Abstract

CR Categories:

Keywords:  Links:  DL  PDF

1 Introduction

Existing Monte-Carlo light transport simulation methods sample the paths either from the eye [Kajiya 1986] (eye path) or from the light source [Dutré et al. 1993] (light path). The motivation to choose these two kinds of paths is due to their predictable contributions to the final rendered results. However, although it is reasonable to sample eye/light paths, due to the complex geometry or materials in various scene settings, the spatial distribution of the sampled paths is difficult to optimum and hence lots of paths usually have to be sampled in order to reach a good convergence.

In this paper, we present a new type of intermediate path, which only bounces between the surface geometry or within the volume media. The intermediate path serves as a kind of connection path that can effectively connect the sampled eye/light paths to form a network graph for more efficient light transport. Via intermediate path, a full light transport path now can be connected by a light path, a number of intermediate paths, and an eye path. We adopt the vertex merging technique [Georgiev et al. 2012] to achieve the connection between intermediate path and the eye/light path. By successfully deriving how to compute the probability and the weight of each constructed full path using the multiple importance sampling framework [Veatch and Guibas 1995], the light transport via intermediate path is formulated into a standard path integral process [Veatch 1997]. The experimental results of rendering the surface geometry and volume media demonstrate the merits of intermediate path.

2 Related Works

Path Tracing (PT). Kajiya [1986] first formulated light transport as a rendering integral equation. The rendering integral can be evaluated by randomly sampling paths with different Monte Carlo sampling methods, e.g., path tracing [Kajiya 1986], which sample paths from the eye, or light tracing [Dutré et al. 1993], which sample paths from the light sources. Since some difficult paths with high contributions are hard to sample using purely path or light tracing, bidirectional path tracing (BPT) [Lafortune and Villermin 1993; Veatch and Guibas 1994] is proposed to sample sub-paths from both light sources and eyes and construct full paths by connecting them. BPT can be considered as a combination of several different sampling strategies (i.e. constructing a path with different length of eye sub-paths), each of which has a different weight. How to choose weights largely affects the speed of convergence. Multiple importance sampling (MIS) [Veatch and Guibas 1995] gives various weighting heuristics which largely improves the robustness of BPT. Based on the metropolis sampling method, Veach and Guibas [1997] proposed Metropolis light transport (MLT) which sample paths through random walking in the path space, and it has shown to be more effective in sampling high contribution paths. Jakob and Marschner [2012] proposed manifold exploration, which reduces the path space to a low dimensional manifold in a high dimensional space, and improves (MLT) in handling paths with specular transports. All above methods aim at producing unbiased results.

Photon Mapping. Photon mapping (PM) [Jensen 1996; Jensen 2001] traces photons from the light source into the scene in the first pass, then in the second pass, the radiance at any point in the scene can be reconstructed through a density estimation (blurring kernel) from nearby photons. Photon mapping is especially efficient in rendering SDS paths, however, it is a biased method due to the use of density estimation. The error vanishes to zero only in theory when infinite number of photons are used, which cannot be completely stored in memory. Progressive photon mapping (PPM) [Hachisuka et al. 2008; Hachisuka and Jensen 2009; Knavs and Zwicker 2011] extends the original PM algorithm to a multi-pass consistent algorithm whose error vanishes to zero in the limit. In each pass, only a limited number of photons are used and hence a limited amount of memory is needed. Hachisuka and Jensen [2011] improve PPM to render certain illumination conditions like close-up views by adaptive photon tracing. Kaplanyan and Dachsbacher [2013] improve convergence rate of PPM by adaptively choosing kernel sizes.

Virtual Point Lights. Virtual point lights (VPLs) methods [Keller 1997] are another stream of widely used biased methods. Different from photon mapping, they trace VPLs instead of photons into the scene, and the radiance at any point in the scene is computed by accumulating contributions from all VPLs, instead of density estimation. Since the cost of evaluating contributions from all VPLs is high, especially when the number of VPLs are large, various methods [Walter et al. 2005; Walter et al. 2006; Hašan et al. 2007; Ou and Pellacini 2011] have been proposed to improve the scalability of VPL methods, by only evaluating a small set of VPLs. VPL methods have also been extended to handle glossy materials [Davidić et al. 2010; Hašan et al. 2009].

MIS based methods. A recent trend is to formulate biased solutions (e.g., photon mapping and VPL) into the framework of path tracing and integration, and use multiple importance sampling (MIS) to adaptively combing light and eye sub-paths. Different from the original PM method, first, Bidirectional photon mapping (BPM) [Vorba 2011] allows eye sub-paths with various lengths, besides, they express photon density estimation as a path sampling strategy, which constructs paths by merging endpoints of light and view sub-paths. The weight of each path is adaptively determined by MIS heuristics. Vertex Connection and Merging (VCM) [Georgiev et al. 2012; Hachisuka et al. 2012] combines BPM and BPM in the same path sampling framework, where BPT and BPM are treated as different path sampling methods. They also optimally selecting paths generated by BPM and BPM using MIS heuristics, and hence retain the benefits of both BPM and BPM. Walter et al. [2012] extend VPL methods by allowing eye sub-path length larger than one. Similar to BPM, paths are constructed by connecting light sub-path vertices and eye sub-path vertices. In-
stead of fully evaluating all pairs of light and eye sub-paths, they propose bidirectional lightcuts which only consider a small fraction of pairs and maintain accuracy.

**Participating Media Rendering.** Those global illumination methods for surfaces can all be extended to handle participating media, like bidirectional path tracing [Lafortune and Willems 1996], metropolis light transport [Paugy et al. 2000], and photon mapping [Jensen and Christensen 1998]. To obtain accumulated radiance along an eye sub-path, instead of applying radiance estimation on multiple points as in volumetric photon mapping, Jarosz et al. [2008] proposed beam radiance estimate which only needs a single beam query. Various methods [Jarosz et al. 2011a; Jarosz et al. 2011b; Sun et al. 2010; Novák et al. 2012] have also been proposed to further improve participating media rendering by using lines instead of points as intermediate lighting representations.

3 Background

The Monte Carlo path tracing framework evaluates the radiance value arrived at a pixel as an integration over the space of all paths:

$$I = \int f(\bar{p}) \, d\bar{p} = \mathbb{E}\left[ \frac{f(\bar{p})}{\bar{p}} \right].$$

where \( I \) is the radiance value. \( \bar{p} = p_0 \ldots p_L \) denotes a path, starting from vertex \( p_0 \) (on a light source) and ending at vertex \( p_L \) (on the eye). Given a sampling method to randomly generate a path \( \bar{p} \), the integral can be evaluated as the expectation of the measurement contribution \( f(\bar{p}) \) of that path divided by the probability distribution \( p(\bar{p}) \) of generating it.

Bidirectional path tracing constructs a full path by connecting the endpoints of two sub-paths, i.e., a light sub-path and an eye sub-path, known as vertex connection. A light sub-path is traced from the light source while an eye sub-path is traced from the eye. E.g., denoting a light sub-path \( \bar{y} = y_0 \ldots y_s \) (\( y_0 \) is on a light source), and an eye sub-path \( \bar{z} = z_0 \ldots z_t \) (\( z_0 \) is on the eye), they can be combined to form a full path \( \bar{p} = y_0 \ldots y_s z_s \ldots z_0 \) with length \( s + t + 1 \).

Photon mapping [Jensen 2001], bidirectional photon mapping [Vorba 2011] can be also viewed as a path sampling method through vertex merging. A full path is constructed through merging the endpoints of a light sub-path and an eye sub-path, if their two endpoints are nearby. E.g., denoting a light sub-path \( \bar{y} = y_0 \ldots y_s \), and an eye sub-path \( \bar{z} = z_0 \ldots z_t \) (\( z_0 \) and \( y_s \) are nearby), they can be combined to form a full path \( \bar{p} = y_0 \ldots y_s z_s \ldots z_0 \). Note that this forms a path of length \( s + t \). These methods are biased since the merging step is an approximation.

The efficiency of photon mapping based techniques is achieved by well exploiting of path re-using. Through merging end-points of light sub-paths and eye sub-paths, a much larger number of full paths are generated efficiently (i.e. without any more path tracing or visibility tests), only at the cost of introducing some bias. However, considering a full path, only one vertex on the full path is merged (i.e. only the end vertex of both sub-paths). There is still a lot of room to further exploit path re-using, by allowing multiple vertices on a full path to be merged.

4 Our Approach

Existing bidirectional methods [Veach and Guibas 1994; Veach and Guibas 1995; Vorba 2011; Georgiev et al. 2012; Walter et al. 2012] construct a full path (i.e. from the light source to the eye) by connecting or merging endpoints from a light sub-path and an eye sub-path. To allow more reusing of paths, we propose a new type of sub-paths, called intermediate sub-path, which starts from a randomly sampled point in the scene and is followed by random bounces in the scene. Under our formulation, a full path can be constructed by sequentially joining a light sub-path, any number of intermediate sub-paths, and an eye sub-path. Details of path and sub-path definitions can be found in Section 4.1. After that, we then explain in detail how to compute the probability (Section 4.2) and the weight (Section 4.3) of each constructed full path using the multiple importance sampling framework [Veach and Guibas 1995].

**Kun:** This paragraph explains how to generate paths. Our approach proceed as follows. In the first step, we sample a number of light sub-paths. Each light sub-path starts from a point on the light source and randomly traces into the scene. This step is the same as existing methods like bidirectional photon mapping [Vorba 2011]. After that, we sample intermediate sub-paths. The start vertex of each intermediate sub-path is randomly sampled from the surface, subsequent vertices are then determined by randomly tracing taking BRDF important sampling into account. Note that, to obtain the position of starting vertices, instead of uniformly sampling on the surface, we also employ an important sampling scheme. First, we should sample more on surface locations whose light vertices are denser, secondly, we should also sample more on diffuse surfaces, than on glossy or specular surfaces. Hence, the sampling pdf at a specific position \( p \) is defined to be proportional to \( N^L_w / \rho^L_w \), where \( N^L_w \) denotes the number of light vertices around \( p \) (i.e. within a given radius), and \( \rho^L_w \) denotes the BRDF glossiness parameter of \( p \). To compute the full contribution of each pixel, a naive method is to generating eye sub-paths for each pixel, enumerate over all constructed full paths, each of which is generated through sequentially merging a light sub-path, any number of intermediate sub-paths and an eye sub-path, and then accumulate their weighted contributions using multiple importance sampling framework. However, the cost will be prohibitively high since the number of full paths is large. Instead, by relying on the fact that full paths share sub-paths a lot with each other, we propose an efficient method to compute the accumulated contributions.

To do so, before sampling any eye sub-paths, we use a path graph data structure to organize the light sub-paths and intermediate sub-paths, where each starting vertex of an intermediate path is considered as a major node, other vertices are considered as minor nodes, and each length-1 segment of a sub-path is a directed edge. We observe that the throughput (weighted contribution divided by probability) \( \text{Kun: Maybe this word "throughput" is not exactly correct for this term, we need to change its name.} \) on a full path can be decomposed as the product of sub-throughputs, and each sub-throughput is only dependent on the corresponding sub-path without the knowledge of the full path. Hence, instead of accumulating path by path, we can accumulate sub-throughput at major nodes which largely improves efficiency. Computing sub-throughput at major nodes is equivalent to a random walk on the path graph, and is formulated as a linear system, which is further solved in an iterative fashion. Details are given in Section 4.5.

After building the path graph, we now trace eye sub-paths for each pixel. For each eye vertex along the eye sub-path, the received contribution is computed by merging with vertices in the path graph. The final contribution to the eye is evaluated as a weighted sum of contributions at all eye vertices. Details are given in Section 4.6.
4.1 Types of Paths and Sub-paths

Intermediate sub-path. Light and eye sub-paths are sub-paths which start from a light source and the eye, respectively. Besides those two types of sub-paths, we define another type of sub-path, called intermediate sub-path. Intermediate sub-path starts from a surface point (starts from any point if participating media is considered) in the scene, which is randomly selected, then traces in the scene and stops according to the Russian roulette. The difference of intermediate sub-paths with light/view sub-paths only lies in the generation of the starting point. The rest processes of tracing/stopping are the same.

Constructing a full path. Suppose that we have sampled numerous light, intermediate, and eye sub-paths in the scene, let’s see how to construct a full path through these three types of sub-paths. Intuitively speaking, A full path can be constructed by concatenating a light sub-path $\bar{y}$, $k$ intermediate sub-paths ($k \geq 0$, $w_i$, $0 \leq i < k$), and an eye sub-path $\hat{a}$, i.e. $\bar{p} = \bar{y}(w)^k \hat{a}$. The two endpoints of every two concatenated sub-paths should be nearby and can be merged.

Such an example is given in Figure 1. In this example, we have three full paths constructed with different number of intermediate sub-paths. The path $y_{1,0}y_{1,1}y_{1,2}(z_3)z_2z_1z_0$ is constructed with no intermediate sub-path, and only one vertex (i.e. $y_{1,2}(z_3)$) is merged. The path $y_{0,0}y_{0,1}(w_{0,0})w_{0,1}(z_2)z_0$ has one intermediate sub-path, and two vertices (i.e. $y_{0,1}(w_{0,0})$ and $w_{0,1}(z_2)$) are merged. The path $y_{0,0}y_{0,1}(w_{0,0})w_{0,1}(z_2)w_{2,1}(z_2)z_0$ has two intermediate sub-paths, and three vertices (i.e. $y_{0,2}(w_{1,0})$, $w_{1,1}(w_{2,0})$, and $w_{2,1}(z_2)$) are merged.

For denotation consistency, we denote the state of a merged vertex (like $y_{1,2}$) as merged, and the state of of a not merged vertex (like $y_{1,1}$) as connected.

Partial Path. As shown in Figure 2, a partial path is defined as the part of a full path which excludes the eye sub-path. A partial path is constructed by concatenating a light sub-path with several (or zero) intermediate sub-paths. In the next sub-section, we will discuss how to calculate the probability and weight (i.e. for multiple importance sampling) for generating a partial path.

4.2 Probability

Given a partial path $x = x_0 \ldots x_k$, let’s see how to define the probability of generating that path. Note that the starting vertex $x_0$ is on a light source, and the ending vertex $x_k$ is merged with an eye sub-path. All other vertices ($x_1, \ldots, x_{k-1}$) can be merged or connected.

The probability density function (pdf) $p(x)$ of the partial path $x = x_0 \ldots x_k$ is defined as the joint pdf of its all vertices:

$$p(x) = \prod_{i=0}^{k} p(x_i) \quad (2)$$

$p(x_0)$ is the pdf of sampling $x_0$ on a light source. For consequent vertices ($x_1, \ldots, x_k$), since it is traced from previous vertices, its pdf is defined as:

$$p(x_i) = \begin{cases} \pi r^2 \cdot p_o(x_{i-1} \rightarrow x_i) \cdot g(x_{i-1} \rightarrow x_i) & i=k \\ \pi r^2 \cdot p_i(x_i) \cdot p_o(x_{i-1} \rightarrow x_i) \cdot g(x_{i-1} \rightarrow x_i) & x_i \text{ is merged} \\ p_o(x_{i-1} \rightarrow x_i) \cdot g(x_{i-1} \rightarrow x_i) & x_i \text{ is connected} \end{cases} \quad (3)$$

Kun: The space around the arrow symbol ($\rightarrow$) seems too large, but I didn’t figure out how to make it smaller. $r$ is the radius, and $\pi r^2$ is the size of merging area (i.e. area of density estimation). $p_i(x_i)$ denotes the pdf of sampling the starting vertex of an intermediate sub-path at $x_i$, which is set as $1/A (A$ is the total area size of all surfaces) if a uniform sampling strategy is used. The term $g(\cdot)$ converts measurement unit from solid angle to area size, and is defined as:

$$g(x_{i-1} \rightarrow x_i) = \frac{\cos \theta_{i-1} \cos \theta_i}{|x_{i-1} - x_i|^2} \quad (4)$$

The term $p_o(x_{i-1} \rightarrow x_i)$ denotes the pdf of sampling $x_i$ by tracing from $x_{i-1}$ in a solid angle unit, and is defined as:

$$p_o(x_{i-1} \rightarrow x_i) = \begin{cases} p_o(L)(x_0 \rightarrow x_1) & i=1 \\ p_o(L)(x_{i-1} \rightarrow x_i) & x_{i-1} \text{ is merged} \\ p_o(x_{i-1} \rightarrow x_{i-1}) & x_{i-1} \text{ is connected} \end{cases} \quad (5)$$

$p_o(L)(\cdot)$ denotes the pdf of sampling a direction from the light source position; $p_o(L)(\cdot)$ denotes the pdf of sampling a direction from the starting vertex of an intermediate path, and we set it as a constant $1/(2\pi)$ (i.e. uniformly sampling directions); $p_o(x_{i-2} \rightarrow x_{i-1} \rightarrow x_i)$ denotes the pdf of sampling the outgoing direction with BRDF importance sampling strategy enabled.

Combining all above equations back to Equation 2, the pdf of the whole partial path can be written as:

$$p(x) = \pi r^2 \cdot p_o(L)(x_0 \rightarrow x_1) \prod_{i=1}^{k} g(x_{i-1} \rightarrow x_i) \prod_{i=1}^{k-1} p_i(x_i) \quad (6)$$

where $p^* (\cdot)$ is defined as:

$$p^*(x_i) = \begin{cases} p_o^*(x_i) & x_i \text{ is connected} \\ p_o^*(x_i) & x_i \text{ is merged} \end{cases} \quad (7)$$
where \( p_M^*(x_i) \) and \( p_M^*(x_i) \) are defined as:

\[
\begin{align*}
  p_M^*(x_i) & = p_s(x_{i-1} \rightarrow x_i \rightarrow x_{i+1}) \\
  p_M^*(x_i) & = \pi_i^* p_i(x_{i-1}) p_{s,i}(x_i \rightarrow x_{i+1})
\end{align*}
\]  

(8)

4.3 Weight

Kun: TODO: a short review of the multiple importance sampling (MIS) framework.

A partial path \( \bar{x} = x_0 \ldots x_k \) will have \( 2^{k-1} \) different sampling methods to generate, since all inner vertices \( (x_1, \ldots, x_{k-1}) \) have two different choices: merged \( (M) \) or connected \( (C) \). Such an illustration is given in Figure 3, a partial path \( \bar{x} \) of length 3 (i.e. \( x_0 \ldots x_3 \)) have four different sampling methods.

According to the multiple importance sampling (MIS) framework, the sum of weights of all sampling methods to generate the same partial path should be equal to one. The balance heuristic defines the weight of a partial path by a specific sampling method as:

\[
\mu(\bar{x}) = \frac{p(\bar{x})}{\sum p_S(\bar{x})}
\]  

(9)

where \( S = [s_1, \ldots, s_{k-1}] \) \( s_i \in \{ C, M \}, 1 \leq i \leq k-1 \) denotes a specific sampling method. \( s_i \) denotes the state of vertex \( x_i \) (i.e. \( s_i = C \) means \( x_i \) is connected, and \( s_i = M \) means \( x_i \) is merged). By taking use of Equation 6, the weight can be rewritten as:

\[
\mu(\bar{x}) = \frac{\prod_{i=1}^{k-1} p_i^*(x_i)}{\sum_{S} \left( \prod_{i=1}^{k-1} p^*_S(x_i) \right)}
\]  

(10)

The second equality in the above equation uses polynomial simplification (e.g. \( \sum_{i=0, j=0, l=0, k=0}^{a_i/b_j/c_k} a_i b_j c_k = \frac{(a_0 + a_1)(b_0 + b_1)(c_0 + c_1)}{c_1} \)). The weight can be further rewritten as:

\[
\mu(\bar{x}) = \frac{p^*(x_k)}{p_M^*(x_i) + N_i \cdot p_M^*(x_i)}
\]  

(11)

\( \mu(x_i) \) is denoted as vertex weight, and \( N_i \) is the number of sampled intermediate sub-paths. The weight of the whole partial path is evaluated as the product of the vertex weights of all inner vertices.

4.4 Evaluating gathered radiance at an eye sub-path vertex

Given an eye sub-path \( \bar{z} = z_0 \ldots z_t \), we now explain how to compute the gathered radiance value at vertex \( z_t \) (to vertex \( z_{t-1} \)) from merging from partial paths. Using the multiple importance sampling (MIS) strategy, it can be approximated as weighted summation:

\[
I_{z_{t-1} \rightarrow z_t}^M \approx \sum_{\bar{x}} \rho_{\bar{x}} \cdot \mu(\bar{x}) \cdot \frac{f(\bar{x})}{p(\bar{x})}
\]  

(12)

where \( \bar{x} \) iterates over all partial paths whose ending vertices are closed to (or can be merged with) \( z_t \), \( \rho \) denotes the BRDF of \( z_t \), and we denote the transport as \( T(\bar{x}) = \mu(\bar{x}) \cdot f(\bar{x})/p(\bar{x}) \).

To compute Equation 12, a naive method is to traversal all possible partial paths and compute the sum. However, this can be prohibitively expensive, since the number of partial paths can be very large. In the next subsection, we will introduce a much more efficient method to evaluate the sum.
where lengths of eye sub-paths: the final contribution should mainly use the gathered radiance noises. E.g., if the first eye sub-path vertex $z_1$ is defined as:

$\rho_{1\rightarrow 0}(\bf{y}_{1\rightarrow 0})$ is defined as:

Hence, Equation 15 can also be rewritten as a matrix form $M = A M + B$. We solve $M$ from this equation through an iterative way.

Kun: TO BE MODIFIED: The symbols are not consistent with the next section.

4.6 Evaluating the final output radiance

Recall that we already know how to evaluate the radiance at a specific vertex on an eye sub-path in Section 4.4. Now we explain how to evaluate the final contribution to the eye. Note that theoretically, we can use the gathered radiance at different eye sub-paths to estimate the final contribution $I$: $I \approx I^M = \sum_{i=1}^{n}(\bf{y}_{0\rightarrow_{i-1}} \cdot f(\bf{z}_{0\rightarrow_{i-1}}))$. However, different choices have different errors. E.g., if the first eye sub-path vertex $z_1$ is almost diffuse, we should mainly use the gathered radiance $I^M_1$ at $z_1$ (i.e., eye sub-path with length $1$). On the other hand, if $z_1$ is almost specular, we should mainly use longer eye sub-paths. Hence, we estimate the final contribution $I$ as a weighted contribution from different lengths of eye sub-paths:

$\hat{I} \approx \sum_{t=1}^{\infty} w_t I^M_t$, $w_1 = \nu_1$, $w_t = (1 - \sum_{1 \leq k \leq t-1} w_k) \nu_t$ (16)

where $\nu_t$ is defined as:

$\nu_t = \min(1, \max(0, \frac{n_t^M}{\nu_{\max} - \nu_{\min}}))$ (17)

where the two thresholds are empirically set as $\nu_{\min} = 0.1, \nu_{\max} = 0.9$. $n_t^M$ denotes the number of intermediate (light) sub-path ending vertices merged with $z_1$, $n_t^f$ denotes the glossiness of $z_1$.

5 Achieving Consistency

Similar to photon mapping based methods [Jensen 2001; Vorba 2011; Georgiev et al. 2012], our method is also biased, produced from the vertex merging step. As demonstrated in the progressive photon mapping work [Knaus and Zwicker 2011], by progressively reducing the merging radius, the bias and variance of accumulated rendering results will converge to zero in the limits (i.e. which is called consistent). Now we will show that our method can also be made consistent.

Bias and Variance. Considering a merging vertex $p_i^*$ (i.e., it might be a vertex on an eye sub-path, or the starting vertex of an intermediate sub-path), denote it is merged from $k$ vertices $p_i$ ($0 \leq i < k$), the merged radiance value $L^*$ at $p_i^*$ is evaluated as:

$L = \sum_{0 \leq i < k} L_i W_i$ (18)

where $\nu_t$ denotes the starting point of the sub-path on which vertex $p_i$ is (i.e. $p_i^*$ can be the starting vertex of a light sub-path, or that of an intermediate sub-path). $L_i$ denotes the radiance of vertex $p_i^*$, and $W_i$ denotes the transport function from vertex $p_i^*$ to $p_i$, which combines BRDFs, probability, and weighting functions.

Under the above formulation, for conventional photon mapping, $p_i^*$ always denotes a vertex on the light source, and only the term $W$ will introduce bias due to the merging operator on $p_i^*$. However, in our approach, $p_i^*$ can be the starting point of an intermediate sub-path, whose radiance $L_i$ is also evaluated through merging and will also introduce bias too. The error $\varepsilon L$ of radiance $L$ can be evaluated as:

$\varepsilon L = \sum_{0 \leq i < k} ((L_i + \varepsilon L_i) \cdot (W_i + \varepsilon W_i) - L_i \cdot W_i)$

$\approx \sum_{0 \leq i < k} L_i \cdot \varepsilon L_i + \sum_{0 \leq i < k} W_i \cdot \varepsilon L_i$ (19)

where $\varepsilon L_i$ denotes the error of $L_i$, and $\varepsilon W_i$ denotes the error of $W_i$. In the above equation, we have omitted the term $\varepsilon L_i \cdot \varepsilon W_i$, since it is a high order term and can be regarded as infinitesimal.

[Knaus and Zwicker 2011] has already shown that $\text{Var}(\varepsilon W_i) = O(1/r^2)$, $\text{Bias}(\varepsilon W_i) = O(r^2)$. Suppose for all preceding vertices we also have: $\text{Var}(\varepsilon L_i) = O(1/r^2)$, and $\text{Bias}(\varepsilon L_i) = O(r^2)$. From Equation 19, it is easy to see that the variance and bias of $\varepsilon L$ also satisfies:

$\text{Var}(\varepsilon L) = O(1/r^2)$, $\text{Bias}(\varepsilon L) = O(r^2)$. (20)

Hence, using mathematical induction, it is easy to prove that error of radiance at every merging vertex satisfies the above requirements on variance and bias.

A progressive and consistent solution. Notice that the forms of variance and bias of error in our approach (Equation 20) is the same as those in photon mapping [Knaus and Zwicker 2011], hence, our approach can be made consistent using the same progressive radius reduction scheme. Specifically, in the first iteration, we use $r$ as merging radius; in subsequent iterations, we set merging radius as $r \sqrt[j]{(j \cdot \alpha)}$ (j is the iteration number, and $\alpha$ is a user specified value satisfying $0 < \alpha < 1$). In each iteration, different set of light, eye, and intermediate sub-paths are used (i.e. they are re-sampled). The final result is obtained by accumulating the rendered results of all iterations.
6 Conclusion

References


