# THE FAST CALCULATION OF FORM FACTORS USING LOW DISCREPANCY SEQUENCES

ALEXANDER KELLER

DEPARTMENT OF COMPUTER SCIENCE, KAISERSLAUTERN UNIVERSITY,
POSTFACH 3049, D-67653 KAISERSLAUTERN, GERMANY

Abstract. The calculation of form factors is an important problem in computing the global illumination in the radiosity setting. Closed form solutions often are only available for objects without obstruction and are very hard to calculate. Using Monte Carlo integration and ray tracing provides a fast and elegant tool for the estimation of the form factors. In this paper we show, that using deterministic low discrepancy sample points is superior to random sampling, resulting in an acceleration of more than half an order of magnitude.

## 1. Introduction

In computer graphics most integrals have a discontinuous kernel and as such are hard to be solved analytically. In addition visibility has to be checked, which is an expensive operation. For the evaluation of such integrals, Monte Carlo methods provide a efficient and elegant tool. On a computer the random samples used for Monte Carlo integration, are approximated by means of pseudo-random numbers. But there exist deterministic point sets especially designed for integration, which promise a convergence faster than the Monte Carlo rate of $\mathcal{O}(\frac{1}{\sqrt{N}})$, where $N$ is the number of samples drawn.

The paper now investigates the application of so-called *low discrepancy points* for the form factor integral and compares it to random sampling. Therefore we introduce the *quasi-Monte Carlo method* (for a profound introduction to quasi-Monte Carlo integration and low discrepancy points, see [Nie92b]) in the next section. Then we explain the algorithm used for the calculation of the form factors. After discussing the numerical evidence of some experiments, we draw the conclusions.

## 2. Monte Carlo and Quasi-Monte Carlo Integration

In computer graphics we often have integrands with discontinuities, that are not axis-aligned. So usual quadrature rules do not work at maximum efficiency and we therefore use the Monte Carlo method to approximate an integral on the unit cube by

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

where $P_N = \{x_0, \ldots, x_{N-1}\}$ is a uniformly distributed sequence of points in the $s$-dimensional unit cube $[0,1)^s = I^s$. For the classical Monte Carlo method, these points are chosen randomly (on a computer typically modelled by pseudo-random numbers of a linear congruential generator). For this choice of sampling points the expectation of the error 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 $\sigma^2(g)$ is the variance of $g$. Using quasi-random numbers, i.e. low-discrepancy point sets, for the $x_i$ (see [Nie92b]) results in the Koksma-Hlawka inequality

$$\left|\int_{I^s} g(x)\,dx - \frac{1}{N}\sum_{i=0}^{N-1} g(x_i)\right| \leq V(g)\ D^*(P_N) \tag{2.1}$$

where $V(g)$ is the variation in the sense of Hardy and Krause and $D^*(P_N)$ is the discrepancy of the point set $P_N$, i.e. its deviation from uniform distribution (see definition 2.1). In computer graphics the variation of the integrand $g$ is infinite, and so this upper error bound cannot be applied.

2.1. **Approximation of Continuous Measures by Discrete Measures.** The integrand, which is to be evaluated, often can be split into three factors:

$$\int_{I^s} f(x)\ \chi_A(h(x))\ p(x)\ dx = \int_{I^s} f(y)\ \chi_A(h(y))\ d\mu(y) \tag{2.2}$$

$f$ is a positive, bounded, and discontinuous function. The function $h(x)$ is a mapping from $I^s$ onto the set $S$. $\chi_A$ is the characteristic function of the subset $A \subset S$, i.e. $\chi_A(h(x)) = 1 \Leftrightarrow h(x) \in A$, $\chi_A(h(x)) = 0$ otherwise. The function $p$ is a density function 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, the inverse $\mu^{-1}$ is defined by the multidimensional inversion method[1] (see [HM72] and [Wic74]). For the sequel we also presume $\mu^{-1}$ to be of bounded variation in the sense of Hardy and Krause (see [Nie92b]). We now approximate the measure $\mu$ by a discrete point cloud $C_N = \{y_0, \dots, y_{N-1}\}$ modelled out of the point set $P_N$ by $y_i = \mu^{-1}(x_i)$. For the investigation of the approximation we define

**Definition 2.1.** The discrepancy $D^*(P_N)$ is a measure for the deviation of a point set $P_N$ 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:

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

Using (2.1), the discrepancy between $C_N$ and $p$ is bounded by

$$D^*(p, C_N) := \sup_{J=\prod_{j=1}^s [0,a_j)\subset I^s} \left|\int_{I^s} \chi_J(x)\ p(x)\ dx - \frac{1}{N}\sum_{i=0}^{N-1} \chi_J(y_i)\right| \leq V(\mu^{-1})\ D^*(P_N)\ . \tag{2.3}$$

[1] A more practical introduction to this method including a table of useful transformations can be found in [Shi92] (there the multidimensional inversion method is called *warping*).

*Remark 2.2.* If $p$ is separable, i.e. $p(x) = \prod_{j=1}^{s} p^{(j)}(x^{(j)})$, we have $D^*(p, C_N) = D^*(P_N)$.

These bounds for $D^*(p, C_N)$ are results of [Wic74]. By

$$\int_{I^s} f(y)\, \chi_{A_k}(h(y))\, d\mu(y) \approx \int_{I^s} f(y)\, \chi_{A_k}(h(y))\, d\mu_N(y) := \frac{1}{N} \sum_{i=0}^{N-1} f(y_i)\, \chi_{A_k}(h(y_i)) \tag{2.4}$$

we calculate (2.2) using the discrete measure $\mu_N := \frac{1}{N} \sum_{i=0}^{N-1} \delta_{y_i}$ as approximation for the continuous measure $\mu$. 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 $D^*(P_N)$.

**Low Discrepancy Points.** There exist deterministic point sets and sequences, which have an asymptotically better discrepancy than random points, which is

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

almost surely (see [Nie92b], pp. 166 - 167). These so-called low discrepancy points mostly base on radical inversion and modifications of this inversion. The principle of radical inversion is to transfer the natural number $i$ into base-$b$-representation and to mirror that representation at the decimal point, which results in the radical inverse

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

The simplest points of that kind are the Halton and Hammersley points. The Halton sequence for $s$ dimensions is built by

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

where the base $b_j$, $1 \le j \le s$, mostly is chosen to be the $j$-th prime number. For this sequence the discrepancy of the first $N$ points is bounded by (see [Nie92b])

$$D^*(P_N^{\text{Halton}}) \in \mathcal{O}\left(\frac{\log^s N}{N}\right)$$

The Hammersley point set even has a smaller discrepancy, but is a finite point set of $N$ points in contrast to the infinite Halton sequence:

$$x_i = (\frac{i}{N}, \Phi_{b_1}(i), \ldots, \Phi_{b_{s-1}}(i)) \ .$$

The discrepancy is bounded by (see [Nie92b]):

$$D^*(P_N^{\text{Hammersley}}) \in \mathcal{O}\left(\frac{\log^{s-1} N}{N}\right)$$

So the deterministic low discrepancy points have an asymptotically smaller discrepancy than (pseudo-) random numbers and therefore better uniformity properties, which are important for numerical integration. Other low discrepancy point sets have a different constant, but the order stays the same. In figure 1 code fragments for the calculation of the radical inverse function are shown. The routine on the

```
void Φ(int b, int i)
{
    double x = 0.0;
    double f = 1/b;

    while(i)
    {
        x += f * (double) (i % b);
        i /= b;
        f *= 1/b;
    }

    return x;
}
```

```
void Φ(int b, double x)
{
    double h,hh,r;

    r = 1.0 - x - 1e-10;

    if(1/b < r)
        x += 1/b;
    else
    {
        h = 1/b;

        do
        {
            hh = h;
            h *= 1/b;
        }
        while(h >= r);

        x += hh + h - 1.0;
    }

    return x;
}
```

**Fig. 1.** Direct and incremental calculation of the radical inverse $\Phi_b$

lefthand side directly calculates $\Phi_b(i)$, whereas the righthand routine operates much faster for incrementally calculating $\Phi_b(i)$ out of $x = \Phi_b(i-1)$ (see [HW64]). The speed of generation is slightly slower than the speed of good quality random number generators. Other methods for the fast generation of generalized Halton points can be found in [Str95].

## 3. Calculation of the Form Factors

In computer graphics a 3-dimensional scene is usually given as boundary representation. So the 2-dimensional surface $S = \cup_{k=1}^{K} A_k$ is the union of $K$ surface primitives $A_k$ (e.g. triangles). The form factor

$$f_{i-j} := \frac{1}{\pi |A_i|} \int_{A_i} \int_{\Omega} \chi_{A_j}(h(x, \omega)) \cos\theta(\omega, x)\, d\omega\, dx \tag{3.1}$$

is the percentage of diffuse radiance, that is directly radiated from $A_i$ to $A_j$. Form factors are used in many radiosity algorithms (e.g. several variations of progressive refinement [GCS94]). $h(x, \omega)$ is the first point hit by a ray started in $x \in A_i$ in direction $\omega \in \Omega$. $\Omega$ is the set of all directions over the hemisphere in $x$. The integral (3.1) accounts for all form factors from $A_i$ to all other surface elements $A_j$, $1 \leq j \leq K$. The characteristic function $\chi_{A_j}$ is used to select the hitpoints concerning $A_j$. The cosine, taken from the azimuth angle $\theta$ between the direction $\omega$ and the surface normal in $x$, projects the differential area $dx$ perpendicular to the direction $\omega$.

3.1. **Construction of the Algorithm.** Using the principles of the previous section, especially 2.3 and 2.4, we derive a quasi-Monte Carlo algorithm: First we transform (3.1) into spherical coordinates and onto the 4-dimensional unit cube $I^4 = [0,1)^4$. Then we use the trigonometric terms as probability density and rewrite the integral as Riemann-Stieltjes-integral, using the cumulative distribution function as measure (For the sake of simplicity, we assume $A_j$ to be the unit square. Area preserving mappings for other geometries, like triangles etc., can be found using [HM72] or in [SWZ96] and [Shi92]):

$$\begin{aligned} f_{i-j} &= \frac{1}{\pi|A_i|}\int_{A_i}\int_0^{2\pi}\int_0^{\frac{\pi}{2}} \chi_{A_j}(h(x,\omega(\theta,\phi)))\ \cos\theta\ \sin\theta\ d\theta\ d\phi\ dx \\ &= \frac{\pi}{2}\int_{I^4} \chi_{A_j}(h(x_1,x_2,\omega(\frac{\pi}{2}x_4, 2\pi x_3)))\ \sin\pi x_4\ dx_4\ dx_3\ dx_2\ dx_1 \\ &= \int_{I^4} \chi_{A_j}(h(y_1,y_2,\omega(\frac{\pi}{2}y_4, 2\pi y_3)))\ d\sin^2\frac{\pi}{2}y_4\ dy_3\ dy_2\ dy_1 \\ &\approx \frac{1}{N}\sum_{k=0}^{N-1} \chi_{A_j}(h(y_{k,1},y_{k,2},\omega(\frac{\pi}{2}y_{k,4}, 2\pi y_{k,3}))) =: \tilde{f}_{i-j} \end{aligned} \tag{3.2}$$

Using parts of the integrand as density prevents the expensive rays to be weighted by small trigonometric factors (importance sampling). Instead we generate the samples $y_k$ out of the uniformly distributed $x_k$ using the inverse cumulative distribution function:

$$\begin{aligned} (y_{k,1}, y_{k,2}, y_{k,3}, \sin^2\frac{\pi}{2}y_{k,4}) &= (x_{k,1}, x_{k,2}, x_{k,3}, x_{k,4}) = x_k \\ \Leftrightarrow y_k = (y_{k,1}, y_{k,2}, y_{k,3}, y_{k,4}) &= (x_{k,1}, x_{k,2}, x_{k,3}, \frac{2}{\pi}\arcsin\sqrt{x_{k,4}}) \end{aligned}$$

Although the algorithm resembles the Monte Carlo method used in [Shi91], it is constructed using a totally different theoretical framework (see previous section). The quasi-Monte Carlo algorithm now can be implemented easily: For each sample $x_k$ we calculate $y_k$ as the start parameters for the ray. The object identified by $h(y_{k,1}, y_{k,2}, \omega(\frac{\pi}{2}y_{k,4}, 2\pi y_{k,3}))$ then keeps a counter for the number of hits. After distributing the rays, dividing the counters of each surface element by $N$ gives an approximation for the form factor. By simultaneously calculating all form factors related to $A_i$ by one sequence of sample points, this algorithm does not waste any rays for visibility tests! The counting can be done with integer arithmetic. Note, that $\sum_{j=1}^{K} f_{i-j} = 1$, $f_{i-i} = 0$ for planar elements and $A_i f_{i-j} = A_j f_{j-i}$ for $1 \le i, j \le K$.

3.2. **Numerical Evidence.** To illustrate the superiority of quasi-Monte Carlo integration due to the better discrepancy of the sampling points, we give some numerical evidence. For the simulation we used different random number generators. `drand48()` is the internal random generator of HP-UX. The explicit inversive generator is taken from [Nie92a]. These two pseudo-random number generators approximate discrepancy of order $\sqrt{\frac{\log\log N}{N}}$ of random numbers, whereas the Halton and Hammersley points have a discrepancy of order $\frac{\log^s N}{N}$ and $\frac{\log^{s-1} N}{N}$, respectively. In addition we

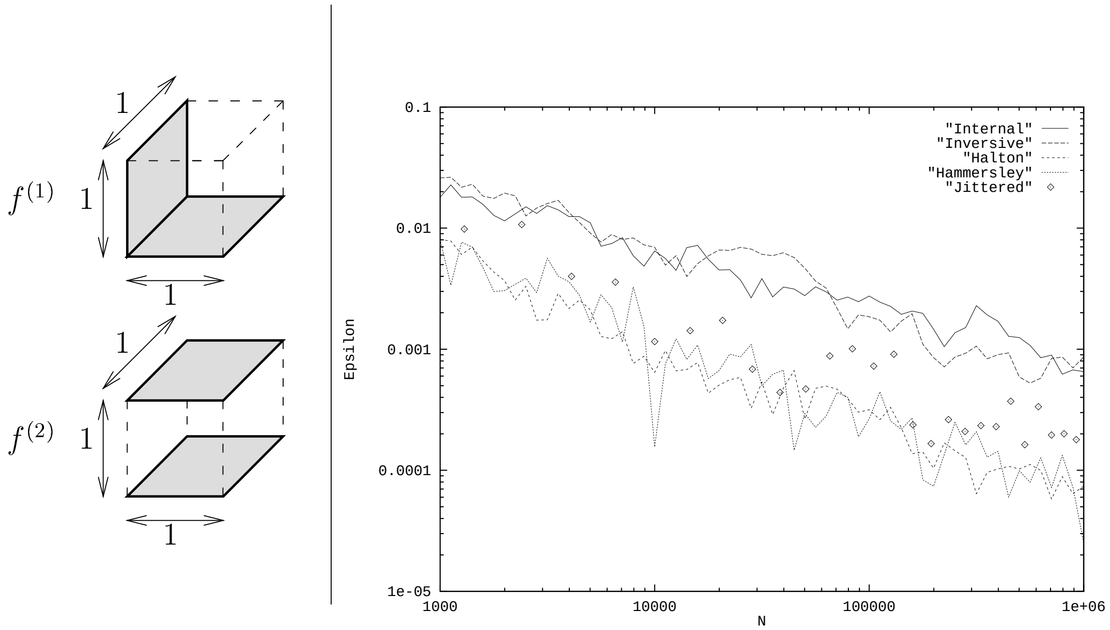


**Fig. 2.** Form factors and $L_\infty$ - error for the measurement in the unit cube.

show the results for jittered sampling [CPC84], where, due to constuction, only the results for those $N$ are shown, where $N = \lfloor \sqrt[s]{N} \rfloor^s$.

**Experiment 1.** The first experiment uses (3.2) for the approximation of the form factors in an empty unit cube. Here the density is separable, and we have $D^*(p, C_N) = D^*(P_N)$ due to remark 2.2. Due to symmetry we have three possible form factors for which there exist analytic, closed form solutions (see [CW93]). The form factor from a unit square to itself is zero (as mentioned before). For two unit squares perpendicular to each other and sharing one edge we have $f^{(1)} = \frac{1}{2} - \frac{1}{\pi}(\frac{1}{4}\ln\frac{4}{3} + \sqrt{2}\tan^{-1}\frac{1}{\sqrt{2}}) \approx 0.2000437761$. For two coplanar unit squares parallel to each other with distance 1 we hence have $f^{(2)} = 1 - 0 - 4f^{(1)} \approx 0.1998248957$ (see the figure 2 for $f^{(1)}$ and $f^{(2)}$).

For the numerical experiment we simultaneously evaluated the six form factors (that is, we only used one point cloud) as seen from the ground plane of the unit cube and compared the numerical results to the analytical solution. In the graph in figure 2 we plotted the absolute error, which is the $L_\infty$-distance of the analytical solution and numerical approximation for a fixed surface element $A_i$:

$$\max_{1 \le j \le K} |f_{i-j} - \tilde{f}_{i-j}|$$

As can be seen in the graph in figure 2, the sampling patterns with the lower discrepancy are superior by one order of magnitude, i.e. the better the measure $\mu$ is approximated by $\mu_N$, the better is the quadrature, and even beat the stratified random pattern, i.e. jittered sampling. Seen from a physical point of view, it is only natural to simulate the density $p$ by particles, since the basic nature of radiation can be interpreted by the corpuscle model. And again it is obvious, that the smaller the discrepancy $D^*(p, C_N)$, the better the real particles are approximated.

**Experiment 2.** In a real setting we are confronted with obstruction and transformations of the unit square onto the surface of the primitives (see [SWZ96]). Due to the transformation of the unit square onto a triangle, the sampling densities are

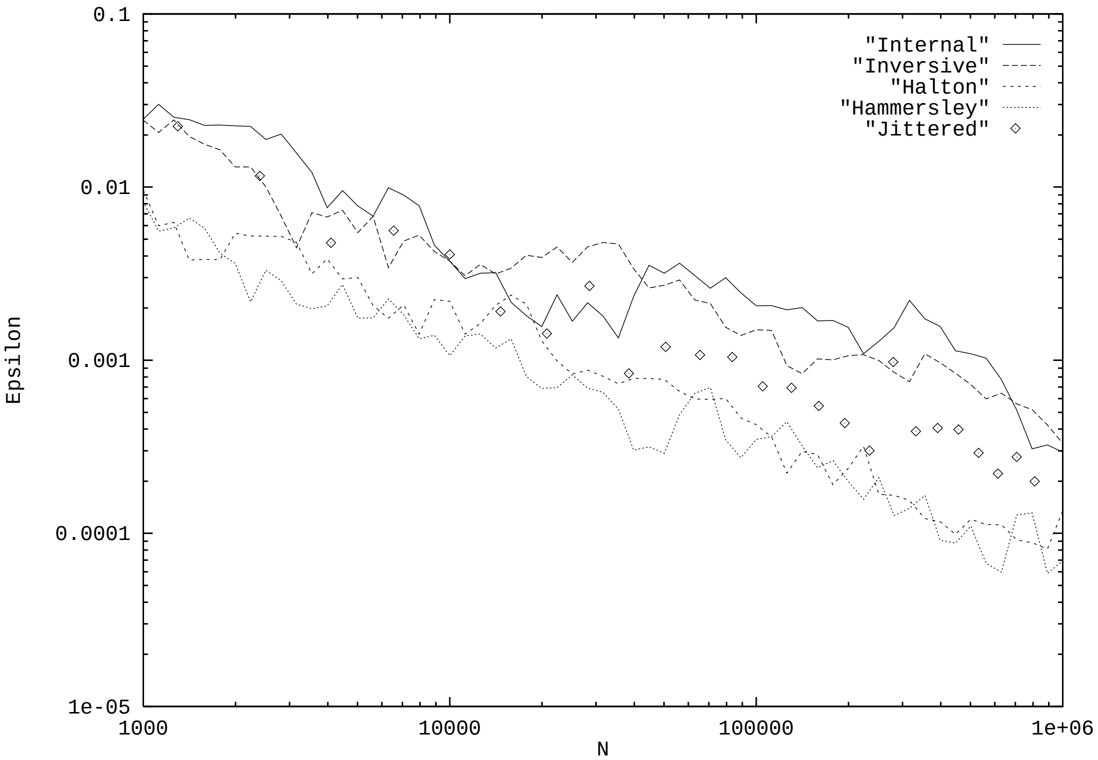


**Fig. 3.** $L_\infty$ - error for measurement in scene 1.

not longer separable, and we have $\mathcal{O}(D^*(p, C_N)) = \mathcal{O}(D^*(P_N))$ according to (2.3), since the variation of this transformation is bounded. The second experiment uses two realistic scenes modelled by triangles. From each scene we selected one triangle and calculated its form factors to all other elements. In the graphs in figure 3 and figure 4 we compared the $L_\infty$-error to a master calculation done by $N = 10^7$ samples (see figure 5 at end of paper for images). Clearly the low discrepancy sampling patterns beat random sampling by a factor $5 \ldots 10$. Compared to jittered sampling, low discrepancy sampling performs superior and is available for any choice of $N$.

## 4. Discussion

Since the Koksma-Hlawka inequality (2.1) cannot be used in this setting (for dimension $s = 2$ see [HK94]) if the discontinuities are not axis-aligned, we considered the difference between Monte Carlo and quasi-Monte Carlo integration, i.e. the sampling pattern. Regarding Monte Carlo and quasi-Monte Carlo methods as generation of discrete measures allows, especially in the domain of quasi-Monte Carlo methods, statements about the quality of such methods for discontinuous integrands in comparison with standard Monte Carlo methods. By theory we showed, that generating a discrete measure by low discrepancy point sets is more exact by order of magnitude than creating it by random points. Obviously the property of uniform distribution, measured by discrepancy, is more important than randomness for integration.

From the graphs we observe, that quasi-Monte Carlo integration consistently outperforms the Monte Carlo rate of $\mathcal{O}(N^{-\frac{1}{2}})$. The algorithm used in this paper is much more exact than the rough estimate of [WEH89] (look at the quality of approximation for the analytical experiment). Including the calculation of extended form factors is straightforward: By a random decision, according to the BRDF, a surface is decided

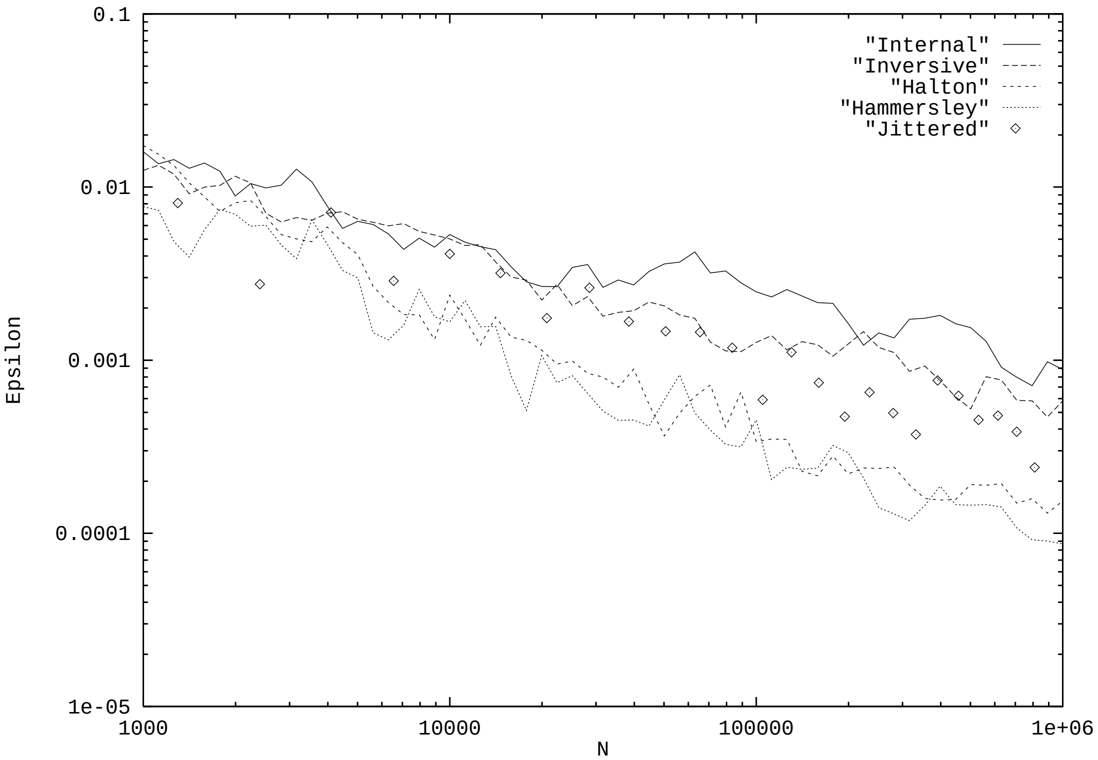


**Fig. 4.** $L_\infty$ - error for measurement in scene 2.

to be diffuse or specular. In the diffuse case the hitpoint contributes to that surface. In the specular case, the ray is scattered and the same procedure is performed on the next element hit.

Concerning aliasing, when repeatedly calculating form factors for example in progressive refinement, this can be avoided by the usual rotation of the local coordinate system of a surface element by some random offset $\phi_\xi$, which is added to $\phi$. Note, that this does not affect the quality of the quadrature formula. As compared to jittered and multi-jittered sampling, our algorithm is suited for any choice of the number $N$ of rays shot. In addition, low discrepancy sequences provide a minimum distance property, while simultaneously minimizing discrepancy. The Hammersley point set is a special case of $N$-rooks sampling, but much more easy to generate.

Obviously, the method has the disadvantage, that at small rates $N$, elements with small form factors may be missed. This can be fixed by increasing $N$ or by using directed shooting as in [WEH89] with low discrepancy points, but then at the cost of a systematic error. Our method can be used not only for the calculation of form factors, but also for any integral of the form 3.2 with a kernel other than the characteristic function. This applies to undirected shooting or gathering as for example in [NNP$^+$95]. Since the method is deterministic, the correction step of overshooting algorithms (see [FP92]) can be calculated exactly.

## 5. Conclusion

For the central problem of many radiosity algorithms, the form factor calculation, we illustrated the superiority of (deterministic) quasi-Monte Carlo integration as compared to Monte Carlo integration and gave numerical evidence for the remarkable speed-up of factor $5 \ldots 10$.

Quasi-Monte Carlo integration is also applicable for pixel oversampling [HK94], the generation of random walks and the final gathering pass in radiosity [Kel95]. In these applications using low discrepancy points sets results in faster algorithms (up to factor $\frac{2}{3}$), because the same accuracy of integration is aquired with less samples than random sampling.

Due to their construction, low discrepancy points replace simple domain stratification. For numerically calculating integrals with high sample numbers they are superior due to their better uniformity properties than random sampling.

## 6. Acknowledgements

The author would like to thank Tim Gleue for the numerical experiments.

## References


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

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

[FP92] M. Feda and W. Purgathofer, *Accelerating radiosity by overshooting*, 3rd Eurographics Workshop on Rendering (Bristol, England), May 1992, pp. 21 – 32.

[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.

[Kel95] A. Keller, *Quasi-Monte Carlo Methods in Computer Graphics: The Global Illumination Problem*, Proc. of the SIAM Conference in Park City (1995), to appear.

[Nie92a] H. Niederreiter, *New methods for pseudorandom number and pseudorandom vector generation*, Winter Simulation Conference (Arlington, VA, 1992) (Piscataway, NJ), IEEE Press, 1992, pp. 264–269.

[Nie92b] H. Niederreiter, *Random number generation and quasi-Monte Carlo methods*, SIAM, Pennsylvania, 1992.

[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.

[Shi91] P. Shirley, *Radiosity via ray tracing*, Graphics Gems II (J. Arvo, ed.), Academic Press, 1991, pp. 306–310.

[Shi92] Peter Shirley, *Nonuniform random point sets via warping*, Graphics Gems III (San Diego) (David Kirk, ed.), Academic Press, San Diego, 1992, summary of various useful sample generation transformations, pp. 80–83.

[Str95] J. Struckmeier, *Fast generation of low-discrepancy sequences*, J. of Computational and Applied Mathematics (1995), no. 61, 29–41.

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

[WEH89] J. Wallace, K. Elmquist, and E. Haines, *A ray tracing algorithm for progressive radiosity*, Computer Graphics, July 1989, pp. 315–324.

[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.

**Fig. 5.** Sample views of the scenes 1 and 2.

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