Selective guided sampling with complete light transport paths

FLORIAN REIBOLD, JOHANNES HANIKA, ALISA JUNG, AND CARSTEN DACHSBACHER, KIT, Germany

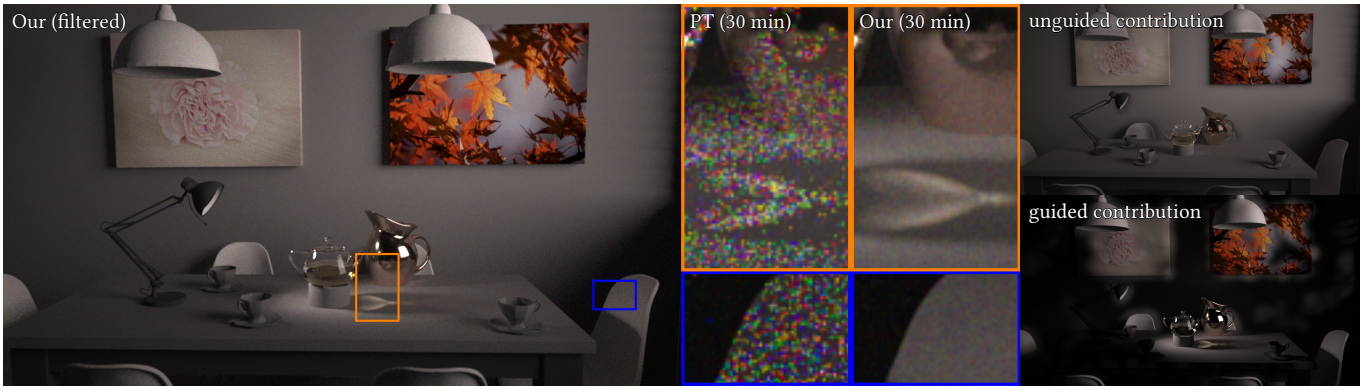

Fig. 1. A dining room lit from the outside by an area light and by a spot light on the table. The insets show that our (unidirectional) guided sampling is able to resolve the caustic much better than path tracing (PT) in the same time. Guided sampling learns the difficult parts of the light transport in 3 minutes (*guided contribution*, bottom right image) and rendered for 27 minutes. Our approach is selective, i.e. the guided sampler automatically focuses only on the parts of the illumination which are poorly sampled by the unguided estimator. The images on the right show the unguided and guided component of our method.

Finding good global importance sampling strategies for Monte Carlo light transport is challenging. While estimators using local methods (such as BSDF sampling or next event estimation) often work well in the majority of a scene, small regions in path space can be sampled insufficiently (e.g. a reflected caustic). We propose a novel data-driven guided sampling method which selectively adapts to such problematic regions and complements the unguided estimator. It is based on complete transport paths, i.e. is able to resolve the correlation due to BSDFs and free flight distances in participating media. It is conceptually simple and places anisotropic truncated Gaussian distributions around guide paths to reconstruct a continuous probability density function (guided PDF). Guide paths are iteratively sampled from the guided as well as the unguided PDF and only recorded if they cause high variance in the current estimator. While plain Monte Carlo samples paths independently and Markov chain-based methods perturb a single current sample, we determine the reconstruction kernels by a set of neighbouring paths. This enables local exploration of the integrand without detailed balance constraints or the need for analytic derivatives. We show that our method can decompose the path space into a region that is well sampled by the unguided estimator and one that is handled by the new guided sampler. In realistic scenarios, we show 4× speedups over the unguided sampler.

CCS Concepts: • **Computing methodologies** → **Ray tracing**;

Additional Key Words and Phrases: Global Illumination, Sampling and Reconstruction, Monte Carlo, Stochastic Sampling

1 INTRODUCTION

Photorealistic image synthesis is important for product visualisation, architecture, and visual effects in movie production. In these fields, Monte Carlo light transport simulation has found widespread adoption due to its ability to accurately simulate complex lighting effects. In this class of techniques, path tracing is still the most frequently used technique among practitioners, due to its simplicity and easy maintenance. This relatively simple solution – starting at the camera sensor, iteratively extending transport paths to more bounces, combined with next event estimation (i.e. deterministically connecting to the light sources) – can efficiently account for the vast majority of rendering problems in practice [Fascione et al. 2017].

However, path tracing often encounters transport phenomena which it fails to sample sufficiently. Although the source of the difficulty may be very localised, it can cause noise virtually everywhere in the image. More robust solutions have been proposed, e.g. [Georgiev et al. 2012; Hachisuka et al. 2012; Veach and Guibas 1997]), but have not found broad adoption so far. The reasons are many: they often suffer from temporal flickering or blurry artifacts, and some cannot handle special requirements such as hair strands or participating media – or the computational or memory overhead they introduce is just too high.

In this paper, we present a novel method that addresses these issues and balances local exploration and global discovery of transport paths. The global discovery uses stratified sampling with uniformly distributed points, e.g. a path tracer using the Halton sequence or bidirectional path tracing (BDPT) [Veach and Guibas 1994]. As a basis for the local exploration, we store samples causing high variance as *guide paths*. The key is to store full paths, as opposed to a disconnected set of end points [Vorba et al. 2014] or path segments [Jensen 1995]. By this, we can make use of all the available information during local exploration by creating new paths which are sampled using Gaussian distributions around the guide paths. These newly spawned paths are often selected as new guide paths for subsequent sampling. In particular, our method

- is selective, i.e. we introduce an iterative learning process which identifies sub-spaces of the integration causing high variance.

Author's address: Florian Reibold, Johannes Hanika, Alisa Jung, and Carsten Dachsbacher, KIT, Am Fasanengarten 5, Karlsruhe, BW, 76131, Germany, florian.simon@kit.edu.

- uses information from complete paths, i.e. guided sampling accounts for path length, Russian roulette, BSDF, visibility, etc.
- samples from a *set* of neighbouring paths. Plain Monte Carlo constructs independent samples, Markov chains mutate from a single current state sample, while we use the full information from multiple previous paths to sample a new one.

We show that our approach can work in a variety of scene configurations, including special cases which are difficult for previous work such as participating media, hair fibers.

2 BACKGROUND AND PREVIOUS WORK

Light transport. This field is well studied [Chandrasekar 1960] and has been formalised in computer graphics using the rendering equation [Kajiya 1986] and the path integral [Veach 1998]. To compute the amount of radiant energy I_p incident on a pixel p, we need to integrate differential spectral radiant flux over a wavelength range (weighted by sensor responsivity), shutter time interval, aperture area of the optical device, and the vertex areas of all relevant points of interaction of the transport path with the scene:

$$I_p = \int_{\mathcal{P}} h_p(\mathbf{X}) \cdot f(\mathbf{X}) \, \mathrm{d}\mathbf{X}, \tag{1}$$

where $\mathcal{P}$ is the *path space* and a path $\mathbf{X}$ of length k is formed by a list of vertices $\mathbf{X} = (\mathbf{x}_0, \mathbf{x}_1, .., \mathbf{x}_{k-1})$; $\mathbf{x}_0$ being on the sensor and $\mathbf{x}_{k-1}$ on a light. Eq. (1) includes the pixel reconstruction filter $h_p(\mathbf{X})$ and the *measurement contribution function* $f(\mathbf{X})$ [Veach 1998]:

$$f(\mathbf{X}) = W(\mathbf{x}_0) \cdot G(\mathbf{x}_0, \mathbf{x}_1) \cdot T(\mathbf{x}_0, \mathbf{x}_1) \cdot L_e(\mathbf{x}_{k-1}) \cdot \prod_{v=1}^{k-2} f_s(\mathbf{x}_v) \cdot G(\mathbf{x}_v, \mathbf{x}_{v+1}) \cdot T(\mathbf{x}_v, \mathbf{x}_{v+1}), \text{ with} \tag{2}$$

$$G(\mathbf{x}, \mathbf{y}) = D(\mathbf{x}, \mathbf{y}) \cdot D(\mathbf{y}, \mathbf{x}) / \|\mathbf{x} - \mathbf{y}\|^2, \text{ and} \tag{3}$$

$$D(\mathbf{x}, \mathbf{y}) = \begin{cases} \langle n(\mathbf{x}), \omega_{\mathbf{x}\to\mathbf{y}} \rangle & \mathbf{x} \text{ on surface,} \\ \sqrt{1 - \langle r(\mathbf{x}), \omega_{\mathbf{x}\to\mathbf{y}} \rangle^2} & \mathbf{x} \text{ on fiber,} \\ 1 & \mathbf{x} \text{ in medium.} \end{cases} \tag{4}$$

For the sake of simpler notation, we omitted the dependencies on wavelength and incident and outgoing directions for the scattering function $f_s(\mathbf{x})$. It takes the form of a *bidirectional scattering distribution function* (BSDF) on surfaces, scattering coefficient times phase function $\mu_s(\mathbf{x}) \cdot \varphi(\cos\theta)$ in volumes, or a curve scattering function on hair fibers. $T(.)$ denotes the volumetric transmittance between the two given points. $D(\mathbf{x}, \mathbf{y})$ accounts for the foreshortening according to the surface normal $n(\mathbf{x})$ or the main fiber direction $r(\mathbf{x})$, and has no effect in media. We will also omit the dependency on pixel position p and wavelength λ wherever possible.

Monte Carlo methods. To compute integrals such as Equation (1), many Monte Carlo (MC) [Sobol 1994] or Markov chain Monte Carlo (MCMC) approaches [Hastings 1970; Metropolis et al. 1953] have been devised and applied to light transport [Kelemen et al. 2002; Pharr et al. 2017; Veach and Guibas 1997].

While MC is usually good at uniformly discovering islands in the path space, MCMC approaches are better at exploring them once found. A hybrid approach exists [Cline et al. 2005], but often requires manual, scene-dependent intervention to determine what should be explored by MC or MCMC to avoid wasteful sampling and unnecessary flickering in animations.

Our technique explicitly determines which parts of the path space get explored well by MC, and will then spend additional work only on the remaining parts. We achieve this by storing transport paths in subspaces which are intricate to sample and by this acquire information about the transport in a scene. This is in contrast to MCMC methods which always only depend on a single current state.

Advanced (MC)MC light transport methods. Previous work exploits additional information to guide local exploration in MCMC. In particular, geometric derivatives [Jakob 2013] have been used to globally optimise all vertices of a path simultaneously. This work clearly shows the interdependencies of path vertices along a full path in form of a block-tridiagonal matrix, which maps half vector constraints to vertex area locations (cf. [Jakob and Marschner 2012, Figure 2(c)] or [Kaplanyan et al. 2014, Equation (11)]). This motivates our approach of working with complete transport paths (see Section 6 in the supplemental document for an extended discussion of the tridiagonal structure).

Analytic derivatives of the measurement contribution have been explored to help with computing step sizes [Fasiolo et al. 2018; Li et al. 2015]. Instead, we employ an observed light field, since fine displacements and visibility make analytical derivatives unstable [Hanika et al. 2015].

MCMC has also been augmented with large caches (photon maps) which represent visibility and information beyond a single state [Gruson et al. 2016; Hachisuka and Jensen 2011; Šik et al. 2016]. However, we want our method to focus only on small subspaces of the path space where exploration is difficult. Moreover, photon maps do not encode the full information about complete transport paths, but represent a 2D marginal distribution (4D if incoming directions are stored).

High-dimensional importance sampling. Metropolis light transport can importance sample complete paths [Veach and Guibas 1997] in the limit by performing many small mutations. There are approaches to incorporate more global information into regular MC sampling. Joint importance sampling [Georgiev et al. 2013] successively derives joint probability density functions (PDFs) for sampling 2 to 3 vertices in participating media at once. Weber et al. [2017] use an uncached, deterministic path construction to construct in the order of 10 vertices inside a medium towards a light source as an extended next event estimation. Closed form approximations have been used to guide scattering events towards a light source in dense media [Křivánek and d'Eon 2014; Meng et al. 2016]. Note that all of these are strictly limited to participating media.

Particle filters. Sequential Monte Carlo or particle filters [Del Moral 1996] are usually used to discover the posterior distribution of a hidden Markov model (for instance tracking one or multiple moving 2D targets over time). They can be used to estimate a PDF by first sampling an initial set of particles followed by a move and resampling step. As this approach suffers from progressive degeneration, i.e. it tends to get stuck in a single mode, many countermeasures have been proposed, see e.g. [Gilks and Berzuini 2001]. In our context, however, we are computing a static equilibrium distribution

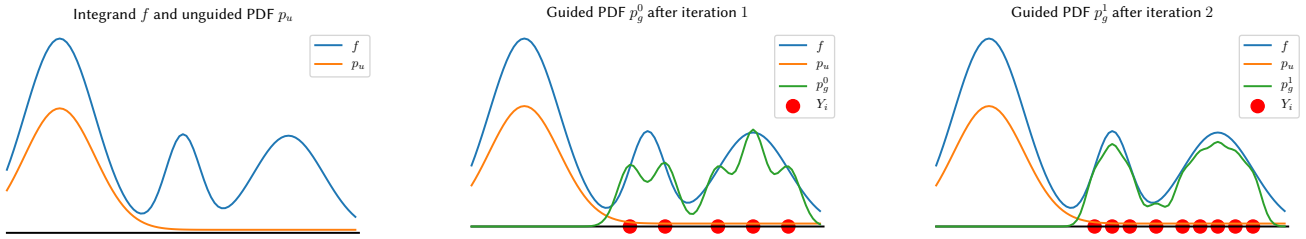

Fig. 2. A 1D illustration of our approach: parts of the integrand (f, blue) are well represented by a PDF (p_u, orange). We wish to approximate the difference between the integrand and the orange PDF. Guiding records ($\mathbf{Y}_i$, red points) are iteratively placed and then used to reconstruct a continuous guided PDF (p_g, green). These two PDF are then combined with multiple importance sampling (balance heuristic) to yield a good Monte Carlo estimator for the integral.

of light, so we do not move our discovered samples over time. Also resampling means discarding some information from the previous iteration which did not work well for light transport. However, we do employ a similar resampling step for dynamic scenes.

Population Monte Carlo (PMC) [Cappé et al. 2004] and adaptive importance sampling [Cappé et al. 2008] also derive parameters for a PDF from previous samples. PMC used to be most effective in 2D and has been used to determine various 2D kernels for instance for lens perturbations [Fan 2006] and energy redistribution path tracing [Lai et al. 2007]. Recent work extends it to higher dimensional cases, e.g. particle MCMC [Andrieu et al. 2010].

Previous work suggests that light transport lends itself well for a decomposition into 2D subdomains (one per bounce), e.g. as done with padded replications for low discrepancy sampling [Kollig and Keller 2002a,b]. However, maintaining the correlation between bounces as dictated, for instance, by non-diffuse BSDFs is not possible. Thus we cannot use a padded 2D replication of particle filters.

Learning-based methods. Despite elaborate sampling techniques, learning-based approaches can improve the simulation by using information from previously sampled paths and guiding MC samples to regions where more samples are required. Care has to be taken to maintain the unbiasedness of an estimator, even when working in image space only [Kirk and Arvo 1991]. Guiding can also be performed by learning the 2D incoming light field at cache records distributed throughout the scene [Vorba et al. 2014], and has been combined with reinforcement learning [Dahm and Keller 2017a,b]. The work of Hey and Purgathofer [2002] uses kernels around guide samples to reconstruct a signal and is in that sense similar to our approach, but only for a 2D hemispherical signal.

Note that all of these approaches only consider a 2D slice of the path space at once and thus need to explicitly differentiate modes at each cache point. Even including BSDF information requires costly processing [Herholz et al. 2016]. Instead of clustering, we store complete paths and by this keep the entire information present during path sampling as well as the dimensions of the path vertices separated. This allows us to fit simpler unimodal distributions to the local surroundings of a path sample, and include correlation between the bounces such as information about the path length.

Spatial subdivision schemes for adaptive sampling, e.g. [Hachisuka et al. 2008; Müller et al. 2017], are appealing because of their simplicity, but are intractable in higher dimensions. Subdividing every dimension only once in a 32-dimensional space would result in $2^{32} \approx 4$G bins to store. In contrast, we store a subset of full transport paths only, and define a continuous PDF around them for guiding.

3 A SELECTIVE GUIDING FRAMEWORK FOR COMPLETE PATHS

Computing the integral over a function f with a regular, unguided MC estimator often works well for some part of the integrand, but others are not well represented by the underlying unguided probability density function (unguided PDF). The goal of our work is to derive and add a new *guided PDF* to the existing one, as to make the sum of both proportional to the integrand f (Section 3.1). This is equivalent to adding a PDF in the context of multiple importance sampling [Veach and Guibas 1995], using the balance heuristic. Since data-driven guiding is expensive, we want to apply it only where needed, i.e. where existing techniques fail to converge in acceptable time. To construct this new PDF, we draw samples from the unguided PDF, and record samples which cause high variance in a cache along with their sample weight, f divided by the sampling density, as *guide samples* (Section 3.2). We use complete transport paths as samples, to retain as much information as possible. From these, we reconstruct a continuous PDF using a Gaussian mixture model. To sample from this PDF, we randomly pick one guide sample by its weight, and then sample from a Gaussian distribution centered around it (Section 3.3). The Gaussian kernel's size depends on the neighbouring guide samples (Section 3.4). Learning the guided PDF is sped up by running this process iteratively on small sample batches, also sampling from the guided PDF learned so far.

3.1 Overview

In Figure 2, an example integrand $f(\mathbf{X})$ is shown in blue, and the existing MC estimator draws samples from the PDF shown in orange, which we denote as the *unguided PDF* $p_u(\mathbf{X})$. More formally, we are approximating the integral using the estimator:

$$\langle I(\mathbf{X})\rangle_u = \frac{f(\mathbf{X})}{p_u(\mathbf{X})}. \tag{5}$$

In our example, the PDF p_u represents the left smooth mode of the integrand well, but misses the other features (two more modes). Thus we construct the additional *guided PDF* $p_g^i(\mathbf{X})$ (shown in green) which is iteratively refined for $i = 0, 1, \ldots$, and captures the differences between the integrand $f(\mathbf{X})$ (blue) and the unguided PDF $p_u(\mathbf{X})$ (orange). The converged guided PDF is shown in Figure 2,

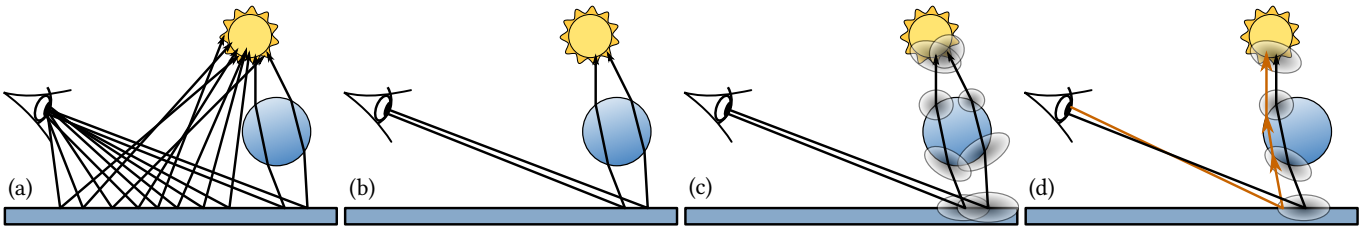

Fig. 3. Schematic overview of the guiding method employing full paths. We start with all path traced trajectories, depicted in (a). From these, we select the ones that are most poorly sampled by the underlying Monte Carlo estimator, shown in (b). For each of these guide paths, we compute anisotropic reconstruction kernels at every path vertex, to fill in the gaps between the samples. This is visualised in (c). To sample from this structure, we first pick a guide path and then successively sample path vertices according to the reconstruction kernels (d). Evaluating the PDF requires to sum up the PDF associated with all guide paths that may create the given path.

right. Eventually, we want to combine these two PDFs using multiple importance sampling (MIS). More precisely, we will be using a single-sample model with the balance heuristic [Veach and Guibas 1995] resulting in a combined estimator with a mixing weight u, i.e.

$$\langle I(\mathbf{X})\rangle_g^i = \frac{f(\mathbf{X})}{u \cdot p_u(\mathbf{X}) + (1-u) \cdot p_g^i(\mathbf{X})}. \tag{6}$$

The guided PDF will be constructed incrementally and with every iteration i we adjust $p_g^i(\mathbf{X})$ to further reduce the variance of Equation (6).

Intuitively, we proceed as illustrated in Figure 3. We begin by letting $p_g^0(\mathbf{X}) \equiv 0$ and effectively sampling $\mathbf{X}$ using only $p_u(\mathbf{X})$ (Figure 3a). From the paths $\mathbf{X}$, we select only a few new guide paths $\mathbf{Y}$ motivated by importance sampling: we first determine whether a path is an outlier causing high variance (via *density-based outlier rejection [DeCoro et al. 2010]*) and only from these we pick the N with the highest contribution to the estimate in Equation (6). We will iteratively add batches of guide paths $\mathbf{Y}_j$ sampled from both p_u and the current guided pdf p_g^i to the cache, and once they are added we keep them unchanged until the end. This is illustrated in Figure 3b and detailed in Section 3.2.

The guide paths $\mathbf{Y}_j$ are turned into a continuous function using a Gaussian reconstruction kernel (Figure 3c). We use a high-dimensional neighbour search to determine large enough reconstruction kernels to close the gaps between paths and to achieve smooth coverage. One challenge is that the PDF evaluation can become slow in Gaussian mixture models. Therefore we use truncated Gaussian kernels to be able to cull away samples efficiently. Our Gaussians are truncated at $\approx 3.330\sigma$, see Section 7 in the supplemental document for details.

To sample from the cache, we first select a guide path $\mathbf{Y}_j$ using a *cumulative density function* (CDF) built on path weights w_j which are updated every iteration i to reflect the new guided PDF $p_g^i(\mathbf{X})$ (see Section 3.4). Then we sample a new path vertex $\mathbf{x}_v$ following the Gaussian reconstruction kernel $(\Sigma, \mu)_v$ around every guide path vertex $\mathbf{y}_v$ (Figure 3d), starting at the sensor (see Section 3.3).

In the subsequent iteration $i+1$, we draw samples from both $p_u(\mathbf{X})$ and $p_g^i(\mathbf{X})$, with probabilities u and $1-u$, respectively. No matter which technique was used, new samples will be considered to be recorded as new guide samples for the next iteration, depending on their contribution $\langle I(\mathbf{X})\rangle_g^i$.

There are concerns whether an estimator is still unbiased when changing weights and kernel parameters of a PDF based on particles adaptively [Douc et al. 2007; Kirk and Arvo 1991]. In our case we will use multiple importance sampling (MIS) with an unguided estimator, which guarantees an unbiased combination. For that we require $p_u(\mathbf{X})$ to form a complete unbiased estimator, in particular $\forall \mathbf{X} : f(\mathbf{X}) > 0 \Rightarrow p_u(\mathbf{X}) > 0$. This will not be the case for our guided PDFs $p_g^i(\mathbf{X})$ which may be zero in areas where the original estimator is deemed sufficiently good already. We chose the balance heuristic to combine the PDFs as it is close to optimal [Veach and Guibas 1995]; note that it is not relevant for the final sampling quality, as the convex combination of both PDFs $p_u(\mathbf{X})$ and $p_g^i(\mathbf{X})$ adapts to be proportional to $f(\mathbf{X})$.
Learning and sampling from $p_g^i(\mathbf{X})$ has several key advantages:

- The algorithm leverages both global stratification via regular (quasi-)Monte Carlo sampling and local exploration due to the Gaussian.
- Sampling is fast, since only one guide sample needs to be chosen and the corresponding Gaussian needs to be sampled. We only need a careful implementation of the PDF evaluation for good performance.
- It focuses on parts of the integration domain where unguided sampling performs poorly.
- Compared to MCMC where step sizes are often guesswork, we use neighbouring samples to derive sampling spread. This is more robust than using analytical derivatives which can fail, e.g. due to fine displacements and visibility.
- Guide path caches can easily be reused across frames in an animation (Section 3.5).

3.2 Selecting new guide paths

In every iteration i we adapt the guiding cache to the residual variance present in the current estimator $\langle I\rangle_g^i$ (Equation (6)). The cache stores full transport paths consisting of one point on the sensor, one point on an emitter, and a list of vertices in between.

We generate a batch of such path samples from both the unguided and guided estimator (e.g. one sample per pixel), and select paths that are causing the highest variance in the current combined estimator $\langle I\rangle_g^i$. These samples need to have a high contribution in order

to cause a big difference to the expected value. High contribution can be caused by high measurement as well as low PDF. The former does not necessarily cause high variance if sampled well (e.g. looking directly into the sun). We therefore use *density-based outlier rejection* (DBOR) [DeCoro et al. 2010; Zirr et al. 2018] to distinguish between high sample contributions caused by bright measurement and those caused by low PDF value. From the latter, we keep the N paths with the highest contribution as new guide paths. This avoids selecting new guide paths in regions that have already been sampled well in previous iterations. For implementation details regarding DBOR, please see Section 3 in the supplemental document. As sample weight associated with each selected path $\mathbf{Y}_j$, we store the value of the estimator Equation (6) (Section 3.4 details how the weights are computed).

We chose to add only N (in the order of several hundred) new samples in every iteration, as in our experiments this led to improved learning performance; an observation supported by previous work [Chopin et al. 2011]. We add unguided as well as guided samples to the cache since both have the ability to discover new features: one globally, the other locally. In contrast to particle filters, once a path is added to the guiding cache, it is never removed or resampled, since in our experiments this led to loss of information, oscillation, and clumping of guide paths (cf. Section 3.4, Section 6). Only the associated weight and kernel parameters change over time.

3.3 Sampling from the cache

To sample from $p_g(\mathbf{X})$, we first select a guide path $\mathbf{Y}$ by sampling the CDF built on the path weights w_j. Having selected a guide path representing the difference between the unguided PDF $p_u(\mathbf{X})$ and the measurement contribution function $f(\mathbf{X})$, we need to reconstruct a continuous PDF $p_g(\mathbf{X})$ around it. This is accomplished with a reconstruction kernel: at every high-dimensional control point (a guide path) we place a truncated Gaussian with a covariance matrix of the same dimensionality adapted to a number of nearest neighbours. Since the measurement contribution function is a separable product of terms and since we will be sampling full paths incrementally by constructing one vertex at a time, we assume that this high-dimensional covariance matrix has a block-tridiagonal structure. Thus we only need to compute a sparse set of coefficients relating up to three consecutive path vertices to each other. Please see Section 6 in the supplemental document for an expanded explanation.

We create path vertices $\mathbf{x}_v$ one by one, starting at the camera: First, we sample $\mathbf{x}_0$ on the camera lens as for regular path tracing. Then, successively for every vertex $v > 0$, we compute the parameters Σ_v, μ_v of a truncated Gaussian distribution $\mathcal{N}[\Sigma_v, \mu_v]$ from which we will sample $\mathbf{x}_v$.

For volumes, we sample a 3D truncated Gaussian $\mathcal{N}[\Sigma_v, \mu_v]$ which is already in vertex area measure. For surface points $\mathbf{x}_v$, we first sample $\mathbf{x}'$ according to the 2D Gaussian in the frame of the next guide vertex, and then trace a ray to project $\mathbf{x}'$ to the surface geometry which yields $\mathbf{x}_v$ (see Figure 4). This step incurs a Jacobian determinant to account for the change of measure:

$$p_g(\mathbf{x}_v|\mathbf{x}_{v-1}) = \mathcal{N}[\Sigma_v, \mu_v](\mathbf{x}') \cdot G(\mathbf{x}_{v-1}, \mathbf{x}_v)/G(\mathbf{x}_{v-1}, \mathbf{x}'). \quad (7)$$

Obviously (near-)specular BSDFs will pose a problem if outgoing directions are chosen this way: the measurement contribution will likely evaluate to (near-)zero. To cover this case, we mix in outgoing directions sampled from the BSDF as well. We stochastically select between BSDF and Gaussian sampling depending on the Beckmann equivalent roughness of the surface material of the current vertex $\mathbf{x}_{v-1}$. The mode of interaction (reflect or transmit) is constrained to be the same as the guide path's. This case can also be combined with volume sampling: we choose the 2D direction by BSDF sampling, and sample the distance by a 1D Gaussian, by taking the conditional of the 3D Gaussian for this fixed direction (see Figure 4c).

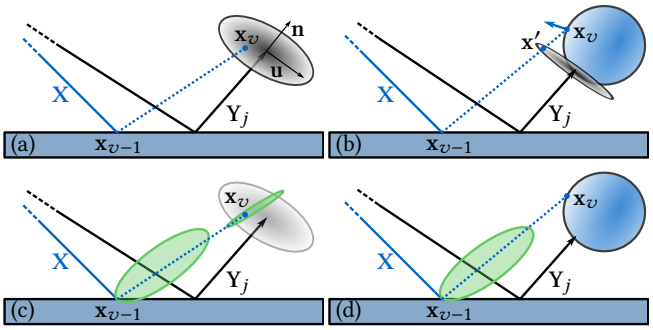

Fig. 4. Sampling the next vertex $\mathbf{x}_v$ in volumes (a,c) and on surfaces (b,d). The Gaussian distributions are defined in a frame ($\mathbf{u}$, $\mathbf{v}$, $\mathbf{n}$) at the guide path vertex, where $\mathbf{n}$ is aligned with the incident path segment. In the surface case (b), the sampled point $\mathbf{x}'$ needs to be projected to the surface geometry. For highly specular BSDF (c,d), we sample from the BSDF instead of from the guide path Gaussian. The volume case (c) then only samples the distance from a 1D Gaussian which is the 3D volume Gaussian conditioned to the sampled ray.

To evaluate the guided PDF $p_g(\mathbf{X})$ for a given path $\mathbf{X}$, we need to sum up the PDF of sampling $\mathbf{X}$ given any of the guide paths $\mathbf{Y}_j$, i.e.

$$p_g(\mathbf{X}) = \sum_j w_j \cdot p_g(\mathbf{X}|\mathbf{Y}_j) \;/\; \sum_j w_j, \quad (8)$$

$$p_g(\mathbf{X}|\mathbf{Y}_j) = p(\mathbf{x}_0) \cdot \prod_{v=1}^{k-1} r \cdot p_g(\mathbf{x}_v|\mathbf{x}_{v-1}) + (1-r) \cdot p_s(\mathbf{x}_v|\mathbf{x}_{v-1}),$$

where we account for the Gaussians $p_g(\mathbf{x}_v|\mathbf{x}_{v-1})$ from Equation (7), the guide path weights w_j as well as for the PDF of BSDF sampling p_s. The weight r is computed from the surface roughness of the vertex $\mathbf{x}_{v-1}$ or the mean cosine in case of volume scattering. Additionally, $p(\mathbf{x}_0)$ includes the PDFs to choose a wavelength, time, and aperture position on the camera.

3.4 Determining reconstruction kernel parameters

It remains to compute the Gaussian parameters Σ and μ, as well as the path weights w_j.

Reconstruction kernel size. The kernels should be wide enough to close the gaps between the guide paths in the path space. Otherwise the guided PDF will leave a part of the difficult sampling domain to the unguided PDF $p_u(\mathbf{X})$ (see Figure 6). This would still result in an unbiased estimator, but may lead to outliers generated by $p_u(\mathbf{X})$. Thus we employ a high-dimensional nearest neighbour search on all guide paths $\mathbf{Y}_j$ of same configuration, i.e. of the same length k

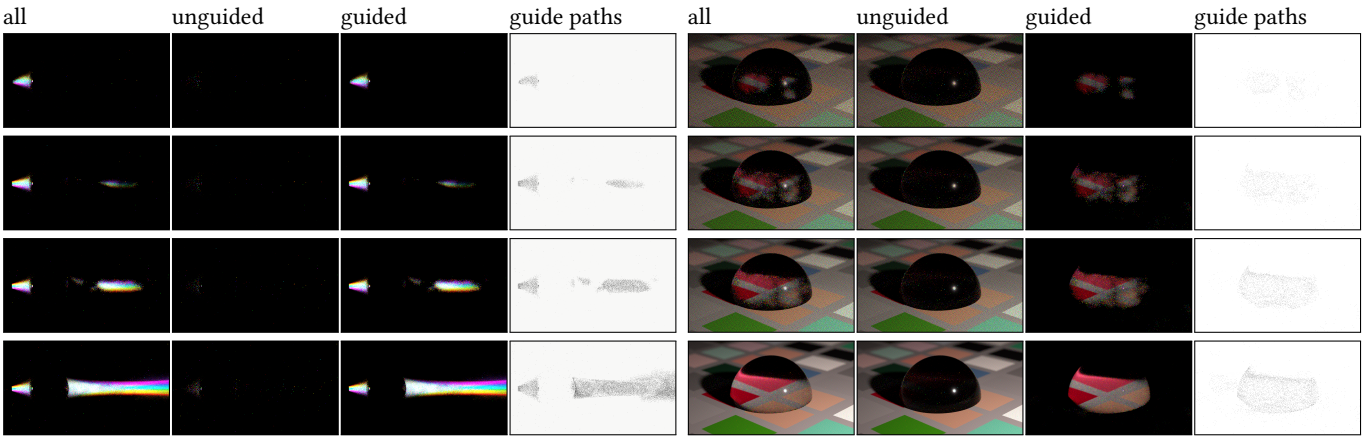

Fig. 5. Left half: a volumetric caustic cast by a coloured spot light through a specular dielectric sphere. Right half: a glass sphere on a plane. The rows represent a selection of the consecutive learning progressions from $i = 2$ to $i = 29$. The renders are broken down into contributions from the guided and unguided samplers, as indicated on the columns. While in the volume caustic, virtually all paths require guiding for efficient sampling, the diffuse ground plane in the right part is robustly separated out by our approach. Additionally, we show the guide path locations as black dots to visualise both the even spacing and the local learning. For illustration, we used a constant 10-pixel wide Gaussian for the first vertex in these images.

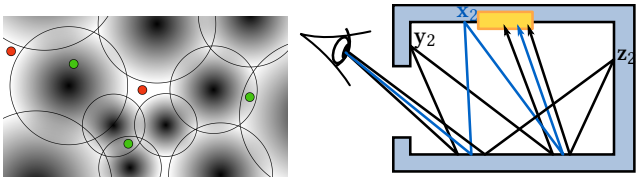

Fig. 6. Left: A 2D illustration of the guided PDF (black Gaussians). If the unguided sampler creates high variance samples inside the support of the guided PDF (green) their contribution will be effectively reduced by MIS. If the high variance samples lie outside of the support of the guided PDF (red) these samples will be added to the cache in the learning phase. However, during rendering the cache remains fixed and these outliers contribute fully to the image. Right: A diffuse path $\mathbf{X}$ with 5 vertices, sampled in between two guide paths $\mathbf{Y}$ and $\mathbf{Z}$. Their covariance matrices Σ_2 around vertex $v = 2$ need to be very large to define an overlap with $\mathbf{x}_2$. If they are too small, the blue path $\mathbf{X}$ cannot be constructed by either of these guide paths, even though they are very close in all other dimensions. This means the gap will be filled by the unguided sampler (potentially with higher variance).

and succession of reflect, transmit, hair fiber, or volume scattering events. As distance metric, we use the sum of the squared Euclidean distances of the path vertices:

$$d(\mathbf{Y}, \mathbf{Z}) = \sum_{v=0}^{k-1} \|\mathbf{y}_v - \mathbf{z}_v\|^2. \tag{9}$$

For every guide path we collect a small number of neighbouring paths (we are using 10, but any number between 4 to 20 yielded similar results). To derive the Gaussian to sample the respective next path vertex $\mathbf{x}_v$ given vertex $\mathbf{x}_{v-1}$, we first compute a covariance matrix $\bar{\Sigma}_v$ and the corresponding mean $\bar{\mu}_v$:

$$\bar{\Sigma}_v = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}, \quad \bar{\mu}_v = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \tag{10}$$

where the block Σ_{11} expresses how the v-th vertices of the neighbour paths are distributed around their mean μ_1, and $\Sigma_{11} \in \mathbb{R}^{2\times2}$ if $\mathbf{x}_v$ is on a surface, or $\Sigma_{11} \in \mathbb{R}^{3\times3}$ if $\mathbf{x}_v$ is in a volume. Σ_{22} is the same for the previous vertex $\mathbf{x}_{v-1}$, and the dimensionality of the off-diagonal blocks follows accordingly. This means that $\bar{\Sigma}_v$ can be 4×4, 5×5 or 6×6-dimensional and captures how the v-th path vertex behaves with respect to the $(v-1)$-th path vertex. We compute the covariance matrix in an orthonormal frame for every path vertex, where one basis vector is aligned with the incoming ray direction $\mathbf{y}_v - \mathbf{y}_{v-1}$ of the guide path.

To derive an accurate sampling distribution, we compute the conditional $\bar{\Sigma}_v|\mathbf{x}_{v-1}$ to obtain a truncated Gaussian distribution $\mathcal{N}[\bar{\Sigma}_v, \bar{\mu}_v]$ from which we will sample $\mathbf{x}_v$:

$$\Sigma_v = \bar{\Sigma}_v|\mathbf{x}_{v-1} = \Sigma_{11} - \Sigma_{12} \cdot \Sigma_{22}^{-1} \cdot \Sigma_{21}, \tag{11}$$

$$\mu_v = \bar{\mu}_v|\mathbf{x}_{v-1} = \mu_1 + \Sigma_{12} \cdot \Sigma_{22}^{-1}(\mathbf{x}_{v-1} - \mu_2). \tag{12}$$

Since Equation (11) does not depend on $\mathbf{x}_{v-1}$, we precompute and store Σ_v, $(\Sigma_{12} \cdot \Sigma_{22}^{-1})$, μ_1, and μ_2 at the guide path vertex $\mathbf{y}_{v-1}$: We perform a singular value decomposition of Σ_v, regularise the singular values, and keep the rotation matrix and singular values suitable to transform an isotropic Gaussian random variable.

The conditional in Equation (11) results in small Gaussians, facilitating our goal to yield good sampling efficiency such that, for instance, the sampled rays do actually intersect the light source they aim for (see Figure 7).

The size of the Gaussian for the first vertex $\mathbf{y}_1$ is not based on the neighbouring paths. Instead, we use a simple radius shrinking scheme similar to progressive photon mapping [Knaus and Zwicker 2011] which is based on a ray differential footprint. We successively shrink the size of the Gaussian at $\mathbf{y}_1$ which allows us to control screen space stratification, and ensures that the first Gaussians are not overlapping excessively, making PDF evaluation faster.

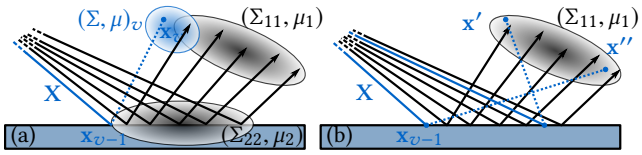

Fig. 7. For most accurate sampling of $\mathbf{x}_v$, we compute a covariance matrix (5×5 in this example, since $\mathbf{x}_{v-1}$ is a surface point and $\mathbf{x}_v$ is in a volume). One block in the matrix, Σ_{11}, expresses how $\mathbf{x}_v$ should be distributed according to the collected data from the guide paths (shown in black). Because we already have a fixed previous vertex $\mathbf{x}_{v-1}$, we derive the conditional of the 5×5 Gaussian accordingly, resulting in a new distribution $\mathcal{N}[\Sigma_v, \mu_v]$ (shown in blue). If there is a non-zero covariance between the coordinates of the guide path vertices $\mathbf{y}_v$ and $\mathbf{y}_{v-1}$ the conditional Gaussian Σ_v will typically be a lot more focused than Σ_{11}. In this example, disregarding the covariance would amount to sampling according to Σ_{11} with much larger spread, as illustrated in (b). Using Σ_v results in higher quality samples which are more likely to be valid paths, as well as faster PDF evaluation due to culling.

An illustration of the progressive sample placement in screen space can be seen in Figure 5, showcasing the separation into the contribution from the guided and the unguided part of our technique as well as global and local exploration. In Section 2 of the supplemental document we show a visualisation of a guide path, its neighbouring guide paths from the cache and some sampled paths for the same scenes.

Guide path weights. In contrast to particle filtering methods, we do not employ a resampling step in the learning phase, and we never remove guide paths. Our experiments confirmed that in regions where only very scarce information can be obtained from the unguided sampler, discarding any path can be a substantial loss. Also, learning performs much faster if important paths attract more samples. Thus we aim to keep all guide paths and reevaluate the weights w_j once new paths are added. If the density of paths becomes higher, the weight should become smaller. For importance sampling of guide paths, we should select them by their contribution to the estimator in Equation (6):

$$w_j = \frac{\langle I(\mathbf{Y}_j)\rangle_g^i}{\sum_l \langle I(\mathbf{Y}_l)\rangle_g^i}. \tag{13}$$

In every new iteration i, we need to recompute these weights for all paths in the cache. Unfortunately, in iteration i Equation (6) depends on the guided PDF $p_g^i(\mathbf{Y}_j)$ which in turn depends on updated weights w_j based on paths sampled in iterations $[0, i]$. This leads to a circular dependency between Equation (8) and Equation (13).

As a solution, we propose to use the PDF of the previous iteration $p_g^{i-1}(\mathbf{Y}_j)$ instead. To avoid oscillation, we take an exponential average of previous weights:

$$w_j^i = t \cdot \frac{\langle I(\mathbf{Y}_j)\rangle_g^{i-1}}{\sum_l \langle I(\mathbf{Y}_l)\rangle_g^{i-1}} + (1-t) \cdot w_j^{i-1}, \tag{14}$$

where we choose $t = 0.5$ for paths that were previously contained in the cache, and $t = 1$ for new additions. Note that since t depends on j, these weights do not sum to one any more, so they have to be explicitly normalised in Equation (8).

3.5 Extension to dynamic scenes

To explicitly improve temporal stability, we employ a resampling step which is in direct correspondence to the resampling step in particle filters for target tracking. When rendering a new frame of an animation, we keep the guide path cache of the previous frame and use it as an initial guided PDF to sample a few paths from. Since the difference between frames is often small due to temporal coherence, this step often yields good samples and makes sure that effects that have been discovered previously have a good chance to be discovered in the subsequent frame, too. After this initial seeding step, the old guide paths are discarded and the learning phase continues as usual on the resampled guide paths. Please see the accompanying video for a visual evaluation, and Section 5 in the supplemental document for more details about how we resample the guide path cache of the previous frame and the relation to classic particle filter methods as they are used for target tracking.

4 IMPLEMENTATION DETAILS

We implemented our method in a custom spectral rendering system. In our evaluation we found 10k-50k guide paths to be sufficient in most cases and therefore we did not optimise our implementation for run time speed or memory.

We store the guide paths and their vertices in two separate linear buffers. A path vertex needs 128 bytes of memory: 112 bytes for sampling information including the eigenvalue decomposition, 12 bytes for the world space position and the remaining 4 bytes are used for flags (e.g. volume, surface, transmit, reflect) and shader information. For each path we store its wavelength, weight, measurement contribution, the individual PDF values p_u and p_g, the index of the first path vertex in the vertex buffer, as well as some additional debugging information. Storing 50k guide paths therefore requires about 150MB of memory.

4.1 PDF evaluation

To evaluate the guided PDF $p_g(\mathbf{X})$ in Equation (8) for a given path $\mathbf{X}$, we have to identify all guide paths that could have sampled $\mathbf{X}$. To reduce the computation, we employ a variety of culling approaches: first, we use a BVH to find only guide paths $\mathbf{Y}_j$ with overlapping truncated Gaussians for $\mathbf{x}_1$ because the PDF conditional on other guide paths would evaluate to zero. Each vertex has a corresponding bounding box that encloses its truncated sampling Gaussian. For surfaces, we determine all potential guide paths by intersecting this structure with the camera ray, enumerating all intersecting 2D Gaussians. For volumes, we can perform a simpler point query to find all overlapping 3D Gaussians. Afterwards, we do a quick culling step based on the path configurations (path length and reflect/transmit/volume scattering sequence have to match). Additionally, we cull a guide path as soon as one Gaussian evaluates to zero. Since the Gaussians are designed to cover a small number of neighbours, we can expect the culling to be effective.

4.2 Covariance matrices

To compute the covariance matrices (cf. Section 3.4) for a guide path, we search for similar guide paths using a high-dimensional nearest neighbour search. Since the number of guide paths is usually low

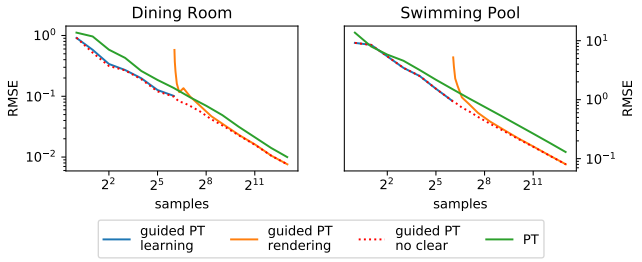

Fig. 8. Convergence plots for the DiningRoom and Pool scene illustrating that asymptotically our proposed method behaves like regular Monte Carlo, but with lower error than plain path tracing. The orange line depicts the error if the learning phase is explicitly terminated after a while and the frame buffer is cleared. The guided PDF retained in the path cache makes sure the convergence quickly recovers. The dashed line indicates the error if the frame buffer is not cleared after learning stopped at 2^6 samples.

(10–100k) and culling using the same BVH as for the PDF calculation is very effective, we employ a simple brute force approach and compute the pairwise distances between the guide paths according to Equation (9). We realise that this is very inefficient asymptotically, but it was not a bottleneck in our experiments. If the need for more guide paths arises, an appropriate acceleration structure for the nearest neighbour search will be needed.

The eigenvalue decomposition of the covariance matrices Σ_v can result in negative eigenvalues due to numerical problems. Therefore we clamp them from below to the pixel footprint at the vertex $\mathbf{x}_1$ of a path. This also counteracts sub pixel clumping of samples.

4.3 Learning mode/Rendering mode

Our method can be used fully progressively, and similar to Müller et al. [2017] it does not strictly require a dedicated learning phase. Figure 8 illustrates the error behaviour (without DBOR) with sample count if we explicitly split the algorithm into learning and rendering phase versus if we just keep rendering. The main difference is that clearing the frame buffer after learning leads to a fresh start without any potentially accumulated outlier samples. If DBOR is used to remove any outliers from the learning phase, we conclude that clearing the frame buffer after learning is not necessary.

Stopping the learning process. For time-limited renders, it can be an advantage to stop the learning process at some point, since excessive learning can fill the cache with a large number of paths, leading to slow PDF evaluation. It is difficult to define a criterion to decide when the learning phase was sufficiently long: theoretically we can never be sure to have explored all important features in the path space since we have to rely on the unguided sampler to find them (which can take arbitrarily long). We found that allotting 10% of the total run time to the learning phase, and the rest accordingly for the rendering phase, works well in our test scenes. We have encountered cases where the DBOR guide path selection detects that all effects are already sufficiently explored and does not admit any more guide paths into the cache. In this case, the algorithm transparently stops recomputing the guide path cache. There are, however, fail cases to this, for instance in scenes where all transport is equally difficult. This leads to an indefinitely growing cache. We would like to explore more robust criteria in the future.

Remaining high variance samples. Stopping the learning process means that there can be regions in the path space that have not been explored yet. If these are sampled during the rendering phase by the unguided sampler, its high variance samples possibly contribute fully to the image as bright outliers. Resolving this is possible by an extended learning phase or by outlier removal. In Figure 12 we show that removing the remaining outliers discards significantly less energy than using unguided sampling only and removing all outliers. We perform the outlier removal with DBOR, which we already use for path classification during the learning phase. Please see Sections 3 and 4 in the supplemental document for details.

4.4 Parameters

Our implementation has parameters that we want to discuss in the following. First, we usually choose the unguided mixing weight from Equation (6) as $u = 0.5$. We ran experiments adjusting it such that the observed variance in the guided and unguided contributions to the image are of equal magnitude. This led to results very close to 0.5, but may be worth reconsidering in the future.

Second, we need to set the number of samples added to the cache in every iteration. We found that using a budget of 0.1–1% of the total number of generated samples per iteration works well in our tests. Since DBOR can be unreliable in the beginning of the learning phase, we use a conservative maximum of 0.2%. We use one heap per thread to efficiently manage the list of highest-ranked paths.

Density-based outlier rejection. As mentioned in Section 3.2, we use density-based outlier rejection [DeCoro et al. 2010; Zirr et al. 2018] during the learning phase to distinguish between high-contribution samples which are frequent enough to not cause high variance, and those which are important as guide paths. The corresponding frequency threshold has an impact on learning: lowering the threshold increases the chances to add irrelevant samples to the guiding cache which the unguided sampler would have been able to handle. This can decrease the efficiency of the guided sampler. On the other hand, increasing the threshold reduces the learning speed, as only very rare samples are added to the cache, and consequently more unguided samples are needed.

Progressive Gaussian shrinking. As mentioned before, we use a progressive radius shrinking scheme based on a ray differential footprint for the size of the Gaussians at the first vertex. We start by a screen space radius of 50 pixels and progressively shrink it during the learning phase depending on the scene. Starting with a large radius is important for discovery in the beginning of the learning phase and allows to share guide paths between pixels. The radius at the end of shrinking has two important impacts. First, a smaller radius results in less overlapping Gaussians and therefore more effective culling and faster PDF evaluation. Second, a smaller radius results in larger cache sizes because guide paths can only be shared between fewer pixels which increases PDF evaluation time. Choosing an optimal minimal radius automatically is challenging and remains future work. Right now, it is a user parameter that is

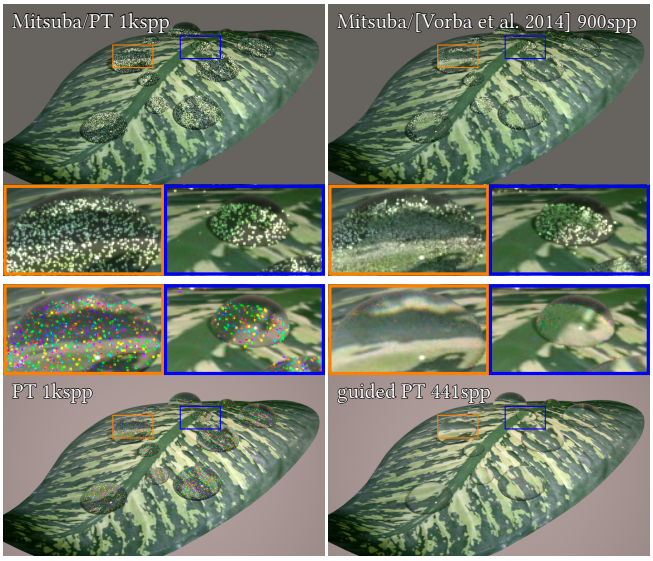

Fig. 9. Roughly equal time comparison to 2D marginalised guide records [Vorba et al. 2014]. The top row has been rendered in Mitsuba, and the bottom row in our renderer (on different machines with different operating systems, one RGB and one spectral). We normalised run time to 1024spp for next event estimation (left). The scene is lit by a key light to the right and a constant environment map. Here, 2D records should work just fine because the problem is essentially 2D. Since our method focuses its efforts on difficult regions (the water droplets), it performs much better using only 950 guide paths (bottom right) even compared to Vorba et al.'s method (top right) using ≈700k guide records, which has to learn the whole scene.

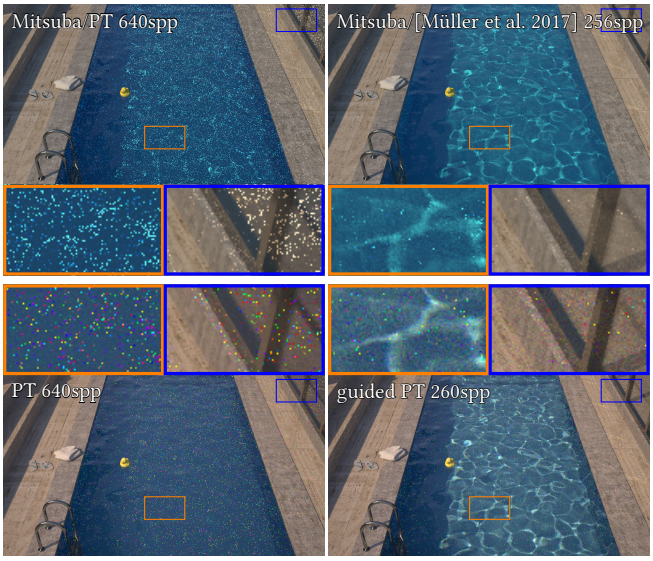

Fig. 10. Roughly equal time comparison to 2D marginalised guide records [Müller et al. 2017]. The top row has been rendered in Mitsuba, and the bottom row in our renderer (on the same machine, one RGB and one spectral). We normalised run time to 640spp/5 min for next event estimation (left). In this case the difficult parts of transport are spread all over the scene, so our method cannot show its full strength by focusing efforts to small areas in screen space. Factoring in the colour noise from spectral rendering, it yields comparable benefits over path tracing.

chosen between 1 (which we use for the Pool and Tumbler scenes) and 10 pixels (which we use for all other scenes in this work).

5 RESULTS

Our results have been rendered on an AMD Ryzen 7 1800X with 64GB of main memory. In all our experiments with the guided sampler, half the samples are guided ($u = 0.5$). The maximum path length is 32 path vertices for all images.

5.1 Evaluation of guided sampling

We evaluated our guided sampling in a variety of scenes with different challenging lighting settings and performed equal-time comparisons between path tracing with next event estimation (PT) and guided path tracing (guided PT). The details about learn and render time, number of guide paths and error metrics compared to a path traced reference are given in Table 1.

DiningRoom. In Figure 1 we show an indoor scene illuminated by a large light outside and a small bright spotlight inside on the table. PT handles the outside illumination that dominates the overall lighting in this scenes very well. However, it struggles to resolve the caustic caused by the lamp and teapot on the table. Our algorithm robustly selects the difficult regions and is able to sample the reflected caustic effectively. The insets show the results of our guided PT with outlier removal. For PT there was no outlier removal otherwise the caustic would be missing completely.

Pool. A scene where the difficult caustic affects a much bigger portion of the image is the first scene in Figure 13. Here a swimming pool is illuminated by environment lighting containing a bright sun which causes intricate caustics on the bottom of the pool. Our guided PT handles this scene equally well and divides the illumination of the sun, which is handled by the guided sampler, from the more uniform rest of the environment illumination that is handled by the unguided sampler. In all comparisons in Figure 13 we used outlier removal with the same threshold for both PT and guided PT.

Tumbler. The second scene in Figure 13 contains a complicated caustic and many small glints on the water surface and the surface of the tumbler. This scene is challenging since it contains many direct and indirect highlights. Again the illumination is robustly divided into a high frequency component for the guided sampler and a low frequency component for the unguided sampler. After outlier removal no glints (highlights) remain in the path traced images whereas our method resolves these small bright features much better.

Dragon. The third scene Figure 13 shows a glossy dragon in slightly forward-scattering fog (mean cosine is 0.5) which is illuminated by several spotlights. The surrounding participating medium complicates the light transport simulation significantly.

We can observe that the unguided sampler computes more of the diffusive multiple scattering illumination. The guided sampler focuses on bright and sharp features such as the indirect glossy reflection of the directly illuminated part of the dragon's wing and the volumetric illumination from the spot lights. This multi-dimensional feature is potentially very hard to grasp with 2D guide records. Note that we cannot show a direct comparison to previous guiding techniques as these do not support participating media.

The bottom inset shows the volume directly in front of one of the spot lights. Usually a specialised technique, such as equi-angular sampling [Kulla and Fajardo 2011], is required to sample such regions efficiently. Our guided sampling improves the rendering quality in these regions significantly and automatically.

LivingRoom. The last scene in Figure 13 shows a living room which is illuminated from the outside with environment lighting and two lamps on the inside. The lamps are spherical emitters surrounded by a dielectric cylinder that prevents next event estimation and causes a high variance for PT. Guided PT focuses on this difficult illumination which results in less noise especially in the region of the picture above the lamps. After outlier removal, PT misses much more energy in this region compared to guided PT.

The insets in Figure 1 and Figure 13 only show versions of PT and guided PT with outlier removal. Figure 12 shows the full images of PT and guided PT with and without outlier removal for the DiningRoom scene as well as the removed outliers. Figure 7 in the supplemental document shows this comparison for all other scenes.

5.2 Comparison to related work

We compare to the previous work of Vorba et al. [2014] in Figure 9 where we render water droplets on a leaf. We used the author's source code available in the rendering framework Mitsuba. This comparison has to be taken with a grain of salt because of the different rendering environment. We perform roughly an equal time comparison (4-5 min) by normalising the performance of the two different rendering systems: we render simple path tracing with next event estimation at 1024 samples per pixel for both, and give the guiding algorithms in the corresponding frameworks the same time. What can be observed from these images is that the 2D caches struggle with separating the illumination from the constant environment map from the diffuse and very small light source present in this scene. Our guided sampling improves this situation: first, keeping complete transport paths can easily distinguish between well-separated light sources. Second, we can sample along guide paths maintaining the path configuration (reflect or transmit). Third, we detect that samples landing on a constant environment map have very little variance and can be handled by the path tracer. Thus the result is smoother with about half the number of samples per pixel while using about three orders of magnitude fewer guide records.

In Figure 10 we compare to the work of Müller et al. [2017] in the Pool scene. We used the author's source code available in the rendering framework Mitsuba. Here again, this comparison has to be taken with a grain of salt because of the different rendering environments. Considering the additional noise introduced by our spectral renderer the qualitative improvement over path tracing with next event estimation is similar for both methods.

In the supplemental document, we compare our method to BDPT, VCM, and several MCMC methods in a closeup of the DiningRoom scene, focusing on the caustic and reflected caustic.

5.3 Path length

We visualised light transport paths appearing in a volume caustic from a specular sphere in an unbounded homogeneous medium in Figure 11. In this case, guided sampling saves a lot of render time because it does not generate overly long scattering paths in uninteresting regions of the path space, as opposed to unguided sampling. This shows how the guide cache with full paths can make Russian roulette for path termination unnecessary.

6 DISCUSSION

Discarded ideas. In the course of this project, we explored various, as it turned out, wrong tracks, but we still want to share these.

Replacing Gaussians during learning by constant disk kernels or cubic b-splines strongly limits the ability of local exploration due to the short or non-existent tails of these kernels.

To speed up the search for neighbouring guide paths, we tried to use a repeated application of a lower dimensional 3D nearest neighbour search using a BVH, once per path vertex instead of a monolithic search for the whole path. In the simplistic example in Figure 6 this might work, but may leave gaps between the guide paths when these are overlaid by many more undirected diffuse paths. 3D neighbour search will underestimate the distances, since many vertices seem close together but belong to paths that are far apart in other bounces. This will result in more residual outliers caused by the unguided contributions.

We experimented with photon relaxation-inspired guide path vertex moving [Spencer and Jones 2013]. However, doing so as a padded replication of 2D/3D operations breaks the correlation between bounces and the resulting paths do not transport much energy any more. We also experimented with high-dimensional move steps (as often found in sequential Monte Carlo) and resampled guide paths by using a Metropolis-Hastings update step similar to PMCMC [Andrieu et al. 2010]. Unfortunately this led to clumping and was working against the stratification in our selection stage. These experiences suggest that expectation maximisation-based fitting of Gaussians [Jakob et al. 2011] also probably does not work well in this context.

Learning. Our experiments showed that selecting too many new guide paths in one iteration hinders the exploration: the sampler requires a few iterations to find the regions of lowest probability density between existing Gaussians for good stratification. Moreover, irrelevant samples should never be chosen as guide paths and should be handled by the unguided sampler. In interesting regions which have already been discovered, adding too many overlapping Gaussians can lead to poor performance when evaluating the PDF. All our attempts to "repair" such scenarios with a resampling step failed. Randomly removing samples from the cache often resulted in oscillation instead of convergence.

Outlier removal. Any kind of adaptive sampling (such as guiding) takes away probability mass from areas that are deemed unimportant, to favour already discovered interesting areas. This is dangerous as it can increase variance in important but undiscovered areas. To balance this effect, we focus on the surroundings of samples which cause high variance and thus high mean squared error (MSE). Consequently, the areas we take probability mass away from have only minor effect on MSE. We do not introduce worse outliers since we combine with the unguided sampler via MIS. However, in the worst case this leads to fewer samples per pixel in the same time for regions marked as unimportant, increasing variance there. In such cases simply removing outlier samples often results in lower MSE (see Figure 12). Note that this effect is found in all adaptive sampling schemes, also in previous low dimensional marginal guiding systems. This is why Vorba et al. [2014] and similar methods always apply MIS with BSDF sampling to counteract the effect. For final renders, we always use outlier removal. Since it is run as a post-process [Zirr et al. 2018], we can always reconstruct the unbiased results which are equal to the filtered result if there are no outliers. Figure 7 in the supplemental material shows results with and without outlier removal for all our test scenes, except the simple scenes in Figures 5, 9 and 10 which do not use any outlier removal.

Animations. Though not part of our original scope, we implemented a straightforward adaptation of our method with explicit inter-frame reuse of guide paths for dynamic scenes (see Section 3.5 and Section 5 in the supplemental material). This is evaluated in a supplemental video containing non-converged images to show temporal convergence characteristics. We demonstrate that reusing caches between frames substantially improves temporal stability and learning speed. We leave it for future work to thoroughly analyse this approach in a variety of scenes and optimise it for reduced flickering, fast learning, and balanced exploration of path space.

Complete paths vs. 2D marginals. We explored the possibilities of caching complete transport paths with all dimensions and their correlations (incoming lighting, BSDF, free flight distance in volumes). In cases where no such correlation exists (e.g. a diffuse Cornell box with no complex visibility which is correlated with path length), our method creates unnecessary overhead. Instead, low-dimensional caches storing 2D marginals [Vorba et al. 2014] may perform better. Also we assume that the unguided sampler works well in a significant portion of the sampling domain. In the VeachDoor scene (see Figure 6 in the supplemental material) this is not true and all transport would profit from guiding.

However, 2D marginal cache records do not consider BSDF or volumes. We are interested in storing reflectance fields, not light fields. Thus, in general the transport operator is 6D up to 9D. Guo et al. [2018] recently showed that guiding with spatial data structures can be challenging with this number of dimensions already.

We argue that storing the full paths is a well-suited data structure since it allows us to clearly distinguish between types of scattering events (reflect, transmit, volume, specular), it encodes potentially long specular chains, and it sidesteps the clustering problem of multi-modal GMM: The path space helps discern different types of incident illumination, so we can use simpler uni-modal Gaussians and thus never run out of lobes as e.g. Vorba et al. [2014].

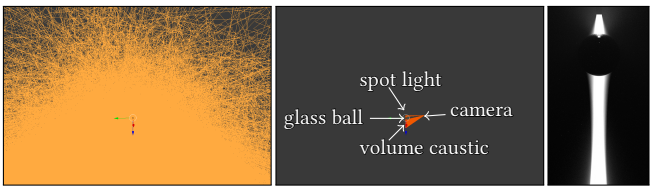

Fig. 11. Path visualisation for a volume caustic scene where a specular sphere casts a caustic from a spot light (right). The left two pictures show screen grabs from a 3D visualisation of traced paths, at same scale. Guiding by complete paths efficiently selects only the relevant paths (middle) out of the complete set of paths a path tracer (left) would trace. Most of these paths will carry very little throughput and could be culled by using Russian roulette, because energy is transported here by paths with five vertices or less. However, without adjoint information Russian roulette can increase variance substantially [Vorba and Křivánek 2016].

We showed we can afford to store important paths and we hope to inspire future work making even better use of the data (e.g. by jumping between paths using them as low-dimensional marginal distributions to improve the performance in the VeachDoor scene).

Curse of dimensionality. Usually, the remaining hard-to-sample part consists of highly directed transport and is effectively low dimensional. Areas with many undirected bounces can cause problems for our guiding, as Gaussian kernels need to become large to close gaps in the sampling domain. Fortunately, the unguided samplers typically perform well in these regions. However, we did encounter situations with very large Gaussians (and thus slow PDF evaluation), or gaps which are filled by high-variance unguided paths (as in any dense, highly scattering medium or in the VeachDoor scene).

Deriving a PDF from many neighbours. One interesting aspect of our guiding is that, in contrast to MC or MCMC, we can make data-driven decisions about the sampling spread around a sample. We found the conditional distribution derived from a 6D covariance matrix very effective in generating accurate samples from very few guide paths. In theory, the transport operator should depend on three path vertices, i.e. the maximum dimension of the covariance matrix should be 9×9. Our implementation assumes that incoming rays have about the same direction since they are coupled to the guide path, but extending this may lead to even better results. Ideally such an approach would incorporate statistics about half vectors, which may avoid the need to sample the BSDF at all. Our fallback BSDF sampling uses independent sampling. At least it would potentially be better to use smaller steps in primary sample space by first deriving the random numbers using an inverse mapping [Bitterli et al. 2017; Otsu et al. 2017; Pantaleoni 2017].

7 CONCLUSION AND FUTURE WORK

We introduced a new guided sampling technique for light transport simulation which employs complete transport paths instead of 2D marginals. This captures diverse kinds of lighting effects and their correlations, including the BSDF for glossy interactions and free flight sampling for volume scattering as well as path configuration (reflect vs. transmit and path length instead of Russian roulette).

This technique is selective, in that it is used only in the regions of the path space where an existing Monte Carlo sampler is inefficient. We automatically detect this sub domain.

This separation into highly directed/high contrast contribution and diffusive/low contrast parts could be exploited in future work to derive specialised variance reduction techniques. For instance the low contrast part could work well with aggressive denoising or gradient domain path tracing [Kettunen et al. 2015].

The method conceptually differs from Markov chain-based methods: it derives sampling of a new path from many input paths instead of just one current state. This enables us to accurately estimate the spread of the distribution. Furthermore, using complete paths as guide records allows us to use simple unimodal Gaussian distributions instead of complicated clustering, while using significantly fewer guide records than previous work.

We would like to point out that a newly proposed guide path does not strictly require a well-defined PDF. Our technique would re-evaluate a PDF based on the current state of the cache. In other words, the guide paths need not be generated by an unbiased Monte Carlo sampler of the path integral. Instead, the cache could be filled by Markov chains violating the detailed balance condition or by hand-chosen paths. The latter could be useful to explicitly steer importance to a key light that is currently being look developed. Conversely, the guide path cache can be used to visualise the hardest transport paths in a scene to reveal potential modelling issues.

This new approach comes with new challenges. We propose a first set of techniques to approach these. In particular, we propose an iterative path selection strategy akin to particle filters which finds difficult regions in the sampling domain and keeps rare trajectories in a cache. From this cache, we derive a continuous, high-dimensional PDF similar to a Gaussian mixture model. We also present a sampling technique which creates full path samples from a set of neighbouring guide paths. We expect each of these steps to improve in the future, such as faster PDF evaluation, or more accurate sampling from fewer guide paths.

REFERENCES

Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. 2010. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society* 72, 3 (2010), 269–342.

Benedikt Bitterli, Wenzel Jakob, Jan Novák, and Wojciech Jarosz. 2017. Reversible Jump Metropolis Light Transport Using Inverse Mappings. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 37, 1, Article 1 (2017), 12 pages.

O. Cappé, R. Douc, A. Guillin, J.-M. Marin, and C. P. Robert. 2008. Adaptive Importance Sampling in General Mixture Classes. *Statistics and Computing* 18, 4 (2008), 447–559.

O. Cappé, A. Guillin, J.-M. Marin, and C. Robert. 2004. Population Monte Carlo. *Journal of Computational and Graphical Statistics* 13, 4 (2004), 907–929.

Subrahmanyan Chandrasekar. 1960. *Radiative Transfer.* Dover Publications Inc. ISBN 0-486-60590-6.

N. Chopin, P. E. Jacob, and O. Papaspiliopoulos. 2011. SMC^2: an efficient algorithm for sequential analysis of state-space models. *ArXiv e-prints* (Jan. 2011). arXiv:stat.CO/1101.1528

David Cline, Justin Talbot, and Parris K. Egbert. 2005. Energy Redistribution Path Tracing. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 24, 3 (2005), 1186–1195.

K. Dahm and A. Keller. 2017a. Learning Light Transport the Reinforced Way. *ArXiv e-prints* (Jan. 2017). arXiv:cs.LG/1701.07403

K. Dahm and A. Keller. 2017b. Machine Learning and Integral Equations. *ArXiv e-prints* (Dec. 2017). arXiv:cs.LG/1712.06115

Christopher DeCoro, Tim Weyrich, and Szymon Rusinkiewicz. 2010. Density-based Outlier Rejection in Monte Carlo Rendering. *Computer Graphics Forum (Proc. Pacific Graphics)* 29, 7 (Sept. 2010), 2119–2125.

Pierre Del Moral. 1996. Non Linear Filtering: Interacting Particle Solution. *Markov Processes and Related Fields* 2, 4 (1996), 555–580.

R. Douc, A. Guillin, J. . Marin, and C. P. Robert. 2007. Convergence of adaptive mixtures of importance sampling schemes. *ArXiv e-prints* (Aug. 2007). arXiv:math.ST/0708.0711

Shaohua Fan. 2006. *Sequential Monte Carlo methods for physically based rendering.* Ph.D. Dissertation.

Luca Fascione, Johannes Hanika, Marcos Fajardo, Per Christensen, Brent Burley, and Brian Green. 2017. Path Tracing in Production – Part 1: Writing Production Renderers. In *SIGGRAPH Courses.*

Matteo Fasiolo, Flávio Eler de Melo, and Simon Maskell. 2018. Langevin incremental mixture importance sampling. *Statistics and Computing* 28, 3 (01 May 2018), 549–561.

Iliyan Georgiev, Jaroslav Křivánek, Toshiya Hachisuka, Derek Nowrouzezahrai, and Wojciech Jarosz. 2013. Joint Importance Sampling of Low-Order Volumetric Scattering. *ACM Trans. on Graphics* 32, 6 (2013), 164.

Iliyan Georgiev, Jaroslav Křivánek, Tomáš Davidovič, and Philipp Slusallek. 2012. Light Transport Simulation with Vertex Connection and Merging. *ACM Trans. on Graphics (Proc. SIGGRAPH Asia)* 31, 6 (2012), 192:1–192:10.

Walter Gilks and Carlo Berzuini. 2001. Following a moving target – Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society* 63, 1 (2001), 127–146.

Adrien Gruson, Mickaël Ribardière, Martin Šik, Jiří Vorba, Rémi Cozot, Kadi Bouatouch, and Jaroslav Křivánek. 2016. A Spatial Target Function for Metropolis Photon Tracing. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 36, 1 (Nov. 2016), 4:1–4:13.

Jerry Jinfeng Guo, Pablo Bauszat, Jacco Bikker, and Elmar Eisemann. 2018. Primary Sample Space Path Guiding. In *Eurographics Symposium on Rendering - Experimental Ideas & Implementations*, Wenzel Jakob and Toshiya Hachisuka (Eds.). The Eurographics Association.

Toshiya Hachisuka, Wojciech Jarosz, Richard Peter Weistroffer, Kevin Dale, Greg Humphreys, Matthias Zwicker, and Henrik Wann Jensen. 2008. Multidimensional Adaptive Sampling and Reconstruction for Ray Tracing. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 27, 3 (Aug. 2008), 33:1–33:10.

Toshiya Hachisuka and Henrik Wann Jensen. 2011. Robust Adaptive Photon Tracing Using Photon Path Visibility. *ACM Trans. on Graphics* 30, 5, Article 114 (Oct. 2011), 11 pages.

Toshiya Hachisuka, Jacopo Pantaleoni, and Henrik Wann Jensen. 2012. A Path Space Extension for Robust Light Transport Simulation. *ACM Trans. on Graphics (Proc. SIGGRAPH Asia)* 31, 6 (2012), 191:1–191:10.

Johannes Hanika, Anton Kaplanyan, and Carsten Dachsbacher. 2015. Improved Half Vector Space Light Transport. *Computer Graphics Forum (Proc. Eurographics Symposium on Rendering)* 34, 4 (June 2015), 65–74.

W Keith Hastings. 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57, 1 (1970), 97–109.

Sebastian Herholz, Oskar Elek, Jiří Vorba, Hendrik Lensch, and Jaroslav Křivánek. 2016. Product Importance Sampling for Light Transport Path Guiding. *Computer Graphics Forum (Proceedings of Eurographics Symposium on Rendering)* 35, 4 (2016), 67–77.

Heinrich Hey and Werner Purgathofer. 2002. Importance sampling with hemi-spherical particle footprints. In *Proceedings of the 18th Spring Conference on Computer Graphics.* 107–114.

Wenzel Jakob. 2013. *Light transport on path-space manifolds.* Ph.D. Dissertation. Cornell University.

Wenzel Jakob and Steve Marschner. 2012. Manifold Exploration: A Markov Chain Monte Carlo Technique for Rendering Scenes with Difficult Specular Transport. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 31, 4 (July 2012), 58:1–58:13.

Wenzel Jakob, Christian Regg, and Wojciech Jarosz. 2011. Progressive Expectation-Maximization for Hierarchical Volumetric Photon Mapping. *Computer Graphics Forum (Proc. Eurographics Symposium on Rendering)* 3, 4 (July 2011), 1287–1297.

Henrik Wann Jensen. 1995. Importance driven path tracing using the photon map. In *Proc. Eurographics Workshop on Rendering.* 326–335.

James T. Kajiya. 1986. The rendering equation. *Computer Graphics (Proc. SIGGRAPH)* (1986), 143–150.

Anton Kaplanyan, Johannes Hanika, and Carsten Dachsbacher. 2014. The Natural-Constraint Representation of the Path Space for Efficient Light Transport Simulation. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 33, 4 (2014), 1–13.

Csaba Kelemen, László Szirmay-Kalos, György Antal, and Ferenc Csonka. 2002. A Simple and Robust Mutation Strategy for the Metropolis Light Transport Algorithm. *Computer Graphics Forum* 21, 3 (2002), 531–540.

Markus Kettunen, Marco Manzi, Miika Aittala, Jaakko Lehtinen, Frédo Durand, and Matthias Zwicker. 2015. Gradient-Domain Path Tracing. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 34, 4 (2015), 123:1–123:13.

David Kirk and James Arvo. 1991. Unbiased sampling techniques for images synthesis. *Computer Graphics (Proc. SIGGRAPH)* 25, 4 (1991).

Claude Knaus and Matthias Zwicker. 2011. Progressive Photon Mapping: A Probabilistic Approach. *ACM Trans. on Graphics* 30, 3, Article 25 (May 2011), 13 pages.

Thomas Kollig and Alexander Keller. 2002a. Efficient Bidirectional Path Tracing by Randomized Quasi-Monte Carlo Integration. In *In Proc. Monte Carlo and Quasi-Monte Carlo Methods 2000.* Springer Berlin Heidelberg, Berlin, Heidelberg, 290–305.

Fig. 12. Applying density-based outlier rejection (DBOR) can remove residual spike noise. Doing so removes a lot of energy when the estimator is not able to handle all features present in the scene. Our guided sampling uses a similar criterion as outlier detection to learn difficult paths, and thus running DBOR on a guided estimator removes much less energy. In fact removing outliers on the guided estimator leads to a *lower* root mean squared error (RMSE) here.

Scene	total render time	learn time	guide paths	spp		RMSE				MAE			
						original		outlier removal		original		outlier removal	
				PT	our	PT	our	PT	our	PT	our	PT	our
DiningRoom	30min	3min	33638	608	479	0.0435	**0.0345**	0.1170	**0.0231**	0.0155	**0.0151**	0.0187	**0.0142**
Pool	30min	3min	104000	2184	517	**0.2550**	0.3162	0.1238	**0.0499**	0.1599	**0.0868**	0.1215	**0.0558**
Tumbler	30min	3min	41985	3902	1448	**0.3888**	0.4347	0.8188	**0.2154**	0.1532	**0.0767**	0.1498	**0.0473**
Dragon	1h	3min	68800	8960	8365	0.5012	**0.2121**	0.6143	**0.2198**	0.1167	**0.0880**	0.1467	**0.0930**
LivingRoom	3h	3min	63448	3840	2749	0.3926	**0.2571**	2.4309	**0.3165**	0.1063	**0.0789**	0.2459	**0.0824**
VeachDoor	90min	3min	99454	7921	4405	**0.7487**	1.3006	**3.4933**	4.8032	**0.4428**	0.6591	**0.7385**	0.9519
Hair	10min	1min	39899	3475	694	**0.0848**	0.0893	1.0013	**0.3542**	0.0281	**0.0219**	0.0279	**0.0158**

Table 1. Detailed figures about the comparisons from Figure 13 and Figure 7 in the supplemental document. We list the root mean squared error (RMSE) as well as the mean absolute error (MAE) with and without outlier removal using DBOR seperately. MAE incurs less severe penalty for spiky outlier noise that would be removed in realistic scenarios and thus reflects the visual error more closely for our method.

Thomas Kollig and Alexander Keller. 2002b. Efficient Multidimensional Sampling. *Computer Graphics Forum (Proc. of Eurographics)* 21, 3 (2002), 557–563.

Christopher Kulla and Marcos Fajardo. 2011. Importance Sampling of Area Lights in Participating Media. In *SIGGRAPH Talks*. 55:1–55:1.

Jaroslav Křivánek and Eugene d'Eon. 2014. A Zero-variance-based Sampling Scheme for Monte Carlo Subsurface Scattering. In *SIGGRAPH Talks*. 66:1–66:1.

Yu-Chi Lai, Shao Hua Fan, Stephen Chenney, and Charcle Dyer. 2007. Photorealistic Image Rendering with Population Monte Carlo Energy Redistribution. In *Proc. Eurographics Workshop on Rendering*. The Eurographics Association, 287–295.

Tzu-Mao Li, Jaakko Lehtinen, Ravi Ramamoorthi, Wenzel Jakob, and Frédo Durand. 2015. Anisotropic Gaussian Mutations for Metropolis Light Transport through Hessian-Hamiltonian Dynamics. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 34, 6 (Nov. 2015), 209:1–209:13.

Johannes Meng, Johannes Hanika, and Carsten Dachsbacher. 2016. Improving the Dwivedi Sampling Scheme. *Computer Graphics Forum (Proceedings of Eurographics Symposium on Rendering)* 35, 4 (June 2016), 37–44.

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. 1953. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics* 21, 6 (1953), 1087–1092.

Thomas Müller, Markus Gross, and Jan Novák. 2017. Practical Path Guiding for Efficient Light-Transport Simulation. *Computer Graphics Forum (Proc. Eurographics Symposium on Rendering)* 36, 4 (June 2017), 91–100.

Hisanari Otsu, Anton S. Kaplanyan, Johannes Hanika, Carsten Dachsbacher, and Toshiya Hachisuka. 2017. Fusing State Spaces for Markov Chain Monte Carlo Rendering. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 36, 4, Article 74 (2017), 10 pages.

Jacopo Pantaleoni. 2017. Charted Metropolis Light Transport. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 36, 4, Article 75 (2017), 14 pages.

Matt Pharr, Wenzel Jakob, and Greg Humphreys. 2017. *Physically Based Rendering: From Theory to Implementation* (3rd ed.). Morgan Kaufmann.

Ilya Sobol. 1994. *A Primer for the Monte Carlo Method.* CRC Press.

Ben Spencer and Mark W. Jones. 2013. Progressive Photon Relaxation. *ACM Trans. on Graphics* 32, 1, Article 7 (Feb. 2013), 11 pages.

Eric Veach. 1998. *Robust Monte Carlo methods for light transport simulation.* Ph.D. Dissertation. Stanford University. AAI9837162.

Eric Veach and Leonidas Guibas. 1994. Bidirectional Estimators for Light Transport. In *Proc. Eurographics Workshop on Rendering*. 147–162.

Eric Veach and Leonidas J Guibas. 1995. Optimally combining sampling techniques for Monte Carlo rendering. *Proc. SIGGRAPH* (1995), 419–428.

Eric Veach and Leonidas J. Guibas. 1997. Metropolis Light Transport. *Proc. SIGGRAPH* (1997), 65–76.

Jiří Vorba, Ondřej Karlík, Martin Šik, Tobias Ritschel, and Jaroslav Křivánek. 2014. On-line Learning of Parametric Mixture Models for Light Transport Simulation. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 33, 4 (Aug. 2014), 101:1–101:11.

Jiří Vorba and Jaroslav Křivánek. 2016. Adjoint-Driven Russian Roulette and Splitting in Light Transport Simulation. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 35, 4 (July 2016), 1–11.

Martin Šik, Hisanari Otsu, Toshiya Hachisuka, and Jaroslav Křivánek. 2016. Robust Light Transport Simulation via Metropolised Bidirectional Estimators. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 35, 6 (Nov. 2016), 245:1–245:12.

Pascal Weber, Johannes Hanika, and Carsten Dachsbacher. 2017. Multiple Vertex Next Event Estimation for Lighting in dense, forward-scattering Media. *Computer Graphics Forum (Proceedings of Eurographics)* (April 2017).

Tobias Zirr, Johannes Hanika, and Carsten Dachsbacher. 2018. Reweighting firefly samples for improved finite-sample Monte Carlo estimates. *Computer Graphics Forum* 37, 6 (2018), 410–421.

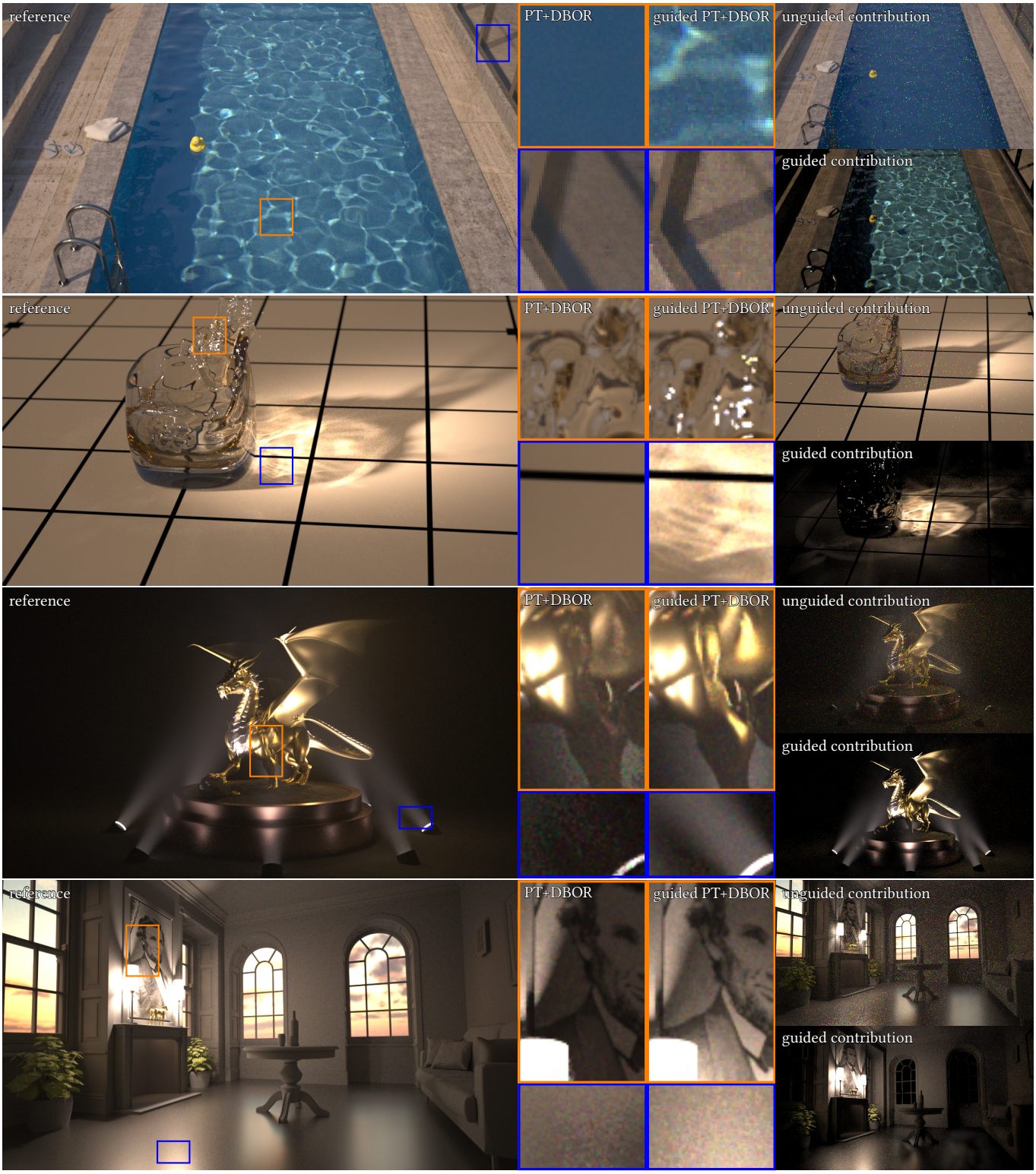

Fig. 13. Equal-time comparisons of PT and our guided PT. The insets show PT and guided PT with outlier removal and therefore PT misses lighting effects it can not render efficiently (e.g. caustics). In all scenes guided PT robustly identifies the part of light transport that can not be handled by PT efficiently and improves upon this using our guided sampling for these parts. As a side effect, we obtain an interesting separation into a smooth/low contrast image (unguided contribution) and a sharp/high contrast image (guided contribution). Please see Figure 7 in the supplemental material for DBOR/no-DBOR comparisons.