On-line Learning of Parametric Mixture Models for Light Transport Simulation

Jiří Vorba[1]* Ondřej Karlík[1]* Martin Šik[1]* Tobias Ritschel[2]† Jaroslav Křivánek[1]‡

[1]Charles University in Prague [2]MPI Informatik, Saarbrücken

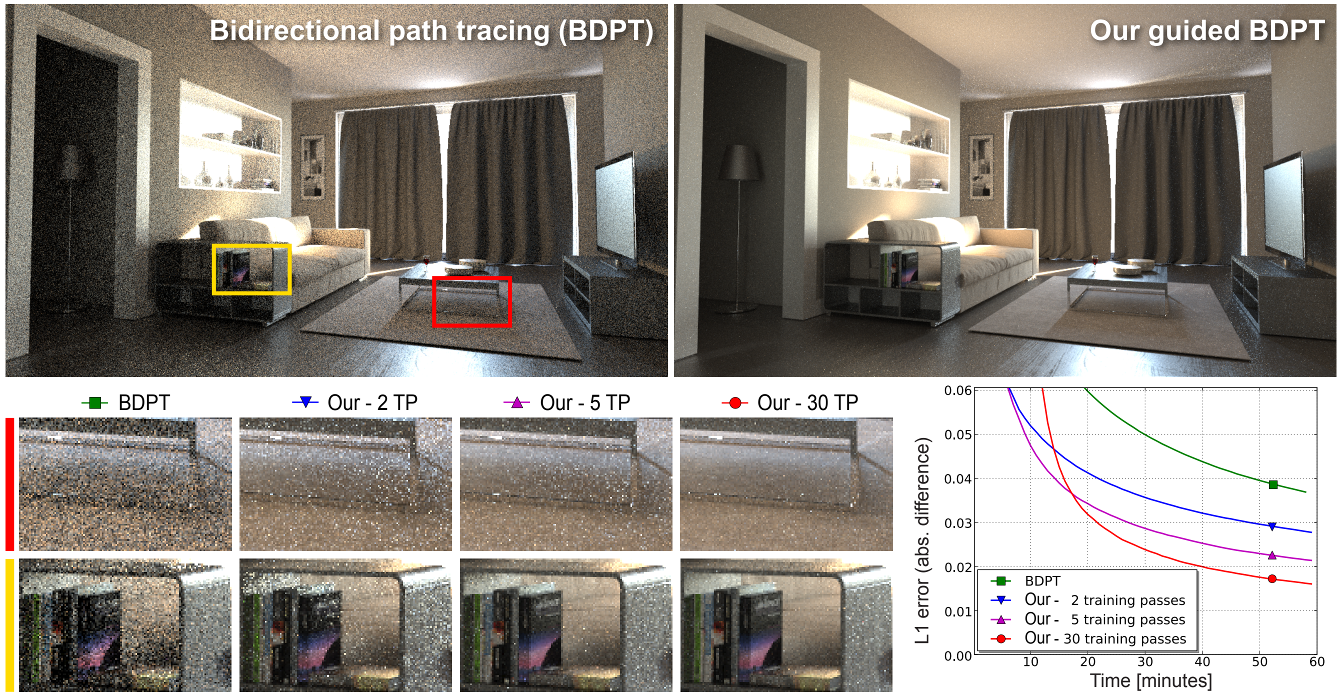

Figure 1: *We render a scene featuring difficult visibility with bidirectional path tracing (BDPT) guided by our parametric distributions learned on-line in a number of training passes (TP). The insets show equal-time (1h) comparisons of images obtained with different numbers of training passes. The results reveal that the time spent on additional training passes is quickly amortized by the superior performance of the subsequent guided rendering.*

Abstract

Monte Carlo techniques for light transport simulation rely on importance sampling when constructing light transport paths. Previous work has shown that suitable sampling distributions can be recovered from particles distributed in the scene prior to rendering. We propose to represent the distributions by a parametric mixture model trained in an on-line (i.e. progressive) manner from a potentially infinite stream of particles. This enables recovering good sampling distributions in scenes with complex lighting, where the necessary number of particles may exceed available memory. Using these distributions for sampling scattering directions and light emission significantly improves the performance of state-of-the-art light transport simulation algorithms when dealing with complex lighting.

CR Categories: I.3.3 [Computer Graphics]: Three-Dimensional Graphics and Realism—Display Algorithms

Keywords: light transport simulation, importance sampling, parametric density estimation, on-line expectation maximization

Links: DL PDF WEB CODE

1 Introduction

Despite recent advances, robust and efficient light transport simulation is still a challenging open issue. Numerous algorithms have been proposed to solve the problem, but certain common lighting conditions, such as highly occluded scenes, remain difficult. Most existing unidirectional and bidirectional methods rely on incremental, local construction of transport sub-paths, which is oblivious to the global distribution of radiance or importance. As a result, the probability of obtaining a non-zero contribution upon sub-path connection in highly occluded scenes is low. This is the main reason why such scenes remain difficult to render. While Metropolis light transport and related methods [Veach and Guibas 1997; Kelemen et al. 2002; Cline et al. 2005] strive for importance sampling on the entire path space, they suffer from sample correlation and are often outperformed by the classic Monte Carlo approaches.

*{jirka,karlik,martin.sik}@cgg.mff.cuni.cz
†ritschel@mpi-inf.mpg.de
‡jaroslav.krivanek@mff.cuni.cz

To address the problem of rendering highly occluded scenes, we augment the sampling of local scattering directions and light emission, employed when constructing transport sub-paths, with global information. We sample the scattering and emission proportionally to an approximation of the equilibrium radiance (for camera sub-paths) or importance (for light sub-paths). As a result, the sub-paths are guided to each other – camera sub-paths toward light sources and light sub-paths toward the camera – which increases the probability of constructing full paths with non-zero contributions. This, in turn, significantly reduces variance without introducing bias.

Our work adopts the idea of reconstructing the sampling distributions from particles [Jensen 1995]. While a number of works have taken this route, they often use inflexible representations of the distributions [Jensen 1995] and incur significant overhead [Hey and Purgathofer 2002]. Most importantly, these methods rely on a limited number of particles, which is usually insufficient to recover useful sampling distributions in highly occluded scenes.

We propose to represent the sampling distributions with the Gaussian mixture model (GMM), extensively used in machine learning [Bishop 2006]. The GMM is efficient to learn, easy to evaluate and sample, and compact to store. The core of our method is an on-line (progressive) learning step: Instead of learning the distributions only once from a limited set of particles, we continuously train them using a potentially infinite stream of particles while keeping a bounded memory footprint. We use importance sampling based on this model in a number of light transport algorithms, including the state-of-the-art bidirectional ones [Veach 1997; Georgiev et al. 2012b; Hachisuka et al. 2012]. Our key contributions are:

- introduction of on-line learning of parametric mixture models to image synthesis,
- a learning procedure that can handle particles with highly varying weights,
- importance-driven particle emission from environment light sources, and
- an efficiency improvement of path sampling-based light transport algorithms in complex, highly occluded scenes.

2 Related Work

Sampling distributions from particles. Jensen [1995] proposed the use of light-carrying particles, or photons, to guide direction sampling in a path tracer. He reconstructs the directional PDFs by counting the number of photons whose directions fall into constant-sized bins. This corresponds to simple histogram density estimation, which is known to be a poor density estimation method prone to under- or over-fitting [Bishop 2006]. The method could be made progressive by accumulating an infinite stream of particles to the bins. However, this progressivity is rather deceiving because the regular histogram grid does not adapt to details, thereby producing poor PDF reconstructions no matter how many particles are used. A similar approach was taken by Steinhurst and Lastra [2006] and Budge et al. [2008]. Peter and Pietrek [1998] extended this idea to using importance particles for guiding photons towards the camera. Hey and Purgathofer [2002] represent the directional PDFs with cones of adaptive width centered on gathered photons' directions. This method yields better results than previous work, but incurs substantial overhead. Pharr and Humphreys [2010] implement a simplified version of Hey and Purgathofer's method with cones of constant width. Its reduced overhead, however, comes at the expense of quality. Our paper follows this general line of work, pointing out that 'reconstructing sampling PDFs from photons' is a general density estimation problem. We adopt parametric mixture model estimation to learn the sampling PDFs in an on-line (progressive) manner.

Adaptive sampling. A number of works propose adaptive construction of sampling distributions during rendering. Lafortune and Willems [1995] store radiance samples in a 5D tree and use this information for importance sampling and as a control variate. Pegoraro et al. [2008a; 2008b] replace the 5D tree with per-pixel directional distributions. Dutré and Willems [1994; 1995] use adaptive sampling similar to the VEGAS algorithm [Lepage 1978] to emit paths from light sources. Cline et al. [2008] use fixed-size adaptive importance sampling tables in a path tracer. A common problem shared by these works is a fixed-resolution or hierarchical representation of the sampling function which makes it difficult to capture high-frequency features. The Gaussian mixture model that we adopt does not suffer from this issue.

Caching. Bashford-Rogers et al. [2012] employ a parametric model for importance sampling, where cosine lobes are used to model directional distributions, cached and re-used across pixels. The idea of caching sampling distributions was also used by Georgiev et al. [2012a], who precompute and cache discrete distributions of the contributions of virtual point lights (VPLs) to scene points. As opposed to these techniques, we store the learned directional distributions in a spatial cache not only to amortize the overhead, but mainly to maintain a persistent representation of the distributions, thereby enabling their progressive refinement through on-line learning. Other solutions such as irradiance and radiance caching [Ward et al. 1988; Křivánek et al. 2005] can be used when systematic error is acceptable; however, we pursue unbiased results.

Sampling emission from environment maps. Tsai et al. [2008] employ spherical Gaussians to sample from the product of environment illumination and BRDFs. Bashford-Rogers et al. [2013] guide emission from environment maps using importance. We address a more general problem of sampling both indirect illumination and environment emission.

Progressive GMM learning in rendering. Jakob et al. [2011] use a Gaussian mixture model to represent spatial distribution of scalar irradiance in participating media. The accelerated expectation maximization (EM) algorithm [Verbeek et al. 2006] used for this purpose allows progressive model updates, but it fundamentally relies on the ability to produce a good fit from the initial batch of particles. Thanks to the maximum a posteriori (MAP) formulation of the model estimation, our on-line technique robustly handles situations where particles are extremely scarce.

3 Background

In this section, we review Monte Carlo integration with importance sampling and learning of parametric mixture models.

3.1 Rendering Equation and Importance Sampling

Light transport simulation involves finding values of the equilibrium outgoing radiance L_o that satisfy the rendering equation:

$$L_\mathrm{o}(\mathbf{x},\omega_\mathrm{o}) = L_\mathrm{e}(\mathbf{x},\omega_\mathrm{o}) + \int_{\mathcal{H}^+} L_\mathrm{in}(\mathbf{x},\omega_\mathrm{i}) f_\mathrm{r}(\mathbf{x},\omega_\mathrm{i},\omega_\mathrm{o}) \cos\theta_\mathrm{i} \mathrm{d}\omega_\mathrm{i}.$$

Here $L_\mathrm{e}(\mathbf{x},\omega_\mathrm{o})$ is the radiance emitted from a point $\mathbf{x}$ in a direction ω_o, $L_\mathrm{in}(\mathbf{x},\omega_\mathrm{i})$ is the incoming radiance from ω_i, f_r denotes the BRDF, and θ_i is the angle between the surface normal at $\mathbf{x}$ and the incoming light direction ω_i [Dutré et al. 2006]. Monte Carlo methods for solving the rendering equation involve *sampling rays in random directions* ω_i over the hemisphere $\mathcal{H}^+$ to recursively estimate the integral in the equation. To reduce the Monte Carlo estimator's variance, it is crucial to employ an importance sampling strategy that draws the direction samples from a distribution closely proportional to the integrand. The traditional BRDF importance sampling can be ineffective when the incoming radiance term $L_{\mathrm{in}}(\mathbf{x}, \omega_{\mathrm{i}})$ is the primary source of the integrand's variation. Common examples include caustics, indirect highlights, and complex visibility. It is therefore beneficial to enhance the importance sampling with an estimate of the directional distribution of incoming radiance. One way to achieve this is to obtain the distribution from a 'photon map' generated by particle tracing before the rendering starts [Jensen 1995]. We follow this general strategy and note that the problem can be viewed as density estimation in the directional domain. We advocate the use of a parametric mixture model to represent the distributions, which enables on-line learning from a potentially infinite stream of particles. Additionally, we build distributions of camera importance for guiding paths starting from the light sources.

3.2 Parametric Mixture Models and the EM Algorithm

We now review the classic *batch EM* (Expectation Maximization) algorithm, which is well known to the graphics community. We then present the *off-line stepwise EM* algorithm, a generalization of batch EM with better convergence properties [Liang and Klein 2009]. The stepwise EM formulation allows deriving the *on-line stepwise EM* algorithm for learning from a potentially infinite stream of particles. While batch EM is reviewed only for reference, both stepwise EM variants are essential components of our method. Bishop [2006] and Cappé [2011] provide more details on EM.

Parametric mixture models. A parametric finite mixture model is a convex combination of simpler parametric distributions. We use the Gaussian mixture model (GMM) with K components:

$$\mathrm{GMM}(\mathbf{s}|\theta) = \sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{s}|\mu_j, \mathbf{\Sigma}_j), \tag{1}$$

where $\mathcal{N}(\mathbf{s}|\mu_j, \mathbf{\Sigma}_j)$ is a Gaussian distribution over $\mathbf{s} \in \mathbb{R}^d$ with a mean μ_j and a covariance matrix $\mathbf{\Sigma}_j$. The *mixing coefficients* π_j satisfy $\pi_j \geq 0$ and $\sum_{j=1}^{K} \pi_j = 1$. The mixture is defined by a parameter vector $\theta = \{\pi_1, \mu_1, \mathbf{\Sigma}_1, \ldots, \pi_K, \mu_K, \mathbf{\Sigma}_K\}$.

Maximum likelihood (ML) estimation. The density estimation problem for a mixture model $p(\mathbf{s}|\theta)$ (e.g. the GMM), consists in finding parameters θ so that $p(\mathbf{s}|\theta)$ is a good approximation of the unknown distribution that generated a given finite set of N observed samples $\mathbf{S} = \{\mathbf{s}_0, \ldots, \mathbf{s}_{N-1} \in \mathbb{R}^d\}$. A common approach is to use the parameter vector θ^{ML} that maximizes the *log-likelihood* $\mathcal{L}(\mathbf{S}, \theta) = \ln p(\mathbf{S}|\theta) = \sum_{q=0}^{N-1} \ln p(\mathbf{s}_q|\theta)$.

Maximum a-posteriori (MAP) estimation. A fundamental problem with ML estimation is over-fitting, i.e. introducing patterns not present in the original distribution [Bishop 2006]. This issue is particularly pressing in our approach, where we may have only a few observed samples available to construct initial estimates. To deal with this issue, we adopt the *maximum a posteriori* solution θ^{MAP}, which seeks the mode of the *posterior* distribution $p(\theta|\mathbf{S})$ over model parameters θ, given by Bayes' theorem: $p(\theta|\mathbf{S}) \propto p(\mathbf{S}|\theta)p(\theta)$ (i.e. posterior $\propto$ likelihood $\times$ prior). With only a few samples, the solution is mostly determined by our *prior* beliefs (e.g. that PDFs with extreme values are unlikely), modeled by the prior distribution $p(\theta)$, which is overridden as more samples are observed. Gauvain and Lee [1994] provide more details.

Batch expectation maximization (Batch EM). Expectation maximization (EM) [Dempster et al. 1977] is an iterative procedure to find the ML or MAP estimates for mixture models. The classic, or *batch* EM algorithm [Liang and Klein 2009] for a finite set of observed samples starts with an initial guess of parameters and proceeds in iterations over the sample set. In each iteration, which consists of the *expectation* (E) and the *maximization* (M) steps, it obtains a new estimate θ^{new} based on the current estimate θ^{old}:

```
repeat
    // E-step: Eq. (3)
    $\mathbf{u}^j_{N-1}$ := COMPUTESUFFICIENTSTATS($\mathbf{S}, \theta^{\mathrm{old}}$)
    // M-step
    $\theta^{\mathrm{new}} := \overline{\theta}(\mathbf{u}^1_{N-1}, \ldots, \mathbf{u}^K_{N-1})$
until CONVERGED();
```

Since the log-likelihood $\mathcal{L}(\mathbf{S}, \theta)$ over the EM iterations is a non-decreasing function of θ, the iterative solution θ^{new} converges to a local maximum. The following condition is often used as a convergence criterion: $\left|\mathcal{L}(\mathbf{S}, \theta^{\mathrm{old}}) - \mathcal{L}(\mathbf{S}, \theta^{\mathrm{new}})\right| < \epsilon\, |\mathcal{L}(\mathbf{S}, \theta^{\mathrm{new}})|$.

In the **E-step**, the *responsibilities* γ_{qj} of every component j for each sample $\mathbf{s}_q$ are evaluated. Informally, they give the probability that the sample $\mathbf{s}_q$ would be drawn from the component j if we sampled from the mixture θ^{old}. For the GMM, the responsibilities are computed as

$$\gamma_{qj} = \frac{\pi_j \mathcal{N}(\mathbf{s}_q \mid \theta^{\mathrm{old}}_j)}{\sum_{h=1}^{K} \pi_h \mathcal{N}(\mathbf{s}_q \mid \theta^{\mathrm{old}}_h)}. \tag{2}$$

With these responsibilities, we can compute the *sufficient statistics* $\mathbf{u}^j_{N-1}$ for every mixture component j as the weighted average

$$\mathbf{u}^j_{N-1} = \frac{1}{N} \sum_{q=0}^{N-1} \gamma_{qj} \mathbf{u}(\mathbf{s}_q), \tag{3}$$

where $\mathbf{u}(\mathbf{s}_q) = (1, \mathbf{s}_q, \mathbf{s}_q\mathbf{s}_q^T)$ is a triplet consisting of the number 1, the vector $\mathbf{s}_q$, and the matrix $\mathbf{s}_q\mathbf{s}_q^T$. The subscript $N-1$ suggests that the sufficient statistics are based on N observed samples $\mathbf{s}_0, \ldots, \mathbf{s}_{N-1}$. The ML and MAP estimates depend on the observed samples in $\mathbf{S}$ only through these sufficient statistics.

In the **M-Step**, the sufficient statistics are used to obtain a new GMM estimate θ^{new} using a closed form update formula $\theta^{\mathrm{new}} = \overline{\theta}(\mathbf{u}^1_{N-1}, \ldots, \mathbf{u}^K_{N-1})$. Details are provided in Sec. 4.2.

Off-line stepwise EM. We now describe the stepwise EM formulation [Liang and Klein 2009], whose on-line variant forms the basis of our approach. In batch EM, the sufficient statistics are recomputed from all N samples (E-step) and only then, the distribution parameters can be updated (M-step). The stepwise formulation, on the other hand, continuously updates the statistics with every observed sample, which enables more frequent parameter updates, and therefore faster convergence.

Off-line stepwise EM, like batch EM, iterates over the sample set until convergence, as shown in Alg. 1. In the E-step, the sufficient statistics for each mixture component j are updated using the formula:

$$\mathbf{u}^j_i = (1 - \eta_i)\mathbf{u}^j_{i-1} + \eta_i\, \gamma_{qj} \mathbf{u}(\mathbf{s}_q), \tag{4}$$

where $i \geq 0$ is an index that increments with each processed sample and $q = i \mod N$ is the index of the sample $\mathbf{s}_q$ in the sample set $\mathbf{S}$. In other words, the samples from $\mathbf{S}$ are processed over and over, while the index i keeps growing. The updated statistics $\mathbf{u}^j_i$ are given by a weighted average of the statistics $\gamma_{qj}\mathbf{u}(\mathbf{s}_q)$ for the currently observed sample $\mathbf{s}_q$ and the statistics $\mathbf{u}^j_{i-1}$ for all the previously observed samples. The weight in this average is given by the decreasing sequence $\{\eta_i\}_{i \geq 1}$ of *stepsizes* that must obey $\sum_i \eta_i = \infty$ and $\sum_i \eta_i^2 < \infty$. A sequence which satisfies these conditions is $\eta_i = i^{-\alpha}$ with the effective values of the stepsize parameter $\alpha \in [0.6, 0.9]$ [Cappé 2011].

The distribution parameters θ^{old} are updated in the M-step after processing every m-th sample ($1 \leq m \leq N$). The formula is the same as in batch EM, with the current sufficient statistics $\mathbf{u}_i$ as inputs:

$$\theta^{\text{new}} = \overline{\theta}(\mathbf{u}_i^1, \ldots, \mathbf{u}_i^K). \tag{5}$$

Algorithm 1: Off-line stepwise EM

```
1  // Index of sufficient statistics
2  $i := 0$
3  repeat
4      // Iterate over a batch of N samples
5      for $q := 0$ to $N - 1$ do
6          // E-step: Eq. (4)
7          $\mathbf{u}_i^j :=$ UPDATESUFFICIENTSTATS($\mathbf{s}_q, \theta^{\text{old}}$)
8          if $i + 1 \bmod m = 0$ then
9              // M-step: every m-th observed sample; Eq. (5)
10             $\theta^{\text{new}} := \overline{\theta}(\mathbf{u}_i^1, \ldots, \mathbf{u}_i^K)$
11         end
12         $i := i + 1$
13     end
14 until CONVERGED();
```

On-line stepwise EM. The batch EM and the off-line stepwise EM algorithms base inference on a finite set of N samples stored in memory. Our method, however, targets scenarios where the number of samples (e.g. photons) necessary for reliable inference would be impractical or even impossible to store. The off-line stepwise EM formulation, unlike batch EM, can be easily modified for this purpose [Sato and Ishii 2000; Liang and Klein 2009; Cappé 2011]. The key is Equation (4) that enables progressive embedding of the information from any number of particles into a small set of statistics. If we consider the input set $\mathbf{S} = \{\mathbf{s}_0, \mathbf{s}_1, \ldots \in \mathbb{R}^d\}$ to be an infinite stream of samples, then the *on-line stepwise EM* algorithm is obtained from Alg. 1 by removing the outer cycle (lines 3 and 14) that iterates over the finite batch of samples. As such, the on-line algorithm continues learning as long as the samples are streamed. Details with respect to our application are given in Sec. 4.2.

4 Our Approach

We now present an overview of our unbiased guiding method (Sec. 4.1) followed by details of our technical contribution: MAP density estimation from weighted particles (Sec. 4.2), caching of distributions (Sec. 4.3), and emission from environment light sources (Sec. 4.4). In Sec. 4.5, we discuss the use of Russian roulette in our method.

4.1 Overview

Our method is split into two strictly separated phases: a) training of hemispherical distributions representing the incoming radiance or importance from particles (training phase), and b) using the trained distributions for importance sampling in rendering (rendering phase). During the training phase, the directional distributions are placed and cached at scene surfaces and progressively updated. The distributions stay fixed during the entire rendering phase.

Training phase. The training phase, depicted in Fig. 2, consists of several *training passes*. Each training pass comprises tracing a batch of importons from the camera followed by a batch of photons from the light sources. We start by tracing a batch of importons without our guiding (Fig. 2a). Every succeeding particle tracing step is then guided by our distributions, which are created and progressively refined throughout the entire training phase. Radiance distributions are trained by photons and used to guide paths from the camera. Conversely, importance distributions are trained by importons and used to guide paths from the light sources. *Guiding* refers to importance sampling of local scattering directions and emission from environment light sources based on our distributions, as described below.

New distributions are created on-the-fly during the guided particle tracing steps (Fig. 2c, f). They are stored in a *spatial cache* so that they can be reused at nearby locations and refined in the subsequent training passes. We keep two separate caches, one for importance and the other for radiance distributions. If the particle tracing process requests a distribution at a certain point and none is available at any nearby location, a new distribution is created and cached. We train the new distribution from the particle map constructed during the preceding particle tracing step (e.g. the importon map if we currently trace a photon) by collecting N nearest particles and using them as the input for *off-line stepwise EM*. As we use the method only for importance sampling of *indirect illumination*, we train the distributions using particles that have scattered at least once. The new distribution is then cached together with meta-information necessary for its refinement in the subsequent training passes (see Sec. 4.2).

After a batch of particles has been traced, we use it to update all distributions in the respective cache (Fig. 2d, e). For each cached distribution, we find the N nearest particles and use *on-line stepwise EM* to update it. At any given time, only the last two particle maps (one for radiance and the other for importance) are kept in memory. We delete the maps before constructing new ones (Fig. 2b, g), and thus our method keeps a *bounded memory footprint* while allowing for an arbitrary number of training passes. Unlike the particle maps, the two distribution caches are persistent and are continuously updated throughout the whole training phase.

The motivation behind our use of interleaved, mutually guided importon and photon tracing steps is that with every training pass, the distributions become more accurate and provide improved importance sampling for the subsequent training passes. This approach significantly improves the efficiency of the training phase.

Rendering phase. With the two caches of distributions obtained in the training phase, we can guide the construction of both camera and light sub-paths in virtually any path sampling-based light transport algorithm (including bidirectional ones). To guide a unidirectional algorithm, such as path tracing, we simply discard the

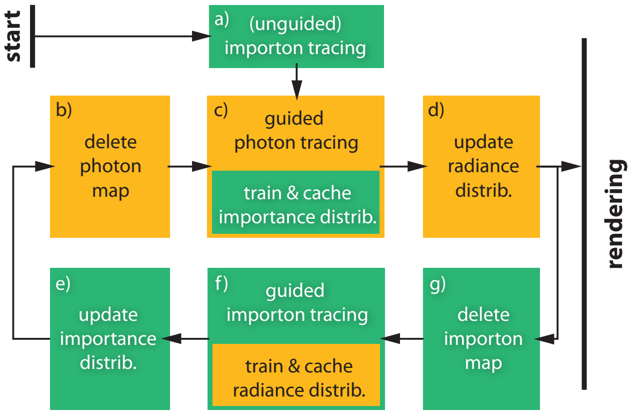
start
a) (unguided) importon tracing
b) delete photon map
c) guided photon tracing
train & cache importance distrib.
d) update radiance distrib.
rendering
e) update importance distrib.
f) guided importon tracing
train & cache radiance distrib.
g) delete importon map

Figure 2: *The training phase preceding the rendering phase. Processes related to importons or importance distributions use green background while processes related to photons or radiance distributions are in orange.*

unnecessary distribution cache. As in the training phase, we use the distributions for importance sampling of the local scattering directions and emission from environment light sources. If no guiding distribution is available at a nearby position, a new one is trained from the latest batch of particles.

Distribution representation. After investigating a number of alternatives, we decided to model the directional distribution $p(\omega|\mathbf{x})$ at a spatial location $\mathbf{x}$ with a mixture of bi-variate Gaussians $\mathrm{GMM}(\mathbf{s})$ on a 2D plane, i.e. $\mathbf{s} \in \mathbb{R}^2$. To do this, we project the hemisphere $\mathcal{H}^+$ onto a unit square using the area preserving mapping $\mathcal{S}$ of Shirley and Chiu [1997].

Sampling from the distribution model. To generate a new direction ω after a particle has collided with a surface at a position $\mathbf{x}$, we randomly choose between BRDF sampling and sampling from our *guiding distribution* $p(\omega|\mathbf{x})$. Both strategies are then combined using multiple importance sampling [Veach 1997]. To sample a direction ω' from our guiding distribution, we first draw a 2D position $\mathbf{s}'$ from the $\mathrm{GMM}(\mathbf{s})$ and then we apply the inverse mapping so that $\omega' = \mathcal{S}^{-1}(\mathbf{s}')$. To compensate for the change of variables, we multiply the PDF value by the Jacobian of $\mathcal{S}^{-1}$, which is a constant $\frac{1}{2\pi}$ for Shirley and Chiu's mapping. Should the sample $\mathbf{s}'$ lie outside of the unit square, the particle path is terminated.

4.2 Learning Distributions from Weighted Particles

A particle tracing algorithm generates particles $\mathbf{p}_i = (\mathbf{x}_i, \omega_i, w_i)$, defined by their position $\mathbf{x}_i$, incoming direction ω_i, and *weight* w_i (also referred to as 'flux' when tracing photons). The particle weight is a product of the particle emission function, BRDFs, and geometry factors divided by the probability density of generating the particle path [Pharr and Humphreys 2010]. The density of the photons or importons together with their weights form an unbiased representation of the equilibrium incoming radiance L_{in} or importance W_{in}, respectively [Veach 1997]. Therefore, an approximation of a sampling PDF $p(\omega|\mathbf{x}) \propto L_{\mathrm{in}}(\mathbf{x}, \omega)$ can be reconstructed from the *directions* and *weights* of photons $\mathbf{p}_i$ in the vicinity of $\mathbf{x}$. Similarly, directions and weights of importons in the vicinity of $\mathbf{x}$ can be used to reconstruct a sampling PDF $p(\omega|\mathbf{x}) \propto W_{\mathrm{in}}(\mathbf{x}, \omega)$.

Here, we present a new generalization of the stepwise EM algorithms (both off-line and on-line) that supports density estimation from a set of *weighted particles*. Before delving into its description, let us clarify, on an example of a single distribution, the use of off-line and on-line stepwise EM in our method (see Fig. 3). A distribution is initialized by *off-line* stepwise EM from the set of particles available in the first training pass. We then store its sufficient statistics $\mathbf{u}_i$ and its counter i so that the learning can be resumed in the subsequent training passes by *on-line* stepwise EM.

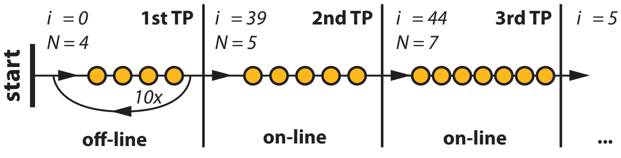

Figure 3: *An example of learning of a single distribution from a stream of particles processed first by the off-line and then by the on-line stepwise EM algorithms. In the first training pass (TP), we use the off-line algorithm that iterates until convergence (10 times in our example) over the $N = 4$ particles available in the first TP. In the subsequent training passes, the on-line algorithm is used, which processes each particle only once. The index i of the sufficient statistics associated with the trained distribution is incremented with each processed particle.*

Save for some subtle differences, our new weight-aware formulation affects both off-line and on-line stepwise EM in the same manner. Thus, in the following statements, we refer to both algorithms as stepwise EM unless the algorithm variant is explicitly stated.

Weighted data log-likelihood. The log-likelihood $\mathcal{L}(\mathbf{S}, \theta)$, as given in Sec. 3.2, does not allow density estimation from a weighted set of samples. Thus, we introduce the following *weighted data log-likelihood*

$$\mathcal{L}(\mathbf{S}, \mathbf{w}, \theta) = \sum_{q=0}^{N-1} w_q \ln p(\mathbf{s}_q|\theta), \tag{6}$$

where $\mathbf{S} = \{\mathbf{s}_0, \ldots, \mathbf{s}_{N-1}\}$ is a set of N observed samples with their corresponding weights $\mathbf{w} = \{w_0, \ldots, w_{N-1}\}$. This definition is in line with the intuition that a weighted sample $(\mathbf{s}_q, w_q)$ corresponds to an unweighted sample $\mathbf{s}_q$ observed w_q times.

Taking measures against over-fitting is extremely important when using weighted particles, because the particle weights might differ by orders of magnitude. To this end, we employ a MAP solution (see Sec. 3.2) based on *conjugate priors*. Conjugate priors have the same functional form as the resulting posterior distribution and therefore lead to a greatly simplified Bayesian analysis [Bishop 2006]. Our approach based on the *weighted data log-likelihood* requires a careful treatment of the MAP approach. This is reflected in both the E-step and the M-step of stepwise EM. In the following paragraphs, we extend the sufficient statistics (E-step) and provide formulae for updating the model parameters. Please refer to the supplemental material for their derivation.

Our weighted stepwise EM: E-step. To account for the weight w_q of an observed sample $\mathbf{s}_q$, we modify the sufficient statistics update formula (4) to

$$\mathbf{u}_i^j = (1 - \eta_i)\mathbf{u}_{i-1}^j + \eta_i w_q \gamma_{qj} \mathbf{u}(\mathbf{s}_q), \tag{7}$$

where $\mathbf{u}(\mathbf{s}_q) = (1, \mathbf{s}_q, \mathbf{s}_q\mathbf{s}_q^T)$. The only difference from Equation (4) is the multiplication of the second summand by w_q. This corresponds to the interpretation of weight w_q as a multiplicity of the new observed sample $\mathbf{s}_q$. Additionally, we keep track of the averaged total particle weight required for normalization of mixture weights in the M-step:

$$\overline{w}_i = (1 - \eta_i)\overline{w}_{i-1} + \eta_i w_q, \tag{8}$$

Our weighted stepwise EM: M-step. We have derived an update function for model parameters, $\theta^{\mathrm{new}} = \overline{\theta}(\mathbf{u}_i^1, \ldots, \mathbf{u}_i^K, \overline{w}_i)$, that takes the current modified sufficient statistics (7) and the averaged total particle weight $\overline{w}_i$. By letting $\mathbf{u}_i^j = ((u_\gamma)_i^j, (\mathbf{s})_i^j, (\mathbf{s}\mathbf{s}^T)_i^j)$, we decompose the sufficient statistics $\mathbf{u}_i^j$ into a real number, a vector and a matrix that are computed from Equation (7). Then the specific formulae defining the vector function $\overline{\theta}$ read as follows:

$$\pi_j^{\mathrm{new}} = \frac{\dfrac{(u_\gamma)_i^j}{\overline{w}_i} + \dfrac{\nu - 1}{n}}{1 + \dfrac{K(\nu - 1)}{n}}, \qquad \mu_j^{\mathrm{new}} = \frac{(\mathbf{s})_i^j}{(u_\gamma)_i^j} \tag{9}$$

$$\mathbf{\Sigma}_j^{\mathrm{new}} = \frac{\dfrac{b}{n}\mathbf{I} + \dfrac{(\mathbf{s}\mathbf{s}^T)_i^j - \mathbf{A} + (u_\gamma)_i^j \mathbf{B}}{\overline{w}_i}}{\dfrac{a - 2}{n} + \dfrac{(u_\gamma)_i^j}{\overline{w}_i}} \tag{10}$$

where

$$\mathbf{A} = (\mathbf{s})_i^j (\mu_j^{\mathrm{new}})^T + \mu_j^{\mathrm{new}} (\mathbf{s}^T)_i^j, \qquad \mathbf{B} = \mu_j^{\mathrm{new}} (\mu_j^{\mathrm{new}})^T,$$

$\mathbf{I}$ is the identity matrix, K is the number of mixture components and n is the total number of observed samples (see the details below). Scalars a, b and ν are parameters of conjugate priors induced by the MAP solution (see the supplemental material for more details).

We ran extensive experiments with both synthetic and real data from light transport simulation and by comparing our algorithm results to the reference solutions, we have concluded that the most suitable values for our application are $a = 2.01$, $b = 5 \times 10^{-4}$, $\nu = 1.01$. Likewise, we found that both the on-line and the off-line stepwise EM algorithms achieve the best results when the M-step is executed every $m = 10$ samples (see Alg. 1) and with the stepsize parameter $\alpha = 0.7$ (see Sec. 3.2). We decided to use $K = 8$ components in the mixture as it proved sufficient in all tested scenes.

Differences between the off-line and the on-line versions. The number of observed samples n in Equations (9) and (10) governs the effect of our prior beliefs. The more samples we have observed the weaker the effect of priors. In *on-line* stepwise EM, we simply set n to the current value of the index of sufficient statistics i. However, to fully exploit the MAP approach and thus to prevent over-fitting in our *off-line* stepwise EM, it is necessary to set $n = \min(i, N)$. This is necessary because the algorithm iterates over the same batch of N samples multiple times (see Fig. 3) before it converges and the index i could be much higher than the actual number of unique observed samples.

4.3 Caching of Distributions

Once the EM algorithm creates a hemispherical distribution $p(\omega|\mathbf{x})$ at $\mathbf{x}$, we cache it for reuse at nearby locations. The main reason for using a persistent cache of distributions is to enable their on-line refinement. Our cache is inspired by the traditional lazy evaluation scheme from (ir-)radiance caching [Ward et al. 1988; Křivánek et al. 2005]. It maintains a set of distributions, and for each query point it either returns an existing distribution or creates (i.e. trains) and stores a new one. Our caching scheme, however, exhibits an important difference from (ir-)radiance caching. While (ir-)radiance caching *blurs* the stored values to obtain a biased, yet perceptually plausible result, we strive for an unbiased result.

Spacing of cached distributions. To achieve a good performance, we space the distributions so that they adapt to the angular frequency of the radiance/importance function, as illustrated in Fig. 4 (e.g. for radiance, more distributions should be created in sharp caustics). To space the distributions, we assign to each of them a *validity radius* that determines the maximum spatial distance where the distribution can be reused. The validity radius for a distribution is computed as a harmonic mean of the validity radii of its individual mixture lobes (i.e. GMM components), weighted by the mixing coefficients π_j. To determine the validity radius of a lobe, we first predict how the lobe would change if we observed the environment from a slightly different position (see Fig. 5). We then set the validity radius such that the Kullback-Leiber divergence [Bishop 2006] between the original and the changed lobe stays below a certain threshold for any location within the validity radius. We additionally improve the spacing of distributions by the neighbour clamping heuristic [Křivánek et al. 2006]. We also clamp the validity radii to be between 0.5 and 1 times the distance between the distribution position $\mathbf{x}$ and the furthest particle used for its training. Details are provided in the supplemental material.

Distribution reuse. We have experimented with different interpolation strategies and concluded that simply re-using the nearest suitable distribution is the most robust solution. When we query the cache at the position $\mathbf{x}$, we search for M nearby distributions and from among those we select one with a suitable position and normal orientation. Specifically, we choose the distribution that minimizes

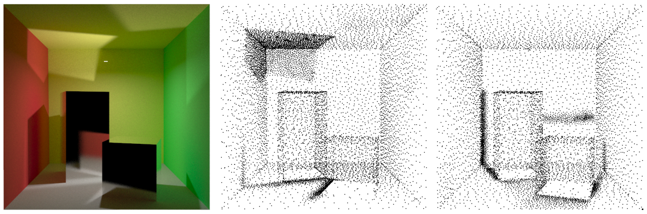

Figure 4: *An example of radiance and importance caches in a Cornell box scene (left). The black dots represent positions of distributions used for guiding camera (center) and light paths (right). The cache adapts to the scene by placing more records where the radiance/importance function contains high angular frequencies.*

$$\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{h} + 2\sqrt{1 - \mathbf{n} \cdot \mathbf{n}_i}, \tag{11}$$

where $\mathbf{x}_i$ and $\mathbf{n}_i$ are the position and the normal of i-th candidate respectively, $\mathbf{n}$ is the normal at $\mathbf{x}$ and h is the distance to the furthest of the M candidates. Finally, we check if the query point $\mathbf{x}$ lies within the validity radius of the selected distribution. If it does, that distribution is reused, otherwise we create a new one.

4.4 Environment Emission Sampling

We have observed that guiding only the scattering directions is insufficient in complex scenes with environment lighting. This is because most particles emitted from the environment fail to enter important parts of the scene through small openings (see Fig. 1). Here, we present our approach for driving the particle emission from an environment light source by importance. Emission from other light source types is left for future work.

An environment light defines the emitted radiance $L_e(\omega)$ over directions ω. Emitting a particle from an environment light requires sampling a joint PDF $p(\mathbf{x}, \omega) = p(\omega)p(\mathbf{x}|\omega)$ that we factor into the PDF $p(\omega)$ to sample the particle direction ω, and $p(\mathbf{x}|\omega)$ to sample the starting position $\mathbf{x}$ on a disk outside the scene that is perpendicular to ω [Georgiev 2012; Pharr and Humphreys 2010]. These distributions are trained by the importons that have left the scene (see Fig. 6a) during the *importance distributions update* step in Fig. 2.

Directional distribution. We compute $p(\omega)$ as a product of $p_{\mathrm{L}}(\omega) \propto L_e(\omega)$ given by the environment map and of $p_{\mathrm{W}}(\omega)$

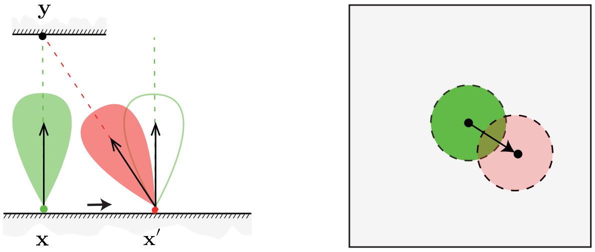

Figure 5: *Calculation of the lobe validity radius. Left: We assume that the green lobe of a distribution at the position $\mathbf{x}$ corresponds to an importance/radiance highlight at the position $\mathbf{y}$ seen from $\mathbf{x}$ along the lobe axis. If we move the lobe to a new position $\mathbf{x}'$, the direction of the lobe (red) pointing towards the highlight changes. The validity radius is set such that the resulting change of the lobe, measured by the KL-divergence, does not exceed a given threshold. Right: The same situation in the unit square domain where the corresponding Gaussian distribution is defined and where the KL-divergence is actually measured.*

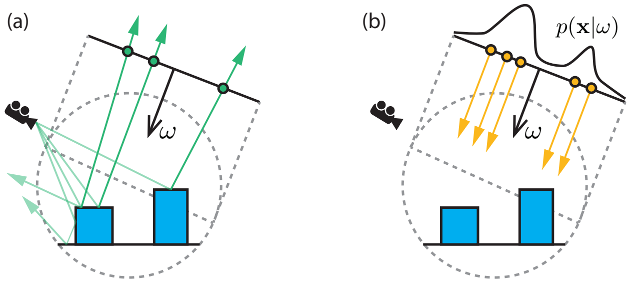

Figure 6: *To learn the importance distribution $p(\mathbf{x}|\omega)$ for sampling starting positions of light paths from the environment, conditional on the direction ω, we construct a disk perpendicular to ω outside the scene. A GMM model of the distribution is trained from importons that leave the scene in a direction less than 1° from ω (a). The position $\mathbf{x}$ of the light paths emitted in a direction close to ω is sampled using the learned distribution $p(\mathbf{x}|\omega)$ (b).*

that is the directional distribution of importance reconstructed from the importons that left the scene. We represent $p_{\mathrm{W}}(\omega)$ with a fixed-resolution bitmap. In each pixel, the PDF is computed from importon directions using a progressive kernel density estimate [Hachisuka et al. 2008]. To avoid introducing bias due to zero probability of sampling directions that did not receive any importons, we combine sampling from the product $p_{\mathrm{L}}(\omega) \times p_{\mathrm{W}}(\omega)$ and from $p_{\mathrm{L}}(\omega)$ via multiple importance sampling [Veach 1997].

Position distributions. The distributions $p(\mathbf{x}|\omega)$ defined on a disk perpendicular to ω (see Fig. 6b) are stored in a unit sphere cache indexed by ω. This allows to reuse the stored distributions for nearby directions and enables their on-line training. To represent $p(\mathbf{x}|\omega)$, we train our GMM (see Sec. 3.2) from importons that left the scene in a direction less than 1° from ω. Prior to the training, these importons are projected onto the perpendicular disk (see Fig. 6a).

4.5 Russian Roulette

Path termination via Russian Roulette (RR) [Arvo and Kirk 1990] affects the distribution of particles together with their weights, without changing the expected value of the quantity they represent. This is commonly called *biasing* in the neutron transport literature [Booth 1985; Booth 2012]. Our method is also a biasing technique: we strive to reduce variance by guiding particles toward important areas (e.g. in front of the camera). However, combining different biasing techniques could be counterproductive. When we use our guiding method, we must not base the random walk termination by RR on a local decision (derived from e.g. the surface albedo) as is commonly done in light transport simulation [Pharr and Humphreys 2010]. This approach might result in a termination of particles that could eventually yield an important contribution.

During particle tracing, we base our Russian roulette on particle weights. We set the RR survival probability to $\min\{^{w_i}/_{w_i^* w_t}, 1\}$, where w_i is the current particle weight, w_i^* is the particle weight before any biasing or scattering (i.e. light source power for photons or source importance for importons) and w_t is a fixed, empirically determined threshold that we set to 10^{-5} for all our renderings. In other words, we do not attempt to terminate particles unless their weight drops at least w_t times from its initial value. Note that due to guiding, important areas contain many particles with very small weights and our RR ensures that these particles are not killed. Ideally, w_t should locally adapt to the distribution of importance or radiance in the scene [Haghighat and Wagner 2003].

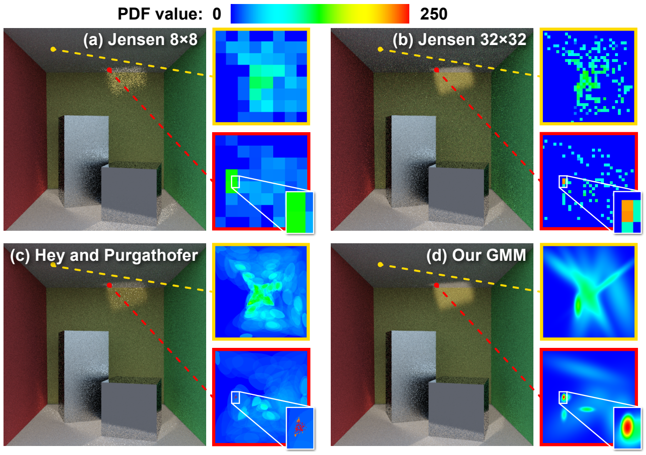

Figure 7: *Demonstration of the superior flexibility of the parametric Gaussian mixture model (GMM) over previously used models. Four renderings of a Cornell box scene with diffuse walls and two glossy blocks lit by the sun are rendered by guided path tracing using Jensen's [1995] method with the histogram resolution of 8×8 (a) and 32×32 (b), Hey and Purgathofer's [2002] hemispherical footprints (c), and with our GMM (d). The distributions trained at two selected locations in the scene are also visualized. One distribution contains low-frequency illumination while the other contains a sharp directional peak caused by a reflection of the sun.*

5 Applications and Results

We first demonstrate the flexibility of the Gaussian mixture model in rendering. Then we show that guiding various path-sampling algorithms using our progressively trained distributions provides superior rendering results in complex, highly occluded scenes.

5.1 Flexibility of the Gaussian Mixture Model

We compare our GMM to Jensen's histograms [1995] and Hey and Purgathofer's hemispherical footprints [2002] in terms of their ability to model distributions encountered in rendering. Fig. 7 shows a simple Cornell box scene with two glossy blocks. It is rendered with path tracing using the histogram model with two different histogram resolutions, the hemispherical footprints model, and finally our GMM. Note that the on-line learning, which is discussed in the following section, is not applied so that the comparison is fair. Instead, during the training phase, we emit 5M photons in a single batch and all the models are trained from 250 nearest photons.

The figure shows that higher histogram resolution captures high frequencies in the incoming radiance distribution inside caustics, while the ability to represent low frequencies deteriorates at the same time due to over-fitting. Thus, while the quality of caustics at the histogram resolution of 32×32 pixels is superior to the quality at the 8×8 resolution, the noise is increased on the walls.

Hemispherical footprints are more flexible than the fixed histogram grid as they take into account the directional density of particles. However, the method is closely related to kernel density estimation and as such it suffers from the optimal bandwidth selection problem [Silverman 1986]. Visualization of a distribution inside the sun's reflection suggests insufficient generalization – the distribution is discontinuous between individual observed samples in its peak.

Finally, the GMM with only 8 components in the mixture exhibits sufficient flexibility to represent both low frequencies and sharp peaks in the distributions. Note that when compared to the other

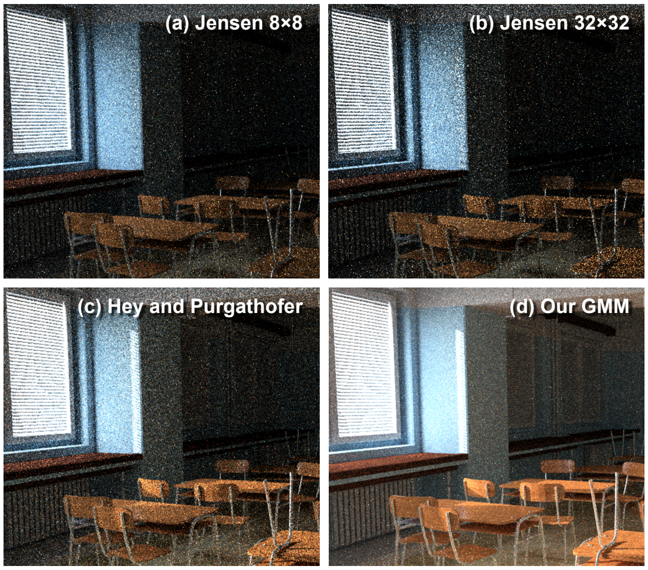

Figure 8: *Equal-time (1h) comparison of different distribution models in a scene with difficult visibility. Light enters the classroom from the sun and the sky through small gaps between the window blinds. The scene was rendered by path tracing guided by 8×8 (a) and 32×32 (b) histogram model, hemispherical footprints model (c), and our Gaussian mixture model (d). Each method was given a fixed number of emitted photons (500M) in one unguided tracing step. Each distribution was then trained from a single batch of 500 nearest photons without additional on-line learning.*

methods, the GMM excels at generalizing from the observed data (i.e. it avoids discontinuities between the individual samples). This is further supported by Fig. 8, which shows that the GMM provides superior rendering results in a scene with difficult visibility.

5.2 On-line Learning Results

In Fig. 9, we compare classical implementations of (1) path tracing (PT), (2) bidirectional path tracing (BDPT) and (3) vertex connection and merging (VCM) [Georgiev et al. 2012b] to the same implementations guided by our method with the distributions trained in the on-line (progressive) manner. We also show images rendered by (4) Veach's Metropolis light transport with manifold exploration (Veach MLT) [Jakob and Marschner 2012]. We also present corresponding L_1 error plots in Fig. 10. The supplemental material contains additional results for progressive photon mapping (PPM) and other flavours of MLT (Kelemen MLT [2002] and energy redistribution path tracing [Cline et al. 2005]) as well as RMSE plots.

Setup. Our method uses 30 training passes in all the presented scenes. We set the maximum path length to 40 bounces. All images, save the references, were rendered in 1 hour (including the training phase) on a single Intel Core i7-2600K CPU using 8 logical cores. The reference images were rendered with BDPT for 10–60 days. We used the Corona Renderer [Karlík 2009] to produce all the results except the MLT images, which were rendered in Mitsuba [Jakob 2010] with default settings. We verified that our BDPT implementations in both renderers converge to the same result.

Scenes. In Fig. 9, we show renderings of three scenes that feature difficult visibility and contain many locations that are poorly sampled with regular path sampling-based methods. Two of the scenes (Living room and Classroom) feature environment lighting and use our method for importance-driven emission sampling. All the light in rendered images is due to indirect illumination.

The **Living room** scene allows only a small fraction of the light from the sun and the sky to enter the room through a glass window and a small gap between the curtains. The walls are diffuse and there are a few semi-glossy objects, including the floor. Note that even BDPT and VCM struggle to resolve the image in the dark closet on the left. The scene was rendered at a resolution of 1024×576 pixels. One million photons and importons were emitted in each training pass. The training phase took 10.3 minutes.

The **Classroom** scene is lit by the sun and the sky through window blinds. For regular MC algorithms, it is especially difficult to sample the dark half of the classroom. Windows and highly glossy chairs and table legs together with a semi-glossy floor create specular–glossy–glossy light transport paths that form many caustics/indirect highlights. While our method improves sampling of the caustics and the dark parts of the scene, sampling of the glossy-glossy highlights remains a challenge. The reason is that we do not sample from the product of the BRDF and the incoming radiance/importance, but rather combine the two via multiple importance sampling. The scene was rendered at a resolution of 960×480 pixels. 0.5 million photons and importons were emitted in each training pass. The training phase took 7 minutes.

In the **Door** scene, light enters the room through a small slit. This is a recreation of the well-known scene from Veach and Guibas' paper [1997] provided by Lehtinen et al. [2013]. To make the scene more realistic, we have made the area light source much smaller and used more light bounces. The scene was rendered at a resolution of 800×600 pixels. 300k photons and importons were emitted in each training pass. The training phase took 3.75 minutes.

Error and convergence. Fig. 10 shows the dependence of the L_1 error on time in all three scenes, comparing both the classical and our guided versions of the algorithms. Fig. 11 demonstrates that our method converges to the reference.

On-line learning. To demonstrate progressive improvement of distributions during the training phase, we rendered the *Living room* scene several times with guided BDPT. We used 2, 5 and 30 training passes, while all other settings were kept the same. In Fig. 1, we present an equal-time comparison (1 hour) including the time spent on the training phase. The results reveal that the time spent on additional training passes is quickly amortized by the superior performance of the subsequent guided rendering.

Discussion. Guided BDPT and PPM (see the supplemental material) yield superior results compared to their classical versions. However, although VCM is a combination of BDPT and PPM, the improvement of guided VCM is only subtle over guided BDPT. We suspect that the guiding might render some path sampling techniques – that would otherwise be an essential component of the combined algorithm – less important.

6 Discussion, Limitations and Future Work

Alternative distribution models. We have experimented with a range of alternatives to Gaussian mixtures. The *von Mises-Fisher* (vMF) distribution [1953] is defined directly on the unit sphere, but its isotropic shape is not suitable for fitting highly anisotropic structures in the directional domain, often produced by caustics. The *Kent* distribution [1982] is a directional distribution that does support anisotropy, but becomes bi-modal for a range of parameters and is numerically unstable. An anisotropic spherical function [Xu et al. 2013] derived from the Bingham distribution [1974] was developed for interactive rendering. However, at the moment, there is no known analytical sampling procedure. We further experimented with an anisotropic distribution similar to the lobe of the *Ward BRDF* [Ward 1992], but its learning was too expensive.

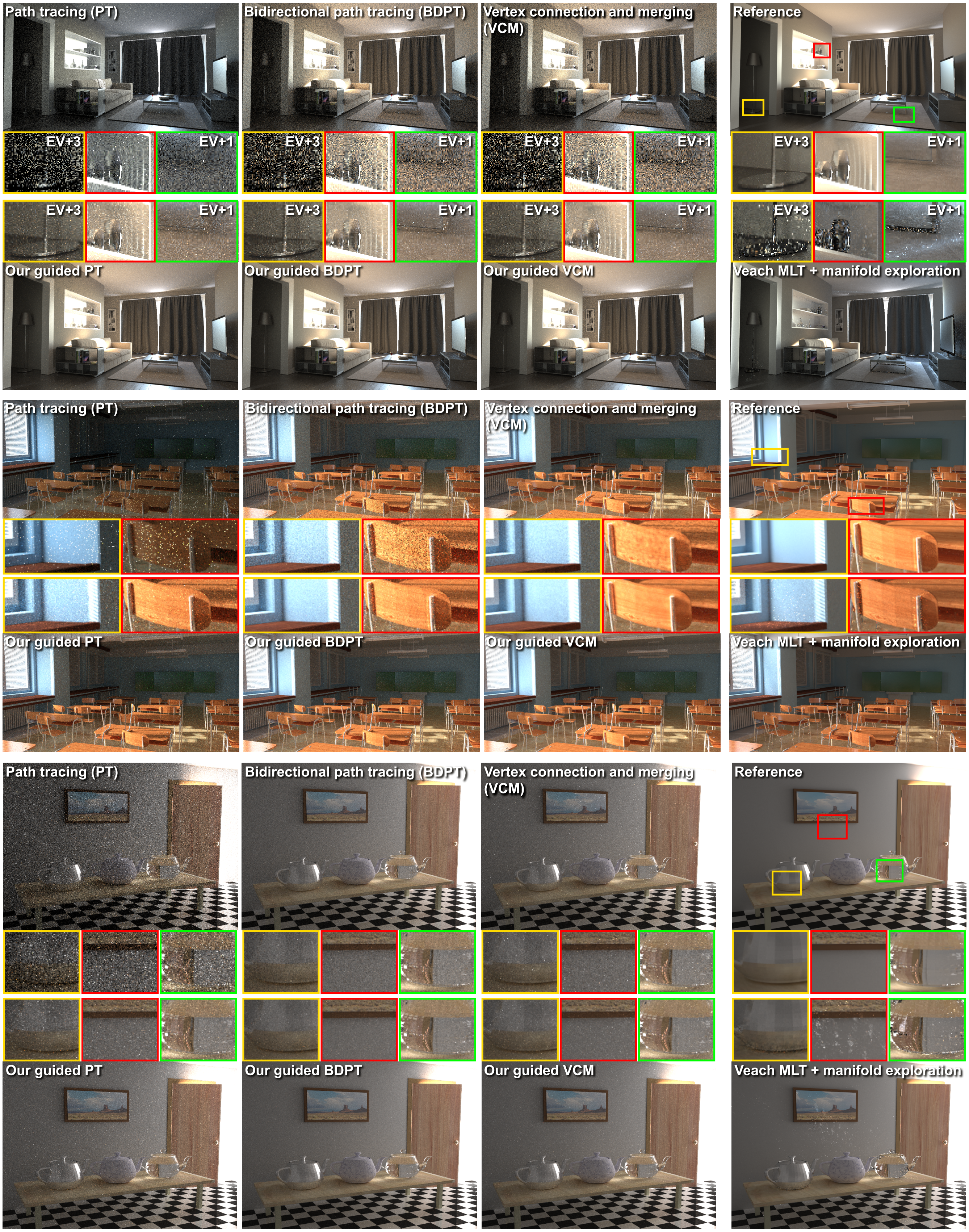
Path tracing (PT)
Bidirectional path tracing (BDPT)
Vertex connection and merging (VCM)
Reference
EV+3
EV+1
Our guided PT
Our guided BDPT
Our guided VCM
Veach MLT + manifold exploration

Figure 9: *Three scenes with complex visibility rendered with path tracing (PT), bidirectional path tracing (BDPT), vertex connection and merging (VCM), and their respective versions guided by our method. We also present reference images and results of Veach's Metropolis light transport (MLT) with manifold exploration. All images, except the references, were calculated in one hour, including the time spent on 30 training passes. The results show that the overhead of our method is amortized by the improved sampling, as the noise levels are reduced in all tested algorithms, especially in dark areas. ("EV+v" in the insets refers to a multiplication of the image brightness by 2^v.)*

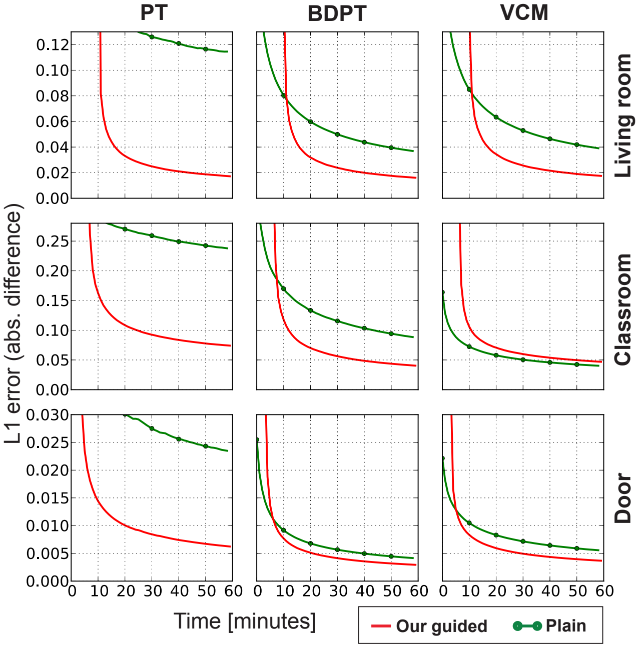

Figure 10: *Time dependence of L_1 error for 60 minutes of rendering. The learning phase of our method is included in the comparison, so the graphs for the guided versions do not start at zero time.*

Limitations and Future Work. The overhead of our method in the rendering phase, that may offset its advantages in simple scenes, comes mostly from querying the cache. This takes 26% of the execution time for path tracing, and up to 45% for bidirectional methods, since they need to access both radiance and importance caches at each path vertex. Optimization should focus on this aspect, since sampling and evaluation of the GMMs each takes only about 5% of the total time.

Importance sampling based on our distributions can occasionally generate excessive particle weights (training phase) or path contributions (rendering phase) and produce spiky image noise. Splitting and Russian roulette based on our importance/radiance distributions and particle weights [Haghighat and Wagner 2003] could resolve this issue and improve overall performance.

While the fixed number of components in the mixture may be insufficient to capture some complex distributions, we have not encountered any problems due to this limitation in our tests. Nonetheless, adaptive determination of the number of components would be an interesting avenue for future work.

Because the distributions created in the later training passes have access to fewer particles, they may be less refined than the distributions created earlier. Adaptive refinement of the distributions during rendering could help resolve this issue.

Since our distributions only model incoming radiance or importance, we cannot provide good importance sampling of some complex glossy-glossy inter-reflections. This could be alleviated by sampling from the product of the incoming radiance or importance and the BRDF.

Our importance-driven emission is currently limited to environment light sources. Using our distributions to sample emission from other light source types, as well as for radiance-driven emission of importons from the camera, would improve the efficiency of our method.

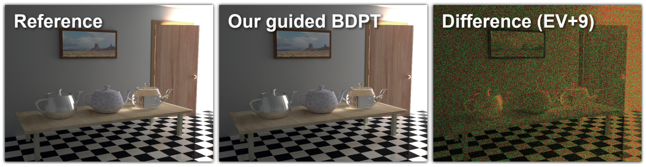

Figure 11: *Our method converges to the reference image computed by plain BDPT (left). We rendered the scene for approximately two days with our guided BDPT that used 30 TP (middle). The reference image was rendered by BDPT for approximately 10 days. The uniform distribution of positive (green) and negative (red) differences (right) suggests that any residual error is only due to variance. Brightness of the difference image was multiplied by 2^9.*

Finally, our method shares the same overall goal with Metropolis light transport, that is globally optimized importance sampling of entire light transport paths. It would be interesting to see if the two approaches can complement each other to achieve further benefits.

7 Conclusion

We have proposed the use of a parametric mixture model to represent directional distributions for importance sampling in Monte Carlo light transport simulation. The core of our approach is an on-line learning procedure that allows one to train the distributions from a potentially infinite stream of particles. With this approach, we can recover good importance sampling distributions in difficult lighting configurations, where an excessively large number of particles would otherwise be necessary. This, in turn, enables rendering scenes with complex visibility, where the existing state-of-the-art methods are ineffective.

Acknowledgements

The work was supported by Charles University in Prague, project GA UK No 580612, by the grant SVV2014260103, and by the Czech Science Foundation grant P202-13-26189S. We thank Jan Beneš, Alexander Wilkie, and Iliyan Georgiev for their feedback on the paper draft and to Ludvík Koutný for providing us with the Living Room scene.

References

Arvo, J., and Kirk, D. 1990. Particle transport and image synthesis. In *Proc. SIGGRAPH '90*, ACM, 63–66.

Bashford-Rogers, T., Debattista, K., and Chalmers, A. 2012. A significance cache for accelerating global illumination. *Computer Graphics Forum 31*, 6, 1837–51.

Bashford-Rogers, T., Debattista, K., and Chalmers, A. 2013. Importance driven environment map sampling. *IEEE Transactions on Visualization and Computer Graphics 19*.

Bingham, C. 1974. An antipodally symmetric distribution on the sphere. *The Annals of Statistics 2*, 6, pp. 1201–1225.

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning*. Springer.

Booth, T. 1985. A sample problem for variance reduction in MCNP. Tech. rep., Los Alamos National Laboratory, Los Alamos, New Mexico 87545.

Booth, T. E. 2012. Common misconceptions in Monte Carlo particle transport. *Applied Radiation and Isotopes 70*, 7.

Budge, B. C., Anderson, J. C., and Joy, K. I. 2008. Caustic forecasting: Unbiased estimation of caustic lighting for global illumination. *Computer Graphics Forum 27*, 7, 1963–70.

Cappé, O. 2011. *Online Expectation Maximisation*. John Wiley & Sons, Ltd, 31–53.

Cline, D., Talbot, J., and Egbert, P. 2005. Energy redistribution path tracing. *ACM Trans. Graph. 24*, 3.

Cline, D., Adams, D., and Egbert, P. 2008. Table-driven adaptive importance sampling. *Computer Graphics Forum 27*.

Dempster, A. P., Laird, N. M., and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B 39*, 1, 1–38.

Dutré, P., and Willems, Y. 1994. Importance-driven Monte Carlo light tracing. In *Eurographics Workshop on Rendering*.

Dutré, P., and Willems, Y. 1995. Potential-driven Monte Carlo particle tracing for diffuse environments with adaptive probability density functions. In *EG Workshop on Rendering*.

Dutré, P., Bala, K., and Bekaert, P. 2006. *Advanced Global Illumination*, 2nd ed. A. K. Peters.

Fisher, R. 1953. Dispersion on a sphere. *Proc. Royal Society of London. Series A. Math. Phys. 217*, 1130, 295–305.

Gauvain, J., and Lee, C.-H. 1994. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Trans Audio Speech Lang Processing 2*, 2, 291–298.

Georgiev, I., Křivánek, J., Popov, S., and Slusallek, P. 2012. Importance caching for complex illumination. *Computer Graphics Forum 31*, 2pt3, 701–10. Proc. of Eurographics.

Georgiev, I., Křivánek, J., Davidovič, T., and Slusallek, P. 2012. Light transport simulation with vertex connection and merging. *ACM Trans. Graph. 31*, 6.

Georgiev, I. 2012. Implementing vertex connection and merging. Tech. rep., Saarland University.

Hachisuka, T., Ogaki, S., and Jensen, H. W. 2008. Progressive photon mapping. *ACM Trans. Graph. 27*, 5 (Dec.).

Hachisuka, T., Pantaleoni, J., and Jensen, H. W. 2012. A path space extension for robust light transport simulation. *ACM Trans. Graph. 31*, 6.

Haghighat, A., and Wagner, J. C. 2003. Monte Carlo variance reduction with deterministic importance functions. *Progress in Nuclear Energy 42*, 1, 25 – 53.

Hey, H., and Purgathofer, W. 2002. Importance sampling with hemispherical particle footprints. In *SCCG*.

Jakob, W., and Marschner, S. 2012. Manifold exploration: A Markov chain Monte Carlo technique for rendering scenes with difficult specular transport. *ACM Trans. Graph. 31*, 4 (July).

Jakob, W., Regg, C., and Jarosz, W. 2011. Progressive expectation-maximization for hierarchical volumetric photon mapping. *Computer Graphics Forum 30*, 4, 1287–1297.

Jakob, W., 2010. Mitsuba renderer. http://mitsuba-renderer.org.

Jensen, H. W. 1995. Importance driven path tracing using the photon map. In *Eurographics Workshop Rendering*, 326–335.

Karlík, O., 2009. Corona Renderer. http://corona-renderer.com.

Kelemen, C., Szirmay-Kalos, L., Antal, G., and Csonka, F. 2002. A simple and robust mutation strategy for the Metropolis light transport algorithm. *Comp. Graph. Forum (Proc. of Eurographics) 21*, 3, 531–540.

Kent, J. T. 1982. The Fisher-Bingham distribution on the sphere. *Journal of the Royal Statistical Society. Series B 44*, 1, 71–80.

Křivánek, J., Gautron, P., Pattanaik, S., and Bouatouch, K. 2005. Radiance caching for efficient global illumination computation. *IEEE Trans. Vis. Comp. Graph. 11*, 5.

Křivánek, J., Bouatouch, K., Pattanaik, S. N., and Žára, J. 2006. Making radiance and irradiance caching practical: Adaptive caching and neighbor clamping. In *Eurographics Symposium on Rendering*, 127–138.

Lafortune, E. P., and Willems, Y. D. 1995. A 5D tree to reduce the variance of Monte Carlo ray tracing. In *Eurographics Workshop on Rendering*. 11–20.

Lehtinen, J., Karras, T., Laine, S., Aittala, M., Durand, F., and Aila, T. 2013. Gradient-domain Metropolis light transport. *ACM Trans. Graph. 32*, 4.

Lepage, G. P. 1978. A new algorithm for adaptive multidimensional integration. *Journal of Comp. Physics 27*, 192–203.

Liang, P., and Klein, D. 2009. Online EM for unsupervised models. In *Human Language Technologies (NAACL '09)*, Association for Computational Linguistics, 611–619.

Pegoraro, V., Brownlee, C., Shirley, P. S., and Parker, S. G. 2008. Towards interactive global illumination effects via sequential Monte Carlo adaptation. In *IEEE Symposium on Interactive Ray Tracing*, 107–114.

Pegoraro, V., Wald, I., and Parker, S. G. 2008. Sequential Monte Carlo adaptation in low-anisotropy participating media. *Computer Graphics Forum 27*, 4, 1097–1104.

Peter, I., and Pietrek, G. 1998. Importance driven construction of photon maps. In *Rendering Techniques*, 269–80.

Pharr, M., and Humphreys, G. 2010. *Physically Based Rendering, Second Edition: From Theory To Implementation*, 2nd ed. Morgan Kaufmann Publishers Inc., San Francisco, CA.

Sato, M.-A., and Ishii, S. 2000. On-line EM algorithm for the normalized Gaussian network. *Neural Comput. 12*, 2 (Feb.).

Shirley, P., and Chiu, K. 1997. A low distortion map between disk and square. *J. Graph. Tools 2*, 3 (Dec.), 45–52.

Silverman, B. W. 1986. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London.

Steinhurst, J., and Lastra, A. 2006. Global importance sampling of glossy surfaces using the photon map. In *IEEE Symposium on Interactive Ray Tracing*, 133–138.

Tsai, Y.-T., Chang, C.-C., Jiang, Q.-Z., and Weng, S.-C. 2008. Importance sampling of products from illumination and BRDF using spherical radial basis functions. *The Visual Computer 24*, 7.

Veach, E., and Guibas, L. J. 1997. Metropolis light transport. In *SIGGRAPH '97*.

Veach, E. 1997. *Robust Monte Carlo methods for light transport simulation*. PhD thesis, Stanford University.

Verbeek, J. J., Nunnink, J. R., and Vlassis, N. 2006. Accelerated EM-based clustering of large data sets. *Data Mining and Knowledge Discovery 13*, 3, 291–307.

Ward, G. J., Rubinstein, F. M., and Clear, R. D. 1988. A ray tracing solution for diffuse interreflection. In *SIGGRAPH '88*, no. 4, ACM, 85–92.

Ward, G. J. 1992. Measuring and modeling anisotropic reflection. In *SIGGRAPH '92*, ACM, 265–272.

Xu, K., Sun, W.-L., Dong, Z., Zhao, D.-Y., Wu, R.-D., and Hu, S.-M. 2013. Anisotropic spherical Gaussians. *ACM Transactions on Graphics 32*, 6, 209:1–209:11.