Interner Bericht

Quasi-Monte Carlo Radiosity

Alexander Keller

279/96

Fachbereich Informatik

Universität Kaiserslautern · Postfach 3049 · D-67653 Kaiserslautern

Quasi-Monte Carlo Radiosity

Alexander Keller

279/96

Universität Kaiserslautern
AG Numerische Algorithmen
Postfach 30 49
67653 Kaiserslautern
Germany

März 1996

Herausgeber: AG Numerische Algorithmen
Leiter: Professor Dr. S. Heinrich

QUASI-MONTE CARLO RADIOSITY

ALEXANDER KELLER

ABSTRACT. The problem of *global illumination* in computer graphics is described by a second kind Fredholm integral equation. Due to the complexity of this equation, Monte Carlo methods provide an interesting tool for approximating solutions to this transport equation. For the case of the radiosity equation, we present the deterministic method of *quasi-random walks*. This method very efficiently uses low discrepancy sequences for integrating the Neumann series and consistently outperforms stochastic techniques. The method of quasi-random walks also is applicable to transport problems in settings other than computer graphics.

1. INTRODUCTION

The fast solution of the global illumination problem is a central problem in computer graphics. It is given by a second kind Fredholm integral equation. Using the Neumann series, the equation can be transformed into a sum of integrals. Due to the high dimension of the integrals and the discontinuities of the integrands, usual quadrature formulae will fail. So only the class of Monte Carlo methods remains as suited tool for integration. Besides the original Monte Carlo method, using (pseudo-) random numbers for sampling, there exists the method of quasi-Monte Carlo integration. This deterministic method uses low discrepancy points for sampling. These point sets are especially designed for integration, and are more uniformly distributed than random numbers. In consequence the convergence rate of quasi-Monte Carlo integration outperforms the Monte Carlo rate of $\mathcal{O}(\frac{1}{\sqrt{N}})$ (see [Nie92]).

In this paper we present the deterministic method of the *quasi-random walk* for the fast approximation of the solution of the radiance equation using low discrepancy points. Compared to stochastic algorithms, the quasi-random walk works operates at a slightly better rate than Monte Carlo algorithms. The algorithm correctly handles textures, is straightforward to implement and interactively can be used with graphics hardware like SGI-Reality Engine^2, SGI-Infinite Reality, or HP-Visualize.

In the next section we give a brief introduction to Monte Carlo and quasi-Monte Carlo integration (for a profound introduction, see [Nie92]). In section 3 we describe the global illumination problem by its integral equation and derive the method of quasi-random walk. Then we illustrate the superiority of our approach as compared to other algorithms by numerical evidence. After discussing some special features of the algorithm, we draw the conclusion, that the quasi-random walk is superior to comparable stochastic algorithms in the setting of computer graphics.

Date: March 16, 1996.

1991 *Mathematics Subject Classification.* Primary 65Y25, 65C05; Secondary 65R20.

2. Monte Carlo and Quasi-Monte Carlo Integration

The Monte Carlo method approximates an integral of a function g on the s-dimensional unit cube $I^s = [0,1)^s$ by averaging N function samples taken at random positions $x_i \in I^s$, $i = 0, \dots, N-1$:

$$\int_{I^s} g(x)\, dx \approx \frac{1}{N} \sum_{i=0}^{N-1} g(x_i) \tag{2.1}$$

The expectation of the error of this stochastic method is bounded by

$$\mathbb{E}\left| \int_{I^s} g(x)\, dx - \frac{1}{N} \sum_{i=0}^{N-1} g(x_i) \right| \leq \sigma(g) \frac{1}{\sqrt{N}}$$

where σ^2 is the variance of g. Since a computer cannot produce real random numbers, pseudo-random numbers are used instead. They are typically generated using the linear congruential method (for a survey on pseudo-random number generation, see [Nie95]). Assuming a finite upper bound on the variance, the rate of the Monte Carlo method is $\mathcal{O}(\frac{1}{\sqrt{N}})$. It can be improved by variance reduction techniques (see [Nie92], pp. 7 or [Sob91]). The technique of domain stratification is very common in computer graphics, where one instance is *jittered sampling* [CPC84]. The quality of sampling patterns can be further enhanced by guaranteeing a minimum distance property of the samples (like in Poisson-disk sampling). From that we observe that increasing the uniformity of the sampling pattern is more important than randomness for the purpose of integration:

Definition 2.1. The discrepancy $D^*(P_N)$ is a measure for the deviation of a point set $P_N = \{x_0, \dots, x_{N-1}\}$ from uniform distribution. $D^*(P_N)$ is defined to be the largest integration error for integrating the characteristic functions of all subcubes J of I^s including the origin:

$$D^*(P_N) := \sup_{J = \prod_{j=1}^{s} [0, a_j) \subset I^s} \left| \int_{I^s} \chi_J(x)\, dx - \frac{1}{N} \sum_{i=0}^{N-1} \chi_J(x_i) \right|$$

For infinitely many random points, we have (see [Nie92])

$$D^*(P_N^{\text{random}}) \in \mathcal{O}\left(\sqrt{\frac{\log\log N}{N}} \right) \text{ almost surely.} \tag{2.2}$$

The best known theoretical lower bound (see [Nie92], pp. 32) for the discrepancy is

$$D^*(P_N) \geq B_s \frac{\log^{\frac{s-1}{2}} N}{N} \tag{2.3}$$

In fact there exist deterministic point sets which acquire a discrepancy of $\mathcal{O}(\frac{\log^s N}{N})$ and fall into the gap between (2.2) and (2.3). Deterministic sample points with a discrepancy of $\mathcal{O}(\frac{\log^s N}{N})$ belong to the class of *low discrepancy points*. For a short survey on generators for such points, see [War95], a profound introduction can be found in [Nie92]. For the sake of simplicity and the needs of the algorithm introduced in the next section, we restrict ourselves to the introduction of the simple, infinite Halton sequence. It is based on radical inversion, which transfers

the natural number i into base-b-representation and mirrors that representation at the decimal point, resulting in the radical inverse function

$$\Phi_b(i) := \sum_{j=0}^{\infty} a_j(i)\, b^{-j-1} \Leftrightarrow i = \sum_{j=0}^{\infty} a_j(i)\, b^j$$

The s-dimensional Halton sequence is built by

$$x_i = (\Phi_{b_1}(i), \ldots, \Phi_{b_s}(i))$$

where the base b_j, $1 \leq j \leq s$, mostly is chosen to be the j-th prime number. For this sequence the discrepancy of the first N points is bounded by (see [Nie92])

$$D^*(P_N^{\text{Halton}}) < \frac{s}{N} + \frac{1}{N} \prod_{j=1}^{s} \left(\frac{b_j - 1}{2 \log b_j} \log N + \frac{b_j + 1}{2} \right) \tag{2.4}$$

The sequence can be calculated very fast using ACM algorithm 247 [HW64]. Since the minimal distance of any two points $x_i \in P_N$ and $x_j \in P_N$ of the Halton sequence is bounded by

$$\inf_{x_i \neq x_j} \|x_i - x_j\|_\infty \geq \frac{1}{N} \max_{1 \leq j \leq s} \frac{1}{b_j}, \tag{2.5}$$

the efforts of jittered sampling and multi-jittered sampling are met, guaranteeing both implicit domain stratification, and minimum distance property. Besides the Halton sequence, there exist other low discrepancy sequences, which have a smaller constant preceeding the order, but often are not suited to our algorithm due to their internal structure. The order of the discrepancy can be reduced by one logarithm by restricting to finite low discrepancy point sets (for example the Hammersley point set), but the algorithm of the next section needs an infinite sequence. Obviously low discrepancy points are deterministic and not random at all, i.e. they fail most of the statistical tests, which (pseudo-)random numbers are supposed to pass. Using the discrepancy of a point set, the error of integration can be bounded by the Koksma-Hlawka inequality:

$$\left| \int_{I^s} g(x)\, dx - \frac{1}{N} \sum_{i=0}^{N-1} g(x_i) \right| \leq V(g)\, D^*(P_N) \tag{2.6}$$

Here $V(g)$ is the variation of g in the sense of Hardy and Krause (see [Nie92]). Roughly speaking this inequality promises a quadratically faster convergence when using low discrepancy samples instead of random samples. But in the setting of computer graphics the function g is of unbounded variation and so we cannot apply this inequality. Restricting g to a convex set $C \subset I^s$ results in (see [Nie92])

$$\left| \int_C g(x)\, dx - \frac{1}{N} \sum_{i=0}^{N-1} g|_C(x_i) \right| \leq (V(g) + g(1, \ldots, 1))\, 8sD^*(P_N)^{\frac{1}{s}}.$$

The order of this upper bound is too rough by far. In [PTVF92] and [MC95] we find arguments for faster convergence rates, which beat random sampling even for functions of unbounded variation, especially characteristic functions. As the numerical experiments will show, these rates are acquired using low discrepancy sampling.

2.1. **Discrete Density Approximation.** We now transfer the principle of importance sampling to quasi-Monte Carlo integration (see also [SM94]). Often a density distribution p can be split off the integrand $g = f\ p$, where the density p is defined on the unit cube I^s with $p(x) \geq 0$ for $x \in I^s$ and $\int_{I^s} p(x)\ dx < \infty$. Assuming the distribution function $\mu(y) = \int_0^y p(x)dx$, $y \in I^s$ to be strictly monotone, an inverse $\mu^{-1} : I^s \to I^s$ can be defined by the multidimensional inversion method (see [HM72] and [Wic74]). For the sequel we also presume μ^{-1} to be of bounded variation in the sense of Hardy and Krause. We now approximate the measure μ by a discrete point cloud $C_N = \{y_0, \ldots, y_{N-1}\}$ modeled out of the point set P_N by $y_i = \mu^{-1}(x_i)$. Using (2.6), the discrepancy between C_N and p is bounded by

$$(2.7) \quad \begin{aligned} D^*(p, C_N) &:= \sup_{J=\prod_{j=1}^s [0,a_j) \subset I^s} \left| \int_{I^s} \chi_J(x)\ p(x)\ dx - \frac{1}{N} \sum_{i=0}^{N-1} \chi_J(y_i) \right| \\ &\leq V(\mu^{-1})\ D^*(P_N)\ . \end{aligned}$$

Note: If p is separable, i.e. $p(x) = \prod_{j=1}^s p^{(j)}(x^{(j)})$, we even have $D^*(p, C_N) = D^*(P_N)$. These bounds for $D^*(p, C_N)$ are results of [Wic74]. We now approximate the continuous measure μ by the discrete measure $\mu_N := \frac{1}{N} \sum_{i=0}^{N-1} \delta_{y_i}$:

$$(2.8) \quad \int_{I^s} f(x)\ p(x)\ dx = \int_{I^s} f(y)\ \ d\mu(y) \approx \int_{I^s} f(y)\ d\mu_N(y) =: \frac{1}{N} \sum_{i=0}^{N-1} f(y_i)$$

Thus the use of different point sets results in different quadrature formulae, which can be distinguished by bounds on the discrepancy $D^*(p, C_N)$. Since we assumed the variation $V(\mu^{-1})$ to be finite, the asymptodic behaviour of the approximation is determined by the discrepancy $D^*(P_N)$.

2.2. **Constructing Low Discrepancy Quadrature Formulae.** The design of low discrepancy quadrature formulae can be split into three steps:

1. Transformation of the integral onto unit cube,
2. possible inversion of densities, and
3. selection of suited low discrepancy sample points.

The first step allows us to use (2.1) for approximating the integral by low discrepancy points. The second step avoids expensive samples to be weighted by small values of a density distribution[1]. These two transformations are not unique, since there exist many different mappings of the unit cube onto itself, and therefore the discrepancy can be changed by (2.7). So constructing μ^{-1} minimizing $V(\mu^{-1})$ is of interest in further research (for example the mapping of the unit square onto a triangle, see [SWZ96]). When choosing the low discrepancy points in the last step, the major decision is whether to choose a finite point set or a infinite point sequence. The finite point sets have a slightly improved order of discrepancy (for instance the Hammersley point set acquires $\mathcal{O}(\frac{\log^{s-1} N}{N})$), whereas the infinite sequences can be used for adaptive sampling. Other sequences differ in the constant of the order. The application of low discrepancy points incorporates the random sampling principles of domain stratification and minimum distance property (2.5) directly into a deterministic quadrature, which in consequence has no variance and a slightly improved performance as compared to the Monte Carlo rate!

[1] For smooth functions there even exist theorems for quasi-importance sampling, see [SM94]

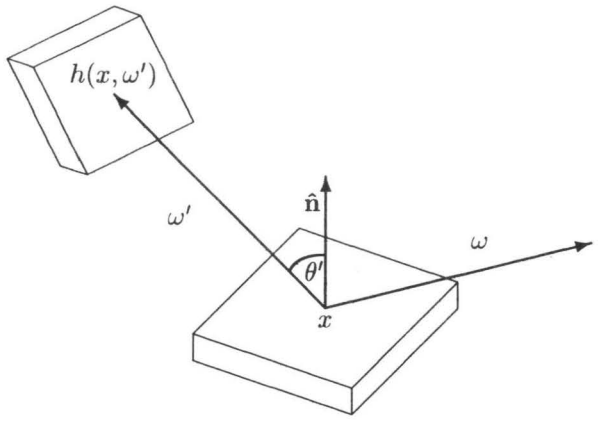

FIGURE 1. The geometry for the radiance equation.

3. SOLVING THE RADIANCE EQUATION

The radiance equation describes the radiance distribution in the scene. It is given by a second kind Fredholm integral equation. The radiance L, which leaves from a point $x \in S$ on the surface of the scene in direction $\omega \in \Omega$, where Ω is the hemisphere in point x, is the sum of the source radiance L_0 and all reflected radiance (see figure 1 for the geometry):

$$\begin{aligned} L(x,\omega) &= L_0(x,\omega) + \int_\Omega f_r(-\omega', x, \omega)\, L(h(x,\omega'), -\omega')\, \cos\theta'\, d\omega' \\ &:= L_0 + T_{f_r} L \end{aligned} \tag{3.1}$$

Here $h(x,\omega') \in S$ is the first point that is hit when shooting a ray from x into direction ω'. This function accounts for visibility in the three-dimensional environment. Its calculation is the most expensive in the whole illumination process and is bounded from below by $\mathcal{O}(\log K')$, where K' is the number of elements in the scene, which have to be tested for intersection. Minimizing the number of calls to h promises the fastest algorithms for the solution of (3.1). The reflected radiance is the integral of the radiance of all points which can be seen through the hemisphere Ω in point x attenuated by the BRDF (bidirectional reflectance distribution function) f_r and the projection term $\cos\theta'$, which puts the surface perpendicular to the ray (x,ω'). θ' is the azimuth angle between the surface normal in x and the direction ω'. Due to physical reasons we have $\|T_f\| < 1$, because any real scene is reflecting less than 100% of the radiance. Another important property used for the solution of the radiance equation is the Helmholtz principle:

$$f_r(-\omega', x, \omega) = f_r(-\omega, x, \omega')$$

This property allows to reverse all light paths. For the radiosity setting, we only consider diffuse reflections. Then L and f_r become independent of direction, and the BRDF $f_r(x)$ can be taken out of the integral:

$$L(x) = L_0(x) + f_r(x) \int_\Omega L(h(x,\omega'))\, \cos\theta'\, d\omega' \tag{3.2}$$

L, L_0 and f_r are spectral distributions. In computer graphics equation (3.1) is usually replicated for the three selected wavelengths of red, green and blue. This simple approximation of the spectrum is sufficient for our purposes.

Now we want to calculate functionals of the solution L. That is, we project the solution onto a set of basis vectors $(\Psi_k)_{k=1}^K$ by $\Phi_k = \langle L, \Psi_k \rangle$. For the sake of simplicity we choose the usual characteristic functions for every surface element A_k, where $\cup_{k=1}^K A_k = S$ and the A_k are disjoint triangles. Hence $\Psi_k(y,\omega) = \chi_{A_k}(h(y,\omega)) \cos\theta$ detects the incoming power Φ_k for the element A_k. Using the Neumann series and the principle of construction from the previous section, we derive the algorithm:

$$
\begin{aligned}
\Phi_k = \langle L, \Psi_k \rangle &= \langle \sum_{j=0}^{\infty} T_{f_r}^j \, L_0, \Psi_k \rangle = \sum_{j=0}^{\infty} \langle T_{f_r}^j \, L_0, \Psi_k \rangle \\
&= \sum_{j=0}^{\infty} \int_S \int_\Omega (T_{f_r}^j \, L_0)(y,\omega') \, \chi_{A_k}(h(y,\omega')) \, \cos\theta' \; d\omega' \, dy \\
&= \sum_{j=0}^{\infty} \int_{\Omega^{j+1}} \int_{S_0} L_0(x_0) \, \chi_{A_k}(h(x_{j+1},\omega_{j+1})) \\
&\qquad \prod_{l=1}^{j} f_r(x_l) \prod_{l=1}^{j+1} \cos\theta_l \; dx_0 \, d\omega_1 \cdots d\omega_{j+1} \\
&= \sum_{j=0}^{\infty} \int_{Q^{j+1}} \int_{S_0} L_0(x_0) \, \chi_{A_k}(h(x_{j+1},\omega_{j+1})) \\
&\qquad \prod_{l=1}^{j} f_r(x_l) \prod_{l=1}^{j+1} \frac{\sin 2\theta_l}{2} \; dx_0 \, d\theta_1 \, d\phi_1 \cdots d\theta_{j+1} \, d\phi_{j+1} \\
&= \sum_{j=0}^{\infty} |S_0| \, \pi^{2j+2} \int_{I^{2j+4}} L_0(x_0(u_0,u_1)) \, \chi_{A_k}(h(x_{j+1},\omega_{j+1})) \\
&\qquad \prod_{l=1}^{j} f_r(x_l) \prod_{l=1}^{j+1} \frac{\sin \pi u_{2l}}{2} \; du_0 \cdots du_{2j+3} \\
&= \sum_{j=0}^{\infty} |S_0| \, \pi^{j+1} \int_{I^{2j+4}} L_0(x_0(v_0,v_1)) \, \chi_{A_k}(h(x_{j+1},\omega_{j+1})) \\
&\qquad \prod_{l=1}^{j} f_r(x_l) \; dv_0 \, dv_1 \, dF(v_2,v_3) \cdots dF(v_{2j+2},v_{2j+3})
\end{aligned}
\tag{3.3}
$$

where

$$
\begin{aligned}
x_l &= h(x_{l-1},\omega_l) \text{ for } l > 0 \,, \\
Q &= [0,\frac{\pi}{2}] \times [0,2\pi] \text{ and} \\
\omega_l &= (\theta_l,\phi_l) = (\frac{\pi}{2} u_{2l}, 2\pi u_{2l+1}) \text{ in } x_{l-1} \,.
\end{aligned}
$$

S_0 is the surface of the lightsources where $L_0 > 0$. By (v_0,v_1) we access the point x_0 on S_0, using an area preserving mapping (see [SWZ96]). The inversion method applied in the last transformation step prevents the integrand to be weighted by small

values of the projection term in the final simulation. The directions $(\frac{\pi}{2}v_{2l}, 2\pi v_{2l+1})$ are modeled with respect to the projection term:

$$dF(v_{2l}, v_{2l+1}) := d\sin^2\frac{\pi}{2}v_{2l}\, dv_{2l+1}$$

For the simulation we assume the source radiance to be given as diffuse emitters with scalar emittance ϵ_k for each surface element as $L_0(x) = \sum_{k=1}^{K} \chi_{A_k}(x)\epsilon_k\tau_k^e(x)$, where τ_k^e is the emission texture. The diffuse BRDF is of the form $f_r(x) = \sum_{k=1}^{K} \chi_{A_k}(x)\frac{\rho_k^d}{\pi}\tau_k(x)$. Here ρ_k^d is the reflectivity and τ_k the texture of surface element A_k. Then $\Phi(x)$ is interpolated out of the Φ_k by applying interpolating display hardware (for details see [CW93]) and the approximation

$$L(x) \approx \overline{L}(x) = L_0(x) + \Phi(x)\sum_{k=1}^{K} \chi_{A_k}(x)\frac{\rho_k^d\tau_k(x)}{\pi|A_k|} \tag{3.4}$$

is displayed using modulating textures.

3.1. **The Quasi-Random Walk Algorithm for the Radiosity Equation.** Previous work (see [Kel95a] and [Kel95b]) showed, that solving the radiance equation 3.1 by fixed length quasi-random walks is superior to using fixed length random walks. Related work done in [Spa95] for similar integral equations also revealed the superiority of low discrepancy sampling. But the algorithms in [Kel95a] and [Kel95b] are biased, since the truncated Neumann series results in an underestimation of the true solution. Experiments show, that using the same number of rays for a random walk with russian roulette for absorption, is superior to the fixed length algorithms, because they acquire the unbiased expectation with a higher accuracy. Obviously the latter random walk performs better, since it uses more paths for low powers of $T_f^i L_0$ than for the higher powers, which are less important due to $\|T_f\| < 1$. [AK90] and [KMS94] mention, that russian roulette, i.e. *discrete absorption*, has a considerably higher variance than using paths of fixed length, i.e. *fractional absorption*. This leads us to the following algorithm:

For the moment let us assume, that we are in a very restricted radiosity setting, where all surfaces have the constant reflectivity ρ, i.e. $f_r = \frac{\rho}{\pi}$. So if we would perform a random walk using discrete absorption with N paths, due to $\|T_f\| = \rho$ only ρN paths are expected to have a path length longer than 1, $\rho^2 N$ paths would be longer than 2, and so on. Then a first quasi-Monte Carlo algorithm using fractional absorption, would be to integrate $\langle T_f^j L_0, \Psi_k\rangle$ using $N_j := \lfloor \rho^j N \rfloor$ low discrepancy samples, where $j = 0, \ldots, l$ and $l := \lfloor \log_\rho \frac{1}{N} \rfloor$. Since the ray shooting is a very expensive operation, the procedure can be speeded up by effectively reusing rays, which are already shot: The first N_1 paths used for calculating $\langle L_0, \Psi_k\rangle$, are continued to calculate $\langle T_f L_0, \Psi\rangle$, then N_2 of these paths are continued for $\langle T_f^2 L_0, \Psi_k\rangle$, and so on (see the scheme in figure 2). Since ρ usually is not constant, we calculate the average reflectivity of the scene by

$$\bar{\rho} := \frac{\sum_{k=1}^{K} |A_k|\rho_k^d}{\sum_{k=1}^{K} |A_k|}$$

which is used in our algorithm instead of ρ. Then the total number N of rays shot in the algorithm is bounded by

$$N = \sum_{j=0}^{l} N_j \leq \sum_{j=0}^{l} \bar{\rho}^j N = \frac{1-\bar{\rho}^l}{1-\bar{\rho}} N < \frac{1}{1-\bar{\rho}} N = \bar{\lambda} N$$

where $\bar{\lambda} := \frac{1}{1-\bar{\rho}}$ is the mean path length. Note, that for $N \to \infty$ the algorithm converges to the true solution without a truncation error since all summands of the Neumann series become included. The pseudo-code for the algorithm is:

```
int j = 0;                  // number of reflections
double Start = N;           // total number of paths
int End;

loop
    End = (int) Start;
    Start *= ρ;

    if((int) Start >= End)
        break;

    for(int i = (int) Start; i < End; i++)
        TracePath(i, j);

    j++;
endloop
```

For the routine `TracePath` the i-th $(2j+4)$-dimensional low discrepancy vector v_i is used for all decisions of the random walk, where j is the number of reflections. $P_0 = \sum_{k=1}^{K} A_k * \epsilon_k$ is the total power of the lightsources. Similar to [NNP+95], the cumulative distribution function $P(j) = \frac{\sum_{k=1}^{j} A_k * \epsilon_k}{P_0}$ is used for to start a particle with power $\frac{P_0}{N}$ in point x_0, modeled by $(v_{i,0}, v_{i,1})$ according to $P(j)$, and traced into direction ω_1, modeled by $(v_{i,2}, v_{i,3})$. In $x_1 = h(x_0, \omega_1)$ its data (direction, position, radiance) is recorded. Then the particle is scattered and attenuated by using $(v_{i,4}, v_{i,5})$. This walk is continued for j reflections.

3.2. **Extensions for the Radiance Equation.** The simulation of more general BRDFs like in [War92] is done by randomization: When a surface is hit, a random decision according to the diffuse ρ_d and specular ρ_s reflectivities is made (see [CRMT91]). In the specular case the ray is randomly scattered using the specular distribution. Otherwise we proceed by diffusely scattering using the low discrepancy sequence. Obviously this procedure meets the expectation and specular surfaces are simply used to lengthen the deterministic paths by random segments. Since only the diffuse radiance is calculated by (3.4), in the general case the specular effects cannot be displayed directly, but must be added by for example ray tracing or using multi-pass techniques with double-buffer hardware.

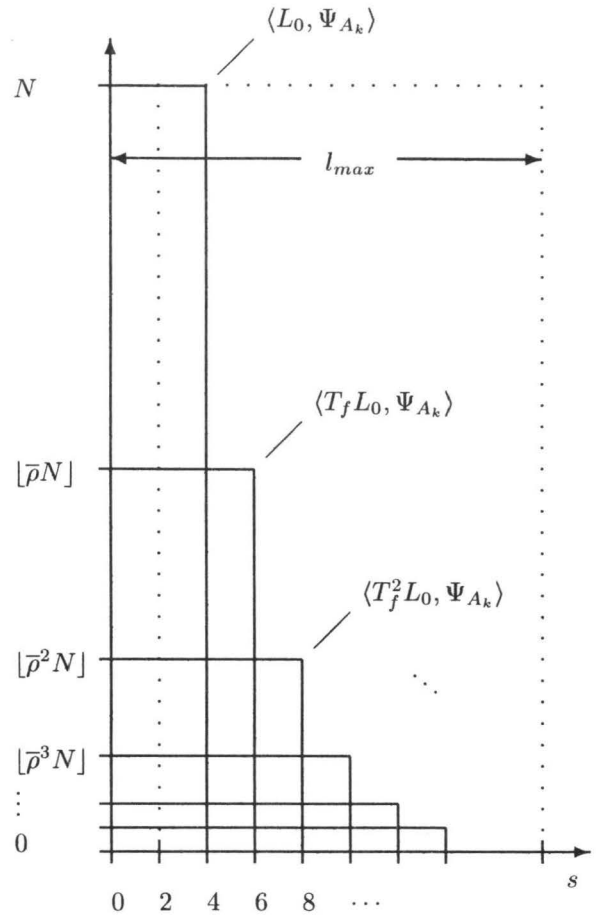

FIGURE 2. Usage of low discrepancy points in the quasi-random walk algorithm

4. Numerical Evidence

Since the available theoretical bounds are far too pessimistic, we analyze the algorithms by numerical experiments. In the experiments we compare the quasi-random walk to a random walk algorithm. For the true solution L_k and the approximation $\tilde{L}_k$ we compare the mean radiance $\overline{L}$ and the weighted mean radiance $\overline{L}^w$, the $\|\cdot\|_2$-, weighted $\|\cdot\|_2^w$-, $\|\cdot\|_\infty$, and weighted $\|\cdot\|_\infty^w$- distance:

not weighted	weighted
$\overline{L} = \frac{\sum_{k=1}^K \tilde{L}_k}{N}$	$\overline{L}^w = \frac{\sum_{k=1}^K A_k \tilde{L}_k}{\sum_{k=1}^K A_k}$
$\|L, \tilde{L}\|_2 = \sqrt{\sum_{k=1}^K (\tilde{L}_k - L_k)^2}$	$\|L, \tilde{L}\|_2^w = \sqrt{K \frac{\sum_{k=1}^K A_k (\tilde{L}_k - L_k)^2}{\sum_{k=1}^K A_k}}$
$\|L, \tilde{L}\|_\infty = \max_{k=1}^K \|\tilde{L}_k - L_k\|$	$\|L, \tilde{L}\|_\infty^w = K \frac{\max_{k=1}^K A_k \|\tilde{L}_k - L_k\|}{\sum_{k=1}^K A_k}$

For the random experiments we show the range over 20 experiments with different random seeds in order to give an impression of the variance. The experiments have been carried out with other generators, too. For other random number generators we observed the same behaviour as for the internal `drand48()` of HP-UX used in the printed graphs, whereas the Halton sequence seems to be best choice as compared to other low discrepancy sequences applied to this problem. Another argument for the Halton sequence is that it generates low discrepancy points faster than other generators. All calculations were done on an HP9000/735 at 125 Mhz in double precision. For the measurements we used the geometry of a Cornell-box. Special care was taken to provide a correct BREP (boundary representation) mesh, i.e. wholes were cut out when one object contacts another.

4.1. **Experiment 1.** For the first measurement we simplify the integral equation (3.2), so that an analytical solution exists for use as benchmark. Let $L_0 = \frac{1}{4}$ and $f_r = \frac{\rho_d}{\pi} = \frac{1}{2\pi}$ then we have

$$\begin{aligned} L &= L_0 + \frac{\rho_d}{\pi} \int_\Omega L \cos\theta(\omega)\, d\omega \\ &= L_0 + T_{f_r} L = \sum_{i=0}^{\infty} T_{f_r}^i L_0 \\ &= \sum_{i=0}^{\infty} \left(\frac{1}{2\pi}\right)^i \frac{1}{4} T^i = \frac{1}{4} \sum_{i=0}^{\infty} \frac{1}{2^i} = \frac{1}{2} \end{aligned}$$

where

$$T = \int_\Omega \cos\theta\, d\omega = \int_0^{2\pi} \int_0^{\frac{\pi}{2}} \cos\theta(\omega)\, \sin\theta\, d\theta\, d\phi = \pi$$

simply is the projection of the unit-hemisphere onto the plane. Then the solution is independent of the scene geometry ! As can be seen in the graphs in figure 3, both algorithms converge to the same solution. Considering the weighted mean $\overline{L}$, the quasi-random walk approximates the solution more smoothly, than the random algorithm. Looking at the unweighted norms, the quasi-random walk always ranges inside the variance of the random method. In the case of the weighted norms, it even performs better, meaning, that the solution is more exact on the larger surface elements, than on the smaller ones.

4.2. **Experiment 2.** We now use the Cornell box with colored faces and with only on light source attached to the ceiling of the box (see figure 4). Since an analytical solution is not yet kown, we calculated a master solution L for the deterministic and stochastic algorithm at $N = 10^7$ paths. The two solutions lie very close together, and we compared the runs for smaller N of both algorithms to their own master. Having now only a small area of the scene as lightsource, the graphs in figure 4 very clearly show the superiority of the quasi-random walk using the Halton sequence. The algorithm approximates the solution more smoothly. Considering the norms, it deterministically outperforms the random experiments (the weighted norms are not displayed, since all triangles were about the same size).

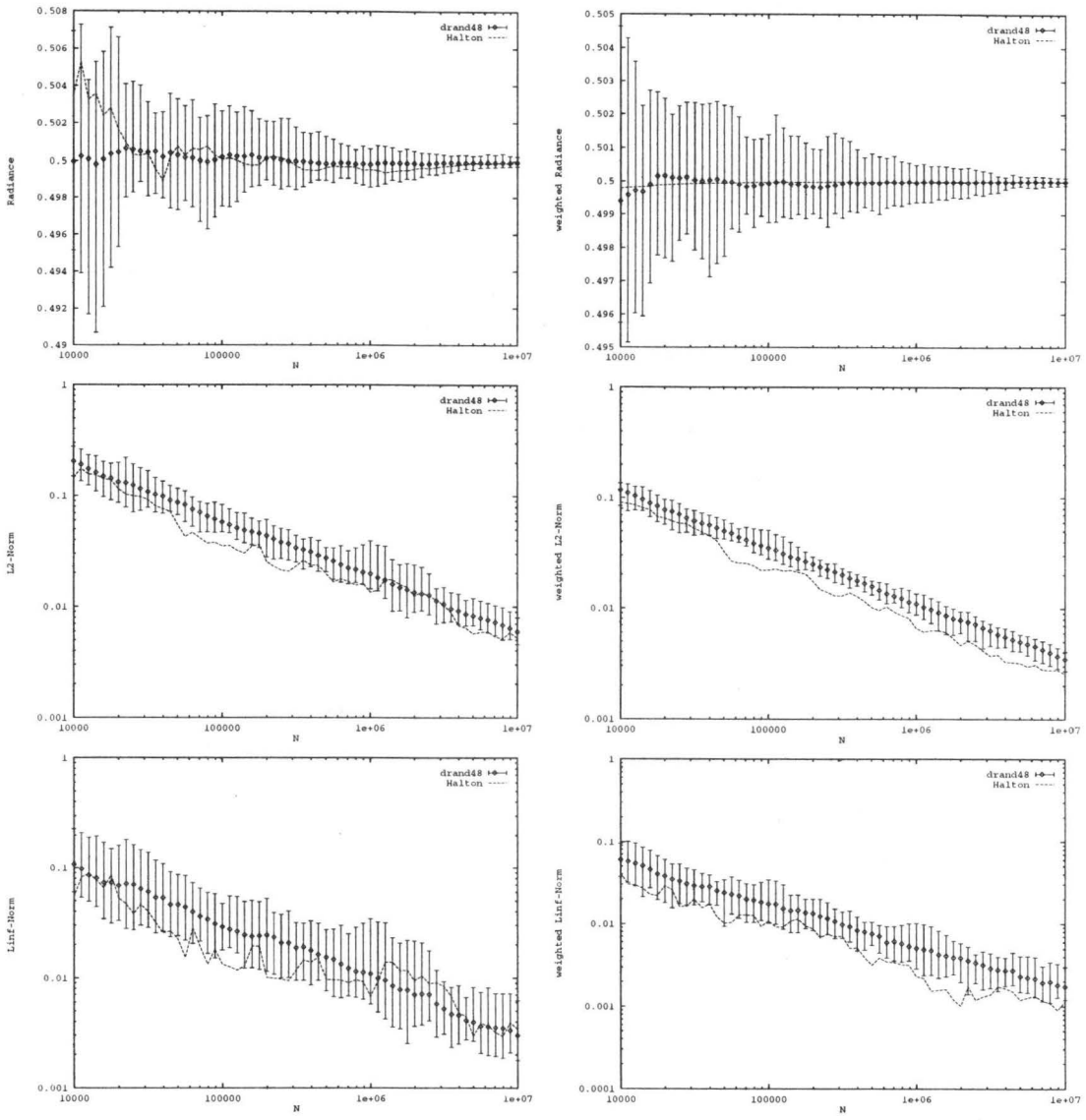

FIGURE 3. Measurements for the analytic Cornell-box

5. Discussion and Applications

The quasi-random walk algorithm produces a deterministic cloud of particles, which are used for calculating functionals of the solution of (3.2). The experiments show, that this new technique exposes a slightly better convergence rate than comparable stochastic methods but without variance. It such is superior to random sampling. Due to the properties of low discrepancy sequences, the method can be seen as one special instance of jittered sampling with importance sampling incorporated in it, except that N can be chosen freely.

Note, that this method does not belong to the class of Galerkin methods like progressive refinement (see [GCS94]) or similar methods [NNP+95], because the

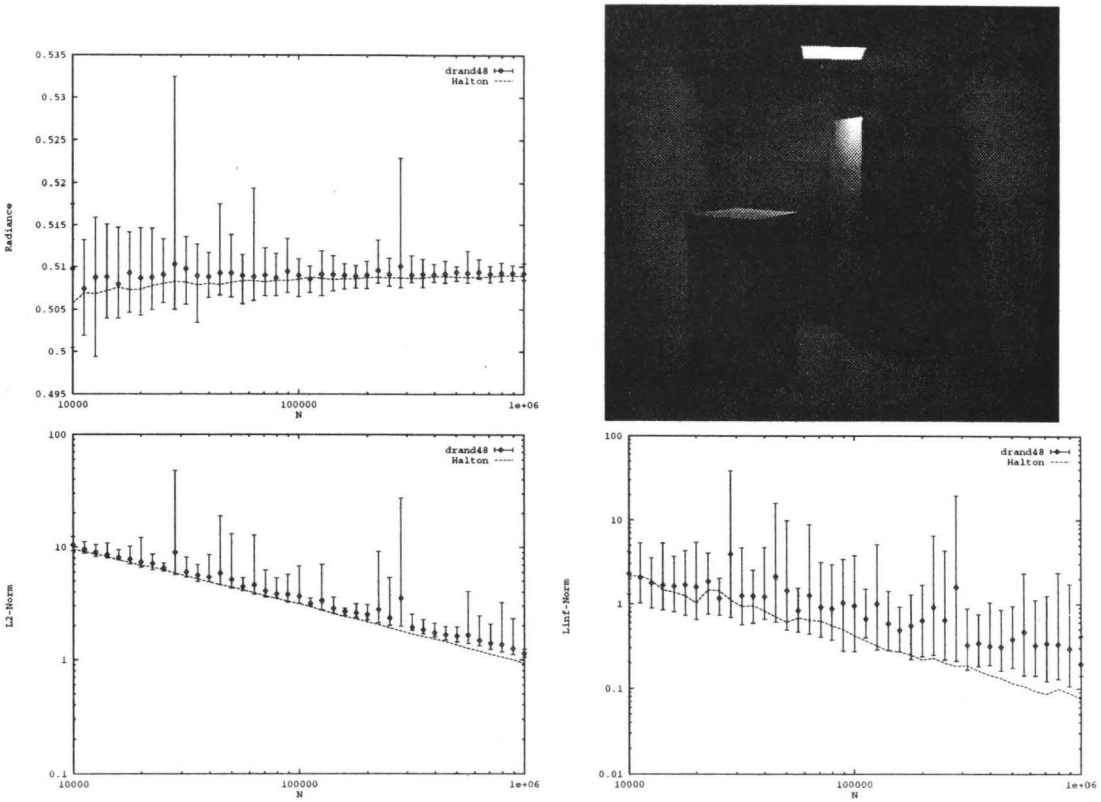

FIGURE 4. Measurements for the colored Cornell-box

quasi-random walk uses the original scene including textures (no tesselation or polygonization is required) as it is modeled. Only the functionals calculated finally introduce a discretization error depending on the basis functions used. In consequence we can expect the results to be more accurate than in the methods mentioned before.

Generating the vectors of the Halton sequence using [HW64] and accelerating the ray shooting by a space order (BSP or Octree) makes the algorithm work very fast. Each ray carries about the same amount of power, and so each ray is equally important. No form factors have to be calculated and no ray is used for binary visibility tests. Compared to [NNP+95], we only need to invert the light density of the sources once starting a path and this distribution in consequence can be be calculated in advance. The algorithm can also be used for progressive illumination: Simulating the paths from $N-1$ down to 0 and taking the intermediate result at a path number $0 \leq N' < N$, multiplying everthing by $\frac{N}{N-N'}$ allows to preview the approximation of the illumination. This temporal behaviour of the algorithm can be seen in figure 5. The algorithm is straightforward to implement and very easy to use, since the only parameter (besides the choice of a low discrepancy sequence, if not the Halton sequence) is the number N of paths to be traced.

The particle cloud generated by the algorithm can also be stored and used like the photon map (see [JC95] and [Jen95]) or as a preprocessing step for photorealistic rendering. The N particles emerging from the lightsources can be used for the

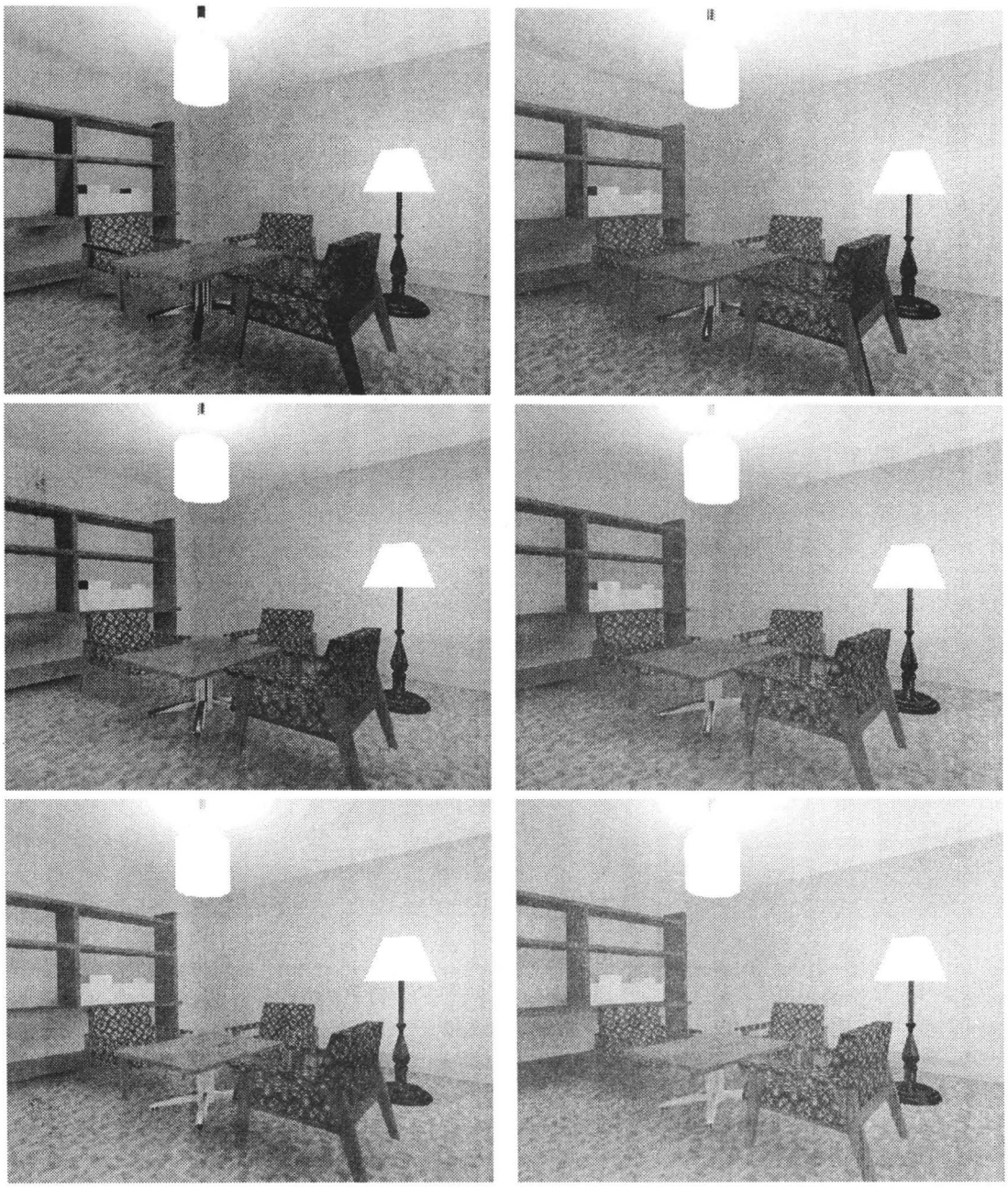

FIGURE 5. Temporal behaviour of the quasi-random walk algorithm, from left to right and top down for 4.0e4, 8.0e4, 1.2e5, 1.6e5, and 2.0e5 of total 2.4e5 paths. The final simulation took about 70 seconds on an HP9000/735 125MHz (12941 triangles)

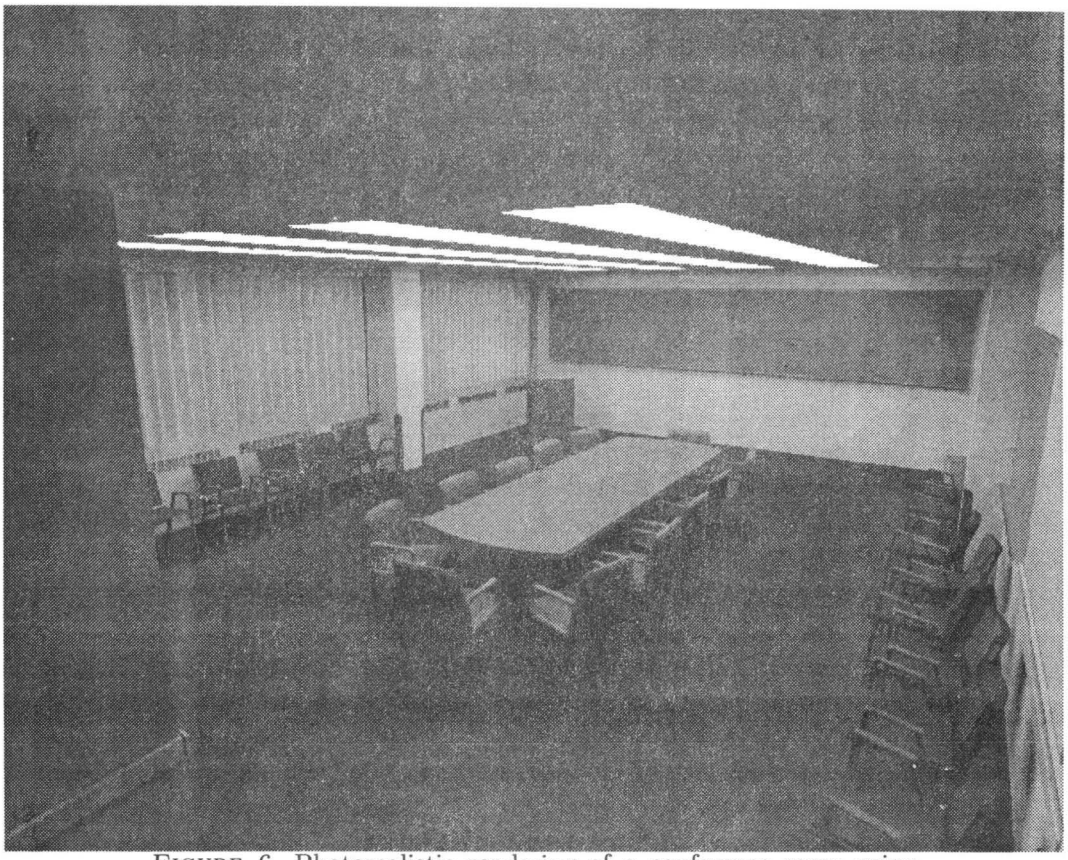

FIGURE 6. Photorealistic rendering of a conference room using 16384 paths in the quasi-random walk preprocess of about 90 seconds

fast generation of an importance function for calculating the direct illumination [SWZ96]. A picture[2] generated with the algorithm in [Kel95b] using the previous ideas can be seen in figure 6. Additionally storing a flag for the last particle in a path and the path number enables the incremental increasing of the number of paths, since the algorithm is deterministic and the paths can be continued by decisions taken from the low discrepancy vector.

6. CONCLUSION

We introduced a new deterministic method for approximating the solution of the radiosity equation based on low discrepancy sequences. Due to their better uniformity properties, low discrepancy sequences are superior to random sampling, when numerically calculating high-dimensional integrals with high sample numbers. This superiority is illustrated by numerical evidence. The convergence speed and simple

[2]The model of the conference room in figure 6 has been modeled by Anat Grynberg and Greg Ward and is available via `http://radsite.lbl.gov/mgf/HOME.html` in the *material and geometry format* MGF.

structure of the quasi-random walk make the new algorithm well-suited for interactive applications, like VRML-viewers, or for fast preprocessing for photorealistic renderers.

Quasi-Monte Carlo integration is also superior in pixel oversampling [HK94], the final gathering pass in radiosity [Kel95b] and the calculation of form factors [Kel96]. In these applications using low discrepancy points sets results in faster algorithms (up to factor $1.5 \cdots 10$), because the same accuracy of integration is aquired with less samples than random sampling.

7. Acknowledgements

The author would like to thank Stefan Heinrich for the valuable discussions and Matthias Schirm for the numerical tables.

References

[AK90] J. Arvo and D. Kirk, *Particle Transport and Image Synthesis*, Computer Graphics, August 1990, pp. 63 – 66.

[CPC84] R. Cook, T. Porter, and L. Carpenter, *Distributed ray tracing*, Computer Graphics **18** (1984), no. 3, 137–145.

[CRMT91] S. E. Chen, H. E. Rushmeier, G. Miller, and D. Turner, *A progressive Multi-Pass Method for Global Illumination*, Computer Graphics, July 1991, pp. 165 – 174.

[CW93] M. Cohen and J. Wallace, *Radiosity and Realistic Image Synthesis*, Academic Press Professional, Cambridge, 1993.

[GCS94] S. Gortler, M. F. Cohen, and P. Slusallek, *Radiosity and relaxation methods*, IEEE Computer Graphics and Applications **6** (1994), 48 – 58.

[HK94] S. Heinrich and A. Keller, *Quasi-Monte Carlo methods in computer graphics, Part I: The QMC-Buffer*, 242/94, University of Kaiserslautern, 1994.

[HM72] E. Hlawka and R. Mück, *Über eine Transformation von gleichverteilten Folgen II*, Computing (1972), no. 9, 127–138.

[HW64] J. Halton and G. Weller, *Algorithm 247: Radical-inverse quasi-random point sequence*, Comm. ACM (1964), no. 12, 701–702.

[JC95] H. Jensen and N. Christensen, *Photon maps in birdirectional monte carlo ray tracing of complex objects*, Computer Graphics **19** (1995), no. 2, 215–224.

[Jen95] H. Jensen, *Importance driven path tracing using the photon map*, Rendering Techniques '95, 1995, pp. 326–335.

[Kel95a] A. Keller, *A quasi-Monte Carlo algorithm for the global illumination problem in the radiosity setting*, Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing (H. Niederreiter and P. Shiue, eds.), Springer, 1995, pp. 239–251.

[Kel95b] A. Keller, *Quasi-Monte Carlo Methods in Computer Graphics: The Global Illumination Problem*, Proc. of the SIAM Conference in Park City (1995), to appear.

[Kel96] A. Keller, *The fast calculation of form factors using low discrepancy sequences*, 278/96, University of Kaiserslautern, 1996.

[KMS94] A. Kersch, W. Morokoff, and A. Schuster, *Radiative heat transfer with quasi-Monte Carlo methods*, Transport Theory and Statistical Physics **7** (1994), no. 23, 1001–1021.

[MC95] W. Morokoff and R. Caflish, *Quasi-Monte Carlo integration*, J. of Computational Physics (1995), no. 122, 218–230.

[Nie92] H. Niederreiter, *Random number generation and quasi-Monte Carlo methods*, SIAM, Pennsylvania, 1992.

[Nie95] H. Niederreiter, *New developments in uniform pseudorandom number and vector generation*, Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing (H. Niederreiter and P. Shiue, eds.), Springer, 1995, pp. 87–120.

[NNP+95] L. Neumann, A. Neumann, W. Purgathofer, P. Eliás, X. Pueyo, R. Tobler, and M. Feda, *The stochastic ray method for radiosity*, 6th Eurographics Workshop on Rendering (Dublin, Ireland), June 1995, pp. 206–218.

[PTVF92] H. Press, S. Teukolsky, T. Vetterling, and B. Flannery, *Numerical Recipes in C*, Cambridge University Press, 1992.

[SM94] J. Spanier and E. Maize, *Quasi-random methods for estimating integrals using relatively small samples*, SIAM Review **36** (1994), no. 1, 18–44.

[Sob91] I. Sobol, *Die Monte-Carlo-Methode*, Deutscher Verlag der Wissenschaften, 1991.

[Spa95] J. Spanier, *Quasi-Monte Carlo methods for particle transport problems*, Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing (H. Niederreiter and P. Shiue, eds.), Springer, 1995, pp. 121–148.

[SWZ96] P. Shirley, C. Wang, and K. Zimmermann, *Monte Carlo Techniques for Direct Lighting Calculations*, To appear in ACM Transactions on Graphics, 1996.

[War92] G. J. Ward, *Measuring and Modeling Anisotropic Reflection*, Computer Graphics, July 1992, pp. 265 – 272.

[War95] T. Warnock, *Computational investigations of low discrepancy point sets II*, Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing (H. Niederreiter and P. Shiue, eds.), Springer, 1995, pp. 354–361.

[Wic74] J. Wick, *Zur Anwendung der Approximation durch endliche Punktmengen auf die Lösung von Integro-Differentialgleichungen*, Jül-1124-MA, Zentralinstitut für Angewandte Mathematik, Kernforschungsanlage Jülich, Oktober 1974.

Department of Computer Science, Kaiserslautern University, Postfach 3049, D-67653 Kaiserslautern, Germany

E-mail address: keller@informatik.uni-kl.de, http://www.uni-kl.de/AG-Heinrich