Noname manuscript No.
(will be inserted by the editor)

Robust and Efficient Adaptive Direct Lighting Estimation

Yu-Chi Lai · Hsuan-Ting Chou · Kuo-Wei Chen · Hao-Yu Chang · Shaohua Fan

Received: date / Accepted: date

Abstract Hemispherical integral is important for the estimation of direct lighting which has major impact on the results of global illumination. This work first proposes the population Monte Carlo hemispherical integral (PMC-HI) sampler to improve the efficiency of hemispherical integral estimation. The sampler is **unbiased** and derived from the population Monte Carlo framework. The hemispherical integral sampler works on a population of samples and learns to be a better sampling function over iterations. Information found in one iteration can be used to guide subsequent iterations by distributing more samples to important sampling techniques such as sampling from light and sampling from BSDF in order to focus more efforts on the sampling sub-domain which has larger contribution to the integral. In addition, a cone sampling is proposed based on the previous successful sampling results to enhance the success rate when complex occlusions exist. In addition, we also adapt the image-plane multidimensional adaptive sampling proposed by Hachisuka et al. [15] for multidimensional hemispherical adaptive sampling (MDHI) to estimate the hemispherical integrals deterministically in the random-number direct lighting space. The images rendered with PMC-HI and MDHI are compared against those rendered with multiple importance sampling (MIS) [30] and adaptive light sample distributions (ALSD) [6]. Our PMC-HI sampler can significantly improve the rendering efficiency. Furthermore, we also show that PMC-HI can be implemented as a plugin to a modern global rendering system.

Keywords Ray-tracing · PMC · Monte Carlo · Global Illumination

Yu-Chi Lai
National Taiwan University of Science and Technology, R.O.C. E-mail: cheeryuchi@gmail.com
Funded by: NSC 99-2218-E-011-005-, Taiwan

Hsuan-Ting Chou
National Taiwan University of Science and Technology, R.O.C.

Kuo-Wei Chen
National Taiwan University of Science and Technology, R.O.C.

Hao-Yu Chang
National Taiwan University of Science and Technology, R.O.C.

Shaohua Fan
Suzhou University, China and NYU Courant Institute of Mathematical Sciences, USA. E-mail: shaohua@cs.wisc.edu
Corresponding author

1 Introduction

When rendering a physical scene, we can separate the lighting effect into two components: direct lighting and indirect lighting. Direct lighting controls the main theme of the rendered image and is the main factor of the final rendered images. The indirect lighting affects the small details of the picture such as color bleeding and is important to perceptual correctness. Traditionally, direct lighting is estimated using Monte Carlo methods but how to increase the efficiency of rendering is still the main research topics in global illumination community. This work presents a sampling technique that significantly improves rendering efficiency for direct lighting. The sampler is derived using the population Monte Carlo (PMC) sampling framework, which is a technique that adapts sampling distributions over iterations according to the information collected in previous iterations.

Traditionally, importance sampling based on the surface bidirectional reflectance distribution functions (BRDFs) [24, 21] or light sources [27, 1] (see Pharr and Humphreys [24] and Dutré [10] for an overview of these) is the primary tool for the estimation of direct lighting, or hemispheric integrals and the choice of the importance function has large

influence on the estimation. Moreover, the best importance function can vary from a shading point to another during the rendering process depending on the surface properties, related light configurations and distributions and etc. Generally, rendering community uses multiple important sampling (MIS) [30] to relieve this issue by generating samples from candidate functions which may be possibly good and then the hemispherical integral is estimated based on the balance or power heuristic weighting scheme to reduce the variance of estimation. The results are generally good but MIS may waste precious samples on those not-ideal candidate importance functions. Therefore, this work proposes the population Monte Carlo hemispherical integral (PMC-HI) sampler to dynamically adjust the choice of importance functions in order to improve sampling efficiency. The sampler iterates on a population of samples which are initialized using stratified MIS sampling on each light to estimate the initial hemispheric integrals on a point. Any information available at this stage can then be used to dynamically adapt a *kernel function* that produces a new population to approach the ideal importance functions.

There is also an adaptive light sample distribution(ALSD) research [6] proposed to dynamically adapt the sample distribution function for each light when estimating direct lighting in a many-light scene. Originally, their algorithm aims at estimating direct lighting of many lights in a scene containing almost diffuse surfaces and thus, the sampling-from-light method can generally get a low-variance estimate at a shading point. However, when a scene contains glossy surfaces, the performance of the sampling-from-light method may deteriorate with the degree of specularity at a shading point. We have improved the rendering efficiency of ALSD by combining it with MIS but sampling-from-BRDF may still waste precious samples because only one light can get hit by the shadow ray generated by the sampling-from-BRDF method when the surface is glossy and MIS sends out BRDF shadow rays for each light. Therefore, our algorithm uses a mixture of sampling from BRDF and light sources and then adapt the weight of the mixture to choose a component based on information collected through the iterations. Additionally, Hachisuka et al. [15] propose an image-plane adaptive sampling method which can consider the adaptation in the image-plane sampling process to generate deterministic result and use a numerical method to solve the path integral problem. However, the number of samples is huge and thus the time to construct and update the Kd-Tree is large and the memory to store all samples in the Kd-Tree is also enormous and therefore, the algorithm generally must divide the entire image plane into a set of tiles to run the adaptation and reconstruction independently for generating a good result. Additionally, the adaptive method can generate good results under the assumption that the integrant is deterministic. In order to have deterministic integrants, the number of dimensions for consideration has to face the curse of dimensionality i.e in order to consider an extra bounce, the dimensionality must increase by 5 (2 for surface bounce, 3 for direct lighting estimation). Section 6.1 will illustrate that we have adapted Hachisuka's algorithm proposed in [15] to estimate the hemispherical integrals called multidimensional hemispherical adaptive sampling (MDHI) by reformulating the probabilistic integral at a shading point as deterministic integral in the random-number space. When using a similar adaptive sample generation and reconstruction, the result is generally good but the cost of building a Kd-Tree and looking up a sample through the Kd-Tree is large. Therefore, when comparing the rendering efficiency with MIS, the improvement is low and even worse in hemispherical integral estimation. Our PMC-HI do not require an extra data structure in the adaptation process, has low adaptation cost and can generate good results with minimum extra cost when comparing to MDHI.

At the end, we have compared the images rendered with PMC-HI and MDHI against those rendered with MIS [30], and adaptive light sample distribution (ALSD) [6] on several scenes. PMC-HI can have a significant improvment in rendering efficiency. Additionally, PMC-HI can be easily implemented as a plugin to a general global rendering system and this work demonstrates the incorporation with direct lighting and photon mapping which is the main chosen rendering algorithm in industry. If an image-plane adaptive sampler is applied, the system can feed pixel-based coherent information to the PMC-HI sampler for even greater improvement in efficiency. We demonstrate this through the incorporation with PMC-IP [19] which can distribute pixel sampling positions iteratively. The contributions of this work is as follows:

- **Population Monte Carlo hemispheric integral sampler (PMC-HI)**: The sampler is derived from population Monte Carlo and can adjust the sampling directions used to evaluate hemispheric integrals at a point and support a variety of importance functions acting together. The sampler can, for instance, avoid over-sampling a light source from a surface point within its shadow, or a BRDF specular lobe that makes no contribution. Furthermore, the sampler can also guide samples toward important illumination directions found by previous samples, without adding bias.
- **Multidimensional hemispherical adaptive sampling sampler (MDHI)**: This sampler is derived from the multidimensional image-plane adaptive sampling method [15] by formulating the direct lighting integral as deterministic integrands in a three-random-number space. Then, the adaptation and reconstruciton presented in [15] can be directly applied to generate better samples for direct lighting estimation.

– **Incorporation with an adaptive image-plane sampler for the pixel-based coherence**: PMC-HI can be implemented as a plug-in to a general rendering system and can be incorporated with an image-plane adaptive sampler such as PMC-IP which can adjust the image-plane sample distribuiton to focus more effor in those high variance regions. In each image-plane adaptation iteration, the hemispherical sampling distribution information can be recorded and used to adjust the initial PMC-HI distribution function for exploring the spatial coherence of further acceleration.

The remainder of this paper is organized as follows: section 2 reviews a number of works related to this algorithm. section 3 gives a short overview of the population Monte Carlo method. Section 4 presents the concept and detail of the PMC-HI sampler. Section 5 describes the incorporation of the PMC-HI and PMC-IP samplers. Section 6 demonstrate the applications of the PMC-HI sampler. Section 7 gives a short discussion of using the PMC-HI sampler. Finally, Section 8 gives the conclusion of our algorithm.

2 Related Work

Direct lighting estimation is important for physically-based rendering but the computation cost may be expensive when area lights and complex occlusion are involved. There are a large number of research proposed to reduce this cost and covering all these is beyond the scope of this paper. Instead, here we focus on a specific area of related work: sampling for hemispheric integrals. For an overview of Monte Carlo rendering in general, see Pharr and Humphreys [24] and Dutré [10].

There is a large body of work on computing hemispheric integrals (direct lighting), mostly concerned with importance sampling functions. Veach's thesis [29] provides a description of the basic methods and analysis of variance. Importance functions are commonly based on surface BRDFs [24, 21] or light sources [27, 33, 23, 13, 1, 31]. However, a poor choice of importance function can *increase* variance, and, moreover, the best importance function can vary throughout a rendering depending on such things as surface properties, lighting configurations and the presence of occlusions. For example, the ideal importance function for a semi-gloss surface depends on whether the primary lobe points toward a light source, or the surface is in shadow, or neither. These configurations vary over a surface and they are difficult to discover before sampling begins, yet the choice of importance functions is typically made once and remains fixed. Recent advances include wavelet-based importance functions for environmental lighting [5], and resampling algorithms [3, 28] that avoid visibility queries for samples that are likely to be unimportant. However, the former is applicable only to environment maps, while the latter throws away samples and still requires a-priori choice of importance functions. No existing importance sampling approach for hemispheric integrals offers adaptable importance functions. This fixed choice may deviate from the optimal condition at some shading points. Multiple important sampling (MIS) [30] is the general choice for alleviating this problem by generating a sample based on light sources and the surface BRDF and the estimated direct lighting is weighed based on its sampling probability of both methods. It generally generate good results but the sampling mechanism may have chances to waste samples on the low-contribution sampling functions.

Hachisuka et al. [15] has proposed a new image-plane adaptive sampling strategy for ray tracing. Their technique is specifically designed to handle multidimensional sample domains, and it is well suited for efficiently generating images with effects such as soft shadows, motion blur, and depth of field. Since the number of samples for generating a smooth result is huge, it takes long time to construct and update the Kd-Tree and do the reconstruction for each pixel. In addition, the memory required to record the Kd-Tree is huge and thus a normal size image must be rendered in a tile style. The adaptation may miss the chance to do across the tile boundaries. Additionally, Hachisuka et al. prefer deterministic samples for the estimation but this preference must face the curse of dimensionality i.e. when a path is allowed to get an extra bounce, the dimension of the space must grow with 5 (2 for surface bounce, 3 for direct lighting estimation). Therefore, we have modified the original algorithm for estimating the hemispherical integrals at each surface point and we call it multidimensional adaptive light sampling (MDHI) which is discuss in Section 6.1. The adaptation and reconstruction mechanism is similar to the one used in [15]. The result is nice. However, the cost of building a Kd-Tree and looking up a node through the Kd-Tree is large and thus, the improvement in rendering efficiency is low. Therefore, this work proposes PMC for hemispheric integrals (PMC-HI) to dynamically adjust the choice of importance functions in order to improve sampling efficiency. The framework has low adaptation cost without the need of extra data structure and the rendered results are generally good.

Work on adaptive PDFs for importance sampling has focused on path tracing or irradiance caching applications. Dutré and Willems [8] used piecewise linear functions to determine shooting directions out of light sources in a particle tracing application. Dutré and Willems [9] use piecewise constant functions and Pietrek and Peter [25] use wavelets to build adaptive PDFs for sampling gather directions in path tracing. A diffuse surface and piecewise constant PDF assumption is required to reduce the number of coefficients to a manageable level, and even then very high sample counts are required. It is important to note that a bad approximation can *increase* variance. Lafortune and Willems [16] used a

5D tree to build an approximation to radiance in the scene, and then use it for importance sampling in a path tracing framework. The same problems with sample counts and approximation errors arise in their work. Our algorithm works with arbitrary BRDFs and uses a low-parameter adaptive model to minimize the sample count required to control adaption. Adaptive algorithms have also been suggested for shadow computations. Ward [32] proposed an algorithm for scenes with many lights, where shadow tests for insignificant lights are replaced by probabilistic estimates. Ward's approach works best with many light sources (tens or hundreds) while our technique works best with few sources. Ohbuchi and Aono [22] adaptively sampled an area light source (which introduces bias). They achieved good stratification by employing quasi-Monte Carlo (QMC) techniques to place the samples, a technique we also use.

Donikian et al. [6] introduce a new multipass algorithm for efficiently computing direct illumination in scenes with many lights and complex occlusion by computing a probability density function (PDF) for distributing samples for light-based sampling based on the accumulated pixel-level and block-level contributions and the sampling success rates. Because the algorithm mainly focuses on estimating many-light direct lighting and the scene contain large portion of diffuse surfaces, the sampling-from-light method is good to generate nice results. But when the portion of glossy surfaces grows, the sampling- from-light method may miss to sample the important feature of the BRDF and the collected samples may contain low contributions and the variance of the estimation grows. Although the light adaptation algorithm can easily incorporate with MIS method as shown in Section 6, when doing BRDF sampling for each light, the failure rate is still high and this wastes the sampling effort. Therefore, our algorithm do the adaptation in choosing the sampling strategy among BRDF and light sources to pick a good one to lower the variance.

A sequential Monte Carlo algorithm, similar in spirit to population Monte Carlo, has recently been applied by Ghosh, Doucet and Heidrich [14] to the problem of sampling environment maps in animated sequences. Their work exploits another property of iterated importance sampling algorithms – the ability to re-use samples from one iteration to the next – and is complementary to our approach. We have modified for the adaptation in the hemisphere and do the comparison among them by using component and mixture sampling.

Population Monte Carlo is another class of iterative sampling methods. The framework is introduced into rendering in [11, 12]. We have adapted the iterative adaptation mechanism for adaptively distributing image-plane samples [19, 20] and adjusting the perturbation mechanism of a Markov chain during energy redistribution process [17, 20, 18]. This work adapted from my dissertation [11] focuses on choosing a good sampling function for hemispherical integration based on the information collected in the previous iteration. Additionally, we also show that the algorithm can easily incorporate with the PMC-IP algorithm [19] to generate better results.

3 Population Monte Carlo (PMC)

The Population Monte Carlo algorithm [4] is an iterated importance sampling scheme. In this scheme, a sample population approximately distributed according to a target distribution is generated at each iteration. Then the samples from all the iterations can be used to form unbiased estimates of integrals under that distribution. It is an adaptive algorithm that calibrates the proposal distribution to the target distribution at each iteration by learning from the performance of the previous proposal distributions.

Assume we have a population of samples denoted by $\left\{X_1^{(t)},\ldots,X_N^{(t)}\right\}$, where t is the iteration number and N is the population size, and we wish to sample according to the distribution proportional to $f(x)$. The generic PMC sampling algorithm is stated in Figure 1.

```
1  generate the initial population, t = 0
2  for t = 1,···,T
3     adapt K^(t)(x^(t)|x^(t−1))
4     for i = 1,···,N
5        generate X̂_i^(t) ~ K^(t)(x|X_i^(t−1))
6        w_i^(t) = f(X̂_i^(t)) / K^(t)(X̂_i^(t)|X_i^(t−1))
7     resample according to w_i^(t) for the new population
```

Fig. 1 The generic Population Monte Carlo algorihtm.

Line 1 creates the population to jump-start the algorithm. Any method can be used to generate these samples provided that any sample with non-zero probability under f can be generated, and the probability of doing so is known.

The outer loop is over iterations. In each iteration of the algorithm, a *kernel function*, $K^{(t)}(x^{(t)}|x^{(t-1)})$, is determined (line 3) using information from the previous iterations. The kernel function is responsible for generating the new population, given the current one. It takes an existing sample, $X_i^{(t-1)}$, as input and produces a candidate new sample, $\hat{X}_i^{(t)}$, as output (line 5). The distinguishing feature of PMC is that the kernel functions are modified after each step based on information gathered from prior iterations. The kernels adapt to approximate the ideal importance function based on the samples seen so far. While this dependent sampling may appear to introduce bias, it can be proven that the result is either unbiased or consistent, depending on whether certain normalizing constants are known (in our case they are

known). The weight computed for each sample, $w_i^{(t)}$, is essentially its importance weight.

The resampling step in line 7 is designed to cull candidate samples with low weights and promote high-weight samples. Resampling is not always necessary, particularly if the kernel is not really a conditional distribution. In our PMC-HI algorithm, we did not use the resampling step.

At any given iteration, an estimator of the integral of interest is

$$\tilde{\pi}(f) = \frac{1}{n}\sum_{i=1}^{n} w_i^{(t)} f(X_i^{(t)}) \tag{1}$$

is unbiased for $\pi(f)$. To see that, we have

$$\begin{aligned} E[w_i^{(t)} f(X_i^{(t)})] &= \int\int \frac{\pi(x)}{K_{it}(x|\zeta)} f(x) K_{it}(x) dx\, g(\zeta) d\zeta \\ &= \int\int f(x)\pi(x) dx\, g(\zeta) d\zeta \\ &= \int f(x)\pi(x) dx = \pi(f) \end{aligned} \tag{2}$$

where ζ is the vector of past random variates which contribute to K_{it} and $g(\zeta)$ is an arbitrary distribution. It concludes that $\tilde{\pi}(f)$ is an unbiased estimator of $\pi(f)$.

However, in most settings, $\pi(x)$ may be known only up to a constant scale factor. Then, an estimator with a self-normalized term has to be used:

$$\hat{\pi}(f) = (\sum_{i=1}^{n} w_i^{(t)})^{-1} \sum_{i=1}^{n} w_i^{(t)} f(X_i^{(t)}) \tag{3}$$

In this case, the unbiasedness property of the estimator is lost, but $\hat{\pi}(f)$ is consistent.

At any given iteration, an estimator of the integral of interest is

$$\int_{\mathscr{D}} f(x) dx = \frac{1}{N}\sum_{i=1}^{N} w_i^{(t)} f(X_i^{(t)} \tag{4}$$

This estimator is unbiased. To see that, we have

$$\begin{aligned} E[w_i^{(t)} f(X_i^{(t)})] &= \int\int \frac{\pi(x)}{K_{it}(x|\zeta)} f(x) K_{it}(x) dx\, g(\zeta) d\zeta \\ &= \int\int f(x)\pi(x) dx\, g(\zeta) d\zeta \\ &= \int f(x)\pi(x) dx = \pi(f) \end{aligned} \tag{5}$$

where ζ is the vector of past random variates which contribute to K_{it} and $g(\zeta)$ is an arbitrary distribution. It concludes that $\hat{\pi}(f)$ is an unbiased estimator of π. However, in most settings, $\pi(x)$ may be known only up to a constant scale factor. Then, an estimator with a self-normalized term has to be used:

$$\tilde{\pi}(f) = (\sum_{i=1}^{n} w_i^{(t)})^{-1} \sum_{i=1}^{n} w_i^{(t)} f(X_i^{(t)}) \tag{6}$$

In this case, the unbiasedness property of the estimator is lost, but $\tilde{\pi}(f)$ is consistent. As with importance sampling, this estimator uses the sample weights to estimate the normalization constant, Z, of the the target distribution $\pi = \frac{1}{Z} f(x)$.

In practice, we can average over all iterations to improve the estimate, even weighting each iteration differently if we choose. A cumulative self-normalized PMC estimator over all T iterations can be defined as

$$\tilde{\pi}_\beta(f) = \sum_{t=0}^{T} \beta_t \left((\sum_{i=1}^{n} w_i^{(t)})^{-1} \sum_{i=1}^{n} w_i^{(t)} f(X_i^{(t)}) \right) \tag{7}$$

where $\beta_t, t = 0, \cdots, T$, are the weights to combine the estimates from different iterations. The optimal choices of β_t, which minimize the variance of $\tilde{\pi}_\beta(f)$, are given by [7]:

$$\beta_t^{min} = \sigma_t^{-2} / (\sum_{t=0}^{T} \sigma_t^{-2}) \tag{8}$$

where σ_t is the variance of the estimator $\tilde{\pi}(f)$ at iteration t. More details in proving the unbiasedness and analyzing the variance of PMC method is given in [2]. Several steps are required to apply PMC to rendering problems:

- Decide the sampling domain and population size. Computational concerns and stratification typically drive the choice of the domain.
- Define kernel functions and their adaption criteria. This is the most important task, and we give examples for our applications and suggest some general principles in the discussion. For rendering applications two key concerns are the degree to which the kernel supports stratification and whether it works with a small population size (as low as 4 in our hemispheric integrals sampler).
- Choose the techniques for sampling from the kernel functions and the resampling step. The deterministic sampling we use significantly reduces variance much like stratification.

The following sections describe our sampler in detail, adaptation of them into a modern rendering system, and a general discussion on PMC for rendering problems before we conclude with results.

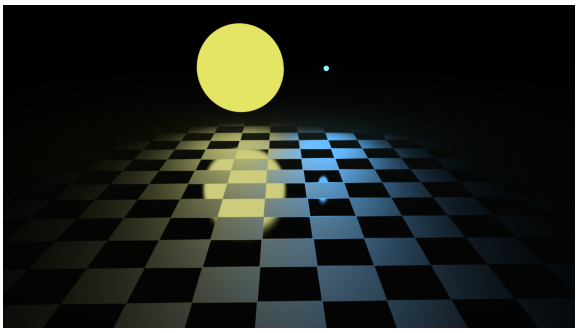
4 PMC-HI: Adaptive Hemispheric Integrals Sampling

Hemispheric samplers generate incoming directions, ω', at a surface point, $\mathbf{x}$. One application is in direct lighting, which assumes that the light leaving a surface point, $L(\mathbf{x}, \omega)$, can be evaluated by the following integral, composed of terms for light emitted from and reflected at $\mathbf{x}$:

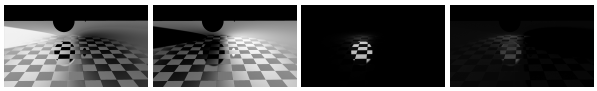
$$L(\mathbf{x}, \omega) = L_e(\mathbf{x}, \omega) + \int_{\Omega} f(\mathbf{x}, \omega, \omega') d\omega' \tag{9}$$

where $L_e(\mathbf{x}, \omega)$ is light emitted at $\mathbf{x}$, Ω is the hemisphere of directions *out* of $\mathbf{x}$ and $f(\mathbf{x}, \omega, \omega')$ is the light reflected at $\mathbf{x}$ from direction $-\omega'$ into direction ω:

$$f(\mathbf{x}, \omega, \omega') = L_{in}(\mathbf{x}, -\omega') f_r(\mathbf{x}, \omega, \omega') |\cos(\theta')| \tag{10}$$



(a) The checker scene



(b) Component sampling weights

Fig. 2 (a) is a scene constructed to demonstrate how the optimal sampling strategy varies over an image. The checkers contains diffuse and glossy squares, with near-pure specular toward the back and rougher toward the front. There are two light sources. (b) are maps which show how the mixture component weights for PMC-HI vary over the image, after two iterations. Bright means high weight. From left to right: $\alpha^{(2)}_{L1}$, the left light's weight; $\alpha^{(2)}_{L2}$, the right light's weight; $\alpha^{(2)}_{BRDF}$; and $\alpha^{(2)}_{cone}$, which can show its usage for generating more samples for glossy surfaces. The large light dominates in regions where no light is seen in a glossy reflection, while the right light is favored in nearby diffuse squares. The BRDF component is favored only when the large light is specularly reflected at a pixel. The images are quite noise-free for such small sample counts (32 total samples per estimate), indicating that the adaption mechanism converges to a consistent result.

where $L(\mathbf{x}, -\omega')$ is the light arriving at $\mathbf{x}$ from direction ω', $f_r(\mathbf{x}, \omega, \omega')$ is the BRDF, and θ' is the angle between ω' and the normal at $\mathbf{x}$.

A standard importance sampling algorithm for $L(\mathbf{x}, \omega)$ samples directions, $\{\omega'_1, \ldots, \omega'_n\}$, out of $\mathbf{x}$ according to an importance function, p, and computes the estimate:

$$\hat{L}(\mathbf{x}, \omega) = \frac{1}{n} \sum_{i=1}^{n} \frac{f(\mathbf{x}, \omega, \omega'_i)}{p(\omega'_i)} \tag{11}$$

The variance of this estimator improves as p more closely approximates f, and is zero when p is proportional to f.

In the local direct lighting situation, one common choice for p is proportional to $L_{in}(\mathbf{x}, -\omega')f_r(\mathbf{x}, \omega, \omega')|\cos(\theta')|$ or a normalized approximation to it. An alternative is to break the integral into a sum over individual light sources and sample points on the lights to generate directions [24, §16.1]. In an environment map lighting situation, the wavelet product approach of Clarberg et al. [5] currently provides the best way to choose p. However, none of these individual importance functions behaves well in all cases.

Figure 2(a) demonstrates the various difficult cases for importance sampling. The floor consists of a checker pattern with diffuse and glossy squares (with two types of gloss settings). There are two lights, one large and one small. In pixels that image diffuse squares, an importance function based on the lights is best. In highly glossy pixels that reflect the large light, BRDF sampling is best. For glossy pixels that do not reflect light, sampling from the light is best, and rough glossy pixels benefit from both BRDF and light sampling, but we have no way of knowing this a-priori, and most practitioners would use BRDF sampling. In rough glossy regions that reflect only one light, sampling from the other light is wasteful, but again most algorithms would sample equally or according to total emitted power.

Multiple importance sampling (MIS) and bidirectional importance sampling address many of these problems, by trying several importance functions and combining their results. While this does very well at reducing variance, it is wasteful in cases where one of the importance functions is much better than the others and could be used alone. Other techniques assume knowledge of which strategy will dominate where.

PMC-HI is a sampler that generates directions out of a point by adapting a kernel function to match the integrand of interest — $L_{in}(\mathbf{x}, -\omega')f_r(\mathbf{x}, \omega, \omega')|\cos(\theta')|$ in the direct lighting case. For example, the lower images in the bottom in Figure 2 indicate the relative usefulness of different importance functions at each pixel. Furthermore, the PMC framework enables important samples from one iteration to guide sampling in subsequent iterations.

4.1 The PMC-HI Kernel Function

Each direct lighting estimate takes place at a single surface point and is only one small step in a larger computation. The same surface point, and hence the same target function, f_r, essentially never re-appears. We choose to adapt on a per-estimate basis, which avoids the need to store information about the adaptation state at the surface points and interpolate to find information at new points. Hence, the number of samples on which to base adaption is low, certainly less than 100 and less than 10 in some of our examples.

A mixture distribution of a few candidate importance functions is a good starting point. At least one such component is likely to be a good approximation to f_r, and we expect to adapt to use that function most often. To catch cases where good sampling directions are hard to find, we include a component, h_{cone}, that samples based on important sample directions from the previous iteration. At one shading point, the mixture is

$$\begin{aligned} K^{(t)}_{IR}(\omega^{(t)}|\mathbf{d}^{(t)}, \beta^{(t)}) = {} & \alpha^{(t)}_{BRDF} h_{BRDF}(\omega^{(t)}) \\ & + \sum_l \alpha^{(t)}_{light_l} h_{light_l}(\omega^{(t)}) \\ & + \alpha^{(t)}_{cone} h_{cone}(\omega^{(t)}|\mathbf{d}^{(t)}, \beta^{(t)}) \end{aligned} \tag{12}$$

There is one term for the BRDF-based importance function, one for a light (or one per light for multiple lights) and

the cone perturbation function. The cone function samples a direction uniformly within a cone of directions with axis $\mathbf{d}^{(t)}$ and half-angle $\beta^{(t)}$, which is set based on the population in the previous iteration. The motivation for this term is because when complex occlusions exist, the estimation may still have a high failure rate for those samples which choose the right importance sampling functions. Therefore, it is helpful to generate samples based on those successful generated samples to increase the success rate. It is particularly useful for situations like partial shadowing where previous samples that found visible portions of the light generate more samples that also reach the light.

The population in PMC-HI is a set of sample directions out of the surface point we are estimating. The population size must be large enough to obtain reasonable estimates for the $\alpha_k^{(t)}$ values at each iteration but not so large as to increase computation times unnecessarily. We typically use $N = 2m$, where m is the number of mixture components. This is a sufficient size to see the benefits of adaption, as the results in the bottom of Figure 2 demonstrate.

4.2 Adapting for PMC-HI

An initial population of N samples, $\left\{\Omega_1^{(0)}, \ldots, \Omega_{n_0}^{(0)}\right\}$, is generated using $\alpha_{cone}^{(0)} = 0$ and the other $\alpha_k^{(0)}$ equal and summing to one. A deterministic mixture sampling is used to select the number of samples from each component. Each sample is tagged with the mixture component that was used to generate it, and their importance weights are computed:

$$w_i^{(0)} = \frac{f(\mathbf{x}, \omega, \omega')}{K_{IR}^{(0)}(\omega^{(0)})} \tag{13}$$

There is no resampling step for direct lighting. The sample size is so small that resampling tends to unduly favor high-weight directions at the expense of others, thus reducing the degree to which sampling explores the domain. Instead, the cone mixture component is used to incorporate the information from previous samples.

The new component weights, $\alpha_k^{(1)}$, can now be determined along with the $\mathbf{d}^{(1)}$ and $\beta^{(1)}$ parameters for h_{cone}. The cone direction $\mathbf{d}^{(1)}$ is found by taking a weighted average of the $t = 0$ population samples, with weights $w_i^{(0)}$. The cone size is set to the standard deviation of those samples. The component weights are set based on the sample importance weights:

$$\alpha_k^{(t)} = \frac{\sum_{i \in \mathscr{S}_k} w_i^{(t-1)}}{\sum_{j=1}^{n} w_j^{(t-1)}} \tag{14}$$

where $\mathscr{S}_k$ is the set of samples that were generated using component k. In the first iteration there is no sample from the cone perturbation, so we set $\alpha_{cone}^{(1)} = 0.2$ and adjust the other α's by a factor of 0.8 to make them all sum to one.

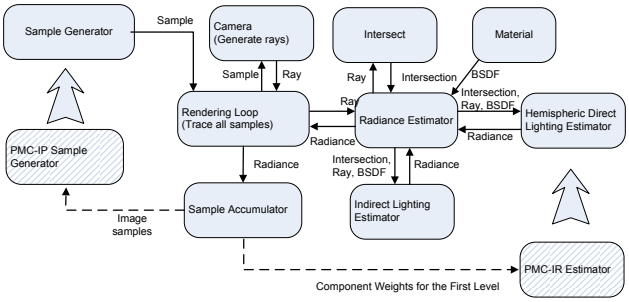


Fig. 3 A block diagram of a plug-in style Monte Carlo rendering system, following Pharr and Humphreys [24]. The PMC-IP sampler replaces a uniform sample generator with the addition of a feedback path from the sample accumulator in order to calculate the perceptual variance. The PMC-HI estimator replaces the direct lighting estimator. We can feed information from one iteration back to the next one to provide initial values for the next iteration.

We now begin the next iteration. A new set of samples is generated using deterministic mixture sampling from the kernel $K_{IR}^{(t)}(\omega^{(t)}|\mathbf{d}^{(t)}, \beta^{(t)})$, weights are computed, and the kernel function is updated based on the weights. To form the estimate, we use Equation 11, with each sample, $\Omega_i^{(t)}$, weighted by $w_i^{(t)}$ from Equation 13.

5 Integrating Samplers into a Rendering System

PMC-HI can be easily implemented as a plug-in for a general rendering system. The PMC-HI sampler is also designed to incorporate with PMC-IP [19] which iteratively distribute pixel samples according to the previous rendering results. Figure 3 shows a modern Monte Carlo rendering framework. The only core modification to the framework required to support adaptive sampling is the addition of a feedback path from the output image generator back to the samplers. This feedback is required to pass information from one sampling iteration back to the samplers for the next iteration. For the PMC-IP sampler, this feedback provides the variance map required to determine pixel sampling weights. In addition, we can improve the performance of the PMC-HI sampler through feedback.

5.1 Improving the PMC-HI with Feedback

Figure 3 shows that the PMC-HI estimator can replace the direct lighting estimator to get a better estimate at a shading point. Therefore, it can directly be used for the final gathering phase in the photon mapping algorithm. The PMC-HI sampler of Section 4 was presented in the context of a single integral estimate. On the first iteration it uses a set of default weights for each mixture component, $\alpha_k^{(0)}$. However, we can explore the pixel-based coherence through the image-plane adaptation. In a complete rendering system, we are able to

record the final adapted weights from all the direct lighting integrals done during one image-plane sampling iteration. The recorded weights can then be used to initialize the PMC-HI $\alpha_k^{(t)}$ values for the next round of sampling.

Consider a single pixel sample at image plane iteration t. A direct lighting estimate is made for the first surface point seen through the pixel location. This direct lighting estimate uses the PMC-HI sampler to adapt a set of mixture weights, which we pass back along with the estimate itself and accumulate in an auxiliary image. Other estimates may also be made for indirect illumination, but we do not record the weights from those.

At the next image-plane iteration, any direct lighting estimate required at a pixel looks to the auxiliary image to find starting weights for the PMC-HI adaption process. This avoids wasting PMC-HI iterations with un-adapted initial parameters, which improves our rendering efficiency.

6 Results

Hachisuka et al. [15] propose an image-plane adaptive sampling method for adaptively generating samples for ray tracing but the memory requirement for storing image-plane samples is too huge to compute a result with a reasonable resolution without using the tiling technique. Therefore, we adapt the concept of adaptive sampling in the hemispherical integral estimation and the resulted sampler is named as multidimensional hemispherical adaptive sampling (MDHI). This section will first give a short description of MDHI and the results rendered with PMC-HI and PMC-IP are presented and compared with other direct lighting methods.

6.1 Multidimensional Hemispherical Adaptive Sampling

In order to have multidimensional adaptive sampling, Eqn. 9 can be rewritten as

$$L(\mathbf{x},\omega) = L_e(\mathbf{x},\omega) + L_d(\mathbf{x},\omega) \tag{15}$$
$$\text{and } L_d(\mathbf{x},\omega) = \int_{\Omega} f(\mathbf{x},u_1,\cdots,u_n)du_1\cdots du_n$$

where u_1, $\cdots$ and u_n are random numbers with a value between 0 and 1, Ω is the set of all possible $(u_1,\cdots,u_n)$ and f is the contribution of the direct ligting when using the random number of $(u_1,\cdots,u_n)$. Based on the concept proposed in [15], the entire domain can be decomposed into a set of disjoin sub-domains, $\Omega_1,\Omega_2,\cdots,\Omega_N$, where $\Omega = \Omega_1 \bigcup \Omega_2 \bigcup \cdots \bigcup \Omega_N$. $L_d((\mathbf{x},\omega))$ can be estimated as follows:

$$L_d(\mathbf{x},\omega) = \sum_{\Omega_i} \int_{\Omega_i} f(\mathbf{x},u_1,\cdots,u_n)du_1\cdots du_n \tag{16}$$

By assuming that the function value within each disjoint sub-domain is constant, the integral in each disjoint sub-domain can be estimated as:

$$\int_{\Omega_i} f(\mathbf{x},u_c,u_{d1},u_{d2})du_c du_{d1} du_{d2} \approx V_{\Omega_i} f_{\Omega_i} \tag{17}$$

where V_{Ω_i} is the volume of the disjoint set Ω_i and f_{Ω_i} is the constant reconstruction of the function within the disjoint sub-domain. The radiance of the direct lighting can be computed as a Riemann sum of these estimates:

$$L_d(\mathbf{x},\omega) = \sum_{\Omega_i} \int_{\Omega_i} f(\mathbf{x},u_c,u_{d1},u_{d2})du_c du_{d1} du_{d2} \tag{18}$$
$$\approx \sum_{\Omega_i} V_{\Omega_i} f_{\Omega_i} \tag{19}$$

When implementiing MDHI, we choose three random number u_c, u_{d1} and u_{d2} to define the domain of the direct lighting sampling. There are two different types of implementation:

- u_c is used to randomly select a light source for sampling from light and u_{d1} and u_{d2} are used to select a shadow ray direction based on the sampling-from-light mechanism.
- u_c is first used to randomly choose whether sampling from BRDF or sampling from light. When samplilng from BRDF is chosen, u_c is renormalized to have a value between 0 and 1 and (u_c,u_{d1},u_{d2}) is used to select a shadow ray direction based on the sampling-from-BRDF mechanism. When sampling from light is chosen, u_c is renormalized to have a value between 0 and 1. and (u_c,u_{d1},u_{d2}) is used to choose a light and generate a shadow ray direction based on the sampling-from-light mechanism.

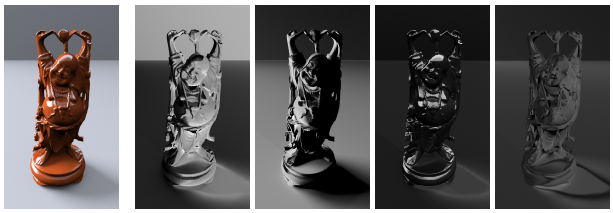
When determining the way of sampling and the direction of the shadow ray, the direct lighting contribution for the ray can be computed accordingly. After defining the sampling domain, the same Kd-Tree-based adaptive sampling and reconstruction methods proposed in [15] are used to estimate the direct lighting of a shading point.

6.2 Rendered Results and Comparisons

All statistics provided in this section are done with a personal laptop whose CPU is Intel i7-3517U 2.4 GHz with 4 GB DDR3 memory and OS is Window 7 64 bits. This section will present the results of using our PMC-HI to estimate the direct lighting of several scenes. These results are compared withmultiple importance sampling (MIS) [30], adaptive light sample distribution with sampling from light (ALSD-L) [6], adaptive light sample distribution with sampling from light and BRDF (ALSD-MIS) [6], multidimensional hemispherical adaptive sampling with sampling from light (MDHI-L), and multidimensional hemispherical adaptive sampling with sampling from light and BRDF (MDHL-MIS) for having the equal number of samples per pixel (SPP) and number

Image	Method	# SPP	# SRL	T(s)	Err	Eff
Checker	MIS	8	16	776.63	2.93e-3	0.439
	ALSD-L	4-4	16	1197.83	1.23e-2	0.0678
	ALSD-MIS	4-4	16	966.48	5.45e-3	0.189
	MDHI-L	8	16-16	3859.43	5.43e-1	4.7e-4
	MDHI-MIS	8	16-16	4278.39	5.44e-3	4.3e-3
	PMC-HI	8	4-12	846.12	1.37e-3	0.8627
	PMC-IP-HI	2-6	4-12	1048.48	3.9e-5	2.45
Budda	MIS	8	8	476.29	5.54e-3	0.379
	ALSD-L	4-4	8	1005.8	2.48e-2	0.04
	ALSD-MIS	4-4	8	786.68	1.30e-2	0.097
	MDHI-L	8	8-8	2079.71	1.33	3.62e-4
	MDHI-MIS	8	8-8	2074.77	4.25e-2	0.0113
	PMC-HI	8	3-5	516.29	2.06e-3	0.94
	PMC-IP-HI	2-6	3-5	739	8.79e-4	1.54
Jack-O	MIS	8	16	1101.09	5.87e-3	0.154
	ALSD-L	4-4	16	1271.58	4.79e-3	0.164
	ALSD-MIS	4-4	16	998.52	8.36e-3	0.12
	MDHI-L	8	16	3672.12	0.584	4.66e-4
	MDHI-MIS	8	16	4071.12	2.74e-2	8.96e-2
	PMC-HI	8	4-12	1211.12	3.79e-3	0.217
	PMC-IP-HI	2-6	4-12	1321.12	1.97e-3	0.445
YearRight	MIS	16	16	1205.58	2.81e-4	2.95
	ALSD-L	8-8	16	2013.16	1.81e-3	2.74
	ALSD-MIS	8-8	16	1502.58	1.77e-3	5.51
	MDHI-L	16	16-16	5973.21	0.636	2.63e-4
	MDHI-MIS	16	16-16	5983.92	3.98e-3	0.42
	PMC-HI	16	4-12	1325.98	1.23e-4	6.13
	PMC-IP-HI	4-12	4-12	1726.22	3.6e-5	16.09

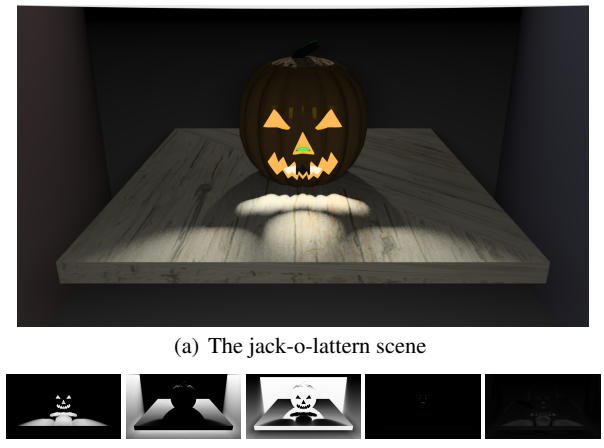
Table 1 This shows the statistics of measurements when rendering Checker, Budda, Jack-O and YearRight with PMC-HI, MIS [30], ALSD-L [6], ALSD-MIS, MDHI-L, MDHI-MIS and PMC-IP-HI which the combination of the image-plane adapting sampler, PMC-IP [19] and the hemispherical sampler, PMC-HI for having the equal number of samples per pixel (SPP) and number of shadow rays per light (SRL) equal. For the ALS and PMC-IP-HI algorithms, N_1-N_2 in the SPPS column denotes the number of samples per pixel in the first and second iterations. and ALS have two image-plane iteration. For the MDHI, PMC-IP and PMC-IP-HI algorithms, N_1-N_2 in the SRL column denotes the number of rays in the initial and adaptive iterations. T denote the time to finish rendering the entire image, Err denote the average variance average error and Eff indicates the efficiency as discussed in [29].



(a) The budda scene (b) Component sampling weights

Fig. 4 (a) is a budda scene with the ground is completely diffuse and the body is glossy. There are two light sources. (b) are maps which show how the mixture component weights for PMC-HI vary over the image, after two iterations. From left to right: $\alpha^{(2)}_{L1}$, $\alpha^{(2)}_{L2}$, $\alpha^{(2)}_{BRDF}$ and $\alpha^{(2)}_{cone}$. We can see that BRDF sampling is preferred for the specularity regions and cone sampling can help for those partial occlusion region such as the partial shadow region in the right-bottom of the image.

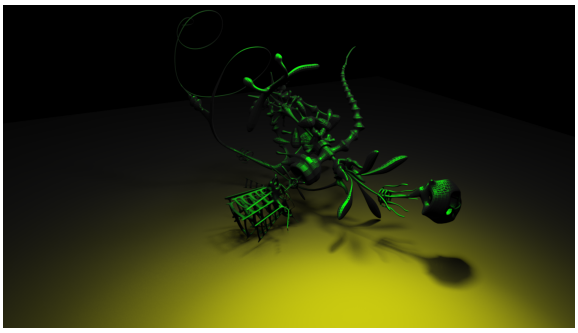
of shadow rays per light (SRL) equal. The computer runs under Windows 7 SP1 (64-bits). As shown in Figure reffig:ChessBoard, reffig:BU, reffig:JA and reffig:YR, there are four test scens: checkerboard(CK), budda(BU), Jack-O-Lattern (JA) and yearright (YR). CK, BU, JA and YR are



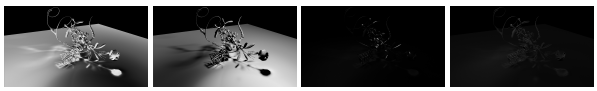
(a) The jack-o-lattern scene

(b) Component sampling weights

Fig. 5 (a) is a jack-o-lattern scenes. Majority of the objects in the scene is diffuse. There are three light sources. (b) are maps which show how the mixture component weights for PMC-HI vary over the image, after two iterations. From left to right: $\alpha^{(2)}_{L1}$, $\alpha^{(2)}_{L2}$, $\alpha^{(2)}_{L3}$, $\alpha^{(2)}_{BRDF}$, and $\alpha^{(2)}_{cone}$. The cone sampling can help to get better results for the regions on the table lighted with the light inside the jack-o-lattern and the sampling-from-BRDF is almost useless in this scene.



(a) The yearright scene



(b) Component sampling weights

Fig. 6 (a) is a yearright scene. The ground is diffuse and the decoration is glossy. There are two light sources. (b) are maps which show how the mixture component weights for PMC-HI vary over the image, after two iterations. From left to right: $\alpha^{(2)}_{L1}$, $\alpha^{(2)}_{L2}$, $\alpha^{(2)}_{BRDF}$ and $\alpha^{(2)}_{cone}$. The cone sampling can help to get better results the right-bottom partial shadow regions to get a better result.

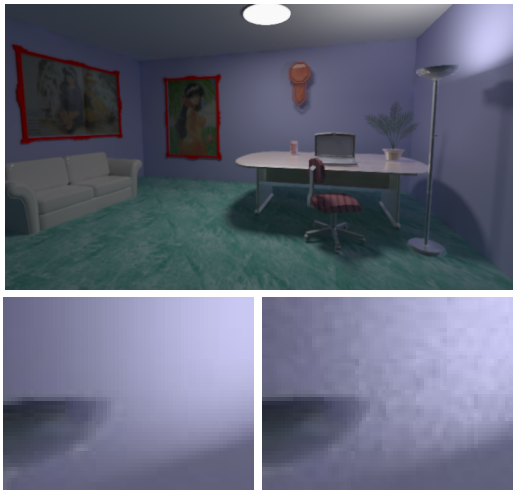


Fig. 8 The top is the result of a final gather with 4 PMC-IP iterations, with each iteration averaging 4 samples per pixel and using the PMC-HI direct lighting sampler augmented with the feedback mechanism. PMC-HI uses 2 iterations and each iterations has 16 shadow rays to estimate direct lighting. The bottom are blown up images of the upper right portion of the room scene. The left was generated with PMC-IP and PMC-HI. The right is the result of a standard photon mapping gather using 16 spps and using 16 shadow rays per light to estimate direct lighting. Note the significant reducitomn in noise with our methods.

rendered with a resolution of 1920×1080, 1080×1920, 1920×1080 and 1920×1080. The timing and the error comparisons with MIS (the best of several existing algorithms we tried on these scenes) appear in Table 1. The checker, budda, jack-o-lattern and yearright scene clearly demonstrates that adaption is a stable process that finds a good kernel function, or evenly weights the components if no component dominates (The bottom of Figure 2, 4, 5 and 6). The results show that PMC-HI and PMC-IP-HI gains an improvement in rendering efficiency over all algorithms. Figure 7 shows the cropped results of the checker, budda, jack-o-lattern and yearright. It is easy to observe that PMC operators can get much better results in the glossy regions in the chessboard scene and the partial shadow regions in the budda scene. The jack-o-lattern and yearight scenes contain a large portion of the diffuse surfaces and thus ALSD can get better results than MIS and close to the results rendered with the PMC operators. Due to the limitation of space, the complete results are presented in http://graphics.csie.ntust.edu.tw/pub/PMCHI/index.html.

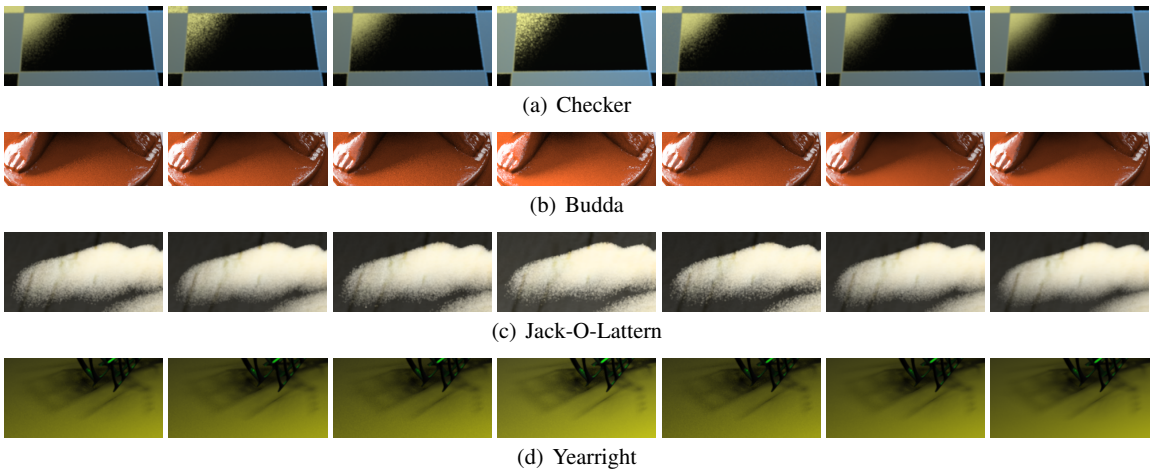
Photon mapping is an industry standard method for global illumination, and PMC-HI can be directly plug-in to estimate the direct lighting in the gather phase of a photonmapping implementation. Figure 8 shows a room scene computed with the final system. Looking at the blown-up images of right wall by the lamp, in Figure 8, we can see that our algorithm converges more rapidly to a smooth image. This is because PMC-HI obtains a better estimate of the direct lighting, and PMC-IP puts more samples in this region because of its high variance nature. Both improve the efficiency of the final result.

7 Discussion

In Equation 13 we use the full mixture distribution as the importance function, $K(\omega'_i)$. This is a form of Rao-Blackwellization, which reduces variance but at the expense of additional computation. The algorithm remains correct if we use only the mixture component from which the sample came, $h_k(\omega'^{(i)})$, and we need not compute the other mixture functions. In some cases, the resulting reduction in computation may exceed the increase in noise, but in rendering the greatest cost is usually in obtaining a sample, rather than evaluating its probabilities.

The PMC algorithms we have presented can be viewed as generalizations of the existing algorithm, MIS. MIS is a special case of deterministic mixture sampling. It corresponds to fixing the α_k weights ahead of time, which fixes the number of samples from each function. The MIS balance heuristic results in the same estimator that we use. We improve upon MIS by adapting the weights over time, which avoids wasting samples on unimportant component functions.

Many PMC kernels in the literature are mixture models. Mixtures are typically formed by combining several components that are each expected to be useful in some cases but not others. The adaption step then determines which are useful for a given input. Mixtures allow otherwise unrelated functions to be combined, such as the light area importance function and the BRDF importance function in Equation 13. If an environment map was present, we could even include



(a) Checker

(b) Budda

(c) Jack-O-Lattern

(d) Yearright

Fig. 7 (a) is the cropped results of the checker scene. (b) is the cropped results of the budda scene. (c) is the cropped results of the jack-o-lattern scene. (d) is the cropped results of the yearright scene. Images from left to right in each column are rendered with MIS, ALSD-L, ALSD-MIS, MDHI-L, MDHI-MIS, PMC-HI and PMC-IP-HI.

the wavelet importance functions of Clarberg et al. [5] in the mixture. Typically, the common rule of choosing importance functions applies here also: when f is a product of several unrelated functions, then a good choice of mixture components is something proportional to each factor.

Other adaptable functions can be used as kernel functions, such as Gaussian distributions parameterized by their standard deviation. Such a choice would be appropriate if a single Gaussian of unknown size was thought to be sufficient even when acting alone, but the ability to simultaneously sample from several functions is lost. The most common reason to use non-mixture kernels is when the aim of PMC is finding the adapted parameters themselves, not the samples, and hence the kernel function is chosen to represented the expected form of the underlying statistical model.

The most notable limitation of PMC is the high sample counts required when the kernel has many adaptable parameters. This precludes, for instance, using one component per light when there are many lights. Such a strategy would be appealing for efficiently sampling in complex shadow situations (some components would see the lights, others would not), but the sample count required to adequately determine the mixture component weights would be too large. Instead we use a single mixture component for all the lights and rely on the cone perturbation component to favor visible lights. This does not work well if the illumination sources are widely spaced.

In Section 5 we presented an image space method for sharing adapted parameters across different estimates. An alternate approach for integrating functions defined on surfaces is to store the mixture component weights in a surface map and interpolate. This amortizes the cost of adapting over many surface points. We did not explore this possibility, but it offers potential for the multi-light problem or cases where many light transport paths must be constructed through a scene, such as bi-directional path tracing or photon mapping.

8 Conclusion

We have shown how the hemispheric integral estimation can be derived from the PMC framework. The algorithm learns to be an effective sampler based on the results generated in early iterations. This alleviates one of the greatest problems in Monte Carlo rendering: the choice of importance functions and other parameters. Although this work currently only applies the PMC-HI sampler to estimating the direct lighting, the sampler could be used in any situation where estimates of an integral over the hemisphere are required. For example, irradiance caching would possibly benefit greatly from the sampler in the computation of each cached value. The PMC-HI direct lighting estimator is implemented as a plug-in and we have shown that it can be easily to incorporate with photon mapping in the final gathering phase to estimate the direct lighting. PMC is just one approach from the family of iterated importance sampling algorithms [26]. The Kalman filter is another well-known example. Common to these techniques is the idea of sample reuse through resampling and the adaption of sampling parameters over iterations. Computer graphics certainly offers further opportunities to exploit these properties.

References

1. Agarwal S, Ramamoorthi R, Belongie S, Jensen HW (2003) Structured importance sampling of environment maps. In: SIGGRAPH '03, pp 605–612

2. Anonymous (2006) Sequential monte carlo methods for physically based rendering. PhD thesis, Anonymous University
3. Burke D, Ghosh A, Heidrich W (2005) Bidirectional importance sampling for direct illumination. In: Proc. of the 16th Eurographics Symposium on Rendering, pp 147–156
4. Cappé O, Guillin A, Marin JM, Robert C (2004) Population Monte Carlo. Journal of Computational and Graphical Statistics 13(4):907–929
5. Clarberg P, Jarosz W, Akenine-Möller T, Jensen HW (2005) Wavelet importance sampling: efficiently evaluating products of complex functions. In: SIGGRAPH '05, pp 1166–1175
6. Donikian M, Walter B, Bala K, Fernandez S, Greenberg DP (2006) Accurate direct illumination using iterative adaptive sampling. IEEE Transactions on Visualization and Computer Graphics 12(3):353–364
7. Douc R, Guillin A, Marin JM, Robert CP (2005) Minimum variance importance sampling via population Monte Carlo. Technical report, University Paris Dauphine
8. Dutré P, Willems YD (1994) Importance-driven Monte Carlo light tracing. In: Proc. of the 5th Eurographics Workshop on Rendering, pp 185–194
9. Dutré P, Willems YD (1995) Potential-driven Monte Carlo particle tracing for diffuse environments with adaptive probability functions. In: Proc. of the 6th Eurographics Workshop on Rendering, pp 306–315
10. Dutre P, Peter Shirley and KB, Bekaert P (2006) Advanced Global Illumination. A K Peters Ltd
11. Fan S, Chenney S, Hu B, Tsui KW, Lai YC (2006) Optimizing control variate estimators for rendering. Computer Graphics Forum 25:351–358
12. Fan S, Lai Y, S C (2007) Population monte carlo samplers for rendering. Technical Report TR-1613, University of Wisconsin
13. Fernandez S, Bala K, Greenberg DP (2002) Local illumination environments for direct lighting acceleration. In: Proceedings of the 13th Eurographics workshop on Rendering, EGRW '02, pp 7–14
14. Ghosh A, Doucet A, Heidrich W (2006) Sequential sampling for dynamic environment map illumination. In: Proc. Eurographics Symposium on Rendering, pp 115–126
15. Hachisuka T, Jarosz W, Weistroffer RP, Dale K, Humphreys G, Zwicker M, Jensen HW (2008) Multidimensional adaptive sampling and reconstruction for ray tracing. ACM Trans Graph 27(3):33:1–33:10
16. Lafortune EP, Willems YD (1995) A 5D tree to reduce the variance of Monte Carlo ray tracing. In: Proc. of the 6th Eurographics Workshop on Rendering, pp 11–20
17. Lai Y (2010) Photorealistic animation rendering with population monte carlo. PhD thesis, University of Wisconsin at Madison
18. Lai Y, Liu F, Zhang L, Dyer C (2008) Efficient schemes for monte carlo markov chain algorithms in global illumination. In: International Symposium on Visual Computing
19. Lai Y, Chenney S, Liu F, Niu S Yand Fan (2010) Animation rendering with population monte carlo image-plane sampler. The Visual Computer pp 543–553
20. Lai Y, S F, C D (2010) Population monte carlo path tracing. Technical Report TR-1614, University of Wisconsin
21. Lawrence J, Rusinkiewicz S, Ramamoorthi R (2004) Efficient brdf importance sampling using a factored representation. ACM Trans Graph 23(3):496–505
22. Ohbuchi R, Aono M (1996) Quasi-Monte Carlo rendering with adaptive sampling. Technical Report RT0167, IBM Tokyo Research Laboratory
23. Paquette E, Poulin P, Drettakis G (1998) A light hierarchy for fast rendering of scenes with many lights. Computer Graphics Forum 17:63–74
24. Pharr M, Humphreys G (2004) Physically Based Rendering from Theory to Implementation. Morgan Kaufmann
25. Pietrek G, Peter I (1999) Adaptive wavelet densities for Monte Carlo ray tracing. In: Skala V (ed) WSCG'99 Conference Proceedings, pp 217–224
26. Robert CP, Casella G (2004) Monte Carlo Statistical Methods, 2nd edn. Springer-Verlag
27. Shirley P, Wang C, Zimmerman K (1996) Monte Carlo techniques for direct lighting calculations. ACM Transactions on Graphics 15(1):1–36
28. Talbot J, Cline D, Egbert P (2005) Importance resampling for global illumination. In: Proc. of the 16th Eurographics Symposium on Rendering, pp 139–146
29. Veach E (1997) Robust Monte Carlo methods for light transport simulation. PhD thesis, Stanford University
30. Veach E, Guibas LJ (1995) Optimally combining sampling techniques for Monte Carlo rendering. In: SIGGRAPH '95, pp 419–428
31. Wald I, Benthin C, Slusallek P (2003) Interactive global illumination in complex and highly occluded environments. In: Proceedings of the 14th Eurographics workshop on Rendering, EGRW '03, pp 74–81
32. Ward G (1991) Adaptive shadow testing for ray tracing. In: Proc. of the 2nd Eurographics Workshop on Rendering, pp 11–20
33. Zimmerman K, Shirley P (1995) A two-pass realistic image synthesis method for complex scenes. In: IN EUROGRAPHICS WORKSHOP ON RENDERING, pp 284–295