Hemicube Shooting for Non-Diffuse Global Illumination

György Antal,*
László Szirmay-Kalos,†
Ferenc Csonka,‡
Technical University of Budapest§

Abstract

The hemicube is a classical tool to transfer the light power in diffuse radiosity algorithms. The main advantage of the hemicube based light transfer is that the visible patches can easily be identified by the graphics hardware. This paper extends the hemicube approach to solve the non-diffuse global illumination problem. In order to get rid of the quadratic complexity of classical radiosity algorithms and to allow specular surfaces without storing directional finite-elements, the original iteration is replaced by stochastic iteration. Unlike classical iteration where all patches should be selected to gather the radiosity or to shoot their unshot radiosity, stochastic iteration can exploit that a randomly selected patch may represent its neighbours as well, thus accurate results can be obtained even if just a fraction of patches are selected at all. Since stochastic iteration requires just a random approximation of the patch radiance, it can use just a single variable per patch even if the general, non-diffuse problem is attacked. Random selection, however, may introduce noise that is particularly significant where the source and receiver patches are close. We also propose a solution strategy to eliminate these artifacts. The paper also discusses further improvements by applying constant radiance step and by the randomization of the hemicube.

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

1 Introduction

Global illumination aglorithms aim at the physically correct simulation of the light propagation and solve some form of the rendering equation [9]

$$L = L^e + \mathcal{T}_{f_r} L,$$

which expresses the radiance $L(\vec{x}, \omega)$ of point $\vec{x}$ at direction ω as a sum of the emission L^e and the reflection of all point radiances that are visible from here. The reflection of the radiance is expressed by an integral operator

$$\mathcal{T}_{f_r} L(\vec{x}, \omega) = \int_\Omega L(h(\vec{x}, -\omega'), \omega') \cdot f_r(\omega', \vec{x}, \omega) \cdot \cos\theta' \, d\omega',$$

which is also called as the *light transport operator*. In this equation h is the visibility function finding that point which is visible from $\vec{x}$ at direction $-\omega'$, f_r is the BRDF and θ' is the angle between the surface normal and direction $-\omega'$.

The *radiosity* method [6] was the first global illumination technique. In order to solve the rendering equation, radiosity algorithms use iteration. Iteration techniques are based on the fact that the solution of the rendering equation is the fixed point of the following iteration scheme:

$$L(m) = L^e + \mathcal{T}_{f_r} L(m-1).$$

If this scheme is convergent, then the pixel colors can be obtained as a limiting value:

$$C = \lim_{m \to \infty} \mathcal{M} L(m)$$

where $\mathcal{M}$ is the measuring operator of the camera.

To store the radiance estimates, finite-element approaches should be used which represent the radiance function in a finite function series form:

$$L(\vec{x}, \omega) = \sum L_j \cdot b_j(\vec{x}, \omega)$$

where functions $b_j(\vec{x}, \omega)$ are pre-defined basis functions and parameters L_j are scalars. In the diffuse case basis functions $b_j(\vec{x}, \omega)$ are independent of the direction ω. In the general, non-diffuse case, these basis functions are usually decomposed to a product of positional ($s_k(\vec{x})$) and directional basis functions ($d_i(\omega)$). The positional basis functions may be either constant or linear on a patch, while the directional basis functions can also be piece-wise constant [7], spherical harmonics [16] or Haar functions [17]. Due to the fact that the radiance has 4 variates and changes quickly, an accurate finite-element representation requires very many basis functions, which makes these algorithms both storage and time consuming. Although, hierarchical or multiresolution methods [3] and clustering [2] can help, the memory requirements are still prohibitive for complex scenes.

*agy@inf.elte.hu
†szirmay@iit.bme.hu
‡sumi@inf.elte.hu
§Budapest, Magyar Tudósok krt. 2, H-1117, HUNGARY

Early radiosity implementations used Gauss-Seidel or Jacobi-iteration to solve the set of radiosity equations. Both iteration methods require the entire form factor matrix, which introduces $\mathcal{O}(N^2)$ memory complexity, where N is the number of patches in the diffuse scene. If the scene is non-diffuse, then N is the number of patches multiplied by the number of directional basis functions, which makes the finite-element approach even more prohibitive to use. Concerning the time complexity, iteration converges with the speed of a geometric series, i.e. the error from the limiting value is in the order of $\mathcal{O}(a^m)$ where a is the contraction of integral operator $\mathcal{T}_{f_r}$ [20]. The contraction is proportional to the average albedo of the surfaces and depends on how open the scene is. Since the contraction is independent on the number of patches, the number of required iterations is constant. However, in a single iteration step an $N \times N$ matrix is multiplied with an N element vector, which is a quadratic operation. Thus the classical iteration is quadratic both in terms of computational time and storage. Therefore it cannot be used for complex scenes.

A more advanced method, called the progressive refinement, is based on the Southwell iteration where it is not necessary to store the entire matrix of form factors, and only one column of the matrix is computed on the fly in each iteration step. Each patch is associated with three values, such as an emission, a radiosity and an unshot radiosity. The algorithm iteratively selects the patch with the highest unshot energy, and shoots the unshot power towards other patches in the scene. This trick reduces the storage complexity and the time complexity of a single operation to $\mathcal{O}(N)$. However, the required number of Southwell iterations will be also linear, thus the whole process is still quadratic [18]. This can also be intuitively understood since, roughly speaking, all patches must be selected to allow them to get rid of their unshot power.

In practical cases the neighboring patches are usually very similar, thus it would be enough to select one of them and transfer the total power of the neighborhood at once from here. This idea is exploited by hierarchical radiosity and also by Monte-Carlo radiosity [10, 14]. Monte-Carlo radiosity selects a patch just with certain probability, but the transferred radiance is divided with this probability. Note that this operation simulates the strategy that instead of all patches, just a few representative ones are picked, but from these representative patches we also transfer the radiance of those patches that have not been selected.

Realizing this, for highly complex scenes the progressive radiosity was randomized and called Monte-Carlo radiosity. The calculation of the necessary form factors can be done by using a hemicube [11] or can be performed by local or global lines of the ray tracing approach [15, 13]. The unification of the Monte Carlo radiosity and hierarchical techniques[1] has also been proposed recently.

When the non-diffuse case is considered, the other main problem is the enormous number of required finite-elements. Randomization can also help solving this problem as has been proposed by the concept of *stochastic iteration* [19]. Suppose that we have a random linear operator $\mathcal{T}^*$ so that

$$E[\mathcal{T}^* L] = \mathcal{T}_{f_r} L \tag{1}$$

for any integrable function L. During stochastic iteration a random sequence of operators $\mathcal{T}_1^*, \mathcal{T}_2^*, \ldots \mathcal{T}_i^* \ldots$ is generated, which are instantiations of $\mathcal{T}^*$, and this sequence is used in the iteration:

$$L(m) = L^e + \mathcal{T}_m^* L_{m-1}. \tag{2}$$

Suppose that at a given point of the iteration, an image estimate is computed from the actual radiance, that is, the measured value in each pixel is

$$C_m = \mathcal{M} L(m). \tag{3}$$

This measured value will also be a random variable which does not converge but fluctuates around the real solution. This problem, however, can be solved, if the image estimates are computed after each iteration step and the final result is obtained as an average of these estimates[19].

Averaging the first M steps, we obtain:

$$C(m) = \frac{1}{m} \cdot \sum_{i=1}^{m} \mathcal{M} L(i) =$$

$$\frac{1}{m} \cdot \mathcal{M} L(m) + \left(1 - \frac{1}{m}\right) \cdot C(m-1).$$

If $\mathcal{T}^*$ is properly constructed, then it does not need the radiance function everywhere it its domain, which helps reducing the astronomical storage requirements of directional dependent finite elements.

As a summary of the previous work we can conclude that stochastic iteration is theoretically an effective tool that can reduce the complexity of global illumination algorithms and can result in non-diffuse global illumination methods that require just a single variable per patch, that is, their memory requirement is the same as that of the diffuse radiosity case. The critical part of the construction of such an algorithm is to find an appropriate random transport operator, that can be efficiently computed preferably with hardware support, thus the computation time of a single step is small. On the other hand, the variance introduced by the randomization should be also small, to keep the number of required iteration steps acceptable.

This paper proposes a stochastic iteration scheme that uses the hemicube to realize the random transport operator.

2 The new algorithm

In order to use the hemicube as the tool to randomly transfer the radiance in the scene, its origin, i.e. the patch and a point of the patch is selected randomly. Having identified the shooting point, its radiance is shot towards other points in the scene. Suppose that patch j is selected with

probability p_j and point $\vec{y}$ on this patch with uniform $1/A_j$ probability density. According to importance sampling, it is worth setting the selection probability p_i proportional to the powers of the patches.

Let us define the random transport operator as transferring the radiance $L(\vec{y}, \omega'_{\vec{y}\to\vec{x}})$ of this point, divided by its selection probability p_j/A_j, to all other visible points $\vec{x}$. When the radiance arrives at point $\vec{x}$, it is reflected according to the BRDF of the material here, and thus results in a new radiance value. Formally, the random transport operator is

$$(\mathcal{T}^* L)(\vec{x}, \omega) =$$

$$\frac{A_j}{p_j} \cdot v(\vec{x}, \vec{y}) \cdot L(\vec{y}, \omega'_{\vec{y}\to\vec{x}}) \cdot f_r(\omega'_{\vec{y}\to\vec{x}}, \vec{x}, \omega) \cdot \frac{\cos\theta'_{\vec{x}} \cdot \cos\theta_{\vec{y}}}{|\vec{x}-\vec{y}|^2}, \tag{4}$$

where $v(\vec{x}, \vec{y})$ is the mutual visibility indicator, which is 1 if the two points are visible from each other.

In order to show that this random transport operator is appropriate, we have to prove that the expected value of its effect gives back the application of the original light transport operator. The expected value of the random radiance after the transfer is:

$$E[\mathcal{T}^* L] = \sum_j p_j \cdot \int_{A_j} (\mathcal{T}^* L)(\vec{x}, \omega) \, \frac{d\vec{y}}{A_j} =$$

$$\sum_j \int_{A_j} v(\vec{x}, \vec{y}) \cdot L(\vec{y}, \omega'_{\vec{y}\to\vec{x}}) \cdot f_r(\omega'_{\vec{y}\to\vec{x}}, \vec{x}, \omega) \cdot \frac{\cos\theta'_{\vec{x}} \cdot \cos\theta_{\vec{y}}}{|\vec{x}-\vec{y}|^2} \, d\vec{y}.$$

Using the formula of solid angles $d\vec{y} \cdot \cos\theta_{\vec{y}}/|\vec{x}-\vec{y}|^2 = d\omega_{\vec{x}}$ and assuming that illumination can only come from surfaces — i.e. there is no external sky light illumination — the integration over all surfaces can be replaced by an integration over all incoming solid angles:

$$E[\mathcal{T}^* L] = \int_{\Omega'} L(h(\vec{x}, -\omega'), \omega') \cdot f_r(\omega', \vec{x}, \omega) \cdot \cos\theta'_{\vec{x}} \, d\omega'_{\vec{x}}.$$

Thus we could prove that the expectation of the application of the random operator really gives back the effect of the real transport operator, thus the requirement of equation 1 is met.

To obtain the radiance on the receiver patch, the radiances of its points, which have been computed according to equation 4, are averaged:

$$L(m)|_i = \frac{1}{A_i} \cdot \int_{A_i} \mathcal{T}^* L(m-1) \, d\vec{x} =$$

$$\frac{A_j}{p_j A_i} \int_{A_i} v(\vec{x}, \vec{y}) \cdot L(\vec{y}, \omega'_{\vec{y}\to\vec{x}}) \cdot f_r(\omega'_{\vec{y}\to\vec{x}}, \vec{x}, \omega) \cdot \frac{\cos\theta'_{\vec{x}} \cdot \cos\theta_{\vec{y}}}{|\vec{x}-\vec{y}|^2} \, d\vec{x}. \tag{5}$$

Let us interpret equation 5. The new radiance of patch i depends on the probability p_j of selecting the shooting patch, the radiance of the shooting point towards the receiver points $L(\vec{y}, \omega'_{\vec{y}\to\vec{x}})$, on a geometric factor

$$G_{\vec{y}\to A_i} = \frac{A_j}{A_i} \int_{A_i} v(\vec{x}, \vec{y}) \cdot \frac{\cos\theta'_{\vec{x}} \cdot \cos\theta_{\vec{y}}}{\pi|\vec{x}-\vec{y}|^2} \, d\vec{x},$$

and on the BRDF at the receiving point from the direction of the shooting point

$$\rho(\vec{y} \to \vec{x}) = f_r(\omega'_{\vec{y}\to\vec{x}}, \vec{x}, \omega) \cdot \pi.$$

Conceptually, this is very similar to the diffuse case except for the facts that we used the direction dependent radiance and the BRDF instead of the direction independent radiosity and diffuse albedo. Note that the formula has been divided by π in the geometric factor and multiplied by π in the BRDF, in order to give back the classical radiosity interpretation in the special case. The geometric factor can also be given a classical interpretation. Note that the integral in the geometric term is the point-to-patch form factor $F_{\vec{y}\to A_i}$. If patch j is small, then this point-to-patch form factor well approximates the patch-to-patch form factor $F_{A_j\to A_i}$, thus

$$G_{\vec{y}\to A_i} = \frac{A_j}{A_i} \cdot F_{\vec{y}\to A_i} \approx \frac{A_j}{A_i} \cdot F_{A_j\to A_i} = F_{A_i\to A_j}$$

according to the symmetry relation of the form factors. Despite to the conceptual similarities, the formulation of equation 5 is more complicated formally, since these factors cannot be decomposed and the radiance cannot be obtained as their simple product. The reason of this notational complexity is that now the radiance and the BRDFs depend on the direction as well. In fact, the product form is valid only for differential surface elements on patch i. However, if patch i is small compared with its distance to point $\vec{y}$, we can still use the following approximation:

$$L(m)|_i \approx \frac{L(\vec{y}, \omega'_{\vec{y}\to\vec{x}}) \cdot F_{A_i\to A_j} \cdot \rho(\vec{y} \to \vec{x})}{p_j} \tag{6}$$

where $\vec{x}$ is the center of patch A_i. We should emphasize that the implementation of the algorithm does not use this approximation. This formula, however, will be useful to understand the heuristic variance reduction technique, which is presented later in subsection 2.5 of this paper.

2.1 Representation of the radiance function

The radiance function now depends on both the surface point and the direction, thus its accurate finite-element representation would be too expensive. Fortunately, the radiance is needed only for computing the random transfer from a single patch and its image contribution. Note that these tasks require the radiance function just in a small subdomain compared to the set of all points and directions.

In order to compute the radiance transfers and the image contribution without explicitly storing the direction dependent radiance function itself, instead of the outgoing radiance L, the incoming radiance I is associated with each patch. If the sender and receiver patches are small compared to their distances, then we can assume that a patch may receive radiance only from a single direction, thus I is non-zero only for the direction pointing from the previously selected patch to the currently selected patch. Thus the incoming radiance is represented by two variables per patch, the intensity I and the direction of the transfer ω'. From the incoming radiance, the outgoing radiance can be obtained by a multiplication with the local BRDF:

$$L(\vec{x},\omega) = I(\omega') \cdot f_r(\omega', \vec{x}, \omega) \cdot \cos\theta'_{\vec{x}}.$$

2.2 Computation of the radiance transfer by hemicubes

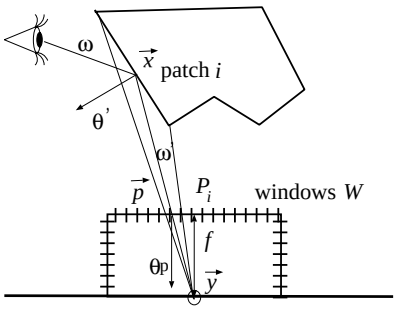

Figure 1: Hemicube shooting

In order to efficiently determine those $\vec{x}$ points that are visible from $\vec{y}$, the classical hemicube method can be used [4]. We can note that the integral in equation (5) can also be evaluated on the five window surfaces (W) that form a hemicube around the source $\vec{y}$ (figure 1). In the remaining part of this section, we re-derive the basic formulae to show that the hemicubes can also be used in cases when the reflection is non-diffuse.

To find formal expressions, let us express the solid angle $d\Omega_p$, in which a differential surface area $d\vec{x}$ is seen through pixel area $d\vec{p}$, both from the surface area and from the pixel area:

$$d\Omega_p = \frac{d\vec{x} \cdot \cos\theta'_{\vec{x}}}{|\vec{y}-\vec{x}|^2} = \frac{d\vec{p} \cdot \cos\theta_p}{|\vec{y}-\vec{p}|^2}, \qquad (7)$$

where θ_p is the angle between the direction pointing to $\vec{x}$ from $\vec{y}$ and the normal of the window (figure 1). The distance $|\vec{y}-\vec{p}|$ between pixel point $\vec{p}$ and the radiance source $\vec{y}$ equals to $f/\cos\theta_p$ where f is the distance from $\vec{y}$ to the window plane, that is also called the *focal distance*. Using this and equation (7), differential area $d\vec{x}$ can be expressed and substituted into equation (5), thus we can obtain:

$$L(m)|_i = \frac{A_j}{p_j A_i f^2} \cdot \int_W v(\vec{y},\vec{x}) \cdot L(\vec{y}, \omega'_{\vec{y}\to\vec{p}}) \cdot f_r(\omega'_{\vec{y}\to\vec{x}}, \vec{x}, \omega) \cdot \cos\theta_{\vec{y}} \cdot \cos^3\theta_p \, d\vec{p}.$$

Let P_i be the set of those pixels in which patch i is visible from $\vec{y}$. P_i is computed by running a z-buffer/constant shading rendering step for each sides of the window surface, assuming that the color of patch i is i, then reading back the "images". The reflected radiance on patch i is approximated by a discrete sum as follows:

$$L(m)|_i \approx \frac{A_j \delta P}{p_j A_i f^2} \cdot \sum_{P_i} L(\vec{y}, \omega'_{\vec{y}\to\vec{p}}) \cdot f_r(\omega'_{\vec{y}\to\vec{x}}, \vec{x}(\vec{p}), \omega) \cdot \cos\theta_{\vec{y}} \cdot \cos^3\theta_p, \qquad (8)$$

where δP is the area of a single pixel in the image. If R is the resolution of the image — i.e. the top of the hemicube contains $R \times R$ pixels, while the side faces contain $R \times R/2$ pixels – then $\delta P = 4f^2/R^2$.

2.3 Randomization of the hemicube

It is easy to see that when computing the point-to-patch form factors by hemicubes, sometimes it is not efficient to render the scene across all sides of the hemicube. This is the situation when the power distributed through a specific side of the hemicube is negligible. This can be caused by the low incoming radiance or by the anti-symmetry of the outgoing radiance. The main cause of this anti-symmetry is the specular characteristic of the surface, which transfers most of the radiance towards the ideal reflection direction. If the ratio of specular albedo and diffuse albedo and the *shine* parameter of the Phong illumination formula is large, most of the light power is transferred through that specific side of the hemicube, which lies in the direction of the ideal reflection. In these cases it is useless to use the other sides of the hemicube.

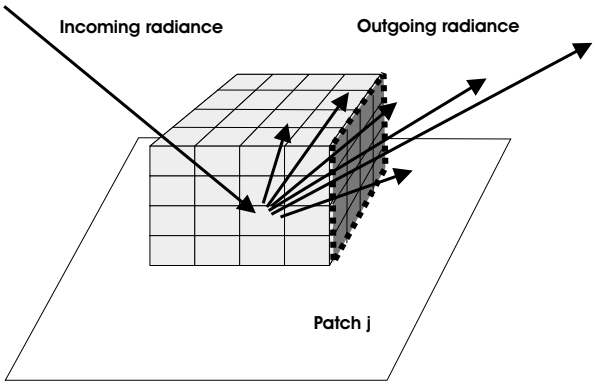

Figure 2: On specular surfaces most of the illumination goes through just 1 side of the hemicube.

Since we do not want to loose the unbiasedness characteristics of the method, but we want to save computation time when it is possible, we introduce randomization into

the process. When the selected surface has quite strong specular characteristics, we randomly select one side of the hemicube and propagate the radiance through just the selected side. In that cases, according to Russian Roulette, the transported radiance is divided by the selection probability.

2.4 Importance sampling

The stochastic iteration will converge quickly if the random noise added by a single iteration step is small. This means that the randomization of the light transfer should not be too strong (the optimal level is determined by the efficiency of transferring the radiance by the random operator and the variance caused by the randomization).

The variance of the random transfer can be decreased by the variance reduction techniques of the Monte-Carlo literature, and particularly by importance sampling. According to importance sampling, the selection probability is good if it mimics the original integrand, thus the total transferred power, which is computed as the real power divided by the selection probability, should be roughly constant. This means that the patches are worth selecting with a probability that is proportional to their output power, and the sides of the hemicube according to the power transferred through them.

Recall that before a given iteration step, we store the emission and the incoming radiance of each patch. If a patch is selected, then it will shoot the following power:

$$\Phi_j = \Phi_j^e + A_j \cdot \cos\theta' \cdot I(\omega'_{m-1}) \cdot a(\omega'_{m-1})$$

where Φ_j^e is the emission power and $a(\omega'_{m-1})$ is the local albedo. Thus in each iteration step, power Φ_j is computed for each patch, and the patch selection is realized with $\Phi_j / \sum_k \Phi_k$ probability.

The second level of randomization controls the identification of those hemicube sides through which the transfer is computed. This should be proportional to the represented solid angle and the average radiance in the directions of the solid angle. We used a very simple heuristic scheme. If the surface is highly specular, the algorithm selects that side which is intersected by the ideal reflection direction by 0.6 probability and all the other sides by 0.1 probability. If the ratio of the specular and diffuse albedos is smaller then 1, more than half of the power is distributed by diffuse light transfer. Since there is a cosine term in the transferred radiance formula, in these cases most of the radiance goes through the top of the hemicube. Thus it is worth selecting the top deterministically and using one from the four sides randomly.

2.5 Variance reduction by trading bias with noise

If we implement and run the algorithm described so far, we can realize that the general convergence of the image will be very fast, but embarrassing noise occurs at corners and at object boundaries (figure 3). This problem is mentioned by the Monte-Carlo radiosity literature, but so far it has not been solved.

Figure 3: An office scene rendered without the biased variance reduction.

The explanation of these irritating artifacts is the following. Darker patches are very seldomly selected by the algorithm. However, when they are selected, the transferred power is divided by the small probability, thus even dark patches can result in large power transfers. When the receiver patches are very close to the shooting patch, then the point-to-patch form factor is large, thus the receiving patch gets too much power and tends to be too bright in the image. As stochastic iteration proceeds this annoying artifact slowly disappears. Theoretically there is nothing bad or unusual with these too bright patches, this behavior is caused by the random noise, which is inherent in all Monte-Carlo methods. As the iteration number goes to infinity, the radiance values will converge to the mean of these random variables.

However, when we want to have accurate and nice images quickly, i.e. after just a few hundred iterations, the fact that the irritating bright patches would disappear if we were running the algorithm for much longer time, is not acceptable. Fortunately a solution exists that can successfully attack this problem, which trades bias for noise in a way that for a given iteration number the result will not be unbiased, but the total error of the bias and the Monte-Carlo noise will be still smaller than the Monte-Carlo noise of the original algorithm.

Let us return to the approximation of the radiance of

patch i after an iteration step (equation 6):

$$L(m)|_i \approx \frac{L(\vec{y}, \omega'_{\vec{y}\to\vec{x}}) \cdot F_{A_i\to A_j} \cdot \rho(\vec{y}\to\vec{x})}{p_j}.$$

The expected value of this random variable is

$$E[L(m)|_i] \approx E[L(\vec{y}, \omega'_{\vec{y}\to x_j})] \cdot F_{A_i\to A_j} \cdot \rho(\vec{y}\to\vec{x}).$$

If M iteration steps are computed altogether, then the probability of selecting patch j for shooting at least once is $1-(1-p_j)^M$. If p_j is really small, because patch j is not a light source, its size is also small and it does not receive significant illumination, then the selection probability is reasonably smaller than one and can be approximated as

$$1-(1-p_j)^M \approx p_j M.$$

The fact that these patches are not selected at all is not a problem in itself since according to the fundamental assumption of hierarchical and Monte-Carlo methods, patches form homogeneous groups and using one patch in these groups can also simulate the radiance transfer of other elements of the group. Suppose that the selected patch is a member of such a homogeneous group consisting of k similar patches. Then the probability of selecting at least one member of this group is $1-(1-kp_j)^M$. The real problem happens when even this group selection probability is much smaller than one. In this case, this probability is roughly kp_jM. Statistically, such a group should not be used for radiance transfer in the M step long iteration, but the random selection might find elements also in this group. If this patch group is selected $n>0$ times, the random estimator is:

$$\hat{L}(m)|_i = \tilde{L}(\vec{y}, \omega'_{\vec{y}\to A_j}) \cdot F_{\vec{y}\to A_i} \cdot \rho(\vec{y}\to\vec{x}) \cdot \frac{n}{kp_jM},$$

where $\tilde{L}$ is the average of the radiances in these transfers. The distance of this estimator from the expected value can be bigger than the distance between the expected value and zero, when it is worth replacing the transferred radiance by zero. If we assume that the average radiance $\tilde{L}$ is approximately equal to its expected value, then the criterion of replacing the transfer by zero is:

$$\hat{L}(m)|_i - E[L(m)|_i] > E[L(m)|_i] - 0 \Longrightarrow p_j < \frac{1}{2kM}.$$

In order to find an upper-bound for k, notice that the geometric parameters of the transfer are characterized by form factor $F_{\vec{y}\to A_i}$. Different patches behave similarly in this transfer if their respective form factors are also similar. The sum of form factors is at most one (exactly one in closed and less than one in open scenes), that is

$$\sum_j F_{A_i\to A_j} \le 1.$$

It means that the number of patches that have roughly this form factor with patch i is bounded by the inverse of the form factor, thus for k we obtain:

$$k \le \frac{1}{F_{A_i\to A_j}}.$$

This allows to establish the limit of probability where the radiosity transfer is not worth executing:

$$p_j < \frac{F_{A_i\to A_j}}{2M}. \tag{9}$$

The modified algorithm works similarly as the previous one, it selects patches randomly and computes the radiance transfer from the selected patch towards those patches that are visible from here. In order to compute the new radiance value of the receivers, the form-factor is also computed. However, when it turns out that selection probability of the shooting patch is smaller as the limit of inequality 9, then this particular receiving patch is assumed to get zero radiance in this iteration step.

Note that this trick steals energy from the system, thus for a fixed iteration number M the result will be biased. However, the error is still less than in the unbiased estimate. On the other hand, the bias disappears as M goes to infinity, thus the method is still unbiased in the asymptotic case.

For very small M values, the missing energy becomes noticeable at the corners since they are darker than expected. Although this is still much better than the too bright patches, this problem can be further reduced by a special type of mean value substitution. When it turns out that the random estimator of the current transfer is too large, then instead of replacing it by zero, it can be replaced by its approximated mean $L(\vec{y}, \omega'_{\vec{y}\to A_j}) \cdot F_{\vec{y}\to A_i} \cdot \rho(\vec{y}\to\vec{x})$. This is as accurate as the radiance L is close to its converged value.

2.6 Application of the constant radiance term

Usually just a fraction of the patches belong to light sources. Importance sampling on the other hand will probably select the shooting patches from the light sources. One alternative for making it better is the first-shot technique, but since it selects the center of the light source patches deterministically, it introduces bias.

Anyway, the hemicube shooting by nature is very good at performing first-shot, so doing a first-shot before starting the algorithm seems unnecessary. On the other hand, the selection probability of the non light source patches can be significantly improved by transforming the radiance function to a function with smaller amplitude.

A constant radiance value is extracted from the solution in every surface point and direction. However, we should be careful, when choosing this constant value. The optimal constant value is hard to find for each patch separately, but even a conservative estimate can improve the convergence.

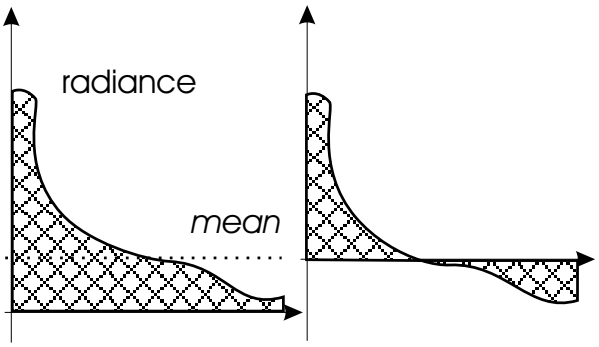

Figure 4: When subtracting the mean, the importance sampling more probably selects the non light sources.

The average radiance $\overline{L}$ can be computed by the assumption that all points have the same BRDF and albedo, that are computed as the average [12].

The average radiance is determined by:

$$\overline{L} = \frac{\int_{\Omega}\int_{S} L^e(\vec{x},\omega)\cos\theta d\vec{x}d\omega}{S\pi(1-a_{mean})}.$$

where a_{mean} is the average albedo of the scene, calculated by:

$$a_{mean} = \frac{1}{S\pi}\cdot\int_{\Omega}\int_{S} a(\vec{x},\omega)\cos\theta d\vec{x}d\omega.$$

Formally, let us decompose the radiance function into this average $\overline{L}$ and a distance from the average ΔL. Substituting $L = \overline{L} + \Delta L$ into the rendering equation, we can obtain:

$$\Delta L(\vec{x},\omega) = L^e(\vec{x},\omega) + (a(\vec{x},\omega)-1)\cdot\overline{L} + \mathcal{T}_{f_r}\Delta L.$$

Note that we obtained a rendering equation for the ΔL term, having modified the emission function with $(a(\vec{x},\omega)-1)\overline{L}$. This new light source term is negative for physically valid scenes, which means that the non light source patches emit negative power. Therefore, when doing importance sampling we should use the absolute values of power for computing the selection probability of the patches. After the iteration finishes and $\Delta L(\vec{x},\omega)$ is obtained, we should add the average radiance $\overline{L}$ to the final result.

3 Results

The presented algorithm have been implemented in C++ in OpenGL environment. The images have been rendered with 500×500 resolution. The faces of the hemicube had 600×600 pixels. The algorithm can render moderately complex scenes within a minute.

We tried the hemicube randomization with scenes of specular reflectance, and according to our experience the

Figure 5: R2D2 meets an alien in the Cornell-box. The image was rendered with the new method using the biased variance reduction and the constant radiance step.

Figure 6: An office scene rendered with the new method using biased variance reduction.

hemicube randomization resulted in 5-15 percent speed-up. Without using the constant radiance step, importance sampling selected the light sources very frequently, i.e. in about 50 percents of iterations. When applying the constant radiance step, this decreased to 32 percent. This trick increased the speed by another ten percent.

When the Cornell box scene with R2D2 and the alien (figure 5) was rendered, the 500 iterations needed 50 seconds on a Pentium III 1Ghz computer using GeForce2 MX graphics hardware. Note the specular highlight on the back wall and on the body of R2D2. The specular albedos were set to 0.27 and 0.2 and the shine parameters of the Phong BRDF to 28 and 10, respectively. Other surfaces also have specular albedos, usually in the range of $0.05 - 0.0$. Rendering the office room (figure 6), which contains a specular vase, took about 60 seconds and needed 600 iterations.

4 Conclusions

This paper has presented a new stochastic rendering technique that is based on the randomization of the classical hemicube approach. This randomization speeds up the convergence of the algorithm and ensures rendering non-diffuse scenes with the same storage space as required by the radiosity method. In order to get rid of the artifacts of the randomization, we proposed trading of noise with bias in a way that the error gets smaller, but the algorithm is still unbiased asymptotically. We also discussed some improvements for the basic algorithm, which included the random selection of the hemicube faces and the application of the constant radiance step.

References

[1] P. Bekaert, L. Neumann, A. Neumann, M. Sbert, and Y. Willems. Hierarchical Monte-Carlo radiosity. In *Rendering Techniques '98*, pages 259–268, 1998.

[2] P. H. Christensen, D. Lischinski, E. J. Stollnitz, and D. H. Salesin. Clustering for glossy global illumination. *ACM Transactions on Graphics*, 16(1):3–33, 1997.

[3] P. H. Christensen, E. J. Stollnitz, D. H. Salesin, and T. D. DeRose. Global illumination of glossy environments using wavelets and importance. *ACM Transactions on Graphics*, 15(1):37–71, 1996.

[4] M. Cohen and D. Greenberg. The hemi-cube, a radiosity solution for complex environments. *Computer Graphics (ACM SIGGRAPH '85 Proceedings)*, pages 31–40, 1985.

[5] M. Cohen, E. Chen, J. Wallace, D. Greenberg. A Progressive Refinement Approach to Fast Radiosity Image Generation. *Computer Graphics (ACM SIGGRAPH '88 Proceedings)*, 22(4), 1988.

[6] C. Goral, K. Torrance, D. Greenbert, B. Battaile. Modeling the Interaction of Light between Diffuse Surfaces. *Computer Graphics (ACM SIGGRAPH '84 Proceedings)*, 18(3), 1984.

[7] D. S. Immel, M. F. Cohen, and D. P. Greenberg. A radiosity method for non-diffuse environments. In *Computer Graphics (SIGGRAPH '86 Proceedings)*, pages 133–142, 1986.

[8] H. W. Jensen. Global illumination using photon maps. *Rendering Techniques '96*, pages 21–30, 1996.

[9] Kajiya, J. T. The Rendering Equation. *Computer Graphics (ACM SIGGRAPH '86 Proceedings)*, pages 143–150, 1986.

[10] L. Neumann. Monte Carlo radiosity. *Computing*, 55:23–42, 1995.

[11] L. Neumann, M. Feda, M. Kopp, W. Purgathofer. A New Stochastic Radiosity Method for Highly Complex Scenes. *Rendering Techniques '94*, pages 201–213, 1994.

[12] L. Neumann, A. Neumann, J. Prikryl, W. Purgathofer The Constant Radiance Term. *Machine Graphics & Vision*, 7(3), pages 535–549, 1998.

[13] M. Sbert. *The Use of Global Directions to Compute Radiosity*. PhD thesis, Catalan Technical University, Barcelona, 1996.

[14] P. Shirley. Time complexity of Monte-Carlo radiosity. In *Eurographics '91*, pages 459–466. Elsevier Science Publishers, 1991.

[15] P. Shirley. Radiosity via Ray Tracing. In James Arvo, editor, *Graphics Gems II*, pages 306–310, Academic Press, San Diego, 1991.

[16] F. X. Sillion, J. R. Arvo, S. H. Westin, and D. P. Greenberg. A global illumination solution for general reflectance distributions. *Computer Graphics (SIGGRAPH '91 Proceedings)*, 25(4):187–198, 1991.

[17] M. Stamminger, A. Scheel, A. Granier, F. Perez-Cazorla, G. Drettakis, and F. Sillion. Efficient glossy global illumination with interactive viewing. In *Graphics Interface'99, Kingston, Ontario*, 1999.

[18] L. Szirmay-Kalos (editor). *Theory of Three Dimensional Computer Graphics*. Akadémia Kiadó, Budapest, 1995. http://www.iit.bme.hu/~szirmay.

[19] L. Szirmay-Kalos. Stochastic iteration for non-diffuse global illumination. *Computer Graphics Forum (Eurographics'99)*, 18(3):233–244, 1999.

[20] L. Szirmay-Kalos. *Photorealistic Image Synthesis with Ray-Bundles*. Hungarian Academy of Sciences, D.Sc. Dissertation, Budapest, 2000. http//www.iit.bme.hu/~szirmay.