A Path Space Extension for Robust Light Transport Simulation

Toshiya Hachisuka[1,3] Jacopo Pantaleoni[2] Henrik Wann Jensen[3]
[1]Aarhus University [2]NVIDIA Research [3]UC San Diego

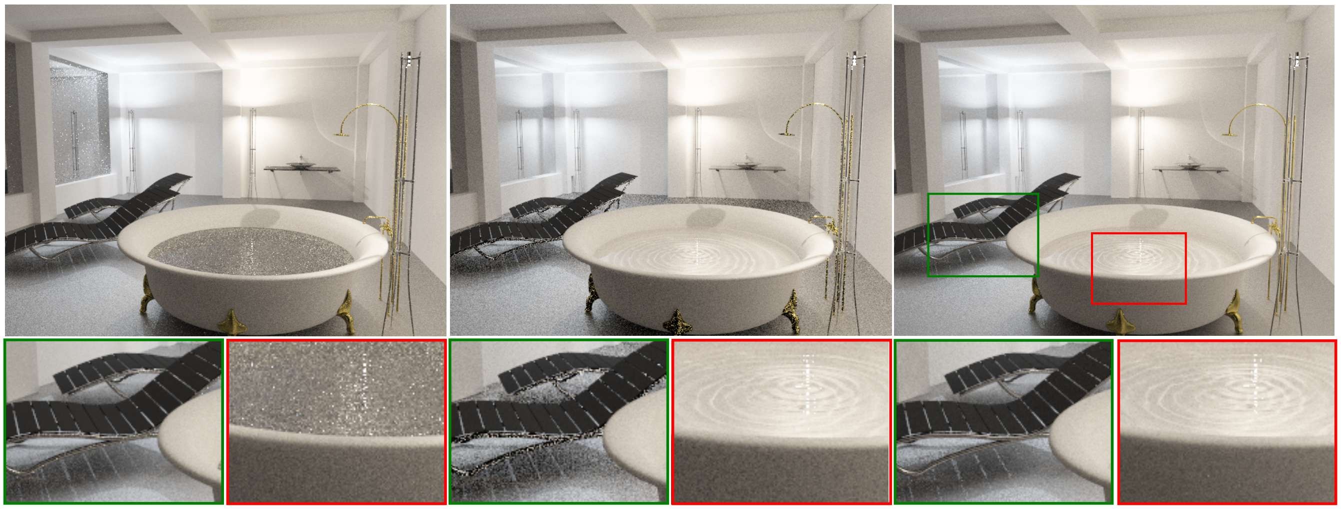

Monte Carlo Path Integration (RMSE: 0.07836)
Photon Density Estimation (RMSE: 0.04638)
Unified Framework (RMSE: 0.01246)

Figure 1: *Equal-time comparison of rendered images of a bathroom scene with realistic lighting fixtures. This scene includes both glossy reflections and complex caustics due to lighting fixtures which are common in interior design. Existing light transport simulation methods including Monte Carlo path integration and photon density estimation cannot efficiently render scenes with such lighting phenomena. Our new framework for light transport simulation automatically combines Monte Carlo path integration and photon density estimation by extending the sampling space of light transport paths, and produces a significantly more accurate solution in the same rendering time.*

Abstract

We present a new sampling space for light transport paths that makes it possible to describe Monte Carlo path integration and photon density estimation in the same framework. A key contribution of our paper is the introduction of vertex perturbations, which extends the space of paths with loosely coupled connections. The new framework enables the computation of path probabilities in the same space under the same measure, which allows us to use multiple importance sampling to combine Monte Carlo path integration and photon density estimation. The resulting algorithm, *unified path sampling*, can robustly render complex combinations and glossy surfaces and caustics that are problematic for existing light transport simulation methods.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Raytracing;

Keywords: global illumination, multiple importance sampling

1 Introduction

Efficiently simulating light transport under general scene configurations is a difficult task. Currently, the most successful approaches are based on Monte Carlo path integration and photon density estimation. Both approaches solve the rendering equation introduced by Kajiya [1986]. Unfortunately, neither Monte Carlo path integration nor photon density estimation can simulate *all* types of light transport efficiently. For example, Monte Carlo path integration has problems with specular-diffuse-specular (SDS) transport, while photon density estimation techniques can suffer in the presence of highly glossy materials.

Recently, Hachisuka and Jensen [2009] introduced stochastic progressive photon mapping. They combined photon density estimation and distributed ray tracing [Cook et al. 1984] to solve the rendering equation in general scenes. This algorithm can handle both SDS paths and glossy reflections efficiently, however, the algorithm uses a manual classification of surface materials into either "glossy" or "diffuse". This binary classification can lead to inefficient sampling in some scenes such as glossy surfaces lit by a directional light source.

A key challenge in combining density estimation and Monte Carlo path integration is the lack of a shared theoretical foundation for both approaches. It is well known that both approaches solve the rendering equation. However, while Monte Carlo path integration has a solid theoretical foundation with the path integral formulation [Veach 1998], photon density estimation has not been formulated under the same foundation. This difference in their theoretical formulations limits the ability to analyze both methods in the same framework.

In this paper, we propose a novel path sampling framework for light transport. The key contribution is a new extension of the path space using *vertex perturbations*, which can describe both density estimation and Monte Carlo path integration techniques under the same unified framework. The insight behind the use of vertex perturbations is that density estimation can be seen as adding a loosely coupled connection to Monte Carlo path integration. Likewise, Monte Carlo path integration can be extended to density estimation by introducing a similar loosely coupled connection, rather than the traditional explicit connection between two path vertices.

This new path sampling framework provides a unified formulation for both photon density estimation and Monte Carlo path integration. Our framework thus allows us to evaluate path probabilities in the same space under the same measure, which makes it possible to use multiple importance sampling [Veach and Guibas 1995] to combine these two approaches. We call this combination *unified path sampling*.

Although one could attempt the use of multiple importance sampling without extending the path space, this attempt would result in a combination that is not scale invariant due to the lack of a proper measure as we will elaborate. This lack of measure is one of the key reasons why Monte Carlo path integration and photon density estimation are usually implemented as separate rendering frameworks. Our unified path sampling algorithm overcomes this limitation. Figure 1 demonstrates how the resulting algorithm can efficiently render both caustics and glossy reflections.

2 Theory

In the following, we first briefly summarize the formulation of multiple importance sampling. We then formally describe the differences between Monte Carlo path integration and photon density estimation which have prevented us to combine these two methods using multiple importance sampling. Finally, to resolve these differences, we introduce the path space extension that expresses the two techniques under a unified definition. Figure 2 illustrates the core idea. Table 1 summarizes the notations used throughout the paper.

Multiple importance sampling [Veach and Guibas 1995] is a powerful tool that enables combining multiple Monte Carlo integration techniques with different probability density functions in order to solve an integral. Given an integral I of the function f

$$I = \int_{\Omega} f(x) d\mu(x) \tag{1}$$

over some domain Ω and a measure μ, multiple importance sampling combines N_t different techniques to generate samples in this same domain Ω.

To build an estimator for the given integral, multiple importance sampling weights contributions of individual samples from different techniques, where each i-th technique has a different probability density function $p_i(\vec{x})$ and approximates different parts of the integrand f better than the others.

To be precise, if the i-th technique is used to generate n_i samples $\{X_{i,k} : i = 1, \ldots, N_t, k = 1, \ldots, n_i\}$, multiple importance sampling gives us the unbiased estimator of I as

$$I = \mathrm{E}\left[\sum_{i=1}^{N_t} \frac{1}{n_i} \sum_{k=1}^{n_i} w_i(X_{i,k}) \frac{f(X_{i,k})}{p_i(X_{i,k})}\right], \tag{2}$$

as long as $\sum_{i=1}^{N_t} w_i(\vec{x}) = 1$ and $w_i(\vec{x}) = 0$ whenever $p_i(\vec{x}) = 0$. In other words, multiple importance sampling will give us a correct estimator of the integral as long as weights sum to one and each estimator is valid.

Efficient weighting strategies such as the balance and power heuristics [Veach and Guibas 1995] compute the weights by evaluating probability densities associated with different sampling techniques at the same sample. Monte Carlo path integration fully adopted such efficient weighting strategies in bidirectional path tracings [Lafortune and Willems 1993; Veach and Guibas 1995]. On the other hand, photon density estimation and its progressive variants [Jensen 1996; Hachisuka et al. 2008; Knaus and Zwicker 2011] typically do not use multiple importance sampling, and rely on a simpler heuristic such as caustics/non-caustics classification.

Only recently Vorba and Křivánek [2011] showed how multiple importance sampling can be used to combine different photon density estimation techniques. However, this combination is still limited within photon density estimation techniques themselves. Some other partial combinations of photon density estimation and bidirectional path tracing have been proposed, and they demonstrated more robust alternatives than using one of the algorithms alone [Bekaert et al. 2003; Tokuyoshi 2009; Hachisuka and Jensen 2009].

Our focus in this paper is to provide a mathematical framework that unifies these two algorithms using multiple importance sampling. In order to do so, we seek for the common space where we can compare probability densities of generating paths under two different algorithms. We then use these probability densities to find weights for multiple importance sampling and construct the combined estimator.

Notation	Description
y_i	ith vertex generated from the light source
z_i	ith vertex generated from the eye
K	density estimation kernel
M	path length
p^{mc}	probability density in MC integration
p^{de}	probability density in density estimation
p^{ups}	probability density in unified path space
C_s^L	throughput of the light path with s vertices
C_t^E	throughput of the eye path with t vertices

Table 1: *Descriptions of the notations used in the paper.*

2.1 Path Integration and Density Estimation

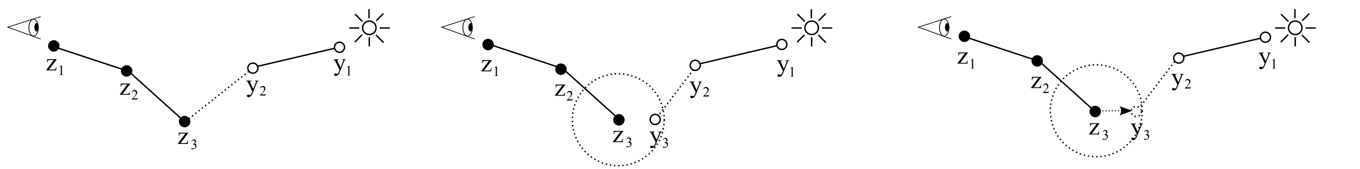
Path Integration: According to the path integral formulation [Veach 1998], light transport simulation can be expressed as

$$I_j = \int_{\Omega} f_j(\bar{x}) d\mu(\bar{x}), \tag{3}$$

where j is an index to the pixel, f_j is the measurement contribution function, and μ is a measure of paths. Ω in this setting is the set of transport paths of all lengths. For formal definitions, please refer to Appendix A.1 in the supplemental document.

Bidirectional path tracing [Lafortune and Willems 1993; Veach and Guibas 1995] solves this integral via Monte Carlo integration. Given a path of length M, bidirectional path tracing considers all the possible *techniques* to sample a path by joining two subpaths with s vertices from the light source and t vertices from the eye, such that $s + t - 1 = M$.

Let us focus on a single technique (s, t), where we have a light subpath $\bar{y}_s = y_1 \ldots y_s$ and an eye subpath $\bar{z}_t = z_t \ldots z_1$. By convention, y_1 is at the light source and z_1 is at the eye, thus the complete path is $\bar{x}_{s,t} = \bar{y}_s \bar{z}_t = y_1 \ldots y_s z_t \ldots z_1$. The overall



(a) A path in MC path integration (b) Corresponding path in density estimation (c) Corresponding path in our extended space

Figure 2: *Path space extension for unifying Monte Carlo path integration and photon density estimation. A path sampled in Monte Carlo path integration corresponds to one of the infinitely many possible paths in photon density estimation. The probability density of the path in (a) is not in the same space as the corresponding path in (b). We propose to use an extended path space of Monte Carlo path integration by considering a random perturbation of the last connecting light vertex within the neighborhood of the originally sampled vertex as in (c). With this extension, both Monte Carlo path integration and photon density estimation cover exactly the same path space.*

throughput of this path is

$$\begin{aligned} & C^{\mathrm{mc}*}_{s,t}(\bar{x}_{s,t}) = f_j(\bar{x}_{s,t}) \\ & = C^L_s(\bar{y}_s) \\ & \cdot f(y_{s-1} \to y_s \to z_t) G(y_s, z_t) f(y_s \to z_t \to z_{t-1}) \\ & \cdot C^E_t(\bar{z}_t) \end{aligned} \tag{4}$$

with

$$\begin{aligned} C^L_s(\bar{y}_s) &= L_e(y_1 \to y_2) G(y_1, y_2) \textstyle\prod_{p=1}^{s-1} f_s(y_{p-1} \to y_p \to y_{p+1}) G(y_p, y_{p+1}) \\ C^E_t(\bar{z}_t) &= W^j_e(z_1 \to z_2) G(z_1, z_2) \textstyle\prod_{p=1}^{t-1} f_s(z_{p-1} \to z_p \to z_{p+1}) G(z_p, z_{p+1}). \end{aligned}$$

The geometry term G includes the visibility term similar to the definition by Veach an Guibas [1995] and the special cases of $s = 0, 1$ and $t = 0, 1$ are also similarly defined.

Assuming that the light path sampling procedure and the eye path sampling procedure are statistically independent, local path sampling methods, the probability density of the whole path is the product of the subpath probability densities:

$$p^{\mathrm{mc}}_{s,t}(\bar{x}_{s,t}) = \left(p(y_1) \prod_{i=1}^{s-1} p(y_{i+1}|y_i) \right) \left(p(z_1) \prod_{i=1}^{t-1} p(z_{i+1}|z_i) \right). \tag{5}$$

We can combine samples from different techniques by weighting their contributions using multiple importance sampling:

$$I_j = \mathrm{E}\left[\sum_{M=1}^{\infty} \sum_{s=0}^{M+1} w_{s,t}(\bar{x}_{s,t}) \frac{C^{\mathrm{mc}*}_{s,t}(\bar{x}_{s,t})}{p^{\mathrm{mc}}_{s,t}(\bar{x}_{s,t})} \right], \tag{6}$$

where we defined $t = M + 1 - s$ for simplifying the notation. Note that we take one sample from each estimator by setting $n_i = 1$ in Equation 2 as in the original formulation by Veach and Guibas.

Density Estimation: Solutions to density estimation methods like photon mapping can also be expressed as solutions of an integral [Silverman 1986]. We can thus express light transport simulation using photon density estimation as:

$$I'_j = \int_{\Omega'} f'_j(\bar{x}) d\mu(\bar{x}) = I_j + B_j, \tag{7}$$

where f'_j is the *extended* measurement contribution function that includes the influence of photon density estimation and Ω' is also the *extended* path space. B_j is bias associated with density estimation (refer to Appendix A.2 in the supplemental document for details).

To build a path of length M with density estimation at the last eye vertex z_t, we need to have a light subpath with $s + 1 = M + 2 - t$ vertices $\bar{y}_{s+1} = y_1 \ldots y_{s+1}$, and an eye subpath $\bar{z}_t = z_t \ldots z_1$. The complete path is given as $\bar{x}_{s+1,t} = \bar{y}_{s+1}\bar{z}_t = y_1 \ldots y_{s+1} z_t \ldots z_1$. The subpaths are combined via density estimation using y_{s+1} and z_t, instead of establishing an explicit connection between y_{s+1} and z_t. The overall throughput of this path is

$$\begin{aligned} & C^{\mathrm{de}*}_{s+1,t}(\bar{x}_{s+1,t}) = f'_j(\bar{x}_{s+1,t}) \\ & = C^L_{s+1}(\bar{y}_{s+1}) \\ & \cdot K(y_{s+1}, z_t) f(y_s \to y_{s+1}, z_t \to z_{t-1}) \\ & \cdot C^E_t(\bar{z}_t) \end{aligned} \tag{8}$$

where K is the density estimation kernel, and the light and eye subpath contributions are defined as Equation 4. The definition of the probability density is the same as before[1]:

$$p^{\mathrm{de}}_{s+1,t}(\bar{x}_{s+1,t}) = \left(p(y_1) \prod_{i=1}^{s} p(y_{i+1}|y_i) \right) \left(p(z_1) \prod_{i=1}^{t-1} p(z_{i+1}|z_i) \right). \tag{9}$$

There are three major differences from bidirectional path tracing:

1. To build a path of length M, we need $M + 2$ vertices rather than $M + 1$.
2. The visibility and geometry terms at the connecting edge are replaced by the density estimation kernel (compare Equation 4 and Equation 8).
3. Density estimation adds a certain amount of bias, which depends on the density estimation kernel.

Despite these differences, if we focus only on photon density estimation techniques, it is possible to apply multiple importance sampling. For example, stochastic progressive photon mapping [Hachisuka and Jensen 2009] combines different techniques based on a binary classification of glossy/non-glossy surfaces. Such a classification corresponds to having a binary weight for each technique in multiple importance sampling. Vorba and Křivánek [2011] provide more detailed definitions on how multiple importance sampling can be used to combine photon density estimation samples. Therefore, we can still use multiple importance sampling as:

$$I'_j = \mathrm{E}\left[\sum_{M=1}^{\infty} \sum_{s=0}^{M+2} w_{s,t}(\bar{x}_{s,t}) \frac{C^{\mathrm{de}*}_{s,t}(\bar{x}_{s,t})}{p^{\mathrm{de}}_{s,t}(\bar{x}_{s,t})} \right], \tag{10}$$

where we have $t = M + 2 - s$.

The more interesting question we have set out to solve is whether photon density estimation *and* Monte Carlo path integration can be similarly combined. Our goal thus is to enable multiple importance sampling in this combination.

[1] Note that while the visibility and geometric terms have been substituted by the density estimation kernel in the path contribution, the kernel affects only the integrand, *not* the probability with which the subpaths (and hence the complete paths) are sampled. Even if range queries can be used to efficiently find the subpaths with non-zero contributions, this does not alter the way the paths are stochastically sampled in the first place.

2.2 Path Space Extension

We recognized that the main problem for combining photon density estimation and Monte Carlo path integration is that path samples of a given length M live in spaces of different dimensionality. The reason is that they use different number of vertices to construct a path of the same length. As we show in Appendix A.2 in the supplemental document, we can describe this difference by the difference in measures of their path spaces. For Monte Carlo path integration, the path space has a product measure proportional to A_{M+1}, whereas for photon density estimation the corresponding path space which accounts for the same light transport phenomena has a measure proportional to A_{M+2}.

We resolve this difference by extending the path space of Monte Carlo path integration by means of a *vertex perturbation*. Given a path of length M, obtained by the bidirectional sampling technique (s, t), we add a new vertex y_{s+1} generated by a random perturbation of the vertex z_t with the probability density equal to the density estimation kernel K. For example, if the density estimation kernel is $K = \pi^{-1}r^{-2}$ with the support r, then we randomly sample the disc with the radius r around z_t to generate y_{s+1} (Figure 2). This perturbation kernel should be the same as the density estimation kernel in order to consider the same set of paths in all the techniques.

A new extended light subpath is $\bar{y}'_s = \bar{y}_s y_{s+1} = y_1 \dots y_s y_{s+1}$ and an eye subpath is $\bar{z}_t = z_t \dots z_1$. The extended complete path is $\bar{x} = \bar{y}'_s \bar{z}_t = y_1 \dots y_s y_{s+1} z_t \dots z_1$, which is in the same form as the path in photon density estimation. Figure 2 illustrates this idea.

Given this path space extension, the overall throughput of such a path becomes exactly analogous to that of density estimation (Equation 8):

$$\begin{aligned} C^{\mathrm{ups}*}_{s,t}(\bar{x}) &= f'_j(\bar{x}) \\ &= C^L_{s+1}(\bar{y}') \\ &\cdot K(y_{s+1}, z_t) f(y_s \to y_{s+1}, z_t \to z_{t-1}) \\ &\cdot C^E_t(\bar{z}), \end{aligned} \tag{11}$$

and the probability density follows the same definition as Equation 9:

$$p^{\mathrm{ups}}_{s,t}(\bar{x}) = p(y_1)\prod_{i=1}^{s-1} p(y_{i+1}|y_i) \cdot p(y_{s+1}|y_s) \cdot p(z_1) \prod_{i=1}^{t-1} p(z_{i+1}|z_i), \tag{12}$$

where we defined $p(y_{s+1}|y_s) = K(y_{s+1}, z_t)$ (i.e., the probability density function used to generate the random perturbation). We intentionally left out the term $p(y_{s+1}|y_s)$ from the product, since we can describe both Monte Carlo path integration and photon density estimation by properly defining $p(y_{s+1}|y_s)$ (e.g., vertex perturbation of z_t to obtain y_{s+1}, or local path sampling of y_{s+1} from y_s).

The combined estimator is defined as:

$$I'_j = \mathrm{E}\left[\sum_{u=0}^{1}\sum_{M=1}^{\infty}\sum_{s=0}^{M+2} w_{s,t,u}(\bar{x}_{s,t,u}) \frac{C^{\mathrm{ups}*}_{s,t}(\bar{x}_{s,t,u})}{p^{\mathrm{ups}}_{s,t}(\bar{x}_{s,t,u})}\right], \tag{13}$$

where we have $t = M + 2 - s$. We defined $u = 0$ to identify samples generated by path integration with a vertex perturbation from z_t to y_s;

$$\bar{x}_{s,t,0} = \bar{y}'_{s-1}\bar{z}_t = \bar{y}_{s-1} y_s \bar{z}_t = \bar{y}_s \bar{z}_t., \tag{14}$$

while we defined $u = 1$ to identify samples generated by photon density estimation;

$$\bar{x}_{s,t,1} = \bar{y}_s \bar{z}_t. \tag{15}$$

3 Unified Path Sampling

Using our framework, both (extended) bidirectional path tracing and photon density estimation sample paths of length M in the same space with the measure A_{M+2}. For a given kernel K, we can hence solve the rendering equation using both techniques with a standard multiple importance sampling weighting strategy such as the balance heuristic [Veach 1998].

In order to utilize multiple importance sampling, given a sampled complete path, we need to evaluate the probability densities under *all* the other techniques by which the same path could have been sampled. In our framework, this process involves evaluating probability densities under *both* path integral techniques and density estimation techniques, regardless how the path was actually sampled. Equation 12 handles the differences in these two sets of techniques by the term $p(y_{s+1}|y_s)$.

In the following, we explain several practical considerations that make our framework more useful.

Efficient Reuse of Paths: Instead of sampling a single light subpath and a single eye subpath at a time, we can sample two families of N light and eye subpaths $\{\bar{y}^i\}_{i=1,\dots,N}$ and $\{\bar{z}^j\}_{j=1,\dots,N}$ and reuse them to generate samples from all techniques. This optimization is in a spirit similar to the subpath reuse in the original formulation of bidirectional path tracing [Veach and Guibas 1994].

For each vertex of each eye subpath $\bar{z}^j$, we evaluate all bidirectional path tracing connections with *a single* light path $\bar{y}^j$, and *all* density estimation connections with *all* the N light subpaths $\{\bar{y}^i\}_{i=1,\dots,N}$.

We choose this approach since finding all the density estimation connections with non-zero contributions can be done in sublinear time using efficient range queries. On the other hand, finding such bidirectional connections involves shooting expensive shadow rays and takes $O(N)$ at least. While this change does not affect estimators, it must be reflected in the calculation of the multiple importance sampling weights by multiplying by N the probability density of photon density estimation.

Consistent Estimation: Since our estimator is biased, in order to obtain a *consistent* estimator, we shrink the support of kernel K over time as in the spirit of progressive photon density estimation [2008]. For simplicity, we employ the probabilistic formulation of Knaus and Zwicker [2011] in this step. Note however that our path space extension is not necessarily tied to the use of a progressive scheme.

One minor issue is that, due to the presence of bias, the optimality of the balance heuristic does not apply to our framework. The derivations by Veach [1998] assume that error of the estimator is solely characterized by variance. However, since the error in biased estimators is characterized by both *bias* and *variance*, having bounded variance does not necessarily mean that a combined estimator has bounded error[2].

In the supplementary document, we show a detailed analysis of the problem and provide derivations on how bias in photon density estimation affects this combination. The end result is simple: in order to build a consistent estimator with the same asymptotic behavior as the optimal combination, we should set the alpha parameter for the radius reduction of progressive density estimation [Hachisuka et al. 2008] to $\frac{2}{3}$. We use this alpha value to generate all the results in this paper.

[2] Bias we consider in this paper is bias due to density estimation, not bias dues to uncovered sampling domain.

Virtual Perturbations: Once we set to solve the original light transport equation by using a consistent estimator, we can also *approximate* the contribution of an extended bidirectional path by not actually performing the perturbations to generate paths (i.e., setting $y_{s+1} = z_{z_t}$).

This extra step significantly simplifies the implementation since generating a new vertex such that it will be consistent with photon density is challenging in practice. For example, we need to sample a point on the surface within the support of the same kernel, which is a 3D kernel project on a potentially complicated surface. Virtual perturbations avoid this complication under some assumptions as follows.

Assuming locally uniform radiance distribution as in existing photon density estimation methods [Jensen 1996; Hachisuka et al. 2008; Knaus and Zwicker 2011] and and a constant kernel $K = \pi^{-1} r^{-2}$, we can prove that the contribution of an extended path in Monte Carlo path integration is approximately equal to that of the original Monte Carlo integration.

First, from the definition of the contribution and the probability density in our extended path space, we have

$$\begin{aligned} & \frac{C^{\text{ups}*}_{s+1,t}(\bar{x}_{s+1,t,0})}{p^{\text{ups}}_{s+1,t}(\bar{x}_{s+1,t,0})} \\ &= \frac{C^L_{s+1}(\bar{y}'_{s+1}) K(y_{s+1}, z_t) f(y_s \to z_t \to z_{t-1}) C^E_t(\bar{z}_t)}{p(y_1) \prod_{i=1}^{s-1} p(y_{i+1}|y_i) p(y_{s+1}|y_s) p(z_1) \prod_{i=1}^{t-1} p(z_{i+1}|z_i)}. \end{aligned} \tag{16}$$

Second, since we defined $p(y_{s+1}|y_s) = K(y_{s+1}, z_t)$,

$$\begin{aligned} &= \frac{C^L_{s+1}(\bar{y}'_{s+1}) \cancel{K(y_{s+1}, z_t)} f(y_s \to z_t \to z_{t-1}) C^E_t(\bar{z}_t)}{p(y_1) \prod_{i=1}^{s-1} p(y_{i+1}|y_i) \cancel{K(y_{s+1}, z_t)} p(z_1) \prod_{i=1}^{t-1} p(z_{i+1}|z_i)} \\ &= \frac{C^L_{s+1}(\bar{y}'_{s+1}) f(y_s \to z_t \to z_{t-1}) C^E_t(\bar{z}_t)}{p(y_1) \prod_{i=1}^{s-1} p(y_{i+1}|y_i) p(z_1) \prod_{i=1}^{t-1} p(z_{i+1}|z_i)}. \end{aligned} \tag{17}$$

Finally, we expand the last term in the product in $C^L_{s+1}(\bar{y}')$. We then use the facts that we added a new vertex y_{s+1} by a random perturbation of the vertex z_t and these vertices are interchangeable if the radiance distribution is locally uniform around z_t;

$$\begin{aligned} &= \frac{C^L_s(\bar{y}_s) f_s(y_{s-1} \to y_s \to y_{s+1}) G(y_s, y_{s+1}) f(y_s \to z_t \to z_{t-1}) C^E_t(\bar{z}_t)}{p(y_1) \prod_{i=1}^{s-1} p(y_{i+1}|y_i) p(z_1) \prod_{i=1}^{t-1} p(z_{i+1}|z_i)} \\ &= \frac{C^L_s(\bar{y}_s) f_s(y_{s-1} \to y_s \to z_t) G(y_s, z_t) f(y_s \to z_t \to z_{t-1}) C^E_t(\bar{z}_t)}{p(y_1) \prod_{i=1}^{s-1} p(y_{i+1}|y_i) p(z_1) \prod_{i=1}^{t-1} p(z_{i+1}|z_i)} \\ &= \frac{C^{\text{mc}*}_{s,t}(\bar{x}_{s,t})}{p^{\text{mc}}_{s,t}(\bar{x}_{s,t})}. \end{aligned} \tag{18}$$

Therefore, the extended Monte Carlo path integration returns the same contribution as the original Monte Carlo path integration if we use $p(y_{s+1}|y_s) = K(y_{s+1}, z_t)$ and the radiance distribution is locally constant around z_t. Note that our path space extension defines exactly the same contribution and the probability density for density estimation.

Even though virtual perturbations are approximations, as long as vertex perturbation is considered in the weights computation (which in turn uses the probability densities), the resulting estimator remains both consistent and scale invariant.

Evaluation of Probability Densities: Given a path of length M, we have the following four cases which we evaluate probability densities under different techniques:

1. The path has $M + 1$ vertices via path integration, and we evaluate a probability density under extended path integration.

   We make the corresponding bidirectional connection, add a vertex with vertex perturbation, and evaluate the probability density in our unified definition.

2. The path has $M + 1$ vertices via path integration, and we evaluate a probability density for under density estimation.

   We first add a vertex with vertex perturbation at the location of density estimation, and evaluate the probability density in our unified definition.

3. The path has $M + 2$ vertices via density estimation, and we evaluate a probability density under extended path integration.

4. The path has $M + 2$ vertices via density estimation, and we evaluate a probability density under density estimation.

   In these two cases, we consider the corresponding path with extended path integration. Since the corresponding path has $M + 1$, the rest is the same as the case 1 (= case 3) and the case 2 (= case 4).

Invalid Sampling Techniques: Given a path length M and the kernel K, Equation 13 defines $M + 3$ techniques for both extended path integration and density estimation. However, not all techniques are valid. For the following invalid techniques, we simply set their probability densities and the corresponding weight to zero:

- $s = 0$ or $t = 0$ for density estimation, since density estimation needs at least one vertex for each subpath.
- $t = 0$ for extended path integration, since vertex perturbation needs at least one vertex for an eye subpath.

We thus have $M + 2$ valid techniques for path integration and $M + 1$ valid techniques for density estimation.

4 Implementation

Figure 3 shows pseudocode of our implementation. We implemented paths reuse by sampling and storing the sets of eye and light subpaths into the EyePaths and LightPaths buffers. The number of samples N in each set of subpaths is set to be equal to the number of pixels in the image NumPixels. Our theory does not require us to do so, however, we have found that this number of subpaths makes the implementation compatible with a typical implementation of bidirectional path tracing as we describe later.

The sampling procedures GENEYEPATH() and GENLIGHTPATH() are exactly the same as the ones for bidirectional path tracing. We then build a photon map over all the light vertices by BUILDPM() for efficient range queries. In our implementation, we used a spatial grid as an acceleration data structure, but it is possible to use different data structures such as a kD-tree.

```
procedure RENDERING(Scene, Camera, Image, N_itr)
  for all Pixels(i, j)
    do { EyePaths(i, j) ← GENEYEPATH(Scene, Camera, $i, j$)
       { LightPaths(i, j) ← GENLIGHTPATH(Scene)
  BUILDPM(LightPaths)
  DERadius ← CALCRADIUS($\alpha$, N_itr)
  for all Pixels(i, j)
    do COMBINEPATHS(EyePaths, LightPaths, $i, j$)

procedure COMBINEPATHS(EyePaths, LightPaths, $i, j$)
  $C_{\text{eye}}$ ← CONNECTEYE(EyePaths(i, j), LightPaths(i, j))
  CONNECTLT(LightImage, EyePaths(i, j).V[0], LightPaths(i, j))
  $C_{\text{de}}$ ← CONNECTDE(EyePaths(i, j), LightPaths)
  EyeImage(i, j) ← EyeImage(i, j) + $C_{\text{eye}}$ + $C_{\text{de}}$/NumPixels
```

Figure 3: *Pseudocode for our framework.*

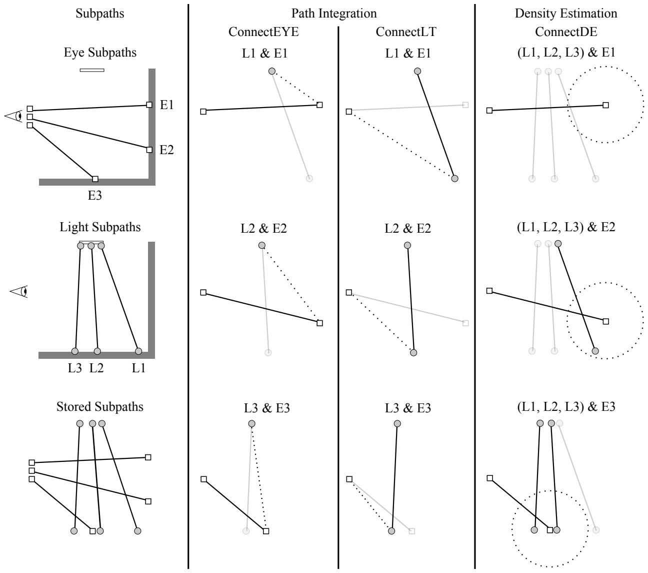


Figure 4: *Connection procedures for paths of length two. We first sample sets of subpaths from light sources and the eye (left). Sampling techniques for Monte Carlo path integration connect each eye subpath with one light subpath (center), whereas sampling techniques for photon density estimation connect each subpath with all the light subpaths (right). Contributions of complete paths are weighted according to a multiple importance sampling strategy.* $L1, L2, L3$ *and* $E1, E2, E3$ *are labels of the paths.*

4.1 Subpath Connections

We have three connection procedures that need to be implemented separately. Each procedure generates samples from a different sampling technique by reusing sampled sets of subpaths in EyePaths and LightPaths. Figure 4 illustrates our connection procedures for paths of length two.

Bidirectional path tracing techniques with $t \neq 1$:
CONNECTEYE() connects the given eye path using at least two vertices and the light path by tracing shadow rays between vertices.

This is the same procedure as bidirectional path tracing, except that the weight computation is extended to include photon density estimation. We also optimized this part by not performing vertex perturbation in the sampling procedure.

Bidirectional path tracing techniques with $t = 1$:
CONNECTLT() connects the first eye vertex and all the vertices in the light path via shadow rays.

This procedure is separated CONNECTEYE() since it accumulates contributions to a separate image buffer (LightImage). This separation is also used as the light tracing part of bidirectional path tracing [Veach 1998].

Photon density estimation techniques:
CONNECTDE() connects the given eye path and *all* the light paths by range queries. This procedure performs a range query at each eye vertex, not just only at the end of the eye path.

The contribution is divided by NumPixels, which is necessary for correctly taking into account the fact that we attempt to connect *all* the NumPixels light subpaths to each eye subpath. We also need to take this factor into account in the weight computation as we describe below.

4.2 Weight Computation

Figure 5 shows the computation of weights for each complete path. Each connection procedure internally calls BALANCEHEURISTIC() in order to properly weight the contribution of *each* sample. Inside this procedure, we call the evaluation procedure of the probability density function based on the definition in Section 2 and the strategy described in Section 3.

The boolean value given to PDF_UPS is true if we are considering the probability density function for extended path integration. This boolean value is used for switching how to handle the term $p(y_{s+1}|y_s)$ as we described in Section 2.2. NumPixels is multiplied to the probability density function of photon density estimation in order to properly account for the difference in the number of samples between bidirectional connections and density estimation connections.

```
procedure BALANCEHEURISTIC(Path, $N_E$, $N_L$)
  PDF_Sum ← 0
  for $s \leftarrow 0$ to Length(Path) + 2
  do
      $t \leftarrow$ Length(Path) $+ 2 - s$
      PDF_MC ← PDF_UPS(Path, $s$, $t$, true )
      PDF_DE ← PDF_UPS(Path, $s$, $t$, false ) $*$ NumPixels
      if ($t = N_E$ & $s = N_L$) PDF_Path ← PDF_MC
      if ($t = N_E$ & $(s + 1) = N_L$) PDF_Path ← PDF_DE
      PDF_Sum ← PDF_Sum + PDF_MC + PDF_DE
  return (PDF_Path/PDF_Sum)
```

Figure 5: *Pseudocode for weight computation.*

4.3 Compatibilities with Other Rendering Methods

Our framework subsumes implementations of multiple rendering methods. If one would like to use bidirectional path tracing, we just need to disable the connection by density estimation (CONNECTDE()) and also disable the corresponding probability density evaluation inside BALANCEHEURISTIC(). Likewise, our framework can be converted into (bidirectional) path tracing, light tracing, (progressive) photon mapping, and stochastic progressive photon mapping just by limiting a set of sampling techniques.

5 Results

We implemented bidirectional path tracing (BPT) [Veach and Guibas 1995], progressive photon mapping (PPM) [Hachisuka et al. 2008], stochastic progressive photon mapping (SPPM) [Hachisuka and Jensen 2009], and our unified path sampling (UPS) using the same rendering system. We will release an example implementation of our framework.

Our theoretical framework supports different radius per pixel, however, we chose to use a global radius for all the photon density estimation for simplicity and picked the initial radius by hand. The reference solution to Figure 8 was rendered by BPT and others were rendered by SPPM with manual classifications of specular/non-specular materials for glossy reflections.

We ran all the experiments on an Intel Core i7-2600 at 3.40 GHz with a single thread. The resolution of the images are either 512×512 or 640×480. We left the images intentionally unconverged to ease comparisons of computation errors. Table 2 summarizes the total average number of samples per pixel in our test cases. Overall, we have found that unified path sampling can take more samples than bidirectional path tracing by counting a complete path as one sample (e.g., a single BPT sample will result in multiple complete samples). This is because connections via photon density estimation are computationally less costly than connections via local path sampling in our implementation.

Figure 1 highlights the advantage of our method in a realistic illumination setting for interior design. We have modeled realistic lighting fixtures with emitters and reflectors. The dominant illumination is due to caustics as is the case in many lighting fixtures of the real world. Bidirectional path tracing, which is labeled as Monte Carlo path integration, is efficient at computing some contributions from glossy reflections, yet indirectly visible caustics exhibit significant amount of noise (e.g., caustics seen through water in the bathtub). Progressive photon mapping, which is labeled as photon density estimation, handles such caustics and reflections of caustics robustly, but a sharp BRDF lobe of the highly glossy material becomes a source of noise. Our unified framework combines the strength of each method under a single framework without any user intervention, and produces a more accurate solution in the same rendering time.

The graph in Figure 6 shows the convergence of the RMS (Root Mean Square) errors of the same scene with different methods. This graph uses the equal number of samples. This comparison is in favor of bidirectional path tracing in our implementation since Table 2 concludes that bidirectional path tracing is the most computationally costly method per sample. Even under such a comparison, the graph confirms that our method provides an order of magnitude more accurate solution than both methods for the same number of samples.

Figure 7 compares all of the rendering methods in our tests for another scene using the same rendering time. This scene also features highly glossy reflections, which are difficult to capture efficiently with photon density estimation, and indirectly visible caustics, which are difficult to capture efficiently with Monte Carlo path integration. This comparison includes SPPM that already demonstrated efficient rendering of glossy reflections by tracing one bounce ray from a visible point through each pixel [Hachisuka and Jensen 2009]. One issue of this approach is that whether we trace such rays or not is based on a heuristic classification of diffuse/non-diffuse materials. Our unified path sampling framework avoids introducing such a heuristic and combines all the possible techniques with a provably good strategy. Note also that diffuse direct illumination is significantly less noisy with our unified path sampling.

Figure 8 shows another equal-time comparison with bidirectional path tracing for a scene that has only diffuse materials. This scene does not feature any light transport that is particularly challenging for bidirectional path tracing. Even in such a scene configuration, our unified path sampling is still comparable to bidirectional path tracing since our framework subsumes bidirectional path tracing.

We emphasize that photon density estimation is important in many real-world scenarios. Figure 9 highlights such a case, where we have two light sources; a blue diffuse area light source, and a yellow diffuse area light source enclosed by a metal tube and a lens. The only difference between these two light sources is whether they are modeled after a realistic lighting fixture or a bare emitter. The blue light source directly illuminates the scene, while the yellow light source illuminates the scene via caustics just like many lighting fixtures in the real world. Our unified path sampling algorithm puts higher weight for Monte Carlo path integration techniques for illumination from the blue light source and photon density estimation techniques for illumination from the yellow light source.

Figure 10 visualizes relative contributions from each set of sampling techniques. Note that density estimation has relatively large contribution in this scene configuration. This is a provably good combination predicted by our theory, and photon density estimation indeed captures a significant portion of overall illumination.

Figure 11 shows sequences of rendered images of a simple scene where we have a Cornell box with a glossy box and a glass box with a small diffuse area light source. Despite its relatively simple configuration, bidirectional path tracing and progressive photon mapping already show their inefficiency at capturing certain light paths. The result shows the advantage of our unified framework even in this simple scene. Since our method captures all the features equally well, it is also possible to quickly identify overall illumination in the scene only after a few samples.

Scene	BPT	PPM	SPPM	UPS	Time [min]
Bathroom	1396	9494	5313	2085	240
Buddha	38	205	112	72	10
Conference	120	799	475	237	30
Cornell	264	1838	679	484	32
Torus	448	2438	1508	738	60
Treasures	367	1386	1184	618	120

Table 2: *Statistics of our experiments. The numbers in the column of each method (BPT: Bidirectional Path Tracing, PPM: Progressive Photon Mapping, SPPM: Stochastic Progressive Photon Mapping, and UPS: Unified Path Sampling) show the average numbers of samples per pixel. The average numbers of samples per pixel per minute over our test cases are 5.33 (BPT), 32.51 (PPM), 17.41 (SPPM), and 9.71 (UPS).*

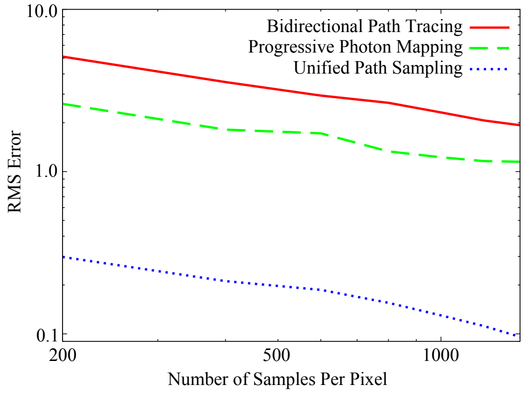
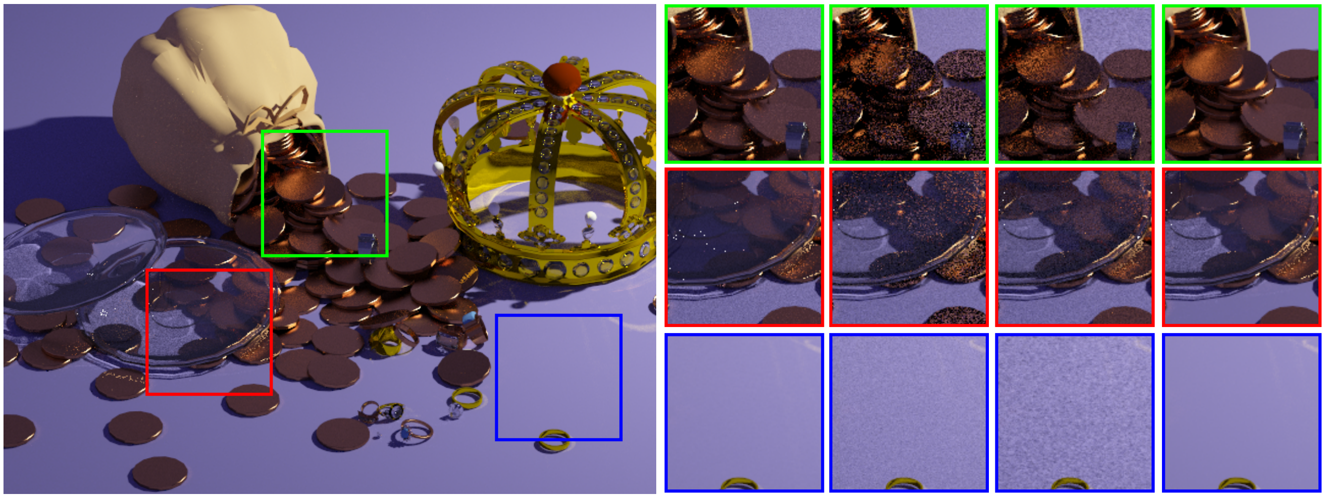


Figure 6: *RMS errors of the bathroom scene using the same average number of samples per pixel. We used equal number of samples as a comparison in this graph, favoring bidirectional path tracing as each sample of bidirectional path tracing takes more computation time in our implementation.*

6 Discussion

A number of researchers have explored applications of multiple importance sampling in the context of regular (not progressive) photon mapping. Bekaert et al. proposed a combination of the regular photon density estimation using multiple importance sampling in the context of their modified photon density estimator [2003]. Due to their connection kernel formulation, their framework cannot handle caustics from specular materials; our method efficiently handles those. Vorba and Křivánek [2011] described how multiple importance sampling can be used to combine only photon density estimation techniques. Unlike their approach, our method does not limit the combinations only to photon density estimation, but provides full combinations of Monte Carlo path integration and photon density estimation.

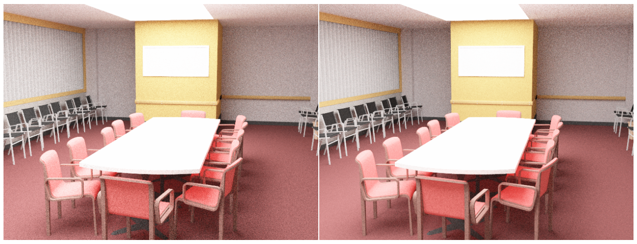
One concurrent work is vertex merging by Georgiev et al. [2011] which is an extension of SPPM based on multiple importance sampling. In the follow up work, Georgiev et al. [2012] reformulated BPT and (S)PPM as two families of sampling techniques via vertex



Rendered Image with UPS

Method	BPT	PPM	SPPM	UPS
RMSE	0.05003	0.05934	0.03324	0.02285

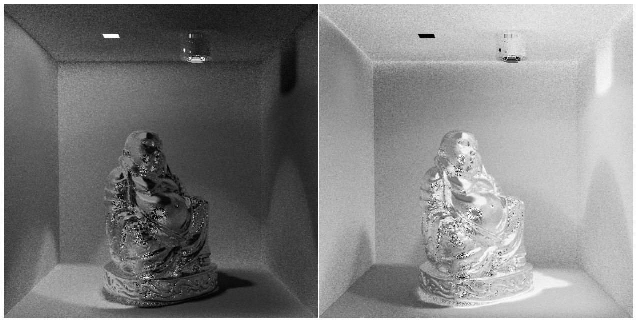
Figure 7: *Scene features high geometric complexity and illumination complexity. The scene has glass plates and coins and a crown with glossy metal illuminated by a small diffuse light source. The image on the left is rendered by our framework (UPS). The close-ups show parts of the images rendered by various methods using the same rendering time (120 min). Bidirectional path tracing (BPT) cannot efficiently render caustics seeing through glass, while glossy reflections are relatively less noisy. Progressive photon mapping (PPM) captures such indirectly visible caustics, but produces noisy results for glossy reflections. Stochastic progressive photon mapping (SPPM) captures all the illumination features reasonably well, but direct illumination is relatively noisy. Our framework (UPS) takes the best of all three approaches and captures all the illumination features efficiently.*



Bidirectional path tracing
RMS Error: 0.02152

Unified path sampling
RMS Error: 0.02255

Figure 8: *Conference room with diffuse surfaces and a large diffuse light sources. For this type of scenes, our method (unified path sampling) performs almost as well as bidirectional path tracing since the contribution of Monte Carlo path integration automatically dominates the final image.*



MC Path Integration

Density Estimation

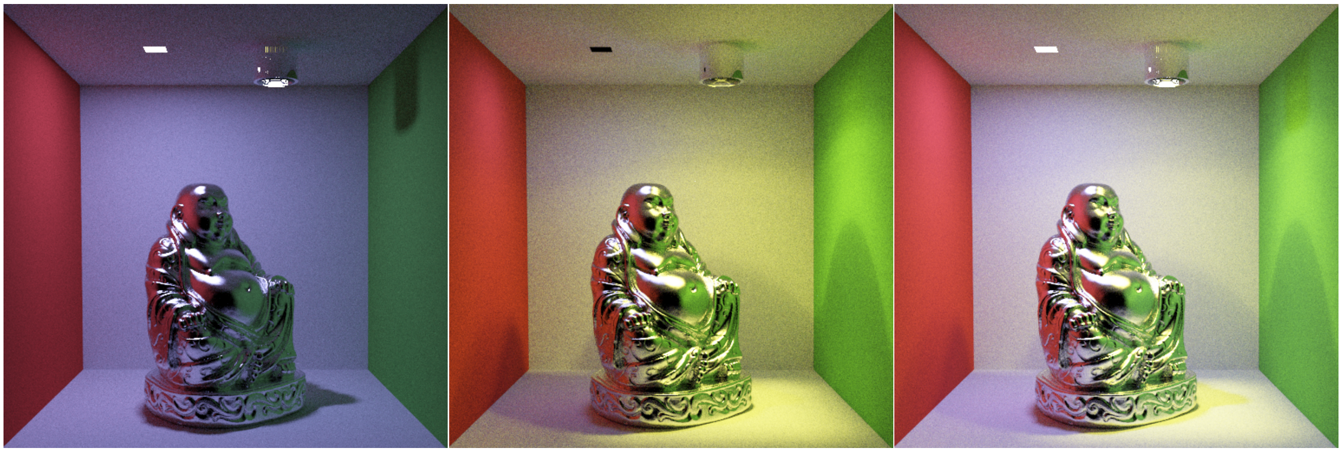
Figure 10: *Visualization of weight for each set of sampling techniques in Figure 9. The images show the ratio of summed weights of all the paths from each set.*

connections by shadow rays and vertex merging. This reformulation in fact results in the same algorithm as ours. One can think of their formulation as an alternative to our unified path space extension which uses vertex merging as a mean of *contraction* of the path space. Our theoretical contributions however are not exactly overlapping. In terms of theoretical contributions, we focus on analysis of MIS for a general combination of unbiased estimators and (not necessarily progressive) biased estimators, whereas their analysis provides more details on the asymptotic behavior of the combination of BPT and (S)PPM.

6.1 Limitations

The form of combinations that we explored in this paper is a weighted sum of different sampling techniques. However, this form is not the only way to combine different sampling techniques, and it is possible that we have a more efficient form of combinations. For example, it may be possible to develop an entirely new sampling technique based on our extended formulation. Although the results demonstrate that our combined technique works better than using one of the techniques alone, we certainly do not claim that our combination of path integration and density estimation is optimal.

We also emphasize that our goal in this paper is not improving efficiency of each individual sampling technique, but finding a better combination by introducing a new set of sampling techniques. This separation also means that our algorithm can still be inefficient if neither Monte Carlo path integration nor photon density estimation performs well for a given path. The rendered images within the red boxes in Figure 7 indeed reveal one such example. Glossy reflections of caustics due to the glass plate on the coins are noisy in all the images including the image with our method. Such paths of light are fundamentally difficult to sample efficiently even with our unified framework. One potential solution is to investigate incorporating advanced sampling methods such as Markov Chain Monte Carlo sampling [Veach and Guibas 1997; Hachisuka and Jensen 2011; Jakob and Marschner 2012] into our framework.



Weighted Contribution of MC Path Integration

Weighted Contribution of Density Estimation

Combined Result

Figure 9: *Weighted contribution of Monte Carlo path integration and photon density estimation within our unified path sampling framework. The scene has a Buddha statue with glossy reflections, and two small light sources with different colors (blue and yellow) where the yellow one is enclosed by a metal tube and a lens. The images show computed illumination due to Monte Carlo path integration (left, equivalent to techniques used in bidirectional path tracing), photon density estimation (center, equivalent to techniques used in progressive photon mapping), and the combined result (right). Each approach covers a different component of illumination based on our unified definition of probability density functions of paths.*

We would also like to point out that implementing our framework can be challenging as it subsumes both bidirectional path tracing and photon density estimation. This means that the engineering effort of implementing our framework is at least equivalent to the engineering effort of implementing those two approaches in total. Likewise, an efficient parallel implementation of our framework might be challenging since this was at least the case for bidirectional path tracing [van Antwerpen 2011].

7 Conclusion

We have presented a new sampling framework for light transport algorithms that combines unbiased Monte Carlo path integration and photon density estimation based on multiple importance sampling. The key idea is to extend the space of Monte Carlo path integration by introducing perturbation of path vertices. This extension provides a unified view of the sampling spaces, and serves as a theoretical foundation for the application of multiple importance sampling to this combination of two different light transport simulation approaches. We have demonstrated the improved robustness and efficiency of the resulting algorithm in comparison to bidirectional path tracing and progressive photon mapping.

We believe that our unified path sampling framework will find many practical applications for photorealistic image synthesis, and also lead to further development of robust light transport simulation methods to handle all kinds of illumination. Future work includes a combination with advanced sampling techniques such as Markov chain Monte Carlo sampling.

Acknowledgements

We would like to thank Youichi Kimura (Studio Azurite) for providing us the bathroom model. The scene in Figure is a modification of "Challenge #19: The King's Treasure" at `www.3drender.com` modeled by Dan Konieczka and Juan Carlos Silva. We thank Jaakko Lehtinen for his feedback on the initial draft. Leonhard Grüenschloß also gave us inspiring comments on the initial development of this project.

References

Bekaert, P., Slussalek, P., Havran, R. C. V., and Seidel, H. 2003. A custom designed density estimator for light transport. Tech. rep., Max-Planck-Institut für Informatik.

Cook, R., Porter, T., and Carpenter, L. 1984. Distributed ray tracing. *Computer Graphics (Proceedings of SIGGRAPH 84)* (July), 137–145.

Georgiev, I., Křivánek, J., and Slusallek, P. 2011. Bidirectional light transport with vertex merging. In *ACM SIGGRAPH Asia 2011 Sketches*, 27:1–27:2.

Georgiev, I., Křivánek, J., Davidovič, T., and Slusallek, P. 2012. Light transport simulation with vertex connection and merging. In *ACM SIGGRAPH Asia 2012 Papers*, XXX:1–XXX:10.

Hachisuka, T., and Jensen, H. W. 2009. Stochastic progressive photon mapping. In *ACM SIGGRAPH Asia 2009 Papers*, 141:1–141:8.

Hachisuka, T., and Jensen, H. W. 2011. Robust adaptive photon tracing using photon path visibility. *ACM Transaction on Graphics 30* (October), 114:1–114:11.

Hachisuka, T., Ogaki, S., and Jensen, H. W. 2008. Progressive photon mapping. In *ACM SIGGRAPH Asia 2008 Papers*, 130:1–130:8.

Jakob, W., and Marschner, S. 2012. Manifold exploration: a Markov chain Monte Carlo technique for rendering scenes with difficult specular transport. In *ACM SIGGRAPH 2012 Papers*, 58:1–58:13.

Jensen, H. 1996. Global illumination using photon maps. In *Rendering Techniques '96*, 21–30.

Kajiya, J. T. 1986. The rendering equation. In *ACM SIGGRAPH 1986 Papers*, vol. 20, 143–150.

Knaus, C., and Zwicker, M. 2011. Progressive photon mapping: A probabilistic approach. *ACM Transaction on Graphics 30* (May), 25:1–25:13.

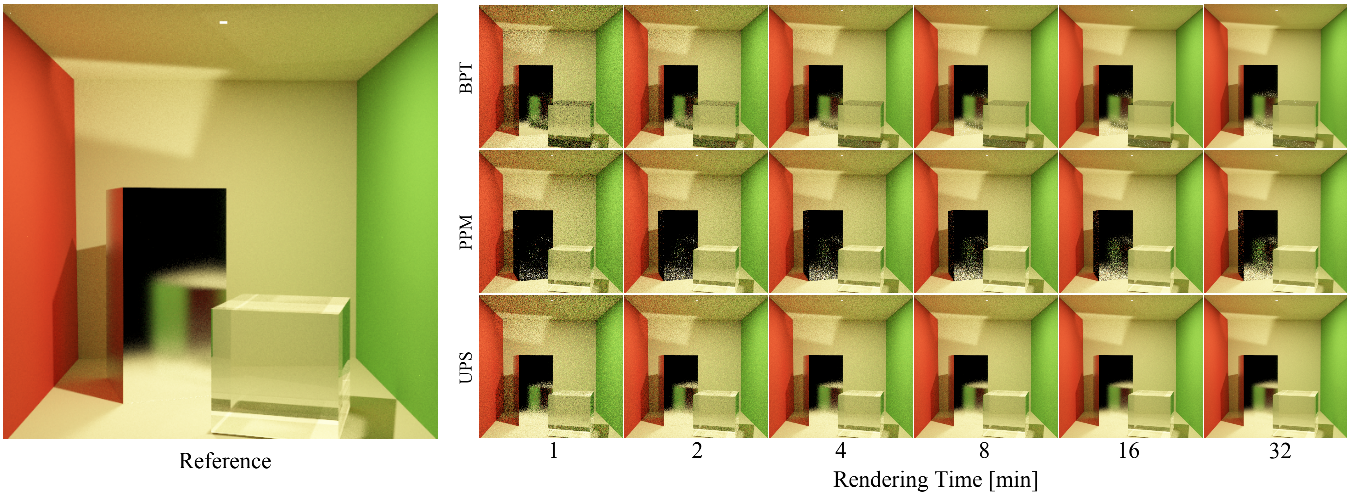


Figure 11: *Sequences of rendered images using different approaches. Even this simple scene reveals issues of using previous approach alone. Bidirectional path tracing (BPT) is inefficient for sampling caustics seen through the glass cube, while progressive photon mapping (PPM) is inefficient for sampling highly glossy reflections. Although these two approaches theoretically guarantee convergence to the correct solution in such cases, the sequences of images show slow convergence in practice. Our framework (UPS) unifies both approaches into a single unified path sampling method and significantly improves convergence speed for such inefficient cases. RMS errors at 32 min are 0.09146 (BPT), 0.08874 (PPM), and 0.01536 (UPS) respectively.*

LAFORTUNE, E. P., AND WILLEMS, Y. D. 1993. Bidirectional path tracing. In *Proceedings of Computer Graphics*, 95–104.

SILVERMAN, B. W. 1986. *Density estimation for statistics and data analysis*. Chapman and Hall.

TOKUYOSHI, Y. 2009. Photon density estimation using multiple importance sampling. In *ACM SIGGRAPH Asia 2009 Posters*, 37:1–37:1.

VAN ANTWERPEN, D. 2011. Improving SIMD efficiency for parallel monte carlo light transport on the GPU. In *ACM SIGGRAPH Symposium on High Performance Graphics*, 41–50.

VEACH, E., AND GUIBAS, L. 1994. Bidirectional estimators for light transport. In *Proceedings of the Fifth Eurographics Workshop on Rendering*, 147–162.

VEACH, E., AND GUIBAS, L. J. 1995. Optimally combining sampling techniques for Monte Carlo rendering. In *ACM SIGGRAPH 1995 Papers*, 419–428.

VEACH, E., AND GUIBAS, L. J. 1997. Metropolis light transport. In *ACM SIGGRAPH 1997 Papers*, 65–76.

VEACH, E. 1998. *Robust Monte Carlo methods for light transport simulation*. PhD thesis, Stanford University, Stanford, CA, USA. AAI9837162.

VORBA, J., AND KŘIVÁNEK, J. 2011. Bidirectional photon mapping. In *Proceedings of CESCG 2011*, The 15th Central European Seminar on Computer Graphics.