METROPOLIS ITERATION FOR GLOBAL ILLUMINATION[1]

László Szirmay-Kalos, Balázs Benedek, Mateu Sbert

Department of Control Engineering and Information Technology, Budapest University of Technology
Budapest, Magyar Tudósok Krt. 2, H-1117, HUNGARY
Email: szirmay@iit.bme.hu

ABSTRACT

This paper presents a stochastic iteration algorithm solving the global illumination problem, where the random sampling is governed by classical importance sampling and also by the Metropolis method. Point pairs where radiance transfer takes place are obtained with random ray shooting. Ray shooting can mimic the source radiance and the geometric factor, but not the receiving capability of the target (i.e. the BRDF and the area), which results in not optimal importance sampling. This deficiency is attacked by the Metropolis method. The pseudo random numbers controlling ray shooting are generated not independently, but by the perturbation of the previously used pseudo random numbers. These perturbations are accepted or rejected according to the change of the contribution of the transfers. The algorithm is mesh based, requires only a few variables per patch, and can render moderately complex glossy scenes in a few seconds.

Keywords: Global illumination, finite-element techniques, Monte-Carlo methods.

1 INTRODUCTION

Global illumination algorithms should compute the average of the radiance values on the area visible through a pixel or leaving surface patches in a given direction:

$$L_i(\omega) = \frac{1}{A_i} \cdot \int\limits_{A_i} L(\vec{x}, \omega) d\vec{x},$$

where A_i is the area of the surface on which the average is computed. The surface of interest A_i can be a patch in mesh-based finite-element algorithms or the surface visible in a pixel in continuous methods. The algorithm presented in this paper belongs to the mesh-based approaches, thus we assume that the surfaces are tessellated to patches $A_1, \ldots, A_N$.

According to the rendering equation [Kaj86], the directional radiance of a point is the sum of the emission and a reflected component. The average of the reflected radiance on patch A_i is:

$$L_i^r(\omega) = \int\limits_{A_i} \int\limits_{S} L(\vec{y}, \omega_{\vec{y}\to\vec{x}}) \cdot G(\vec{x}, \vec{y}) \cdot \frac{f_r(\omega_{\vec{y}\to\vec{x}}, \vec{x}, \omega)}{A_i} \, d\vec{y} d\vec{x}.$$

In this equation f_r is the BRDF and G is the geometric factor:

$$G(\vec{x}, \vec{y}) = v(\vec{x}, \vec{y}) \cdot \frac{\cos\theta_{\vec{x}} \cdot \cos\theta_{\vec{y}}}{|\vec{x} - \vec{y}|^2}$$

where v is the visibility function that is one if the two points can see each other, and $\theta_{\vec{x}}, \theta_{\vec{y}}$ are the angles between the surface normals and the direction $\omega_{\vec{y}\to\vec{x}}$ connecting $\vec{x}$ and $\vec{y}$.

We can observe three main actors in the integrand of the reflected radiance: the *source* at $\vec{y}$ having intensity of radiation $L(\vec{y} \to \vec{x}) = L(\vec{y}, \omega_{\vec{y}\to\vec{x}})$; the *receiver* at $\vec{x}$ having intensity of reception $R_\omega(\vec{y} \to \vec{x}) = f_r(\omega_{\vec{y}\to\vec{x}}, \vec{x}, \omega)/A_i$; and finally the *communication* between the source and the receiver as represented by the geometry factor $G(\vec{x}, \vec{y})$. When developing a good sampling strategy, these actors and their sampling densities

[1] Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Journal of WSCG, Vol.12, No.1-3., ISSN 1213-6972
WSCG'2004, February 2-6, 2004, Plzen, Czech Republic.
Copyright UNION Agency - Science Press

should be taken into account simultaneously. Unfortunately, only partial solutions are available, which rely on only two actors from the possible three.

In practice the computation should be carried out on all patches A_i, thus N separate integrals need to be evaluated. These different integrals can be merged together if the domain of the integral is expanded to the total surface S while multiplying the integrand by the characteristic function $\xi_i(\vec{x})$ of surface area A_i ($\xi_i(\vec{x}) = 1$ iff $\vec{x}$ is in area A_i):

$$L_i^r(\omega) = \int\limits_S \int\limits_S L(\vec{y} \to \vec{x}) \cdot G(\vec{x}, \vec{y}) \cdot R_\omega(\vec{y} \to \vec{x}) \cdot \xi_i(\vec{x}) \, d\vec{y} d\vec{x}.$$

This integral is often evaluated by Monte-Carlo or by quasi-Monte Carlo quadrature. In these methods the original integration domain $S \times S$ is mapped onto a unit cube U where the uniformly distributed pseudo-random or low-discrepancy points can be found [Deá89]. The local expansion between the two spaces is $d\mathbf{u} = t(\vec{x}, \vec{y}) \cdot d\vec{x} d\vec{y}$. Here t is the determinant of the Jacobi matrix. This mapping assigns an $(\vec{x}, \vec{y})$ point pair to point $\mathbf{u}$ in the unit cube, thus the radiance, the geometric factor and the intensity of the reception will also be functions of $\mathbf{u}$. Using variable transformation in the integrand, we can obtain

$$L_i^r(\omega) = \int\limits_U \frac{L(\mathbf{u}) \cdot G(\mathbf{u}) \cdot R_\omega(\mathbf{u}) \cdot \xi_i(\vec{x}(\mathbf{u}))}{t(\mathbf{u})} \, d\mathbf{u}.$$

A classical Monte-Carlo quadrature would take M uniformly distributed random samples in the unit cube and approximate the integral as follows:

$$\frac{1}{M} \cdot \sum_{m=1}^{M} \frac{L(\mathbf{u}_m) \cdot G(\mathbf{u}_m) \cdot R_\omega(\mathbf{u}_m) \cdot \xi_i(\vec{x}(\mathbf{u}_m))}{t(\mathbf{u}_m)}.$$

Note that if $\mathbf{u}$ is sampled uniformly, the probability density of obtaining an $(\vec{x}, \vec{y})$ pair is $t(\vec{x}(\mathbf{u}), \vec{y}(\mathbf{u}))$ due to the local expansion of the mapping from the unit cube onto space $S \times S$. In order to reduce the variance, the variable transformation should result in a flat integrand, that is, the local expansion factor t should mimic the integrand. This approach is called *importance sampling*.

Unfortunately it is impossible to construct a transformation where t mimics all actors of the integrand, the source, the receiver and the communication. It is, however, possible to find a transformation for any two actors from the three. In *ray shooting* we can first sample the source point and the direction proportional to its cosine weighted radiance, then a ray is traced from the selected point at the selected direction to obtain the destination of the transfer. Ray shooting cannot mimic the BRDF and the area of the receiver patch. In *ray gathering* on the other hand, a receiver point and a direction are sampled proportional to its cosine weighted BRDF, and the source of the radiance transfer is found by casting a ray. Ray gathering is not able to follow the radiance of the source patch. Finally the source and radiance can be sampled simultaneously and the visibility is detected by a shadow ray. In this case the sampling does not take the geometric factor into account. Despite to their limitations, these methods, or their combination are worth using, since they can flatten the integrand. However, they are not able to completely eliminate the random fluctuations. We shall call the algorithm of converting points in the unit cube to point pairs for the radiance transfer as the *basic method*.

In order to reduce the variance of the not optimal importance sampling of the basic method, we shall apply Metropolis sampling. Metropolis algorithm obtains samples proportionally with a prescribed scalar importance function in the framework of an adaptive process. We shall set this prescribed importance function to take into account all actors of the transfer.

1.1 Metropolis sampling

Metropolis method constructs a Markovian process in a way that its stationary distribution $p(\mathbf{u})$ is proportional to a prescribed scalar importance function $\mathcal{I}(\mathbf{u})$. This scalar importance function can be the integrand, or the luminance of the integrand if the light is simultaneously transferred on several wavelengths. Thus the Metropolis method can provide optimal importance sampling asymptotically. Note that in our case a separate integral needs to be evaluated for every patch. We can use the same Markovian process for all integrals simultaneously, if the scalar importance function is defined as the common part of all integrands.

The sampling density is then $p(\mathbf{u}) = 1/b \cdot \mathcal{I}(\mathbf{u})$, where $b = \int \mathcal{I}(\mathbf{u}) d\mathbf{u}$ is the normalization constant that makes the density to integrate to one. Thus the quadrature rule for integral $\int F(\mathbf{u}) d\mathbf{u}$ is the following:

$$P = \int F(\mathbf{u}) d\mathbf{u} \approx \frac{1}{M} \sum_{m=1}^{M} \frac{F(\mathbf{u}_m)}{p(\mathbf{u}_m)} = \frac{b}{M} \sum_{m=1}^{M} \frac{F(\mathbf{u}_m)}{\mathcal{I}(\mathbf{u}_m)}.$$

In our case the actual sample $\mathbf{u}_i$, also called the actual state, is a point in the unit cube, which unambiguously determines a pair of surface points $(\vec{x}, \vec{y})$ between radiance transfer takes place.

The next state $\mathbf{u}_{i+1}$ of the sampling process is found by letting an almost arbitrary *tentative transition function* $T(\mathbf{u}_i \to \mathbf{u}_t)$ generate a *tentative sample* $\mathbf{u}_t$ which is either accepted as the real next state or rejected making the next state equal to the actual state. The decision uses the "*acceptance probability*" $a(\mathbf{u}_i \to \mathbf{u}_t)$ that expresses the increase of the scalar importance function (if this "acceptance probability" is greater than one, then the sample is accepted deterministically).

Veach[VG97] recognized that it is worth using also the rejected samples since they also provide useful information. Note that a tentative sample is accepted with probability a, while the original sample is kept with probability $1-a$. Replacing this random variable by its mean, both locations can be contributed but the contributions of the tentative sample and the old sample should be weighted with a and $1-a$, respectively.

The original Metropolis algorithm scales the integral quadrature by the normalization constant of the scalar importance function $b = \int \mathcal{I}(\mathbf{u}) d\mathbf{u}$, thus the integral of the scalar importance function needs to be approximated. Veach proposed a preprocessing step, when b is obtained by a classical Monte-Carlo scheme. This means that the error of computing b is inherited by the Metropolis algorithm, which will not be able to correct it, thus the algorithm will be biased. Summarizing, the pseudo-code of the Metropolis algorithm is as follows:

```
Approximate $b = \int \mathcal{I}(\mathbf{u})\, d\mathbf{u}$
for $i = 1$ to $M$ do
    Sample a tentative $\mathbf{u}_t$ using $T(\mathbf{u}_i \to \mathbf{u}_t)$
    $a(\mathbf{u}_i \to \mathbf{u}_t) = \min\left\{\frac{\mathcal{I}(\mathbf{u}_t)\cdot T(\mathbf{u}_t \to \mathbf{u}_i)}{\mathcal{I}(\mathbf{u}_i)\cdot T(\mathbf{u}_i \to \mathbf{u}_t)}, 1\right\}$
    $P \mathrel{+}= F(\mathbf{u}_i)/\mathcal{I}(\mathbf{u}_i) \cdot b/M \cdot (1-a)$
    $P \mathrel{+}= F(\mathbf{u}_t)/\mathcal{I}(\mathbf{u}_t) \cdot b/M \cdot a$
        // accept with probability $a(\mathbf{u}_i \to \mathbf{u}_t)$
    Generate random number $r$ in $[0,1]$.
    if $r < a(\mathbf{u}_i \to \mathbf{u}_t)$ then $\mathbf{u}_{i+1} = \mathbf{u}_t$
    else                          $\mathbf{u}_{i+1} = \mathbf{u}_i$
endfor
```

This scheme has been used to evaluate integrals in the path space, that is to find complete light paths proportionally to their carried luminance in [VG97]. The basic idea was extended to incorporate participating media in [PKK00]. In [KSKAC02] it was pointed out that perturbations are worth executing in the space where the uniformly distributed pseudo-random numbers are obtained. We shall also use this idea in this paper. An interesting attempt to incorporate the Metropolis sampling in the radiosity method has been published in [BS02]. These methods all belong to the category of random walks, when the integral corresponds to the infinite dimensional path space. However, the Metropolis algorithm is more powerful when the domain of the integral is not high dimensional. This is the particular case in iteration algorithms, when the radiance update in a single iteration requires the evaluation of four dimensional integrals.

In the next section we propose a new global illumination algorithm that applies Metropolis sampling to reduce the variance of a stochastic iteration scheme applying ray shooting to iterate the scene radiance. The new algorithm is the first method that applies Metropolis sampling independently of the random walk context. Similarly to [VG97] we also utilize the rejected samples weighting the tentative and actual samples by the acceptance and the rejection probabilities, respectively. However, unlike in other methods, we propose an adaptive computation of the scaling factor b instead of approximating it separately. This adaptive process guarantees asymptotically unbiased results.

We can also interpret our approach in the following way. The mapping from the unit cube to the space of point pairs, as defined by the basic method (ray shooting), makes t only approximately proportional to the integrand. Then Metropolis method is used to reduce that variation which is left by the improper importance sampling of the basic method. The integrals estimated by the Metropolis method are low dimensional thus we can expect fast convergence and the full power of the optimal sampling.

2 THE NEW ITERATIONAL ALGORITHM

So far we assumed that the radiance is available and we have to compute the average radiance reflection at different patches of the scene. In practical situations this case holds only for final gathering but not during the global illumination solution. To cope with this problem, we can follow an iteration approach. Originally the radiance values are initialized to the emissions, then the emission is increased by the reflection of the current radiance estimate. For the computation of the reflected radiance we use the Metropolis method. This step is repeated to compute further iterations which are responsible for refining the Monte-Carlo estimates of each light bounce and also for the introduction of higher order light bounces. Note, however, that the stochastic iteration of the radiance will not converge, the random radiance estimates fluctuate around the real solution. From these random estimates the final result can be obtained by averaging[SK00]. If the diffuse radiosity problem is solved, averaging can be done in object space. However, in the non-diffuse case the representation of the complete radiance function would require large storage. Thus an image estimate is computed after each iteration step, and averaging takes place in the image space. In our method we shall use both object space and image space averaging. The *main part* of the radiance (i.e. the directional average of the diffuse and specular terms), which is independent of the direction, is stored in the object space, since this is needed to compute the scaling factor b. On the other hand, the fluctuation of the radiance caused by the specular materials is averaged in the image space.

2.1 The basic method

To realize the mapping between the unit cube and the pairs of surface points, we use ray shooting. The first two elements of vector $\mathbf{u}$ is used to select a point proportionally to its radiance. A unit interval is decomposed to intervals proportional to the patch powers and with the first element we find the source patch proportionally to its power. Subtracting the start of the elementary interval from the first random number and multiplying with the length of the elementary interval, the new number will again be distributed in the unit interval. This number together with the second element of vector $\mathbf{u}$ are used to sample a uniformly distributed random point $\vec{y}$ on the selected patch. Finally two additional random variables are needed to sample a direction proportionally to the cosine weighted BRDF. The first one determines whether the diffuse or the specular BRDF is sampled. Scaling the first element after the decision, and taking the second, the direction is obtained by transforming the two values with the selected BRDF function. Casting a ray from $\vec{y}$ into the sampled direction, hit point $\vec{x}$ is obtained. Radiance transfer is computed between points $\vec{x}$ and $\vec{y}$. Note that this ray shooting algorithm defines a correspondence between the points of a four-dimensional unit cube and the pair of surface points.

The basic method mimics the radiance of the source and the geometric factor, but not the albedo and the area of the target. Thus, if we used the basic method alone, the algorithm would be poor at small, high albedo patches, and would devote needlessly many rays to hit larger patches of smaller reflectance. This problem is solved by Metropolis sampling.

2.2 Definition of the scalar importance function

In order to find an optimal sampling, the importance function $\mathcal{I}$ should be proportional to the common part of the integrands (the only factor which is not common is the characteristic function of the patches $\xi_i(\vec{x}(\mathbf{u}))$):

$$\frac{L(\vec{y}(\mathbf{u}) \to \vec{x}(\mathbf{u})) \cdot G(\vec{x}(\mathbf{u}), \vec{y}(\mathbf{u})) \cdot R_\omega(\vec{y}(\mathbf{u}) \to \vec{x}(\mathbf{u}))}{t(\vec{x}(\mathbf{u}), \vec{y}(\mathbf{u}))}$$

Note that this function also depends on viewing direction ω, which is the direction of the eye in case of final gathering, or the direction of the next transfer in an iteration like global illumination algorithm. Since the direction of the next transfer is not known yet, we can use the average cosine weighted intensity of reception, taking into account all directions:

$$R(\vec{y} \to \vec{x}) \approx \frac{1}{\pi} \cdot \int_\Omega R_\omega(\vec{y} \to \vec{x}) \cos\theta \, d\omega = \frac{\rho_i(\omega_{\vec{y}\to\vec{x}})}{\pi A_i}$$

where ρ_i is the albedo of patch i. This approximation can also be given an alternative interpretation. In the next iteration step the direction is obtained by BRDF sampling, thus the probability density will follow the cosine weighted BRDF. The weight of the transfer, that is the ratio of the cosine weighted BRDF and the probability density will be the albedo. It means that the replacement of the BRDF by the albedo does not introduce additional variance.

The other problem of the definition of the importance function is that the radiance and the albedo are not scalars but rather vectors having elements for every wavelength on which the transfer is computed. The scalar importance function should express where the elements of these vectors are large, therefore we use the luminance function $\mathcal{L}$, which computes the weighted sum of the elements. Thus the scalar importance function is

$$\mathcal{I}(\mathbf{u}) =$$

$$\mathcal{L}\left(\frac{L(\vec{y}(\mathbf{u}) \to \vec{x}(\mathbf{u})) \cdot G(\vec{x}(\mathbf{u}), \vec{y}(\mathbf{u})) \cdot R(\vec{y}(\mathbf{u}) \to \vec{x}(\mathbf{u}))}{t(\vec{x}(\mathbf{u}), \vec{y}(\mathbf{u}))}\right)$$

where $\mathcal{L}$ computes the luminance of a spectral distribution.

2.3 The primary Monte-Carlo estimate

The primary estimate of the Monte-Carlo quadrature is the integrand divided by the probability density. Taking into account that in the Metropolis algorithm the probability density of the samples is $\mathcal{I}(\mathbf{u})/b$ asymptotically, the primary estimate is

$$\frac{L(\vec{y}(\mathbf{u}) \to \vec{x}(\mathbf{u})) \cdot R_\omega(\vec{y}(\mathbf{u}) \to \vec{x}(\mathbf{u})) \cdot \xi_i(\vec{x}(\mathbf{u}))}{\mathcal{L}\left(L(\vec{y}(\mathbf{u}) \to \vec{x}(\mathbf{u})) \cdot R(\vec{y}(\mathbf{u}) \to \vec{x}(\mathbf{u}))\right)} \cdot b.$$

Note that this estimate has small variance since the geometric factor and the density of the ray shooting are completely compensated, and the radiance of the shooter and the receiving capability of the receiver are well mimicked. In fact, the only factor which can introduce larger variance is the characteristic function ξ_i of patch i. Suppose, for example, that the surfaces are diffuse and the radiance is computed on a single wavelength. In this case all rays would result in a reflected contribution equal to b. The Metropolis algorithm is responsible for distributing the rays in a way, that the higher radiance patches are hit more frequently.

In moderately complex scenes there are a few tens of thousands of patches, thus we can expect acceptable results in about a million stochastic iterations, i.e. having computed about a million rays (we recall that in stochastic iteration algorithms a few rays per patch usually give satisfying results even with poorer importance sampling [Bek99]). This number of rays are enough even if the image has a higher resolution or the lighting distribution is very heterogeneous (e.g. only a small part of the scene is illuminated by strong light sources). Note that this ray number would only be enough for the identification of the points visible in different pixels if a stochastic ray-tracing approach were applied, thus our method is expected to be much faster than random walk methods.

Scaling factor b is the integral of the importance function over the whole domain:

$$b = \int_U \mathcal{I}(\mathbf{u})\ d\mathbf{u} =$$

$$\mathcal{L}\left(\int_S \int_S L(\vec{y} \to \vec{x}) \cdot G(\vec{x}, \vec{y}) \cdot R(\vec{y} \to \vec{x})\ d\vec{x} d\vec{y}\right).$$

Since $R(\vec{y} \to \vec{x}) = \rho_i/\pi A_i$ if $\vec{x}$ is on patch i, the integral of b is the luminance of the total, average reflected radiance assuming that all surfaces are diffuse and having the albedo of the original surfaces. In order to compute this term, we separate the constant main part of the reflected radiance. The estimates of this main part are averaged in each iteration step, thus it will converge to a stable value. Let us store the directional average of the reflected radiance in variable $L^{d,(n)}$ in each patch i computed as

$$L_i^{d,(n)} = \frac{1}{n} \cdot \sum_{m=1}^{n} I_i(m) \cdot \frac{\rho_i(\omega_m)}{\pi},$$

where $\rho(\omega)$ is the albedo of the material and $I(m)$ is the irradiance of iteration step m (i.e. irradiance I is the incoming radiance estimate multiplied by the cosine of the incoming angle). Then scaling factor b is the luminance of this main part taking into account all $\vec{x}$ points:

$$b = \mathcal{L}\left(\sum_{i=1}^{N} L_i^{d,(n)}\right).$$

We have to emphasize that L^d is not equal to the diffuse component, but also includes the average of the specular reflection.

2.4 Representation of the radiance function

Because of the computation of scaling factor b, we have to determine the main part of the radiance function in object space. This main part can also be used to reduce the fluctuation of the patch radiances if we express the reflected radiance as the sum of this main part and a difference part computed from the difference BRDF $\Delta f_r = f_r - \rho/\pi$:

$$L_i^{r,(n)}(\omega) = L_i^{d,(n)} + I_i(n) \cdot \Delta f_i(\omega_n, \omega).$$

As mentioned, only the main part converges, the reflected radiance does not, but fluctuates due to the random $I_i(n) \cdot \Delta f_i(\omega_n, \omega)$ term. The final result is obtained by computing an image estimate in each iteration step, then averaging these image estimates.

If a patch is hit by a ray in iteration n, its irradiance $I(n)$ and the direction of the ray ω_n are stored on the patch. For those patches that are not hit by the ray, the irradiance of this iteration step is zero. Examining this sequence, we can note that it has a high fluctuation, it is mostly zero but when the patch is lucky enough to be hit by a ray, then it gets a larger contribution.

The variance of the whole method can be reduced if the fluctuation of this sequence is decreased. The general idea is to replace sequence $I(n)$ by another sequence which is smoother but still results in the correct reflected radiance when averaging takes place [KBSK03]. To obtain the new sequence, zero samples are ignored, large samples of the original sequence will be scaled down and small samples will be scaled up. In order not to distort the average computed from the sequence, a scaled down larger value will appear more times in the new sequence. In the optimal case the scaling would make the luminance of the reflected radiances of all elements in the sequence equal. The average reflected luminance of the sequence is:

$$C = \frac{1}{n} \cdot \sum_{m=1}^{n} \mathcal{L}(I(m)\Delta a).$$

In this formula Δa is the albedo of the absolute value of the difference BRDF Δf_r. Thus an appropriate scaling of $I(n)$, which makes the reflected luminance constant, is

$$\frac{I(n)}{n \cdot \mathcal{L}(I(n)\Delta a)} \cdot C.$$

The average will be correct if we can guarantee that $I(n)$ is expected to appear $\mathcal{L}(I(n)\Delta a)/C$ times. A sampling scheme that can produce samples proportionally with $\mathcal{L}(I(m)\Delta a)$ is based on random acceptance and rejection of the Metropolis method. At each iteration step the new irradiance $I(n)$ is compared with the stored irradiance $I(m)$. If $\mathcal{L}(I(n)\Delta a)$ is greater or equal than $\mathcal{L}(I(m)\Delta a)$, then the new irradiance will replace $I(m)$ in the random representation of the radiance. However, when $\mathcal{L}(I(n)\Delta a)$ is smaller than $\mathcal{L}(I(m)\Delta a)$, the new irradiance is accepted with probability $\mathcal{L}(I(n)\Delta a)/\mathcal{L}(I(m)\Delta a)$.

Note that this approach has the advantage that the representation of the patch radiance requires just a few variables: the main part L^d, the irradiance I together with the direction ω_n, the average reflected luminance C, and the eye radiance. Thus the storage requirement of this method is comparable to that of the diffuse radiosity algorithms, although our method can also solve the glossy global illumination problem.

2.5 Perturbation strategy

The Metropolis algorithm is driven by the applied perturbation strategy. We use uniform perturbations in a small neighborhood of the actual sample. The edge size of this small neighborhood is 0.1. In order to guarantee the ergodicity of the process, i.e. to make sure that all non-zero importance point will be generated by positive probability sooner or later, we randomly introduce large perturbations among the small ones. These large perturbations may obtain any point of the unit cube with uniform probability density and independently of the actual sample [KSKAC02]. The probability of these large perturbations is currently 0.5.

Summarizing, the pseudo-code of the proposed iteration algorithm is:

```
for $i = 1$ to $M$ do
    large_step = a random binary value
    if (large_step) then Generate a random $\mathbf{u}_t$
    else $\mathbf{u}_t$ = small perturbation of $\mathbf{u}_i$
    $a(\mathbf{u}_i \to \mathbf{u}_t) = \min\left\{\frac{\mathcal{I}(\mathbf{u}_t)}{\mathcal{I}(\mathbf{u}_i)}, 1\right\}$
    Find $\vec{x}_i, \vec{y}_i$ from $\mathbf{u}_i$ using ray shooting
    Reflect $\frac{L(\mathbf{u}_i)\cdot R_\omega(\mathbf{u}_i)}{\mathcal{L}(L(\mathbf{u}_i)\cdot R(\mathbf{u}_i))} \cdot b \cdot (1-a)$
            at the patch of $\vec{x}_i$
    Find $\vec{x}_t, \vec{y}_t$ from $\mathbf{u}_t$ using ray shooting
    Reflect $\frac{L(\mathbf{u}_t)\cdot R_\omega(\mathbf{u}_t)}{\mathcal{L}(L(\mathbf{u}_t)\cdot R(\mathbf{u}_t))} \cdot b \cdot a$
            at the patch of $\vec{x}_t$
    Average the patch radiances
          // accept with probability $a(\mathbf{u}_i \to \mathbf{u}_t)$
    Generate random number $r$ in $[0, 1]$.
    if $r < a(\mathbf{u}_i \to \mathbf{u}_t)$ then $\mathbf{u}_{i+1} = \mathbf{u}_t$
    else                                    $\mathbf{u}_{i+1} = \mathbf{u}_i$
endfor
```

3 RESULTS

The proposed method has been tested with a moderately complex scene of a room with columns and a Beethoven, tessellated to 22767 patches and illuminated by an area light source in the non-visible corner of the room (figure 3). First we assumed that all surfaces are diffuse and rendered a close-up of Beethoven. The image resolution was 800×900. We used only one million rays, which required 20 second on a 1.4 GHz computer both for the stochastic ray shooting and for the Metropolis algorithms (the overhead of Metropolis sampling is negligible). The Beethoven is the most difficult part of the scene since it consists of small, high albedo patches, which are far from the light source. This means that ray-shooting would assign very few primary and secondary rays these patches, which is responsible for the high variance of figure 1. This problem can be solved by the Metropolis sampler, which takes into account not only the radiance of the sources but also the size and the albedo of the targets. Thus Metropolis will allocate more rays for the difficult part of the scene and reduce the ray numbers at large low albedo patches, where few rays can provide accurate results. The reduction of the variance is shown in figure 2.

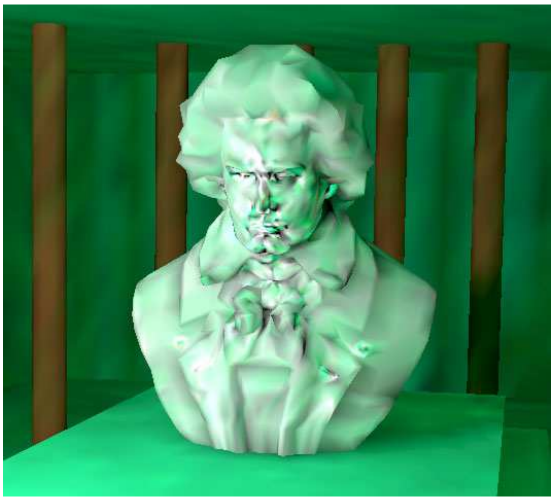

Figure 1: Stochastic ray-shooting

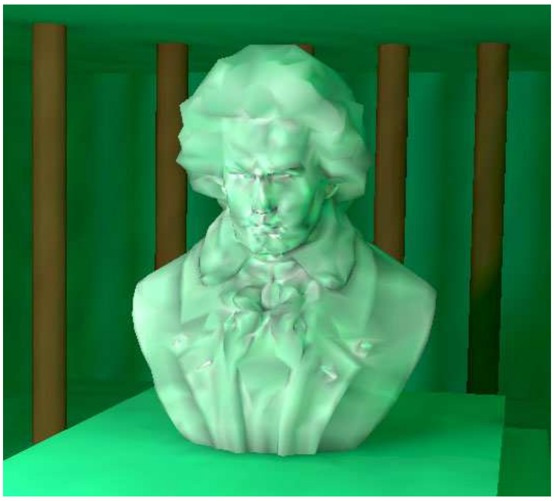

Figure 2: Stochastic ray-shooting with Metropolis sampling

When we turned the surfaces specular we had to realize that the finite-element decomposition becomes visible at specular surfaces due to Gouraud shading. Thus we decided to compute the direct illumination separately with Phong shading and use the proposed algorithm only for calculating the indirect illumination. The separate computation of the direct illumination increased the rendering time by 4 seconds. The result rendered on 800 × 600 resolution is shown in figure 3.

4 CONCLUSIONS

In this paper we proposed a stochastic iteration algorithm that uses both classical importance sampling (i.e. ray shooting) and Metropolis method to reduce the random noise introduced in each iteration step. The ray shooting can mimic the source radiance and the geometric factor, but not the receiving capability of the target, which results in not optimal importance sampling. This deficiency is attacked by the Metropolis method, which incorporates the albedo and the area of the target into importance sampling. Thus the new method is particularly efficient in scenes where the patch size and the reflectance vary significantly. This is the case in architectural scenes, for example, when the walls and the external scenery are usually poorly tessellated but the internal curved objects are decomposed to a large number of small polygons.

There is room for several improvements. We can, for example, use a better basic method. Instead of ray shooting, better results can be obtained with a method which combines ray shooting, ray gathering and connecting the shooter and the receiver. The combination can be made quasi-optimal by multiple importance sampling. The method can also be made bi-directional. When a point pair is established, the radiance or importance can be transferred into both directions. The proposed algorithm, as all shooting methods, generates rays that hit surfaces proportionally to the carried radiance. However, this is not optimal in the last step to compute the eye contribution, which would require samples according to BRDF sampling. To attack this problem, weighted importance sampling seems very promising [SKAB03].

5 ACKNOWLEDGEMENT

This work has been supported by the National Scientific Research Fund (OTKA ref. No.: T042735), IKTA ref. No.: 00159/2002, the Bolyai Scholarship and the Slovene-Hungarian Fund, by Intel, and by TIC 2001-2416-C03-01 from the Spanish Government.

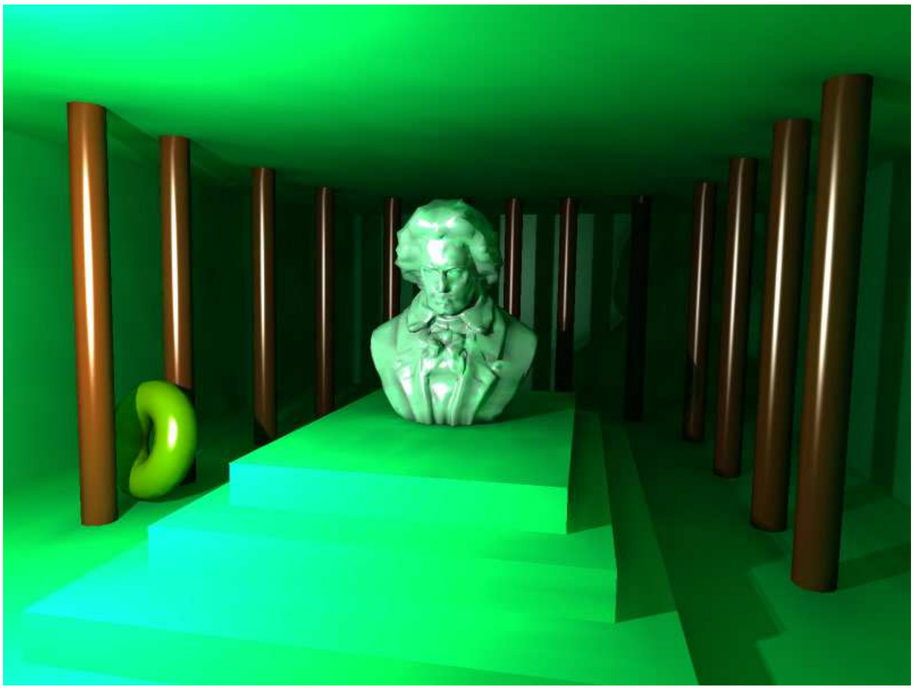

Figure 3: A specular scene rendered by the proposed Metropolis sampling (24 secs rendering time)

REFERENCES

[Bek99] P. Bekaert. *Hierarchical and stochastic algorithms for radiosity*. PhD thesis, University of Leuven, 1999.

[BS02] A. Brusi and M. Sbert. An application of metropolis method to radiosity. Technical Report IIiA 02-04-RR, University of Girona, 2002.

[DDM02] C. Damez, K. Dmitriev, and K. Myszkowski. Global illumination for interactive applications and high-quality animations. Eurographics, September 2002. STAR.

[Deá89] I. Deák. *Random Number Generators and Simulation*. Akadémia Kiadó, Budapest, 1989.

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

[KBSK03] Cs. Kelemen, B. Benedek, and L. Szirmay-Kalos. Bi-directional rays in global illumination. In *WSCG 2003 Conference, Posters, Plzen*, 2003.

[KSKAC02] Cs. Kelemen, L. Szirmay-Kalos, G. Antal, and F. Csonka. A simple and robust mutation strategy for the Metropolis light transport algorithm. In *Eurographics '02*, pages 531–540, 2002.

[PKK00] M. Pauly, T. Kollig, and A. Keller. Metropolis light transport for participating media. In *Rendering Techniques*, pages 11–22, 2000.

[SK00] L. Szirmay-Kalos. *Photorealistic Image Synthesis with Ray-Bundles*. Hungarian Academy of Sciences, D.Sc. Dissertation, 2000. www.iit.bme.hu/˜szirmay/ Thesis-SzKL.htm.

[SKAB03] L. Szirmay-Kalos, Gy. Antal, and B. Benedek. Weighted importance sampling in shooting algorithms. In *Spring Conference on Computer Graphics 2003*, pages 192–198, 2003.

[VG97] E. Veach and L. Guibas. Metropolis light transport. *Computer Graphics (SIGGRAPH '97 Proceedings)*, pages 65–76, 1997.

[ZBP99] J. Zaninetti, P. Boy, and B. Peroche. An adaptive method for area light sources and daylight in ray tracing. *Computer Graphics Forum*, 18(3):139–150, 1999.