Non-diffuse, Random-walk Radiosity Algorithm with Linear Basis Functions

László Szirmay-Kalos, Tibor Fóris, Werner Purgathofer
Department of Control Engineering and Information Technology, Technical University of Budapest,
Budapest, Műegyetem rkp. 11, H-1111, HUNGARY
szirmay@fsz.bme.hu

Abstract. This paper presents an efficient method to solve the general rendering equation, using a combined finite element and quasi-random walk approach. Applying point collocation method, the surfaces are decomposed into planar patches where the directional distribution of the radiance is assumed to be a linear combination of the distributions at the vertices. The direction dependent radiance function of the vertices is then computed by random or quasi-random walk.

Key words: Rendering equation, quasi-monte carlo quadrature, radiosity, point-collocation method.

1. Introduction

The fundamental task of computer graphics is to solve a Fredholm type integral equation describing the light transport. This equation is called *the rendering equation* and has the following form:

$$L(\vec{x},\omega) = L^e(\vec{x},\omega) + \int^{4\pi} L(h(\vec{x},-\omega'),\omega') \cdot \cos\theta' \cdot f_r(\vec{x},\omega',\omega)\; d\omega'$$

where $L(\vec{x},\omega)$ is the radiance of the surface in point $\vec{x}$ at direction ω, $h(\vec{x},\omega')$ is the visibility function defining the point that is visible from point $\vec{x}$ at direction ω', θ' is the angle between the surface normal and direction $-\omega'$, and $f_r(\vec{x},\omega',\omega)$ is the bi-directional reflection/refraction function (figure 1).

Generally two methods exist that can solve the rendering equation without simplifying the transport laws:

- finite element methods [3] which convert the integral equation into an approximating linear equation that can be solved by straightforward techniques,
- random walk methods [8] that expand the rendering equation into a Neumann series, and evaluate its terms by multi-dimensional quadrature.

We encounter a serious difficulty when the finite element techniques are applied to the radiance equation. The function to be found is 4-variate and has high-frequency components, thus the number of required basis functions, that approximate the solution as a linear combination, should be prohibitively high.

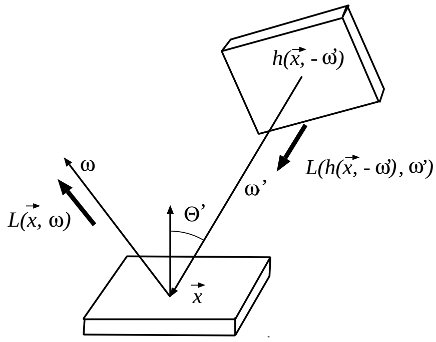

Fig. 1. Geometry of the rendering equation

Random walk methods, on the other hand, require the solution of very high-dimensional integrals. The approximation error of integrals may be different from pixel to pixel, resulting in a noisy image. In order to reduce noise, the number of walks is increased and the image is generated as an average of the contributions of individual walks.

2. Previous Work on Non-diffuse Radiosity

Finite element techniques that aim at the solution of the non-diffuse case can be traced back to the finite element approximation of the directional functions using *partitioned sphere* [4] and to the application of *extended form factors* [7].

The first method calculates and stores radiance at each solid angle derived from a partitioned sphere. An improved version applies spherical harmonics to represent the directional distribution. Extended form factors, on the other hand, represent both the direct-diffuse and the indirect-coherent light transfer of patches, and are usually evaluated by recursive ray-tracing.

Monte-Carlo random walk methods generate finite samples of the light propagation from the lightsources to the eye. Since a photon may bounce on surfaces several times until it reaches the eye, the propagation consists of several surface to surface interaction, called steps. The series of these steps composes the walk.

In computer graphics the first Monte-Carlo random walk algorithm — called *distributed ray-tracing* — was proposed by Cook *et al.* [2], which spawned to a set of variations, including *path tracing* [5], *light-tracing* [13], *Monte-Carlo radiosity* [9][17][10], and two-pass methods which combine radiosity and ray-tracing [6].

Random-walks can be classified according to the direction they follow the light. A random-walk can be initiated from the lightsources and be traced in the direction of the light (*shooting*), or it can be started at the eye position and traced in the direction opposite to the light (*gathering*). Shooting type walks are also called *light walks*, while gathering type walks are called *eye walks*, emphasizing the starting point of the walk.

The problem of these methods is that the probability that a shooting path finds the eye is zero for a pin-hole camera or very small if non-zero aperture camera modell is used, while the probability that a gathering random path ends in a lightsource may be very little if the lightsources are small.

Thus random walk must be combined with a deterministic step that forces the walk to go to the eye or to find a lightsource. *Light tracing* connects each bounce position to the eye deterministically [13]. *Bi-directional ray-tracing* methods start a shooting type and a gathering type walks and connect the bounce positions of the two walks. In a method proposed by Veach and Guibas [15] only the last positions of the two walks are connected. Lafortune [14], on the other hand, suggested that all positions visited during light-walks may be combined with the positions of eye-walks using a weighted sum.

Quasi-Monte Carlo techniques have been first applied to solve the diffuse radiance equation by Keller [18], where the integrand was generally discontinuous and therefore of infinite variation, thus the superiority of quasi-Monte Carlo method could not been theoretically justified. However, the numerical evidence showed that quasi-Monte Carlo methods can slightly be better than Monte-Carlo techniques.

Due to the visibility term $h(\vec{x}, -\omega')$ in the rendering equation, an estimation of d-th bounce of the light requires the solution of d number of visibility problems. The most obvious way for this is ray-shooting which emanates a ray from point $\vec{x}$ at direction ω' and finds the nearest intersection.

However, this approach may waste a lot of computation by ignoring all the intersection objects but the one closest to the eye. Thus it seems worth using a set of *global directions* [19] and [17] for the complete scene instead of solving the visibility problem independently for different points $\vec{x}$. Moreover, ray-shooting is a simple but by no means the most effective visibility algorithm. Other methods, such as the z-buffer, painter's, Warnock's, *etc.* algorithms can be considered as handling a bundle of parallel rays and solving the visibility problem for all of them simultaneously.

3. The Combined Finite-Element and Random-Walk Method

This section presents a new algorithm that decomposes the scene into a finite mesh. The light transfer between the patches are described by an integral equation consisting of surface and directional integrals. First, the surface integral is handled by finite element methods — namely the *point collocation* method — then the resulting integral equation, which contains only directional integration, is solved by quasi-Monte Carlo walks.

3.1. Reformulation of the Rendering Equation Using Finite-Element Method

Using finite element concepts, the radiance function is searched in the following form:

$$L(\vec{x}, \omega) \approx L^{(n)}(\vec{x}, \omega) = \sum_{j=1}^{n} L_j(\omega) \cdot b_j(\vec{x}), \tag{1}$$

where $L^{(n)}(\vec{x}, \omega)$ is the approximating radiance and $b_j(\vec{x})$ is a complete function system. In this paper b_j is selected to be linear.

Intuitively it means that the surfaces are decomposed into planar triangles and the radiosity variation is assumed to be linear on these triangles. Thus, each vertex i of the triangle mesh will correspond to a "tent shaped" basis function b_i that is 1 at this vertex and linearly decreases to 0 on the triangles incident to this vertex (figure 2).

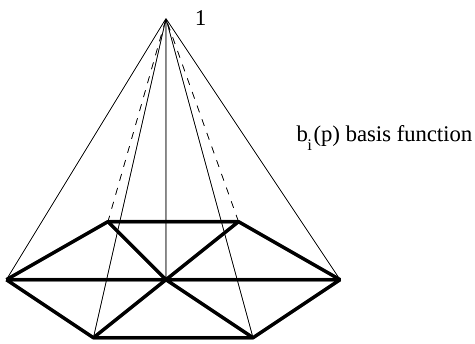

Fig. 2. Linear basis function in 3-dimension

Substituting (1) into the original rendering equation we obtain:

$$\sum_j L_j(\omega) \cdot b_j(\vec{x}) = \sum_j L_j^e(\omega) \cdot b_j(\vec{x}) + \int^{4\pi} \sum_j L_j(\omega') \cdot b_j(h(\vec{x}, -\omega')) \cdot \cos\theta' \cdot f_r(\vec{x}, \omega', \omega)\, d\omega'.$$

In the point-collocation method, the unknown directional functions $L_j(\omega)$ is determined to ensure that the residual of the approximation is zero at the vertices of the triangle mesh. The residual is zero at vertex $\vec{x}_i$ if

$$L_i(\omega) = L_i^e(\omega) + \sum_j \int^{4\pi} L_j(\omega') \cdot b_j(h(\vec{x}_i, -\omega')) \cdot \cos\theta' \cdot f_r(\vec{x}_i, \omega', \omega)\, d\omega',$$

since at point $\vec{x}_i$ all basis functions are zero except for b_i which has a value 1.

The same equation can also be presented in a matrix form:

$$\mathbf{L}(\omega) = \mathbf{L}^e(\omega) + \int^{4\pi} \mathbf{F}(\omega', \omega) \cdot \mathbf{A}(\omega') \cdot \mathbf{L}(\omega')\, d\omega',$$

where $\mathbf{L}(\omega)|_i = L_i(\omega)$ is the vector of radiance values, and $\mathbf{F}(\omega', \omega)$ is the diagonal matrix of the BRDF functions at the vertices, and

$$\mathbf{A}(\omega')|_{ij} = b_j(h(\vec{x}_i, -\omega')) \cdot \cos\theta'$$

is the geometry matrix. The complete transport operator is the product of the geometry and the BRDF matrices:

$$\mathbf{T}(\omega', \omega) = \mathbf{F}(\omega', \omega) \cdot \mathbf{A}(\omega').$$

3.2. Quasi-Random Walk Methods

In order to simplify the notations, the integral operator of the rendering equation is denoted by $\mathcal{T}$:

$$\int^{4\pi} \mathbf{T}(\omega', \omega) \cdot \mathbf{L}(\omega')\, d\omega' = \mathcal{T}\mathbf{L}(\omega).$$

Thus the short form of the rendering equation is:

$$\mathbf{L}(\omega) = \mathbf{L}^e(\omega) + \mathcal{T}\mathbf{L}(\omega). \tag{2}$$

In (2) the unknown radiance function $\mathbf{L}(\omega)$ appears on both sides. Substituting the right side's $\mathbf{L}(\omega)$ by the complete right side, which is obviously $\mathbf{L}(\omega)$ according to the equation, we get:

$$\mathbf{L}(\omega) = \mathbf{L}^e(\omega) + \mathcal{T}\mathbf{L}(\omega) = \mathbf{L}^e(\omega) + \mathcal{T}(\mathbf{L}^e(\omega) + \mathcal{T}\mathbf{L}(\omega)) = \mathbf{L}^e(\omega) + \mathcal{T}\mathbf{L}^e(\omega) + \mathcal{T}^2\mathbf{L}(\omega).$$

Repeating this step m times, the original equation can be expanded into a Neumann series:

$$\mathbf{L}(\omega) = \sum_{i=0}^{m} \mathcal{T}^i\mathbf{L}^e(\omega) + \mathcal{T}^{m+1}\mathbf{L}(\omega).$$

If integral operator $\mathcal{T}$ is contractive, that is if

$$||\mathcal{T}\mathbf{L}(\omega)|| \leq \lambda \cdot ||\mathbf{L}(\omega)||, \qquad \lambda < 1,$$

then $\lim_{m\to\infty} \mathcal{T}^{m+1}\mathbf{L} = 0$, thus

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

The contractive property of $\mathcal{T}$ comes from the fact that a reflection or refraction always decreases the energy. Using, for example, the infinite norm, we obtain

$$||\mathcal{T}\mathbf{L}(\omega)||_\infty \leq \max_i \int^{4\pi} f_r(\vec{x}_i, \omega', \omega) \cdot \cos\theta'\, d\omega' \cdot ||\mathbf{L}(\omega)||_\infty = \max_i a_i(\omega) \cdot ||\mathbf{L}(\omega)||_\infty,$$

where $a_i(\omega)$ is the *albedo* [12] of the material at vertex i. For physically plausible material models the albedo must be less than 1.

The terms of this infinite Neumann series have intuitive meaning as well: $\mathcal{T}^0\mathbf{L}^e(\omega) = \mathbf{L}^e(\omega)$ comes from the emission, $\mathcal{T}^1\mathbf{L}^e(\omega)$ comes from a single reflection (called 1-bounce), $\mathcal{T}^2\mathbf{L}^e(\omega)$ from two reflections (called 2-bounces), etc.

Using the definition of integral operator $\mathcal{T}$, the full form of the Neumann series is:

$$\begin{aligned}
\mathbf{L}(\omega) = \mathbf{L}^e(\omega) \quad &+ \quad \int^{4\pi} \mathbf{T}(\omega'_1, \omega) \cdot \mathbf{L}^e(\omega'_1)\, d\omega'_1 \\
&+ \quad \int^{4\pi}\int^{4\pi} \mathbf{T}(\omega'_1, \omega) \cdot \mathbf{T}(\omega'_2, \omega'_1) \cdot \mathbf{L}^e(\omega'_2)\, d\omega'_2 d\omega'_1 \\
&+ \quad \int^{4\pi}\int^{4\pi}\int^{4\pi} \mathbf{T}(\omega'_1, \omega) \cdot \mathbf{T}(\omega'_2, \omega'_1) \cdot \mathbf{T}(\omega'_3, \omega'_2) \cdot \mathbf{L}^e(\omega'_3)\, d\omega'_3 d\omega'_2 d\omega'_1 + \ldots
\end{aligned}$$

In practice the infinite sum of the Neumann series is always approximated by a finite sum. The number of required terms is determined by the contractivity λ of operator $\mathcal{T}$. Let us denote the maximum number of calculated bounces by D. The truncation of Neumann series introduces a bias in the estimation, which can be tolerated if D is high enough.

In order to simplify the notations, we introduce the *max d-bounce irradiance* $\mathbf{I}_d$ for $d = 1, 2, \ldots$ as follows:

$$\begin{aligned}
\mathbf{I}_0(\omega'_D) \quad &= \quad \mathbf{A}(\omega'_D) \cdot \mathbf{L}^e(\omega'_D) \\
\mathbf{I}_d(\omega'_{D-d}, ..., \omega'_D) \quad &= \quad \mathbf{A}(\omega'_{D-d}) \cdot \big(\mathbf{L}^e(\omega'_{D-d}) + 4\pi \cdot \mathbf{F}(\omega'_{D-d+1}, \omega'_{D-d}) \cdot \mathbf{I}_{d-1}(\omega'_{D-d+1}, ...\omega'_D)\big).
\end{aligned}$$

Limiting the walk to D number of bounces, the solution of the rendering equation is obtained as a single $2D$-dimensional integral:

$$\mathbf{L}(\omega) = \mathbf{L}^e(\omega) + (\frac{1}{4\pi})^{D-1} \int^{4\pi}\int^{4\pi} \ldots \int^{4\pi} \mathbf{F}(\omega'_1, \omega) \cdot \mathbf{I}_D(\omega'_1, \omega'_2, \ldots, \omega'_D)\, d\omega'_D \ldots d\omega'_2 d\omega'_1.$$

This high-dimensional integral can be estimated by quasi-Monte Carlo quadrature.

3.3. Quasi-Monte Carlo Integration

Since the low-discrepancy sequences are defined in a unit-cube of appropriate dimension, the points in the unit cube should be projected onto the directional hemisphere. In order to generate a uniform distribution on the surface of the directional sphere, the following transformation can be used [8]:

$$\psi = 2\pi u, \qquad \theta = \arccos(1 - 2v).$$

In this case

$$d\omega = det \begin{bmatrix} \frac{\partial\psi}{\partial u} & \frac{\partial\psi}{\partial v} \\ \frac{\partial\theta}{\partial u} & \frac{\partial\theta}{\partial v} \end{bmatrix} \cdot \sin\theta(\omega(u, v))\; du\; dv = 4\pi \cdot du\; dv.$$

thus the spatial distribution is uniform on the sphere if it was in the unit square. Note that the fact that this transformation projects equal areas to equal solid angles does not mean that the transformation also projects equal distances to the same spherical distances.

The quasi-Monte Carlo approximation of the radiance vector is:

$$\mathbf{L}(\omega) = \mathbf{L}^e(\omega) + (\frac{1}{4\pi})^{D-1} \int^{4\pi}\int^{4\pi} \ldots \int^{4\pi} \mathbf{F}(\omega'_1, \omega) \cdot \mathbf{I}_D(\omega'_1, \omega'_2, \ldots, \omega'_D)\ d\omega'_D \ldots d\omega'_2 d\omega'_1$$
$$\approx (\frac{1}{M}) \sum_{i=1}^{M} \left(\mathbf{L}^e(\omega) + 4\pi \cdot \mathbf{F}(\omega_1^{(i)}, \omega) \cdot \mathbf{I}_D(\omega_1^{(i)}, \omega_2^{(i)}, \ldots, \omega_D^{(i)})\right)$$

where M is the number of samples in the quadrature.

In order to evaluate this formula, M number of quasi-random walks should be generated, where the directions are sampled from a $2D$-dimensional low-discrepancy sequence, such as the $2D$-dimensional *Hammersley sequence* [11]. When the D-bounce irradiance is available, it is multiplied by the BRDF defined by the last direction ω_1 and the viewing direction ω to find the radiance of this patch that is visible from the eye position. Note that this step makes the algorithm view-dependent. The viewing direction is different from patch to patch if the eye is not at infinity. The complete algorithm — which requires two variables per vertex, one for the max d-bounce irradiance $\mathbf{I}_v$ and one for the accumulating radiance $\mathbf{L}_v$ — is summarized in the following:

for $i = 1$ **to** M **do** // *global light walks*
  Generate the ith low-discrepancy point: $(\omega_1^{(i)}, \omega_2^{(i)}, \ldots, \omega_D^{(i)})$
  $\mathbf{I} = 0$
  for $d = 0$ **to** $D - 1$ **do** // *quasi-random walk*
    $\mathbf{I} = \mathbf{A}(\omega_{D-d}^{(i)}) \cdot \left(\mathbf{L}^e(\omega_{D-d}^{(i)}) + 4\pi \cdot \mathbf{F}(\omega_{D-d+1}^{(i)}, \omega_{D-d}^{(i)}) \cdot \mathbf{I}\right)$
  endfor
  for each vertex $\vec{x}_v$ **do** // *accumulation of the contributions*
    Calculate viewing direction ω
    $\mathbf{L}_v \mathrel{+}= (\mathbf{L}_v^e(\omega) + 4\pi \cdot \tilde{f}_r(\vec{x}_v, \omega_1^{(i)}, \omega) \cdot \mathbf{I}_v)/M$
  endfor
endfor
Render Image

4. Calculation of the Transport Matrix

A crucial part of the method is the determination of the radiance transfer to different vertices, which requires the evaluation of the following formula

$$\mathbf{I}_v = \sum_j b_j(h(\vec{x}_v, -\omega'_{D-d})) \cdot \cos\theta' \cdot (\mathbf{L}_v^e(\omega'_{D-d}) + \tilde{f}_r(\vec{x}_v, \omega'_{D-d+1}, \omega'_{D-d}) \cdot \mathbf{I}_j) \qquad (3)$$

Note that $b_j(h(\vec{x}_v, -\omega'_{D-d}))$ is non-zero for those j indices which represent a vertex of the patch visible from $\vec{x}_v$ at direction ω'_{D-d}. The exact value can be derived from the calculation of the height of the linear "tent" function at point $h(\vec{x}_v, -\omega'_{D-d})$.

In the algorithm the evaluation of (3) is needed for all v indices in a fixed direction ω'_{D-d}. This is a global visibility problem where the viewing direction is fixed but the eye position visits all vertices.

An efficient way to solve this problem is based on the famous painter's algorithm. The triangular patches are sorted in direction ω'_{D-d}, then painted one after the other into an image buffer (such a sorting is not obvious if patches overlap in this direction, but this can be solved by a clipping as proposed in the painter's algorithms [1]). For vertex j of each triangle, the "color" is set to $\cos\theta' \cdot (\mathbf{L}^e_v(\omega'_{D-d}) + \tilde{f}_r(\vec{x}_v, \omega'_{D-d+1}, \omega'_{D-d}) \cdot \mathbf{I}_v$ and the linear interpolation hardware (Gouraud shading) is used to determine the color inside the triangle.

If the triangles are rendered into the image buffer in the reverse order determined by direction ω'_{D-d}, then the radiance illuminating the vertices of the given triangle is readily available in the current image buffer.

Thus the processing of a single direction for all patches require a sorting step and the rendering of each triangle into a temporary buffer. This can be done in $O(n \log n)$ time.

5. Simulation results

The presented algorithm has been simulated for the Cornell-box which is a usual benchmark for random-walk methods [21]. The reciprocal Phong BRDF [16] with parameters $(kd = 0.4, ks = 0.5, n = 10)$ was used to model the specular reflectivity of the surfaces. The low-discrepancy sequence has been generated by the incremental Halton generator proposed in [20]. Figure 3 contains two images of the same scene after 100 and 1000 iterations. The first image is still noisy, however, the high-frequency variations are filtered out by the linear basis functions.

6. Acknowledgements

This work has been supported by the National Scientific Research Fund (OTKA), ref.No.: F 015884 and the Austrian-Hungarian Action Fund, ref.No.: 29ö4 and 32öu9 and the Spanish-Hungarian Fund ref.No.: E/9.

7. Conclusions

This paper presented a light-tracing method which generates quasi-random walks and follows light interactions that are parallel with the given directions.

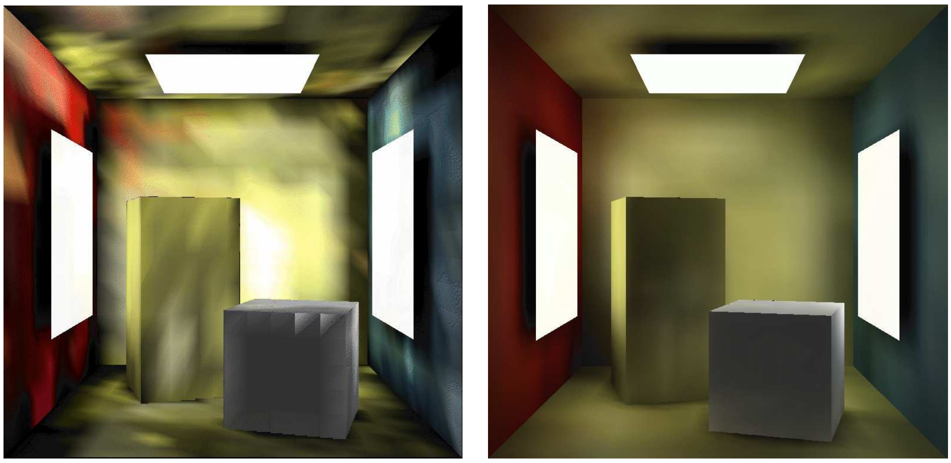

Fig. 3. An image of the Cornell Box after 100 and 1000 iterations

This method can solve the non-diffuse radiosity problem — that is the general rendering equation — without increasing the number of free variables and the required storage unlike other finite-element methods. Since in the finite-element phase the patches are assumed to have position independent radiance distribution, and in the random-walk phase the directional integrals are approximated by quasi-Monte Carlo quadrature, the method is accurate if the radiance function is quite smooth (perfect mirrors, for example, can produce artifacts).

References

1972

[1] Newell M.E., Newell R.G., Sancha T.L.: A New Approach to the Shaded Picture Problem. Proceedings of the ACM National Conference, 443.

1984

[2] Cook R.L., Porter T., Carpenter L.: Distributed Ray Tracing . Proceedings of SIGGRAPH '84, Computer Graphics, 137-145.

1985

[3] Cohen M., Greenberg D.: The Hemi-cube, a Radiosity Solution for Complex Environments. Proceedings of SIGGRAPH '85, 31-40.

1986

[4] Immel D.S., Cohen M.F., Greenberg D.P.: A Radiosity Method for Non-Diffuse Environments. Proceedings of SIGGRAPH '86, Computer Graphics, 133-142.

[5] Kajiya J. T.: The Rendering Equation. Proceedings of SIGGRAPH '86, Computer Graphics, 143-150.

1987

[6] Wallace J.R., Cohen M.F., Greenberg D.P.: A Two-Pass Solution to the Rendering Equation: A Synthesis of Ray Tracing and Radiosity Methods. Proceedings of SIGGRAPH '87, Computer Graphics, 311-324.

1989

[7] Sillion F., Puech C.: A General Two-Pass Method Integrating Specular and Diffuse Reflection. SIGGRAPH 89, 335-344.

1991

[8] Sobol I.: Die Monte-Carlo-Methode. Deutscher Verlag der Wissenschaften (in German).

[9] Shirley P.: Discrepancy as a Quality Measure for Sampling Distributions. Eurographics '91, 183-194.

1992

[10] Pattanik S.N., Mudur S.P.: Adjoint Equations and Random Walks for Illumination Computation. ACM Transactions on Graphics, 14(1), 77-102.

[11] Niederreiter H.: Random number generation and quasi-Monte Carlo methods. SIAM, Pennsilvania.

1993

[12] Lewis, R.: Making Shaders More Physically Plausible. Proc. 4th EG Workshop on Rendering, 47-62.

[13] Dutre P., Lafortune E., Willems, Y.D.: Monte Carlo Light Tracing with Direct Computation of Pixel Intensities. Compugraphics '93, 128-137.

[14] Lafortune E., Willems Y.D.: Bi-directional Path-Tracing. Compugraphics '93, 145-153.

1994

[15] Veach E., Guibas L.: Bidirectional Estimators for Light Transport. Proc. 5th Eurographics Workshop on Rendering, 147-162.

[16] Lafortune E., Willems, Y.: Using the Modified Phong Reflectance Model for Physically Based Rendering. Department of Computing Science, K.U. Leuven.

1995

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

[18] Keller A.: A quasi-Monte Carlo algorithm for the global illumination in the radiosity setting. Monte-Carlo and Quasi-Monte Carlo Methods in Scientific Computing, Springer, 239-251.

1996

[19] Sbert M.: The Use of Global Directions to Compute Radiosity. Ph.D. Thesis, Catalan Technical University, Barcelona.

[20] Keller A.: The fast Calculation of Form Factors using Low Discrepancy Sequences. Proc. Spring Conference on Computer Graphics (SCCG '96), 195-204.

1997

[21] Neumann L., Neumann A., Bekaert P.: Radiosity with Well Distributed Ray Sets. Computer Graphics Forum (Eurographics'97). 16(3), 261-270.

[22] Szirmay-Kalos, L. and Fóris, T. and Neumann, L. and Csébfalvi, B.: An Analysis to Quasi-Monte Carlo Integration Applied to the Transillumination Radiosity Method. Computer Graphics Forum (Eurographics'97), 16(3), 271-281.