Volume xx (200y), Number z, pp. 1–11

A Significance Cache for Accelerating Global Illumination

paper1093

Abstract

Rendering using physically-based methods requires substantial computational resources. Most methods that are physically-based use straightforward techniques that may excessively compute certain types of light transport, while ignoring more important ones. Importance sampling is an effective and commonly used technique to reduce variance in such methods. Most current approaches for physically-based rendering based on Monte Carlo methods sample the BRDF and cosine term, but are unable to sample the indirect illumination as this is the term that is being computed. Knowledge of the incoming illumination can be especially useful in the case of hard to find light paths, such as caustics or scenes which rely primarily on indirect illumination. To facilitate the determination of such paths, we propose a caching scheme which stores important directions, and is analytically sampled to calculate important paths. Results show an improvement over BRDF sampling and similar illumination importance sampling.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Display Algorithms; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Ray Tracing; I.6.8 [Simulations and Modeling]: Types of Simulation—Monte Carlo;

1. Introduction

Global illumination algorithms simulate the propagation of light within a virtual scene by solving the rendering equation [Kaj86]:

$$L_o(x,\omega_o) = L_e(x,\omega_o) + \int_{\Omega_{2\pi}} f_r(x,\omega_i,\omega_o) L_i(x,\omega_i)(n\cdot\omega)d\omega_i \quad (1)$$

where $L_o(x,\omega_o)$ is the outgoing radiance at a point x in the direction ω_o, which is the quantity we want to measure. $L_e(x,\omega_o)$ is the emitted radiance from the point. The function f_r is the BRDF, L_i is the incoming radiance and n is the surface normal. The rendering equation is typically solved using numerical methods, such as Monte Carlo integration, a powerful technique for evaluating integrals of high dimensions. Importance sampling is an effective variance reduction technique for accelerating Monte Carlo integration by sampling from a distribution that approximates the integrand as closely as possible, with the ideal being proportional. For rendering this is often difficult to achieve, so most commonly used approaches draw samples from the BRDF and the cosine term (or the light source in the case of direct lighting). However, global illumination calculations have to take into account the multitude of lighting situations that arise even in simple scenes. As light propagates around a scene, paths which are difficult to find commonly contribute a great deal to image quality. Examples of these paths are difficult indirect paths or caustics paths. For such complex paths, information about the incoming radiance can be a more important sampling factor than the cosine and the BRDF.

In this paper, we introduce a method that permits improved path generation via importance sampling based on a cached approximation of the incoming radiance (L_i), which is also combined with the other terms in Equation 1. This is made possible by the use of a caching structure we term the Significance Cache (SC), which stores the sampled radiance from previous computations, and re-uses them to sample new directions in an unbiased manner. The SC is inspired by the irradiance/radiance cache methods [WRC88] [KGPB05] but is not directly used for image generation. Using our method, whenever a new sample is required, the SC is consulted to identify important directions. The directions chosen are based on sampling the cached distribution, such that the choice of directions remains unbiased. While there exist other methods that sample the product distribution, our method differs in that we store the incoming radiance in an adaptive caching structure, based on cosine lobes, which do not require discretization of the hemisphere. The SC adapts to the incoming radiance of the scene as the rendering is computed, and new, important directions are used to update current samples, unlike methods which rely on a pre-pass, such as the technique presented by Jensen [Jen95]. The use of the SC does not require any pre-computation, and is used on the fly. The SC is not specific to any rendering algorithm, and can be used for path generation in conjunction with most rendering methods, such as path tracing [Kaj86] and Metropolis Light Transport (MLT) [VG97]. Moreover, unlike other screen space importance sampling methods [PBSP08], the SC is temporally coherent and can be used in subsequent frames and at different viewpoints. In this paper we present the SC as an acceleration structure, demonstrate how to obtain samples from it, and state how the cached records are updated. Furthermore, we describe how it can be used in conjunction with rendering algorithms, demonstrate results for SC used to accelerate path tracing, including path tracing with participating media, and show results for the SC used in conjunction with MLT and Progressive Photon Mapping [HOJ08].

2. Related Work

Most importance sampling techniques aim at sampling the product of the BRDF and cosine term in the rendering equation [Kaj86]. However some have taken the approach of sampling the product of the BRDF, incoming radiance and cosine term. Multiple Importance Sampling [VG95] is one such method, where it was shown that samples drawn from multiple distributions could be weighted via simple heuristics in order to optimally combine them.

Many of the methods that sample multiple distributions are often targeted at direct illumination from high dynamic range environment maps. Such approaches are generally referred to as Bidirectional Importance Sampling techniques. Burke et al. [BGH05] demonstrated that rejection sampling or sampling importance resampling [TCE05] could be used to generate samples proportional to the product of the BRDF and environment lighting. Samples are initially drawn from the higher frequency component, and are then weighted by the other component in order to generate a better distribution of the samples. Wang et al. [WÅ09] used lightcuts and clustering to draw samples from a direct lighting distribution. Their method relies on taking several samples from the BRDF, and the weighting probability of sampling a cluster from the lightcut by the proportion of BRDF samples which lie within the bounding volume of the cut.

2.1. Pre-pass methods

Several methods have been proposed which apply a pre-pass to generate an approximate lighting distribution which can guide sampling. Jensen [Jen95] used the directions of the photons in the photon map to guide importance sampling. As a first pass, this method shoots photons from the light sources and performs a random walk for each photon. The photons are then stored in the scene in a kd-tree. These photons are then exploited in a second pass to suggest sampling directions for path tracing. At each intersection point during the path tracing pass, nearby photons are gathered from the kd-tree, and weighted by their BRDF and power. These are used to create a discrete PDF over the hemisphere, from which samples are drawn, so that more rays are shot towards important areas of the scene. We compare our method with this technique in Section 4. Szirmay-Kalos et al. [SKCP99] used a similar method, except the photons were stored on patches in the scene. Hey et al. [HP02] extended Jensen's technique by splatting the particles onto a hemisphere and sampling proportional to these photon splats. This method is shown in the context of final gathering. These techniques have the disadvantage in that they do not adapt, based on the new information gathered as the computation proceeds. Budge et al. [BAJ08] shot photons through caustic objects, and clustered them in order to improve importance sampling in a subsequent path tracing pass. A 5D tree based method was also presented by [GXS09] who created the tree with a finite number of nodes at fixed positions in the scene, based on a particle tracing pass. Each node contains a spherical tree of directions, which is sampled, and subdivides based on incoming radiance samples during rendering. This has the issue that if the particle tracing pass is not adequate to capture the lighting, the structure may provide less efficient importance sampling.

Similar techniques have been used to guide photon mapping [Jen96]. Peter et al. [PP98] and Keller et al. [KW00] shot importance particles (importons) from the eye, and in a subsequent pass, used the gathered information to send photons to areas of high visual importance.

2.2. Adaptive Methods

Methods have been proposed to draw samples from the product of the terms in the rendering equation in a more adaptive context. LaFortune et al. [LW95] used an adaptive 5D tree to store incoming radiance. When a sample is requested, several samples drawn initially from the BRDF are weighted by the stored incoming radiance values (created by traversing the tree, and discretizing the values in the hemisphere), and one of these samples is picked based on the weight. Although this weights the samples based on the product, peaks in the stored radiance distribution may be missed due to the fact that samples were only taken from the BRDF. This method was recently extended to participating media by Pegoraro et al. [PWP08], who stored the incoming radiance as hierarchical low order B-Splines. Both of these methods rely on very specific refinement criteria, and can face a large memory consumption ([PWP08] reported use of up to a gigabyte of memory). Image space approaches have been proposed, such as Pegoraro et al. [PBSP08] and Cline et al. [DCE08], however these are limited to a single bounce of global illumination.

In the context of light tracing, Dutré and Willems [DW94] used importance sampling at the light source to send particles in visually important directions. They also tackled the problem of importance sampling in a particle tracing context [DW95]. They stored an adaptive piecewise constant PDF at discrete patches in the scene, and applied this to guide the walk from the light source. This has a disadvantage that the structure used cannot refine in areas of high frequency potential, and is wasteful in areas of low frequency.

2.3. Other sampling approaches

Other approaches have tackled similar issues. Hachisuka et al. [HJW*08] used adaptive sampling over higher dimensions to more efficiently render effects such as motion blur and depth of field. For direct lighting, approaches such as Shirley et al. [SWZ96] have shown how sampling distributions can be designed to improve efficiency. A visibility caching scheme for direct lighting by Clarberg and Akenine-Möller. [CAM08] used a low resolution visibility approximation at various image locations as control variates to reduce variance. Other techniques besides importance sampling have also been applied, for example Szirmay-Kalos et al. [SKS09] used error diffusion to generate samples.

2.4. Cosine Lobes

Cosine lobes have been used in graphics to represent distributions. Phong [Pho75] represented the glossy component of a BRDF by a cosine lobe, an idea that was made physically correct by Lewis [Lew94]. LaFortune et al. [LFTG97] also used several cosine lobes to represent a BRDF. All the components of the rendering equation were approximated as cosine lobes by Meunier et al. [MPA*10], who used an approximation of the product of several lobes when shading. However, this cannot be used in an unbiased context due to the approximations in the model, that radiance is assumed to come only from spherical light sources, and also that this requires the BRDF to be represented as a set of cosine lobes, which is often unfeasible in rendering systems.

3. The Significance Cache

The central idea behind the SC is to cache information about incoming indirect radiance to an area of the scene. The SC can also be used to store importance in the context of a light tracer (shoots rays from the light, instead of the eye). The design of such a structure ideally satisfies two often opposing features:

1. To provide a structure which allows the sampling of important directions in the scene, preferably proportional to their contribution.
2. To allow further exploration of the scene to ensure that all the above important directions are found.

The above often conflict, as the exploration of the scene (via a method such as BRDF or cosine sampling) lessens the desirable effects of sampling important directions. Whereas if only a few directions are sampled, the lack of exploration may lead to an inaccurate representation of the incoming radiance in a scene. Memory consumption should also be minimized. Many caching schemes have a potentially large memory footprint. We aim to minimize this by storing incoming radiance in a sparse representation.

The following sections describe how the SC meets these goals, what information is stored in the SC, and how it is updated, and sampled. Finally, we discuss the application of SC to participating media.

3.1. Overview

Algorithm 1 shows an overview of the SC's function. The SC is queried each time a decision on which direction to sample is required. Initially, a search is performed (line 1) to identify the relevant cache points within the vicinity of the shading point x. If these are identified, the SC is used to choose a direction (lines 4 to 14) ω_i (see Section 3.3). If no cache points were found, some other criteria is used for sampling (line 17), such as the BRDF (sampleBRDF()). Then in lines 22 to 44, the SC is then updated with the new information from the sample, that is, incoming radiance and direction from shootRay() in line 20 (see Section 3.4).

3.2. Representation

The SC stores the incoming illumination in a scene as a relatively sparse set that is capable of representing this information (that is positional, directional and incoming radiance components). There are many choices regarding how to store directionally varying quantities, such as hierarchies [LW95] and spherical harmonics [KGPB05]. We have chosen a differing route, that is, to store the incoming radiance via cosine lobes. These are well understood, and are steerable with intuitive parameters. The generalized cosine lobe model is similar to the LaFortune BRDF [LFTG97] and the cosine lobe rendering model [MPA*10]. In the LaFortune BRDF, the cosine lobes are fitted to data as a preprocess, and in the rendering model many assumptions are made about the geometry and light sources, and this does not take into account indirect lighting. In our system, lobe adaptation has to be continual, and has to deal with minimal data per step (i.e. incoming direction ω_i and the incoming radiance that we want to store L_i). We choose to store the incoming radiance as cosine lobes for three main reasons:

1. Cosine lobes can be easily and analytically sampled.
2. They can be made adaptive in an intuitive manner.
3. They have a very low memory cost for storage.

As mentioned above, spherical harmonics are also an option. We chose not to use them for two main reasons. Firstly, cosine lobes use few, very intuitive parameters and can easily represent highly directional effects. Highly directional lighting (for example a caustic) can be represented by a small amount of data in the cosine lobe representation; whereas spherical harmonics may need many coefficients to represent the same data. The second reason is that by using spherical harmonics, the entire lighting function is forced to be represented at a set of points in the scene. The cosine lobe representation decouples the direction of the lighting from both position and geometry, which effectively spreads this information to a local area of the scene. This allows lighting effects which may be missed by some lobes to be captured by other lobes in the local neighborhood.

The data structure of the SC consists of records stored in an acceleration structure for rapid access. Each cache record (c) consists of:

- The position in world space, $\tilde{x} \in \mathbb{R}^3$.
- The world space direction, $\tilde{\omega} \in \mathbb{R}^2$ for which samples are to be drawn from (represented in a spherical coordinate system).
- The average incoming radiance relevant to the lobe, $\tilde{L}$, either stored as an RGB or spectral quantity, or a scalar value such as luminance, where $\tilde{L} \in \mathbb{R}^3$ or $\tilde{L} \in \mathbb{R}$ respectively.
- The average radiance for all samples at the cache point $\tilde{I}$, where $\tilde{I} \in \mathbb{R}^3$ or $\tilde{I} \in \mathbb{R}$ depending on if RGB or luminance is used.
- The exponent of the cosine lobe, $e \in \mathbb{R}$.
- The total number of samples contributing to the cache point $S \in \mathbb{N}$.
- Useful sample count $Sr \in \mathbb{R}$.
- Total sample count $Sa \in \mathbb{R}$.

The reasons for storing this information are explained in the following sections.

3.3. Sampling

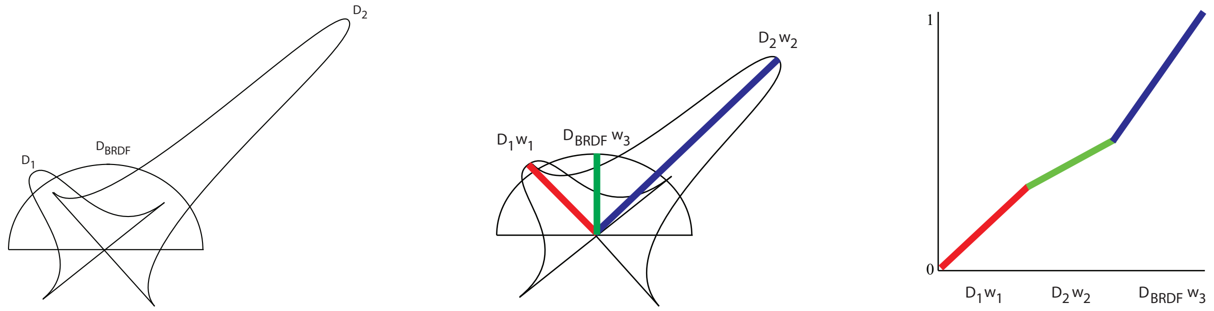
In this section we give details of the sampling procedure outlined in lines 4 to 14 in Algorithm 1. The overall goal is to allow importance sampling to be carried out, which is roughly proportional to the product of the BRDF, incoming luminance and the cosine falloff factor. This is accomplished via three steps. The first builds a distribution which approximates the incoming radiance. The components of this distribution are then weighted to better approximate the rendering equation (Equation 1). Finally, a component of this distribution is selected, and a new sample (direction) is generated. These steps are discussed further below.

3.3.1. Distribution

Our sampling mechanism draws samples from a weighted set of oriented cosine lobes gathered around a shading point x (see Figure 1(a)). We also include a weight for a distribution that is defined over every part of the hemisphere in order

Input: context (position, orientation, surface)
1 $cachePoints \leftarrow searchSC(context)$
2 **if** $cachePoints$ **then**
3   `// Use cache points found in SC`
4   **for** $i \leftarrow 1$ **to** $cachePoints$ **do**
5         $\psi \leftarrow orientation \cdot \tilde{\omega}_i$
6         $w_i \leftarrow BRDF * \psi * \tilde{L}_i$
7   **end**
8     $w_{cachePoints+1} \leftarrow BRDFWeight$
9     $lobeCDF \leftarrow createCDF(w)$
10     $lobe \leftarrow sampleCDF(lobeCDF)$
11     $\omega_{in} \leftarrow sampleLobe(lobe)$
12   **for** $i \leftarrow 1$ **to** $cachePoints$ **do**
13         $p_i \leftarrow (\omega_i \cdot \omega_{in})^{\lceil e_i \rceil} / V_i$
14   **end**
15     $p \leftarrow (w_{cachePoints+1}) p(BRDF)) + \sum_{i=1}^{cachePoints} w_i p_i$
16 **else**
17   `// No cache points found`
18     $\omega_{in} \leftarrow sampleBRDF(context)$
19     $p \leftarrow p(BRDF)$
20 **end**
21 $L_{in} \leftarrow shootRay(in\ \omega_{in}\ direction)$
22 `// Update cache points found in SC`
23 $useful\_lobes \leftarrow 0$
24 **for** $i \leftarrow 1$ **to** $cachePoints$ **do**
25     $S_i{+}{+}$
26     $s \leftarrow max((\omega_i \cdot \omega_{in})^{e_i}, 0)$
27   **if** $T(\tilde{L}_i, L_{in}) == 1$ **then**
28         $Sr_i \leftarrow Sr_i + (1 - s)$
29   **end**
30     $Sa_i \leftarrow Sa_i + (1 - s)$
31     $e_i \leftarrow \max(\frac{Sa_i}{Sr_i} - 1, 1)$
32     $\tilde{I}_i \leftarrow \frac{(S_i - 1)\tilde{I}_i + L_{in}}{S_i}$
33   **if** $L_{in} < I_i$ **then**
34         $useful\_lobes{+}{+}$
35   **else**
36   **if** $s \geq 0.5$ **then**
37             $\tilde{L}_i \leftarrow \frac{(S_i - 1)\tilde{L}_i + L_{in}}{S_i}$
38             $\tilde{\omega}_i \leftarrow \frac{(S_i - 1)\tilde{\omega}_i + \omega_{in}}{S_i}$
39             $useful\_lobes{+}{+}$
40   **end**
41   **end**
42 **end**
43 **if** $useful\_lobes == 0$ **then**
44     $insertCachePoint(position, \omega_{in}, L_{in})$
45 **end**

Algorithm 1: Algorithm showing the usage of the Significance Cache at each surface interaction.



(a) Importance distribution of incoming radiance represented as oriented cosine lobes

(b) Lobes weighted by the BRDF and cosine term

(c) Resultant CDF used to pick a lobe

Figure 1: *The steps involved in CDF construction.*

to keep the algorithm unbiased. The resulting distribution, $\tilde{D}$ is a mixture distribution (lines 4 to 7):

$$\tilde{D} = \left(\sum_{i=1}^{N_{cache}} w_i D_i(x) \right) + w_{N_{cache}+1} D_{BRDF}(x). \quad (2)$$

Where w_i is the ith mixture weight, and $D_i(x)$ is the ith distribution (i.e. a cosine lobe) of the N_{cache} nearest cache points to the shading point x. These points are stored in an acceleration structure for fast access. We use a uniform grid, where each grid cell holds a list of cache points, however any other acceleration structure could be used. The structure is searched for the closest N_{cache} points, which are used for sampling. Points are considered acceptable to be used for sampling when they are oriented in the direction of the surface normal, and are within a certain distance from the sampling point (for example a small fraction of the diagonal of the scene bounding box). Distance weighting is not used as when N_{cache} is small, the lobes in the vicinity can be expected to approximately match the radiance distribution in the local area. The sum is over $N_{cache}+1$ lobes, with the final lobe $D_{BRDF}(x)$ representing a distribution defined over all directions of the hemisphere. This is in order to define a probability of sampling every direction of the hemisphere to keep the sampling unbiased, and to promote exploration within the scene. We can either use a cosine lobe with an exponent of one (a hemisphere), or we can use the BRDF determined from the material properties at x. Using the BRDF is advantageous, as there is still a probability of sampling high frequency areas of the BRDF; such as the peaks in a glossy BRDF, which may not be adequately covered by the cosine lobes.

3.3.2. Weighting

Each component of this distribution has to be assigned a weight w_i where $\sum_{i=0}^{N_{cache}+1} w_i = 1$ (see Figure 1(b) and lines 4 to 7 in Algorithm 1). The weights w_i are calculated to weight each lobe based on the product of the incoming radiance, the BRDF in the direction of the lobe, and the cosine term. The ideal weighting would be to project the triple product of the BRDF, incoming radiance (or importance) and cosine term onto each lobe in the following manner:

$$w_i = \int_{\Omega_{2\pi}} f_r(x, \omega_o, \omega)(n \cdot \omega) L(\omega) D_i(\omega) d\omega. \quad (3)$$

However, this ideal weighting is impossible to achieve in practice. Therefore, we use an approximation for the weighting of the lobes:

$$w_i \approx f_r(x, \omega_o, \omega_i)(n \cdot \omega) L_i, \quad (4)$$

where the subscript i refers to the ith lobe, ω_o is the incoming ray direction, and n is the surface normal. This weights each lobe by the BRDF, from the material at x evaluated in the lobe direction, and the cosine term. The weight for the final lobe ($w_{N_{cache}+1}$) can be assigned any value; however similar to [HP02] we assign a constant weight of:

$$w_{N_{cache}+1} = \sum_{i=1}^{N_{cache}} w_i \quad \text{(0.5 after normalization)}. \quad (5)$$

These lobe weights are then normalized by $\sum_{i=1}^{N_{cache}+1} w_i$ and are then used to construct a piecewise constant Probability Distribution Function (PDF). One exception to this rule is when the shading point has a specular BRDF; in this case BRDF sampling is used alone.

3.3.3. Sampling

The sampling procedure has to first pick a lobe to be sampled (see Figure 1(c) and lines 8 to 14 in Algorithm 1). A sample is drawn from the associated CDF which defines the lobe to be sampled. If the lobe selected is $N_{cache}+1$, then the BRDF is sampled with a probability of $p=0.5$ as $w_{N_{cache}+1}=0.5$. Otherwise, one of the cosine lobes is picked, and a sample is drawn. As the lobes are off normal, samples are drawn in the following way:

1. Draw a sample from the lobe according to $(\theta,\phi) = (\arccos(\sqrt[\lceil e\rceil+1]{\xi_2}), 2\pi\xi_3)$.
2. If the sample lies beneath the tangent plane at the shading point (x), go to step 1.
3. Normalize the PDF.

The PDF has to be normalized as the sampling area is no longer over the hemisphere, but over a spherical diagon (the portion of the hemisphere where the PDF >0), see [Dut03] for a detailed description of this process. Normalizing by V, the unnormalized PDF over the spherical diagon, ensures that each PDF integrates to 1. Therefore, the probability of sampling the direction ω_{in} is:

$$p_i = \sum_{i=1}^{N_{cache}} w_i \frac{(\omega_i \cdot \omega_{in})^{\lceil e_i \rceil}}{V_i} + w_{N_{cache}+1} p_{BRDF}(\omega_{in}), \qquad (6)$$

where $p_{BRDF}(\omega_{in})$ is the probability of sampling ω_{in} from this BRDF. As each component of the PDF integrates to 1, the overall PDF described by the mixture distribution also integrates to 1, and thus the single sample model from multiple importance sampling [VG95] is used to weight the samples.

Figure 1 illustrates the sampling pipeline. The image on the left (1(a)) shows the importance distribution stored as cosine lobes. The middle image (1(b)) shows the weighted lobes (the length of the lines represent the weight each lobe would be assigned by multiplying the BRDF, incoming radiance and cosine term), and the right image (1(c)) shows the CDF created for picking a lobe to sample.

3.4. Updating

In this section we discuss the process of updating the cosine lobes as outlined in lines 22 to 44 in Algorithm 1. Once the incoming radiance L_{in} is returned, cache points are updated based on information returned from shooting the ray. The same set of points that are found in the initial sampling pass are updated, in order to save a cache lookup. Three components of each cosine lobe have to be updated, that is the lobe direction $\tilde{\omega}$, the lobe exponent e and the radiance contained within the lobe $\tilde{L}$. The other parameters are also updated, but their function is to aide updating of the three components.

First, the number of samples that contribute to the ith lobe is updated by incrementing S_i. Next, the relevance of the sampled direction ω_{in} to the lobe is calculated:

$$s = \max((\omega_i \cdot \omega_{in})^{e_i}, 0). \qquad (7)$$

This value expresses how closely the sampled direction matches the direction of the cosine lobe, with samples which are close to the lobe direction being assigned a value around 1, and samples which are less likely to have come from the lobe having lower values. This is used in two places during the updating process. The first is calculating how the exponent of the lobe should adapt in Section 3.4.1, and the second is updating the lobe direction and stored radiance in Section 3.4.2.

3.4.1. Updating the exponent

The exponent is related to the ratio of all the samples at the cache point Sa to the number of relevant samples Sr. The intuition behind this is if useful samples are coming from directions within a small solid angle, fewer samples would normally be generated in those directions (if uniform sampling was used for example). This therefore would require a cosine lobe to have a higher exponent in order to place the majority of samples in the relevant solid angle. Likewise, if useful samples come from larger solid angle, a smaller exponent would be required. This leaves two issues to be resolved; firstly is how to determine if a sample is relevant to the lobe, and secondly how to weight the samples. The first is solved by the relevance function $T(\tilde{L}, L_{in})$ in line 26 of Algorithm 1. This function assesses how relevant the sample is to the lobe, based on the radiance returned from sampling, and the stored radiance in the lobe. There are several possibilities for how to create this function, for instance to use thresholding and only accept samples which are close to the predicted value, or to maintain an estimate of the variance of the samples contributing to the lobe, and accept samples within a predefined value such as one standard deviation. We use the former as it is simple to implement, requires no additional memory, and performed well in the results. Therefore, $T(,)$ is defined for the ith cache point as:

$$T(\tilde{L}_i, L_{in}) = \begin{cases} 1 & \text{if } \frac{abs(\tilde{L}_i - L_{in})}{\tilde{L}_i} < 0.5 \\ 0 & \text{otherwise} \end{cases} \qquad (8)$$

The number of relevant samples is updated with $Sr_i = Sr_i + T(\tilde{L}_i, L_{in})(1-s)$, and the total number of samples is also updated in a similar way $Sa_i = Sa_i + (1-s)$. Both are updated with $(1-s)$ as the samples close to the lobe direction are likely to have come from the lobe, and should have less effect on both accumulated quantities. The exponent is then calculated from the ratio of the accumulators:

$$e_i = max(\frac{Sa_i}{Sr_i} - 1, 1). \qquad (9)$$

This ensures that as more useful samples contribute to a given lobe the exponent will decrease, and otherwise the exponent will increase to reflect the fact the useful samples are clustered over a small solid angle. This leads to a

Table 1: *Predicted vs Experimental exponents.*

Solid Angle	Experimental	Predicted
0.31	13	12
0.62	6	7
0.94	4	5
1.25	3	4
1.57	2	3

self correcting system in which the lobe fluctuates between slightly overestimating and underestimating the correct exponent. However, the magnitude of these fluctuations decreases as rendering progresses. Therefore, if the exponent is too low at a point, more samples are generated for which $T(\tilde{L}_i, L_{in})$ is 0, which increases the exponent. Likewise, if the exponent is too high at a point, more samples are generated for which $T(\tilde{L}_i, L_{in})$ is 1, which decreases the exponent. The subtraction of 1 is used to align the exponent with experimentally observed values. An experiment was performed to determine how closely the exponents predicted by this model match experimentally determined values. A simple scene was created in which light was defined to emit from a variety of solid angles over the hemisphere. Exponents were then generated so that 50% of the generated samples would fall within this solid angle. Table 3.4.1 shows this comparison, and the model predicts an exponent which is slightly too large for most of the solid angles, hence the difference of 1 being subtracted from the ratio above.

3.4.2. Updating the radiance and direction

The next stage is to update the stored radiance and the direction in which the lobe is pointing. A sample is considered to be relevant to update these if it falls within the upper majority of the cosine lobe, that is $s > 0.5$, and if it is greater than the stored average radiance at the lobe. The estimate of the average radiance of all the samples $\tilde{I}_i$ stored at the ith lobe is updated as:

$$\tilde{I}_i = \frac{(S_i - 1)\tilde{I}_i + L_{in}}{S_i}, \tag{10}$$

and it is used to check whether the incoming radiance is greater than the average radiance value $L_{in} > \tilde{I}$. If it is, and $s > 0.5$, the stored radiance is updated in a similar way:

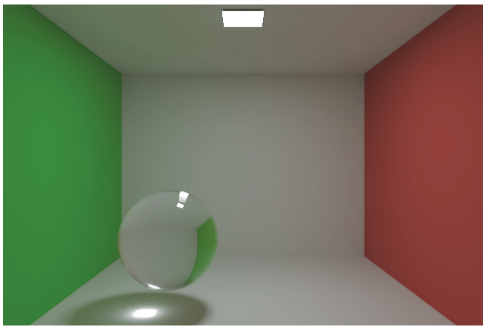
$$\tilde{L}_i = \frac{(S_i - 1)\tilde{L}_i + L_{in}}{S_i} \tag{11}$$

The lobe direction is also updated:

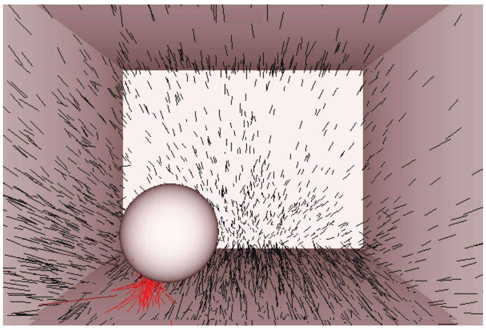
$$\tilde{\omega}_i = \frac{(S_i - 1)\tilde{\omega}_i + \omega_{in}}{S_i}. \tag{12}$$

This ensures that all relevant samples contribute to the updating process, but that later samples have less of an effect once a cache point is established. If for all of the lobes $L_{in} > \tilde{I}$ and $s < 0.5$, then the sample doesn't belong to any of the existing cache points, so a new one should be created ($insertCachePoint(position, \omega_{in}, L_{in})$ on line 42 in Algorithm 1).

Figure 2 gives an overview of the process. The image on the left (Figure 2(a)) shows the path taken from the eye to the light. Initially (Figure 2(b)), we have no other information about the distribution of light in the scene, so the lobes generated are oriented hemispheres. However, after several samples, we have a better idea of the distributions of the light in the scene, so the lobes adapt (Figure 2(c)).



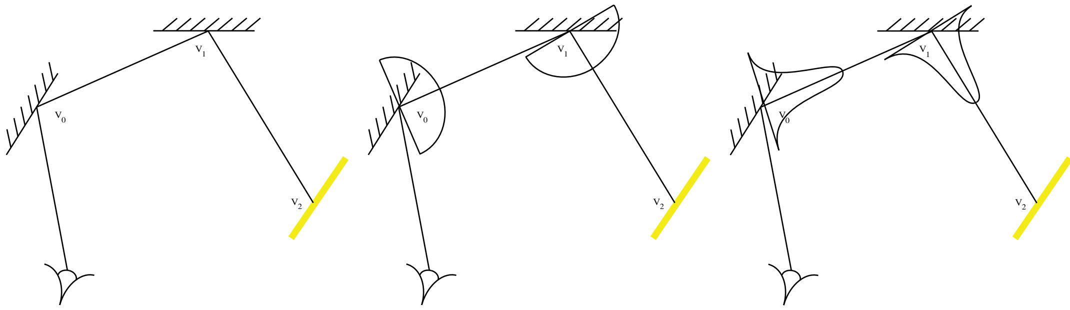
(a) Reference Cornell Box Image



(b) Visualization of the cache points and directions generated for this particular scene, with red cache points symbolizing greater values of $\tilde{L}$. The length of the lines in the image represents the exponent of the lobe.

Figure 3: *The distribution of cache points in the Cornell Box scene.*

Some less useful incoming samples are discarded based on the method outlined above, and leads to a slight under or overestimation of the radiance. However, this doesn't add bias to the algorithm, as although low contribution samples do not contribute towards updating the PDF, they are still used when rendering the image. This means that fewer low contribution samples will be generated, but due to the importance sampling during rendering, will be weighted higher.



(a) An initial path generated with no information about the indirect lighting in the scene

(b) Initial lobes

(c) Adapted lobes

Figure 2: *Lobe adaption as rendering progresses.*

The updating methods for the SC have the advantage that they make the SC naturally adapt to changes in incoming radiance, and that cache points are only placed in relevant areas. For instance, in a path tracer, the cache will be far more dense where there are paths to the light source, and sparse or empty in irrelevant areas. Figure 3 shows an example of how the cache adapts around the caustic formed on the floor of the Cornell Box. If multiple important directions are present at a point, the combination of the sampling, and the cache updating process, leads to the important directions being located and exploited.

3.5. Participating Media

The SC, with some modifications, can also be applied to participating media. Several approaches have been taken when rendering participating media for instance photon mapping [JC98], or bi-directional path tracing [LW96]. Approaches can be categorized into two types; single scattering and multiple scattering. Single scattering allows direct lighting calculations within the medium; whereas multiple scattering performs a random walk with several interactions through the medium. We focus on multiple scattering, as more challenging light paths can be computed (such as volumetric caustics).

The SC can be used to accelerate the rendering of participating media with only minor changes from the previous sections. When updating the cache, we can use the same methods described in Algorithm 1. However, when sampling, a couple of modifications must be made. The BRDF is not applicable in participating media, so all references to the BRDF in Section 3.3 should use the phase function. Also, instead of sampling the lobe above the tangent plane, which is not applicable in a participating medium, the entire lobe is used.

4. Results

In this section we present the results of the SC algorithm. We primarily compare SC against path tracing (PT). In all the results, the SC is used to sample a direction at every shading point in a path. We ran our results on an Intel Quad Core Q6850 running at 3GHz and 2GB RAM. All rendering algorithms were implemented in the same code base; we only altered the sampling code from a BRDF sampler to use the SC. The images were rendered at a resolution of 600×400. All reference images were generated using path tracing. We also compare our algorithm against importance sampling using the photon map [Jen95]. Our implementation shoots 100000 photons in order to create the initial photon distribution, and search for the nearest 50 photons when sampling. The number of photons shot is larger than the one recommended in Jensen et al. [Jen95] due to the complexities of the scenes, yet gave reasonable times for the kd-tree lookups. Images of the scenes used for the results are shown in Figures 4, 5, 6 and 7. Each figure includes the reference image and a series of images showing the difference at different SPPs for SC, PT and with importance sampling using the photon map (PM) Jensen et al. [Jen95]. All images are tone mapped for display. Figure 8 (left) shows RMSE graphs corresponding to the images shown at different numbers of samples per pixel (SPP). We also show efficiency graphs, see Figure 8 (right), which take into account the time taken to compute each image. The efficiency is calculated as $\frac{1}{time \times RMSE^2}$. These results are captured at the same number of pixels and although SC is slightly slower (15% on average) than PT for the same number of SPP, due to the overhead of cache lookups and the time taken to sample the lobes, the efficiency results reflect the sampling improvements. Note, that the SPP along the x-axis does not grow linearly.

Figure 4 shows the Sibenik scene. This is lit from an area light pointing downwards from behind the camera. The zoomed in images show an indirectly lit area of the ceiling at varying numbers of samples per pixel. RMSE demonstrates an improvement of SC over PT of $30-40\%$, with greater improvements occurring at larger numbers of samples per pixel due to the cache adapting to the indirect lighting in the scene. Compared to PM, our method offers a greater improvement as our data structure adapts as rendering progresses, whereas the initial photon distribution is inadequate in this scene leading to less efficient importance sampling using PM.

Figure 5 shows an office scene which is almost totally indirectly lit through a doorway. This is a challenging scene, as in many places the minimum path length to a light source is three. Our algorithm quickly adapts to the lighting in the scene, enabling more of these paths to be found. RMSE demonstrates an improvement of between $40-50\%$ in this scene due to the difficult lighting configuration. Again, as the cache adapts to the lighting, there is a gradual improvement in the RMSE value over path tracing. Our method shows a large improvement over PM. This is due to the photons being shot from the light source, so there is an inadequate number around the surfaces visible from the camera, whereas SC adapts to the positioning of the camera and lights in the scene.

Figure 6 shows a caustic being cast onto a highly glossy floor. This was modeled as a Ward BRDF [War92] with a specular weight of 0.95. Therefore, using the standard technique of only sampling the BRDF, as is common with path tracing, most of the samples will point in the direction of the specular component, and miss the caustic paths. As soon as our technique finds these paths, it is able to draw samples from the distribution of the incoming radiance, and forms a far more converged caustic. There is a $60-75\%$ improvement in RMSE for SC over PT in this scene. There is greater efficiency at fewer samples per pixel in this scene, due to the early benefit of using the cache to sample the caustic. In this scene, we modified PM to sample the BRDF 50% of the time, due to the extremely glossy surface (similar to Hey et al. [HP02]). Initially PM outperforms SC in this scene, due to the difficult lighting scenario, and the low probability of finding the caustic paths. However, the SC adapts to the lighting, and outperforms PM at under 500 SPP. The SC can also use a photon shooting pass to initialize the cache, which would lead to a greater performance at lower SPP.

In Figure 7, we demonstrate the application of our caching structure to multiple scattering participating media. We placed a glass sphere in the center of a Cornell Box filled with homogenous participating media. In this scene, the RMSE results for SC are improved by 30% over PT, as our algorithm finds the caustic region, and adapts the sampling accordingly, enabling a more converged caustic at the same number of samples. We do not compare with PM in this scene, as the method presented in Jensen et al. [Jen95] doesn't support participating media.

In Table 2, we report results for the memory usage and numbers of cache points in the above scenes. In general, the memory usage is modest, however the Participating Media scene uses more memory, due to the larger number of cache points. This is due to cache points being created in the medium and not only on surfaces. Likewise, as the glossy caustic scene contains simpler geometry, the number of cache points is smaller than the more complicated scenes, except in the caustic area where it adapts to the higher frequency component of the illumination. The memory usage and the number of cache points were recorded once the cache had stabilized in terms of cache points being added.

Table 2: *This table shows the memory usage in MB for the scenes, and the number of cache points used.*

Scene	Memory (MB)	Cache Points
Sibenik	1.18	28,056
Office	1.23	29,303
Participating Media	30.72	732,104
Glossy Caustic	0.06	1,503

5. Discussion

Our algorithm does have certain drawbacks. The use of the SC adds time to the rendering (approximately 15%), however, as results show, the overall performance benefits from improved efficiency. Using methods such as distributed ray tracing [CPC84], which can amortize the cost of cache lookups over many samples, can also alleviate the problem. Also, sampling efficiency could be increased if the BRDF were to be represented by cosine lobes, then samples could be drawn from an approximate cosine lobe product (see [MPA*10]). This is not currently implemented as this requires the BRDFs to be fitted to a cosine lobe representation as a pre-process, which is not always feasible for a general rendering system.

Path tracing is not the only rendering algorithm which can be used in conjunction with the SC. In the following sections we elaborate on other rendering methods which can be used with the SC.

5.1. Metropolis Light Transport

One comparison that can be made with SC is to Metropolis Light Transport (MLT) [VG97, KSKAC02]. MLT exploits the correlation between samples, and is very effective at sampling harder to find paths. Our algorithm can be combined with MLT in order to improve performance in this type of scene. This uses the cache points to suggest mutations which have a higher chance of being accepted, and therefore increases convergence. In order to be able to calculate transition probabilities, each path stores a snapshot of the cache points which contribute to each vertex in the path. These are then discarded when the path is no longer needed. Figure 9 demonstrates results for MLT and SC accelerated MLT with the same SPP for the same area shown in Figure 5.

5.2. Particle tracing

The SC is also applicable to particle tracing, such as Progressive Photon Mapping [HOJ08]. These walks start at the light source, are transported around the scene based on the BRDF, and are deterministically connected to the eye. Importance, the adjoint quantity to light which is emitted by the eye, can also be cached and used to guide the light walks. The cache does not need to be altered in any way in order to be used with this type of algorithm.

The SC can be extended to bi-directional path tracing (invented independently by Veach et al. [VG94] and LaFortune et al. [LW93]). Bi-directional path tracing shoots rays from both the light sources and the camera, and performs shadow ray checks to connect the vertices in each path. In this case, two copies of the cache can be stored, one for the camera paths, and the other for the light paths. Cache points in the relevant cache can then be updated based on the results of the shadow rays. For instance, if a vertex on a camera path (v_c) is connected to a vertex on a light path (v_l), then the cache that stores radiance is updated based on the incoming radiance at v_c, and the cache that stores importance is updated based on the incoming importance at v_l.

6. Conclusions and Future Work

In this paper, we have presented a novel algorithm for adaptively storing incoming radiance in order to more effectively sample harder to find paths. We have shown how this can be applied to a variety of lighting scenarios and participating media. The incoming radiance is stored as cosine lobes, and the structure is updated and refined as the simulation progresses. We showed how this can result in improved performance in hard cases compared to normal sampling methods.

As future work, we would like to focus on extending the cache to higher dimensions (i.e. a temporal component for dynamic scenes). This would allow subsequent frames to use knowledge of the incoming radiance gained in the current frame. Future work will look into extending this method to GPU, whereby the cache could be stored in a read/write buffer, and accessed and updated during rendering.

References

[BAJ08] BUDGE B., ANDERSON J., JOY K.: Caustic forecasting: Unbiased estimation of caustic lighting for global illumination. In *Computer Graphics Forum* (2008), vol. 27, Blackwell Science Ltd, Osney Mead, Oxford, OX 2 0 EL, UK,, pp. 1963–1970. 2

[BGH05] BURKE D., GHOSH A., HEIDRICH W.: Bidirectional importance sampling for direct illumination. In *Proceedings of Eurographics Symposium on Rendering 2005* (June 2005), Eurographics. 2

[CAM08] CLARBERG P., AKENINE-MOLLER T.: Exploiting visibility correlation in direct illumination. In *Computer Graphics Forum* (2008), vol. 27, Wiley Online Library, pp. 1125–1136. 3

[CPC84] COOK R., PORTER T., CARPENTER L.: Distributed ray tracing. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques* (1984), ACM, p. 145. 9

[DCE08] DAVID CLINE D. A., EGBERT P.: Table-driven Adaptive Importance Sampling. *Computer Graphics Forum 27*, 4 (2008). 2

[Dut03] DUTRÉ P.: Global illumination compendium. *Computer Graphics, Department of Computer Science Katholieke Universiteit Leuven* (2003). 6

[DW94] DUTRÉ P., WILLEMS Y. D.: Importance-driven Monte Carlo light tracing. In *Proceedings of the Fifth Eurographics Workshop on Rendering* (June 1994), Eurographics, pp. 185–194. 3

[DW95] DUTRÉ P., WILLEMS Y. D.: Potential-driven Monte Carlo particle tracing for diffuse environments with adaptive probability density functions. In *Rendering Techniques '95 (Proceedings of the Sixth Eurographics Workshop on Rendering)* (1995), Springer-Verlag, pp. 306–315. 3

[GXS09] GENG Z., XU Q., SUN J.: Object Space Adaptive Sampling for Global Illumination. In *Computer Science and Information Engineering, 2009 WRI World Congress on* (2009), vol. 2, IEEE, pp. 653–657. 2

[HJW*08] HACHISUKA T., JAROSZ W., WEISTROFFER R., DALE K., HUMPHREYS G., ZWICKER M., JENSEN H.: Multidimensional adaptive sampling and reconstruction for ray tracing. In *ACM SIGGRAPH 2008 papers* (2008), ACM, pp. 1–10. 3

[HOJ08] HACHISUKA T., OGAKI S., JENSEN H.: Progressive photon mapping. In *ACM SIGGRAPH Asia 2008 papers* (2008), ACM, pp. 1–8. 2, 10

[HP02] HEY H., PURGATHOFER W.: Importance sampling with hemispherical particle footprints. In *In Proc. of the 18th spring conference on Computer Graphics* (2002), ACM Press, pp. 107–114. 2, 5, 9

[JC98] JENSEN H. W., CHRISTENSEN P. H.: Efficient simulation of light transport in scenes with participating media using photon maps. *Computer Graphics 32* (July 1998), 311–320. Proc. of SIGGRAPH'98. 8

[Jen95] JENSEN H. W.: Importance Driven Path Tracing Using the Photon Map. In *Rendering Techniques '95 (Proceedings of the Sixth Eurographics Workshop on Rendering)* (New York, NY, 1995), Hanrahan P. M., Purgathofer W., (Eds.), Springer-Verlag, pp. 326–335. 2, 8, 9

[Jen96] JENSEN H. W.: Global illumination using photon maps. In *Rendering Techniques '96*, Pueyo X., Schröder P., (Eds.). Springer-Verlag, 1996, pp. 21–30. 2

[Kaj86] KAJIYA J. T.: The rendering equation. In *Computer Graphics* (1986), pp. 143–150. 1, 2

[KGPB05] KRIVÁNEK J., GAUTRON P., PATTANAIK S., BOUATOUCH K.: Radiance caching for efficient global illumination computation. *IEEE Transactions on Visualization and Computer Graphics 11*, 5 (2005), 550–561. 1, 3

[KSKAC02] KELEMEN C., SZIRMAY-KALOS L., ANTAL G., CSONKA F.: A simple and robust mutation strategy for the metropolis light transport algorithm. In *Computer Graphics Forum* (2002), vol. 21, pp. 531–540. 9

[KW00] KELLER A., WALD I.: Efficient importance sampling techniques for the photon map. In *Proceedings of the Fifth Fall Workshop on Vision, Modeling, and Visualization* (Nov. 2000), IEEE, pp. 271–279. 2

[Lew94] LEWIS R.: Making shaders more physically plausible. In

Computer Graphics Forum (1994), vol. 13, John Wiley & Sons, pp. 109–120. 3

[LFTG97] LAFORTUNE E. P. F., FOO S.-C., TORRANCE K. E., GREENBERG D. P.: Non-linear approximation of reflectance functions. *Computer Graphics 31*, Annual Conference Series (1997), 117–126. 3

[LW93] LAFORTUNE E., WILLEMS Y.: Bi-directional path tracing. In *Proceedings of CompuGraphics '93* (Alvor, Portugal, Dec. 1993), pp. 145–153. 10

[LW95] LAFORTUNE E. P., WILLEMS Y. D.: A 5D Tree to Reduce the Variance of Monte Carlo Ray Tracing. In *Rendering Techniques '95 (Proceedings of the Sixth Eurographics Workshop on Rendering)* (New York, NY, 1995), Hanrahan P. M., Purgathofer W., (Eds.), Springer-Verlag, pp. 11–20. 2, 3

[LW96] LAFORTUNE E. P., WILLEMS Y. D.: Rendering participating media with bidirectional path tracing. In *Rendering Techniques '96 (Proceedings of the Seventh Eurographics Workshop on Rendering)* (1996), Springer-Verlag, pp. 91–100. 8

[MPA*10] MEUNIER S., PERROT R., AVENEAU L., MENEVEAUX D., GHAZANFARPOUR D.: Cosine lobes for interactive direct lighting in dynamic scenes. *Computers & Graphics* (2010). 3, 9

[PBSP08] PEGORARO V., BROWNLEE C., SHIRLEY P. S., PARKER S. G.: Towards Interactive Global Illumination Effects via Sequential Monte Carlo Adaptation. In *Proceedings of the 3rd IEEE Symposium on Interactive Ray Tracing* (2008), pp. 107–114. 2

[Pho75] PHONG B.: Illumination for computer generated pictures. *Communications of the ACM 18*, 6 (1975), 311–317. 3

[PP98] PETER I., PIETREK G.: Importance driven construction of photon maps. In *Rendering Techniques '98 (Proceedings of the Ninth Eurographics Workshop on Rendering)* (1998), Springer-Verlag, pp. 269–280. 2

[PWP08] PEGORARO V., WALD I., PARKER S. G.: Sequential monte carlo adaptation in low-anisotropy participating media. *Computer Graphics Forum 27*, 4 (June 2008), 1097–1104. 2

[SKCP99] SZIRMAY-KALOS L., CSÉBFALVI B., PURGATHOFER W.: Importance driven quasi-random walk solution of the rendering equation. *Computers & Graphics 23*, 2 (1999), 203–211. 2

[SKS09] SZIRMAY-KALOS L., SZÉCSI L.: Deterministic Importance Sampling with Error Diffusion. In *Computer Graphics Forum* (2009), vol. 28, Wiley Online Library, pp. 1055–1064. 3

[SWZ96] SHIRLEY P., WANG C., ZIMMERMAN K.: Monte Carlo techniques for direct lighting calculations. *ACM Transactions on Graphics 15*, 1 (1996), 1–36. 3

[TCE05] TALBOT J., CLINE D., EGBERT P. K.: Importance resampling for global illumination. In *Rendering Techniques* (2005), pp. 139–146. 2

[VG94] VEACH E., GUIBAS L.: Bidirectional estimators for light transport. In *Proceedings of the Fifth Eurographics Workshop on Rendering* (June 1994), Eurographics, pp. 147–162. 10

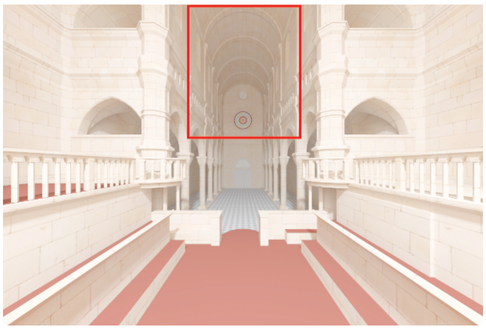
[VG95] VEACH E., GUIBAS L. J.: Optimally combining sampling techniques for monte carlo rendering. *SIGGRAPH (ACM Transactions on Graphics)* (1995), 419–428. 2, 6

[VG97] VEACH E., GUIBAS L. J.: Metropolis light transport. *SIGGRAPH (ACM Transactions on Graphics)* (1997), 65–76. 2, 9

[WÅ09] WANG R., ÅKERLUND O.: Bidirectional importance sampling for unstructured direct illumination. *Computer Graphics Forum 28*, 2 (Apr. 2009), 269–278. 2

[War92] WARD G. J.: Measuring and modeling anisotropic reflection. In *Proceedings of SIGGRAPH '92* (1992), ACM Press, pp. 265–272. 9

[WRC88] WARD G. J., RUBINSTEIN F. M., CLEAR R. D.: A ray tracing solution for diffuse interreflection. *SIGGRAPH (ACM Transactions on Graphics)* (1988), 85–92. 1



(a) Reference Image

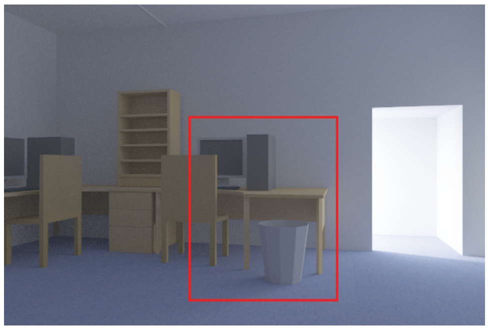
(b) SC 500SPP 1072s (c) PT 500SPP 928s (d) PM 500SPP 1172s

(e) SC 1000SPP 2094s (f) PT 1000SPP 1854s (g) PM 1000SPP 2360s

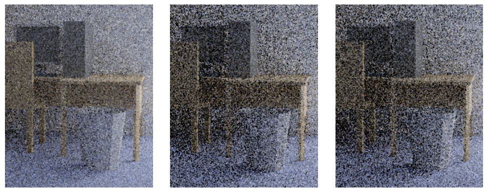
(h) SC 2000SPP 4274s (i) PT 2000SPP 3703s (j) PM 2000SPP 4719s

(k) SC 3000SPP 6283s (l) PT 3000SPP 5553s (m) PM 3000SPP 7110s

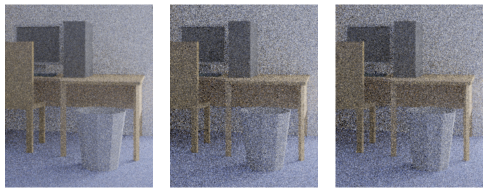
Figure 4: *Sibenik scene results. The images show a magnified region of the roof of the Sibenik Cathedral. SC gives a reduction ion variance compared to both PT and PM.*



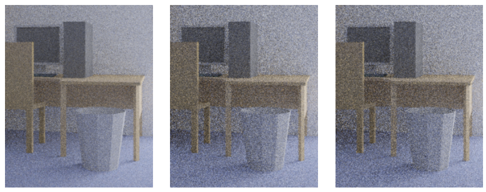
(a) Reference Image



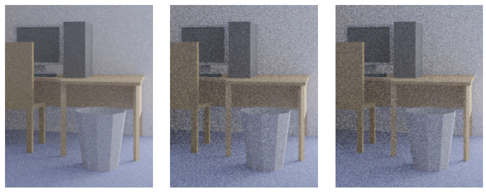
(b) SC 100SPP 89s (c) PT 100SPP 77s (d) PM 100SPP 99s



(e) SC 500SPP 439s (f) PT 500SPP 390s (g) PM 500SPP 463s

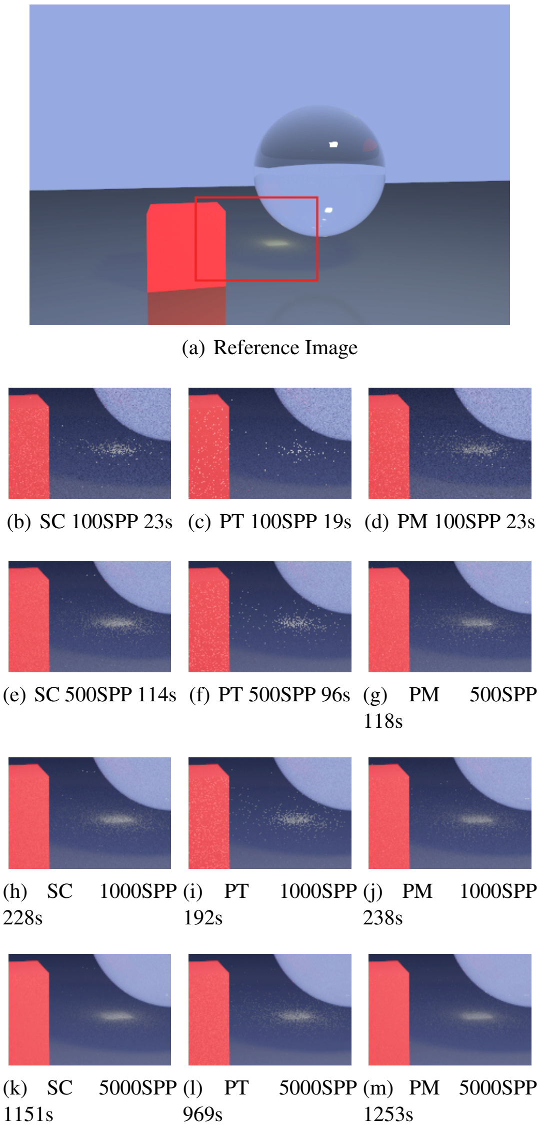


(h) SC 1000SPP 876s (i) PT 1000SPP 784s (j) PM 1000SPP 865s



(k) SC 2000SPP 1678s (l) PT 2000SPP 1492s (m) PM 2000SPP 1683s

Figure 5: *Office scene results. The images show an indirectly light region of the scene, and demonstrate that SC is more efficient than both PT and PM.*



(a) Reference Image

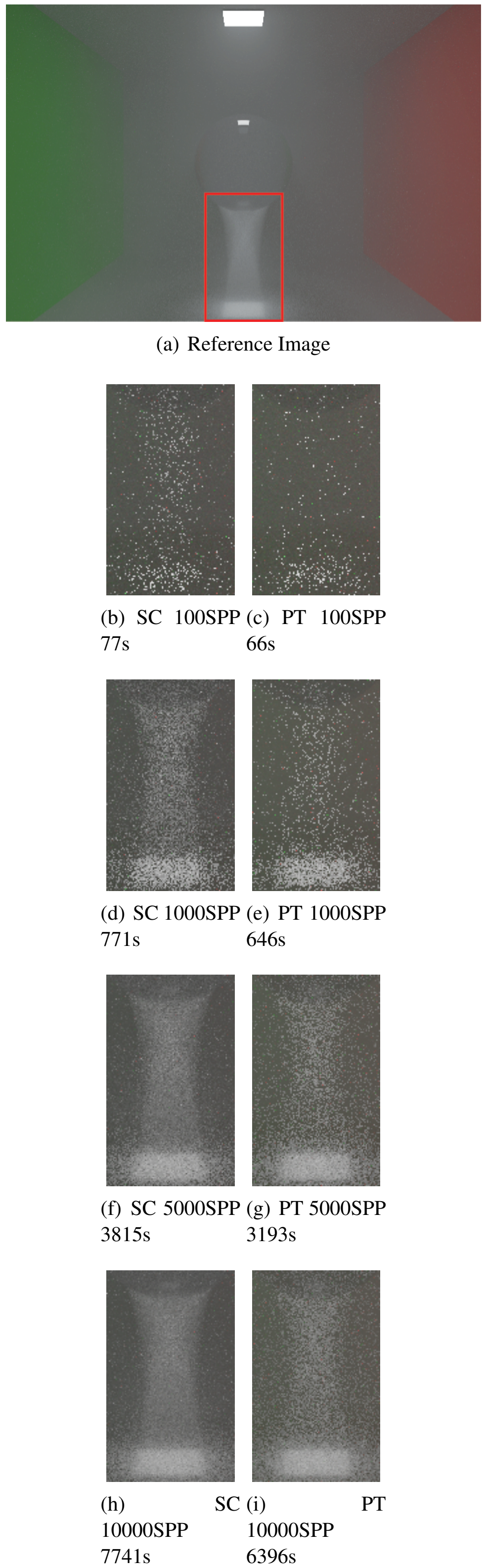
(b) SC 100SPP 23s (c) PT 100SPP 19s (d) PM 100SPP 23s

(e) SC 500SPP 114s (f) PT 500SPP 96s (g) PM 500SPP 118s

(h) SC 1000SPP 228s (i) PT 1000SPP 192s (j) PM 1000SPP 238s

(k) SC 5000SPP 1151s (l) PT 5000SPP 969s (m) PM 5000SPP 1253s

Figure 6: *Glossy Caustic scene. This scenario shows a caustic forming on a highly glossy surface from light focused through the glass sphere. SC outperforms PT, and once the cache has established PM.*



(a) Reference Image

(b) SC 100SPP 77s (c) PT 100SPP 66s

(d) SC 1000SPP 771s (e) PT 1000SPP 646s

(f) SC 5000SPP 3815s (g) PT 5000SPP 3193s

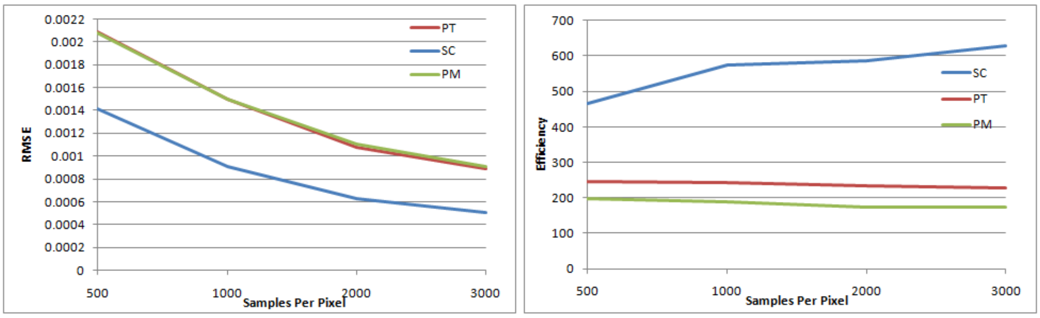
(h) SC 10000SPP 7741s (i) PT 10000SPP 6396s

Figure 7: *Multiple scattering Participating Media scene, in which a volumetric caustic forms in fewer samples with SC than PT.*

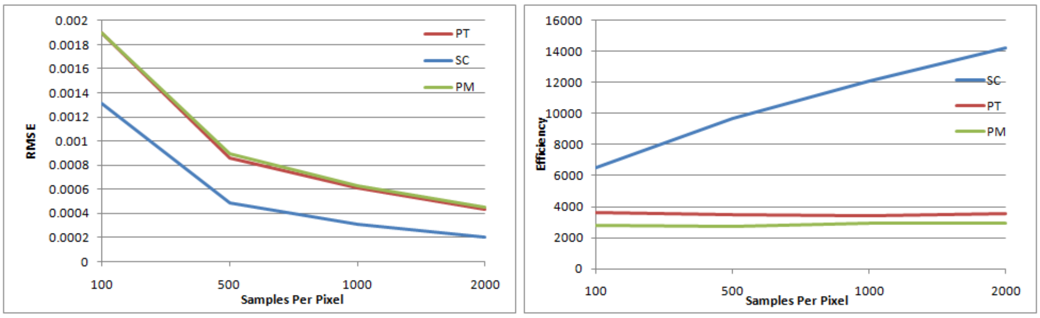
(a) SC accelerated MLT, RMSE = 0.0029, Efficiency = 90.22

(b) MLT, RMSE = 0.0037, Efficiency = 60.42

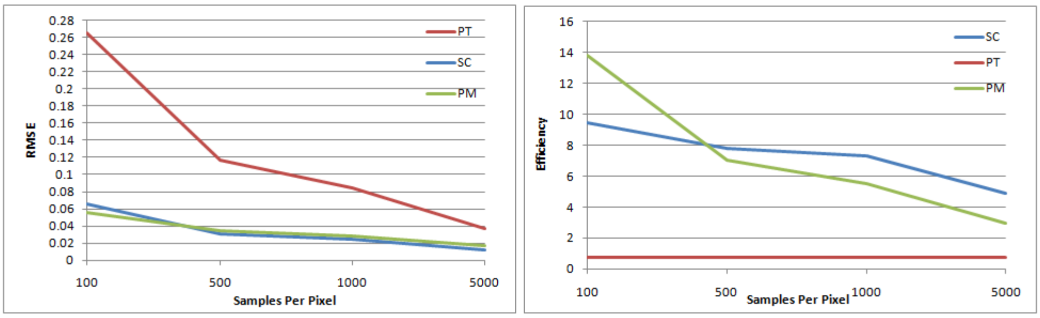
Figure 9: *Results for SC in conjunction with MLT for the office scene.*



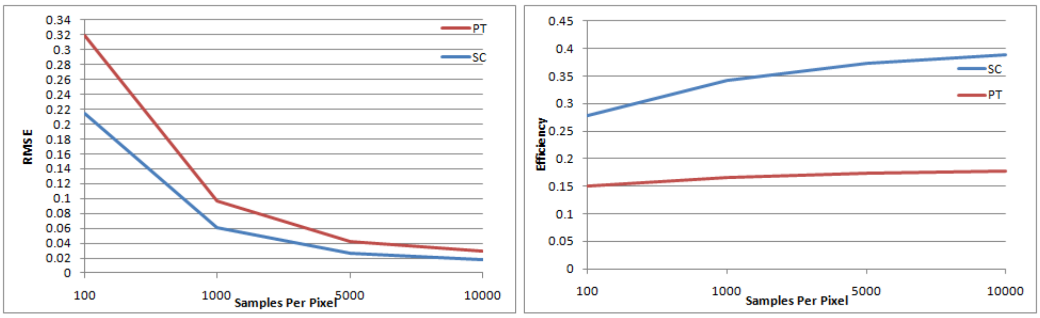
(a) Sibenik



(b) Office



(c) Glossy Caustic Scene.



(d) Participating Media

Figure 8: *RMSE (left) and Efficiency results (right). The efficiency results also take into account the time taken to render the image and are calculated as* $\frac{1}{time \times RMSE^2}$.