Adaptive Sampling for Global Illumination Rendering

László Szécsi
TU Budapest, Hungary
szecsi@iit.bme.hu

László Szirmay-Kalos
TU Budapest, Hungary
szirmay@iit.bme.hu

Murat Kurt
International Computer Institute, Ege University, Izmir, Turkey
murat.kurt@ege.edu.tr

Balázs Csébfalvi
TU Budapest, Hungary
cseb@iit.bme.hu

Contents

Abstracts:

This paper proposes an adaptive sampling algorithm Monte Carlo integration and thus global illumination rendering. Unlike importance sampling, adaptive sampling does not try to mimic the integrand with analytically integrable and invertible densities, but approximates the integrand with analytically integrable functions, thus it is more appropriate for complex integrands, which correspond to difficult lighting conditions and sophisticated BRDF models. We develop an adaptation scheme that is based on ray differentials and does not require neighbor finding and complex data structures in higher dimensions. As a side result of the adaptation criterion, the algorithm also provides error estimates, which are essential in predictive rendering applications. We conclude that although the method attacks the curse of dimension, it is particularly effective when the dimension of the integration domain is moderate. Thus, we propose it for environment mapping computations.

1. Introduction

Rendering is mathematically equivalent to the evaluation of high-dimensional integrals. In order to avoid the curse of dimensionality, Monte Carlo or quasi-Monte Carlo quadratures are applied, which take discrete samples and approximate the integral as the weighted sum of integrand values in these samples. The challenge of these methods is to find a sampling strategy that generates a small number of samples providing an accurate quadrature.

Importance sampling would mimic the integrand with the density of samples. However, finding a proper density function and generating samples with its distribution are non-trivial tasks. There are two fundamentally different approaches for generating samples with a probability density. The *inversion method* maps uniformly distributed samples with the inverse of cumulative probability distribution of desired density. However, this requires that the density is integrable and its integral is analytically invertible. The integrand of rendering equation is a product of BRDF and cosine factor of multiple reflection, and of the incident radiance at the end of a path. Due to the imposed requirements, importance sampling can take into account only a single factor of this product form integral, and even the single factor is just approximately mimicked except for some very simple BRDF models. *Rejection sampling* based techniques, on the other hand, do not impose such requirements on the density. However, rejection sampling may ignore many, already generated samples, which may lead to unpredictable performance degradation. Rejection sampling inspired many sampling methods [1; 2; 3; 4; 5; 6]. We note that in the context of Metropolis sampling there have been proposals to exploit even the rejected samples having reweighted them [7; 8; 9; 10]. *Hierarchical sampling* strategies attack product form integrands by mimicking just one factor initially, then improving the sample distribution in the second step either by making the sampling distribution more proportional to the integrand [3; 4; 11; 12; 5] or by making the empirical distribution of samples closer to the continuous sample distribution [1; 2; 6].

An alternative method of finding samples for integral quadratures is *adaptive sampling* that uses the samples to define an approximation of the integrand, which is then analytically integrated. Adaptation is guided by the statistical analysis of samples in the subdomains. Should the variance of integrand in a subdomain be high, the subdomain is broken down to smaller subdomains. Adaptive sampling does not pay attention to the placement of samples in the subdomains. However, placing samples in a subdomain without considering the distribution of samples in neighboring subdomains will result in very uneven distribution in higher dimensional spaces. On the other hand, variance calculation needs neighbors finding, which gets also more difficult in higher dimensions. Thus, adaptive sampling usually suffers from the curse of dimensionality.

The *VEGAS* method [13] is an adaptive Monte Carlo technique that generates a probability density for importance sampling automatically and in a separable form. The reason of requirement of separability is that probability densities are computed from and stored in histogram tables and s number of one-dimensional (1D) tables need much less storage space than a single s-dimensional table.

Adaptive sampling methods have been less popular in rendering than importance sampling mainly due to the fact that the storage requirements of higher dimensional tables grow exponentially with the dimension of the domain, i.e. the curse of dimension prohibits these techniques to be applied in general rendering algorithms.

This paper proposes a multidimensional adaptive sampling scheme, which also addresses the curse of dimension and exploits low-discrepancy sequences. Low-discrepancy sequences fill higher dimensional spaces better than regular grids and their discrepancy is just moderately dependent on the dimension of the domain. The main idea of our approach is to implicitly encode tables by low-discrepancy sequences, thus to avoid exponential explosion in higher dimensions. The resulting algorithm:

- has small additional overhead both in terms of storage and computation, because it does not require complex data structures or neighborhood searches as other adaptive algorithms,
- provides consistent, deterministic estimates with small and predictable quadrature error (the term unbiasedness is not applicable for deterministic sampling algorithms).

This paper is the extended version of [32]. In this current version we examine the mathematical problem more generally and show that it can be used in higher dimensional problems as well and include additional results.

2. Previous Work

This paper combines different methods popular in computer graphics, thus its previous work roots in different fields.

Adaptive sampling: Adaptive sampling uses samples to approximate the integrand, thus concentrates samples where the integrand changes significantly. The targeted approximation is usually piecewise constant or piecewise linear [14]. Image space adaptive sampling was used even in the first ray tracer [15], and has been improved by randomization [16] and by the application of information theory tools [17]. Adaptive sampling methods should decompose the sampling domain, which is straightforward in two-dimensional (2D) but needs special data structures like the kd-tree in higher dimensions [18].

Ray and path differentials: A ray differential means the derivative of contribution of a path with respect to some variable defining this path. Igehy introduced ray differentials with respect to screen space coordinates [19]. Ray differentials were used in filtering applications where the differentials define the footprint of samples, i.e. the domain the filter. Suykens and Willems [20] generalized this idea for arbitrary sampling parameters defining the path. *Photon differentials* [21] also use this concept to enhance radiance estimates in photon mapping. We note that unlike these techniques, we do not use path differentials in this paper for setting up filters, but to guide the sampling process.

Error estimation in rendering: In predictive rendering we need not only an estimate of the image, but also error bounds describing the accuracy of method [22]. Unfortunately, the computation of error of a rendering algorithm is even more complex than producing the image [23]. Error analysis was proposed in the context of finite-element based radiosity methods [24; 25; 16; 26]. For Monte Carlo methods, we can use the statistics of samples to estimate the variance of estimator. For deterministic approaches, in theory, the Koksma-Hlawka inequality could be used. However, due to the infinite variation of typical integrands in rendering, this provides useful bounds only in exceptional cases [27].

3. Proposed Sampling Scheme

For the sake of notational simplicity, let us consider first a 1D integral of a possibly vector valued integrand $f(t)$ in the domain of $[0,1]$. Suppose that the integration domain is decomposed to intervals $\Delta^{(1)},\Delta^{(2)},\ldots,\Delta^{(M)}$ and one sample point $t^{(i)}$ is selected in each of them. The integral is estimated in the following way:

$$\int_0^1 f(t)\mathrm{d}t \approx \sum_{i=1}^{M} f(t^{(i)})\Delta^{(i)}. \tag{1}$$

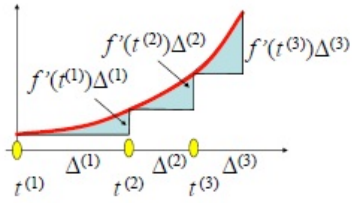

Figure 1: The error of adaptive sampling is the total area of triangles of bases $\Delta^{(i)}$ and heights that are proportional to $\Delta^{(i)}$ and to the absolute value of derivative of integrand f.

Looking at Figure 1, we can conclude that the error of integral is the sum of small triangles in between the integrand and its piecewise constant approximation, which can be expressed as follows if derivative f' exists:

$$\left| \int_0^1 f(t)\mathrm{d}t - \sum_{i=1}^{M} f(t^{(i)})\Delta^{(i)} \right| \approx \sum_{i=1}^{M} |f'(t^{(i)})| \frac{\left(\Delta^{(i)}\right)^2}{2}. \tag{2}$$

We minimize the error with the constraint $\sum_{i=1}^{M} \Delta^{(i)} = 1$ using the Lagrange multiplier method. The target function that also includes the constraint scaled by λ is

$$\sum_{i=1}^{M} |f'(t^{(i)})| \frac{\left(\Delta^{(i)}\right)^2}{2} - \lambda \left(\sum_{i=1}^{M} \Delta^{(i)} - 1 \right).$$

Computing the partial derivatives with respect to $\Delta^{(i)}$ and making it equal to zero, we obtain:

$$|f'(t^{(i)})|\Delta^{(i)} = \lambda, \quad i = 1, 2, \ldots, M,$$

thus, the variation in all subdomains should be equal, i.e. the sampling scheme should concentrate samples where the integrand changes quickly. Note that this requirement is orthogonal to the requirement of importance sampling that places samples where the integrand is large.

3.1 Extension to Higher dimensional Integrands

In rendering high dimensional integrands need to evaluated, where every light bounce increases the dimension of the domain by 2. In the case of environment mapping, the integrand is only two dimensional. For the sake of notational simplicity, when presenting the generalization of the adaptive sampling scheme, we assume that the integrand is two dimensional, but the same approach can also be used in higher dimensions as well. The high dimensional domain should be transformed to a unit cube, which in two dimensions is a unit square $\mathscr{U}$. A point in the square is denoted by $\mathbf{u} = (u_1, u_2)$. The integrand can only be point sampled, but together with point sampling, its derivative can also be obtained according to the concept of *ray differentials*.

Adaptive sampling should decompose the integration domain, i.e. the unit square to M subdomains. Subdomain i has edge lengths $\Delta_1^{(i)}, \Delta_2^{(i)}$ and area

$$\Delta^{(i)} = \Delta_1^{(i)} \cdot \Delta_2^{(i)}.$$

Each subdomain contains a single sample $\mathbf{u}^{(i)}$ from which the integral is estimated as:

$$\int_{\mathscr{U}} F(\mathbf{u})\mathrm{d}\mathbf{u} \approx \sum_{i=1}^{M} F(\mathbf{u}^{(i)})\Delta^{(i)}. \tag{3}$$

In order to find the error of this 2D integral, we approximate the integrand by the first-order Taylor series in each subdomain:

$$F(\mathbf{u}) \approx F(\mathbf{u}^{(i)}) + \frac{\partial F(\mathbf{u}^{(i)})}{\partial u_1}(u_1 - u_1^{(i)}) + \frac{\partial F(\mathbf{u}^{(i)})}{\partial u_2}(u_2 - u_2^{(i)}).$$

Substituting this approximation into the quadrature error, we get:

$$\left| \int_{\mathscr{U}} F(\mathbf{u})\mathrm{d}\mathbf{u} - \sum_{i=1}^{M} F(\mathbf{u}^{(i)})\Delta^{(i)} \right| \approx$$

$$\sum_{i=1}^{M} \left[\left| \frac{\partial F(\mathbf{u}^{(i)})}{\partial u_1} \right| \frac{\Delta_1^{(i)}}{2} \Delta^{(i)} + \left| \frac{\partial F(\mathbf{u}^{(i)})}{\partial u_2} \right| \frac{\Delta_2^{(i)}}{2} \Delta^{(i)} \right]. \tag{4}$$

We minimize this error by finding the subdomain sizes $\Delta_d^{(i)}$ satisfying the constraints that subdomains are disjunct and their union is the original unit square.

If subdomain $\Delta^{(i)}$ is subdivided along axis d, the error is reduced proportionally by

$$\varepsilon_d^{(i)} = \left| \frac{\partial F(\mathbf{u}^{(i)})}{\partial u_d} \right| \frac{\Delta_d^{(i)}}{2} \Delta^{(i)}.$$

To maximize the error reduction, that subdomain and dimension should be selected for subdivision where this product is maximum. As each subdomain stores a single sample, the reduced error can also be assigned to the samples. Samples are generated one-by-one. When we are at sample $\mathbf{u}^{(i)}$, the integrand as well its partial derivatives are computed, and each sample is given the vector of possible error reductions:

$$\boldsymbol{\varepsilon}^{(i)} = \left[\varepsilon_1^{(i)}, \varepsilon_2^{(i)} \right].$$

The maximum error reduction with respect to all samples and dimensions tells us which subdomain should be subdivided and also specifies the axis of subdivision. The subdomain is subdivided generating new samples in the neighborhood of sample associated with the subdivided cell.

3.2 Sample Generation and Subdomain Subdivision

In order to explore the integration domain and establish the subdivision scheme, we take a low-discrepancy series [28; 29; 30]. We work with the Halton series, but other low discrepancy series, such as the Sobol series or (t,m,s)-nets with $t=0$ could be used as well [31]. For the 1D Halton sequence in base B, the elemental interval property means that the sequence will visit all intervals of length $[kB^{-R}, (k+1)B^{-R})$ where exponent R determines the length of interval, before putting a new point in an interval already visited. As the sequence is generated, the algorithm produces a single point in each interval of length B^{-1}, then in each interval of length B^{-2}, etc.

For a 2D sequence of relative prime bases B_1, B_2, this property means that the sequence will insert a sample in each cell

$$[k_1 B_1^{-R_1}, (k_1+1)B_1^{-R_1}) \times [k_2 B_2^{-R_2}, (k_2+1)B_2^{-R_2}),$$

where R_1 and R_2 define the horizontal and vertical sizes of cell, before placing a second sample in any of these cells. The elemental interval property thus implicitly defines an automatically refined subdomain structure, which can also be exploited in adaptive sampling. If we decide to subdivide the subdomain of sample i along coordinate axis d, then the following new samples must be generated:

$$i + B_1^{R_1} B_2^{R_2},\ i + 2B_1^{R_1} B_2^{R_2},\ \ldots,\ (B_d - 1) B_1^{R_1} B_2^{R_2}.$$

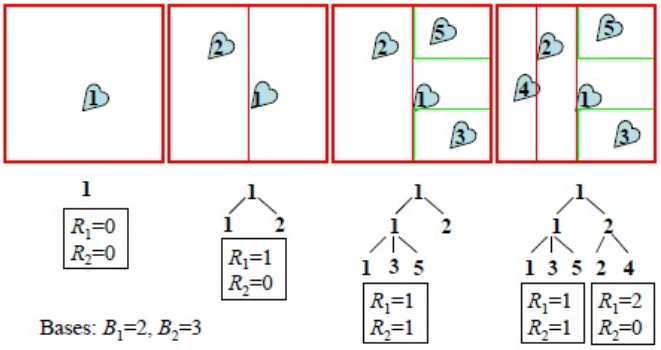

Figure 2: Evolution of samples and their corresponding subdomains in 2D with a Halton sequence of bases $B_1 = 2$ and $B_2 = 3$.

In Figure 2, we show the evolution of cell structure in 2D as new samples are introduced. First, we have a single sample $j=1$ in a cell of volume 1 ($R_1 = 0, R_2 = 0$). Then, this cell is subdivided along the first axis where the basis is $B_1 = 2$, generating a new sample $j = 1 + B_1^{R_1} B_2^{R_2} = 2$. Simultaneously, the size of cell is reduced to half ($R_1 = 1, R_2 = 0$). In the third step, the cell of sample $j=1$ is subdivided again, but now along the second axis. This happens by producing new samples $j = 1 + B_1^{R_1} B_2^{R_2} = 3$ and $j = 1 + 2B_1^{R_1} B_2^{R_2} = 5$, which results in cell sizes as ($R_1 = 1, R_2 = 1$). Finally the cell of sample $j=2$ is subdivided into two parts.

The sampling process can be visualized as a tree (see Figure 2) where upper level nodes are the subdivided subdomains, and leaves are the cells containing exactly one sample. The number of children of a node equals to the corresponding prime basis number of Halton sequence. Although only leaves are relevant for the approximation, it is worth maintaining the complete tree structure since it also helps to find that leaf which should be subdivided to reduce the error most effectively. In upper level nodes, we maintain the maximum error of its children.

The generation of a new sample is the traversal of this tree. When traversing a node, we continue descending into the direction of child with maximum error. When we arrive at a leaf, the dimension which reduces the error the most is subdivided and the leaf becomes a node with children including the original sample and the new samples. While ascending on the tree, the maximum error of new nodes are propagated.

The recursive function implementing this scheme gets a node c. Calling this function with the root node, it generates a new sample that reduces the integration error the most, and updates the complete tree as well:

```
Process( c )
        if ( c is a leaf)
                i = sample id of this leaf
                (R1, R2) = subdivision levels of this node
                (ε1, ε2) = errors of this node
                Find the dimension d of the highest error ε_d
                for k = 1 to B_d − 1 do
                        Generate sample with id j = i + k B_1^R1 B_2^R2
                        Compute F(u^(j)) and the partial derivatives
                        Add sample j as child to node c
                endfor
                Update subdivision levels in sample i
                Update error in sample i
                Add sample i as child to node c
                ε^(c) = maximum error of the children
        else
                Find the child node c_first with maximum error
                Find the second largest error ε_second
                ε^(c) = max( Process( c_first ), ε_second )
        endif
        return ε^(c)
end
```

3.3 Robustness Issues

So far we assumed that the derivative exists and is an appropriate measure for the change of function in a subdomain. This is untrue where the function has a discontinuity or when the subdomain is large. A *consistent estimator* should decompose a subdomain even if the derivative is zero at the examined sample when the number of samples goes to infinity. To obtain this behavior, a positive constant is added to the derivative before the error is computed. This constant is reduced as size of the subdomain gets smaller, corresponding to the fact that for smaller subdomains the derivative becomes a more reliable measure of the change of a function.

Note also that the derivative does not exist where the integrand is discontinuous. This is typical when the integrand includes the visibility factor. Ignoring the visibility factor in the derivative computation, we can still apply the method since we use the derivatives only to decide the "optimal" decomposition of integration domain, and not directly for the estimation of integral. If the decomposition is not "optimal" due to the ignored factors, the method will still converge to the real value, but the convergence will be slower.

The proposed method uses the Halton sequence which can fill D-dimensional spaces uniformly with $O(\log^D N/N)$ discrepancy. However, we cannot claim that our method remains similarly stable in very high dimensions. The problem is the index generation needed to find additional samples in the neighborhood of a given sample, can result in fast growing sequences [31]. So we propose application fields where the dimension of integration domain is moderate. Such an application having particular importance in rendering is environment mapping.

4. Application to Environment Mapping

Environment mapping [32] computes the reflected radiance of point $\vec{x}$ as

$$L(\vec{x},\vec{\omega}) = \int_{\Omega} L^{\text{env}}(\vec{\omega}') f_r(\vec{\omega}',\vec{x},\vec{\omega}) \cos\theta'_{\vec{x}} v(\vec{x},\vec{\omega}') \mathrm{d}\omega',$$

where Ω is the set of all directions, $L^{\text{env}}(\vec{\omega}')$ is the radiance of environment map at *illumination direction* $\vec{\omega}'$, f_r is the BRDF, $\theta'_{\vec{x}}$ is the angle between the illumination direction and the surface normal at shaded point $\vec{x}$, and $v(\vec{x},\vec{\omega}')$ is the indicator function checking whether no virtual object is seen from $\vec{x}$ at direction $\vec{\omega}'$, so environment lighting can take effect. We suppose that the BRDF and the environment lighting are available in analytic forms.

In order to transform the integration domain, i.e. the set of illumination directions, to a unit square $u_1, u_2 \in [0,1]$, we find a parametrization $\vec{\omega}'(u_1,u_2)$. We shall consider different options for this parametrization, including a global parametrization that is independent of the surface normal of shaded point, and BRDF based parameterizations where the Jacobi determinant of mapping compensates the variation of BRDF and cosine factor.

The integral of reflected radiance can also be evaluated in parameter space:

$$L(\vec{x},\vec{\omega}) = \int_0^1 \int_0^1 L^{\text{env}}(\vec{\omega}'(u_1,u_2)) R(u_1,u_2) v(\vec{x},\vec{\omega}'(u_1,u_2)) \mathrm{d}u_1 \mathrm{d}u_2,$$

where

$$R(u_1,u_2) = f_r(\vec{\omega}'(u_1,u_2),\vec{x},\vec{\omega}) \cos\theta'_{\vec{x}}(u_1,u_2) \frac{\partial\omega}{\partial u_1 \partial u_2}$$

is the *reflection factor* defined by the product of BRDF, the cosine of angle between the surface normal and the incident direction, and also the Jacobi determinant of parametrization.

The derivatives of integrand $F = L^{\text{env}} \cdot R \cdot v$ of environment mapping are ($d = 1,2$):

$$\frac{\partial F}{\partial u_d} = \frac{\partial L^{\text{env}}}{\partial u_d} R v + L^{\text{env}} \frac{\partial R}{\partial u_d} v.$$

Thus, we need to evaluate the derivative of reflection factor and of environment lighting. The visibility factor is piecewise constant, so its derivative would be zero.

4.1 Global Parametrization

In the case of *global parametrization* the unit square is mapped to the set of directions independently of the local orientation of surface. The incident direction is obtained in spherical coordinates $\phi = 2\pi u_1$, $\theta = \pi u_2$, which are converted to Cartesian coordinates of the system defined by basis vectors $\vec{i}, \vec{j}, \vec{k}$:

$$\vec{\omega}'(u_1, u_2) =$$

$$\vec{i}\cos(2\pi u_1)\sin(\pi u_2) + \vec{j}\sin(2\pi u_1)\sin(\pi u_2) + \vec{k}\cos(\pi u_2).$$

The derivatives of incident direction are as follows:

$$\frac{\partial \vec{\omega}'}{\partial u_1} = -2\pi\vec{i}\sin(2\pi u_1)\sin(\pi u_2) + 2\pi\vec{j}\cos(2\pi u_1)\sin(\pi u_2),$$

$$\frac{\partial \vec{\omega}'}{\partial u_2} = \pi\vec{i}\cos(2\pi u_1)\cos(\pi u_2) + \pi\vec{j}\sin(2\pi u_1)\cos(\pi u_2) - \pi\vec{k}\sin(\pi u_2). \tag{5}$$

The Jacobi determinant of global parametrization is

$$\frac{\partial \omega}{\partial u_1 \partial u_2} = 2\pi^2 \sin\theta = 2\pi^2 \sin(\pi u_2).$$

4.2 Diffuse BRDF Parametrization

We can also use the parametrization developed for BRDF sampling. The illumination direction is generated in the *tangent space* of shaded point $\vec{x}$, defined by surface normal $\vec{N}_{\vec{x}}$, arbitrary tangent $\vec{T}_{\vec{x}}$ that is perpendicular to the normal, and binormal $\vec{B}_{\vec{x}}$ that is perpendicular both to the normal and the tangent, and it should mimic the cosine distribution:

$$\vec{\omega}' = \vec{T}_{\vec{x}}\cos(2\pi u_1)\sqrt{u_2} + \vec{B}_{\vec{x}}\sin(2\pi u_1)\sqrt{u_2} + \vec{N}_{\vec{x}}\sqrt{1-u_2}.$$

The derivatives are:

$$\frac{\partial \vec{\omega}'}{\partial u_1} = -2\pi\vec{T}_{\vec{x}}\sin(2\pi u_1)\sqrt{u_2} + 2\pi\vec{B}_{\vec{x}}\cos(2\pi u_1)\sqrt{u_2},$$

$$\frac{\partial \vec{\omega}'}{\partial u_2} = \vec{T}_{\vec{x}}\frac{\cos(2\pi u_1)}{2\sqrt{u_2}} + \vec{B}_{\vec{x}}\frac{\sin(2\pi u_1)}{2\sqrt{u_2}} - \vec{N}_{\vec{x}}\frac{1}{2\sqrt{1-u_2}}. \tag{6}$$

The Jacobi determinant compensates the cosine term:

$$\frac{\partial \omega}{\partial u_1 \partial u_2} = \frac{\pi}{\cos\theta_{\vec{x}}} = \frac{\pi}{\sqrt{1-u_2}}.$$

4.3 Specular BRDF Parametrization

In case of a Phong BRDF, the direction is generated in *reflection space* defined by the reflection direction

$$\vec{\omega}_r = 2\vec{N}_{\vec{x}}(\vec{N}_{\vec{x}} \cdot \vec{\omega}) - \vec{\omega}$$

and two other orthonormal vectors $\vec{T}_r$ and $\vec{B}_r$ that are perpendicular to $\vec{\omega}_r$. The mapping should follow the $(\vec{\omega}_r \cdot \vec{\omega}')^n$ function:

$$\vec{\omega}'(u_1, u_2) = \vec{T}_r\cos(2\pi u_1)\sqrt{1-(1-u_2)^{\frac{2}{n+1}}} +$$

$$\vec{B}_r\sin(2\pi u_1)\sqrt{1-(1-u_2)^{\frac{2}{n+1}}} + \vec{\omega}_r(1-u_2)^{\frac{1}{n+1}}.$$

The derivatives are obtained as:

$$\frac{\partial \vec{\omega}'}{\partial u_1} = -2\pi \vec{T}_r \sin(2\pi u_1)\sqrt{1-(1-u_2)^{\frac{2}{n+1}}} +$$

$$2\pi \vec{B}_r \cos(2\pi u_1)\sqrt{1-(1-u_2)^{\frac{2}{n+1}}},$$

$$\frac{\partial \vec{\omega}'}{\partial u_2} = \vec{T}_r \cos(2\pi u_1)\frac{(1-u_2)^{\frac{1-n}{n+1}}}{(n+1)\sqrt{1-(1-u_2)^{\frac{2}{n+1}}}} +$$

$$\vec{B}_r \sin(2\pi u_1)\frac{(1-u_2)^{\frac{1-n}{n+1}}}{(n+1)\sqrt{1-(1-u_2)^{\frac{2}{n+1}}}} - \vec{\omega}_r \frac{1}{(n+1)(1-u_2)^{\frac{n}{n+1}}}. \tag{7}$$

The Jacobi determinant of mapping is

$$\frac{\partial \omega}{\partial u_1 \partial u_2} = \frac{2\pi}{(n+1)(\vec{\omega}_r \cdot \vec{\omega}')^n} = \frac{2\pi}{(n+1)(1-u_2)^{\frac{n}{n+1}}}.$$

4.4 Derivatives of the Reflection Factor

The reflection factor includes the Jacobi determinant of mapping, so it must be discussed taking into account the parametrization. If we use diffuse BRDF parametrization for a diffuse surface, the Jacobi determinant will compensate the cosine term, making the reflection factor constant, and consequently its derivative equal to zero. Similarly, the specular BRDF parametrization of a specular surface also results in a zero derivative. For more complex BRDFs that have no exact importance sampling, we can still use BRDF parametrization of a similar BRDF, for example, that of the diffuse or the Phong solutions. Of course, their Jacobi determinants will not fully eliminate the BRDF and the cosine term, so we have some non-constant function that needs to be derived.

Let us now consider global parametrization, where the Jacobi determinant is $2\pi^2 \sin(\pi u_2)$. If the material is diffuse with diffuse reflectivity k_{dif}, and the surface normal at the shaded point $\vec{x}$ is $\vec{N}_{\vec{x}}$, the partial derivatives of reflection factor including the BRDF, the cosine term, and the Jacobi determinant are ($d = 1,2$):

$$\frac{\partial R}{\partial u_1} = 2\pi^2 k_{\text{dif}} \left(\frac{\partial \vec{\omega}'}{\partial u_1} \cdot \vec{N}_{\vec{x}}\right) \sin(\pi u_2),$$

$$\frac{\partial R}{\partial u_2} = 2\pi^2 k_{\text{dif}} \left(\left(\frac{\partial \vec{\omega}'}{\partial u_2} \cdot \vec{N}_{\vec{x}}\right) \sin(\pi u_2) + \pi(\vec{\omega}' \cdot \vec{N}_{\vec{x}}) \cos(\pi u_2)\right),$$

if $\vec{\omega}' \cdot \vec{N}_{\vec{x}} \geq 0$ and zero otherwise. The derivatives of incident direction are given in Equation (5).

The derivatives can also be computed for specular BRDFs, for example, for the Phong BRDF:

$$R = 2\pi^2 k_{\text{spec}} (\vec{\omega}' \cdot \vec{\omega}_r)^n \sin(\pi u_2),$$

where k_{spec} is the specular reflectivity. Applying the rules of derivation systematically, we get:

$$\frac{\partial R}{\partial u_1} = 2\pi^2 n k_{\text{spec}} (\vec{\omega}' \cdot \vec{\omega}_r)^{n-1} \left(\frac{\partial \vec{\omega}'}{\partial u_1} \cdot \vec{\omega}_r\right) \sin(\pi u_2),$$

$$\frac{\partial R}{\partial u_2} = 2\pi^2 n k_{\text{spec}} (\vec{\omega}' \cdot \vec{\omega}_r)^{n-1} \left(\frac{\partial \vec{\omega}'}{\partial u_2} \cdot \vec{\omega}_r\right) \sin(\pi u_2) +$$

$$2\pi^3 k_{\text{spec}} (\vec{\omega}' \cdot \vec{\omega}_r)^n \cos(\pi u_2).$$

4.5 Derivatives of the Environment Lighting

The environment lighting may be defined analytically or by a texture map.

4.5.1 Simple Sky Illumination

For example, a sky illumination with the Sun at direction $\vec{\omega}_{\text{sun}}$ can be expressed as

$$L^{\text{env}}(\vec{\omega}') = L_{\text{sky}} + L_{\text{sun}}(\vec{\omega}' \cdot \vec{\omega}_{\text{sun}})^s,$$

where L_{sky} and L_{sun} are the sky and sun radiances, respectively, and s is the exponent describing the directional fall-off of sun radiance. With this environment radiance, the derivatives of environment lighting are ($d = 1,2$):

$$\frac{\partial L^{\text{env}}}{\partial u_d} = sL_{\text{sun}}(\vec{\omega}' \cdot \vec{\omega}_{\text{sun}})^{s-1}\left(\frac{\partial \vec{\omega}'}{\partial u_d} \cdot \vec{\omega}_{sun}\right).$$

4.5.2 Texture Based Environment Lighting

If the environment lighting is defined by a texture map, we first project it into a spherical harmonics basis [33]:

$$L^{\text{env}}(\vec{\omega}') = \sum_l L_{l0}^{\text{env}} Y_l^0(\cos\theta) +$$

$$\sum_l \sum_{m=-1}^{-l} L_{lm}^{\text{env}} Y_l^m(\cos\theta, \sin(-m\phi)) + \sum_l \sum_{m=1}^{l} L_{lm}^{\text{env}} Y_l^m(\cos\theta, \cos(m\phi)),$$

where Y_l^m are the spherical harmonics basis functions.

Global spherical angles ϕ and θ can be expressed as

$$\cos\theta = z, \quad \cos\phi = \frac{x}{\sqrt{1-z^2}}, \quad \sin\phi = \frac{y}{\sqrt{1-z^2}},$$

where x, y, z are the components of incident direction vector

$$\vec{\omega}'(u_1, u_2) = (x(u_1, u_2), y(u_1, u_2), z(u_1, u_2)).$$

The derivatives of environment requires the derivatives of spherical basis functions. In practice the number of bands l is quite small, thus it is worth precomputing these basis functions in algebraic form by hand.

The first 9 spherical basis functions are obtained as:

$$\begin{aligned}
Y_0^0 &= 0.282095, \\
Y_1^{-1} &= 0.488603y, \\
Y_1^0 &= 0.488603z, \\
Y_1^1 &= 0.488603x, \\
Y_2^{-2} &= 1.092548xy, \\
Y_2^{-1} &= 1.092548yz, \\
Y_2^0 &= 0.315392(3z^2 - 1), \\
Y_2^1 &= 1.092548xz, \\
Y_2^2 &= 0.546274(x^2 - y^2).
\end{aligned}$$

The derivatives with respect to u_d ($d = 1,2$) of the basis functions can be obtained by a direct derivation of the formulae:

$$\begin{aligned}
\frac{\partial Y_0^0}{\partial u_d} &= 0, \\
\frac{\partial Y_1^{-1}}{\partial u_d} &= 0.488603\frac{\partial y}{\partial u_d}, \\
\frac{\partial Y_1^0}{\partial u_d} &= 0.488603\frac{\partial z}{\partial u_d}, \\
\frac{\partial Y_1^1}{\partial u_d} &= 0.488603\frac{\partial x}{\partial u_d}, \\
\frac{\partial Y_2^{-2}}{\partial u_d} &= 1.092548\left(\frac{\partial x}{\partial u_d}y + x\frac{\partial y}{\partial u_d}\right), \\
\frac{\partial Y_2^{-1}}{\partial u_d} &= 1.092548\left(\frac{\partial y}{\partial u_d}z + y\frac{\partial z}{\partial u_d}\right),
\end{aligned}$$

$$\frac{\partial Y_2^0}{\partial u_d} = 0.315392\left(6z\frac{\partial z}{\partial u_d}\right),$$
$$\frac{\partial Y_2^1}{\partial u_d} = 1.092548\left(\frac{\partial x}{\partial u_d}z + x\frac{\partial z}{\partial u_d}\right),$$
$$\frac{\partial Y_2^2}{\partial u_d} = 0.546274\left(2x\frac{\partial x}{\partial u_d} - 2y\frac{\partial y}{\partial u_d}\right).$$

5. Results

The environment mapping algorithm has been implemented using the Direct3D 9 framework and run on an nVidia GeForce 8800 GFX GPU.

In environment mapping, adaptive sampling generates points in a unit square according to the product form integrand

$$L^{\text{env}}(\vec{\omega}'(u_1,u_2))f_r(\vec{\omega}'(u_1,u_2),\vec{x},\vec{\omega})\cos\theta'_{\vec{x}}(u_1,u_2)\frac{\partial\omega}{\partial u_1\partial u_2},$$

that includes the environment illumination, the cosine weighted BRDF and the Jacobi determinant of mapping.

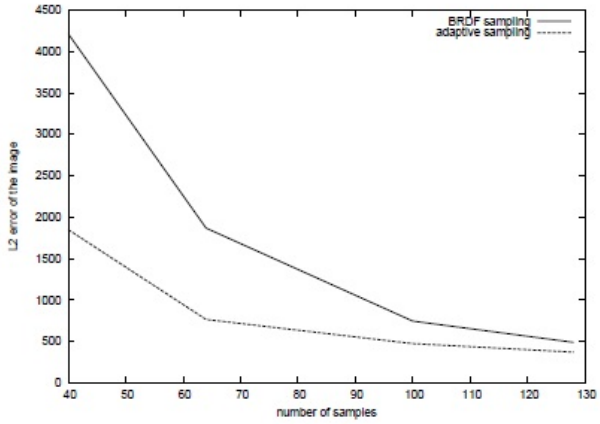

Figure 3: L^2 error of adaptive sampling with diffuse BRDF parametrization compared to classic BRDF sampling with respect to the number of samples per pixel.

The L^2 errors of images rendered with adaptive and classic BRDF sampling are shown in Figure 3 as a function of the samples per pixel. Adaptive sampling roughly halves the error for the price of a ten percent higher computation cost.

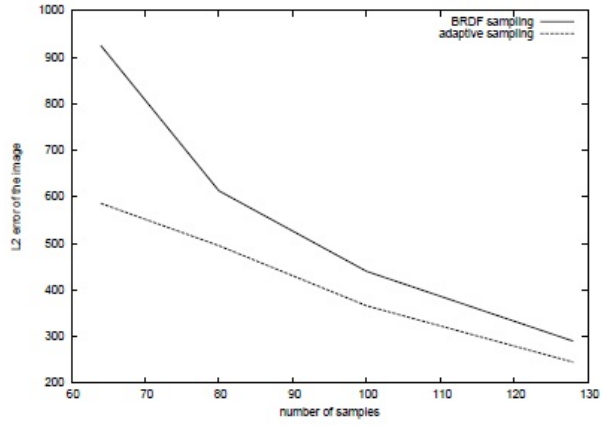

Figure 4: L^2 error of adaptive sampling with Phong BRDF parametrization compared to classic BRDF sampling with respect to the number of samples per pixel.

The L^2 image error for specular surface rendering is shown in Figure 4.

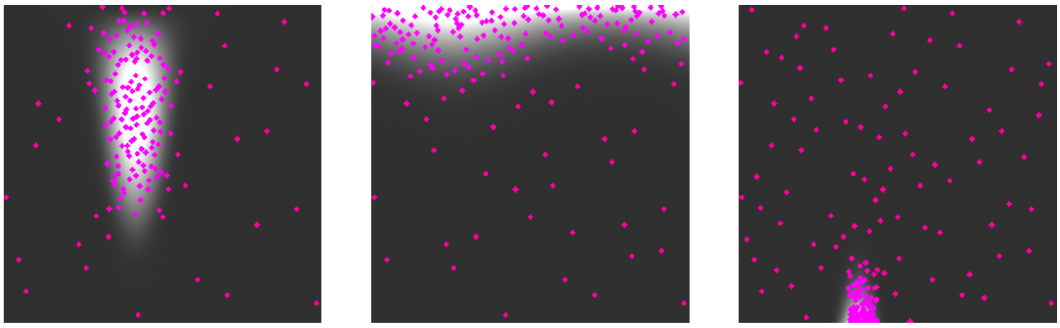

Figure 5: Samples generated with BRDF parametrization displayed over the transformed integrand

In Figure 5 we depicted the samples assigned to a single shaded point. Samples and the transformed integrand are shown in parameter space. Note that the proposed adaptive sampling scheme places samples at regions where the illumination changes quickly and also takes care of the stratification of the samples. Note that samples are well stratified and are focused at regions where the environment lighting changes quickly.

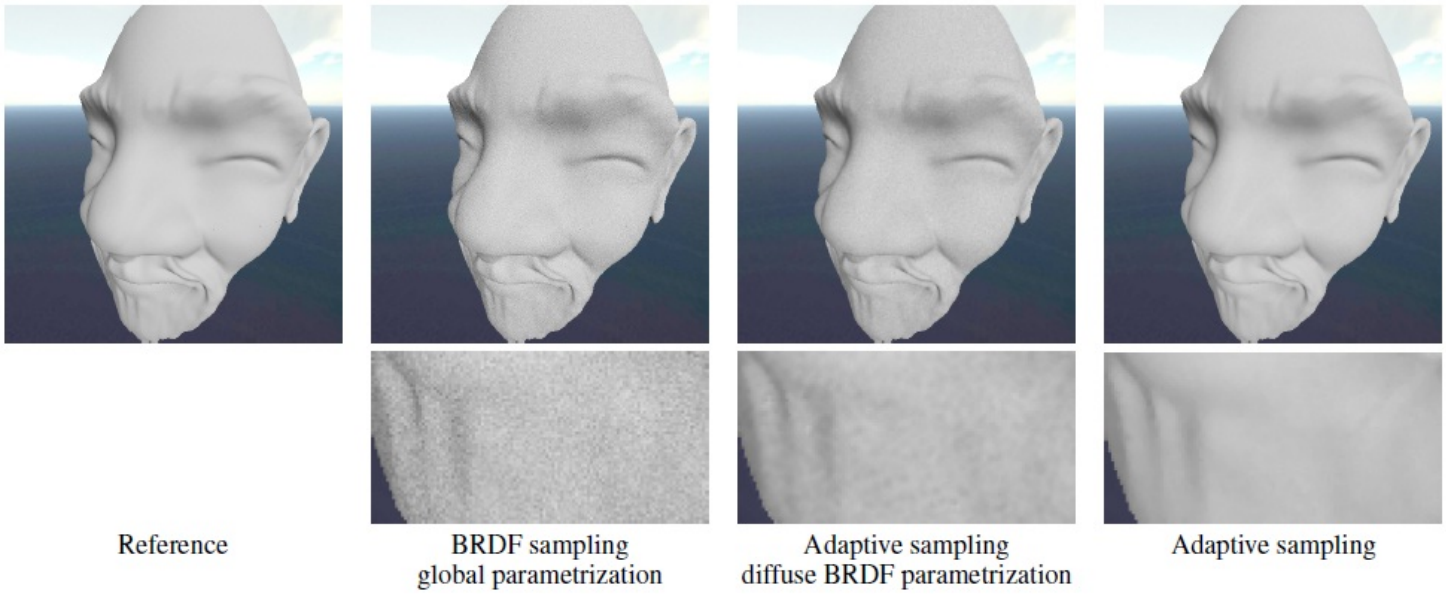

Figure 6: Diffuse objects rendered with BRDF sampling, adaptive sampling with global parametrization, and adaptive sampling with diffuse BRDF parametrization. The lower images are zoom-ins of Ming's chin.

In Figure 6 we compare the adaptive sampling scheme involving global parametrization and diffuse BRDF parametrization to classical BRDF sampling. The dynamic range of environment map is 1 to 320. We used the Halton sequence with random initial index in BRDF sampling and in adaptive sampling as well, thus the low-discrepancy properties of samples have been exploited both in the classical and the new methods. All methods were given the same computation time (50 sec), while we traced 100 rays per pixel with BRDF sampling and 90 rays with adaptive sampling. The ten percent reduction of number of samples is due to the overhead of sample management in adaptive sampling. The proposed adaptive scheme has significantly reduced the noise. BRDF parametrization is better than global parametrization as it automatically ignores that part of the integration domain where the cosine term would be zero.

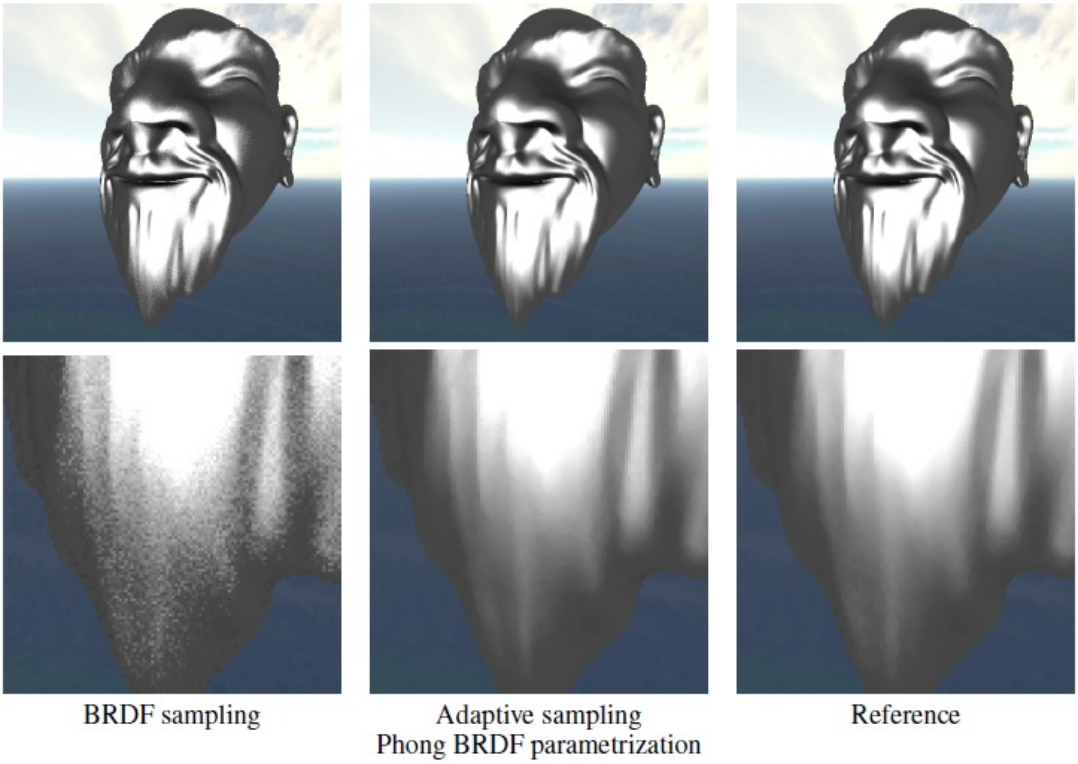

Figure 7: Specular objects rendered with BRDF sampling (80 samples per pixel), adaptive sampling with Phong parametrization (70 samples per pixel), and a reference obtained with 500 samples generated with BRDF sampling. The lower images are zoom-ins of Ming's chin.

Finally in Figure 7 we demonstrate the adaptive sampling approach in specular surface rendering. This figure compares the images rendered by classical BRDF sampling using 80 samples per pixel and adaptive sampling with Phong parametrization working with 70 samples per pixel. Both methods were given the same computation time (45 seconds). We also show a reference image for comparison, that was obtained with 500 samples generated by BRDF sampling. The exponent of Phong BRDF is set to 5. The dynamic range of environment map is 6 to 1320.

6. Conclusions

This paper proposed an adaptive sampling scheme based on the elemental interval property of low-discrepancy series and the derivatives of integrand at the samples. The derivatives provide additional information to the sampling process, which can place samples where they are really needed. The low-discrepancy series not only distributes samples globally, but it also helps refining the sample structure locally without neighborhood searches. Having generated the samples and evaluated the integral, we have all information needed to evaluate Equation (4), which provides error bounds for the estimate. Such bounds are badly needed in predictive rendering applications. We proposed the application of method in integration problems of moderate dimension, for which environment mapping is a straightforward choice.

Acknowledgement

This work has been supported by the TeraTomo project of the National Office for Research and Technology, OTKA K-719922, F-68945, and by the scientific program of the ``Development of quality-oriented and harmonized R+D+I strategy and functional model at BME" (Project ID: ТБМОР-4.2.1/B-09/1/KMR-2010-0002).

References

[1] S. Agarwal, R. Ramamoorthi, S. Belongie, and H. W. Jensen, Structured Importance Sampling of Environment Maps, *ACM Transactions on Graphics*, Vol. 22(3), pp.605–612, 2003.

[2] V. Ostromoukhov, C. Donohue, and P.-M. Jodoin, Fast Hierarchical Importance Sampling with Blue Noise Properties, *ACM Transactions on Graphics*, Vol. 23(3), pp.488–495, 2004.

[3] D. Burke, A. Ghosh, and W. Heidrich, Bidirectional importance sampling for direct illumination. In *Proc. of Eurographics Symposium on Rendering*, pp.147-156, 2005.

[4] J. F. Talbot, D. Cline, and P. K. Egbert, Importance resampling for global illumination. In *Proc. of Eurographics Symposium on Rendering*, pp.139-146, 2005.

[5] F. Rousselle, P. Clarberg, L. Leblanc, V. Ostromoukhov, and P. Poulin, Efficient Product Sampling Using Hierarchical Thresholding, *The Visual Computer*, Vol. 24(7), pp.465–474, 2008.

[6] L. Szirmay-Kalos, and L. Szécsi, Deterministic Importance Sampling with Error Diffusion, *Computer Graphics Forum (Proceedings of Eurographics Symposium on Rendering)*, Vol. 28(4), pp.1055–1064, 2009.

[7] E. Veach, and L. J. Guibas, Metropolis light transport. In *Proc. of SIGGRAPH 97*, pp.65-76, 1997.

[8] C. Kelemen, L. Szirmay-Kalos, G. Antal, and F. Csonka, A Simple and Robust Mutation Strategy for the Metropolis Light Transport Algorithm, *Computer Graphics Forum (Proceedings of Eurographics 2002)*, Vol. 21(3), pp.531–540, 2002.

[9] O. Cappé, A. Guillin, J.-M. Marin, and C. P. Robert, Population Monte Carlo, *Journal of Computational and Graphical Statistics*, Vol. 13, pp.907–929, 2004.

[10] Y.-C Lai, S. H. Fan, S. Chenney, and C. Dyer, Photorealistic image rendering with Population Monte Carlo energy redistribution. In *Proc. of Eurographics Symposium on Rendering*, pp.287–295, 2007.

[11] P. Clarberg, W. Jarosz, T. Akanine-Möller, and H. W. Jensen, Wavelet Importance Sampling: Efficiently Evaluating Products of Complex Functions, *ACM Transactions on Graphics*, Vol. 24(3), pp.1166–1175, 2005.

[12] D. Cline, P. K. Egbert, J. F. Talbot, and D. L. Cardon, Two stage importance sampling for direct lighting. In *Proc. of Eurographics Symposium on Rendering*, pp.103–113, 2006.

[13] G. P. Lepage, VEGAS: An Adaptive Multidimensional Integration Program, Tech. Rep. CLNS-80/447, Cornell University, 2000.

[14] A. Keller, Hierarchical Monte Carlo Image Synthesis, *Mathematics and Computers in Simulation*, Vol. 55(1-3), pp.79–92, 2001.

[15] T. Whitted, An Improved Illumination Model for Shaded Display, *Communications of the ACM*, Vol. 23(6), pp.343–349, 1980.

[16] M. R. Bolin, and G. W. Meyer, An error metric for Monte Carlo ray tracing. In *Proc. of Eurographics Workshop on Rendering*, pp.57–68, 1997.

[17] J. Rigau, M. Feixas, and M. Sbert, Refinement criteria based on f-divergences. In *Proc. of Eurographics Workshop on Rendering*, pp.260–269, 2003.

[18] T. Hachisuka, W. Jarosz, R. P. Weistroffer, K. Dale, G. Humphreys, M. Zwicker, and H. W. Jensen, Multidimensional Adaptive Sampling and Reconstruction for Ray Tracing, *ACM Transactions on Graphics*, Vol. 27(3), pp.33:1–33:10, 2008.

[19] H. Igehy, Tracing ray differentials. In *Proc. of SIGGRAPH 99*, pp.179-186, 1999.

[20] F. Suykens, and Y. D. Willems, Path differentials and applications. In *Proc. of Eurographics Workshop on Rendering*, pp.257–268, 2001.

[21] L. Schjøth, J. R. Frisvad, K. Erleben, and J. Sporring, Photon differentials. In *Proc. of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia*, pp.179-186, 2007.

[22] C. Ulbricht, A. Wilkie, and W. Purgathofer, Verification of Physically Based Rendering Algorithms, *Computer Graphics Forum*, Vol. 25(2), pp.237–255, 2006.

[23] J. Arvo, K. Torrance, and B. Smits, A framework for the analysis of error in global illumination algorithms. In *Proc. of SIGGRAPH 94*, pp.75-84, 1994.

[24] D. Lischinski, B. Smits, and D. P. Greenberg, Bounds and error estimates for radiosity. In *Proc. of SIGGRAPH 94*, pp.67-74, 1994.

[25] P. Bekaert, and Y. D. Willems, Error control for radiosity. In *Proc. of Eurographics Workshop on Rendering*, pp.153–164, 1996.

[26] M. Sbert, Error and Complexity of Random Walk Monte Carlo Radiosity, *IEEE Transactions on Visualization and Computer Graphics*, Vol. 3(1), pp.23–38, 1997.

[27] 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 (Proceedings of Eurographics 97)*, Vol. 16(3), pp.271–281, 1997.

[28] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA 1992.

[29] T. Kolling, and A. Keller, Efficient Multidimensional Sampling, *Computer Graphics Forum (Proceedings of Eurographics 2002)*, Vol. 21(3), pp.557–563, 2002.

[30] K. Dimitriev, S. Brabec, K. Myszkowski, and H. -P. Seidel, Interactive global illumination using selective photon tracing. In *Proc. of Eurographics Workshop on Rendering*, pp.25–36, 2002.

[31] A. Keller, Myths of computer graphics. In *Proc. of Monte Carlo and Quasi-Monte Carlo Methods 2006*, pp.217–243, 2006.

[32] P. Debevec, Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proc. of SIGGRAPH 98*, pp.189–198, 1998.

[33] R. Ramamoorthi, and P. Hanrahan, An efficient representation for irradiance environment maps. In *Proc. of SIGGRAPH 2001*, pp.497–500, 2001.

[32] L. Szécsi, L. Szirmay-Kalos, M. Kurt, B. Csébfalvi: Adaptive Sampling for Environment Mapping, Spring Conference on Computer Graphics. Budmerice, 2010. pp. 75-82