Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computers & Graphics 29 (2005) 738–755
COMPUTERS & GRAPHICS
www.elsevier.com/locate/cag

Adaptive density estimation using an orthogonal series for global illumination

Kam Wah Wong[a,*], Wenping Wang[b]

[a]School of Creative Media, City University of Hong Kong, China
[b]Department of Computer Science, The University of Hong Kong, China

Abstract

In Monte-Carlo photon-tracing methods energy-carrying particles are traced in an environment to generate *hit points* on object surfaces for simulating global illumination. The surface illumination can be reconstructed from particle hit points by solving a density estimation problem using an orthogonal series. The appropriate number of terms of an orthogonal series used for approximating surface illumination depends on the numbers of hit points (i.e. the number of samples) as well as illumination discontinuity (i.e. shadow boundaries) on a surface. Existing photon-tracing methods based on orthogonal series density estimation use a pre-specified or fixed number m of terms of an orthogonal series; this results in undesirable visual artifacts, i.e. either near-constant shading across a surface which conceals the true illumination variation when m is very small or excessive illumination oscillation when m is very large. On the other hand, interactive user specification of the number of terms for different surface patches is inefficient and inaccurate, and thus is not a practical solution. In this paper an algorithm is presented for automatically determining on the fly the optimal number of terms to be used in an orthogonal series in order to reconstruct surface illumination from surface hit points. When the optimal number of terms required is too high due to illumination discontinuity of a surface, a heuristic scheme is used to subdivide the surface along the discontinuity boundary into some smaller patches, called *sub-patches*, so as to allow a smaller number of terms in the orthogonal series to optimally represent illumination on these sub-patches. Experimental results are presented to show that the new method improves upon other existing orthogonal series-based density estimation methods used for global illumination in both running time and memory requirements.

Keywords: Photon-tracing; Monte Carlo simulation; Probability and statistics; Density estimation

1. Introduction

Methods for global illumination computation are categorized into two classes: view-dependent methods and view-independent methods [1]. In a view-dependent method, such as the classical ray tracing, a viewpoint is specified by the user and lighting information depending on the viewpoint is computed to synthesize an image. In a view-independent method, such as the classical radiosity, only the lighting information independent of the viewpoint is computed and stored for subsequent image rendering. The lighting information independent of the viewpoint includes all the light energy interactions between surfaces. Multi-pass methods combining both view-independent and view-dependent light information are the state of the art in global illumination computation.

*Corresponding author. Tel.: +852 2788 7204; fax: +852 2788 7165.
E-mail address: smkam@cityu.edu.hk (K.W. Wong).

0097-8493/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cag.2005.08.012

Although view-independent methods do not provide a complete global illumination solution, they increase the realism of three-dimensional (3D) virtual environments significantly.

Monte-Carlo methods based on particle tracing is an approach towards computing global illumination that is currently under active research [2–17]. There are two stages in a photon-tracing global illumination method: a light transport stage and an illumination reconstruction stage. In the light transport stage, energy-carrying particles are emitted from light sources, traced through the environment, and reflected from surfaces until they are absorbed probabilistically. In the illumination reconstruction stage, illumination on a surface is estimated from the distribution of all particle-hit points on the surface. The process is illustrated in Fig. 1.

Walter et al. [5] formulated the two-stage photon-tracing global illumination method as a *density estimation framework*, based on the fact that illumination reconstruction from particle hits is an instance of the well-known *density estimation* problem in statistics. Several techniques can be found in the statistics literature for solving the density estimation problem [18], some of which have already been applied to solve problems in computer graphics; these include the histogram method [4,19], the kernel method [5,20], and the orthogonal series estimator [3].

The success of a density estimation framework depends on the performance of the density estimation method used in the framework. In this paper we study the application of the orthogonal series estimator in a photon-tracing global illumination method. A critical parameter in an orthogonal series estimator is the number m of terms in a series used to best approximate illumination on a surface patch; this number m should be different for different surfaces, since it depends on the number of samples (i.e. hit points) on the surface as well as illumination discontinuity along shadow boundaries. However, existing density estimation frameworks based on an orthogonal series estimator use a fixed m for all surface patches in a scene, thus causing either an overly flat shading that conceals the true illumination variation when m is relatively small or excessive illumination oscillation when m is relatively large. Moreover, as will be seen later in this paper, the typical values of m vary in a large range. Hence, it is inefficient, and inaccurate, for the user to specify appropriate values of m for different surfaces.

We present a new adaptive orthogonal series estimator that is integrated with a surface subdivision scheme. Invoking established results from statistics, we determine automatically the optimal number of terms for illumination approximation on a surface patch. When this optimal number of terms becomes too large due to significant illumination discontinuity, we subdivide the surface patch into several smaller surface patches, called *sub-patches*, so that the number of terms that should be used on each sub-patch can be kept small, since high-degree polynomials not only take extra time and memory to process but also cause undesirable illumination oscillations. Our experimental results show that this new density estimator outperforms other existing photon-tracing global illumination methods based on an orthogonal series density estimator.

The remainder of this paper is organized as follows: in Section 2 we review some typical methods for solving the density estimation problem in global illumination computation. In Section 3 we consider determining the optimal number of terms used in an estimation function for illumination representation. In Section 4 we present the adaptive nonuniform subdivision scheme. The implementation of our method, experimental results, and comparison with other methods are given in Section 5. The conclusion is drawn in Section 6.

2. Density estimation and photon-tracing

The density estimation problem can be stated as follows: given n independent samples $\{X_1, \ldots, X_n\}$ drawn from an unknown probability density function $f(x)$, derive a density function $\hat{f}(x)$ to approximate $f(x)$.

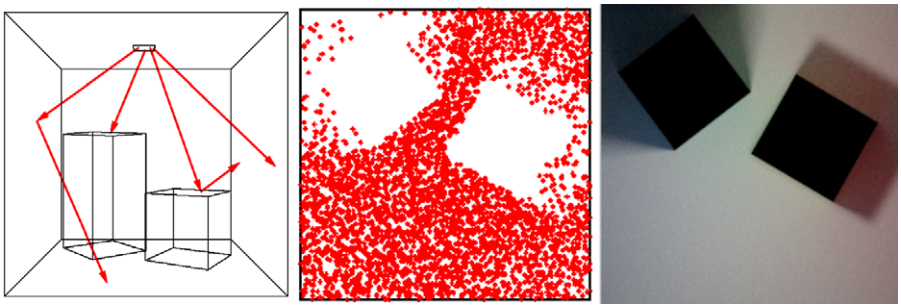

Fig. 1. The photon-tracing process. Left: particles are traced through the scene. Middle: particle-hit points on the floor. Right: estimated illumination of the floor.

It is well known [19,21] that the illumination reconstruction process in a photon-tracing method is a density estimation problem—the particles-hit points are the sample points $\{X_i\}$, and the irradiance is a scaled probability density function $f(x)$.

Note that all formulations in this paper assume a single wavelength. In our implementation three color components (R,G,B) are traced and computed independently. One simple extension to more color bands is to trace photons for each color band independently.

There are three commonly used techniques for solving the density estimation problem [18], the histogram method, the kernel method, and the orthogonal series estimator, all of which have been used in computer graphics to solve the photon-tracing global illumination problem. Below we give a brief review of these techniques. We use the one-dimensional case to illustrate the idea, but note that it is the two-dimensional (2D) case that is used in global illumination applications.

- *Histogram*: Suppose that the domain $[a,b]$ of a variable x is divided into sub-intervals $[a+mh, a+(m+1)h]$, each of which is called a *bin* with width h. Suppose that there are n samples in $[a,b]$ with a distribution $f(x)$. Then the histogram estimator gives the estimation function of $f(x)$ by

$$\hat{f}(x) = \frac{1}{nh} * (\textit{the number of samples in the same bin of } x),$$

which is a piecewise constant function. Fig. 2 shows an example of the histogram estimator.

The "meshing" technique [2,22,23] in computer graphics is a variant of the histogram method, in which each surface is normally subdivided into some patches prior to the particle-tracing stage. The illumination of each patch is a constant proportional to the number of particle hits on the patch. Since the resulting estimation of illumination is a piecewise constant function over a collection of patches representing a smooth surface, a bi-linear interpolate method, such as Gouraud shading, is usually used to produce smooth illumination on a surface.

One problem with the "meshing" technique is that it is unknown beforehand as to how finely a surface should be subdivided; on the one hand, a very fine subdivision may cause some patches to not capture many particles sufficiently, resulting in noise and artifacts in the estimated illumination, and on the other hand, a very coarse subdivision may cause the estimated illumination to appear overly flat, thus concealing the details of illumination variation. This problem can be addressed by the "adaptive meshing" technique [4] which attempts to divide a surface adaptively on the fly during the particle-tracing stage, based on the current number of particles captured by the patch. However, as constant illumination is assumed, it still requires a fine surface subdivision, or equivalently, a large number of small surface patches, to represent a smoothly varying illumination or illumination discontinuity. This means slow convergence and excessive space consumption for attaining a high accuracy of illumination computation.

- *Kernel estimator*: Let $K(x)$ be a kernel function satisfying the condition $\int K(x)\,dx = 1$. A kernel function is normally unimodal, i.e. having one peak over its support. A kernel estimator gives the approximate function by

$$\hat{f}(x) = \frac{1}{nh} * \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right),$$

where h, called the *kernel width*, is the *smoothing parameter*, which controls the smoothness of estimation. Fig. 3 shows examples of a kernel estimator with different kernel widths. The kernel function used here is the Epanechnikov kernel [5] defined by

$$K(x) = \begin{cases} \frac{3}{4}(1 - |x|^2) & \text{if } |x| \leqslant 1, \\ 0 & \text{otherwise.} \end{cases}$$

From Fig. 3 we see that the choice of h affects the smoothness of the resulting estimation; however, it is not a trivial task to choose a suitable h [21]. The kernel width should be narrow enough to capture the details in density distribution, and at the same time, it should be wide enough to avoid spurious fine structures. A shortcoming of a kernel-based method is that all samples need to be stored in order to evaluate $\hat{f}(x)$. Hence the storage requirement becomes prohibitive for a large sample size, such as in the photon-tracing global illumination computation. Walter et al. [5] propose a photon-tracing algorithm that uses a kernel estimator to reconstruct surface illumination. In this algorithm the kernel width is

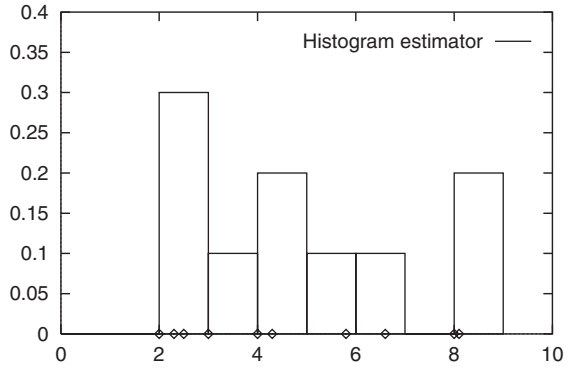

Fig. 2. The histogram estimator for the samples {2, 2.3, 2.5, 3, 4, 4.3, 5.8, 6.6, 8, 8.1}.

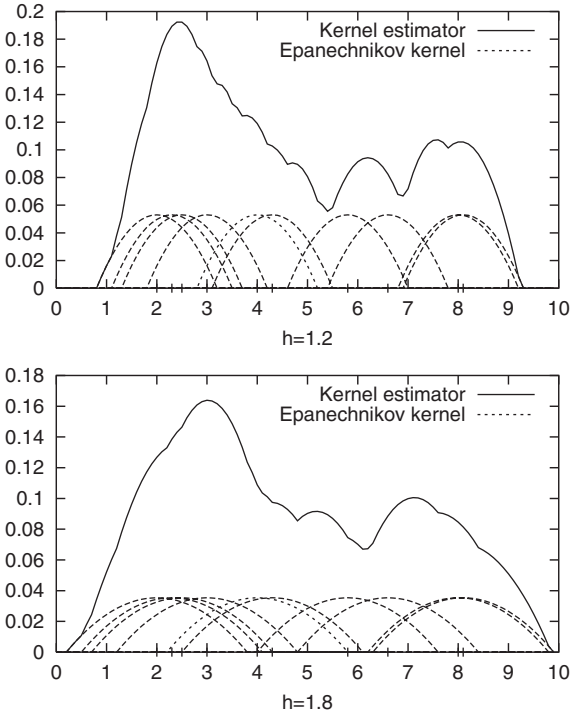

Fig. 3. Kernel estimators with different kernel width.

selected using a heuristic method which requires a large secondary storage to store all particle-hit points.

- *Orthogonal series estimator*: Let $\{\phi_i\}_{i\geqslant 0}$ be a complete set of orthonormal basis functions on an interval I. Suppose that a density function $f(x)$ is represented by

$$f(x) = \sum_{i=0}^{\infty} a_i \phi_i(x), \quad x \in I,$$

where $a_i = \int_I f(x)\phi_i(x)\,\mathrm{d}x$. An orthogonal estimator is then given by

$$\hat{f}_m(x) = \sum_{i=0}^{m} \hat{a}_i \phi_i(x) \tag{1}$$

for some integer $m \geqslant 0$, where $\hat{a}_i$ is an estimation of a_i. Since the samples $\{X_i\}$ are drawn from the probability density function $f(x)$,

$$\begin{aligned} a_i &= \int_I f(x)\phi_i(x)\,\mathrm{d}x \\ &= E[\phi_i(X)] \\ &\approx \frac{1}{n}\sum_{j=1}^{n} \phi_i(X_j) \\ &\equiv \hat{a}_i, \end{aligned} \tag{2}$$

where X is a random variable following the probability density function $f(x)$. Note that $\hat{a}_i$ is an unbiased estimator of a_i [18]. In general, m in Eq. (1) is much smaller than the sample size n; hence, instead of storing all particle-hit points, one only needs to update $\hat{f}_m(x)$ according to Eq. (2) with the arrival of each new particle on a surface and to store a small number of coefficients $\{\hat{a}_i : i = 0, \ldots, m\}$ to represent $\hat{f}_m(x)$.

Clearly, the optimal value of m, i.e. the number of terms in Eq. (1), depends on the number of sample points and the variation of $f(x)$. It is a tricky task to pre-specify appropriate values of m for different surfaces in a complex scene, since the number of hit points and illumination discontinuity on a surface patch are unknown at the beginning and they only become available gradually with the progress of the particle-tracing process.

Feda [3] and Tobler et al. [4] give two photon-tracing algorithms based on the orthogonal series estimator. In both methods the user needs to specify the parameter m for each surface. Our investigation shows, as will be seen in detail shortly, that this treatment is tedious and prone to inaccuracy, and is thus not a practical solution for a complex scenario.

2.1. An overview of our method

We propose a new photon-tracing global illumination method that combines two ideas to make the orthogonal series estimator practical for photon-tracing applications. First, our method automatically determines for each surface the appropriate number of terms that should be used in the orthogonal series. Second, when illumination discontinuity entails a large number of terms, we subdivide the surface adaptively to better capture illumination discontinuity. Hence, the method combines the advantage of an orthogonal series estimator for modeling smooth illumination variation and that of adaptive subdivision for modeling illumination discontinuity.

The integration of an orthogonal series estimator and surface subdivision has been motivated by the work of Tobler et al. [4]. However, there are two major differences between our method and the method of Tobler et al. [4]: (1) the optimal number of terms used in an orthogonal series for each surface patch is determined automatically in our method, while this number is specified by the user in the method of Tobler et al; and (2) using image-processing techniques and orthogonal series over quadrilateral and triangular domains, we adaptively subdivide a surface patch along a direction that best fits illumination discontinuity, while only uniform subdivision of rectangular patches is considered by Tobler et al.

This paper reports on an extension to and improvements over our previous work presented at Pacific

Graphic 2000 Conference [24] in terms of the following aspects:

- Experimental evidence is now provided to show the need for using different number of orthogonal polynomials terms for different surfaces in order to achieve optimal illumination reconstruction. This evidence was absent in [24].
- Non-uniform surface subdivision is now incorporated into our method, and for this purpose, computation using orthogonal polynomials on the triangular domain is now studied. In comparison, our preliminary work in [24] considers orthogonal polynomials on rectangular domains and thus supports only uniform subdivision.
- A novel application of adaptive-width low-pass filters is proposed in the present paper for removing discontinuity between reconstructed illumination approximating functions on adjacent surface patches. This improves significantly the visual quality of output images, as compared with the treatment based on polynomial interpolation in [24].

Since our method falls into the category of a view-independent global illumination algorithm, this method can deal with a static scenario with camera movement. Moreover, the method can only deal with diffuse surfaces, because in the density estimation stage we only consider the particle-hit positions without considering the incoming direction of the particles.

3. Optimal number of terms for best approximation

In this section we discuss the motivation and technique for determining the optimal number of terms in an orthogonal series for approximating surface illumination.

3.1. Error of an orthogonal series estimator

We first consider the error of an orthogonal series estimator. Throughout this paper we use a bivariate orthogonal series over a quadrilateral domain that is derived from the Legendre polynomials and an orthogonal series over a triangular domain that is derived from the Jacobi polynomials (see the description of these two orthogonal series in Appendices A1 and A2, respectively). However, the ideas presented apply to other orthogonal series as well. There are different orthogonal series with different approximation properties. The comparison of these different orthogonal series regarding their applications to solving the density estimation problem is beyond the scope of this paper.

The error of an estimation based on a particular set of samples $\{X_1,\ldots,X_n\}$ is measured by the *integrated-square error* (ISE) [18] defined by

$$ISE = \int_I [\hat{f}(x) - f(x)]^2 \,\mathrm{d}x.$$

The average error over all possible sample sets is measured by the *mean-integrated-square error* (MISE) defined by

$$MISE = E\left\{\int_I [\hat{f}(x) - f(x)]^2 \,\mathrm{d}x\right\}. \tag{3}$$

Since an orthogonal series estimator is used, we substitute the expression of $\hat{f}_m(x)$ in Eq. (1) for $\hat{f}(x)$ into Eq. (3) to obtain

$$\begin{aligned} MISE &= E\left\{\int_I \left[\sum_{i=0}^{m}(\hat{a}_i - a_i)\phi_i(x) - \sum_{i=m+1}^{\infty} a_i\phi_i(x)\right]^2 \mathrm{d}x\right\} \\ &= E\left[\sum_{i=0}^{m}(\hat{a}_i - a_i)^2 + \sum_{i=m+1}^{\infty} a_i^2\right] \\ &= \sum_{i=0}^{m} Var(\hat{a}_i) + \sum_{i=m+1}^{\infty} a_i^2, \end{aligned} \tag{4}$$

where $Var(\hat{a}_i)$ is the variance of $\hat{a}_i$, and $\hat{a}_i$ is an unbiased estimator of a_i.

Eq. (4) illustrates the well-known tradeoff between variance and bias in the density estimation problem [18]. The first term is the error introduced by the variance of the estimator due to the error in approximating a_i by $\hat{a}_i$, which is in turn caused by an insufficient number of samples, and the second term is the bias of the estimator due to truncation of the infinite series $f(x) = \sum_{i=0}^{\infty} a_i\phi_i(x)$.

Fig. 4 shows the consequences when the values of m are not set properly. Given a fixed sample set, in the left image all surfaces use only the constant term in the orthogonal series, i.e. $m = 0$, leading to flat shading. This occurs because truncating off all the terms except for the constant one of the infinite series introduces a large bias in the estimator. In the middle image all surfaces use $m = 45$ terms in the series, which are too many for those surfaces with few particle hits; as a consequence, the variance in the coefficients $\hat{a}_i$ causes excessive illumination oscillation. In the right image, appropriately chosen values of m are used for different surfaces—relatively small values of m are used for those surfaces with few particle hits or smooth illumination, and relatively large values of m for other surfaces. (See Fig. 5 later in this section for the distribution of m.) The resulting image is much improved as compared with the other two.

We now return to the analysis of the error measured by (4). Clearly, $Var(\hat{a}_i) \to 0$ when n approaches ∞, and in this case a larger m leads to a smaller error, which is $\sum_{i=m+1}^{\infty} a_i^2$. However, since n is finite, the variance of $\hat{a}_i$ may contribute significantly to the total error, especially

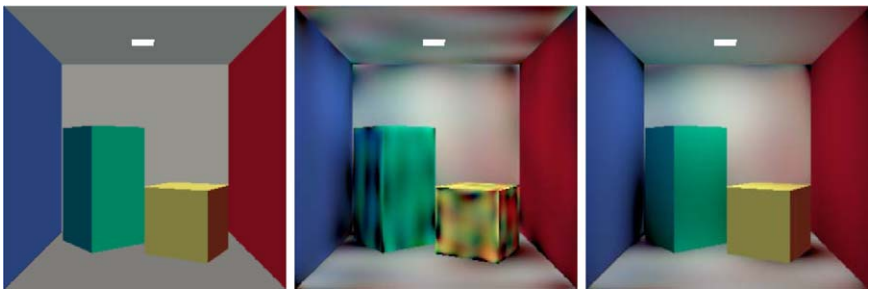

Fig. 4. Different numbers (m) of terms to use: fixed *vs* adaptive. Left: all surfaces use $m = 0$, and the illumination appears flat. Middle: all surfaces use $m = 45$, and illumination oscillation occurs on some surfaces. Right: different values of m are used for different surfaces, producing a better image.

when n is relatively small. Therefore, given a finite sample set $\{X_1, \ldots, X_n\}$, we should select an appropriate value of m to attain a balance between the first term and the second term of Eq. (4) and, as a result, to minimize *MISE*. Substituting (2) into (4), we obtain

$$MISE \approx \frac{1}{n-1}\sum_{i=0}^{m}[2\hat{d}_i - (n+1)\hat{a}_i^2] + \sum_{i=0}^{\infty} a_i^2, \tag{5}$$

where $\hat{d}_i = \frac{1}{n}\sum_{j=1}^{n}\phi_i^2(X_j)$. The details of deriving (5) can be found in [25,26]. So we need to choose a number m which minimizes the MISE, or equivalently, minimizes the function

$$J(m) = \frac{1}{n-1}\sum_{i=0}^{m}[2\hat{d}_i - (n+1)\hat{a}_i^2],$$

since the second term on the right-hand side of Eq. (5) is independent of m.

3.2. Characteristics of $J(m)$

In this subsection we use several scene setups to study the characteristics of $J(m)$ and develop a strategy for determining the optimal value of m. Consider the five scenes shown in Fig. 5. In the first three scenes the receiving surfaces contain a smooth illumination, while in the last two scenes the receiving surfaces contain illumination discontinuity. In the left column of Fig. 6 the functions $J(m)$ are plotted against m, where m is in the range $[0, \ldots, 60]$, for the receiving surfaces in the five scenes, with three cases of the surfaces capturing 400, 1600, and 6400 particles, respectively, for each setup. The right column of Fig. 6 shows the computed coefficients $\hat{a}_i$ plotted against i.

The following observations can be made from the charts in Fig. 6:

- The patterns of $J(m)$ fall into two categories: (i) For the first three scenes in which the irradiance is smooth, an optimal m, denoted by m^*, exists such that it minimizes $J(m)$. In this case $J(m)$ increases or remains unchanged for all $m \geqslant m^*$. (ii) For the last two scenes where there is strong illumination discontinuity, there is no such "optimal" m, and in this case $J(m)$ decreases as m increases. Therefore, to reduce the error in the latter case one would have to use a very large value of m, i.e. to include many terms in the orthogonal series estimator, but this would be inefficient and may even introduce more errors due to the inherent oscillation of a high-degree polynomial when the number of surface hit point is relatively small; large local oscillations may not be revealed by MISE, which is an L_2-type error measurement. This observation suggests that a surface subdivision scheme should be used to better capture illumination discontinuity.
- The value of m^* is not stable for smooth irradiances when the number of particle hits, denoted by n, is small (e.g. $n = 400$), i.e. m^* may shift to a larger value when the number of particles captured increases. However, m^* becomes stable when n is relatively large, e.g. for $n = 1600$ or $n = 6400$. So we suppose that $n \geqslant 2000$ is sufficiently large for obtaining a stable m^*. This observation suggests a "check point" in our algorithm, i.e. when the algorithm should verify and determine whether a surface has illumination discontinuity or not; this will be discussed in more detail later.
- The function $J(m)$ is not necessarily monotonic; it may have multiple local minima in some cases. This is illustrated by the first scene in Fig. 5 and its charts in Fig. 6, when $n = 400$. Since the illumination is symmetric on the receiver, there is no linear variation over the entire surface; consequently, the coefficients for the linear terms (with $i = 1, 2$) should be zero, and thus including these terms in the series should not cause any problem. However, as variance exists, the computed coefficients $\hat{a}_1$ and $\hat{a}_2$ are non-zero, so including these two linear terms may increase the error. Including further terms in the series (e.g. the three quadratic terms with $i = 3, 4, 5$) will then

reduce the error. This explains why the function $J(m)$ may have local minima which are not globally minimal.

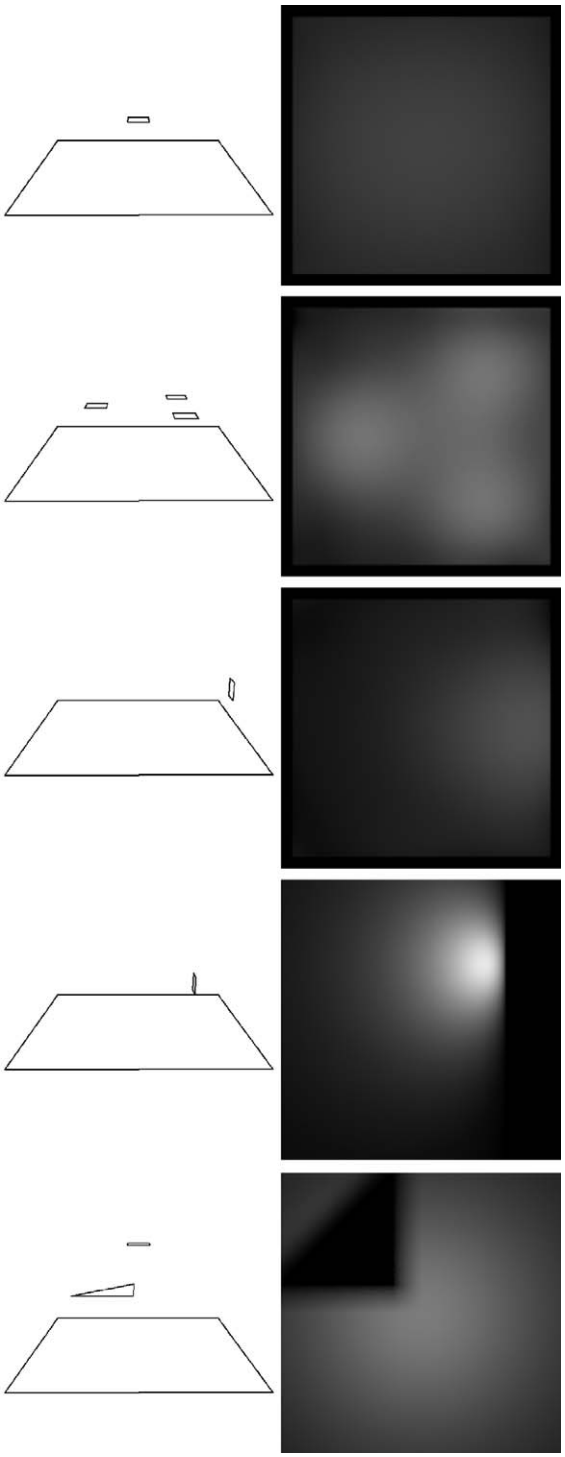

Fig. 5. Test scenes for investigating the behavior of $J(m)$. The left column shows the scene setup, and the right column shows illumination on receivers. From top to bottom: one light source without blocking, three light sources without blocking, one light source placed perpendicular to the receiver, one light source placed perpendicular to the receiver and moved inward, and one light source with a triangle-shaped obstacle. Each light source is represented by a rectangle.

We have also examined the function $J(m)$ for two other more complex scenes. One is the scene of a Cornell box with one light source and 18 surfaces, and the other is an office scene with 4 light sources and more than 300 surfaces, as shown in Fig. 7. We observe that $J(m)$ exhibits the same behavior in these two scenes as in the simple scenes in Fig. 5. Moreover, for all of the scenes we experimented with, the values of m^* for most surfaces fall in the range $[0,\ldots,30]$, except for a few surfaces with strong illumination discontinuity, which require m^* to be larger than 30. Furthermore, within the range $[0,\ldots,30]$, the values of m^* are distributed in a rather random manner, and, in fact, are strongly dependent on the smoothness of illumination. This implies that it would be an extremely difficult and tedious task for the user to specify even a nearly optimal value of m for each surface in a small scene, let alone a complex scene. These observations provide the motivation, as well as basis, for devising an automatic method for selecting the best value of m for each surface.

3.3. Automatic determination of m

Since $J(m)$ may have multiple local minima, to avoid being trapped at a local minimum, one cannot simply increase m incrementally until $J(m+1)>J(m)$. Rather, the following strategy is adopted. Based on our extensive testing, we set $M=30$ to be the upper bound on the number of terms that will be used in the orthogonal series estimator for any surface. $M=30$ is appropriate in the sense that it is large enough to allow enough terms for modeling smooth illumination accurately and, at the same time, not too large to compromise computational efficiency; after all, if the optimal value of m exceeds $M=30$, it is normally the case where there is strong illumination discontinuity, and in this case it should be more efficient to subdivide the surface than using excessively many terms in the estimator.

With M fixed, we choose a value m^* that minimizes $J(m)$ for all $m\leqslant M$. Note that the value of m^* of a surface depends on the number of particles captured by the surface, especially when the sample size is small. However, regardless of the sample size, our algorithm always yields the most appropriate m^* for each surface, subject to $m^*\leqslant 30$.

One remark is in order about calculating the function $J(m)$. Since $J(m)$ involves the particle-hit points, it

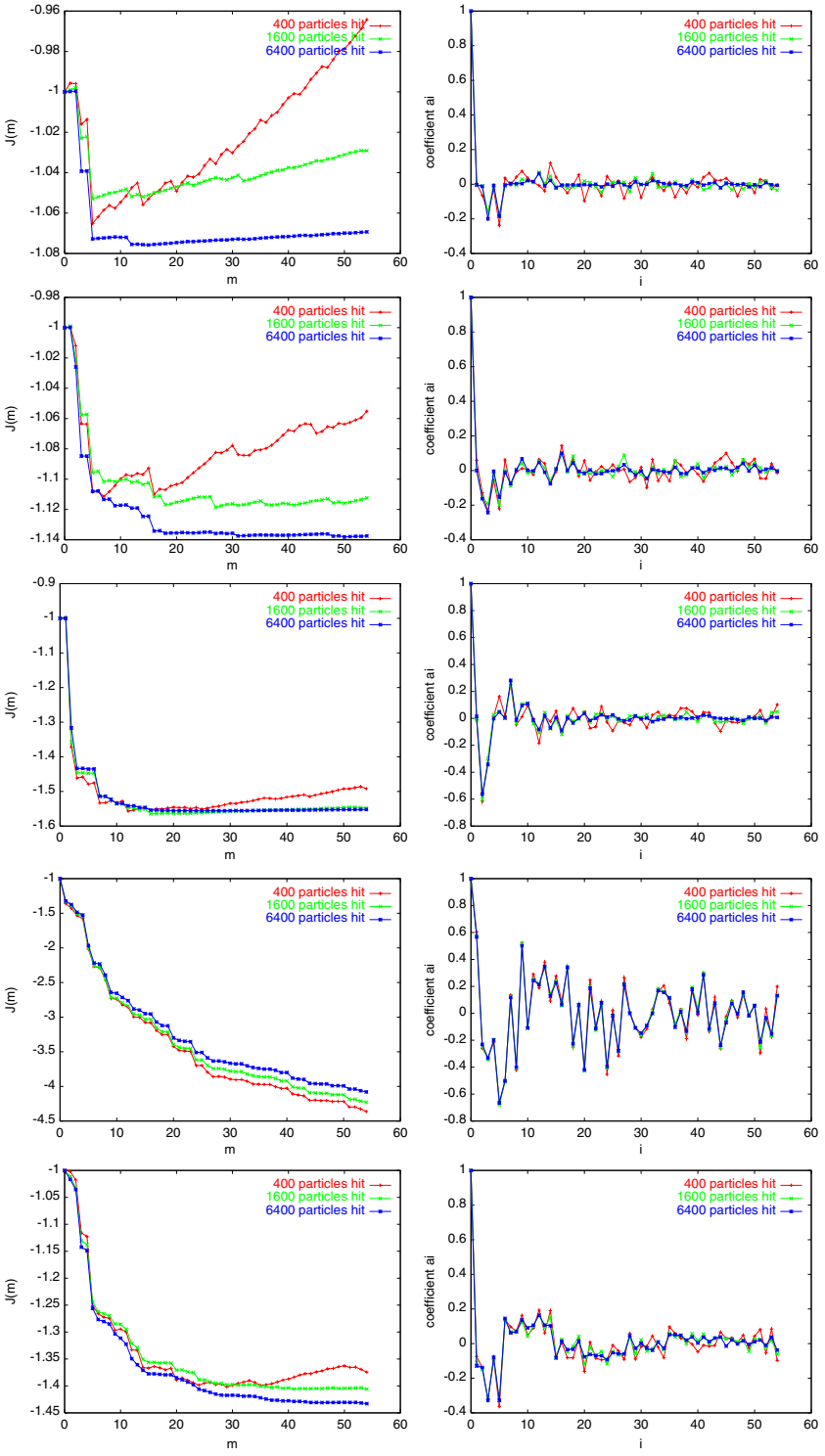

Fig. 6. The behavior of $J(m)$ for the five scenes shown in Fig. 5. The left column shows $J(m)$ against m, and the right column shows the coefficients $\hat{a}_i$ against i.

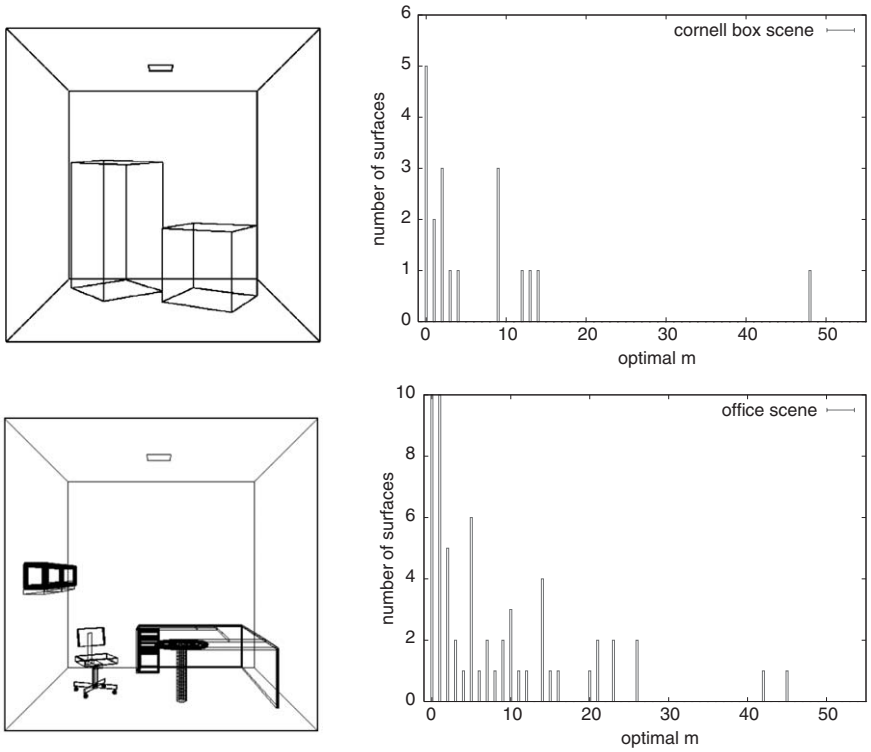

Fig. 7. Distribution of the optimal values of m in the Cornell box scene (top) and an office scene (bottom).

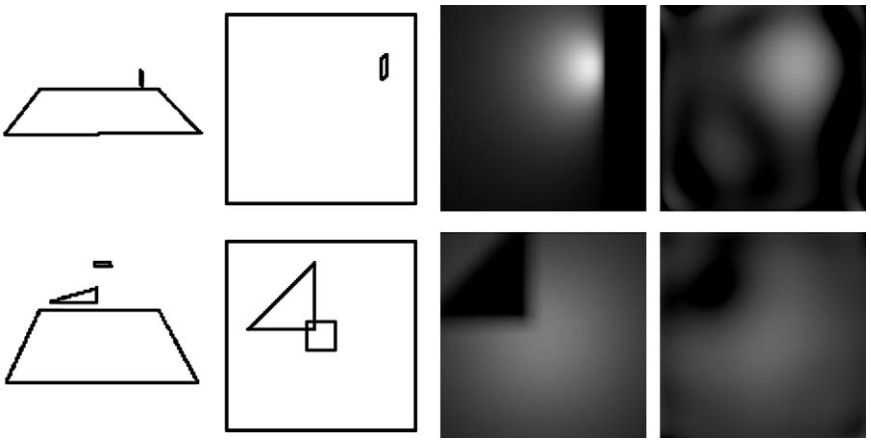

Fig. 8. Results without surface subdivision. From left to right: front view, top view, analytical solution, and approximating function for reference. The estimator uses $m = 45$ terms.

cannot be pre-computed. However, even though $\hat{d}_i$ and $\hat{a}_i$ need to be updated on every particle hit, $J(m)$ does not—we compute $J(m)$ only when we wish to find out the m^* for a surface, either before the final rendering (i.e. after all particles have been traced) or for deciding whether a surface should be subdivided or not. This is described in detail in the next section.

4. Adaptive surface subdivision

Simply increasing the number of terms in the estimation function would not produce a good approximation of illumination discontinuity on a surface, because of the inherent oscillation of a high-degree polynomial. Fig. 8 shows two simple scenes to illustrate

this difficulty. Even though a large number of terms are used, the polynomial still fails to produce a good approximation of illumination for the two scenes. Moreover, the visual artifact is even worse in the region with a low gradient, due to oscillations inherent to high-order terms.

A natural solution to the above problem is to subdivide a surface patch with a high illumination gradient into small sub-patches so that lower degree polynomials can be used to better approximate the illumination distribution on these sub-patches. In this section we will address the following issues with this subdivision scheme: (1) detection of surface patches with high illumination gradients; (2) non-uniform subdivision of such surfaces along the edge of an illumination gradient; and (3) rendering with reconstructed irradiance.

4.1. Detection of illumination discontinuity

Since a surface patch with a high illumination gradient normally entails a large number of terms in an orthogonal series for illumination approximation, we detect such surfaces by verifying the value m^* it requires, i.e. if $m^* > 30$. Specifically, during the particle-tracing stage, we compute the optimal m^* for the surface after it has captured N particles. If this m^* reaches a pre-defined value M, say 30, then the surface needs to be subdivided. The value of N can be chosen within the range from 2000 to 5000, which, as we have observed, ensures a sufficient number of particle hits to yield a reliable estimation of m^*. Fig. 9 shows the results with surface subdivision for the same two scenes as in Fig. 8. Note that a uniform subdivision is used in these examples for simplicity of discussion, while our general scheme allows a non-uniform subdivision as will be introduced below.

4.2. Non-uniform surface subdivision

Both uniform and non-uniform surface subdivisions are widely applied in computer graphics. Uniform subdivision means that a surface element is subdivided in a regular manner, such as dividing a rectangle into four smaller ones of the same size that have the same aspect ratio as the original rectangle. By non-uniform subdivision we mean here cutting along a straight line to divide a quadrilateral or a triangle into some quadrilaterals or triangles. Hence, uniform subdivision is a special case of repeated non-uniform subdivision.

Although non-uniform subdivision allows the subdivision boundary to better fit the boundary of illumination discontinuity than uniform subdivision does, and is thus more efficient in terms of both time and memory, only a uniform subdivision has been used in adaptive meshing for Monte-Carlo photon-tracing algorithms, such as in [4]. This is because most adaptive

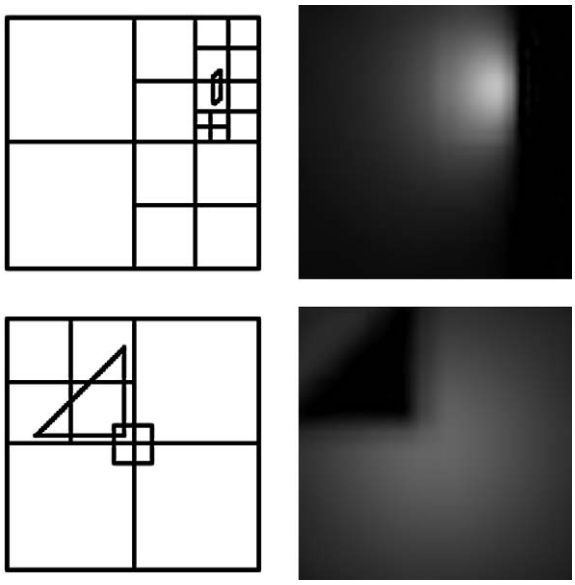

Fig. 9. The result of the new estimator with surface subdivision added.

meshing methods for photon-tracing use the so-called *preview level* technique to decide whether a surface should be subdivided. More specifically, with the *preview level* technique, a surface patch at the *current level* is subdivided into a finer *preview level* consisting of several sub-patches. Then the sub-patches at the preview level are used to keep track of particle hits. After a sufficient number of particles have been captured, the respective numbers of particles captured by the sibling sub-patches at the preview level are compared to decide whether the parent patch at the current level should be subdivided; if yes, the sub-patches at the preview level become surface patches at the next level. Since a surface at the current level is subdivided into the preview level before illumination estimation takes place, no information about illumination discontinuity is available to help decide the best way of subdivision. As a consequence, it is only natural for these methods to subdivide surfaces uniformly.

In contrast, non-uniform subdivision is used in our method. When a surface patch is detected to contain illumination discontinuity, we subdivide the surface along a line that fits the boundary of the illumination discontinuity. There are different ways of finding the boundary of illumination discontinuity. One may compute the derivatives of the estimated illumination function derived from the sample points using a similar formulation of Eqs. (1) and (2). For example, the second derivative of an estimation function can be estimated by

$$\hat{f}''_k(x) = \sum_{i=0}^{k} \hat{b}_i \phi_i(x), \quad \text{for some integer } k \geqslant 0, \tag{6}$$

where

$$\hat{b}_i = \frac{1}{n}\sum_{j=1}^{n} \phi_i''(X_j). \tag{7}$$

Details of this derivation can be found in [27]. A problem with this approach is that extra memory is needed to store the coefficients $\{\hat{b}_i\}$ for each surface; moreover, the running time would also increase since more computation is needed for every particle hit.

We use the following alternative in our implementation. The illumination estimation on a surface patch is first treated as a 2D image and then sharp illumination changes are located with a standard edge-detector based on the zero-crossing property of Laplacian filtering [28]. Specifically, when a surface needs to be subdivided, the estimated illumination of the surface is rendered as a low-resolution image (usually 256×256 is sufficient). Then the image is convolved with the Laplacian of a 2D Gaussian function in the form

$$G(x, y) = \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right), \tag{8}$$

where σ is the standard deviation of the Gaussian function. Then the zero-crossing of the convolved image indicates the edge where there is a high-illumination gradient. Finally, the Hough transform [28] is used to find a straight line that best fits the edge. (Hough transform is commonly used in image analysis to fit a line to a set of nearly collinear points.) This process is illustrated in Fig. 10.

Although we use the zero-crossing technique, we believe that any other edge-detection technique should be equally applicable. No matter which particular edge detector is used, edge detection is not, in general, a very efficient procedure. However, in our method edge detection is performed only when necessary, i.e. when a surface is detected to contain a high-illumination gradient, a condition indicated by $m^* > 30$. Therefore edge detection is not triggered for smoothly illuminated surfaces, which are normally the majority of all surfaces.

Fig. 11 shows the results of the non-uniform subdivision scheme. Note that in order to have a regular functional domain on each subsurface, the resulting subsurfaces are triangulated after each subdivision. We see in these images that the non-uniform surface subdivision scheme cuts the surfaces at a location closer to illumination changes than uniform subdivision does; hence, it produces a better image quality than the latter.

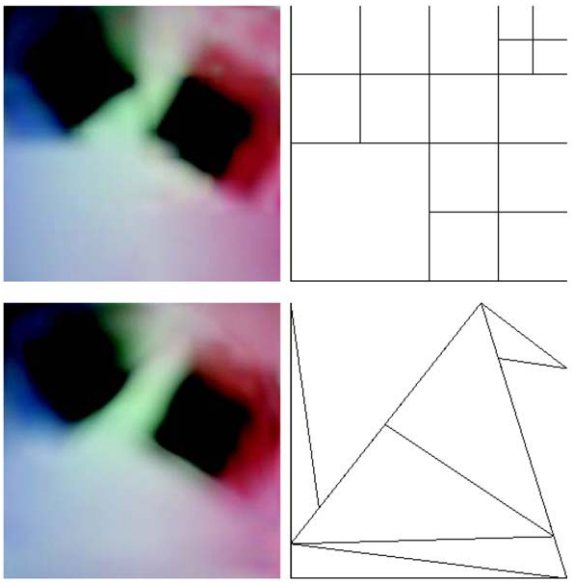

Fig. 11. Non-uniform surface subdivision. The first image on the first row is generated using uniform subdivision, and the first image on the second row is generated using non-uniform subdivision. Note that, with fewer subsurfaces, non-uniform subdivision produces better results than uniform subdivision does.

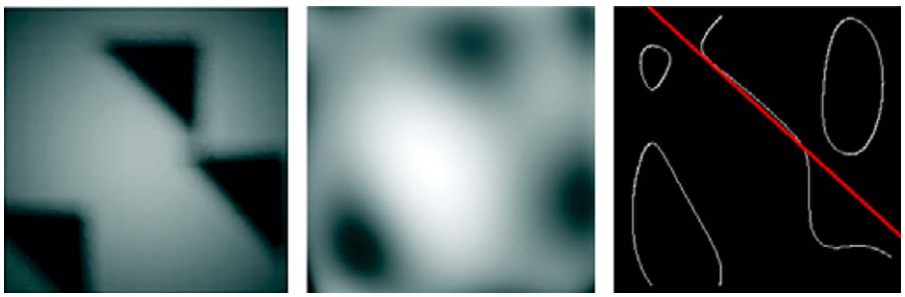

Fig. 10. The zero-crossing method for edge detection: (left) illumination on a surface by an analytical solution; (middle) estimated illumination before subdivision; (right) the white curves indicate zero-crossings of the convolved image, and the red line represents a subdivision line generated by our algorithm.

4.3. Illumination composition and rendering

Upon completion of the particle-tracing stage, we need to compose the illumination estimations at all subdivision levels to produce the final illumination. When a parent patch is subdivided, each of its sub-surfaces will have its own density estimation function, and subsequent particles hits on the surface will be captured by the sub-surface. Note that the coarse illumination estimation of the parent patch must be retained, because it accounts for the contribution of the energy transported by particles before surface subdivision; this coarse estimation function will be superimposed to its sub-patches' illumination estimations before rendering the final image.

Since illumination estimations of adjacent sub-patches are approximated by different functions which may not agree with each other along their shared boundaries, the image rendered with the composed illumination directly without any processing would contain discernible discontinuities along subdivision boundaries. To mitigate this problem, we propose a solution based on a low-pass image filter. We first render the estimated illumination of all sibling sub-patches to form an image of their parent patch. Then we apply a 2D low-pass image filter [28] to smooth out the discontinuity across subdivision boundaries. Since we only wish to remove the visual artifacts along the subdivision boundaries without affecting the estimated illumination in other continuous regions, we use a filter with adaptive width. Ideally, the filter width should be large near discontinuity across subdivision boundaries, and should taper down to zero toward the interior of a sub-patch.

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Fig. 12. The Sobel masks. (left) The horizontal mask and (right) the vertical mask.

However, when a non-uniform surface subdivision is applied, the subdivision boundaries on a surface form an irregular pattern. Moreover, if we look at the illumination across a subdivision boundary, there exist both smooth illuminations and discontinuities along the same boundary. These features make it hard to decide the filter width by merely considering the location of the subdivision boundaries.

Our algorithm chooses the filter width directly from the image before smoothing. First, we find the illumination gradients of each pixel in the image by applying a gradient mask, such as the Sobel masks shown in Fig. 12. Then for each pixel the low-pass filter width will be chosen depending on the illumination gradient at that pixel: a large filter width will be used if the gradient is large; otherwise, a small filter width will be used.

Note that this method does not consider the location of surface subdivision boundaries. We only consider the illumination gradients of the surface. This may lead to one problem: the illumination will be smoothed out in some regions away from subdivision boundaries. However, as continuous polynomials are used as the basis functions, the illumination gradients in an interior region will be much smoother than the discontinuities across a subdivision boundary. There artifacts are generally minimal and not noticeable, and satisfactory results have been obtained with this method in our experiment.

Fig. 13 shows this rendering process. The image on the left shows the illumination of a surface without interpolation or smoothing, in which we can clearly see the illumination discontinuity due to surface subdivision. The illumination gradient of the first image is shown in the second image, where the darker color

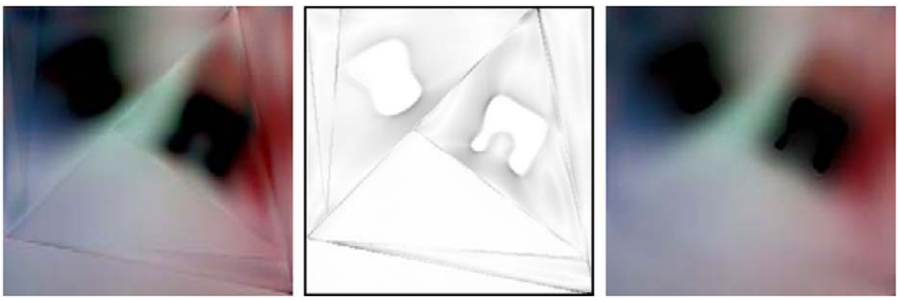

Fig. 13. Adaptive width low-pass filtering. From left to right: (1) a rendered image without interpolation; (2) illumination gradients and filter widths to be used, the darker color being for a larger filter width; (3) the final rendered image after adaptive low-pass filtering.

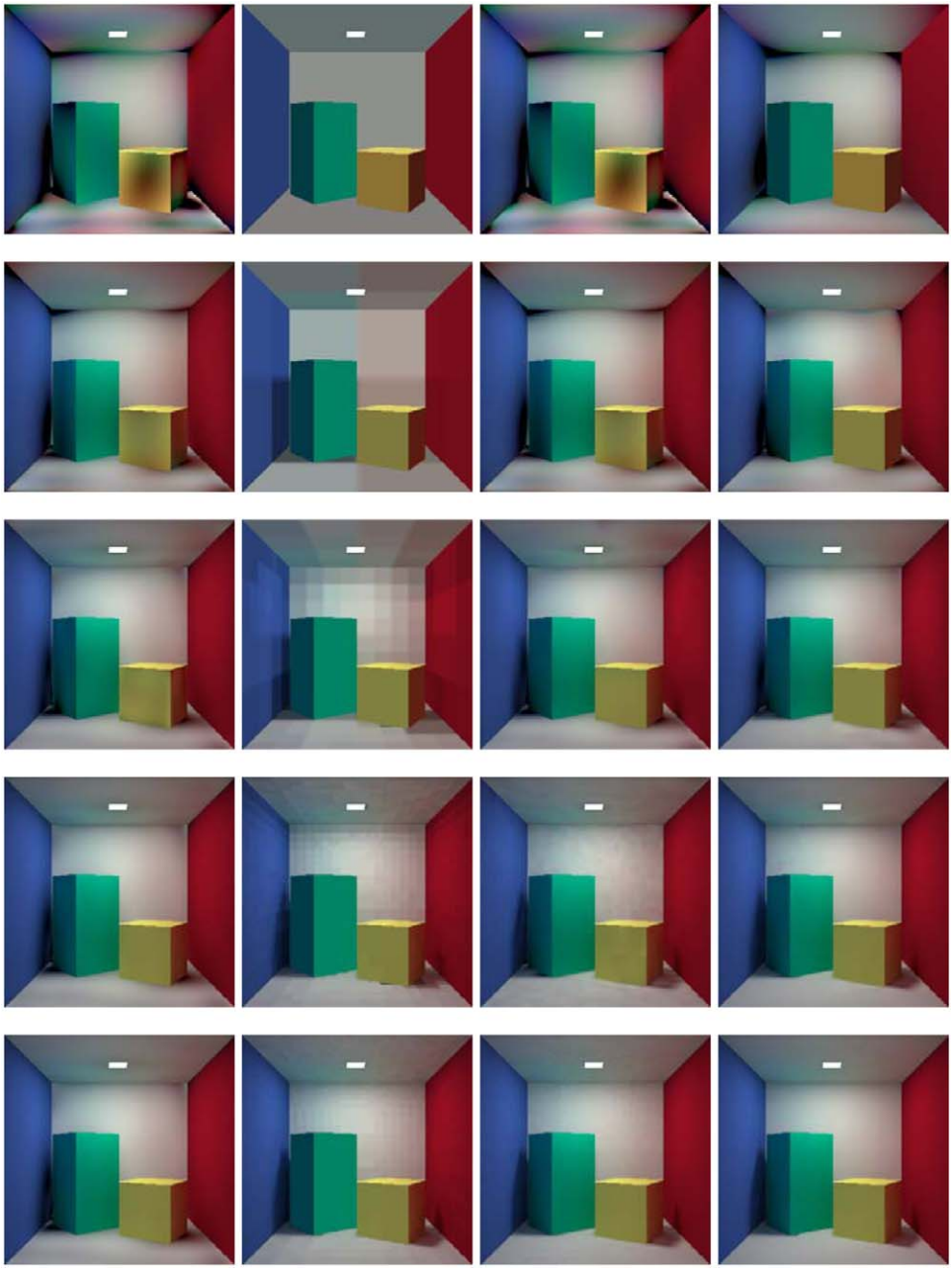

Fig. 14. Final rendered images. From left to right: *FOSE, AM-C, AM-H, NEW*. From top to bottom: 10^3, 10^4, 10^5, 10^6, and 10^7 particles are traced.

indicates a higher illumination gradient, and therefore a larger filter width. The rightmost image is the final rendered image after low-pass filtering is applied.

5. Implementation and results

In this section we present experimental results on the new photon-tracing global illumination method we have proposed. Since our method is a combination of the orthogonal series estimator and the adaptive meshing approach, the following four different methods have been implemented for comparison in terms of running time, memory requirement, and L_2 *error* (ISE).

(1) *FOSE*: The orthogonal series estimator with a user-defined m, i.e. the number of terms in the

illumination estimation function, and without surface subdivision.

(2) *AM-C*: The adaptive meshing method of Tobler et al. [4] with constant illumination.

(3) *AM-H*: Same as (2), but with a higher order estimation function for each surface. The number of terms used, i.e. m, is defined by the user (see below).

(4) *NEW*: Our new method.

These four methods are applied to the scene of a Cornell box. Two surface patches were selected for study—the floor, which has a large illumination variation, and the red wall on the right, which has a smooth illumination variation. To accommodate for their difference in illumination smoothness, 45 terms were used for the floor and 15 terms for the red wall in both *AM-H* and *FOSE*. For error measurement, the illumination computed with the four methods was compared against a reference solution which was generated by shooting 10^8 particles in the scene. A texture of size 200×200 was used to capture particle hits on each surface (Figs. 14, 15).

The results are shown in Fig. 16 and Fig. 17. Running time, memory used, and L_2 *error* (ISE) of the two surfaces are plotted against the number of particles traced. Also, Fig. 14 shows a sequence of images generated by the four methods for the Cornel box scene with an increasing number of particles traced.

The statistics and images lead to the following observations:

- *Error vs. the number of particles*: *FOSE* gives the largest error. The error is bounded from below and not improved by increasing the number of particle hits. *AM-C* and *AM-H*, i.e., the two adaptive meshing methods, produce smaller errors, thanks to

Fig. 15. A complex scene rendered by the new method. Artifacts on the chair show a shortcoming of the photon-tracing global illumination method.

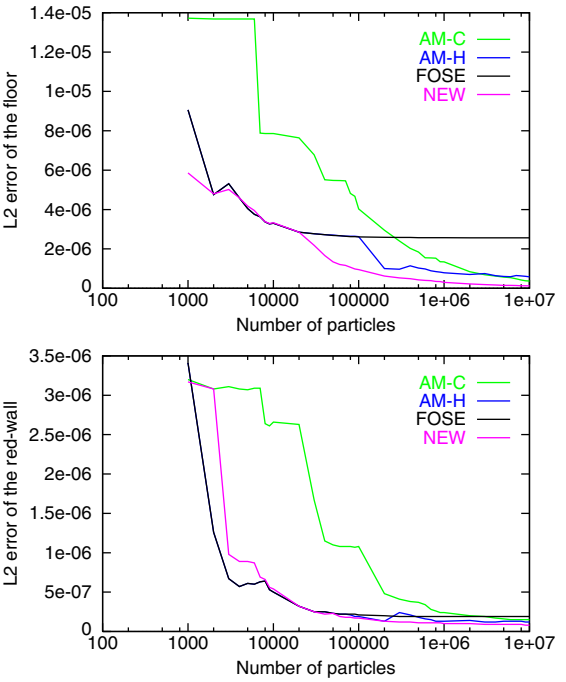

Fig. 16. The L_2 error of the methods.

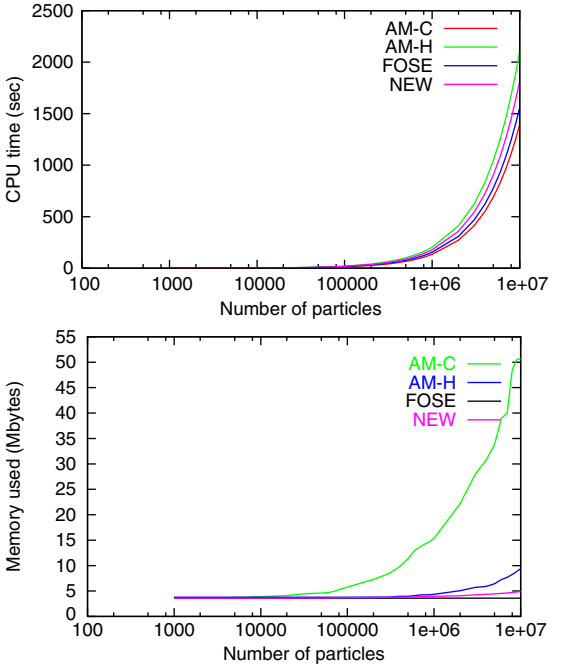

Fig. 17. CPU time and memory used by different methods.

the use of surface subdivision. But *AM-C* uses excessive memory to store the mesh structure in order to achieve this small error. Meanwhile, although the *AM-H* method avoids this storage problem, this method is impractical for a complex scene since the user would have to specify a suitable m for each surface, which is a very difficult task as explained earlier in Section 3. Finally, the error produced by our method is the smallest among the four methods, provided that reasonably many particles are traced, e.g. greater than 10,000. More importantly, our method does not require the user to specify m^*, the optimal number of terms in the illumination estimation function for each surface.

- *Running time vs. the number of particles*: as shown in Fig. 17, *AM-H* takes the longest time to trace the same number of particles, because the method has to evaluate a high-order irradiance at both the preview level and the current level for every particle hit. Our new method takes a longer time than *AM-C* or *FOSE* because it has to evaluate the function $J(m)$ for all $m \leqslant M$, in order to obtain a more accurate illumination estimation. The timing statistics were obtained on a PC with a P4 3.0 GHz processor and 1 GB main memory.
- *Running time vs. error*: Fig. 18 plots the L_2 error against the running time used in the four methods. It shows that, given the same running time, the new method attains the smallest error among the four methods. In other words, to achieve the same error level, our method requires to shoot fewer particles and runs faster.

Fig. 15 shows a synthesized image of a computer laboratory generated by the new method. The model contains 16,000 triangles. In total 10^7 particles were shot from the light sources and traced until absorbed probabilistically, up to a maximum of 10 rebounds. The computation took 2.2 h on the same PC with P4 3.0 GHz processor and 1 GB memory.

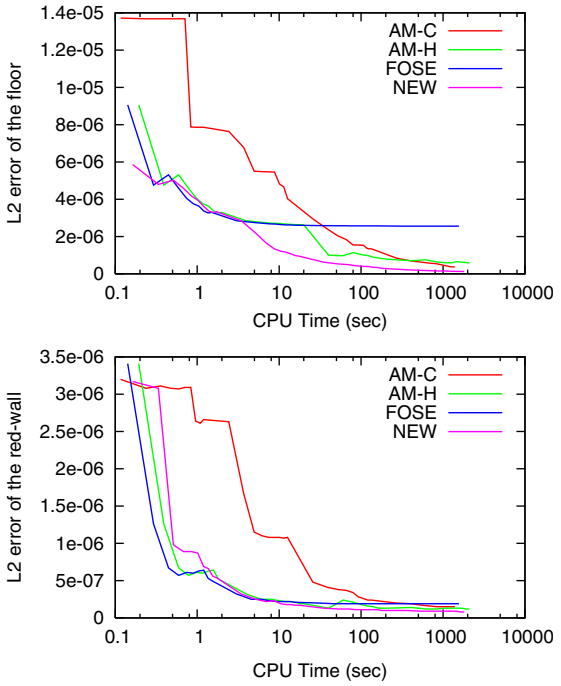

Fig. 18. L_2 error against running time.

6. Conclusions

We have presented a new method for estimating surface illumination in photon-tracing global illumination applications. The method improves the standard orthogonal series estimator in terms of two aspects. First, the new method determines adaptively and automatically the appropriate number of terms from the orthogonal series that should be used for each surface. Second, to improve accuracy, the new method adaptively subdivides a surface which has illumination discontinuity. As a result, to achieve the same error level, the new method requires shooting fewer particles, and therefore uses less memory, and runs faster. These improvements make the orthogonal series estimator more practical and usable for illumination computation.

There is still much work to be done for future research. As shown by the chair in the lower-right corner of the image in Fig. 15, tiny surfaces may not receive enough particle hits unless an extremely large number of particles are shot, which is clearly a computationally inefficient approach. Hence, an important future research problem is to find a functional representation of the irradiance that depends only on surface geometry, but is independent of a particular representation of this geometry, i.e. the underling mesh size.

There are many kinds of orthogonal basis functions, such as Fourier series and wavelets, with different approximation properties. Therefore, another problem is to study whether any of these basis functions is most suitable for photon-tracing global illumination applications.

Acknowledgements

The work of Wenping Wang was partially supported by a HKU CRCG Grant and by a National Key Basic Research Project of China (2004CB318000).

Appendix A. 2D orthogonal basis functions

A.1. Basis functions for quadrilateral surfaces

The 1D Legendre polynomials are generated by the following recursive formulas [29]:

$$P_0(x) = 1,$$

$$P_1(x) = x,$$

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x).$$

The normalized Legendre polynomials are

$$P_n^*(x) = \sqrt{n+\frac{1}{2}}P_n(x).$$

A 2D basis $\{Q_i(x,y)\}$ over a quadrilateral domain can be generated by multiplying two 1D polynomials in different variables [29]. Therefore, the first six terms in the 2D Legendre basis are

$$Q_0(x,y) = P_0^*(x)P_0^*(y), \quad Q_3(x,y) = P_2^*(x)P_0^*(y),$$

$$Q_1(x,y) = P_1^*(x)P_0^*(y), \quad Q_4(x,y) = P_1^*(x)P_1^*(y),$$

$$Q_2(x,y) = P_0^*(x)P_1^*(y), \quad Q_5(x,y) = P_0^*(x)P_2^*(y)$$

and so on. Fig. 19 shows the graphs of these six terms.

A.2. Basis functions for triangles

The basis functions given in the last section are not orthogonal over a triangular domain. However, most modeling packages in the computer graphics industry tessellate all input primitives into triangles. Hence, it is of paramount importance to consider how to use orthogonal series to represent irradiances over a triangle. In the following we briefly introduce the method given in [29] for constructing an orthonormal basis which can be used for triangular domains.

The method starts with 1D Jacobi polynomials. With arbitrary $\alpha, \beta > -1$, 1D Jacobi polynomials are generated by the following recursive formula:

$$J_0^{(\alpha,\beta)}(x) = 1,$$

$$J_1^{(\alpha,\beta)}(x) = \tfrac{1}{2}[(\alpha+\beta+2)x + (\alpha-\beta)],$$

$$\begin{aligned} J_{n+1}^{(\alpha,\beta)}(x) &= \frac{(2n+\alpha+\beta+1)(2n+\alpha+\beta+2)}{2(n+1)(n+\alpha+\beta+1)} xJ_n^{(\alpha,\beta)}(x) \\ &\quad + \frac{(2n+\alpha+\beta+1)(\alpha^2-\beta^2)}{2(n+1)(n+\alpha+\beta+1)(2n+\alpha+\beta)} J_n^{(\alpha,\beta)}(x) \\ &\quad - \frac{(n+\alpha)(n+\beta)(2n+\alpha+\beta+2)}{(n+1)(n+\alpha+\beta+1)(2n+\alpha+\beta)} J_{n-1}^{(\alpha,\beta)}(x). \end{aligned}$$

The normalized Jacobi polynomials are

$$\begin{aligned} &\hat{J}_n^{(\alpha,\beta)}(x) \\ &\quad = \sqrt{\frac{n!(\alpha+\beta+n+1)!(\alpha+\beta+2n+1)}{2^{\alpha+\beta+1}(\alpha+n)!(\beta+n)!(\alpha+\beta+n+1)}} J_n^{(\alpha,\beta)}(x). \end{aligned}$$

The Jacobi polynomials are orthogonal over the interval $[-1,1]$ with respect to the weight function $(1-x)^\alpha(1+x)^\beta$. Therefore,

$$\int_{-1}^{1}(1-x)^\alpha(1+x)^\beta \hat{J}_n^{(\alpha,\beta)}(x)\hat{J}_m^{(\alpha,\beta)}(x)\,\mathrm{d}x = \delta_{n,m}.$$

Note that the Legendre polynomials used in [24] are a special case of the Jacobi polynomials, with both α and β set to zero.

Now, with arbitrary $\alpha, \beta, \gamma > -1$, define

$$\begin{aligned} A_{n,k}(x,y) &= A_{n,k}^{(\alpha,\beta,\gamma)}(x,y) \\ &= (1-x)^k \hat{J}_{n-k}^{(2k+\beta+\gamma+1,\alpha)} \\ &\quad \times (2x-1)\hat{J}_k^{(\gamma,\beta)}\left(\frac{2y}{1-x}-1\right). \end{aligned}$$

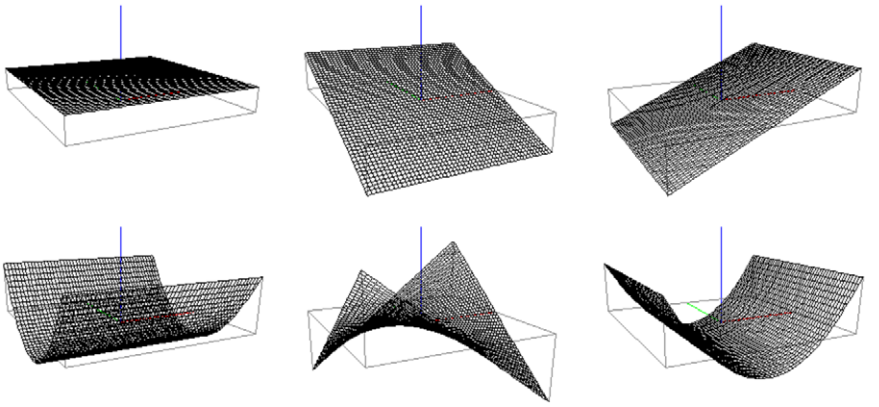

Fig. 19. The first six terms of 2D Legendre basis functions over a quadrilateral domain.

Below we show that the polynomials $\{A_{n,k}(x,y)\}$ are orthogonal over the triangular domain $T = \{(x,y) : 0 \leqslant x, y \leqslant 1, 0 \leqslant x+y \leqslant 1\}$ with respect to the weight function $x^\alpha y^\beta (1-x-y)^\gamma$.

$$
\begin{aligned}
I &= \int_{x=0}^{1}\int_{y=0}^{1-x} x^\alpha y^\beta (1-x-y)^\gamma A_{n,k}(x,y)A_{m,s}(x,y)\,\mathrm{d}x\,\mathrm{d}y \\
&= \int_{x=0}^{1} (1-x)^{k+s} x^\alpha \hat{J}_{n-k}^{(2k+\beta+\gamma+1,\alpha)} \\
&\quad \times (2x-1)\hat{J}_{m-s}^{(2s+\beta+\gamma+1,\alpha)}(2x-1) * \\
&\quad \times \left[\int_{y=0}^{1-x} y^\beta (1-x-y)^\gamma \hat{J}_k^{(\gamma,\beta)}\left(\frac{2y}{1-x}-1\right)\right. \\
&\quad \left. \times \hat{J}_s^{(\gamma,\beta)}\left(\frac{2y}{1-x}-1\right)\mathrm{d}y\right]\mathrm{d}x.
\end{aligned}
$$

Let $2y/(1-x)-1 = t$. Then

$$
\begin{aligned}
I &= \int_{x=0}^{1} (1-x)^{k+s} x^\alpha \hat{J}_{n-k}^{(2k+\beta+\gamma+1,\alpha)} \\
&\quad \times (2x-1)\hat{J}_{m-s}^{(2s+\beta+\gamma+1,\alpha)}(2x-1) * \\
&\quad \times \left[\left(\frac{1-x}{2}\right)^{\beta+\gamma+1} \int_{t=-1}^{1} (1+t)^\beta (1-t)^\gamma \hat{J}_k^{(\gamma,\beta)}(t)\right. \\
&\quad \left. \times \hat{J}_s^{(\gamma,\beta)}(t)\,\mathrm{d}t\right]\mathrm{d}x.
\end{aligned}
$$

The term inside the square brackets is zero if $k \neq s$. When $k = s$,

$$
\begin{aligned}
I &= \frac{1}{2^{\beta+\gamma+1}} \int_{x=0}^{1} (1-x)^{2k+\beta+\gamma+1} x^\alpha * \\
&\quad \times \hat{J}_{n-k}^{(2k+\beta+\gamma+1,\alpha)}(2x-1)\hat{J}_{m-k}^{(2k+\beta+\gamma+1,\alpha)}(2x-1)\,\mathrm{d}x.
\end{aligned}
$$

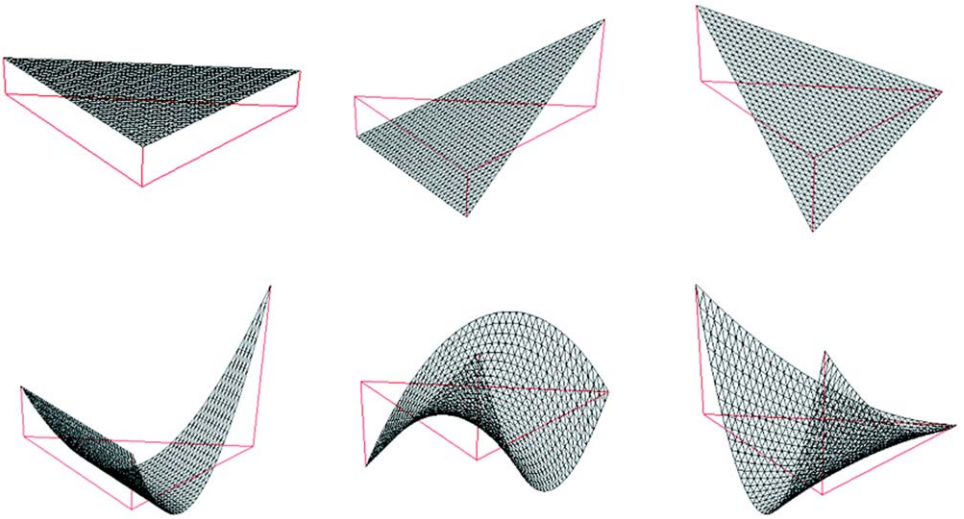

Fig. 20. The first six terms of 2D basis polynomials over a triangular domain.

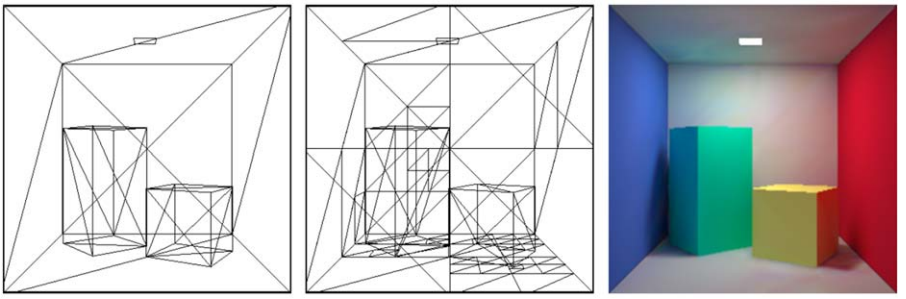

Fig. 21. Cornell box made up with triangles. From left to right: surfaces before subdivision, after subdivision, and the final rendered image.

Let $2x-1=t$. Then

$$I=\frac{1}{2^{2k+2\gamma+2\beta+\alpha+3}}\int_{t=-1}^{1}(1-t)^{2k+\beta+\gamma+1}(1+t)^{\alpha}* \times \hat{J}_{n-k}^{(2k+\beta+\gamma+1,\alpha)}(t),\hat{J}_{m-k}^{(2k+\beta+\gamma+1,\alpha)}(t)\,\mathrm{d}t = \frac{1}{2^{2k+2\gamma+2\beta+\alpha+3}}\delta_{n,m}. \quad (9)$$

To simplify computation, we can simply pick $\alpha=\beta=\gamma=0$ so that the weight function becomes 1. Also, from Eq. (9), we know that $A_{n,k}(x,y)$ can be normalized by multiplying the term $\sqrt{2^{2k+3}}$. As a result, the orthonormal basis polynomials for the triangle T are

$$\hat{A}_{n,k}(x,y)=\sqrt{2^{2k+3}}(1-x)^k\hat{J}_{n-k}^{(2k+1,0)} \times(2x-1)\hat{J}_k^{(0,0)}\left(\frac{2y}{1-x}-1\right).$$

Fig. 20 shows the graphs of the first six terms of $\hat{A}_{n,k}(x,y)$. As an illustration, Fig. 21 shows a triangulation of the Cornell box scene and the corresponding rendered image.

References

[1] Sillion FX, Puech C. Radiosity and global illumination. Morgan Kaufmann; 1994.

[2] Pattanaik SN, Mudur SP. Computation of global illumination by Monte Carlo simulation of the particle model of light, Third Eurographics workshop on rendering, 1992. p. 71–83.

[3] Feda M. A Monte Carlo approach for Galerkin radiosity. The Visual Computer 1996;12(8):390–405.

[4] Tobler RF, Wilkie A, Feda M, Purgathofer W. A hierarchical subdivision algorithm for stochastic radiosity methods. In: Eurographics rendering workshop, 1997. p. 193–204.

[5] Walter B, Hubbard PM, Shirley P, Greenberg DF. Global illumination using local linear density estimation. ACM Transactions on Graphics 1997;16(3):217–59.

[6] Myszkowski K. Lighting reconstruction using fast and adaptive density estimation techniques. In: Eurographics rendering workshop, 1997. p. 251–62.

[7] Bekaert P, Neumann L, Neumann A, Sbert M, Willems YD. Hierarchical Monte Carlo radiosity. In: Eurographics rendering workshop, June 1998. p. 259–68.

[8] Suykens F, Willems YD, Density control for photon maps. In: Rendering techniques, June 2000, p. 23–35.

[9] Volevich V, Myszkowski K, Khodulev A, Kopylov EA. Using the visual differences predictor to improve performance of progressive global illumination computation. ACM Transactions on Graphics 2000;19(2):122–61.

[10] Havran V, Bittner J. Lcts: ray shooting using longest common traversal sequences. Computer Graphics Forum 2000;19(3):59–70.

[11] Granier X, Drettakis G. Incremental updates for rapid glossy global illumination. Computer Graphics Forum 2001;20(3):268–77.

[12] Serpaggi X, Peroche B. An adaptive method for indirect illumination using light vectors. Computer Graphics Forum 2001;20(3):278–87.

[13] Szirmay-Kalos L, Csonka F, Antal G. Global illumination as a combination of continuous random walk and finite-element based iteration. Computer Graphics Forum 2001;20(3):288–98.

[14] Hey H, Purgathofer W. Advanced radiance estimation for photon map global illumination. Computer Graphics Forum 2002;21(3):541–5.

[15] Kelemen C, Szirmay-Kalos L, Antal G, Csonka F. A simple and robust mutation strategy for the metropolis light transport algorithm. Computer Graphics Forum 2002;21(3):531–40.

[16] Roland S. Bias compensation for photon maps. Computer Graphics Forum 2003;22(4):729–42.

[17] Castro F, Sbert M, Neumann L. Fast multipath radiosity using hierarchical subscenes. Computer Graphics Forum 2004;23(1):43–53.

[18] Silverman BW. Density estimation for statistics and data analysis. Chapman and Hall; 1986.

[19] Heckbert PS. Adaptive radiosity textures for bidirectional ray tracing. In: Computer Graphics (SIGGRAPH '90 Proceedings), vol. 24. 1990. p. 145–54.

[20] Chen SE, Rushmeier HE, Miller G, Turner D. A progressive multi-pass method for global illumination. In: Computer Graphics (SIGGRAPH '91 Proceedings), vol. 25. 1991, p. 165–74.

[21] Shirley P, Wade B, Hubbard P, Zareski D, Walter B, Greenberg DP. Global illumination via density estimation. In: Eurographics rendering workshop, 1995.

[22] Campbell III, AT, Fussell DS, Adaptive mesh generation for global diffuse illumination. In: Computer Graphics (SIGGRAPH '90 Proceedings), vol. 24. 1990. p. 155–64.

[23] Lischinski D, Tampieri F, Greenberg DP. Discontinuity meshing for accurate radiosity. IEEE Computer Graphics and Applications 1992;12(6):25–39.

[24] Wong KW. A new adaptive density estimator for particle-tracing radiosity. In: Pacific graphics, 2000. p. 62–70.

[25] Kronmal RA, Tarter ME. The estimation of probability densities and cumulatives by Fourier series methods. Journal of American Statistical Association 1968;63:925–52.

[26] Diggle PJ, Hall P. The selection of terms in an orthogonal series density estimator. Journal of American Statistical Association 1986;81:230–3.

[27] Rao BLSP. Nonparametric estimation of the derivatives of a density by the method of wavelets. Bulletin of Informational Cybernetics 1996;28:91–100.

[28] Gonzalez RC, Woods RE. Digital image processing. Reading: Addison-Wesley; 1992.

[29] Suetin PK. Orthogonal polynomials in two variables. London: Gordon and Breach; 1999.