# Importance Driven Environment Map Sampling

Thomas Bashford-Rogers, Kurt Debattista and Alan Chalmers
WMG, University of Warwick, UK

**Abstract**—In this paper we present an efficient method for supporting Image Based Lighting (IBL) for bidirectional methods. This improves both sampling of the environment, and the detection and sampling of important regions of the scene, such as windows and doors. These parts of the scene often have a small area proportional to that of the entire scene, so paths which pass through them are generated with a low probability. The method proposed in this paper improves sampling efficiency, by taking into account view importance, and modifies the lighting distribution to use light transport information from the camera. This method automatically constructs a sampling distribution in locations which are relevant to the camera position, thereby improving sampling of light paths. This approach can be applied to several bidirectional rendering methods, and results are shown for Bidirectional Path Tracing, Metropolis Light Transport and Progressive Photon Mapping. When compared to other methods, efficiency results demonstrate speed ups of orders of magnitude.

**Index Terms**—I.3.7 Three-Dimensional Graphics and Realism, I.3.7.f Raytracing, I.6.8.g Monte Carlo

## 1 Introduction

Natural lighting is often a vital component for many computer graphics applications. Examples include architectural visualization, film production and simulations. Lighting in these scenarios frequently relies on a captured representation of the lighting at a point in reality [2]. This representation is used to relight a virtual scene; a rendering process normally referred to as Image Based Lighting (IBL). Most current IBL approaches aim to sample direct lighting from the environment for path tracing based methods, where all rendering is computed using backward ray tracing. However, in order to efficiently render many complex scenes, several commonly used approaches such as Bidirectional Path Tracing [3], [4], Photon Mapping [5], Instant Radiosity [6], Progressive Photon Mapping [7] and Metropolis Light Transport [8] start paths at light sources. Although all of these methods need to generate camera paths, the motivation for using such methods relies on the assumption that a significant portion of the lighting will come from efficiently sampling light paths. When generating paths from the environment for IBL, a 4D distribution, which is commonly factored into two 2D distributions, has to be sampled. The first distribution is used to pick the direction of the incoming light to the scene. The second is to pick a starting point of the path onto the scene. Once these distributions are sampled, the light paths interact with surfaces, and are connected to the camera.

In common situations (such as a camera inside a building lit by an environment map) it can be very difficult to generate a path between the light and the camera (see Figure 1). This difficulty in sampling leads to inefficient estimators, and slower rendering times. Therefore, improving sampling of light paths can directly benefit rendering efficiency.

In this paper we propose a method which we term Importance Driven Environment Map Sampling (IDES), to address these issues, and improve rendering performance of IBL for bidirectional methods. IDES provides two major contributions over other methods: improved selection of an outgoing direction from an environment map; and efficient sampling of starting points of light paths.

Normal sampling methods for the selection of the direction of the rays from the environment map use traditional techniques, such as uniform, stratified, or light source sampling [9], [10]. However, this is not an ideal sampling distribution; IDES improves this distribution by taking into account light transport information, which in turn leads to an increase in efficiency.

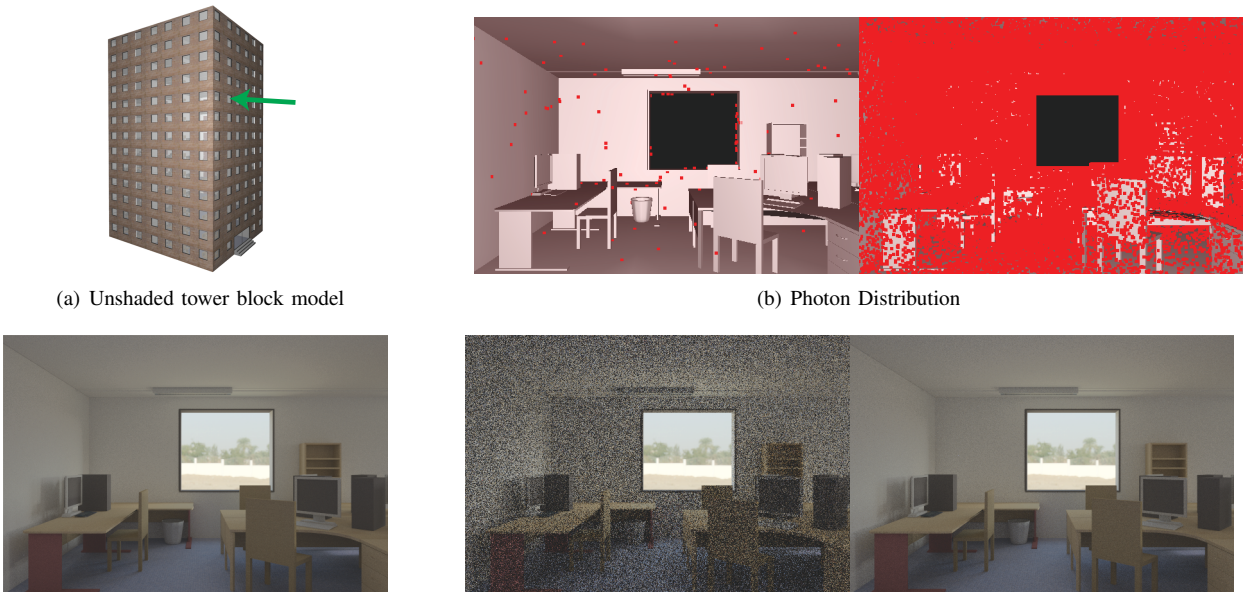
Generating paths from the environment map also requires the selection of the starting points of the paths. In most common scenes, the region of the scene which can be lit from the environment maps is a small proportion of the scene (for instance, a room in a skyscraper). If traditional techniques, such as sampling the bounding sphere of the scene [11], are used, then many samples will be wasted as they cannot contribute towards the image. IDES aims to improve this by generating a distribution from which starting points of the paths will be generated with a high probability of contributing to the image, thereby improving rendering efficiency.

To summarize, this paper makes the following contributions:

- Improved direction selection for IBL using a short path tracing pre-pass.
- Improved position sampling using a distribution generated during the pre-pass.

## 2 Background

A large body of research has been published aimed at improving importance sampling for environment maps. The vast majority of this work focuses on either generating samples on the environment map (that is a set of directional lights), or utilising screen space information to improve sampling. Both of these cases have drawbacks, in that information about importance (the adjoint of radiance) is generally not taken into consideration when generating samples on the environment map. Screen space methods



(a) Unshaded tower block model

(b) Photon Distribution

(c) Reference Image

(d) BPT after 30 Minutes

Fig. 1. When rendering within one of the rooms of the tower block model shown in 1(a), commonly few of the light paths/photons make it through, see 1(b) left. However, when using IDES many more make it through, 1(b) right. This results in images rendered using IDES, 1(d) right, converging much quicker than using alternative methods, 1(d) left, such as the plane sampling method by Dammertz and Hanika [1].

which use information about direct lighting at points seen through pixels [12], [13] are not robust for the general case. For example, screen space methods would show no improvement in a situation where light enters a room through a light shaft, and all pixels are lit purely by indirect illumination. These methods will not help when using bidirectional rendering algorithms where some paths start at the light sources.

Similarly to sampling area lights, environment sampling has to select from two distributions in order to generate an outgoing path. For area lights, there exist several successful approaches. For instance importance driven methods [14] and adaptive methods [15]. However, there has been less work on efficiently sampling the directional and position distributions in order to generate paths for the environment sampling case. The following sections cover existing work in the field of lighting from an environment map.

### 2.1 Direction Selection

Methods for sampling starting directions from an environment map can be broadly divided into two methods: those that generate a set of light sources, such as median cut [9], and those that use importance sampling of the luminance in the environment map [10]. Frequently, as stated above, the importance sampling methods also generate samples based on other attributes at the point being sampled such as the Bidirectional Reflectance Distribution Function (BRDF) ( [16]–[18]). However these methods are not directly applicable to bidirectional methods. None of these methods take into account the fact that the contribution of a light path is not only governed by the lighting distribution, but by both this, and the distribution of the paths connecting the lighting to the camera. Until now, to the best of our knowledge, no work has fully taken into account both distributions when sampling. Lawrence et al. [10], additionally use local environment map sampling, purely based on surface orientation, which does not take into account camera location, and view importance. This method also faces another limitation, that of long pre-computation times.

### 2.2 Position Selection

Sampling the starting position of a light path is currently carried out by a number of methods ( [1], [11], [19]). The method outlined by Pharr and Humphreys [11] creates a disk at the edge of the bounding sphere of the scene. This disk has the same radius as the bounding sphere, and therefore bounds the scene. A starting point for a ray is generated by uniformly sampling the disk.

Another approach by Dammertz and Hanika [1] projects the corner vertices of the scene's bounding box onto the direction of the light. When rendering, the 2D bounding box that encloses these vertices is calculated, and this box is sampled to generate directions for photons. Although this approach presents gains over the disk sampling method, it still does not exploit any view importance information. This therefore does not contain any information about important regions, which can make a substantial difference when sampling many scenes.

Schregle [19] use user interaction to manually place sampling geometry into the scene, to generate the starting points for light paths. Despite providing a two times improvement to sampling efficiency, the method is reliant on

user intervention; placement of geometry into the scene. Informed guesswork is also required for placement of sampling geometry In the case of animations, the user is forced to manually move the sampling geometry each frame, which is highly impractical.

An interactive approach based on portals was proposed by Yue et al [20]. This method uses portals to calculate the lighting response of a scene based on an exterior environment map, where the light passes through portals into the scene. This uses spherical harmonics for the indirect component, and step functions to represent direct lighting. This approach relies on manual assignment of portals, and like other pre-computed radiance transfer methods, it relies on pre-computation, and a non-exact solution.

### 2.3 Rendering of Interior Scenes

IDES is also related to the large body of research on rendering interior scenes. Many of these approaches are based on pre-computed visibility. We refer the reader to Cohen-Or et al. [21] for a detailed survey. Most of these methods pre-compute a cell and portal representation to store visible geometry for each cell, where visibility from one cell to another is handled through portals.

Efficient light transport in interior scenes has also been tackled by several authors. Fan et al. [22] uses Metropolis sampling, along with optional user input to suggest useful paths, to improve rendering quality with photon mapping. Similarly, Chen et al. [23] apply Metropolis sampling to Stochastic Progressive Photon Mapping [24]. Hachisuka and Jensen [25] use a simple form of replica exchange to efficiently sample light paths, including outdoor to indoor scenes. These approaches are compatible with our approach, and would indeed further benefit from IDES.

Other approaches make efficient use of generated paths. Methods have been proposed which combine the strengths of Bidirectional Path Tracing and Progressive Photon Mapping, which enable more efficient rendering [26], [27]. Georgiev et al. [28] also propose an importance sampling based approach for Instant Radiosity based approaches. As these approaches are concerned with utilising generated paths, their methods are complimentary to IDES, in that they would benefit from improved environment sampling.

## 3 Importance Driven Environment Map Sampling

IDES accelerates IBL for bidirectional methods by sampling from a 4D distribution of starting directions and points of a ray when generating paths from an environment light source. This 4D distribution is commonly factored into two 2D distributions. One selects a starting direction of a ray, the second selects a starting position based on the previously selected direction. The first distribution, with PDF (Probability Density Function) $p(\omega_l)$, picks a direction $\omega_l$ from which light will enter the scene. This is equivalent to generating a directional light source in this direction. The second distribution with PDF $p(x|\omega_l)$ is required to generate starting positions for the light paths. Note that this distribution selects a position $x$, depending on the selected direction $\omega_l$.

IDES consists of two phases; a short initialization phase where a pre-pass shoots importance from the camera, which is used to construct efficient distributions (for directions, $p(\omega_l)$, and positions, $p(x|\omega_l)$), and secondly the rendering phase. Improved direction selection, $p(\omega_l)$, is described in Section 4 and position selection, $p(x|\omega_l)$, in Section 5. Algorithm 1 broadly describes our algorithm.

```
INITIALIZATION:
begin
    Path tracing pre-pass
    Create path map for direction selection (Section 4)
    Create disk distribution for position selection
    (Section 5)
end

SAMPLING FROM LIGHTS:
begin
    Sample direction (Equation 4)
    Sample a starting position (Equation 7)
end
```

**Algorithm 1:** IDES Overview

## 4 Direction Selection

As stated in the previous section, two distributions have to be sampled in order to generate paths starting at environment maps. This section focuses on improving sampling of the directional component $p(\omega_l)$. We first present an improved density for selecting directions towards the scene, and then provide an approximation in Section 4.1, which is quickly computed from a first pass before rendering.

Light transport between lights (in our case, an environment map), and the camera, can be described succinctly by the path integral form of the rendering equation [29]:

$$L(\bar{x}) = \int_{\mathbb{P}} f(\bar{x}) d_{\mu}(\bar{x}) \tag{1}$$

where $\mathbb{P}$ is path space [29], and $\mu$ is the area measure on path space. $\bar{x} = x_0 ... x_k$ is therefore a path of length $k$. $f(\bar{x})$ is given by:

$$\begin{aligned} f(\bar{x}) =& L_e(x_0 \rightarrow x_1) W_c(\bar{x}) \\ W_c(\bar{x}) =& G(x_0 \leftrightarrow x_1) \\ & \left[ \prod_{i=1}^{k-1} f_r(x_{i-1} \rightarrow x_i \rightarrow x_{i+1}) G(x_i \leftrightarrow x_{i+1}) \right] \\ & W_e(x_{k-1} \rightarrow x_k) \end{aligned} \tag{2}$$

where $W_c$ is the importance from the camera reaching the light (often termed path throughput), $L_e$ is the emitted radiance from a light source, $f_r$ is the BRDF at a point and $W_e$ is the importance at the camera. $G$ here is the generalized geometry term [30] which takes into account the environment map. For environment maps, lighting coming from $x_0$ is equivalent to a directional light in the direction $\omega_l$.


a) b) c)

Fig. 2. The initial environment map is multiplied with the path map, to produce an improved sampling distribution for the environment.

An efficient sampling strategy for selecting directions from an environment map would utilize information about the radiance distribution in the environment map, and the contribution from each direction $\omega_l$ from the environment map to the image plane over all possible paths. The latter is the amount of importance transported from the camera to each direction on the environment map, expressed as:

$$C(\omega_l) = \int_{\mathbb{P}} W_c(x_0 \oplus \overline{x'}) d_\mu(\overline{x'}) \tag{3}$$

where $\overline{x'} = x_1 ... x_k$ and $\oplus$ is a concatenation operator which concatenates path segments. This calculates the importance transported from the camera to the environment in the direction specified by $\omega_l$. It can be used to drive an importance sampling approach, which generates samples proportional to radiance in the environment map, and the contribution to the image plane. However, this efficient importance sampling is impossible in practice, as constructing the required probability density function requires knowing the solution of the function it is approximating. Therefore, approximations to this distribution have to be generated.

Uniformly sampling a direction from the full sphere of directions is the simplest approach (where in this case $p(\omega_l) = 1/4\pi$). This has the limitations of poor variance reduction properties, due to the mismatch in the uniform sampling density to the peaks and troughs present in Equation 1. Importance sampling is an effective method of reducing variance, especially as the importance distribution becomes closer to being proportional to the function it approximates. This is used in more common approaches to draw directional samples proportional to the known and easily calculated part of the integrand in Equation 2, $L_e(\omega_l)$ (see Section 2.1). Typically this involves calculating the luminance $\mathfrak{L}$ for each direction (that is pixel) $\mathfrak{L}(L_e(\omega_l))$ in the environment map, and drawing samples proportional to this value. Many approaches for the typical environment sampling problem [10] use this approach. However, this still has the drawback of not sampling according to the entire distribution $L_e(\omega_l)C(\omega_l)$, but only part of it $p(\omega_l) \propto L_e(\omega_l)$, see Figure 2(a).

An improved sampling strategy would draw samples proportional to $L_e(\omega_l)C(\omega_l)$, as this better approximates Equation 2. However, a scalar density function is required, so samples can be generated proportionally to the luminance of the product $\mathfrak{L}(L_e(\omega_l)C(\omega_l))$:

$$p(\omega_l) = \frac{\mathfrak{L}(L_e(\omega_l)C(\omega_l))}{\int_\Theta \mathfrak{L}(L_e(\omega)C(\omega))d\omega} \tag{4}$$

where $\Theta$ represents the full sphere of directions. This resultant 2D distribution can then be directly sampled through existing methods in order to generate an outgoing direction from the environment. Any existing method can be used to generate directions using this improved distribution, such as through the use of marginal and conditional probabilities [10], or wavelets [18].

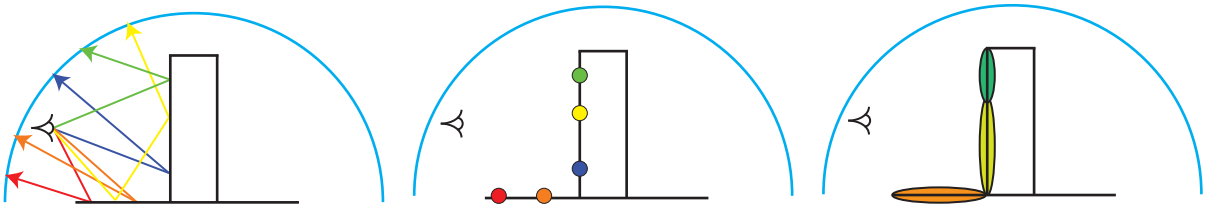
### 4.1 Path Map Construction

Thus far, we have assumed that $C(\omega_l)$ is known. However, it is not practical to compute this accurately, as that would require calculating all possible light paths between the camera and the environment. Therefore, we propose a pre-pass, where importance is shot from the camera to approximate $C(\omega_l)$ from Equation 3, as shown in Figure 2(b). Then an element wise multiplication with $L_e$, Figure 2(a), is performed, which leads to an approximation to the efficient directional distribution, Figure 2(c).

To trace importance from the camera to the environment, path tracing is used in the pre-pass. The importance is accumulated into a buffer we term the *path map*, Figure 2(b). Each time a path hits the environment, it splats importance onto the path map in the outgoing direction of the path. Splatting has several advantages:

- it approximates $C(\omega_l)$ with a finite number of samples
- it removes high frequency noise from the path map, which can result from the finite number of samples and glossy surfaces on the scene
- it is a more conservative approximation (important directions are less likely to be missed due to the finite number of samples used to generate the path map).

The splatting process initially transforms the outgoing direction of the path into the coordinate system used to store the environment/path map. For example spherical coordinates are used for a latitude-longitude representation. Once the coordinates on the path map have been computed, a splat of size $r_{splat}$ is drawn to the path map. We heuristically determined a good value for $r_{splat}$ to be $\sqrt{\frac{EnvSize}{\sqrt{PTSamples}}}$, where $EnvSize$ is the number of pixels in the environment map, and $PTSamples$ is the number of samples used in the path-tracing pass. This gives larger splat radii when the environment map is large, and few



(a) Paths generated in the pre-pass (b) The set $\chi$ consisting of the last path vertex on the scene (c) Sampling distribution consisting of disks generated from the set of points $\chi$

Fig. 3. Overview of starting position selection. 3(a) shows the paths which are traced during the pre-pass. The last vertex of these paths forms the set $\chi$, represented as the points in 3(b). Finally, a distribution consisting of a set of disks is generated from $\chi$, and is used when sampling, as illustrated in 3(c)

samples are used; and decreases the size as more samples are used. We used a Gaussian kernel for splatting, such that the splatted importance was weighted by Gaussian weights, however any kernel can be used.

As the path-tracing pass may not have found all the paths connecting the environment map to the camera, and to keep the sampling unbiased, a small quantity $\varepsilon_l$ is added to any pixels in the path map that initially have a value of zero. This is calculated from the existing values in the path map:

$$\varepsilon_l = \frac{\delta}{|NZ|} \sum_{\theta \in \Theta} \widehat{C}(\omega_\theta) \tag{5}$$

where $\delta$ is a small fraction (we use 0.001) and $NZ$ is the total number of non-zero elements in the path map. $\widehat{C}(\omega_\theta)$ is the accumulated approximation to $C(\omega_\theta)$ from the splatting process. As before, $\Theta$ is all of the directions (pixels) stored in an environment map, and $\theta$ is a direction (pixel).

This element-wise multiplication based approach has an additional advantage in scenes containing dynamic lighting in the environment map. If the scene and camera are static, the path map only has to be computed once. Therefore, the multiplication only has to be performed for each change in the environment map.

When rendering, samples are drawn from the resultant distribution shown in Figure 2(c) to generate starting directions for paths into the scene. This sampling process can be performed through the use of existing algorithms [10], [18].

## 5 Position Selection

Once a direction into the scene $\omega_l$ has been selected, a starting point of the ray into the scene must be chosen. As noted in the introduction, it is desirable for the starting positions of the paths into the scene to be proportional to their contribution to the image. This is a similar goal to the improved direction selection, and therefore a similar concept can be used. The direction selection pre-pass shoots importance from the camera into the scene; and when these paths randomly intersect the environment, their contribution is recorded in the path's exitant direction. These paths can also be used to generate improved starting points of paths from the environment to form the distribution $p(x|\omega_l)$. We note that all the final interactions of the paths with the scene in the pre-pass can be used to create an approximate point based density of the important areas of the scene, which are visible from the environment.

Specifically, we define a set $\chi$ of path vertices representing the last interaction of the pre-pass paths before these paths hit the environment. Next, these path vertices can be used to define a density for sampling starting points for paths shot into the scene. Instead of using the set $\chi$ directly, we create a distribution from $\chi$, which is more amenable for sampling. We build a set of oriented disks in the scene (from $\chi$), which then can be directly sampled when generating light paths. Disks were chosen because of a low pre-computation cost when building, and a low sampling cost during rendering. See Figure 3 for an illustration of this process.

This is similar to the concept of importons [31]. However, instead of using importons to build a hemispherical distribution, which obviously does not apply for environment maps, we build a distribution in object space (i.e. the scene).

### 5.1 Construction

In order to move from a point based representation of the important areas of the scene ($\chi$) to a useable sampling density, we apply a K-nearest neighbour search to form oriented disks in the scene. This can then be directly sampled when generating starting positions. Each point in the set of points $\chi$ from the pre-pass stores a position $\chi_x$, an orientation $\chi_\omega$, and importance $\chi_I$. Similarly, each disk stores a position $x_d$, an orientation $\omega_d$, importance $I_d$ and radius $R_d$. We aim to reduce the number of disks used when sampling as more disks increase sampling time. Therefore, the number of disks is chosen to be a small fraction $\sigma < 1$ of the points generated in the pre-pass. In order to generate the disks, we select a random point $\chi(s)$ from $\chi$, locate the $N = 1/\sigma$ nearest valid points $\chi^N \subset \chi$. A point $\chi(j) \in \chi^N$ is considered valid if $\chi_\omega(s) \cdot \chi^N_\omega(j) > \tau$, that is both points point in a similar direction given by a limit $\tau$ (we use a value of 0.5). A disk is then generated using $\chi(s)$ and $\chi^N$ at the average position, with average direction, with an importance value from the density estimation:

$$R_d = \max(|\chi_x(s) - \chi_x^N(j)|),\ j = 1..N$$
$$x_d = \frac{1}{M}\sum_{j=1}^{M}\chi_x^N(j)$$
$$\omega_d = \frac{1}{M}\sum_{j=1}^{M}\chi_\omega^N(j)$$
$$I_d = \frac{1}{\pi R_d^2}\sum_{j=1}^{M}\chi_I^N(j) \tag{6}$$

The set of points $\chi(s)$ and $\chi^N$ are then removed from the set $\chi$, and the process repeats. Once the disks have been constructed, an optional final step expands the radius of each disk to attempt to fill in any holes in the representation, resulting from the finite size of $\chi$ not providing complete coverage of the important regions. To fix these holes, we simply expand the radius of each disk, so that it overlaps with its $K$'th nearest neighbour. If the disk already encompasses its $K$'th nearest neighbour then no expansion takes place. The importance associated with each disk is re-computed from $\chi$ at this point to account for the larger radius.

### 5.2 Sampling Starting Positions

In order to be able to generate starting points for the rays, the information about disks computed above is used to improve sampling. Firstly, a starting direction $\omega_l$ is selected according to Section 4. Based on this direction, a disk is randomly selected; then a point $x_s$ is generated on that disk. The starting point of the path into the scene is then assigned as $x_l = x_s - (\omega_l * 2r_{bounds})$, to ensure the path from the environment starts outside the scene ($r_{bounds}$ is the radius of the scene's bounding sphere).

Formally, the set of disks to be sampled forms a mixture distribution:

$$p(x|\omega_l) = \sum_{i=1}^{N_{disks}} w(i,\omega_l)p_d(i|\omega_l) + w(bounds)p(bounds) \tag{7}$$

where, $w(i,\omega_l)$ is a weight assigned to the $i$'th disk and $p_d(i|\omega_l)$ is the PDF associated with sampling the $i$'th disk. We also include an additional distribution $p(bounds)$ for randomly sampling anywhere on the scene, in order to keep sampling unbiased. $w(bounds)$ is the probability associated with sampling the entire scene. This takes a low value as the disks provide a good sampling distribution. In order for this mixture distribution to be valid, $\sum_{i=1}^{N_{disks}} w(i,\omega_l) + w(bounds) = 1$.

When sampling, the following process has to be performed:

1) Generate weights $w(i,\omega_l)$ based on incoming direction $\omega_l$
2) Normalise weights and create a piecewise constant CDF (Cumulative Distribution Function)
3) Sample the CDF to pick a disk
4) Sample the selected disk to generate the point $x_s$
5) Calculate the starting point of the path
6) Calculate the PDF of sampling this disk according to Equation 7.

Firstly, unnormalized weights are calculated for each disk via the following function:

$$\widehat{w}(i|\omega_l) = max(-\omega_l \cdot \omega_d(i), 0)\, I_d(i). \tag{8}$$

The dot product in this equation calculates how the disk is oriented with respect to the incoming direction (the minus sign is due to the disk pointing towards the environment, and the direction $\omega_l$ pointing towards the scene). If this is less than 0, the disk is not visible from $\omega_l$ and should not be sampled. This term is then multiplied by the importance stored with the $i$'th disk, $I_d(i)$. This is to ensure more samples are generated on disks which are likely to transfer more radiance to the camera.

Once these weights have been computed for all the disks, they are normalised:

$$w(i|\omega_l) = (1 - w(bounds))\frac{\widehat{w}(i|\omega_l)}{\sum_{j=1}^{N_{disks}}\widehat{w}(j|\omega_l)} \tag{9}$$

where $(1 - w(bounds))$ takes into account the weight for sampling the entire scene. A piecewise constant CDF is then calculated for these weights, and sampled. A sample is uniformly generated on a unit disk, then transformed into world space resulting in the point $x_s$. If the final term of the mixture distribution is selected, we use the plane sampling method by [1].

Once a point is selected, it is moved to be outside the scene via $x_l = x_s - (\omega_l * 2r_{bounds})$. We then calculate the PDF of generating that point through any of the other possible disks and the bounds, weighted as in Equation 7. The PDF for generating a sample on the $i$'th valid disk disk (where $w(i|\omega_l) > 0$) is:

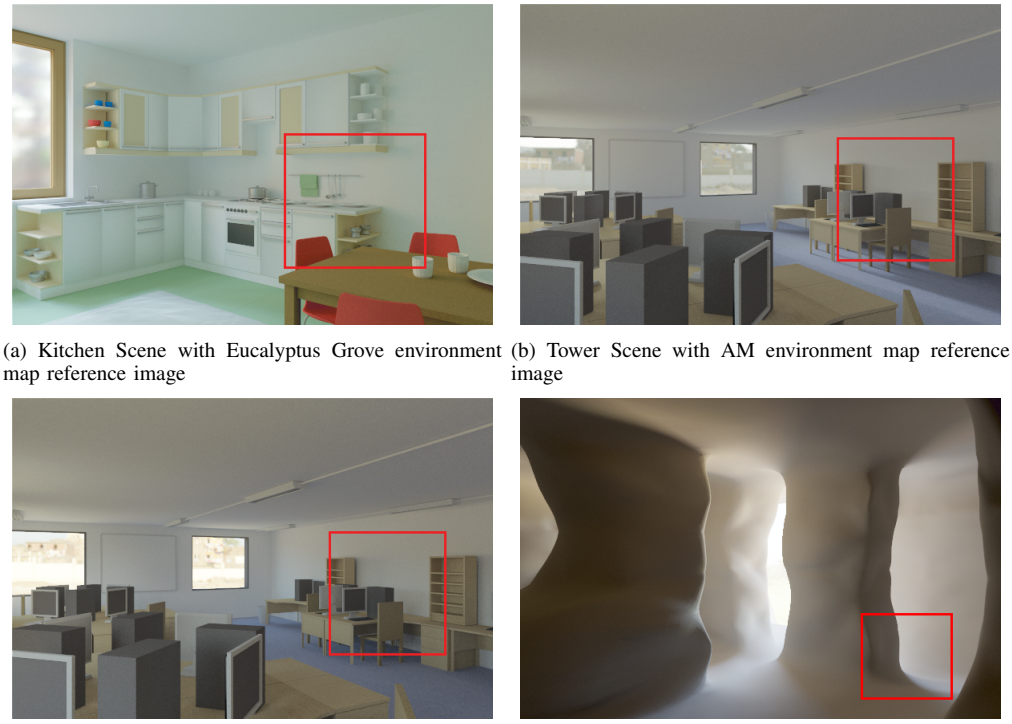
$$p_d(i|\omega_l) = \frac{1}{-(\omega_l \cdot \omega_d(i))\pi R_d(i)^2}. \tag{10}$$

Note this PDF is only required when $w(i|\omega_l) > 0$ and so $-(\omega_l \cdot \omega_d(i))$ is therefore guaranteed to be positive. When $w(i|\omega_l) = 0$, the PDF is not required. $p(bounds)$ is calculated as by Dammertz and Hanika [1].

## 6 Results

This section presents results for IDES. All results were calculated on an Intel Xeon E5-2687W running at 3.1GHz and 32GB RAM. IDES was compared to Plane sampling [1] and Bounding Disk sampling [11]. Starting directions were generated using marginal and conditional PDFs [10]. The directional distribution was calculated based on the luminance of the values in the environment map for the plane and bounding disk sampling methods, and based on the product of the environment map and the path map for IDES. We use values for $\sigma = 1/256$ and $K = 5$ for the values in Section 5.1, and $w(bounds) = 0.001$ from Section 5.2.

All scenes use lighting from captured environment maps, and no other light sources. Three rendering algorithms were



(a) Kitchen Scene with Eucalyptus Grove environment map reference image

(b) Tower Scene with AM environment map reference image

(c) Tower Scene with PM environment map reference image

(d) Mountain Scene with noon environment map reference image

Fig. 4. Reference images for the results scenes. Further results are shown below for the highlighted regions.

used to demonstrate results. The first is Bidirectional Path Tracing (BPT) [3], [4], where sub paths are started at the camera and the lights (in our case the environment map), and vertices from the sub paths are connected to form a full path. The second algorithm is Progressive Photon Mapping (PPM) [7], where the first pass generates camera hit points, and then all paths in the remaining passes sample from the light sources in a manner akin to photon mapping; again with all paths starting from the environment map. The third algorithm is Primary Sample Space Metropolis Light Transport (MLT) [32] to demonstrate that IDES works with Markov Chain Monte Carlo methods which efficiently explore path space.

We demonstrate results for four scenes, see Figure 4. Results are shown in Table 2 for the highlighted region in Figure 4(a) after 30 minutes of rendering for the various methods. Firstly, the Kitchen scene is lit through a single window by the Eucalyptus Grove environment map. This serves to illustrate the effectiveness of IDES when there are strong sources of light from several directions around the environment map.

The Tower block scene is a large architectural type model with the camera located in one of the corner rooms. This is lit through several windows on two sides of the building, see Figure 1(a). This is a challenging but realistic (from an architectural perspective) scenario, as the windows in the room in which the camera is located only constitutes a small percentage of the outside area. This leads to a low probability of a light path entering the room using both plane sampling and bounding disk sampling. We show results with two environment maps; one in the morning (Figure 4(b) and Table 3) when sun is shining at the room, then one in the afternoon (Figure 4(c) and Table 4) when the sun is facing the other side of the building. The PM environment map scene is especially a challenge for traditional methods which generate samples based on the luminance of the environment map. These generate most samples in directions which cannot be seen by the camera. The path map described in Section 4 largely alleviates this problem by generating the majority of the samples in directions which are visible to the camera.

Finally, the Mountain scene shows a small cave in a large mountain, where the camera is located inside the cave, as is shown in Figure 5. This is challenging as there is a very small probability of light paths entering the cave without the guidance of IDES, which is reflected in the results (Table 5).

We show RMSE results compared with a fully converged image in each of the figures. In order to take into account both time taken for sampling, and variance reduction properties of the algorithms, we use efficiency [11]. This is calculated as $\frac{1}{time \times RMSE^2}$, and is averaged over a half hour rendering for BPT, PPM and MLT. The efficiency values for all the methods, and a summary of the speed-ups in terms of efficiency for BPT, PPM, and MLT for the scenes are shown in Table 1. Both RMSE and Efficiency were calculated over the entire image.

Results show an improvement in both RMSE and efficiency for all scenes, due to IDES improving direction and position selection. Bi-directional path tracing results

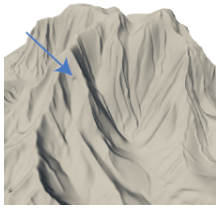


Fig. 5. Model of the mountain scene. The arrow shows the opening of the cave.

indicate a large improvement in efficiency, especially for more complicated scenes where the results in this paper indicate an improvement of one to four orders of magnitude improvement in efficiency. PPM performs similarly, with most scenes showing around an order of magnitude improvement in efficiency. Even MCMC methods show an improvement, as demonstrated by the MLT results. MLT alone can be very effective in the challenging scenarios presented in this paper; however the addition of IDES still gains a $1.3\times$ to $12.9\times$ improvement in efficiency. This is due more path perturbations being accepted, due to the guidance from IDES's distributions.

In terms of efficiency, IDES does face a couple of additional costs over simpler algorithms. Firstly, the pre-processing time; although small in comparison to the overall rendering time, slightly lowers the efficiency of the algorithm. Nevertheless, this has little impact on even simple scenes. Specifically, the Kitchen scene took $0.4s$, the Tower scene took $1.5s$ and the Mountain scene was $1.3s$. $70-80\%$ of this time is spent on tracing rays; the rest is building the path map and disk distribution. Secondly, there is also an overhead for sampling the disks; however, this is negligible compared to the benefits of the IDES. Note that these overheads are taken into account in the efficiency results.

To test how IDES behaves when using differing scene configurations, we performed two experiments. One was to test the performance of IDES with differing window sizes, and the second considered the effectiveness of the algorithm when scenes contained an object of varying glossiness (using a modified Phong BRDF [33]). Results are shown in Table 6. This illustrates that the improvement of IDES, compared to the other algorithms, increases when lighting configurations are more complicated, as is expected. Glossy objects do not significantly affect the performance of the algorithm, as is shown by an almost constant difference between methods.

## 7 Discussion and Future Work

Although, as shown in the previous section, IDES delivers significant speed-ups compared to other algorithms, sampling efficiency is not ideal, as both the directional and position distributions are not perfect due to the finite number of paths in the pre-pass. A solution to this is to adapt the distributions as rendering proceeds to more closely match the ideal distributions. However, one has to keep in mind that runtime adaptation has an impact on certain rendering algorithms (for instance combining MLT with adaptation is non-trivial). Runtime adaptation also has overheads, and the improvement in efficiency has to justify these overheads. We have not implemented runtime adaptation in our results, as for all the scenes tested we found the initial distributions to be good enough to generate significant speedups.

Sampling and calculation of the PDF for the disk representation does add time to the sampling process, although the improvements in efficiency negate this slow-down. However, as future work, we intend to store the disk representation in a tree data structure. Sample generation could be performed via a traversal of the tree, thereby lazily calculating weights; and PDF construction could be accelerated in a similar manner. This would turn the sample generation from $O(N)$ to close to $O(log(N))$, where $N$ is the number of disks used in the position distribution. However, this has to be balanced by increased pre-computation costs and approximate weight generation, leading to a decrease in useful samples compared to the approach in this paper.

Another potential source of inefficiency in IDES is that directions coming from the entire hemisphere above each disk are considered for sampling. In scenes with very complicated occlusion, this can lead to less efficiency when rendering. As future work, we intend to encode visibility information from the environment with each disk, and use this to generate improved weights when sampling based on the incoming direction. Alternatively, the environment could be quantized into a set of directions, and only disks visible from those directions could be sampled.

## 8 Conclusions

This paper has presented a method which improves the efficiency of generating paths starting at an environment light source. Based on a short pre-pass, starting directions are sampled from a distribution which takes into account both the luminance of the pixels in the environment map, and the contributions of the environment map to the camera. Similarly, starting positions are generated using an automatically determined disk distribution. Results illustrate that the combination of these distributions reduce variance effectively over a variety of scenes.

## Acknowledgements

We would like to thank the anonymous reviewers for their comments. Thanks to Jassim Happa for the Tower and Mountain scenes, and to Paul Debevec for the Eucalyptus Grove environment map. This work was supported by EPSRC grant EP/I038780/1.

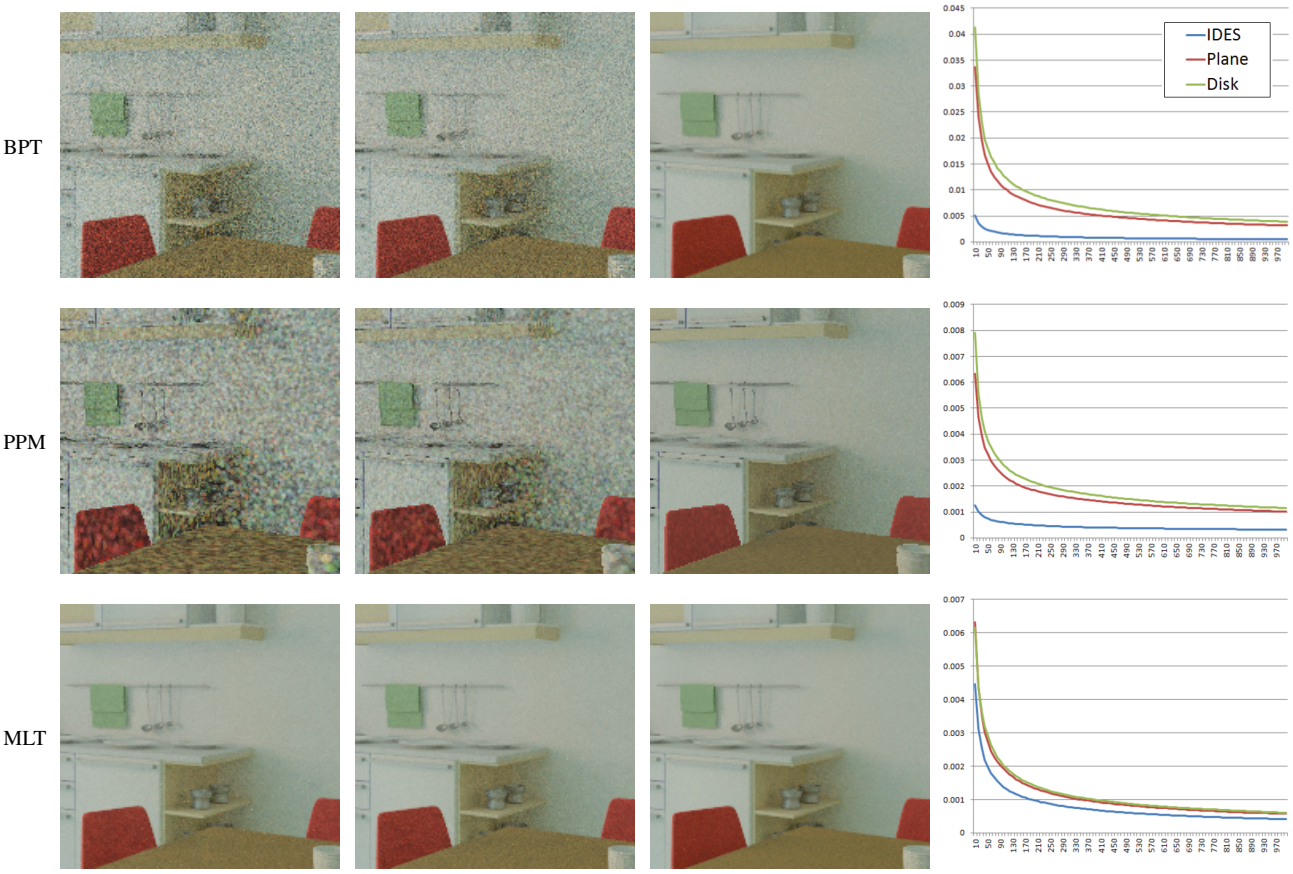
TABLE 1

Summary of efficiency results (calculated as $\frac{1}{time \times RMSE^2}$). The numbers in brackets show times improvement of IDES versus the other algorithms.

| Scene | BPT | | | PPM | | | MLT | | |
|---|---|---|---|---|---|---|---|---|---|
| | Bounding Disk | Plane | IDES | Bounding Disk | Plane | IDES | Bounding Disk | Plane | IDES |
| Kitchen | 135 (28.5×) | 196 (19.7×) | 3858 | 1230 (16.1×) | 2205 (9.0×) | 19811 | 3717 (1.6×) | 3733 (1.6×) | 5905 |
| TowerAM | 60 (64.1×) | 79 (48.4×) | 3833 | 1102 (38.7×) | 1433 (28.8×) | 42679 | 3209 (1.9×) | 3148 (1.9×) | 6127 |
| TowerPM | 97 (151.8×) | 150 (97.9×) | 14721 | 1779 (17.0×) | 3578 (8.5×) | 30269 | 5804 (1.3×) | 4576 (1.7×) | 7608 |
| Mountain | 0.3 (16671.8×) | 0.5 (10594.1×) | 5792 | 2674 (31.9×) | 4260 (20.0×) | 85363 | 1020 (12.9×) | 2116 (6.2×) | 13187 |

TABLE 2

Kitchen Scene with Eucalyptus Grove environment map after 30 minutes rendering. The horizontal axis on the graphs shows SPP and the vertical is RMSE.

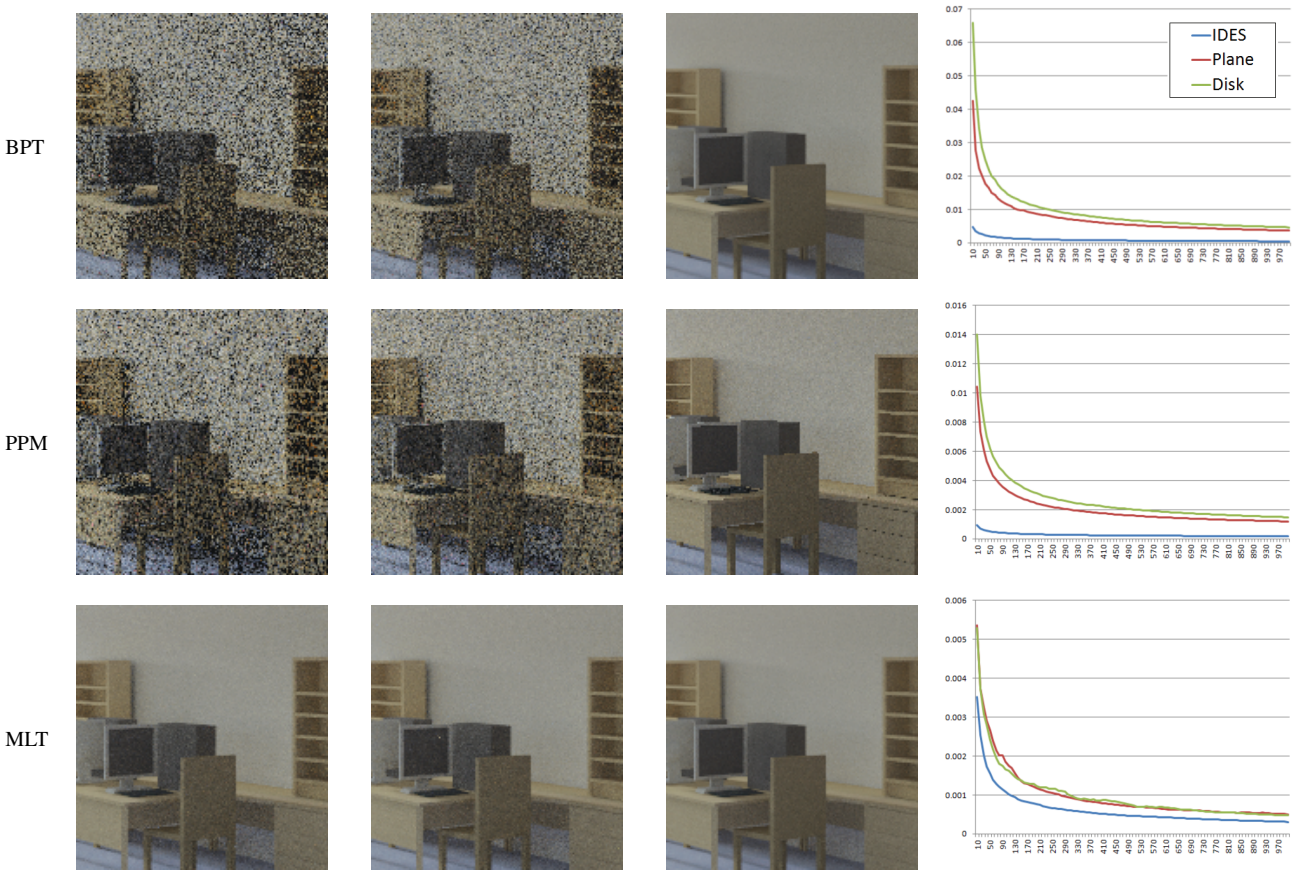



Bounding Disk Plane IDES RMSE

**Thomas Bashford-Rogers** has a degree in Computer Science from the University of Bristol and a doctorate in Computer Graphics from the University of Warwick. His research interests include global illumination, raytracing and Monte Carlo methods. He is currently a Research Fellow at the University of Warwick.

**Kurt Debattista** received his PhD degree in Computer Science from the University of Bristol. He is an Associate Professor at the University of Warwick. His research interests include physically-based rendering, interactive global illumination, high dynamic range imaging, and parallel computing.

TABLE 3
Tower scene with the AM environment map after 30 minutes rendering. The horizontal axis on the graphs shows SPP and the vertical is RMSE.




Bounding Disk | Plane | IDES | RMSE

**Alan Chalmers** is a Royal Society Industrial Fellow and a Professor of Visualisation at WMG, University of Warwick, UK. He has an MSc with distinction from Rhodes University, 1985 and a PhD from University of Bristol, 1991. He is Honorary President of Afrigraph and a former Vice President of ACM SIGGRAPH. He is Founder and Innovation Director of goHDR Ltd. His research goal is "Real Virtuality", delivering physically-based, multisensory, realistic virtual environments at interactive rates.

## References


[1] H. Dammertz and J. Hanika, "Plane Sampling for Light Paths from the Environment Map," *Journal of Graphics, GPU, & Game Tools*, vol. 14, no. 2, pp. 25–31, 2009.

[2] P. E. Debevec, "Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography," in *SIGGRAPH*, 1998, pp. 189–198.

[3] E. Veach and L. Guibas, "Bidirectional estimators for light transport," in *Proceedings of the Fifth Eurographics Workshop on Rendering*. Eurographics, Jun. 1994, pp. 147–162.

[4] E. P. Lafortune and Y. D. Willems, "Bi-directional path tracing," in *Proceedings Of Third International Conference On Computational Graphics And Visualization Techniques (Compugraphics '93)*, 1993, pp. 145–153.

[5] J. Jensen, "Realistic Image Synthesis Using Photon Mapping," *AK Peters Natick, Massachusetts*, 2001.

[6] A. Keller, "Instant radiosity," *Computer Graphics*, vol. 31, no. Annual Conference Series, pp. 49–56, 1997.

[7] T. Hachisuka, S. Ogaki, and H. Jensen, "Progressive photon mapping," in *ACM SIGGRAPH Asia 2008 papers*. ACM, 2008, pp. 1–8.

[8] E. Veach and L. J. Guibas, "Metropolis light transport," *SIGGRAPH ( ACM Transactions on Graphics)*, pp. 65–76, 1997.

[9] P. Debevec, "A median cut algorithm for light probe sampling," in *ACM SIGGRAPH 2005 Posters*, ser. SIGGRAPH '05. ACM, 2005.

[10] J. Lawrence, S. Rusinkiewicz, and R. Ramamoorthi, "Adaptive numerical cumulative distribution functions for efficient importance sampling," in *Eurographics Symposium on Rendering*, 2005, pp. 11–20.

[11] M. Pharr and G. Humphreys, *Physically Based Rendering from Theory to Implementation (Second Edition)*. Morgan-Kaufmann, 2010.

[12] A. Ghosh, A. Doucet, and W. Heidrich, "Sequential sampling for dynamic environment map illumination," *Rendering Techniques 2006*, p. 115, 2006.

[13] A. Ghosh and W. Heidrich, "Correlated visibility sampling for direct illumination," *The Visual Computer*, vol. 22, no. 9, pp. 693–701, 2006.

[14] P. Dutré and Y. D. Willems, "Importance-driven Monte Carlo light tracing," in *Proceedings of the Fifth Eurographics Workshop on Rendering*. Eurographics, Jun. 1994, pp. 185–194.

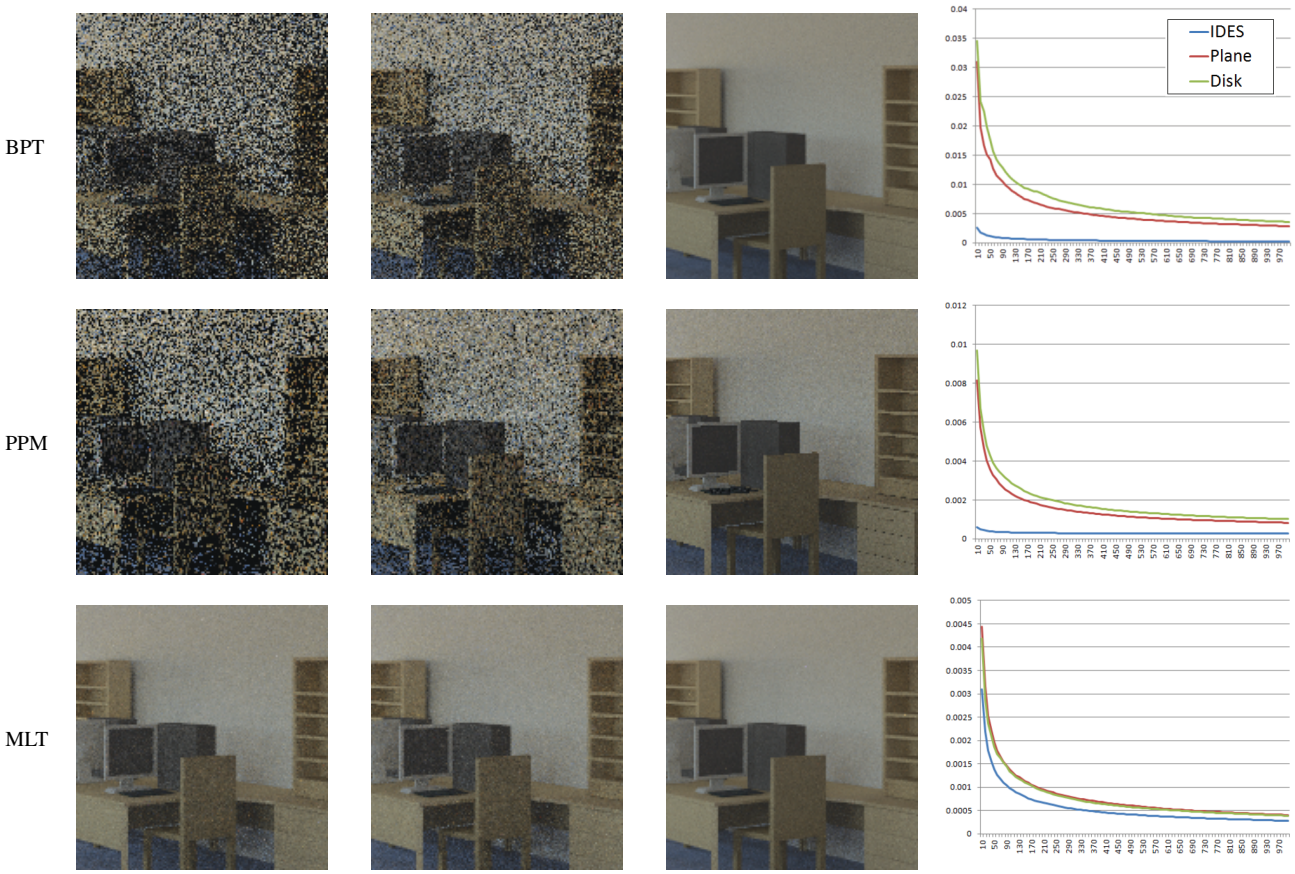
[15] G. Ward, "Adaptive shadow testing for ray tracing," in *Proceedings of the Second Eurographics Workshop on Rendering*, 1991.

[16] D. Burke, A. Ghosh, and W. Heidrich, "Bidirectional importance sampling for direct illumination," in *Proceedings of Eurographics Symposium on Rendering 2005*. Eurographics, June 2005.

[17] J. Talbot, D. Cline, and P. K. Egbert, "Importance resampling for global illumination," in *Rendering Techniques*, 2005, pp. 139–146.

[18] P. Clarberg, W. Jarosz, T. Akenine-Möller, and H. Jensen, "Wavelet

TABLE 4
Tower scene with the PM environment map after 30 minutes rendering. The horizontal axis on the graphs shows SPP and the vertical is RMSE.





importance sampling: efficiently evaluating products of complex functions," in *ACM SIGGRAPH 2005 Papers*. ACM, 2005, pp. 1166–1175.

[19] R. Schregle, *Daylight simulation with photon maps*, 2004.

[20] Y. Yue, K. Iwasaki, B. Chen, Y. Dobashi, and T. Nishita, "Interactive Rendering of Interior Scenes with Dynamic Environment Illumination," in *Computer Graphics Forum*, vol. 28, no. 7. Wiley Online Library, 2009, pp. 1935–1944.

[21] D. Cohen-Or, Y. Chrysanthou, C. Silva, and F. Durand, "A survey of visibility for walkthrough applications," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 9, no. 3, pp. 412–431, 2003.

[22] S. Fan, S. Chenney, and Y.-C. Lai, "Metropolis photon sampling with optional user guidance," in *Rendering Techniques '05 (Proceedings of the 16th Eurographics Symposium on Rendering)*. Eurographics Association, 2005, pp. 127–138.

[23] J. Chen, B. Wang, and J. Yong, "Improved stochastic progressive photon mapping with metropolis sampling," in *Computer Graphics Forum*, vol. 30, no. 4. Wiley Online Library, 2011, pp. 1205–1213.

[24] T. Hachisuka and H. Jensen, "Stochastic progressive photon mapping," *ACM Transactions on Graphics (TOG)*, vol. 28, no. 5, pp. 1–8, 2009.

[25] ——, "Robust adaptive photon tracing using photon path visibility," *ACM Transactions on Graphics (TOG)*, vol. 30, no. 5, p. 114, 2011.

[26] I. Georgiev, J. Křivánek, T. Davidovič, and P. Slusallek, "Light transport simulation with vertex connection and merging," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 192:1–192:10, Nov. 2012.

[27] T. Hachisuka, J. Pantaleoni, and H. W. Jensen, "A path space extension for robust light transport simulation," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 6, p. 191, 2012.

[28] I. Georgiev, J. Křivánek, S. Popov, and P. Slusallek, "Importance caching for complex illumination," *Computer Graphics Forum*, vol. 31, no. 2, 2012, eUROGRAPHICS 2012.

[29] E. Veach, "Robust monte carlo methods for light transport simulation," Ph.D. dissertation, 1997.

[30] W. Jakob and S. Marschner, "Manifold exploration: a markov chain monte carlo technique for rendering scenes with difficult specular transport," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, p. 58, 2012.

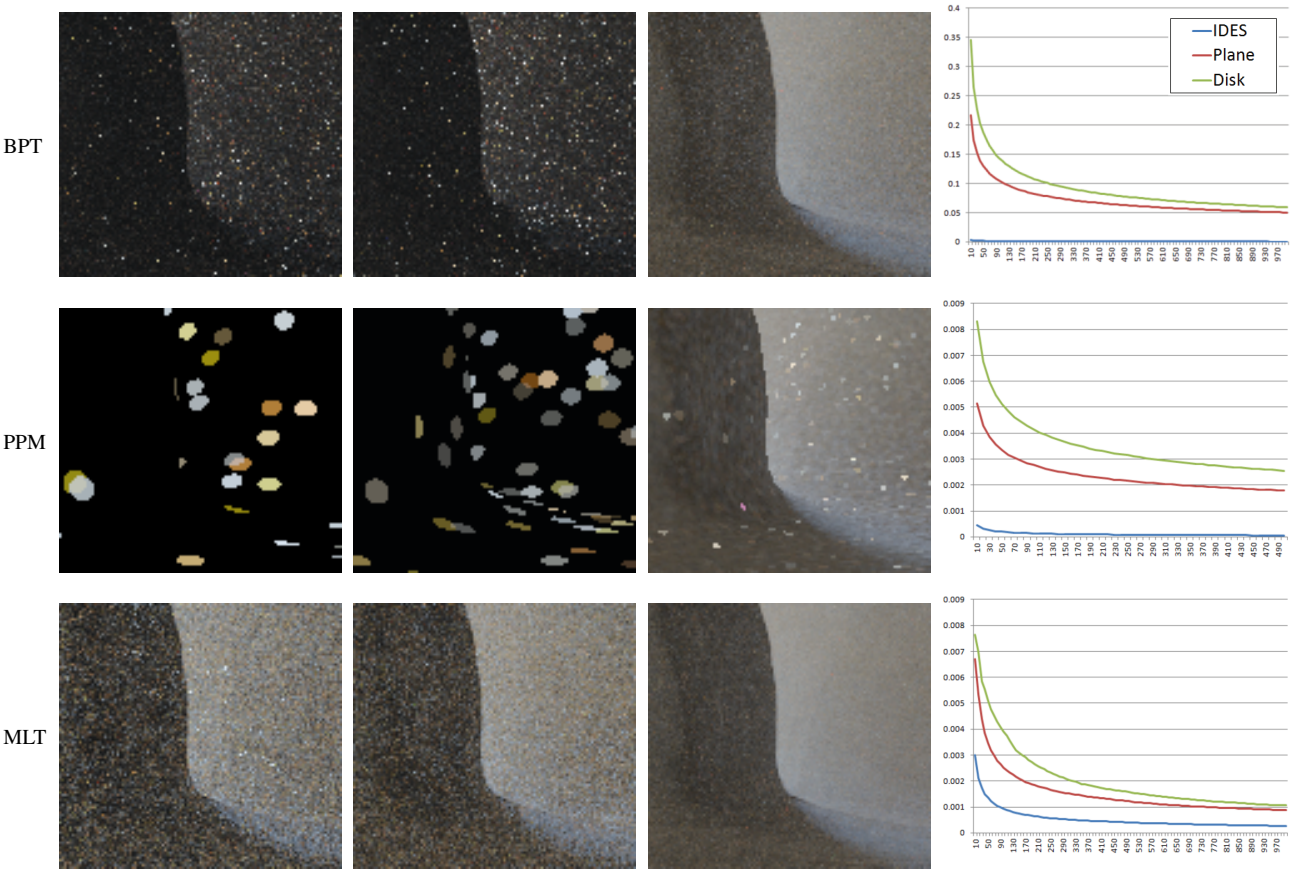
[31] I. Peter and G. Pietrek, "Importance driven construction of photon maps," in *Rendering Techniques '98 (Proceedings of the Ninth Eurographics Workshop on Rendering)*. Springer-Verlag, 1998, pp. 269–280.

[32] C. Kelemen, L. Szirmay-Kalos, G. Antal, and F. Csonka, "A simple and robust mutation strategy for the metropolis light transport algorithm," in *Computer Graphics Forum*, vol. 21, 2002, pp. 531–540.

[33] E. Lafortune and Y. Willems, "Using the modified phong reflectance model for physically based rendering," 1994.

TABLE 5

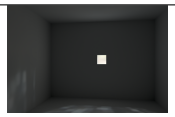
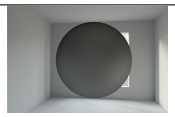
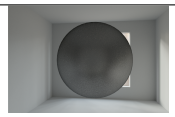
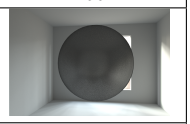
Mountain scene after 30 minutes rendering. The horizontal axis on the graphs shows SPP and the vertical is RMSE.




Bounding Disk | Plane | IDES | RMSE

TABLE 6

Experiment results for varying window sizes, and a scene containing an object of differing glossy levels (the numbers refer to the Phong exponent). **PC** refers to pre-computation time (path tracing pre-pass, path map construction, and disks construction). The bottom three rows are RMSE values (computed after 10 minutes of rendering).

| Scene | Small Window | Medium Window | Large Window | 5 | 50 | 100 |
|---|---|---|---|---|---|---|
| Reference |  |  |  |  |  |  |
| **PC** (s) | 0.34 | 0.59 | 2.06 | 1.39 | 1.38 | 1.38 |
| IDES | 0.000077 | 0.000270 | 0.001338 | 0.000444 | 0.000645 | 0.000715 |
| Plane | 0.000147 | 0.000704 | 0.002287 | 0.001179 | 0.001640 | 0.002066 |
| Bounding Disk | 0.000182 | 0.000917 | 0.002627 | 0.001668 | 0.001814 | 0.002377 |