BI-DIRECTIONAL PATH TRACING

Eric P. Lafortune, Yves D. Willems

Department of Computing Science
Katholieke Universiteit Leuven
Celestijnenlaan 200A, 3001 Leuven, Belgium
Eric.Lafortune@cs.kuleuven.ac.be

ABSTRACT

In this paper we present a new Monte Carlo rendering algorithm that seamlessly integrates the ideas of shooting and gathering power to create photorealistic images. The algorithm can be explained as a generalisation of the well-known path tracing algorithm. Test results show that it performs significantly better for typical indoor scenes where indirect lighting is important.

Key Words: Rendering and visualisation; global illumination and photorealistic rendering.

INTRODUCTION

Both the eye point of the viewer and the set of primary light sources in a scene have always been identified as being important for solving the global illumination problem to create a realistic rendering. Some algorithms such as ray tracing are entirely built around the importance of the viewing point. Other algorithms such as the progressive radiosity method put a great emphasis on the contributions of the light sources. Ideally one would want an algorithm which takes into account the importance of both the light sources and the viewing point. In this paper we will present a new Monte Carlo algorithm which treats light sources and the viewing point on an equal basis.

RELATED WORK

An important milestone in the development of the global illumination theory for computer graphics was the introduction of the radiosity method by Goral et al. [Goral *et al.*84]. Originally developed within the field of heat transfer it is based on the energy equilibrium between diffuse emitters and reflectors of radiative energy such as heat or light. It requires the scene to be discretised into patches or elements and as such it is a finite element method. The radiosity solution is view-independent; the solution does not take into account the eventual viewpoint. Smits et al. [Smits *et al.*92] introduced the notion of importance and adapted the radiosity algorithm to tune the solution towards the final rendering. Their progressive radiosity algorithm shoots light from the light sources and importance from the viewpoint.

In [Kajiya86] Kajiya presented the rendering equation and introduced path tracing as a Monte Carlo algorithm to solve it. The idea is to sample the flux through the pixels, gathering light by following all light paths back to the light sources. As such it is entirely view-dependent. Various other algorithms are based on the same principle [Cook *et al.*84, Shirley90]. Monte Carlo techniques are capable of handling the most general class of lighting effects but are generally slow to converge.

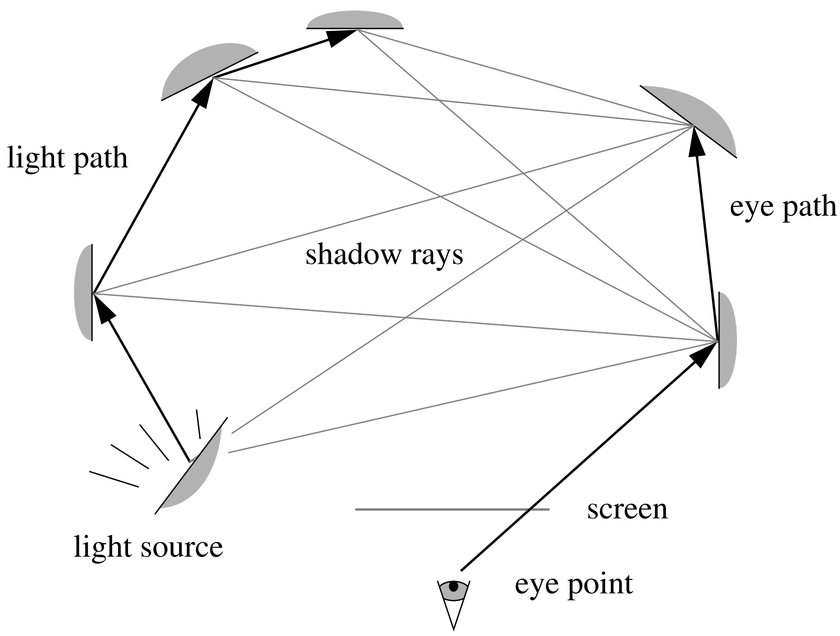

Figure 1: A schematic representation of the bi-directional path tracing algorithm.

Most research on the global illumination problem is currently directed towards two-pass methods [Chen *et al.*91]. These compute diffuse lighting components in a first extended progressive radiosity pass and specular lighting components in a second view-dependent pass. The diffuse components are computed using a deterministic method [Wallace *et al.*87, Ward *et al.*88, Sillion-Puech89, Lange91, Pattanaik-Mudur92] or a Monte Carlo technique [Feda-Purgathofer93, Pattanaik93]. The final image including the specular reflections is usually rendered using some variant of distribution ray tracing or path tracing. Some algorithms attempt to reconstruct not only the diffuse component but also the directional component of the emitted light in order to find a completely view-independent solution of the problem [Immel *et al.*86, Sillion *et al.*91]. The main problem with the latter approach is the huge amount of storage that is required to represent the lighting function.

BI-DIRECTIONAL PATH TRACING

The algorithm we present differs from distribution ray tracing or path tracing in its computation of a primary estimator for the flux through each pixel. The basic idea is that particles are shot at the same time from a selected light source and from the viewing point, in much the same way. All hit points on the respective particle paths are then connected using shadow rays and the appropriate contributions are added to the flux of the pixel in question (Fig. 1). Using this approach various lighting contributions are taken into account, not only from primary light sources but in a probabilistic way also from important secondary, tertiary, etc. light sources.

We will now present in greater detail which probability distribution functions (*pdf*s) govern the random walks and how the effective contributions to the flux are then calculated. It can be proven mathematically that these lead to a correct solution of the rendering equation or its adjoint formulation.

Performing the random walks

As shown in Fig. 2 the random walks can be written as:

- $x_0, x_1, x_2, \ldots, x_{N_l}$ for the light path, where x_{i+1} is the point seen by point x_i along direction Θ_{x_i}, and
- $y_0, y_1, y_2, \ldots, y_{N_e-1}$ for the eye path, where y_{j+1} is the point that sees point y_j along direction $\Theta_{y_{j+1}}$.

The initial points and directions of the random walks have to be selected first. Using the

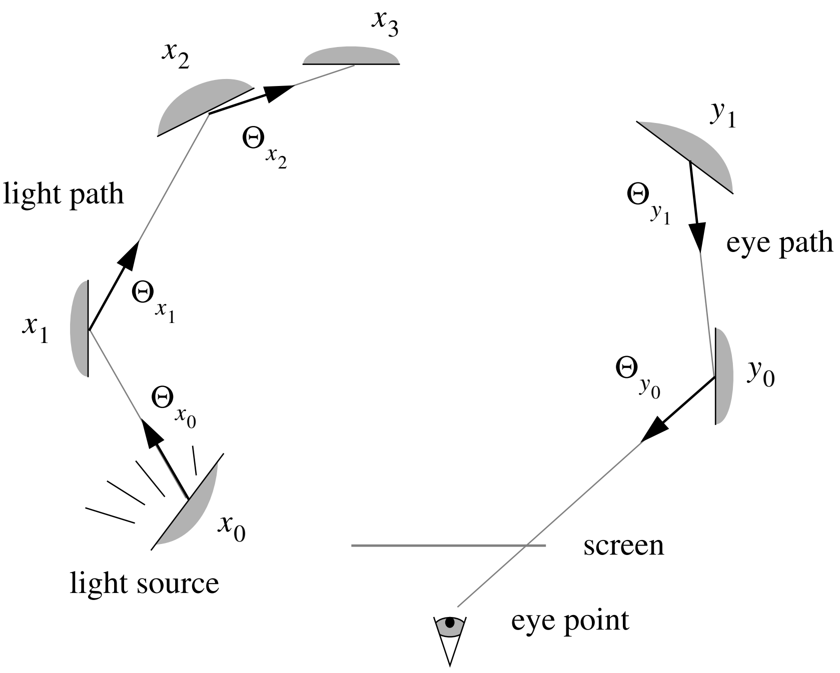

Figure 2: Naming conventions for $x_0, \Theta_{x_0}, \ldots, \Theta_{x_{N_l-1}}$ and $y_0, \Theta_{y_0}, \ldots, \Theta_{y_{N_e-1}}$ to denote the light path and the eye path respectively. In this example N_l equals 3 and N_e equals 2.

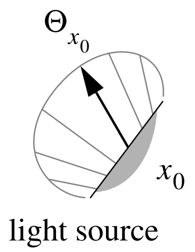

Figure 3: Sampling x_0 and Θ_{x_0} according to the self-emitted radiance of light sources.

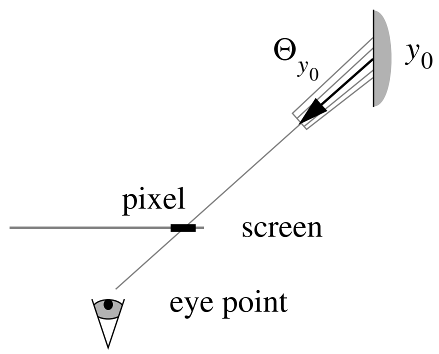

Figure 4: Sampling y_0 and Θ_{y_0} with respect to the pixel under consideration.

naming conventions above we do this for the eye path by sampling x_0 and Θ_{x_0} according to the following *pdf*, which is based on the principle of importance sampling (Fig. 3):

$$pdf(x, \Theta_x) = \frac{L_e(x, \Theta_x)\,|\Theta_x \cdot N_x|}{L} \qquad (1)$$

with the normalisation factor of the *pdf*

$$L = \int_A \int_{\Omega_x} L_e(x, \Theta_x)\,|\Theta_x \cdot N_x|\, d\omega_x d\mu_x$$

where $L_e(x, \Theta_x)$ is the self-emitted radiance at point x in direction Θ_x) and where $|\Theta_x \cdot N_x|$ is the absolute value of the cosine of the angle between Θ_x and the normal vector at x. The importance sampling ensures that more light particles are shot from bright emitters and in bright directions, instead of distributing them uniformly and weighting their contributions to the flux later on.

Similarly we select samples y_0 and Θ_{y_0} for the eye path according to the following *pdf* (Fig. 4):

$$pdf(y, \Theta_y) = \frac{g(y, \Theta_y)\,|\Theta_y \cdot N_y|}{G} \qquad (2)$$

with the normalisation factor of the *pdf*

$$G = \int_A \int_{\Omega_y} g(y, \Theta_y)\,|\Theta_y \cdot N_y|\, d\omega_y d\mu_y$$

where $g(y, \Theta_y)$ the function is 1 for all pairs of points and directions (y, Θ_y) on the surfaces that contribute to the flux, and 0 otherwise.

Once the initial points and directions have been chosen the rest of the random walks is determined by sampling the directions $\Theta_{x_{i+1}}$ and $\Theta_{y_{j+1}}$ respectively. These variables determine x_{i+1} and y_{j+2} unambiguously.

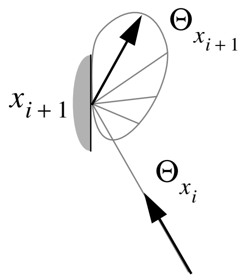

Figure 5: Sampling $\Theta_{x_{i+1}}$ according to the *brdf* of the surface at point x_{i+1} and the incoming direction Θ_{x_i}.

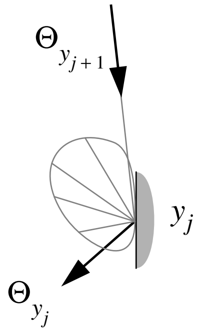

Figure 6: Sampling $\Theta_{y_{j+1}}$ according to the *brdf* of the surface at point y_j and the incoming direction Θ_{x_j}.

The *subcritical pdf* (*spdf*) for $\Theta_{x_{i+1}}$ is chosen as follows, again on the basis of importance sampling (Fig. 5):

$$pdf(\Theta) = f_r(x_{i+1}, \Theta_{x_i}, \Theta) \left|\Theta_x \cdot N_{x_{i+1}}\right| \quad (3)$$

where $f_r(x, \Theta_{in}, \Theta_{out})$ is the bi-directional reflection distribution function (*brdf*). The *pdf* is subcritical because it does not integrate to 1 over all possible angles , at least for physically valid *brdf*s. The actual value of the integration gives the chance that the random walk is continued, which ensures that the random walk terminates. This technique is commonly called Russian roulette.

The subcritical *pdf* for $\Theta_{y_{j+1}}$ is chosen in the same way (Fig. 6):

$$pdf(\Theta) = f_r(y_j, \Theta, \Theta_{y_j}) \left|\Theta_y \cdot N_{y_j}\right| \quad (4)$$

The random walks are independent of one another. The *spdf*s for the directions can be made identical simply by renaming the variables and using the bi-directional property of the *brdf*. This property implies that after the initialisation both random walks can be performed by a single algorithm.

Estimating the flux

Now that we have shown how the stochastic variables for the integrals and for the random walks are selected we will actually evaluate the final result. For this purpose all points on the respective random walks are linked using shadow rays. The primary estimator for the flux can then be derived as a sum of weighted partial estimates:

$$\langle \Phi \rangle = \sum_{i=0}^{N_l} \sum_{j=0}^{N_e} w_{ij} \langle C_{ij} \rangle \quad (5)$$

The factors $\langle C_{ij} \rangle$ express the estimates of the flux found by i reflections on the light path and j reflections on the eye path. Three cases have to be distinguished when evaluating them:

- $i = 0,\ j = 0 : \langle C_{00} \rangle = G \times L_e(y_0, \Theta_{y_0})$

  This term is an estimate for the flux received from a light source that is directly seen through the pixel under consideration (Fig. 7).

- $i = 0,\ j > 0 : \langle C_{0j} \rangle =$

  $$L' \times G \times L_e(x_0, \Theta_{x_0 \to y_{j-1}}) \times f_r(y_{j-1}, \Theta_{x_0 \to y_{j-1}}, \Theta_{y_{j-1}}) \times \frac{\left|\Theta_{x_0 \to y_{j-1}} \cdot N_{x_0}\right| \left|\Theta_{x_0 \to y_{j-1}} \cdot N_{y_{j-1}}\right|}{\|x_0 - y_{j-1}\|^2}\, v(x_0, y_{j-1})$$

  with $L' = \frac{L}{\int_{\Omega_x} L_e(x_0, \Theta_x) \left|\Theta_x \cdot N_{x_0}\right| d\omega_x}$,

  where $\Theta_{x \to y}$ is the direction from point x to point y and where $v(x, y)$ is the visibility function which is 1 if point x sees point y and 0 otherwise. This value is found by means of the shadow ray between point x and point y.

  The term is an estimate for the flux that reaches the eye from the light source through the eye path, as in classical path tracing (Fig. 8).

- $i > 0,\ j > 0 : \langle C_{ij} \rangle =$

  $$L \times G \times f_r(x_i, \Theta_{x_{i-1}}, \Theta_{x_i \to y_{j-1}}) \times f_r(y_{j-1}, \Theta_{x_i \to y_{j-1}}, \Theta_{y_{j-1}}) \times \frac{\left|\Theta_{x_i \to y_{j-1}} \cdot N_{x_i}\right| \left|\Theta_{x_i \to y_{j-1}} \cdot N_{y_{j-1}}\right|}{\|x_i - y_{j-1}\|^2}\, v(x_i, y_{j-1})$$

  This term is an estimate for the flux that reaches the eye from the light source, through i reflections on the light path and j reflections on the eye path (Fig. 9).

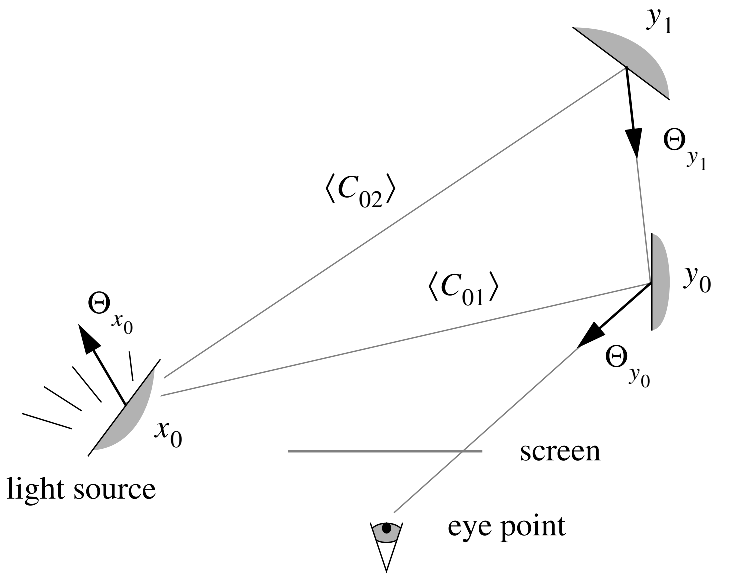

Figure 8: The contribution $\langle C_{0j} \rangle$ is an estimate for the flux that reaches the eye from the light source through the eye path.

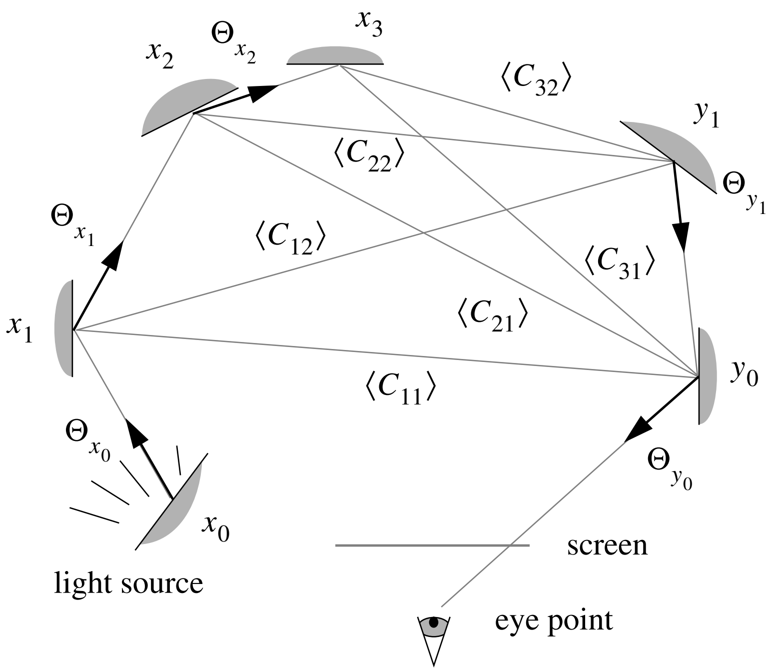

Figure 9: The contribution $\langle C_{ij} \rangle$ is an estimate for the flux that reaches the eye through both the light path and the eye path.

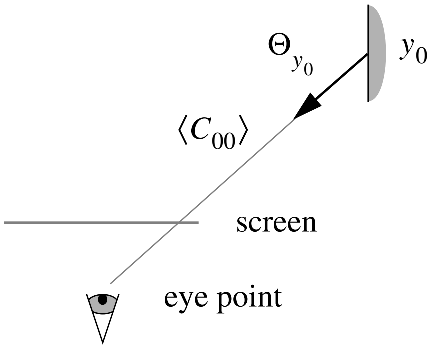

Figure 7: The contribution $\langle C_{00}\rangle$ is an estimate for the flux emitted by a light source that is seen directly by the viewer.

Selecting the weights

At this point the algorithm is still generic as there are various alternatives to choose the weights w_{ij} for the contributions $\langle C_{ij}\rangle$, as long as they comply with the condition that $\sum_{i=0}^{N} w_{i,N-i} = 1$ $(N = 0, 1, \ldots)$. This condition can be derived theoretically; its physical meaning is that the sets of weights for the estimates of the fluxes arriving at the eye via one, two, etc. reflections respectively all have to add up to 1. One can verify that the following instantiation yields the classical path tracing algorithm:

$$\begin{aligned} w_{ij} &= 1 \text{ for } i = 0, \text{ and} \\ &= 0 \text{ otherwise.} \end{aligned}$$

This selection does not fully use the information of the sampling process however. The following alternative uses both particle paths more effectively:

$$\begin{aligned} w_{ij} &= \prod_{k=0}^{j-2} W_k \text{ for } i = 0, \\ &= 0 \qquad\quad \text{for } j = 0, \text{ and} \\ &= (\prod_{k=0}^{j-2} W_k)(1 - W_j) \text{ otherwise.} \end{aligned}$$

where the weights W_j $(j = 0, \ldots, N_e - 1)$ still leave some degrees of freedom.

The idea behind this particular choice is that at each point on the eye path the estimates for the indirect lighting via the rest of the eye path and via the light path are weighted (Fig. 10). For specular surfaces one would rather rely on the estimate found by following the eye path. For diffuse surfaces the estimate found through the contributions

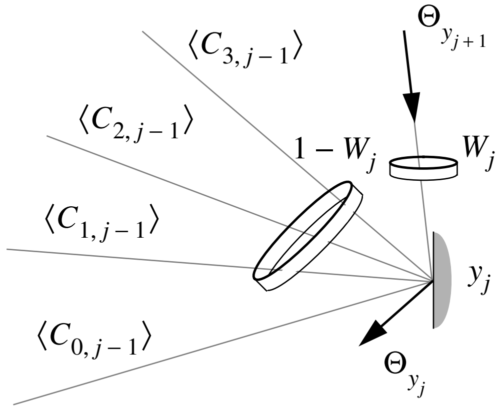

Figure 10: The weights W_j for the contributions via the eye path are selected proportional to the degree of specularity at point y_j. The contributions arriving from the light path via the shadow rays have the weight $1 - W_j$.

of the light path is more likely to be important. Therefore the weight W_j is chosen proportional to a measure of the degree of specularity of the surface at point y_j on the eye path. For highly specular surfaces it approaches 1, for diffuse surfaces it goes to 0. Tests on practical scenes have shown that this technique greatly improves the quality of the images, especially when rendering scenes containing mirrors.

Computing a secondary estimator

The primary estimator of the flux will still have a large variance, which will clearly show up in the image under the form of random noise. As with all Monte Carlo methods a secondary estimator is therefore computed by averaging the results of several primary estimators for a single pixel. The variance of the eventual result will be reduced by a factor $\sqrt{N}$ if N is the number of primary samples. For path tracing N reportedly typically ranges between 40 and 500, but the optimal number largely depends on the complexity of the scene and the desired accuracy. Some heuristic adaptive sampling techniques are usually applied to find a balance between the computational work and the quality of the results.

IMPLEMENTATION

We have implemented the bi-directional path tracing algorithm as described above. The program has been written in the programming language C on an IBM RS/6000-320. It is based on the library routines of the public domain ray tracing program 'Rayshade'.

The *brdf*s have been modeled using a Phong model which has been modified slightly as to make it reciprocal and energy-conserving. It allows the few constants in the equations and the *pdf*s to be computed analytically. More complicated models will in general require numerical techniques.

Several optimised sampling strategies have been implemented. Importance sampling and Russian roulette have been applied as explained in the previous paragraphs. Furthermore stratified sampling has been used. This technique consists in subdividing the sampling intervals and selecting samples from each of these, instead of just selecting all samples randomly over the whole interval. The uniform samples selected in this way may be transformed into non-uniform samples, so that the technique is combined easily with importance sampling.

RESULTS

Based on the implementation we have performed some tests, comparing our bi-directional path tracing algorithm with classical path tracing. The largest amount of work in both algorithms consists in performing ray intersection tests. So in order to obtain a fair comparison approximately the same numbers of rays are used by the respective algorithms in each test. Table 1 gives an overview of the results. All images have been rendered at a resolution of 400×400 pixels. Both implementations use the optimised sampling strategies such as importance sampling, Russian roulette and stratified sampling. Neither of the algorithms performs adaptive sampling of the pixels.

The scene consists of coloured diffuse walls, a slightly specular floor and a mixture of opaque and transparent objects. Both algorithms accurately render typical global illumination effects such as diffuse and glossy reflections, soft shadows and colour bleeding. The results for the directly illuminated scene show little difference between them. For the indirectly illuminated scene however our bi-directional algorithm produces visibly less noise for the same amount of work (it is even noticeable on the printed images, in spite of the dithering applied to produce different colour shades). An intuitive explanation is that the light particle paths help the indirect light to meet the eye paths halfway, thereby producing more reliable estimates and less noise.

CONCLUSION

We have presented bi-directional path tracing as a new Monte Carlo algorithm for physically-based rendering. It can be explained in a general theoretical framework in which the existing path tracing algorithm is a special case.

- Similarly to other Monte Carlo techniques the algorithm is very general: it can handle extensive classes of geometrical objects and optical properties. Diffuse lighting effects, soft shadows, specular and glossy reflections and refractions, and if required even depth of field and motion blur are all simulated correctly. Anti-aliasing is integrated in a natural way.

- Experiments show that the algorithm performs better than path tracing for typical indoor scenes where indirect illumination is important.

- The method requires no meshing and thus avoids all the associated problems. Also because of this the method requires little memory. The description of the scene is accessed in a read-only fashion.

- Importance sampling is used extensively, reducing the variance drastically. Without adaptive sampling however convergence is rather slow. This is partly because the exact solution is sought for each pixel, as opposed to finite element methods where the eventual solution is interpolated over larger

Test	Algorithm	Number of samples per pixel	Total number of rays	Time (sec)
1	Path tr.	60	45×10^6	15 897
2	Bi-dir. path tr.	20	30×10^6	11 036
3	Path tr.	60	35×10^6	18 251
4	Bi-dir. path tr.	20	31×10^6	14 929

Table 1: Overview of the test results.

surface areas most of the time. Adaptive and hierarchical techniques and filtering should help to find a proper balance between 'oversolving' and 'undersolving' in the future.

REFERENCES

Chen, S.E., Rushmeier, H.E., Miller, G., Turner, D. 1991. A progressive multi-pass method for global illumination. In *Computer Graphics*, volume 25, pages 164–174.

Cook, R.L., Porter, T., Carpenter, L. 1984. Distributed ray tracing. In *Computer Graphics*, volume 18, pages 137–145.

Feda, M., Purgathofer, W. 1993. Progressive ray refinement for monte carlo radiosity. In *Proceedings of the Fourth Eurographics Workshop on Rendering*, pages 15–25, Paris, France.

Goral, C.M., Torrance, K.E., Greenberg, D.P., Battaile, B. 1984. Modelling the interaction of light between diffuse surfaces. In *Computer Graphics*, volume 18, pages 213–222.

Immel, D.S., Cohen, M.F., Greenberg, D.P. 1986. A radiosity method for non-diffuse environments. In *Computer Graphics*, volume 20, pages 133–142.

Kajiya, J.T. 1986. The rendering equation. In *Computer Graphics*, volume 20, pages 143–150.

Lange, B. 1991. The simulation of radiant light transfer with stochastic ray-tracing. In *Proceedings of the Second Eurographics Workshop on Rendering*, Barcelona, Spain.

Pattanaik, S.N., Mudur, S.P. 1992. Computation of global illumination by monte carlo simulation of the particle model of light. In *Proceedings of the Third Eurographics Workshop on Rendering*, pages 71–83, Bristol, UK.

Pattanaik, S.N. 1993. *Computational Methods for Global Illumination and Visualisation of Complex 3D Environments*. PhD thesis, Birla Institute of Technology & Science, Pilani, India.

Shirley, P. 1990. *Physically Based Lighting Calculations for Computer Graphics*. PhD thesis, University of Illinois.

Sillion, F., Arvo, J.R., Westin, S.H., Greenberg, D.P. 1991. A global illumination solution for general reflectance distributions. In *Computer Graphics*, volume 25, pages 187–196.

Sillion, F., Puech, C. 1989. A general two-pass method integrating specular and diffuse reflection. In *Computer Graphics*, volume 23, pages 335–344.

Smits, B.E., Arvo, J.R., Salesin, D.H. 1992. An importance-driven radiosity algorithm. In *Computer Graphics*, volume 26, pages 273–282.

Wallace, J.R., Cohen, M.F., Greenberg, D.P. 1987. A two-pass solution to the rendering equation: a synthesis of ray tracing and radiosity methods. In *Computer Graphics*, volume 21, pages 311–320.

Ward, G.J., Rubinstein, F.M., Clear, R.D. 1988. A ray tracing solution for diffuse interreflection. In *Computer Graphics*, volume 22, pages 85–92.