional path tracing multipass method with the previous setup. Note the two caustics on the floor. The one on the left (originating from the right light) is formed by $L(S_t)(S_t)D_rE$ paths. The caustic on the right however is formed by $LD_r(S_t)(S_t)D_rE$ paths, more specifically by paths originating on the left light and diffusely bouncing of the white panel above it, making this a kind of 'diffuse caustic' effect.

Figure 6 shows the scene rendered with bidirectional path tracing alone. Regions where diffuse interreflection is important show significantly more noise than in the multipass method, as can be expected (See also the close-ups in figure 8). Note however the diffuse caustic on the right part of the floor. It contains considerably less noise than the multipass solution. The next section explains why this is so, and how this can be alleviated.

5.2 Optimisation for diffuse caustics

To see why the diffuse caustic is noisier in the multipass approach we will investigate how the paths that contribute to this caustic are generated in both methods. As mentioned before, the diffuse caustic is formed by $LD_r(S_t)(S_t)D_rE$ paths.

The multipass algorithm uses storage readout for LD^+X^*E paths. This includes the paths generating the diffuse caustic. Since readout is done for eye subpaths, the probability for generating a diffuse caustic path is as follows. First a ray from the eye is traced and must hit the floor where the diffuse caustic is visible. No problem so far, because this will happen anyway when the affected pixels in the image are sampled. On the floor a (diffuse) reflected ray will be spawn. This ray must hit the specular transmitting sphere in such a way that a new refracted ray leaving the sphere hits the diffuse white panel above the light. Here readout will take place. Because the panel is small, the probability for generating such a diffuse reflected ray is small.

When using standard bidirectional path tracing it turns out that the diffuse caustic is mostly formed by paths consisting of a light subpath of length 5 (light $\rightarrow$ white panel $\rightarrow$ entering sphere $\rightarrow$ leaving sphere $\rightarrow$ floor) and an eye subpath of length 1 (the eye itself). In the example scene this is a better way to compute the diffuse caustic. So we want to adapt the multipass setup, to have these type of paths handled by BPT.

It is not difficult to do this. We introduce a new SPaR that stores direct light on diffuse surfaces, $M_{direct} = L(D_r)$. This SPaR is computed by the radiosity preprocess at no additional cost. The readout strategy is chosen such that only $M_{direct}(S)^+(D)E$ paths are computed (A list for A_k is easily computed for this). The contribution of these paths is *subtracted* from the final result. Together with the other SPaR M_{rad} the following paths are covered : $L(D_r)^+X^*E \setminus L(D_r)(S)^+(D_r)E$. The final BPT pass computes all paths not covered and thus computes the diffuse caustic. The result is shown in Figure 7, close-ups are shown in figure 9.

Note that when the diffuse panel were larger, readout might become better than BPT for the caustic. Therefore we're currently looking into making a weighted combination of the two methods, taking into account the probability densities of the generated paths, similar to Veach's multiple importance method [VG95]. Veach's method cannot be directly applied because the different methods compute different integrals, due to the readout.

6 CONCLUSION AND FUTURE WORK

We have presented a flexible multipass framework for global illumination. It is shown how bidirectional path tracing can be used as a final pass that automatically ensures that a complete solution to the global illumination problem can be obtained. The method allows a wide variety of preprocessing passes and imposes no constraints on the reflection or transmission characteristics of surfaces in the scene. Results indicate that the strengths of preprocess steps can be combined with those of bidirectional path tracing.

Many promising improvements and extensions are still possible. As is done in Lighting Networks [SSH$^+$98] the domain of SPaR's can be restricted to certain parts of the scene. The BPT pass can be easily adapted to incorporate these restrictions. One could for instance only store caustics on a few surfaces in the scene. Caustics on other surfaces will be computed by the BPT pass automatically.

Since many paths may be handled by the preprocessing steps together with their readout strategy, a lot of bidirectional paths carry a small amount of energy. Adaptive importance sampling for light paths can improve this [DW95], and ensure that more paths are generated with significant energy, thereby reducing the noise in the final image. Metropolis light transport [VG97] offers efficient implicit importance sampling, and can be adapted in the same way as bidirectional path tracing here.

As shown in the example, it is not always best to handle as many paths as possible with the readout. An intelligent weighted combination of the readout and bidirectional paths could alleviate the problem of manually deciding what path types need to be handled by what method.

ACKNOWLEDGEMENTS

Thanks to Philippe Bekaert for adapting his Galerkin radiosity implementation to my needs and for helpful comments on the text. This research was supported by FWO Grant #G.0263.97

REFERENCES

[CRMT91] Shenchang Eric Chen, Holly E. Rushmeier, Gavin Miller, and Douglass Turner. A progressive multi-pass method for global illumination. *Computer Graphics (SIGGRAPH '91 Proceedings)*, 25(4):165–174, July 1991.

[CSS96] P. H. Christensen, E. J. Stollnitz, and D. H. Salesin. Global Illumination of Glossy Environments Using Wavelets and Importance. *ACM Transactions on Graphics*, 15(1):37–71, January 1996.

[CW93] M. Cohen and J. Wallace. *Radiosity and Realistic Image Synthesis.* Academic Press Professional, San Diego, CA, 1993.

[DW95] Philip Dutre and Yves D. Willems. Potential-Driven Monte Carlo Particle Tracing for Diffuse Environments with Adaptive Probability Density Functions. In P. M. Hanrahan and W. Purgathofer, editors, *Rendering Techniques '95 (Proceedings of the Sixth Eurographics Workshop on Rendering)*, pages 306–315, New York, NY, 1995. Springer-Verlag.

[GTGB84] C. M. Goral, K. E. Torrance, D. P. Greenberg, and B. Battaile. Modelling the interaction of light between diffuse surfaces. *Computer Graphics (SIGGRAPH '84 Proceedings)*, 18(3):212–22, July 1984.

[Hec90] Paul S. Heckbert. Adaptive radiosity textures for bidirectional ray tracing. *Computer Graphics*, 24(4):145–154, August 1990.

[Jen96] Henrik Wann Jensen. Global illumination using photon maps. In Xavier Pueyo and Peter Schröder, editors, *Eurographics Rendering Workshop 1996*, pages 21–30, New York City, NY, June 1996. Eurographics, Springer Wein. ISBN 3-211-82883-4.

[Kaj86] J. T. Kajiya. The rendering equation. In *Computer Graphics (SIGGRAPH '86 Proceedings)*, volume 20, pages 143–150, August 1986.

[Kel97] Alexander Keller. Instant radiosity. *Computer Graphics*, 31(3A):49–56, August 1997.

[LW93] Eric P. Lafortune and Yves D. Willems. Bi-directional Path Tracing. In H. P. Santo, editor, *Proceedings of Third International Conference on Computational Graphics and Visualization Techniques (Compugraphics '93)*, pages 145–153, Alvor, Portugal, December 1993.

[LW94] E. P. Lafortune and Y. D. Willems. A theoretical framework for physically based rendering. *Computer Graphics Forum*, 13(2):97–107, June 1994.

[RPV93] Holly E. Rushmeier, Charles Patterson, and Aravindan Veerasamy. Geometric simplification for indirect illumination calculations. In *Proc. Graphics Interface '93*, pages 227–236, Toronto, Ontario, May 1993. Canadian Inf. Proc. Soc.

[RT90] Holly E. Rushmeier and Kenneth E. Torrance. Extending the radiosity method to include specularly reflecting and translucent materials. *ACM Transactions on Graphics*, 9(1):1–27, January 1990.

[SAWG91] F. Sillion, J. Arvo, S. Westin, and D. Greenberg. A global illumination solution for general reflectance distributions. *Computer Graphics (SIGGRAPH '91 Proceedings)*, 25(4):187–196, July 1991.

[Shi90] Peter Shirley. A ray tracing method for illumination calculation in diffuse specular scenes. In *Proceedings of Graphics Interface '90*, pages 205–212, Toronto, Ontario, May 1990.

[SP89] François Sillion and Claude Puech. A general two-pass method integrating specular and diffuse reflection. *Computer Graphics*, 23(3):335–344, July 1989.

[SSH+98] Ph. Slusallek, M. Stamminger, W. Heidrich, J.-C. Popp, , and H.-P. Seidel. Composite lighting simulations with lighting networks. *IEEE Computer Graphics and Applications*, 18(2):22–31, March 1998.

[SWZ96] Peter Shirley, Changyaw Wang, and Kurt Zimmerman. Monte Carlo techniques for direct lighting calculations. *ACM Transactions on Graphics*, 15(1):1–36, January 1996.

[Vea97] Eric Veach. *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, December 1997.

[VG94] Eric Veach and Leonidas Guibas. Bidirectional Estimators for Light Transport. In *Fifth Eurographics Workshop on Rendering*, pages 147–162, Darmstadt, Germany, June 1994.

[VG95] Eric Veach and Leonidas J. Guibas. Optimally Combining Sampling Techniques for Monte Carlo Rendering. In *Computer Graphics Proceedings, Annual Conference Series, 1995 (ACM SIGGRAPH '95 Proceedings)*, pages 419–428, 1995.

[VG97] Eric Veach and Leonidas J. Guibas. Metropolis light transport. *Computer Graphics*, 31(3A):65–76, August 1997.

[WCG87] John R. Wallace, Michael F. Cohen, and Donald P. Greenberg. A two-pass solution to the rendering equation: A synthesis of ray tracing and radiosity methods. *Computer Graphics*, 21(4):311–320, July 1987.

[Whi80] T. Whitted. An improved illumination model for shaded display. *Communications of the ACM*, 23(6):343–349, 1980.

[Zat93] H. R. Zatz. Galerkin radiosity: A higher order solution method for global illumination. In *Computer Graphics Proceedings, Annual Conference Series, 1993*, pages 213–220, 1993.

Figure 3: Radiosity solution. No caustics or transparency, important transport paths are missing

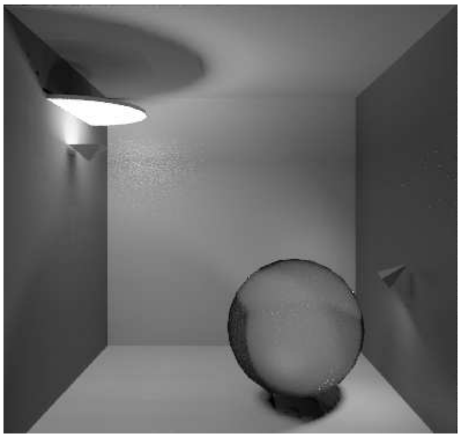

Figure 4: Radiosity with path tracing as a second pass. The sphere looks transparent but still no caustics are present (9 samples/pixel)

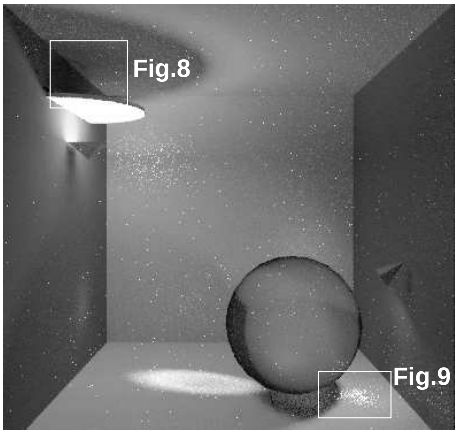

Figure 5: Radiosity combined with bidirectional path tracing (4 samples/pixel for all BPT images, same amount of work as in figure 4)

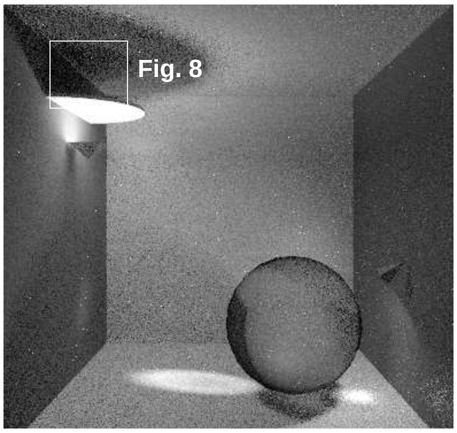

Figure 6: Bidirectional path tracing alone. Parts with important diffuse interreflection are much noisier than in figure 5

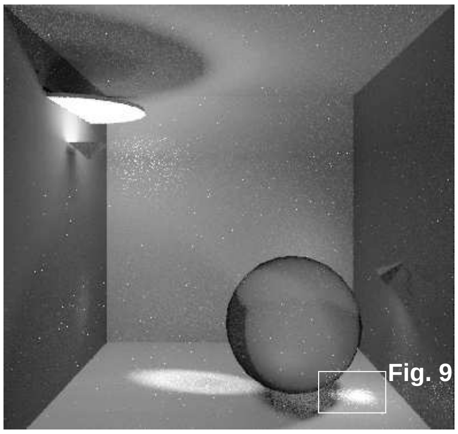

Figure 7: Radiosity and Bidirectional path tracing with caustic optimisation

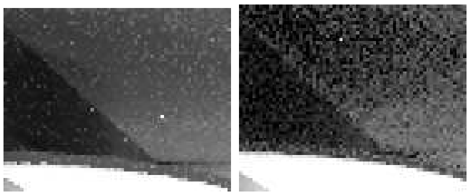

Figure 8: Close-up of upper left corner of figure 5 and figure 6. Note the noisier result in the right image.

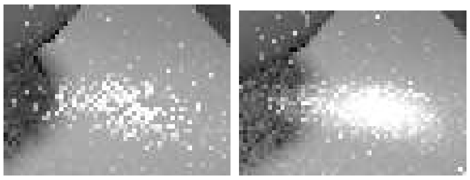

Figure 9: Close-up of 'diffuse' caustic of figure 5 and figure 7. Note the noisier result in the left image.