Technical Report/TR-186-2-98-22 (*20 August 1998*)

Analysis of the Quasi-Monte Carlo Integration of the Rendering Equation

László Szirmay-Kalos, 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

Quasi-Monte Carlo integration is said to be better than Monte-Carlo integration since its error bound can be in the order of $O(N^{-(1-\epsilon)})$ instead of the $O(N^{-0.5})$ probabilistic bound of classical Monte-Carlo integration if the integrand has finite variation. However, since in computer graphics the integrand of the rendering equation is usually discontinuous and thus has infinite variation, the superiority of quasi-Monte Carlo integration has not been theoretically justified. This paper examines the integration of discontinuous functions using both theoretical arguments and simulations and explains what kind of improvements can be expected from the quasi-Monte Carlo techniques in computer graphics.

Keywords:

Rendering equation, quasi-Monte Carlo quadrature, Hardy-Krause variation

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:

$$L(\vec{x},\omega) = L^e(\vec{x},\omega) + \int_{\Omega'} L(h(\vec{x},-\omega'),\omega') \cdot \cos\theta' \cdot f_r(\vec{x},\omega',\omega)\, d\omega' \qquad (1)$$

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 ω', and $f_r(\vec{x},\omega',\omega)$ is the bi-directional reflection/refraction function (figure 1).

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

$$\int_{\Omega'} L(h(\vec{x},-\omega'),\omega') \cdot \cos\theta' \cdot f_r(\vec{x},\omega',\omega)\, d\omega' = \mathcal{T}L(\vec{x},\omega). \qquad (2)$$

Thus the short form of the rendering equation is:

$$L = L^e + \mathcal{T}L. \qquad (3)$$

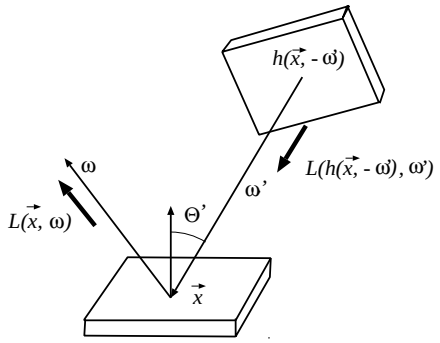

Figure 1: *Geometry of the rendering equation*

Generally two methods exists that can solve this equation without simplifying the transport laws:

- finite element methods [4] which convert the integral equation into an approximating linear equation that can be solved by straightforward techniques,
- random walk methods [18] that expand the rendering equation into a Neumann series, and evaluate its terms by multi-dimensional, Monte-Carlo quadrature.

This paper investigates the random-walk solution when the walk is not governed by random decisions but deterministic selection from appropriate uniform sequences.

In equation 3 the unknown function L appears on both sides. Recursively substituting the right side's L by $L^e+\mathcal{T}L$, which is obviously L according to the equation, we get the following Neumann series:

$$L = L^e + \mathcal{T}L = L^e + \mathcal{T}(L^e + \mathcal{T}L) =$$

$$L^e + \mathcal{T}L^e + \mathcal{T}^2 L = \sum_{i=0}^{n} \mathcal{T}^i L^e + \mathcal{T}^{n+1} L. \qquad (4)$$

If integral operator $\mathcal{T}$ is contractive, that is if $||\mathcal{T}L|| \leq s \cdot ||L||,\ s < 1$, then $\lim_{n\to\infty} \mathcal{T}^{n+1}L = 0$, thus

$$L = \sum_{i=0}^{\infty} \mathcal{T}^i L^e. \qquad (5)$$

The contractive property of $\mathcal{T}$ comes from the fact that a reflection or refraction always decreases the energy.

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

In order to study the structure of $\mathcal{T}^i L^e$, let us consider the case of $i = 2$:

$$\mathcal{T}^2 L^e = \mathcal{T}(\mathcal{T}L^e) =$$

$$\int_{\Omega_1} (\mathcal{T}L^e)(h(\vec{x}, -\omega_1'), \omega_1') \cdot \cos\theta_1' \cdot f_r(\vec{x}, \omega_1', \omega)\ d\omega_1' =$$

$$\int_{\Omega_1'} \int_{\Omega_2'} L^e(h(h(\vec{x}, -\omega_1'), -\omega_2'), \omega_2') \cdot \cos\theta_1' \cdot \cos\theta_2' \cdot$$

$$f_r(\vec{x}, \omega_1', \omega) \cdot f_r(h(\vec{x}, -\omega_1'), \omega_2', \omega_1',)\ d\omega_2' d\omega_1'. \qquad (6)$$

To evaluate the integrand at point $(\vec{x}, \omega, \omega_1', \omega_2')$, the following algorithm must be executed:

1. Surface point $\vec{y}_1 = h(\vec{x}, -\omega_1')$ — that is the point which is visible from $\vec{x}$ at direction $-\omega_1'$ — must be determined. This can be done by sending a ray from $\vec{x}$ into direction $-\omega_1'$ and identifying the surface that is first intersected.
2. Surface point $\vec{y}_2 = h(h(\vec{x}, -\omega_1'), -\omega_2')$ — that is the point visible from $\vec{y}_1$ at direction $-\omega_2'$ is identified. This means the continuation of the ray tracing at direction $-\omega_2'$.
3. The emission intensity of the surface at $\vec{y}_2$ in the direction of ω_2' is obtained and is multiplied with the cosine terms and the BRDFs of the two reflections.

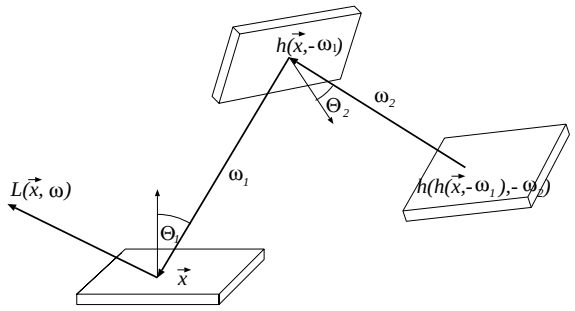

Figure 2: *The integrand of $\mathcal{T}^2 L^e$ is a two-step walk*

This algorithm can easily be generalized for arbitrary number of reflections. In order to compute the contribution of the nth reflection of the light to $L(\vec{x}, \omega)$, the following function should be integrated: A ray is emanated recursively from $\vec{x}$ at direction $-\omega_1'$, then from the found surface at $-\omega_2'$, etc. until $-\omega_n'$. The emission intensity at the end of the walk is read and multiplied by the BRDFs and the cosine terms of the stages of the walk.

These walks provide the value of the integrand at "point" $\omega_1', \omega_2', \ldots, \omega_n'$. The integral is estimated from the values at these points using some quadrature rule. The calculated integral has usually high dimension. 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.

Due to the visibility term $h(\vec{x}, -\omega')$ in the rendering equation, an estimation of dth 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 $-\omega'$ and finds the nearest intersection.

Since classical quadrature techniques based on uniform grids, such as trapezodial rule, may have integration error of order $O(1/N^{1/s})$ where N is the number of dot points and s is the dimension of the domain, in order to guarantee a predefined error, the number of dot points should be in the order of $O(N^s)$, which is prohibitive for high-dimensional integrals. This phenomenon is called as the *dimensional explosion.*

The dimensional explosion can be avoided by Monte-Carlo or Quasi-Monte Carlo integration methods. Monte-Carlo methods trace back the estimation of an integral to the

calculation of an expected value, which is estimated by averaging random samples. Quasi-Monte Carlo techniques, on the other hand, use deterministic samples that are uniformly distributed in the integration domain.

Monte-Carlo quadrature considers the integral as an expected value which can be estimated by averaging random points:

$$\int_V f(\mathbf{z})\ d\mathbf{z} = V \cdot \int_V f(\mathbf{z}) \cdot \frac{1}{V}\ d\mathbf{z} = V \cdot E[f(\mathbf{z})] \qquad (7)$$

since $1/V$ can be considered as the probability density of a random variable uniformly distributed in V.

If N number of points are used to estimate an integral of f in the domain of size V, the Monte-Carlo quadrature will have the following (stochastic) error bound:

$$\int_V f(\mathbf{z})\ d\mathbf{z} = V \cdot \int_V f(\mathbf{z}) \cdot \frac{1}{V}\ d\mathbf{z} \approx \frac{V}{N} \cdot \sum_{i=1}^{N} f(\mathbf{z}_i) \pm V \cdot \frac{\sigma}{\sqrt{N}}. \qquad (8)$$

The variance σ is defined by

$$\sigma^2 = E[(\hat{f}(\mathbf{z}) - E[f(\mathbf{z})])^2] = \int_V (\hat{f}(\mathbf{z}) - E[f(\mathbf{z})])^2 \cdot \frac{1}{V}\ d\mathbf{z}. \qquad (9)$$

If $f(\mathbf{z})$ is square-integrable function, then σ is finite, thus the stochastic error bound will be in the order of $O(N^{-0.5})$.

Quasi-Monte Carlo integration, on the other hand, uses deterministic point-sets that contain very uniformly distributed points in the integration domain. For the normalized, s-dimensional integration domain $[0,1]^s$, the quasi-Monte Carlo approximation is:

$$\int_{[0,1]^s} f(\mathbf{x})\ d\mathbf{x} \approx \frac{1}{N} \sum_{i=1}^{N} f(\mathbf{x}_i). \qquad (10)$$

Sample points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ should be selected to minimize the error of the integral quadrature. Intuitively this error must depend on two independent factors. On the one hand, if the distribution of the sample points is not uniform, then there are large regions where there are no sample point at all, which increases the error. From mathematical point of view, the uniformity is characterized by the discrepancy of a point set. The star-discrepancy is defined by

$$D^*(\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_N) = \sup_A \left| \frac{m(A)}{N} - V(A) \right| \qquad (11)$$

where A is an s-dimensional sub-cube parallel to the coordinate axes and originating at the center, $V(A)$ is its volume, and $m(A)$ is the number of sample points inside this subcube. For carefully selected sample points, called *low-discrepancy sequences* [14], the discrepancy can be in the order of $O(\log^{s-1} N/N)$.

On the other hand, the error of quadrature also depends on how quickly the function changes between the sample points. If the function can change significantly in small domain, then the error can be quite large. However, if the slope of the function is small, then nothing dramatic happens between the sample points, thus the error will be small.

Measures describing how rapidly a function can change are called **variations**. For a 1-dimensional function, the *variation in the sense of Vitali* is defined as:

$$\mathcal{V}_{\mathrm{V}}(f(x)) = \limsup \sum_{i=1}^{n} |f(x_{i+1}) - f(x_i)|. \qquad (12)$$

For a 2-dimensional function, the definition is analogous:

$$\mathcal{V}_{\mathrm{V}}(f(x,y)) =$$

$$\limsup \sum_{i=1}^{n} \sum_{j=1}^{m} |f(x_{i+1}, y_{j+1}) - f(x_{i+1}, y_j) - f(x_i, y_{j+1}) + f(x_i, y_j)|. \qquad (13)$$

Note that for higher dimensions, the variation of Vitali does not provide a good measure: if the function is constant in x, for instance, then the variation is zero, regardless how the function changes depending on y. Thus, it is worth using a somehow more stronger variation type, called the *Hardy-Krause variation.* The variation in the sense of Hardy and Krause is the sum of the variations of the function and its restrictions to the end of the domain. For dimension 2, the new variation is:

$$\mathcal{V}_{\mathrm{HK}}(f(x,y)) = \mathcal{V}_{\mathrm{V}} f(x,y) + \mathcal{V}_{\mathrm{V}} f(x,1) + \mathcal{V}_{\mathrm{V}} f(1,y). \qquad (14)$$

If a function has bounded and continuous mixed derivatives, then its variation is finite. For a 2-dimensional function meeting this requirement, the variation can be given by the following formula:

$$\mathcal{V}_{\mathrm{HK}}(f(u,v)) = \int_0^1 \int_0^1 \left| \frac{\partial^2 f(u,v)}{\partial u \partial v} \right| du\ dv +$$

$$\int_0^1 \left| \frac{\partial f(u,1)}{\partial u} \right| du + \int_0^1 \left| \frac{\partial f(1,v)}{\partial v} \right| dv. \qquad (15)$$

The property that a function is not continuous does not necessarily mean that the variation is infinite. If at most finite or countably infinite discontinuity occurs at hyperplanes parallel to the coordinate axes, then the variation is still finite. An example of a discontinuous function that have finite variation is

$$f(x,y) = \begin{cases} 1 \text{ if } x > x_0, \\ 0 \text{ otherwise.} \end{cases} \qquad (16)$$

However, when the discontinuity is not parallel to the coordinate axes, then the variation is infinite. A simple function

of infinite variation is [5]:

$$f(x,y) = \begin{cases} 1 \text{ if } x > y, \\ 0 \text{ otherwise.} \end{cases} \quad (17)$$

From these two factors — the discrepancy of the sample locations and the variation of the function — an upper-bound can be established for the error of the quadrature. If the integrand f has finite variation in the sense of Hardy and Krause, then the error of the quasi-Monte Carlo quadrature can be bounded as stated by the Koksma-Hlawka inequality:

$$\left| \int_{[0,1]^s} f(\mathbf{x})\, d\mathbf{x} - \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}_i) \right| \le \mathcal{V}_{\mathrm{HK}} \cdot D^*(\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_N), \quad (18)$$

where $\mathcal{V}_{\mathrm{HK}}$ is the variation in the sense of Hardy and Krause and $D^*(\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_N)$ is the star-discrepancy of the sample points [14]. As mentioned, for low-discrepancy point sets the discrepancy can be in the order of $O(\log^{s-1} N/N)$, which can provide much better error bounds than the $O(1/N^{1/s})$ bound of classical quadrature rules or the $O(1/\sqrt{N})$ probabilistic error bound of Monte Carlo methods. Moreover, quasi-Monte Carlo methods guarantee this accuracy in a deterministic way, unlike Monte Carlo methods where the error bound is also probabilistic.

2. Generation of Low-Discrepancy Sequences

As a conclusion of the error analysis we can state that we need very uniform sequences for quasi-Monte Carlo quadrature. The discrepancy of the best sequences known is in the order of $O(\log^s N/N)$ or even in the order of $O(\log^{s-1} N/N)$ if N is known before starting the sequence.

There are many sequences published in the literature [14, 20, 5, 11]. The most famous one is probably the *Halton sequence*.

The element i of the one-dimensional Halton sequence of base b is defined as the radical inverse of the expansion of i in base b. This means that number i is expanded in radix b, then the number is mirrored onto the "radix" point.

Why is this sequence uniform? Note that the construction algorithm generates as binary form of i all binary combinations of length k before producing a combination of length $k+1$. This means that after the radical inverse the sequence H_i will visit all intervals of length 2^{-k} before putting a new point in an interval already visited.

On the other hand, as k increases, the algorithm produces a single point in each interval of length $1/2$, then in each interval of length $1/4$, etc. thus the sequence is not only asymptotically uniform, but also the first N points are fairly uniformly distributed (this is guaranteed by the property that the radical inverse makes the most significant bit the most rapidly changing bit).

Note that the same construction can be used for any base, thus in this way infinitely many different Halton sequences can be constructed.

For higher dimensional Halton sequences, the interdependence of different coordinates should be as little as possible. Thus an appropriate way of the construction is selecting relative primes (say $b_1, b_2, \ldots b_s$) as bases for the different coordinates. A two-dimensional Halton sequence, for instance, would visit all grid cells of size $b_1^{-k_1}, b_2^{-k_2}$ before putting a new point into a cell if b_1 and b_2 are relative primes.

A C++ class that can be initialized to an arbitrary Halton point and then it can generate incrementally all subsequent points using a very fast algorithm [10] is presented in the following:

```
class Halton {
    double value, inv_base;
public:
    Number( long i, int base ) {
        double f = inv_base = 1.0/base;
        value = 0.0;
        while ( i > 0 ) {
            value += f * (double)(i % base);
            i /= base;
            f *= inv_base;
        }
    }
    void Next( ) {
         double r = 1.0 - value - 0.0000000001;
         if (inv_base < r) value += inv_base;
         else {
              double h = inv_base, hh;
              do {
                  hh = h;
                  h *= inv_base;
              } while ( h >= r );
              value += hh + h - 1.0;
         }
    }
    operator double( ) { return value; }
};
```

3. Integrating functions of unbounded variation

As mentioned, the integrand of the original form of the rendering equation is discontinuous where the discontinuity is not aligned with the coordinate axes, thus its variation is infinite. These discontinuities are usually produced by the projected object boundaries.

This property makes the Koksma-Hlawka inequality not appropriate for the error analysis of the solution of the rendering equation and for the prediction of the convergence rates.

In this section the convergence speed is examined for functions which are generally smooth but have general discontinuities of finite "length". First the domain of the integration is assumed to be 2-dimensional, then the results will be generalized to arbitrary dimension.

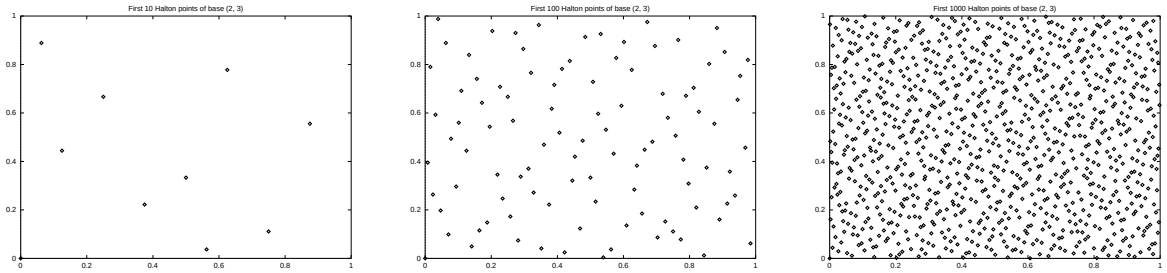

Figure 3: *The distribution of the first 10,100 and 1000 Halton points in 2 dimensions*

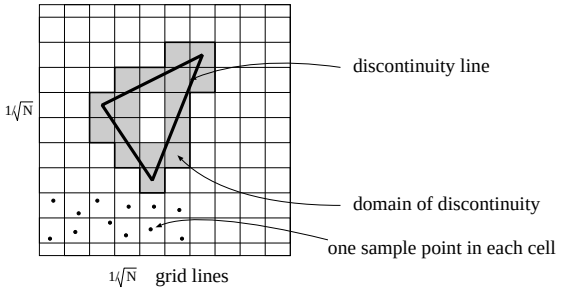

Figure 4: *A typical integrand of the rendering equation*

Suppose that N number of samples has been generated to estimate the integral of a function such as in figure 4 using a low-discrepancy sequence. In order to overcome the difficulty that the integrand f has infinite variation, the function is decomposed into two functions, one is smooth $\tilde{f}$ having continuous mixed derivatives and the other $\hat{f}$ inherits the discontinuity of f (figure 5).

Low-discrepancy sequences generate points in a cellular grid in a way that the difference of the number of points in two cells is at most 1. If there are already N number of points, the size of a cell on the finest, filled level is approximately $1/\sqrt{N} \times 1/\sqrt{N}$. Let us define the domain of $\hat{f}$ as the set of those cells that are intersected by the discontinuity. This domain is called the **domain of discontinuity**. The number of such cells is in the order of the "digital length" of the discontinuity curve, which is the product of the maximum extent l and the resolution of the grid $\sqrt{N}$. Since each cell has at least 1 (and at most 2) points, the number of points in this domain is at least $l\sqrt{N}$.

The error of quadrature is as follows:

$$\left| \int_{\mathbf{z}\in[0,1]^2} f(\mathbf{z})\, d\mathbf{z} - \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{z}_i) \right| \leq$$

$$\left| \int_{\mathbf{z}\in[0,1]^2} \tilde{f}(\mathbf{z})\, d\mathbf{z} - \frac{1}{N}\sum_{i=1}^{N} \tilde{f}(\mathbf{z}_i) \right| +$$

$$\left| \int_{\mathbf{z}\in[0,1]^2} \hat{f}(\mathbf{z})\, d\mathbf{z} - \frac{1}{N}\sum_{i=1}^{N} \hat{f}(\mathbf{z}_i) \right|. \qquad (19)$$

Since $\tilde{f}$ has finite variation, the first term in the error is bounded by $\mathcal{V}_{\mathrm{HK}}(\tilde{f}) \cdot D^*(\mathbf{z}_1, \mathbf{z}_2, \ldots \mathbf{z}_N)$.

Concerning the second term, the integration of $\hat{f}$ is estimated taking $l\sqrt{N}$ samples and averaging the result. Since the samples and the discontinuity are not related in any way, we can suppose that this is a normal Monte-Carlo integration [15]. The uniform property of low-discrepancy sequence guarantees that this pseudo-random set can be assumed to have uniform distribution.

If Δf is the difference between the maximum and minimum values in the domain of discontinuity, then $\sigma^2 \leq (\Delta f)^2$. In our case the number of sample points M is $l\sqrt{N}$ and $V = l/\sqrt{N}$, thus we obtain:

$$\int_V \hat{f}(\mathbf{z})\, d\mathbf{z} = \frac{V}{M} \cdot \sum_{i=1}^{M} \hat{f}(\mathbf{z}_i) \pm V \cdot \frac{\Delta f}{M} =$$

$$\frac{1}{N} \cdot \sum_{i=1}^{M} \hat{f}(\mathbf{z}_i) \pm \Delta f \cdot \sqrt{l} \cdot N^{-3/4}. \qquad (20)$$

Taking into account that $\hat{f}$ is zero outside the domain of discontinuity, equality 19 can be expressed as:

$$\left| \int_{\mathbf{z}\in[0,1]^2} f(\mathbf{z})\, d\mathbf{z} - \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{z}_i) \right| \leq$$

$$\mathcal{V}_{\mathrm{HK}}(\tilde{f}) \cdot D^*(\mathbf{z}_1, \mathbf{z}_2, \ldots \mathbf{z}_N) + \Delta f \cdot \sqrt{l} \cdot N^{-3/4}. \qquad (21)$$

For large N values the second term will be dominant, which results in $O(N^{-3/4})$ error bound. This is poorer than the

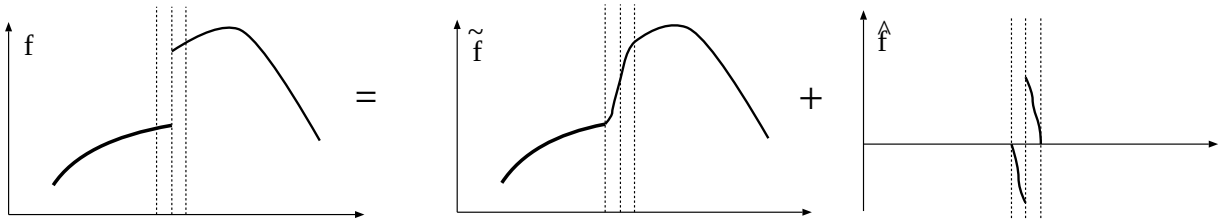

Figure 5: *Decomposition of f into a smooth ($\tilde{f}$) and a discontinuous ($\hat{f}$) function*

$O(\log^2 N/N)$ bound suggested by the Koksma-Hlawka inequality. Note that the point from where the second term dominates the first one depends on "intensity" Δf and size of the discontinuity $\sqrt{l}$.

The same analysis can be carried out in higher dimensions as well. In D dimensions a discontinuity of size l would intersect $V = l \cdot N^{-1/D}$ volume of cells which would contain $M = l \cdot N^{(D-1)/D}$ sample points. Thus the general error bound is:

$$\left| \int\limits_{\mathbf{z}\in[0,1]^D} f(\mathbf{z})\, d\mathbf{z} - \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{z}_i) \right| \leq$$

$$\mathcal{V}_{\mathrm{HK}}(\tilde{f}) \cdot D^*(\mathbf{z}_1, \mathbf{z}_2, \ldots \mathbf{z}_N) + \Delta f \cdot \sqrt{l} \cdot N^{-\frac{(D+1)}{2D}}. \quad (22)$$

Thus, for quasi-Monte Carlo integration of discontinuous functions, the order of the error bound will be in between the $O(N^{-(1-\epsilon)})$ bound of finite variation functions and $O(N^{-0.5})$ stochastic bound of Monte-Carlo quadrature. The higher the dimension of the integral, the closer the two methods get in this order. However, the proportionality ratio of the bounds may still be very different. Thus it is still worth using quasi-Monte Carlo quadrature if the size of the discontinuity l is not very large, since in this case the error could be significantly less then for Monte-Carlo quadrature.

When quasi-Monte Carlo quadrature is used for the solution of the rendering equation, it should be taken into account that the Neumann series contains a series of integrals of increasing dimension. If shooting type walks are used, then the effect of higher dimensional integrals gets lower due to the dissipation of energy. Thus the degradation of the convergence for higher dimensional integrals is partly compensated by their reduced importance.

Unfortunately this does not happen for gathering type walk, since here the final power may result from any step.

4. Numerical evidence for simple functions

In order to demonstrate the previous results, the convergence of a 2-dimensional and a 3-dimensional functions are examined, that are simple enough to analytically compute their integrals.

The 2-dimensional function is:

$$f_2(x,y) = \begin{cases} (x+y)\cdot a + 1 - 2\cdot a \text{ if } x+y > 1, \\ (x+y)\cdot a \text{ otherwise}, \end{cases} \quad (23)$$

where a is a free parameter in the range of $[0, 0.5]$. Note that by setting a appropriately, the intensity of the discontinuity can be controlled without altering either the value of the integral or the variation of the continuous part. If $a = 0.5$, then the function has finite variation, otherwise it has infinite variation.

The results of the simulation are shown in figure 6. This figure shows the maximum error after a given number of samples.

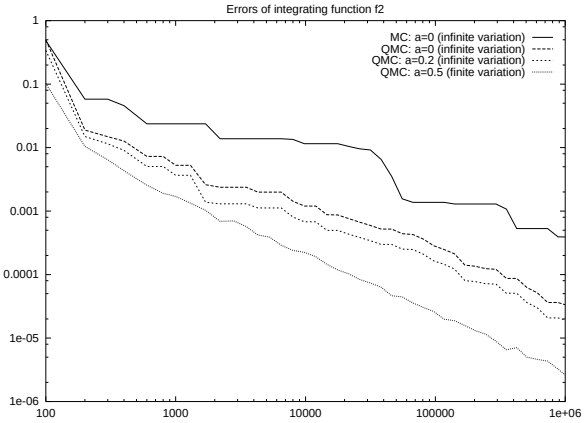

Figure 6: *Error of integrating f_2*

The 3-dimensional function is:

$$f_3(x,y,z) = \begin{cases} (x+y+z)\cdot a + 0.6 - 1.8\cdot a \\ \quad \text{if } x+y+z > 1, \\ (x+y+z)\cdot a \text{ otherwise}, \end{cases} \quad (24)$$

where a is a free parameter in the range of $[0, 1/3]$. If $a = 1/3$, then f_3 has finite variation, otherwise it has not. The error of integration of f_3 is summarized in figure 7.

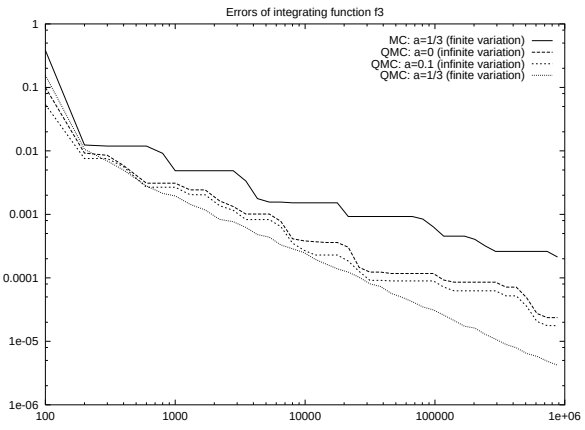

Figure 7: *Error of integrating f_3*

5. Numerical evidence for rendering

In order to obtain accurate error measurements for rendering algorithms, we need scenes for which the exact solution is known. Generally, we have two opportunities. Either, we use simple scenes for which analytical solution is possible, or numerical methods are applied using so many samples that the approximate solution can be accepted as a reference.

One of the famous analytically solvable case is the homogeneous environment, where the geometry is arbitrary, but all surfaces have the same diffuse reflection and emission [17, 8]. In this scene, however, the emission distribution is very homogeneous, thus quasi-Monte Carlo methods would be given an unfair handicap.

Thus we shall use another reference scene proposed by [7], which is an inner surface of a sphere, where the reflection is diffuse and homogeneous, but only a part of the surface has non-zero emission. Due to the not axis-aligned discontinuity of the lightsource the integrand of the rendering equation has infinite variation.

Here we derive the analytical solution somehow differently than in [7], since we also show that not only the complete solution can be obtained but also the partial solutions of different bounces. The knowledge of the contribution of different bounces allows us to use the reference solution to evaluate biased, finite random walk algorithms.

5.1. Analytical solution of the reference scene

Let the scene be an inner surface S of a sphere of radius R, the BRDF be constant $f_r = \varrho/\pi$ and the emission be diffuse and constant L^e in a subset A_e of the spherical surface and zero otherwise.

Using the

$$d\omega' = \frac{\cos\theta_{\vec{y}} \cdot d\vec{y}}{||\vec{y} - \vec{x}||^2}$$

substitution for the solid angle, we obtain the following form of the one bounce:

$$(\mathcal{T}L^e)(\vec{x},\omega) = \int_S f_r \cdot \cos\theta_{\vec{x}} \cdot L^e(\vec{y}) \cdot \frac{\cos\theta_{\vec{y}}}{||\vec{y} - \vec{x}||^2}\, d\vec{y} \quad (25)$$

Since $L^e(\vec{y})$ is constant and non-zero only inside A_e we get:

$$\int_S f_r \cdot \cos\theta_{\vec{x}} \cdot L^e(\vec{y}) \cdot \frac{\cos\theta_{\vec{y}}}{||\vec{y} - \vec{x}||^2}\, d\vec{y} =$$

$$f_r \cdot L^e \cdot \int_{A_e} \frac{\cos\theta_{\vec{x}} \cdot \cos\theta_{\vec{y}}}{||\vec{y} - \vec{x}||^2}\, d\vec{y}. \quad (26)$$

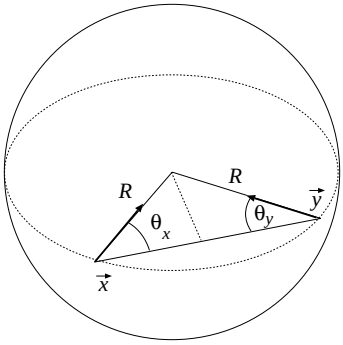

Figure 8: *Geometry of the reference scene*

Looking at figure 8, we can see that inside a sphere

$$\cos\theta_{\vec{x}} = \cos\theta_{\vec{y}} = \frac{||\vec{y} - \vec{x}||}{2R},$$

thus we can obtain the following final form for the one-bounce:

$$f_r \cdot L^e \cdot \int_{A_e} \frac{\cos\theta_{\vec{x}} \cdot \cos\theta_{\vec{y}}}{||\vec{y} - \vec{x}||}\, d\vec{y} = \frac{f_r \cdot L^e}{4R^2} \cdot A_e = \varrho \cdot \tilde{L}^e \quad (27)$$

where $\tilde{L}^e = L^e \cdot A_e/S$ is the average emission of the total spherical surface.

Summarizing, after the first bounce the radiance distribution is constant. Substituting this into L^e and replacing A_e by the total spherical surface S, we can derive the two bounce radiance transfer as well:

$$(\mathcal{T}^2 L^e)(\vec{x},\omega) = \left(\frac{f_r}{4R^2}\right) \cdot S \cdot \varrho \cdot \tilde{L}^e = \varrho^2 \tilde{L}^e. \quad (28)$$

Similarly, the n bounce radiance distribution is:

$$(\mathcal{T}^n L^e)(\vec{x},\omega) = \varrho^n \tilde{L}^e. \quad (29)$$

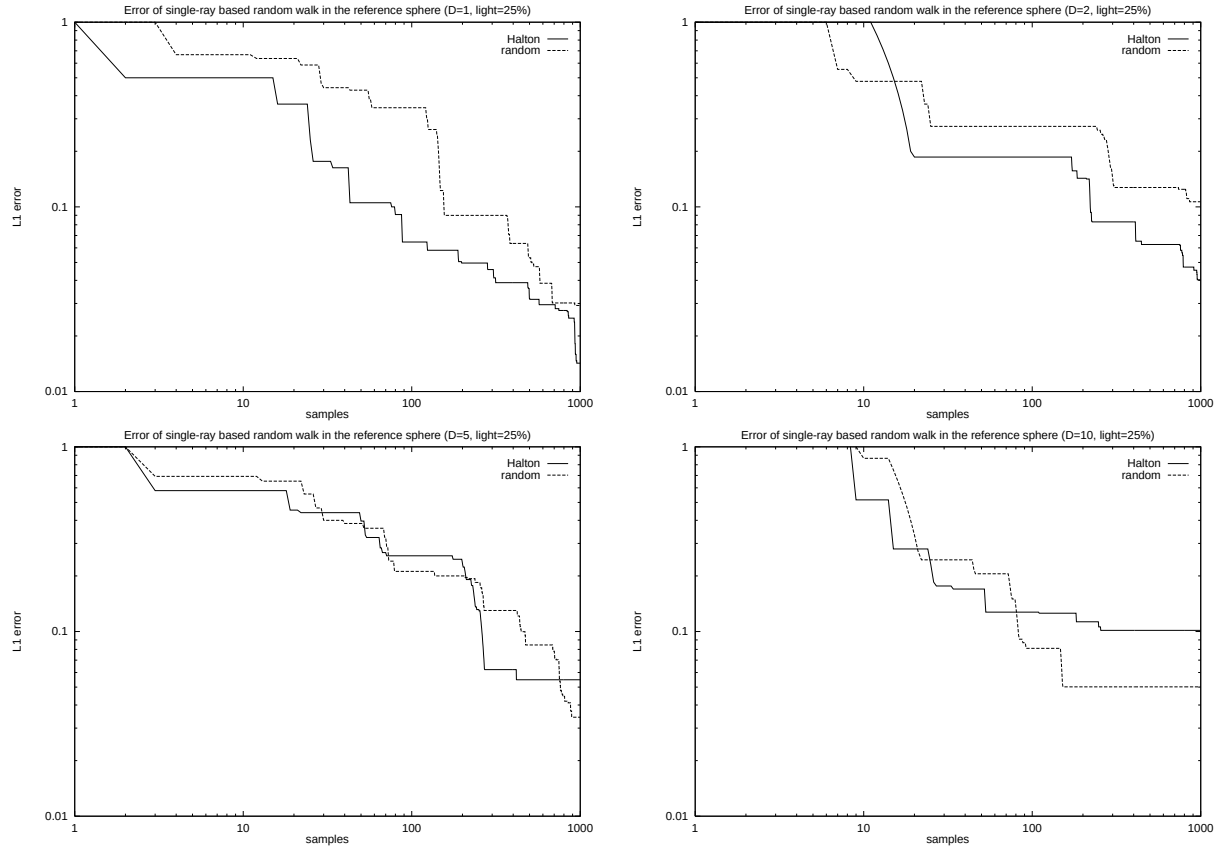

Figure 9: *Error measurements for 1, 2, 5 and 10 bounces*

5.2. Error measurements

The efficiency of Monte-Carlo and quasi-Monte Carlo quadratures have been tested for the presented spherical scene assuming a single pixel camera. The error has been measured separately for the different bounces.

Looking at the error measurements of figure 9, we can see that even for integrands of infinite variation quasi-Monte Carlo methods are still better but they lose their advantage when computing higher bounces as predicted by the theoretical results. The other important problem in higher dimensions is that although a low-discrepancy series has almost linearly decreasing discrepancy in the asymptotic sense, this discrepancy can still be high for not very many points (in the solution of the rendering equation we rarely use more than 1000 samples for the estimation of a single pixel). In the case of the Halton series, for example, the *base* of the series strongly affects the initial behavior of the discrepancy. These base numbers are different prime numbers for different dimensions, thus for high-dimensional integrals the base numbers can be quite high, which results in degraded performance.

6. Conclusions

This paper reviewed the application of quasi-Monte Carlo integration to solve the rendering equation. It showed that although the integrand of the rendering equation is not of finite variation, an error analysis is still possible that combines the Koksma-Hlawka inequality with the probabilistic bounds of Monte-Carlo techniques. This error analysis leads us to the conclusion that the convergence rate of quasi-Monte Carlo integration can be better than Monte-Carlo quadrature even for the discontinuous rendering equation, especially when the required dimension — that is the average reflectivity of the scene — is not very high. The theoretical results have also been demonstrated by simulation.

7. Acknowledgments

This work has been supported by the National Scientific Research Fund (OTKA), ref.No.: F 015884 and the Austrian-Hungarian Action Fund, ref.No.: 29p4, 32öu9 and 34öu28.

References

1. J. Arvo. Stratified sampling of spherical triangles. In *Computer Graphics (SIGGRAPH '95 Proceedings)*, pages 437–438, 1995.
2. P. Bekaert, L. Neumann, A. Neumann, M. Sbert, and Y. Willems. Hierarchical Monte-Carlo radiosity. In *Rendering Techniques '98*, pages 259–268, 1998.
3. F. Castro, R. Martinez, and M. Sbert. Quasi Monte-Carlo and extended first-shot improvements to the multi-path method. In *Spring Conference on Computer Graphics '98*, pages 91–102, 1998.
4. Michael Cohen and Donald Greenberg. The hemi-cube, a radiosity solution for complex environments. In *Computer Graphics (SIGGRAPH '85 Proceedings)*, pages 31–40, 1985.
5. I. Deák. *Random Number Generators and Simulation.* Akadémia Kiadó, Budapest, 1989.
6. D. P. Dobkin, D. Eppstein, and D. P. Mitchell. Computing the discrepancy with applications to supersampling patterns. *ACM Transactions on Graphics*, 15(4):354–376, 1996.
7. M. Hyben, I. Martisovits, and A. Ferko. Scene complexity for rendering in flatland. In L. Szirmay-Kalos, editor, *Spring Conference on Computer Graphics*, pages 112–120, 1998.
8. A. Keller. A quasi-Monte Carlo algorithm for the global illumination in the radiosity setting. In H. Niederreiter and P. Shiue, editors, *Monte-Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, pages 239–251. Springer, 1995.
9. A. Keller. Quasi-Monte Carlo Radiosity. In X. Pueyo and P. Schröder, editors, *Rendering Techniques '96 (Proc. 7th Eurographics Workshop on Rendering)*, pages 101–110. Springer, 1996.
10. A. Keller. The fast Calculation of Form Factors using Low Discrepancy Sequences. In *Proc. Spring Conference on Computer Graphics (SCCG '96)*, pages 195–204, Bratislava, Slovakia, 1996. Comenius University Press.
11. D.E. Knuth. *The art of computer programming. Volume 2 (Seminumerical algorithms).* Addison-Wesley, Reading, Mass. USA, 1981.
12. D. Mitchell. Ray Tracing and Irregularities of Distribution. In *Rendering Techniques '92 (Proc. 3rd Eurographics Workshop on Rendering)*, pages 61–69, Bristol, UK, 1992.
13. D. P. Mitchell. Consequences of stratified sampling in graphics. *Computer Graphics (SIGGRAPH '96 Proceedings)*, pages 277–280, 1996.
14. H. Niederreiter. *Random number generation and quasi-Monte Carlo methods*. SIAM, Pennsilvania, 1992.
15. William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C (Second Edition)*. Cambridge University Press, Cambridge, USA, 1992.
16. P. Shirley. Discrepancy as a quality measure for sampling distributions. In *Eurographics '91*, pages 183–194. Elsevier Science Publishers, 1991.
17. P. Shirley. Time complexity of Monte-Carlo radiosity. In *Eurographics '91*, pages 459–466. Elsevier Science Publishers, 1991.
18. I. Sobol. *Die Monte-Carlo Methode.* Deutscher Verlag der Wissenschaften, 1991.
19. L. Szirmay-Kalos, T. Fóris, L. Neumann, and B. Csébfalvi. An analysis to quasi-Monte Carlo integration applied to the transillumination radiosity method. *Computer Graphics Forum (Eurographics'97)*, 16(3):271–281, 1997.
20. T. Warnock. Computational investigations of low-discrepancy point sets. In H. Niederreiter and P. Shiue, editors, *Monte-Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, pages 354–361. Springer, 1995.