Weighted Multi-pass Method based on Stochastic Iteration and Random Walk

György Antal, László Szirmay-Kalos, Ferenc Csonka

Department of Control Engineering and Information Technology
Technical University of Budapest
Budapest, Pázmány Péter s. 1/D, H-1117
Hungary
agy@inf.elte.hu

ABSTRACT

Multipass algorithms implement different rendering methods and combine their results. If carefully implemented, these algorithms can keep the advantages of individual methods. Algorithms for multipass global illumination may handle disjoint parts of light transport paths in different passes, and include all light transport only once in the final solution. On the other hand, they may also generate the same paths, thus their contribution should be weighted in order to get an unbiased estimation. In this paper a weighted combination of global ray bundle iteration and path tracing is presented. Heuristics for the weights are derived to get the benefits from both approaches. Results show significant improvements compared to both ray bundle iteration and path tracing.

Keywords: Global illumination, stochastic iteration, multipass algorithm

1 INTRODUCTION

Global illumination algorithms determine the radiance distribution arriving at the eye. They build up light paths connecting light sources to the eye through one or more reflections and add their contribution. The radiance arriving at the camera through a pixel is determined by:

$$L(x_1, \omega_p) = L^e(x_1, \omega_p) + \int l(x_1, \omega_1, \ldots, \omega_n, \ldots) d\omega_1 \ldots d\omega_n \ldots \qquad (1)$$

where $l(x_1, \omega_1 \ldots, \omega_n, \ldots)$ is the contribution of a single light path defined by directions $(\omega_1, \ldots, \omega_n, \ldots)$. Considering the exploration of the space of paths, global illumination algorithms can follow two strategies. One approach is the *depth-first strategy*, which usually results in some kind of random walk algorithms. This can also be classified further into gathering or shooting, depending on if the walk is originated from the eye (path tracing) or from the light sources (light tracing). Another approach is the *breadth-first strategy*, where in a single step all paths are advanced simultaneously. These techniques are based on the iteration solutions of the rendering equation.

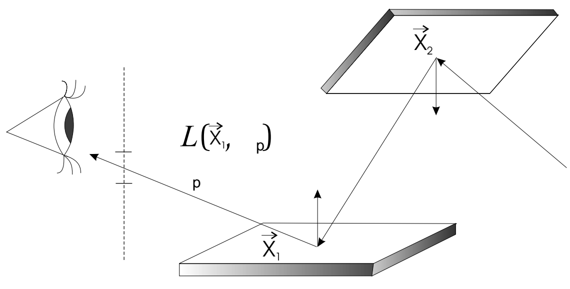

Figure 1: Light path arriving at the camera

Within this context, different path building strategies have been published, and each of them is good for certain path types. It can hardly be expected that a completely different path construction algorithm will appear, and overcome all other methods. Thus, instead of developing a new algorithm, it is worth mixing existing algorithms of different strengths and weaknesses together in order to combine their merits. One must be careful when performing this combination. As Cohen and Wallace stated: "When combining algorithms in two-pass or multipass approaches, care must be taken to avoid counting the same paths more than once" [Cohen93]. There are three possible approaches to handle this problem.

The first family of multipass methods works by completing incomplete light paths. Usually a partial light path exploration is performed in a global preprocessing step, which is completed by a final gather. Such algorithm was presented by Wallace and Cohen [Cohen87], and was generalized in [Sillion89], where a radiosity method was extended to handle perfect mirrors. This idea was also applied later in [Shirley90] [Chen91] [Bouatouch92] [Jensen96]. A common feature of these approaches is that they use an object space rendering algorithm, like e.g. radiosity (or photon map) to calculate the diffuse reflections very efficiently. They may take a light tracing pass, which can handle caustics quite well, and then they usually use a path tracing step to read out the predetermined partial radiance solutions computed in the previous passes.

Members of the second category of multipass approaches combine different algorithms that construct different, but complete light paths [Granier00]. The strategies handle perfectly disjunct parts of the path domain. Therefore, their combination (which is just an arithmetic addition in this situation) does not count the same path more than once.

Approaches of the third type work by combining strategies, which cover the same complete light paths. To avoid counting the same path more than once, the contribution of different strategies should be weighted. This approach was first suggested by Lafortune [Lafortune93] for bi-directional path tracing. Later a strict mathematical groundwork was presented by Veach [Veach95], who showed how the weighted combination of different sampling techniques can produce a better, but still unbiased result. Based on these results a weighted multipass framework was proposed by Suykens and Willems [Suykens99] and was successfully applied for the combination of radiosity and bi-directional path tracing. Compared to bi-directional path tracing, this combination is more difficult since it mixes completely different strategies.

In this paper we present a new weighted multipass method belonging to the third category. The combined algorithms are the global ray bundle tracing and the path tracing. The combination is performed in a way that the benefits of the passes are preserved.

2 WEIGHTED MULTIPASS

The radiance, obtained as the solution of the global illumination problem, can be written as an integral over all covered paths [Spanier69], [Pauly00]:

$$L = \int_P l(z)dz. \tag{2}$$

where $z = (\omega_1, \omega_2 \ldots)$ is a light path and P is the space of path of all length.

Monte-Carlo global illumination algorithms generate paths with certain probability density $p_i(z)$. For a given path z, denote the primary estimator of the method i by $l(z)/p_i(z)$. Then the secondary estimator is:

$$\langle L_p \rangle_i = \frac{1}{N_i} \sum_{j=1}^{N_i} \frac{l(z_{i,j})}{p_i(z_{i,j})}$$

where path $z_{i,j}$ is sampled using p_i and N_i is the number of samples. These estimators compute the same integral. The variance decreases as p_i mimics l more accurately[Spanier69], [Owen99] (importance sampling). For some subdomain of P, p_i could be more efficient than p_k, or vice versa. For example when calculating caustics, light source sampling strategy is more practical. However, when seeing through a mirror, pixel sampling followed by BRDF sampling is more suitable.

Therefore we would like to combine the estimators using weighting functions $w_i(z)$ which keeps the strengths of each estimator:

$$\langle L \rangle_c = \sum_{i=1}^{n} \frac{1}{N_i} \sum_{j=1}^{N_i} w_i(z_{i,j}) \frac{l(z_{i,j})}{p_i(z_{i,j})}. \tag{3}$$

The combined estimator should be unbiased in order to get correct results. This requirement is satisfied if its expected value is equal to the estimated integral.

Denote the total number of samples by $N = \sum_{i=1}^{n} N_i$, where the N_i values are fixed before any samples are taken. The "average probability density" of selecting sample z is then

$$\widehat{p}(z) = \sum_{i=1}^{n} \frac{N_i}{N} \cdot p_i(z).$$

Thus the integral quadrature using these samples is

$$\int_P l(z)dz = \int_P \frac{l(z)}{\widehat{p}(z)} \cdot \widehat{p}(z)dz \approx \frac{1}{N} \sum_{i=1}^{n} \sum_{j=1}^{N_i} \frac{l(z_{i,j})}{\widehat{p}(z_{i,j})}$$

$$= \sum_{i=1}^{n} \frac{1}{N_i} \sum_{j=1}^{N_i} w_i(z_{i,j}) \cdot \frac{l(z_{i,j})}{p_i(z_{i,j})}.$$

where $z_{i,j}$ is the jth sample taken from the ith distribution, and the weights[1] are

$$w_i(z) = \frac{N_i \cdot p_i(z)}{\sum_{k=1}^{n} N_k \cdot p_k(z)}. \qquad (4)$$

In order to have an unbiased estimation $\sum_{i=1}^{n} w_i(z) = 1$ should hold for all z. It means that when combining different algorithms we have to calculate the probabilities that a given path is obtained by each of the methods.

3 Path Tracing

The path tracing is based on the Neumann series solution of the rendering equation:

$$L = \sum_{i=0}^{\infty} \mathcal{T}^i L^e. \qquad (5)$$

The mathematical basis of this algorithm was presented by Kayija [Kajiya86] as a generalization of distributed ray tracing. It is an image based algorithm, which recursively samples random directions $\omega_1', \omega_2', ..., \omega_n'$ to follow the light paths backward and the emission of all visited points are gathered and transferred to the eye. These walks provide the value of the integrand at "point" $z = (\omega_1', \omega_2', \ldots, \omega_n', \ldots)$. Note that a single walk can be used to estimate the 1-bounce, the 2-bounce, etc. n-bounce transfer simultaneously. The expansion results in an infinite Neumann series, which creates the problem of calculating an infinite dimensional integral. Practical implementations usually truncate the infinite Neumann series, which introduces some bias, or

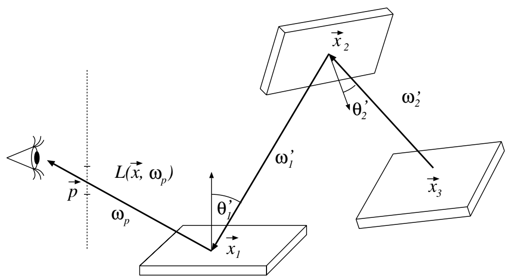

Figure 2: Path tracing

stop the walks randomly, which reduces the samples of higher order inter-reflections. This method called Russian roulette [Arvo90] gives a still unbiased estimation. It is straightforward that this random decision increases the variance of the estimator.

The strength of path tracing is that, since it generates directions according to a surface BRDF, it treats some specular or glossy ($L[D|G|S]^*[G|S]E$ paths) effects quite easily. The weakness of the algorithm is that, in spite of many advancements, building a single path is computationally expensive.

4 Global Ray Bundle Iteration

As proposed in [Szirmay99], [Neumann95] stochastic iteration using ray bundles can handle non-diffuse global illumination as well. Iteration algorithms are based on the fact that the solution of the rendering equation is the fixed point of the following iteration scheme:

$$L_{i+1} = L^e + \mathcal{T} L_i. \qquad (6)$$

The stochastic iteration is originated from equation (6), where the determininistic operator $\mathcal{T}$ is replaced by a random operator $\mathcal{T}^*$, which behaves as the original one in the average case.

$$L_{i+1} = L^e + \mathcal{T}^* L_i, \qquad E[\mathcal{T}^* L] = \mathcal{T} L.$$

In this stochastic iteration scheme, the radiance function and its functionals such as the image do not converge, but fluctuate around the real solution. However, the average of the values in subsequent steps will really converge to the required solution.

In ray bundle iteration the randomization happens by choosing a random direction using a uniform distribution, and the radiance of all patches

[1] One can recognize that Veach called these weights $w_i(z)$ balanced heuristics in [Veach95]. Nevertheless, balanced heuristics were introduced during the minimization of the variance of estimator (3), while we used another approach.

is transferred into this direction, then the transferred radiance is reflected towards the eye and to the next random direction. Note that the algorithm requires just one variable for each patch i, which stores the incoming irradiance $L[i]$. The radiance transfer is performed by a transillumination buffer or by a hardware supported Z-buffer algorithm [Szirmay98].

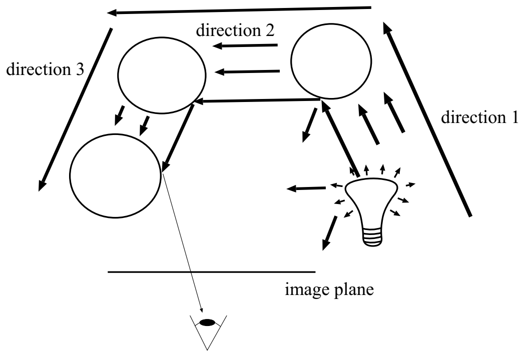

Figure 3: A path of ray bundles

The strength of ray bundle iteration is that the algorithm exploits coherence, therefore it is very fast. Combined with Gouraud or Phong shading the resulting pictures are not noisy. The weakness of the algorithm is that by sampling global directions uniformly over the bounding sphere, it cannot take into account locally important directions. For example, it is very unlikely to sample near ideal mirror reflection directions.

5 Calculating probabilities

According to equation (4), when determining the weight of a contribution of strategy i, we should be able to determine all $p_j(z)$ probabilities with which the sample z would be generated if strategy j was applied. It is relatively easy if the path building strategies are similar (e.g. bi-directional path tracing), but it is difficult when the strategies are very different [Suykens99] as happens in our case.

Focusing on the context of this paper, note that in equation (4) N_1 is the number of pixel samples in path tracing, and N_2 is the iteration number of the ray bundle iteration. Consider path $z = (\omega_1, \omega_2, \ldots)$. Note that it is a path of infinite length. The first question that has to be answered is what the probability of sampling this path by the two algorithm.

Two problems emerge when calculating integral (2) by Monte-Carlo methods, which result in estimations of the following form

$$\frac{1}{N}\sum_{i=1}^{N}\frac{l(z)}{p(z)}.$$

The first problem is that the integrand l is an infinite dimensional function. To evaluate this infinite dimensional function we sometimes truncate the variables of the function at a given depth n and ignore the steps after n. Thus instead of $l(\omega_1, \omega_2, \ldots \omega_n \ldots)$ we use $l(\omega_1, \omega_2, \ldots \omega_n)$. Another strategy is that we terminate the walk at a random depth and compensate for the missing steps. This is achieved in the context of Russian roulette, which at step i decides either to finish the walk with probability $1 - s_i$ and the contribution of the tail is considered to be zero, or to continue the walk and the contribution of the tail is multiplied by $1/s_i$.

Note that this means that when the Neumann series is truncated, then the probability of sampling all variables of the integrand is zero. Since multiple importance sampling uses this probability to weight the samples of a particular method, the truncation results in zero weight, which is obviously not acceptable. To overcome this problem, we shall assume that when Russian roulette terminates the walk, it actually generates samples for all integrand variable vectors where the first variables correspond to the really computed part of the walk. The question is then what kind of probability density can be assumed for the not sampled variables. Since we do not have any information, we shall assume that this probability density is uniform. Note that this approach emphasizes samples obtained with Russian roulette more than they deserve. However, this is still better than not taking them into account at all.

The second problem with equation (2) is that the integration domain is of infinite volume. Thus any pdf function that is defined over this domain (because it should be non-negative and integrated to 1) goes to zero. E.g. the pdf of a uniformly sampled direction is $\frac{1}{2\pi}$. Over an infinite dimensional domain the pdf $(\frac{1}{2\pi})^n$ becomes zero. Thus using $p(z)$ in the Monte-Carlo estimation raises computational problems. This problem is solved by transforming the domain of the paths onto the unit cube. Since a path $z = (\omega_1, \omega_2, \ldots \omega_n \ldots)$ is defined by a sequence of directions, we discuss this transformation for a single direction in this sequence. Of course, the same procedure should be repeated at all reflection points. Let us thus consider a single directional integral of the multi-directional integral on the path space and express the direction by angles of a spherical coordinate system:

$$\int_{\Omega} l(\omega)\ d\omega = \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi/2} l(\phi,\theta) \cdot \sin\theta\ d\phi d\theta.$$

Let us now transform the domain of the angles onto the unit interval using the following transformation:

$$\phi = 2\pi \cdot u, \qquad \theta = \arccos v.$$

The inverse transformation is

$$u = \frac{\phi}{2\pi}, \qquad v = \cos\theta,$$

thus the directional integral can be written in the following form as an integral over the unit square:

$$\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi/2} l(\phi,\theta) \cdot \sin\theta\ d\phi d\theta = \int_{u=0}^{1} \int_{v=0}^{1} \frac{l(u,v)}{2\pi}\ dudv.$$

Note that we could use other transformations as well, which would map the space of directions onto the unit integral. However, this particular formulation will be very convenient when the probabilities of paths in ray bundle iteration and path tracing are computed.

Different algorithms use particular densities $p(\omega)$ to obtain random directions according to, for example, importance sampling. This means that our new u, v variables are also sampled from density $p_{uv}(u,v)$ that should correspond to the probability density of the directions. Using the proposed transformations also for the densities, we obtain:

$$p_{uv}(u,v) = p(\omega) \cdot \frac{d\omega}{dudv} =$$
$$p(\omega) \cdot \sin\theta \cdot det \begin{bmatrix} \frac{\partial\phi}{\partial u} & \frac{\partial\phi}{\partial v} \\ \frac{\partial\theta}{\partial u} & \frac{\partial\theta}{\partial v} \end{bmatrix} = p(\omega) \cdot 2\pi. \quad (7)$$

5.1 The pdf for ray bundle iteration

In ray-bundle iteration we transfer the energy of the patches parallel to a uniformly sampled direction in both ways. [Szirmay99]. Using this strategy, we always continue the walk. According to equation (7), the direction is sampled uniformly when u, v parameters have uniform distribution in the unit square, thus the probability density used in a single step of ray-bundle iteration is

$$p_{uv}(u,v) = 1.$$

Since the directions of the different iteration steps are independent, the density of a path will be the product of these 1 values, which means that the probability density of an arbitrary path is 1.

5.2 The pdf of path tracing pass

When we perform path tracing, importance sample is performed, thus $p(\omega)$ is not uniform, but rather mimics the local BRDF and the cosine of the angle between the generated direction and the surface normal. Density $p_{uv}(u,v)$ is calculated from $p(\omega)$ according to equation (7).

For example, if the surface is diffuse, BRDF sampling using the $p(\omega) = \cos\theta/\pi$ density, thus the transformed probability is:

$$p_{uv}(u,v) = \frac{\cos\theta}{\pi} \cdot 2\pi = 2 \cdot \cos(\arccos v) = 2v.$$

Since path tracing uses Russian roulette, it does not compute all reflections with the same probability as happens theoretically in iteration. Suppose that in the first step we decide whether or not to continue the path with probability s_1. When the integral is estimated with no sample, we assume that this corresponds to sampling uniformly as it was discussed before in this section. Thus when terminating after the first step, the approximated pdf becomes:

$$p_1^*(u_1,v_1) = s_1 \cdot p_{uv}(u_1,v_1) + (1-s_1).$$

Similarly the approximated pdf after the second step is:

$$p_2^*(u_1,v_1,u_2,v_2) = s_1 \cdot p_{uv}(u_1,v_1) \cdot s_2 \cdot p_{uv}(u_2,v_2) +$$
$$s_1 \cdot p_{uv}(u_1,v_1) \cdot (1-s_2) + (1-s_1). \quad (8)$$

According to equation (4) we must compute p^* both in path tracing and ray bundle iteration. The difference between them is that ray bundle iteration follows the light paths from the light sources, while the path tracing traces light backwards from the eye. Both algorithms are partitioned into steps, thus deriving a recursive formula for calculating p^* is straightforward.

Since in path tracing the paths are built up from the eye, the formula of p^* is divided into two parts ($p^* = q + r$), and the recursive form becomes:

$$\begin{aligned} q_0 &= 1, \qquad r_0 = 0, \\ q_i &= s_i \cdot p_{uv}(u_i,v_i) \cdot q_{i-1}, \\ r_i &= (1-s_i) \cdot q_{i-1} + r_{i-1}. \end{aligned} \quad (9)$$

where q is the probability of unterminated path, while r is the probability of those paths that were terminated.

Since ray bundle iteration builds up path from the light source, the calculation of p^* is more straightforward:

$$\begin{aligned} p_1^* &= s_1 \cdot p_{uv}(u_1,v_1) + 1 - s_1, \\ p_i^* &= s_i \cdot p_{uv}(u_i,v_i) \cdot p_{i-1}^* + (1-s_i). \end{aligned} \quad (10)$$

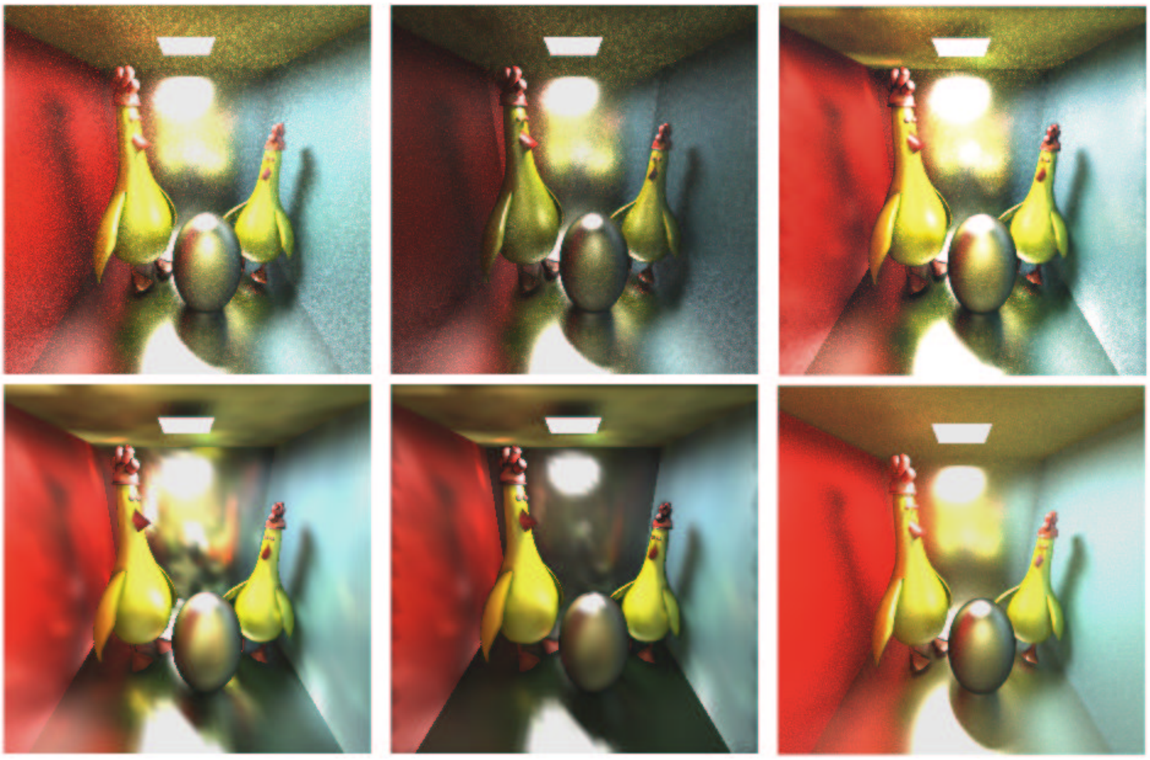

Figure 4: Cornell chickens scene images: a. non-weighted path tracing, b. weighted path tracing, c. combined image, d. non-weighted ray bundle, e. weighted ray bundle, f. reference image.

6 Implementation and Results

The presented algorithms have been implemented in C++ in OpenGL environment. The ray bundle iteration used Gouraud shading for the final image estimation. The transillumination buffer contained 1000×1000 pixels. The main task was the calculation of p^* according to equation (9) in path tracing and using equation (10) when performing bi-directional ray bundle iteration. The storage requirements for ray bundle iteration are increased since there is a need for another variable p^* for each patch.

The first test scene is the Cornell Chickens scene (25K patches), where the floor and the back wall has dominant specular characteristics. Their specular albedo is 0.85 and their shininess parameter is 50. The diffuse albedo, specular albedo and the shininess parameter of the chickens are 0.5, 0.5 and 10, respectively. The specular albedo of the egg is 0.8, its diffuse albedo is 0.1 and its shininess parameter is 10.

In theory the ray bundle iteration is able to sample all possible paths, but the probability of sampling a direction near the ideal reflection is quite low. However, path tracing handles this specular part of the integration well, but it is poor in areas of diffuse reflection. In areas like this the ray bundle iteration is more pleasing to the human eye, since it lacks the high variance of pixel intensities.

The non-weighted and weighted images can be seen in figure 4a, 4b, 4d, 4e. The combination of them is a simply arithmetic image addition of the weighted images shown in figure 4c. The reference image was rendered with bi-directional path tracing (figure 4f). Note that the specular regions (floor, back wall) are reproduced much better, because important path tracing paths get larger weights for this feature. On the other hand, the noise of path tracing (diffuse, indirect lit areas) is also reduced due to weighting. Each wall is tesselated into 1800 patches, that is in spite of Gouraud shading, visible in ray bundle images. However this discretisation error is also improved.

The second example is an architectural scene modelled in ArchiCAD. The images in figure 5 are snapshots of an architectural walk-through.

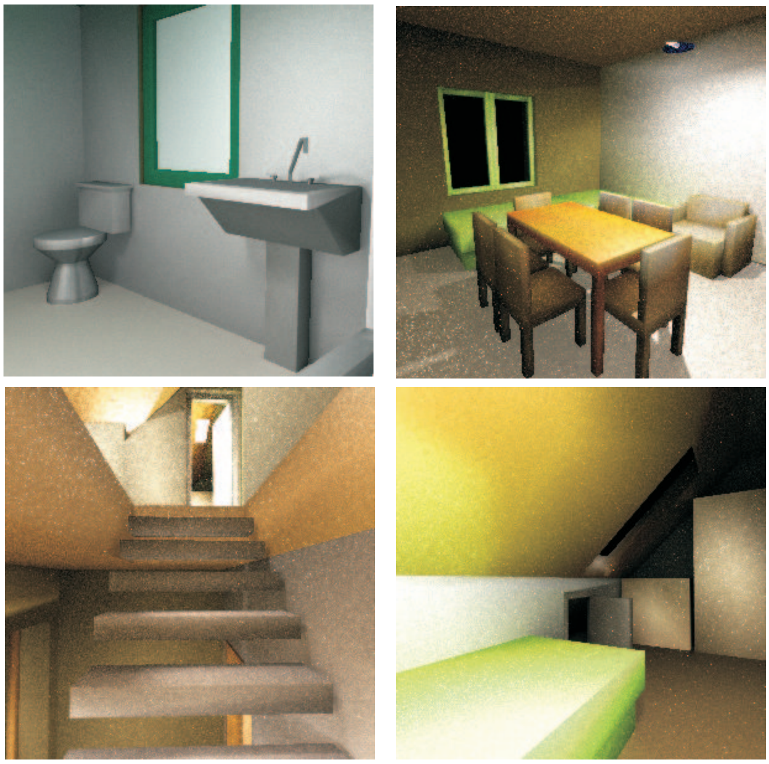

Figure 5: Images of an architectural scene.

The scene contains 80K of patches. The rendering times were 2 minutes for ray bundle iteration and 10 minutes for path tracing on a PentiumIII 800Mhz computer.

7 Conclusion and Future Work

This paper proposed a combination of path tracing and global ray bundle iteration in a way that the result is unbiased. It uses a provably efficient strategy determining the weights of the different passes.

The resulted algorithm exploits the advantages of both underlying algorithms, namely the fast image generation of ray bundle iteration, and the precise specular artifact calculation of path tracing.

While this combination can add the specular effects to the ray bundle iteration, the path tracer is poor to render caustics. Another algorithm such as light tracing or bi-directional path tracing may need to add caustics as well.

8 Acknowledgement

This work has been supported by National Scientific Research Fund (OTKA ref. No.: T029135), the Eötvös Foundation and the IKTA-00101/2000 project. The architectural scenes have been modelled by ArchiCAD that was donated by Graphisoft.

REFERENCES

[Arvo90] Arvo, J. and Kirk, D.: *Particle Transport and Image Synthesis* Computer Graphics, (SIGGRAPH '90 Proceedings), pp. 63-66, 1990.

[Bouatouch92] Bouatouch, K. and Tellier, P.: A two-pass physics-based global lighting model. In Proceedings of Graphics Interface'92, pp. 319-328, Toronto, Ontario 1992.

[Chen91] S. E. Chen, H. E. Rushmeier, G. Miller, D. Turner: A Progressive Multipass method for Global Illumination. Computer Graphics (SIGGRAPH '91 Proceedings), 25(4), pp. 165-174, 1991.

[Cohen87] J. R. Wallace M. F. Cohen, D. P. Greenberg: A two-pass solution to the rendering equation: A synthesis of ray tracing and radiosity methods. Computer Graphics, 21(4), pp. 311-320, 1987.

[Cohen93] M. F. Cohen and J. R. Wallace: Radiosity and Realistic Image Synthesis. San Diego, CA. Academic Press Professional, 1993.

[Granier00] Xavier Granier, George Drettakis and Bruce Walter: Fast Global Illumination Including Specular Effects. Eurographics Rendering Workshop'00, 2000.

[Jensen96] H. W. Jensen: Global Illumination Using Photon Maps. in Eurographics Rendering Workshop 1996 (X. Pueyo and P. Schroder, eds), (New Your City, NY), pp. 21-30, Eurographics, Springer Wien, ISBN 3-211-82883-4. 1996.

[Kajiya86] J. T. Kajiya: The Rendering Equation. Computer Graphics (SIGGRAPH' 86 Proceedings), pages 143-150, 1986.

[Lafortune93] Eric P. Lafortune, Yves D. Willems: Bi-directional Path-Tracing. Proceedings of Compugraphics '93, pp. 145-153, Alvor, Portugal 1993.

[Neumann95] L. Neumann: Monte Carlo Radiosity. Computing, 55:23-42, 1995.

[Owen99] A. Owen and Y. Zhou: Safe and effective importance sampling. Stanford University, Goldman-Sachs, 1999.

[Pauly00] Pauly, M. and Kollig, T. And Keller, A.: Metropolis Light Transport for Participating Media. Rendering Techniques 2000 (Proc. 11th Eurographics Workshop on Rendering), 2000.

[Shirley90] P. Shirley: A Ray Tracing Method for Illumination Calculation in Diffuse-Specular scenes. Graphics Interface'90, pp. 205-212, 1990.

[Sillion89] F. Sillion and C. Puech: A general two-pass method integrating specular and diffuse reflection. Computer Graphics, 23(3), pp. 335-344, 1989.

[Spanier69] J. Spanier and E. Gelbard: Monte Carlo Principles and Neutron Transport Problems. Addison-Wesley, 1969.

[Suykens99] Frank Suykens, Yves D. Willems: Weighted Multipass Methods for Global Illumination. Computer Graphics Forum (Eurographics'99), 1999.

[Szirmay98] László Szirmay-Kalos, Werner Purgathofer: Global Ray-bundle Tracing with Hardware Acceleration. Rendering Workshop '98, Vienna, 1998.

[Szirmay99] László Szirmay-Kalos: Stochastic Iteration for non-diffuse Global Ilumination. Computer Graphics Forum (Eurographics'99) 18(3):233-244, 1999.

[Veach95] Eric Veach, L. J. Guibas: Optimally Combining Sampling Techniques for Monter Carlo Rendering, Computer Graphics (SIGGRAPH '95 Proceedings), pp. 419-428, 1995.

[Veach97] Eric Veach: Robust Monte Carlo Methods for Light Transport Simulation. PhD thesis, Stanford University, 1997.