DOI: 10.1111/cgf.14881
Eurographics Symposium on Rendering 2023
T. Ritschel and A. Weidlich
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 42 (2023), Number 4
e14881 *pp. 1 - 15*

Markov Chain Mixture Models for Real-Time Direct Illumination

Addis Dittebrandt[1] and Vincent Schüßler[1] and Johannes Hanika[1] and Sebastian Herholz[2] and Carsten Dachsbacher[1]
[1] Karlsruhe Institute of Technology, [2] Intel Corporation

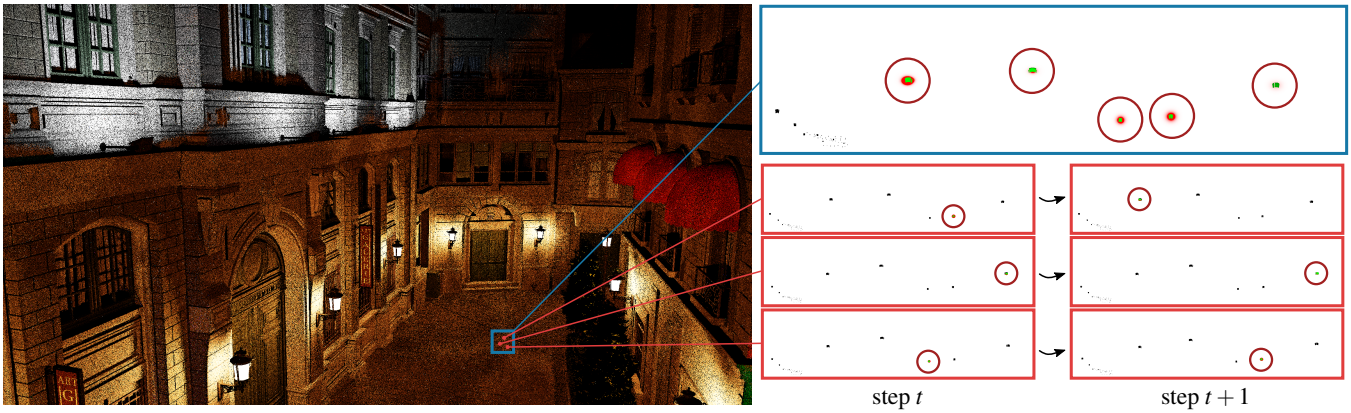

Figure 1: *Left: Direct illumination from many small light sources rendered using our mixture model with 8spp (10.8ms at 1080p), without explicit light source sampling. Right: a visualization of the mixture model in latitude/longitude projection. Red insets: At any time step, every pixel holds only a single vMF lobe (red), aimed at one of the light sources (black), the overlap is shown in green. Through a Markov chain process, lobes are adapted and exchanged locally. Blue inset: We can observe the mixture model as different slices of this construction: spatially over an image region, as well as temporally over multiple time steps. For better visibility, we surround lobes with a dark red circle.*

Abstract

We present a novel technique to efficiently render complex direct illumination in real-time. It is based on a spatio-temporal randomized mixture model of von Mises-Fisher (vMF) distributions in screen space. For every pixel we determine the vMF distribution to sample from using a Markov chain process which is targeted to capture important features of the integrand. By this we avoid the storage overhead of finite-component deterministic mixture models, for which, in addition, determining the optimal component count is challenging. We use stochastic multiple importance sampling (SMIS) to be independent of the equilibrium distribution of our Markov chain process, since it cancels out in the estimator. Further, we use the same sample to advance the Markov chain and to construct the SMIS estimator and local Markov chain state permutations avoid the resulting bias due to dependent sampling. As a consequence we require one ray per sample and pixel only. We evaluate our technique using implementations in a research renderer as well as a classic game engine with highly dynamic content. Our results show that it is efficient and quickly readapts to dynamic conditions. We compare to spatio-temporal resampling (ReSTIR), which can suffer from correlation artifacts due to its non-adapting candidate distributions that can deviate strongly from the integrand.While we focus on direct illumination, our approach is more widely applicable and we exemplarily show the rendering of caustics.

CCS Concepts

• ***Computing methodologies*** → *Ray tracing;*

1. Introduction

The simulation of light transport in a scene is fundamental for photorealistic rendering and most often done using Monte Carlo integration [PJH23]. It has found broad adoption in the visual effects industry [FHH*19] and due to the availability of fast ray tracing cores [NVI18] and noise reduction filters [SKW*17] it is also becoming increasingly relevant for real-time applications with dynamic environments such as video games. This is boosted by recent improvements of direct illumination resampling [BWP*20].

Our work also primarily targets the sampling of direct illumina-

tion. To arrive at a general technique that can, in the future, be extended more easily to indirect illumination, we do not perform next-event estimation but use only hemispherical samples to discover lights. In the spirit of recent path guiding approaches [VKŠ*14], we use parametric mixture models to represent and sample a target distribution (e.g., the incident radiance field). In contrast to resampling schemes, our approach creates new samples by drawing from a continuous distribution instead of selecting from a fixed, previously discovered set of samples. However, to represent the target distribution sufficiently, the number of required mixture components can be large, and since it is unknown a priori, additional statistics need to be stored to refine or merge components [RHL20] adaptively. Due to this storage and evaluation overhead, realizing an efficient guiding solution using discrete mixture models is challenging, especially if we want to store one mixture model per pixel. Our technique combines the simplicity and GPU-friendly fixed memory footprint of a single component per pixel [Der22] with the expressiveness of a mixture of components by replacing a deterministic mixture with the equilibrium distribution of a Markov chain (fig. 1), whose state space defines the parameter-tuple of a component. In detail, we use mixtures based on von Mises-Fisher (vMF) lobes. The Markov chains transition by exchanging the vMF lobe parameters between pixels, and adapting the mean and concentration parameters using a maximum likelihood estimate based on the history of samples. Large steps are facilitated by injecting independent hemisphere samples. Since each pixel hosts their own lobe, we adapt this data model after every sample without synchronization overhead.
Our contributions are:

- **A theoretical foundation of a Markov chain-controlled randomized vMF mixture model.** We use this foundation to sample the parameter space which defines vMF lobes approximating the incident radiance times material evaluation.
- **An efficient unbiased estimator based on stochastic multiple importance sampling (SMIS).** We are, thereby, leveraging the fact that, when using stochastic MIS, the equilibrium distribution cancels out from the estimator, and thus it is not required to know or compute the equilibrium distribution of the Markov chain process. This leaves us ample freedom when designing appropriate transition strategies to balance temporal adaptation in dynamic scenes and accurate representation of the incident light field.
- **An efficient way of reusing ray traced samples between the Markov chain and the estimator.** This allows us to build a light-weight, real-time path guiding technique for direct illumination, which is able to track dynamic lights, and an exemplary extension to underwater caustics.

We analyze the behavior of our method in simple experiments and demonstrate its performance in two open-source rendering systems, Nvidia's Falcor [KCK*22] and the Quakespasm engine to evaluate highly dynamic environments. Since the Markov chain process bears a resemblance to particle filters, it is well suited to track dynamic light sources moving at moderate speeds, which is also confirmed by our experiments.

2. Background

Monte Carlo Light Transport Simulating surface-based light transport in a scene requires solving the *rendering equation* [Kaj86], e.g. via path tracing. A crucial component of the rendering equation is the integral of the reflected radiance, which in the case of direct illumination simplifies to

$$L_r(\mathbf{x}_0,\mathbf{x}_1) = \int_\Omega f_r(\mathbf{x}_0,\mathbf{x}_1,\mathbf{x}_2)\, L_e(\mathbf{x}_2,-\omega)\cos\theta\, d\omega, \tag{1}$$

where $\mathbf{x}_0$ is the point on the image plane, $\mathbf{x}_1$ is the point at the primary ray intersection, and $\mathbf{x}_2$ is a point on a light source that is hit by a ray traced from $\mathbf{x}_1$ in direction ω. The BSDF $f_r(\mathbf{x}_0,\mathbf{x}_1,\mathbf{x}_2)$ relates the differential radiance scattered at $\mathbf{x}_1$ towards $\mathbf{x}_0$ to the differential irradiance at $\mathbf{x}_1$ received from $\mathbf{x}_2$. For brevity, we will use the shorthand notations $L_r(\mathbf{x}_1)$, $f_r(\mathbf{x}_1)$, and $L_e(\mathbf{x}_2)$, which omit dependencies on previous or following vertices.

Monte Carlo integration is used to estimate the solution of the rendering equation due to its high dimensionality. The Monte Carlo estimator for direct illumination, based on a set of N random directions $\{\omega_1,\ldots,\omega_N\}$ with PDF $p(\omega_n)$, is

$$\langle L_r(\mathbf{x}_1)\rangle = \frac{1}{N}\sum_{n=1}^{N}\frac{f_r(\mathbf{x}_1)\, L_e(\mathbf{x}_{2,n})\cos\theta_n}{p(\omega_n)}. \tag{2}$$

The variance of this estimator depends on the sampling strategy, i.e. how close $p(\omega)$ is to the optimal PDF, which is proportional to the integrand of eq. (1):

$$p_{\text{optimal}}(\omega) \sim f_r(\mathbf{x}_1) L_e(\mathbf{x}_2)\cos\theta. \tag{3}$$

Parametric Mixture Models Parametric mixture models are widely used in machine learning and computer graphics to approximate complex distributions in a compact analytical form. Especially mixtures of von-Mises Fisher distributions (vMF) are useful in rendering due to their natural definition in the spherical domain:

$$p(\omega) = \sum_{k=1}^{K}\pi_k V(\omega \mid \mathbf{t}_k), \tag{4}$$

where $\pi_k \in [0,1]$ defines the weight of the k-th component and $\mathbf{t}_k$ its parameters, and $\sum_{k=1}^{K}\pi_k = 1$. A vMF distribution V defined by $\mathbf{t} = (\mu,\kappa)$ is an isotropic distribution from the exponential family with concentration κ around mean direction μ (see [Jak12] for more details). A sample of the mixture can be generated by selecting a component proportional to its weight and then sampling the selected component.

Markov chains and Markov chain Monte Carlo (MCMC) Markov chains model stochastic processes as a series of states (discrete or continuous), where the probability of going from one state to the next only depends on the current state. In essence, a Markov chain has no notion of history. For certain Markov chains, the distribution of possible states approaches a unique distribution, the equilibrium distribution, as more transitions are applied from any initial state.

The basic idea of Markov chain Monte Carlo algorithms is to construct a Markov chain with an equilibrium distribution equal (or proportional) to the integrand f. Satisfying the detailed-balance

condition gives this guarantee:

$$\forall x, y : f(x)P(x \to y) = f(y)P(y \to x), \tag{5}$$

where x and y represent two states and P is the transition kernel. Since f is fixed, P must be tailored such that this condition is met.

The Metropolis-Hastings algorithm splits P into a user-supplied proposal distribution T and acceptance probability A. Since T is fixed, A must be chosen such that detailed-balance is met:

$$A(x \to y) = \min\left(1, \frac{f(y)}{f(x)} \frac{T(y \to x)}{T(x \to y)}\right) \tag{6}$$

The algorithm iteratively produces a proposal state according to T from the current state and actually transitions to it according to A. Otherwise it stays in the current state. This construction allows for expert knowledge to be supplied in the form of tailored T in order to better explore important parts of the integrand.

3. Previous Work

Light source sampling The large body of work on sampling of direct illumination in rendering can be roughly divided into techniques which are either concerned with selecting important light sources using a hierarchy [EK18; MPC19; Yuk19], or with sampling points on a chosen light source, e.g. proportional to (projected) solid angle [Arv95; HPM*20; Pet21]. All of these rely on the availability of explicit light source information, which is sometimes hard to obtain, e.g. with textured emission or procedurals, and are lacking important information about occlusion. In contrast, we only require initial hemisphere or BSDF samples, and steer our mixture model towards highly-contributing directions.

Resampled importance sampling and ReSTIR Efficient resampling schemes [BWP*20] have shown impressive performance for sampling direct illumination of many light sources, and several extensions to global illumination have been proposed [OLK*21; LWY21; Boi21; BJW21]. However, since these works are based on resampled importance sampling (RIS) [TCE05], they do have conceptual limitations [LKB*22]: resampling cannot discover new light sources, but only selects from a known set. That is, no new information is gained by resampling, and other techniques such as Markov chain transitions need to be mixed in to fulfill this task [SLK*22]. Our work is also based on selection by resampling and Markov chain transitions, but in a different order and different state space.

Dependent Monte Carlo Sampling Our technique constructs an unbiased Monte Carlo estimator on top of dependent samples from a Markov chain. Previous approaches achieved this by deterministic groups of samples [WND*14; HDF15] or by explicit estimation of the reciprocal integral of the PDF [Boo07; ZGJ20]. We employ continuous MIS [WGGH20] to achieve this.

Metropolis light transport Metropolis light transport [VG97] simulates a certain target distribution by estimating a histogram of the posterior of the Markov chain. The posterior is a unique equilibrium distribution which does not depend on the initial state. To utilize this to estimate integrals, the mean image brightness has to be computed separately and the posterior of the Markov chain has to be known and is explicitly controlled by using transition densities to compute an acceptance probability after Metropolis-Hastings. We do not require either: our algorithm still works if the equilibrium distribution is not unique, and we can tune acceptance probabilities to focus on fast adaptation to dynamic content without evaluating transition densities or mean image brightness.

There exist other relevant extensions to MLT that share samples in image space through replica exchange [GWH20] or ensembles of light transport paths to guide future transition kernels [BSMD21]. Nevertheless, these works still rely on detailed balance conditions in the same fashion as classic MLT.

Adaptive MCMC Adaptive MCMC [HST01] generally requires to remove the adaptation of the transition densities in the limit. While our requirements on the Markov chain are lighter, our adaptation will also vanish if the scene remains static.

Ensemble and Population Monte Carlo There exist many Monte Carlo algorithms which utilize a population or ensembles of samples to improve an estimator [MP92; SW86]. Our algorithm is similar to Population Monte Carlo (PMC) [CGMR04], which draws samples from a population of lobes. After the sampling step, the lobe means are selected from the sample locations via weighted resampling. Variants such as Optimized-PMC [EC22] also estimate the lobe covariances. In our work, the populations are tailored to groups of pixels and threads. We explicitly have a population of lobe means and concentrations, and the mean will be updated from the samples via a maximum likelihood step.

PMC is known to reduce the diversity of the samples due to the resampling step. Elvira et al. [EMLB17] survey different resampling strategies (global, local, independent) and propose a reduced degeneracy strategy. Our resampling strategy is similar to what they call *independent*, but is designed to adapt quickly to dynamically changing target functions and for simple thread synchronization.

There exist prior approaches which employ MCMC sampling in combination with a Monte Carlo estimator [SK21; RS20]. This is useful to draw more information from rejected samples. Our Markov chain operates on a parameter space controlling the lobes of the sampling PDF and is not directly used to estimate an integral.

Layered adaptive importance sampling [MELC15] is a very related technique which estimates lobe means via MCMC and samples from these in combination with multiple importance sampling (MIS). Unlike our work, they use fixed lobe covariances and there is no sample reuse between the MCMC and the MC step.

Path guiding Data-driven adaptive sampling approaches such as *path guiding* have proven to be valuable tools in offline production rendering [VHH*19] to reduce variance when simulating complex global light transport. These gather information about light transport during rendering or in a pre-processing step to propose samples proportional to a guiding target function, e.g. incident radiance or its product with the BSDF. Various approaches have been presented, which differ in used spatial or directional data structures [Jen95; LW95; HP02; BDC12; MGN17], domains [ZZ18; RHJD18], or target functions [RGH*20]. Closely related to this work are approaches based on parametric mixture models [VKŠ*14; HEV*16; HZE*19; RHL20; DPÖM21; SHJD22] of

distributions from the exponential family, like Gaussian or von-Mises Fisher (vMF). They are usually fitted using weighted expectation maximization (wEM) [VKŠ*14]. Ruppert et al. [RHL20] present a variance-aware split and merge algorithm to increase the robustness of wEM, while determining the optimal number of mixture components.

Due to the strict constraints on compute time and memory bandwidth, real-time guiding approaches [DHD20; Pan20; Der22] cannot afford complex data structures or fitting methods. Inspired by Derevyannykh [Der22], we store a single vMF component per pixel. However, we additionally combine vMF lobes of neighboring pixels to build a randomized mixture model.

4. An Estimator using a Randomized Mixture Model

The core idea pursued in this work is to replace a mixture model with a discrete and fixed number of components and their weights (eq. (4)) by a continuous equivalent:

$$\mathrm{p}(\omega) = \int \mathrm{p}(\mathbf{t}) V(\omega \mid \mathbf{t}) \, \mathrm{d}\mathbf{t}, \tag{7}$$

where $\mathbf{t}$ represents the component parameters and $\mathrm{p}(\mathbf{t})$ the component weight.

While sampling such a mixture is typically trivial, estimating the reciprocal PDF to construct an estimator is often expensive [Boo07; ZGJ20]. We instead use the n-sample *stochastic multiple importance sampling* (SMIS) estimator proposed in continuous multiple importance sampling [WGGH20, eq. (12)]:

$$\langle L_r(\mathbf{x}_1) \rangle_{\mathrm{SMIS}} = \sum_{i=1}^{n} \frac{f_r(\mathbf{x}_1) L_e(\mathbf{x}_{2,i}) \cos\theta_i}{\sum_{j=1}^{n} \mathrm{p}(\omega_i | \mathbf{t}_j)}. \tag{8}$$

We essentially approximate the full PDF through a stochastic selection of lobes. Note that $\mathrm{p}(\mathbf{t})$ is not present in the expression, since it canceled out: we do not need to explicitly compute or even know about the specific distribution of $\mathrm{p}(\mathbf{t})$. We exploit this property in the following.

We define $\mathrm{p}(\mathbf{t})$ through a Markov chain operating on a state space which allows us to compute the lobe's parameters $\mathbf{t} = (\mu, \kappa)$. A one-dimensional example is depicted in fig. 2, where the orange line represents the integrand. As the Markov chain transitions (first four plots), it deposits probability mass resulting in an (unknown) equilibrium distribution $\mathrm{p}(\mathbf{t})$. The resulting sample distribution $\mathrm{p}(\omega)$ can be observed in the last plot. Contrary to Metropolis-Hastings, which gives strong guarantees about the *exact* equilibrium distribution through detailed balance conditions, we do not depend on this reasoning. As a consequence, there is a lot of freedom designing transition steps in the Markov chain, to, for example, emphasize temporal adaptation without causing bias.

For our use-case (direct illumination), we store one Markov chain state per pixel in a simple screen space data structure (fig. 3). We make use of the flexibility in state transitions and perform them in an iterative fashion per frame, incorporating neighbor states to facilitate information sharing, as detailed in section 4.1.

West et al. [WGGH20] have made an additional assumption when proving unbiasedness of the SMIS estimator: All the pairs of lobes with their drawn samples need to be independent among each other. In particular, this means that (1) samples must be independent from each other and (2) techniques must be independent from drawn samples and (3) also from other techniques. These requirements are initially violated when using our inherently correlated Markov chain process, resulting in a biased estimator. In section 4.2, we show how the first two requirements can be upheld in an efficient fashion, by shuffling states between neighboring pixels. In appendix A, we prove that the last requirement is actually not needed at all, thus making our estimator unbiased even in the case of correlation. This is important when reusing Markov chain states from the framebuffer.

4.1. A Markov Chain Mixture Model

In this section, we detail the Markov chain process leading to a continuous mixture model of vMF lobes. This mechanism has two partially opposing goals: (i) the equilibrium distribution should faithfully represent the actual target distribution, (ii) it should adapt fast to dynamically changing content.

Extended State Space To address the first goal, we use a maximum a-posteriori (MAP) approach [BG88; BHO10] to compute accurate distribution parameters $\mathbf{t} = (\mu, \kappa)$. This is directly reflected in our state space definition of the Markov chain (table 1): it does not incorporate the parameter tuple, but rather sufficient statistics of a maximum likelihood estimator, from which the parameters are inferred (algorithm 2, with more details in about the MAP priors in section 5). This way, previous samples can be incorporated as well. Note that this makes the reduced state space over (μ, κ) non-Markovian, since transitions are affected by past states (section 7).

We additionally store a scoring term f. It is used to compute acceptance probabilities and is simply a one-sample estimate of the direct-illumination integral (eq. (1)).

Table 1: *Markov chain state $S = (\bar{\mathbf{y}}, \bar{r}, \bar{w}, N, f)$. We access elements of this tuple in the form of $S.\bar{w}$.*

Symbol	Meaning
$\bar{\mathbf{y}}$	weighted mean of light source vertex positions
$\bar{r}$	mean cosine of weighted directions
$\bar{w}$	sum of weights (estimates of eq. (1))
N	number of samples which went into this state so far
f	score (estimate of eq. (1))

State Transitions The maximum a-posteriori approach is sometimes at odds with the second goal which requires that newly appearing lights should be picked up very quickly. To facilitate this, we exchange Markov chain state between neighboring pixels. This results in a collaborative algorithm where information about new or undersampled features spreads quickly over the frame buffer.

Within one frame, each pixel will independently (and in parallel) mutate their state in a loop with n iterations. Each iteration is composed of three stages as illustrated in fig. 4 (we refer to table 2 for symbol meanings):

1. We look at the states $S_q^{(i)}$ in other pixels q in a window around our pixel p and potentially use their state to overwrite ours.

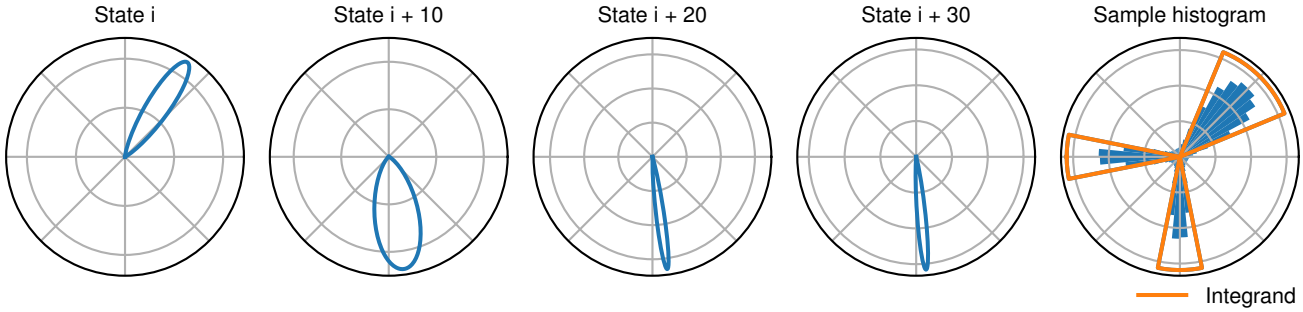

Figure 2: *1D visualization of our Markov chain process. The first four plots exemplarily show the evolution of the Markov chain over the iterations. The last plot shows the target integrand and resulting sample distribution after 800000 iterations.*

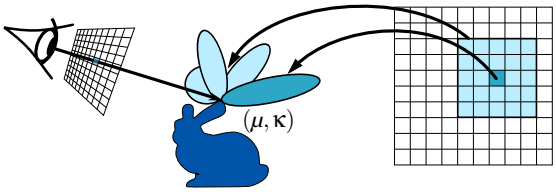

Figure 3: *The state space of our Markov chains defines the parameters of a single von-Mises-Fisher (vMF) lobe (μ, κ), and we run one Markov chain per pixel p. A random instance of the full mixture model for any pixel can be generated by collecting vMF lobes over the time progression of the Markov chain, or in a window around the query position (shown in light blue).*

2. We create a new path vertex $\mathbf{x}_2$ by drawing either an independent hemispherical sample or by using the current vMF lobe of this pixel S.
3. Through a mutation/acceptance step, we keep a current state S_c over all iterations which will eventually be written out as $S_p^{(i+1)}$.

In the following paragraphs we detail the three stages of the Markov chain transition steps.

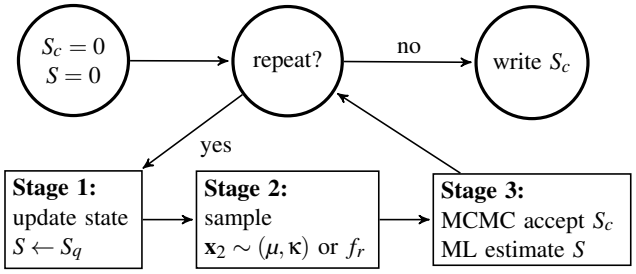

Figure 4: *Overview of the Markov chain transitions in a single pixel. An internal sampling state S is updated from neighboring pixels q. Then a new path vertex is drawn from the vMF lobe or a hemispherical PDF. In a third step, a current state S_c is stashed away if the state S passes a MCMC acceptance test. S_c will be written out to be shared with the other pixels in the next frame.*

Table 2: *Glossary of symbols used in the following stages.*

Symbol	Meaning
S_p, S_q	Markov chain states for pixels p and q, respectively
S	Tentative state used for sampling
S_c	Current state to be written after rendering the frame

Stage 1: collaborative discovery/inter-pixel update Algorithm 1 details this stage: as mentioned, we potentially replace the pixel state S by those found in a framebuffer storing the states of the neighbor pixels written out in the last frame $S_q^{(i)}$. That means they have one frame lag and need to be accessed after correcting the pixel position q with estimated motion vectors to account for dynamic changes (e.g. camera or object motion) between frames. We randomly select a pixel q in a configurable window around p by sampling a B-spline kernel of degree 5. The offset is the sum of four random numbers [SSA05].

The state found, S_q, provides a light position $\mathbf{y}$ but no information about the shading point that lead to the score f which is also provided in the buffer. This means that we have to assume some smoothness over both space and time such that this mutation type makes sense. This has been successfully exploited by similar approaches in the past [BWP*20; VG97].

In practice, we use a window size of 91×91 at a resolution of 1920×1080. While it appears large, it is a trade off between "spreading the news" quickly over the image, and the potential mismatch in score function between pixels due to occlusion. Cheap rejection heuristics (e.g. based on the shading point $\mathbf{x}$) are possible to use here without causing bias, but we simply use $\text{score}(S) = S.f$.

Like population Monte Carlo methods, we perform weighted resampling of the neighbor states: we use the score function as weight and use reservoir-style sampling in a linear scan to randomly select a neighbor state proportional to this weight [CHA82].

In the algorithm listings, a state S is considered *valid*, if it accumulated non-zero weight, i.e. $\bar{w} > 0$. Also note that in line 1 an additional shuffling step is performed. It is included here already to provide a coherent view of the program flow but the necessity will later be explained in section 4.2.

Algorithm 1 Collaborative discovery/inter pixel update

$S \leftarrow$ shuffleUp subgroups ▷ see section 4.2
sum $\leftarrow$ score(S)
for some random neighbor pixels q, until score$(S) > 0$ **do**
  $S_q \leftarrow$ load state $S_q^{(i)}$ at motion corrected q
  $s_q \leftarrow$ score(S_q)
  $\xi \leftarrow$ uniform random in $[0,1)$
  if not valid(S) or $\xi < s_q/(s_q+\text{sum})$ **then**
    $S \leftarrow S_q$
  end if
  sum $\leftarrow$ sum $+s_q$
end for

Stage 2: sampling, ray casting To generate initial samples as well as large steps in the Markov chain we use hemispherical sampling. This can be achieved by sampling from the BSDF or simply by using a cosine distribution. We use either in our two separate implementations. This is performed if the state S holds no valid vMF lobe as well as at random. We use a fixed probability of 0.2. This also makes our sampling defensive, since the estimator now always contains BSDF or cosine sampling as a backup-strategy (see appendix A). In case the sample is drawn from the vMF, μ and κ are first reconstructed from the internal state representation (see algorithm 2).

Algorithm 2 Sampling and ray casting

$\xi \leftarrow$ uniform random in $[0,1)$
if not valid(S) or $\xi < 0.2$ **then**
  $p(\omega) \leftarrow \frac{\cos\theta}{\pi}$ or $f_r(\mathbf{x}_1)$ ▷ cosine or BSDF sampling
else
  $\mu \leftarrow (S.\bar{\mathbf{y}} - \mathbf{x}_1)/\|S.\bar{\mathbf{y}} - \mathbf{x}_1\|$ ▷ convert to direction
  $r \leftarrow (N^2 \cdot S.\bar{r} + N_p \cdot r_p)/(N^2 + N_p)$ ▷ apply prior
  $\kappa \leftarrow (3r - r^3)/(1 - r^2)$ ▷ approx. lobe width
  $p(\omega) \leftarrow V(\omega|\mu,\kappa)$ ▷ vMF distribution
end if
sample $\omega \sim p(\omega)$
shoot ray to find $\mathbf{x}_2$

Stage 3: internal pixel state update After tracing the new direction and finding a path vertex $\mathbf{x}_2$, we compute the score f of this connection. Then, we use a Metropolis acceptance step to decide whether to advance the current state S_c. After the MCMC step the maximum a-posteriori adaptation of the lobe is performed. Note that this includes zero-contribution samples which will make the lobe a bit narrower. Algorithm 3 shows this procedure.

The score function f is evaluated as the luminance of the emission L_e times BSDF $f = \text{lum}(L_e(\mathbf{x}_2) \cdot f_r(\mathbf{x}_1))/p(\omega)$. We divide out the solid angle PDF to make f a coarse estimator for the pixel contribution. That is we include the BSDF which technically also depends on the incident direction. Since this varies slowly over the image it improves sampling of glossy objects considerably.

Algorithm 3 Pixel state update, given a complete path $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2$

$f \leftarrow \text{lum}(L_e(\mathbf{x}_2) \cdot f_r(\mathbf{x}_1))/p(\omega)$
$a \leftarrow \min\{1, f/S_c.f\}$
$\xi \leftarrow$ uniform random in $[0,1)$
if $f > 0$ and (not valid(S_c) or $\xi < a$) **then** ▷ MCMC transition
  if hemisphere sample **then**
    $S \leftarrow \{0\}$ ▷ reset lobe if $\mathbf{x}_2$ is not from vMF
  end if
  $S.f \leftarrow f$
  $S_c \leftarrow S$
end if
$S.N \leftarrow \min\{S.N + 1, 1024\}$ ▷ maximum likelihood update of S
$\alpha \leftarrow \max\{1.0/S.N, 0.1\}$
$\bar{w} \leftarrow \text{mix}(S.\bar{w}, f, \alpha)$
$S.\bar{\mathbf{y}} \leftarrow \text{mix}(S.\bar{w} \cdot S.\bar{\mathbf{y}}, f \cdot \mathbf{x}_2, \alpha)/\bar{w}$ ▷ comp. μ as in algorithm 2:
$\omega \leftarrow \text{mix}(S.\bar{w} \cdot S.\bar{l} \cdot \mu, f \cdot (\mathbf{x}_2 - \mathbf{x}_1)/\|\mathbf{x}_2 - \mathbf{x}_1\|, \alpha)/\bar{w}$
$S.\bar{l} \leftarrow \|\omega\|$
$S.\bar{w} \leftarrow \bar{w}$

4.2. Sample decorrelation

Using a sample both to advance the Markov chain and construct the SMIS estimator leads to dependencies between a sample and subsequent techniques and samples (since the drawn sample influences the next lobe), resulting in a biased SMIS estimator. A naïve solution is to draw a separate sample each for both tasks. While this removes these dependencies, drawing a sample in our context is generally an expensive operation, since it requires to evaluate complex shading models and to trace a ray. Thus, reusing all samples for the SMIS estimator is paramount.

In the following, we leverage that we are running multiple Markov chains in parallel - one for each pixel. By permuting Markov chain states between pixels after each step, we break any sample dependencies in the SMIS estimator. We discuss *temporal* and *spatial* schemes as depicted in fig. 5 that differ in where techniques constituting the SMIS estimator are sourced from.

Temporal SMIS (TSMIS) The mutation/acceptance loop that is executed n times in each pixel leaves a history of n vMF lobes that together form a mixture. A sample is drawn from each lobe and the SMIS estimator is evaluated accordingly. In practice, this requires us to store an array of n sample contributions, locations, and vMF parameters used to sample these. Computing the value of the estimator is then quadratic in n (eq. (8)). For moderate values of n we found this to be a negligible cost as compared to the ray tracing.

As mentioned, we cannot easily reuse the sample used to advance the Markov chain for the pixel estimate due to sample dependence. To decorrelate the two processes, we leverage the fact that we are running a whole population of Markov chains, with one state per pixel. We completely reuse all the MCMC samples, the large as well as the small step mutations, in the SMIS estimator. To make sure MCMC and SMIS are completely independent in one pixel, it is enough to simply shuffle up the MCMC state after mutation and before evaluating the SMIS estimator (fig. 5, left): we divide the frame into tiles and cyclically shift states between pixels in a tile, rotating the last one back to the first. As long as the tile

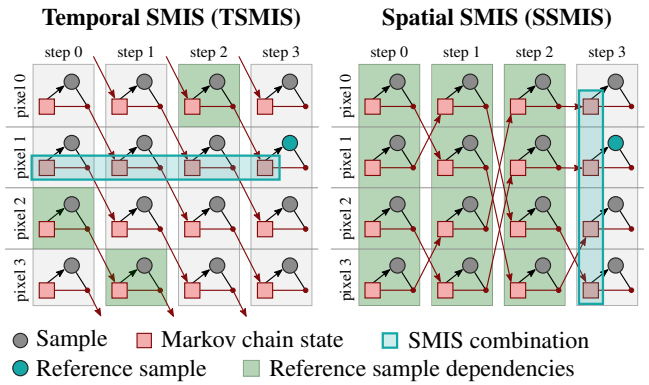

Figure 5: *Construction of our two unbiased estimators. After each step (sampling and state update, black arrows), we exchange Markov chain states between pixels (red arrows) to allow for unbiased SMIS. Temporal SMIS (left) decorrelates states by cyclic shifts between pixels and constructs a 4-SMIS estimator over the independent states of all 4 steps. Spatial SMIS (right) constructs a 1-SMIS estimator over the random set of states in the tile at a single step. It selects a state for each pixel to sample from by random permutation. In both cases, we can see for a given reference sample (marked blue) that its dependencies (green) do not overlap with the techniques in the SMIS estimator (blue box).*

size is larger than the SMIS sample set size n, the estimator for each pixel will be constructed from completely independent samples.

We choose tile sizes below or equal to 32 pixels (e.g. 8×4). On contemporary GPUs, the threads processing the individual pixels execute in groups of usually 32 (referred to as *subgroups*) and state can be exchanged very efficiently between them. We use this to exchange Markov chain states instead of slow video memory.

Spatial SMIS (SSMIS) For larger n the required storage and quadratic cost in n of temporal SMIS can be limiting factors. An alternate construction scheme can be derived by constructing an estimator over the set of vMF lobes at a given step in the pixel tile.

The resulting estimator is no longer a true n-sample SMIS estimator, but multiple realizations of a one-sample SMIS estimator over n techniques. Essentially, we perform a stratified selection of one technique using uniform probabilities, resulting in:

$$\langle L_r(\mathbf{x}_1)\rangle^1_{\text{SMIS}} = \frac{f_r(\mathbf{x}_1)L_e(\mathbf{x}_2)\cos\theta}{\frac{1}{n}\sum_{j=1}^{n} \mathrm{p}(\omega|\mathbf{t}_j)}. \tag{9}$$

The estimator can be directly evaluated after a new sample is generated in each mutation/acceptance step. Additionally, the storage for all vMF lobes is distributed among the threads processing the individual pixels in the tile. As a result, the storage requirements are constant per thread. n is now also independent on the number of steps, which allows one to use fewer steps and still have a large SMIS set to reduce variance.

The selection is implemented as a permutation on the entire Markov chain state to allow using the generated sample to advance the Markov chain. Permutations are particularly important in this instance to avoid loosing states through duplicate selections. In practice, we implement the permutation by xoring a random mask on the subgroup invocation id.

5. Implementation details

We implemented our approach in two open-source rendering systems: The NVIDIA Falcor research renderer [KCK*22] and the classic Quakespasm engine. While Falcor allows us to evaluate with reference implementations of various algorithms, Quakespasm features scenes with highly dynamic content. Initial samples are generated using BSDF (Falcor) and cosine (Quakespasm) sampling. We use classic geometric motion vectors (Falcor) and optical flow [HTD21] (Quakespasm) to gather Markov chain states from the previous frame. The Quakespasm framework does not provide light lists to be sampled. Lights are merely represented as (animated) *fullbright* textures, and we did not attempt to implement next event estimation (NEE) on top of this. While Falcor does contain explicit representations for emitters for NEE, we do not use them to demonstrate the technique's independence of such representations. We will release both implementations with the final version of the paper under a permissive license.

Sufficient statistics To account for parallax between different pixels towards a light, the sufficient statistics (table 1) incorporate a weighted mean of light source vertex positions $\bar{\mathbf{y}}$. The mean of the vMF μ can be recovered for a given position $\mathbf{x}$ as $\mu = (\bar{\mathbf{y}}-\mathbf{x})/\|\bar{\mathbf{y}}-\mathbf{x}\|$. This representation has the advantage that it does not depend on a common pivot point compared to using a direction and distance [RHL20]. Therefore the statistics can be exchanged directly between neighbors.

N is employed to compute a more accurate arithmetic average for the first few samples before switching over to an exponential average to better adapt to dynamic changes. It is also used to support custom priors which vanish over time. Such a prior is defined as N_p, the virtual number of samples it accounts for, as well as r_p, the mean cosine corresponding to the prior lobe width.

Uninformed vs. informed priors We use maximum a-posteriori parameter estimation with priors to stabilize lobe width estimation with very few samples. This is common due to the reset of lobe statistics when switching to different or new features. To get accurate fits quickly, we try to use all available prior knowledge about the scene's light sources. In the Falcor scenes, the size of lights is unrestricted and we thus resort to an uninformed (uniform) prior with a mean cosine of zero. For Quakespasm, we leverage the quantization of meshes to a size of 1.0 in world-space. There, we use an informed prior that fits the mean cosine to this size based on distance, resulting in a more uniform or peaky lobe for close or distant lights, respectively (see fig. 6). Tracking of emitting particles benefits most from this, since particles are sized exactly according to this quantization. In both implementations, the prior vanishes in a squared relationship to the number of samples. This is crucial for adaptation under dynamic conditions and also because fits are lost regularly due to initial samples, thus mandating fast recovery towards a feature.

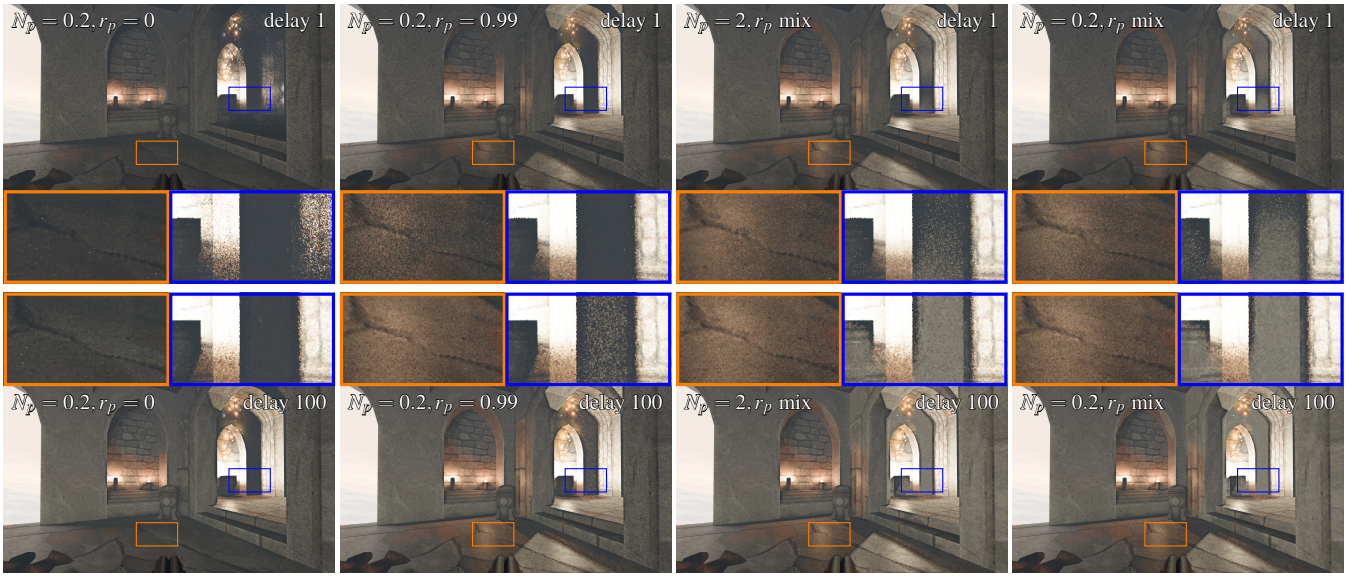

Figure 6: *The effect of different priors with mean cosines r_p and weight N_p. After a one frame they differ significantly (top row) but begin to look similar after 100 frames delay (bottom row). Here the lights are very small, so an informed prior with a focused mean cosine of $r_p = 0.99$ generally performs better than $r_p = 0$. The label "r_p mix" signifies a heuristic prior which becomes more diffuse ($r_p = 0$) as the initially discovered light comes closer to the shading point. Frame time was 20ms on an Nvidia RTX 2080 Ti and 10ms on the RTX 3080 Ti.*

Apparent targets For caustics rendering, retargeting a vMF lobe to the interface (the first intersection from the originating surface) produces suboptimal results. We use the concept of apparent distance to the emitter [RHL20]. Since our vMF lobes are represented with respect to the target and not direction, we use an apparent target instead. Intuitively, we prolong the point at the interface in the originating direction using the distance from interface to emitter, scaled by a correction factor [RHL20, eq. (35)].

6. Evaluation

The Falcor implementation is evaluated on an Nvidia RTX 3080 using openly available [Lum17] and custom test scenes. The Quakespasm implementation is evaluated on an Nvidia RTX 2080 Ti and 3080 Ti and uses scenes from the *Arcane Dimensions* mod (`https://www.moddb.com/mods/arcane-dimensions`), which introduces much higher geometric complexity as compared to the original game. Since we mostly present rendering of direct illumination, we focus our comparisons to ReSTIR, which represents the state of the art in this area, using the publicly available RTXDI implementation in Falcor. In particular, we do not compare against works that cannot represent high-frequency direct illumination adequately [DHD20; Der22].

We report rendering times at display resolutions of 1920×1080. In Falcor, we only include the techniques themselves, not including rasterization of the G-Buffer or post-processing passes such as denoising. The Quakespasm render times include a ray traced G-Buffer and temporal anti-aliasing (TAA).

We refer to temporal and spatial SMIS as A-TSMIS or A-SSMIS, respectively. A is the tile size (e.g. 8-TSMIS operates on a 4×2 tile). We also denote the effective ray-traced samples per pixel (spp) for each image. For TSMIS, the spp are always a multiple of A.

Comparison to ReSTIR We compare our approach to ReSTIR [BWP*20] using the RTXDI production implementation in figure 7. RTXDI offers different candidate generation methods and also biased and unbiased variants. We show comparisons against RTXDI using just one BSDF as well as additionally 24 next event estimation (NEE) candidates. While the latter gives drastic quality improvements to ReSTIR, the former is more directly comparable to our approach, given that we rely purely on BSDF samples to find new features. We observe that our approach is competitive to ReSTIR in certain areas that are illuminated by few light sources (blue inset), while regions more evenly lit by many light sources are much darker in comparison. The unbiased variant of ReSTIR using a single BSDF candidate exhibits more darkening due to stronger noise. We also apply SVGF to all approaches. While the results are comparable, our approach tends to produce more splotches, likely induced by outlier samples. Execution times are generally similar, where our parameterization is slightly faster than the single BSDF variant. The other variants of ReSTIR are more expensive due to additional processing.

Glossy Surfaces Figure 8 shows how the algorithm adapts to product sampling situations with a glossy BSDF even when the initial samples are cosine distributed.

Caustics Our independent sampling scheme allows the application to rendering of caustics. This is shown with both our Quakespasm (fig. 9) and Falcor (fig. 10) implementations. The dynamic adapta-

Figure 7: *Comparison of spatial SMIS against variants of RTXDI, a production implementation of ReSTIR [BWP*20] with and without application of a denoiser (SVGF [SKW*17]). Our approach is competitive in areas illuminated primarily by few light sources. Biased variants of ReSTIR retain image brightness better in complex regions.*

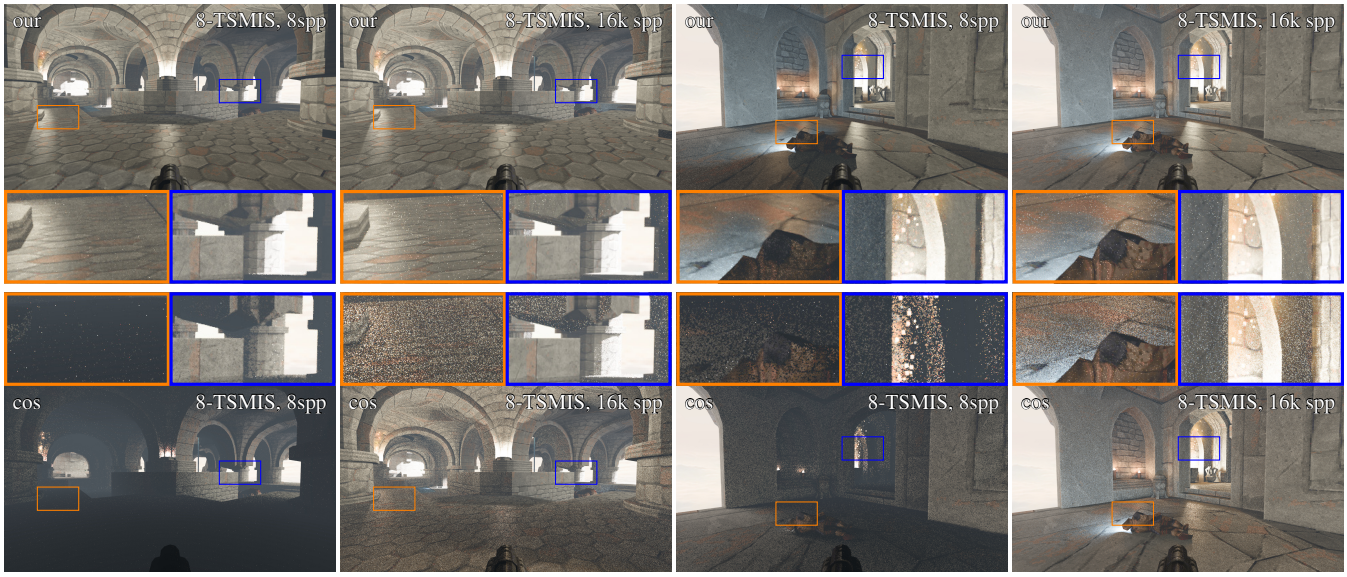

Figure 8: *Glossy materials rendered with vMF lobe or cosine-hemisphere sampling only, none of the approaches use BSDF importance sampling. Using the BSDF in the score function results in effective product sampling. Frame times on an Nvidia RTX 3080 Ti were 15ms and 9ms. Unlike fig. 11, these images are acquired after a few frames of adaptation, so the remaining image errors are due to SMIS estimating the PDF from a small subset of lobes. TAA is used to improve edge rendition.*

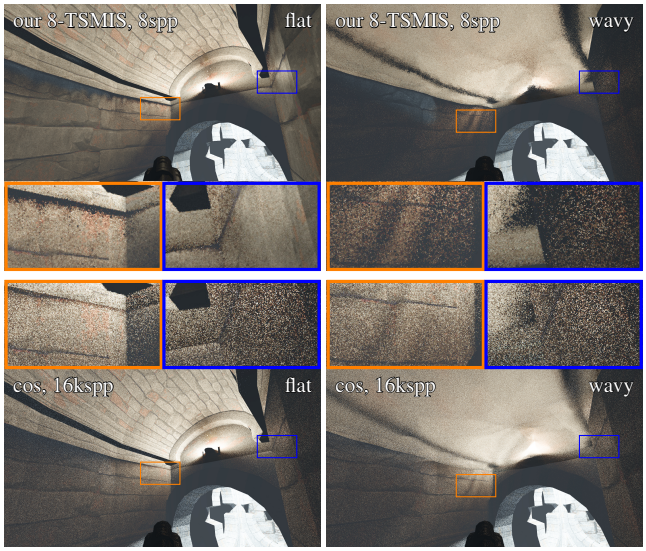

Figure 9: *Underwater renders showing caustics cast from the torch light at the back wall to the underwater wall on the left. The wavy water surface consists of procedurally animated noise with 15 octaves, the torch emits animated particles. Frame time was 71ms on an Nvidia RTX 2080 Ti and 29ms on 3080 Ti.*

tion is fast enough to track moving caustics due to moving emitter (Quakespasm, Falcor) and animated water surface (Quakespasm).

Temporal behavior in dynamic scenes We have tailored the method to quickly react to dynamic changes. We demonstrate this with our Quakespasm implementation in fig. 11. Over the span of 48 frames, a rocket is launched (frame 12), leaving a trail of emitting particles. Just before the rocket impacts the wall (frame 60), the walls already receive stronger illumination compared to cosine sampling. However, the sampling distribution is still not optimal, given that the accumulated image is much brighter in comparison.

Mixing behavior We observe that the Markov chain process exhibits a stronger preference for the light source with the most contribution as can be seen in fig. 12. The light sources are colored to make correspondence of samples more obvious. We can clearly see that the color does not follow a smooth transition when moving between light sources, but rather a defined barrier, which is blurred to some degree through neighborhood resampling. This is a limitation of our current approach and leads to increased variance, especially where many light sources contribute similarly to a surface region.

Effect of SMIS set size in many-lights scenarios A core issue with SMIS is that every feature needs to be adequately represented in order to not produce outlier (firefly) samples. In a mixture model, this boils down to one component for every prominent feature. While our spatio-temporal mixture has a very large number of components, the use of SMIS effectively reduces this set to a small fixed-size stochastic selection during evaluation. This issue becomes apparent when surface regions are evenly lit by many light sources, as depicted in the blue insets of fig. 13 or the left part of fig. 14. The result is a general darkening of the image with an increase in outlier samples. Note that the technique is unbiased, thus more energy is focused in the outlier samples. Increasing the SMIS sample set size reduces this issue, such that the image approaches the brightness of the reference. Intuitively, the likelihood of a sample being represented by a vMF lobe increases when the set size is increased, thus leading to fewer outliers and an increase in brightness. When a surface is primarily illuminated by a single light source (orange inset), small SMIS sample set sizes suffice.

Proximity bias when sharing lobes between pixels Figure 15 illustrates a situation where bright lights are propagated between pixels but are occluded in some of them. In this case re-using the estimate from the neighboring pixels is sub-optimal because sampling efforts will be spent on invisible lights. This degrades the quality of the estimator but does not cause bias: this can be observed by looking at the orange inset, where SVGF successfully averages the noisy fringe around the edge closely to the correct value. It fails to do the same for shadow edges since there is no auxiliary buffer to support this edge (blue insets).

Unbiasedness We prove unbiasedness of our technique in appendix A. In fig. 16 we accompany this proof with a log-log convergence plot when rendering the Cornell box scene. Note that local permutations of Markov chain states are needed to attain an unbiased estimator as discussed in section 4.2.

7. Discussion, Limitations and Future Work

In the following section we summarize our key findings and lay out the limitations of our approach and possible future work.

Optimality of importance sampling Since we do not apply Metropolis-Hastings, we do not know the actual distribution of the lobes $p(\mathbf{t})$. While it is great to enjoy the freedom to tweak the transition process to ones individual needs (prefer fast, reactive adaptation for dynamic environments) it would certainly be interesting to devise a Markov transition that obeys detailed balance and steers the equilibrium exactly towards optimal importance sampling. We expect this to perform worse in dynamic environments but better on static scenes in the long run.

Markov chain property Since we employ a maximum likelihood step which is storing intermediate values which are fed by multiple previous samples, the sampling is non-Markovian in path space (it is Markovian in a state space that includes the intermediates). As adaptive MCMC [HST01] we can see that the chain is still ergodic and converges to a unique equilibrium if the adaptation vanishes with increasing mutation count, i.e. the maximum likelihood estimate converges.

Many lights sampling While the use of SMIS is a crucial aspect in the evaluability of the model, any features not present in the set during evaluation can and do easily lead to outlier samples. This makes our approach less practical for scenes with millions of disjoint light sources. While an increase in the SMIS set size offsets this issue to some degree, it becomes increasingly expensive to evaluate, which results in a practical limit. With SSMIS, a set size of 32 posed a

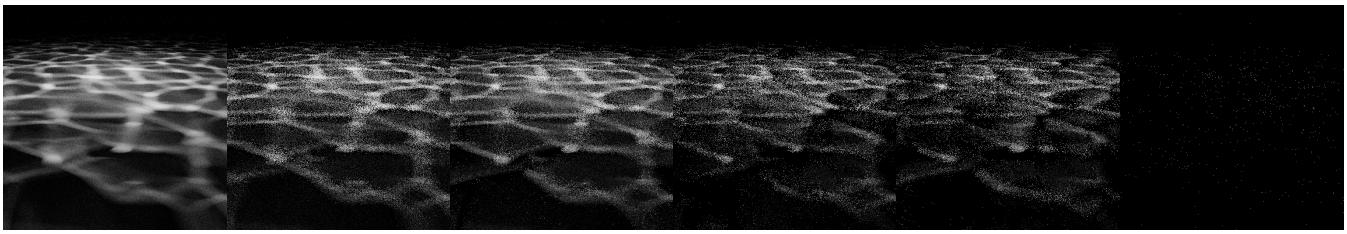

Figure 10: *Caustics induced by a small area light over a water surface. When the light is static, our technique (8-SMSIS) can resolve the caustics robustly. When the light is moving, adaptation depends on how long sample history is kept. Higher α-values (shorter history) allow the technique to readapt quicker.*

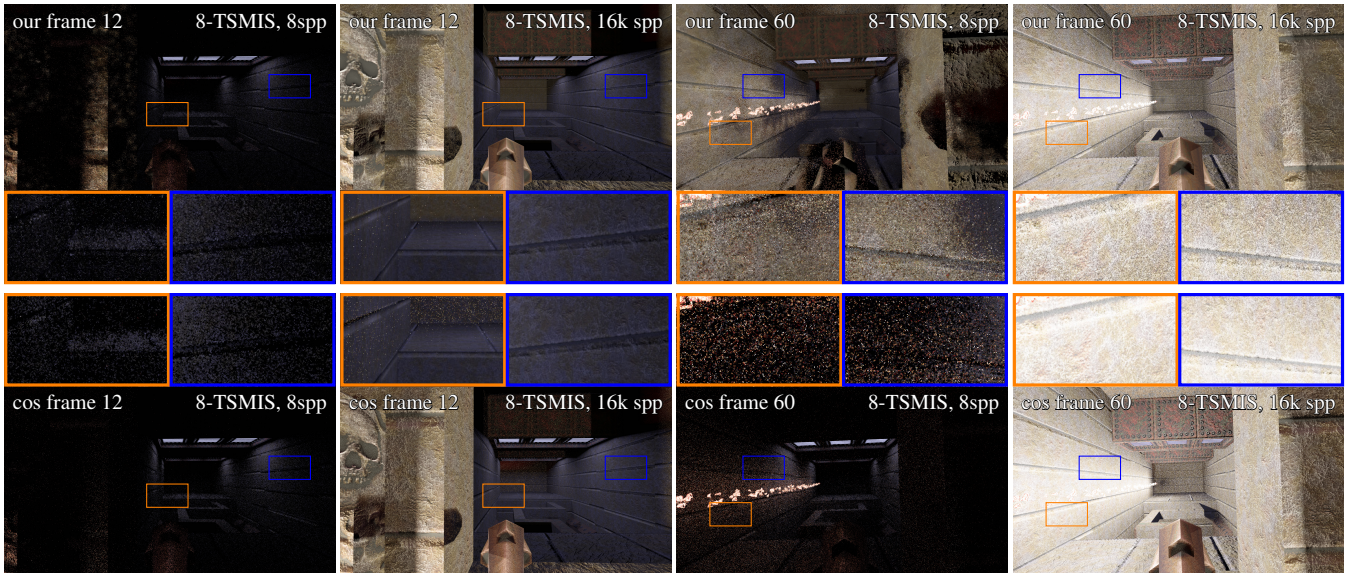

Figure 11: *Testing temporal adaptation: this figure shows the same scene at different points in time: frame 12 and frame 60. In frame 12 the rocket has not yet been fired, to illustrate the darkness of the environment. In frame 60, the rocket has not yet impacted in the back wall, so the emitting particles of the trail are a dynamic change. Our algorithm picked up the light sources already, but has not yet converged to a good sampling distribution: this can be observed by looking at the accumulated 2k frames versions which are notably brighter. Frame times were 15–16ms on an Nvidia RTX 2080 Ti and 8–9ms on 3080 Ti.*

negligible overhead, so it can likely be increased a bit further without too much cost.

Next Event Estimation (NEE) Practical renderers usually incorporate some form of direct light sampling (NEE) for increased robustness. While we do not make use of NEE to stress test our algorithm, NEE can be combined in the same fashion as BSDF or cosine sampling. While the usage of SMIS increases variance compared to evaluating the full mixture (which is intractable in our case), cases that are reliably sampled by NEE (e.g. small, distant light sources) will still see large variance reductions. Furthermore, if NEE can reliably sample most relevant light sources, our technique could focus the mixture on the harder cases [KŠV*19].

Indirect illumination In principle, our approach is not limited to guiding of direct illumination for primary surfaces. This is only due to our choice of a screen space data structure for simplicity. Guiding of direct illumination for deeper path vertices can be achieved by moving towards a world-space data structure in a similar fashion [Boi21]. Guiding of indirect illumination additionally requires changes in data gathering (backtracking of paths to compute contribution for all path vertices). Furthermore, incoming radiance becomes a stochastic quantity which likely has side-effects on the fitting process.

8. Conclusions

We have presented a lightweight guiding algorithm for sampling direct illumination based on a spatio-temporal randomized Markov

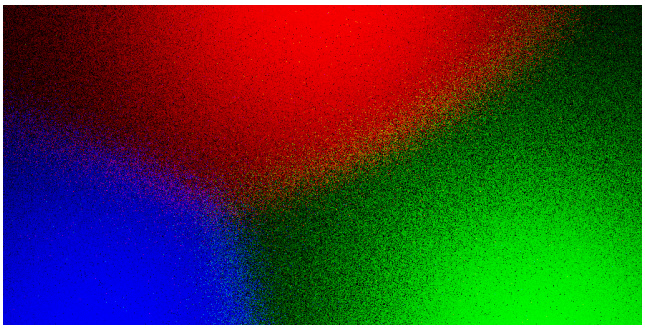

Figure 12: *Mixing behavior of three colored light sources. The Markov chain process has a stronger preference for the biggest contributor. This is hidden to some degree through neighborhood resampling as can be seen in the band that forms between the influence regions of the lights. Note that the asymmetry stems from using luminance in the scoring term.*

chain mixture model. We laid the theoretical foundation to give the Markov chain process full flexibility in its state transitions if desired. We demonstrated this by steering it towards tracking of dynamic lights, at the cost of non-optimal variance reduction which manifests itself in this setting as a tendency of overrepresenting of important light sources. We hope to inspire future work where Markov chain transitions are tailored for other use-cases such as offline rendering. While we exemplarily demonstrate simple animated caustics using our algorithm, we believe that our randomized mixture model is also applicable to, and beneficial for sampling other lighting effects.

9. Acknowledgements

Open Access funding enabled and organized by Projekt DEAL.

References

[Arv95] ARVO, JAMES R. "Stratified Sampling of Spherical Triangles". *Annual Conference Series (Proc. SIGGRAPH)*. Aug. 1995, 437–438. DOI: 10/bdrqbf 3.

[BDC12] BASHFORD-ROGERS, THOMAS, DEBATTISTA, KURT, and CHALMERS, ALAN. "A significance cache for accelerating global illumination". *Computer Graphics Forum*. Vol. 31. 6. Wiley Online Library. 2012, 1837–1851 3.

[BG88] BAGCHI, PARTHASARATHY and GUTTMAN, IRWIN. "Theoretical considerations of the multivariate von Mises-Fisher distribution". *Journal of Applied Statistics* 15.2 (Jan. 1988), 149–169 4.

[BHO10] BANGERT, MARK, HENNIG, PHILIPP, and OELFKE, UWE. "Using an infinite von Mises-Fisher mixture model to cluster treatment beam directions in external radiation therapy". *Conference on Machine Learning and Applications*. 2010 4.

[BJW21] BOKSANSKY, JAKUB, JUKARAINEN, PAULA, and WYMAN, CHRIS. "Rendering Many Lights with Grid-Based Reservoirs". *Ray Tracing Gems II: Next Generation Real-Time Rendering with DXR, Vulkan, and OptiX*. 2021, 351–365. DOI: 10.1007/978-1-4842-7185-8_23 3.

[Boi21] BOISSÉ, GUILLAUME. "World-Space Spatiotemporal Reservoir Reuse For Ray-Traced Global Illumination". *SIGGRAPH Asia 2021 Technical Communications*. 2021. DOI: 10.1145/3478512.3488613 3, 11.

[Boo07] BOOTH, T. E. "Unbiased Monte Carlo estimation of the reciprocal of an integral". *Nuclear Science and Engineering* 156.3 (2007), 403–407 3, 4.

[BSMD21] BASHFORD-ROGERS, THOMAS, SANTOS, LUÍS PAULO, MARNERIDES, DEMETRIS, and DEBATTISTA, KURT. "Ensemble Metropolis Light Transport". *ACM Trans. Graph.* 41.1 (Dec. 2021). ISSN: 0730-0301. DOI: 10.1145/3472294 3.

[BWP*20] BITTERLI, BENEDIKT, WYMAN, CHRIS, PHARR, MATT, et al. "Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting". *ACM TOG (Proc. SIGGRAPH)* 39.4 (July 2020). DOI: 10/gg8xc7 1, 3, 5, 8, 9.

[CGMR04] CAPPÉ, O, GUILLIN, A, MARIN, J. M, and ROBERT, C. P. "Population Monte Carlo". *Journal of Computational and Graphical Statistics* 13.4 (2004), 907–929. DOI: 10.1198/106186004X12803 3.

[CHA82] CHAO, M. T. "A general purpose unequal probability sampling plan". *Biometrika* 69.3 (Dec. 1982), 653–656. DOI: 10.1093/biomet/69.3.653 5.

[Der22] DEREVYANNYKH, MIKHAIL. "Real-Time Path-Guiding Based on Parametric Mixture Models". *Eurographics 2022 - Short Papers*. 2022. DOI: 10.2312/egs.20221024 2, 4, 8.

[DHD20] DITTEBRANDT, ADDIS, HANIKA, JOHANNES, and DACHSBACHER, CARSTEN. "Temporal Sample Reuse for Next Event Estimation and Path Guiding for Real-Time Path Tracing". *Proc. EGSR – DL-only Track*. 2020. DOI: 10.2312/sr.20201135 4, 8.

[DPÖM21] DODIK, ANA, PAPAS, MARIOS, ÖZTIRELI, CENGIZ, and MÜLLER, THOMAS. "Path Guiding Using Spatio-Directional Mixture Models". *CGF* 41.1 (2021), 172–189. DOI: 10.1111/cgf.14428 3.

[EC22] ELVIRA, VÍCTOR and CHOUZENOUX, ÉMILIE. "Optimized Population Monte Carlo". *IEEE Transactions on Signal Processing* 70 (2022), 2489–2501. DOI: 10.1109/TSP.2022.3172619 3.

[EK18] ESTEVEZ, ALEJANDRO CONTY and KULLA, CHRISTOPHER. "Importance Sampling of Many Lights with Adaptive Tree Splitting". *Proc. ACMCGIT* 1.2 (Aug. 2018), 25:1–25:17. DOI: 10/ggh89v 3.

[EMLB17] ELVIRA, VÍCTOR, MARTINO, LUCA, LUENGO, DAVID, and BUGALLO, MÓNICA F. "Population Monte Carlo schemes with reduced path degeneracy". *2017 IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*. 2017, 1–5. DOI: 10.1109/CAMSAP.2017.8313090 3.

[FHH*19] FASCIONE, LUCA, HANIKA, JOHANNES, HECKENBERG, DANIEL, et al. "Path Tracing in Production: Part 1: Modern Path Tracing". *ACM SIGGRAPH 2019 Courses*. 2019, 19:1–19:113. DOI: 10.1145/3305366.3328079 1.

[GWH20] GRUSON, ADRIEN, WEST, REX, and HACHISUKA, TOSHIYA. "Stratified Markov Chain Monte Carlo Light Transport". *Computer Graphics Forum* 39.2 (2020), 351–362. DOI: 10.1111/cgf.13935 3.

[HDF15] HANIKA, JOHANNES, DROSKE, MARC, and FASCIONE, LUCA. "Manifold Next Event Estimation". *CGF (Proc. EGSR)* 34.4 (June 2015), 87–97. DOI: 10.1111/cgf.12681 3.

[Hes95] HESTERBERG, TIM. "Weighted Average Importance Sampling and Defensive Mixture Distributions". *Technometrics* 37.2 (1995), 185–194. DOI: 10.1080/00401706.1995.10484303 15.

[HEV*16] HERHOLZ, SEBASTIAN, ELEK, OSKAR, VORBA, JIŘÍ, et al. "Product Importance Sampling for Light Transport Path Guiding". *CGF (Proc. EGSR)* (2016). DOI: 10/f842dt 3.

[HP02] HEY, HEINRICH and PURGATHOFER, WERNER. "Importance Sampling with Hemispherical Particle Footprints". *Proc. SCCG*. Apr. 2002, 107–114. DOI: 10/fmx2jp 3.

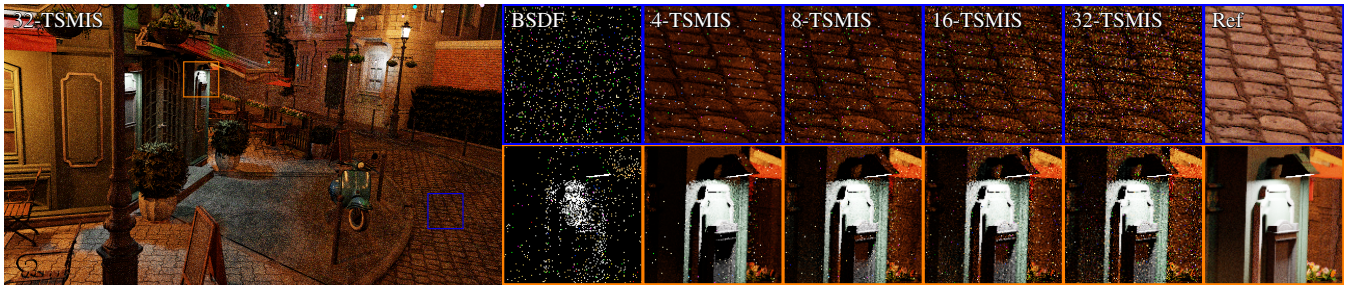

Figure 13: *32spp renderings using different SMIS sample set sizes. Surfaces illuminated by many light sources benefit most from larger SMIS sample sets (blue inset), but smaller sets are sufficient otherwise (orange inset).*

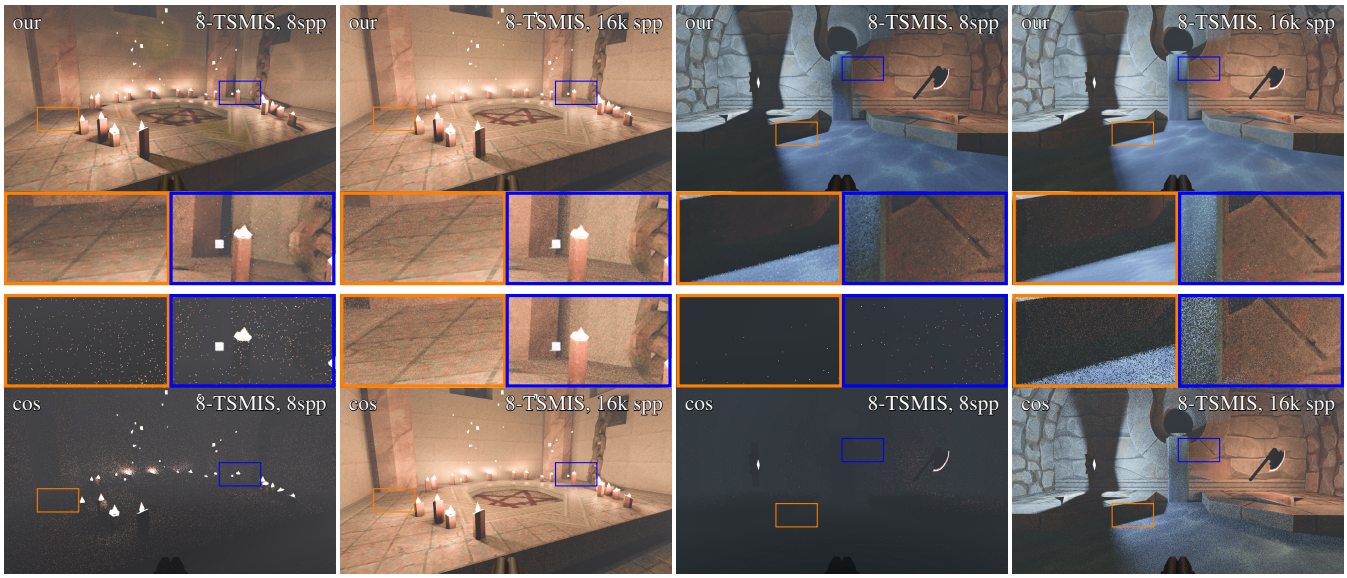

Figure 14: *Many light sources (left: candles and the animated spark particles emitted from them) pose a hard problem for the stochastic lobe mixtures. Right: two smoothly overlapping lights (blue and red) showing the transition between two dominant lobes. Unlike fig. 11, these images are acquired after a few frames of adaptation, so the remaining image errors are due to SMIS estimating the PDF from a small subset of lobes. TAA is used to improve edge rendition. Frame times were 14.9ms (candles) and 18ms (dual lights) on an Nvidia RTX 2080 Ti, and 8ms and 10ms respectively on a 3080 Ti.*

[HPM*20] HART, DAVID, PHARR, MATT, MÜLLER, THOMAS, et al. "Practical Product Sampling by Fitting and Composing Warps". Vol. 39. 4. 2020, 149–158. DOI: 10.1111/cgf.14060 3.

[HST01] HAARIO, HEIKKI, SAKSMAN, EERO, and TAMMINEN, JOHANNA. "An adaptive Metropolis algorithm". *Bernoulli* 7.2 (2001), 223–242. DOI: 10.2307/3318737 3, 10.

[HTD21] HANIKA, JOHANNES, TESSARI, LORENZO, and DACHSBACHER, CARSTEN. "Fast temporal reprojection without motion vectors". *JCGT* 10.3 (Sept. 2021), 19–45. ISSN: 2331-7418. URL: http://jcgt.org/published/0010/03/02/ 7.

[HZE*19] HERHOLZ, SEBASTIAN, ZHAO, YANGYANG, ELEK, OSKAR, et al. "Volume Path Guiding Based on Zero-Variance Random Walk Theory". *ACM TOG* 38.3 (June 2019). DOI: 10.1145/3230635 3.

[Jak12] JAKOB, WENZEL. *Numerically stable sampling of the von Mises-Fisher distribution on S^2 (and other tricks)*. Tech. rep. Cornell University, July 2012 2.

[Jen95] JENSEN, HENRIK WANN. "Importance driven path tracing using the photon map". *Rendering Techniques (Proc. EGWR)*. 1995 3.

[Kaj86] KAJIYA, JAMES T. "The Rendering Equation". *Computer Graphics (Proc. SIGGRAPH)* 20.4 (Aug. 1986), 143–150. DOI: 10/cvf53j 2.

[KCK*22] KALLWEIT, SIMON, CLARBERG, PETRIK, KOLB, CRAIG, et al. *The Falcor Rendering Framework*. Aug. 2022 2, 7.

[KŠV*19] KARLÍK, ONDŘEJ, ŠIK, MARTIN, VÉVODA, PETR, et al. "MIS Compensation: Optimizing Sampling Techniques in Multiple Importance Sampling". *ACM Trans. Graph. (SIGGRAPH Asia 2019)* 38.6 (2019). DOI: 10.1145/3355089.3356565 11.

[LKB*22] LIN, DAQI, KETTUNEN, MARKUS, BITTERLI, BENEDIKT, et al. "Generalized Resampled Importance Sampling: Foundations of ReSTIR". *ACM TOG (Proc. SIGGRAPH)* 41.4 (July 22, 2022), 75:1–75:23. DOI: 10/gqjn7b 3.

[Lum17] LUMBERYARD, AMAZON. *Amazon Lumberyard Bistro, Open Research Content Archive (ORCA)*. July 2017 8.

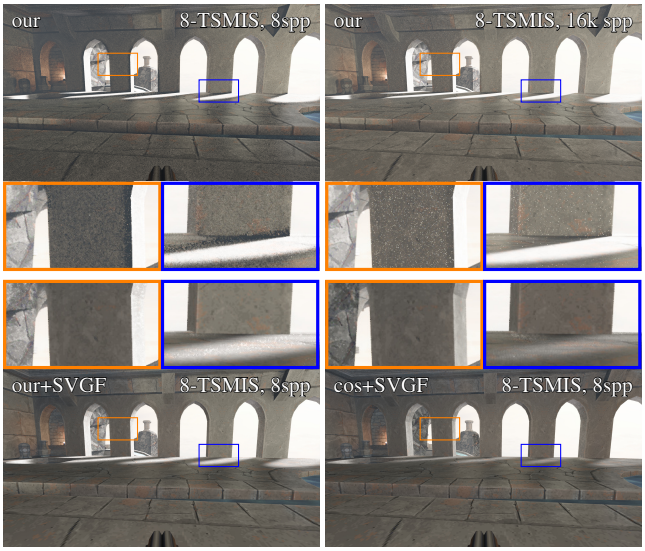

Figure 15: *Proximity bias when estimating lobes from neighbor pixels manifests itself as high-variance fringes around edges or shadow boundaries. A seemingly very important light propagates quickly to the neighbor pixels but then turns out to be occluded. In our estimator this does not cause bias but can lead to blurry contact shadows after applying SVGF (bottom images). Frame time was 20ms on an Nvidia RTX 2080 Ti and 10ms on a 3080 Ti.*

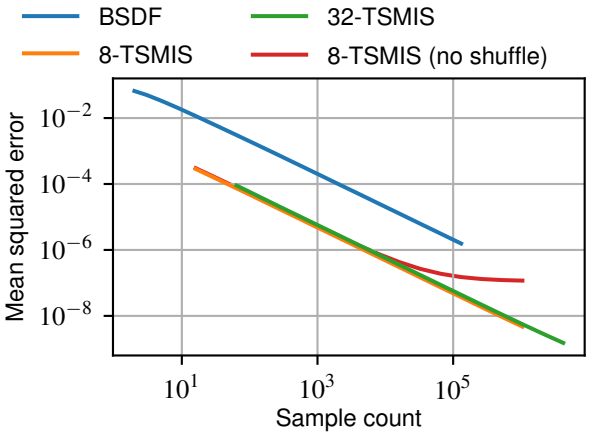

Figure 16: *Log-log plot showing the convergence of our technique when rendering a Cornell box scene (orange line). Note that due to the simple lighting, using a larger SMIS sample set size does not reduce the error further (green line). Performing no state shuffling results in a biased estimator (red line).*

[LW95] LAFORTUNE, ERIC P. and WILLEMS, YVES D. "A 5D tree to reduce the variance of Monte Carlo ray tracing". *Rendering Techniques*. 1995, 11–20 3.

[LWY21] LIN, DAQI, WYMAN, CHRIS, and YUKSEL, CEM. "Fast Volume Rendering with Spatiotemporal Reservoir Resampling". *ACM TOG (Proc. SIGGRAPH)* 40.6 (Dec. 10, 2021), 279:1–279:18. DOI: 10/grrjd6 3.

[MELC15] MARTINO, L., ELVIRA, V., LUENGO, D., and CORANDER, J. "Layered adaptive importance sampling". *Statistics and Computing* 27.3 (2015), 599–623 3, 15.

[MGN17] MÜLLER, THOMAS, GROSS, MARKUS, and NOVÁK, JAN. "Practical Path Guiding for Efficient Light-Transport Simulation". *CGF (Proc. EGSR)* 36.4 (June 2017), 91–100. DOI: 10/gbnvrs 3.

[MP92] MARINARI, E. and PARISI, G. "Simulated Tempering: A New Monte Carlo Scheme". *Europhysics Letters* 19.6 (July 1992), 451. DOI: 10.1209/0295-5075/19/6/002 3.

[MPC19] MOREAU, PIERRE, PHARR, MATT, and CLARBERG, PETRIK. "Dynamic Many-Light Sampling for Real-Time Ray Tracing". *Proc. HPG – Short Papers*. 2019. DOI: 10/ggh89m 3.

[NVI18] NVIDIA. *NVIDIA Turing GPU Architecture*. 2018 1.

[OLK*21] OUYANG, YAOBIN, LIU, SHIQIU, KETTUNEN, MARKUS, et al. "ReSTIR GI: Path Resampling for Real-Time Path Tracing". *CGF* 40.8 (2021), 17–29. DOI: 10/gqwmdx 3.

[Pan20] PANTALEONI, JACOPO. "Online path sampling control with progressive spatio-temporal filtering". *CoRR* abs/2005.07547 (2020). arXiv: 2005.07547 4.

[Pet21] PETERS, CHRISTOPH. "BRDF Importance Sampling for Polygonal Lights". *ACM TOG (Proc. SIGGRAPH)* 40.4 (July 2021), 140:1–140:14. DOI: 10/gqjn8c 3.

[PJH23] PHARR, MATT, JAKOB, WENZEL, and HUMPHREYS, GREG. *Physically Based Rendering: From Theory to Implementation*. 4th. The MIT Press, 2023. ISBN: 978-0-26-204802-6 1.

[RGH*20] RATH, ALEXANDER, GRITTMANN, PASCAL, HERHOLZ, SEBASTIAN, et al. "Variance-Aware Path Guiding". *ACM TOG (Proc. SIGGRAPH)* 39.4 (Aug. 2020). DOI: 10.1145/3386569.3392441 3.

[RHJD18] REIBOLD, FLORIAN, HANIKA, JOHANNES, JUNG, ALISA, and DACHSBACHER, CARSTEN. "Selective Guided Sampling with Complete Light Transport Paths". *ACM TOG (Proc. SIGGRAPH Asia)* 37.6 (Dec. 2018), 223:1–223:14. DOI: 10/gf2g93 3.

[RHL20] RUPPERT, LUKAS, HERHOLZ, SEBASTIAN, and LENSCH, HENDRIK P. A. "Robust Fitting of Parallax-Aware Mixtures for Path Guiding". *ACM TOG (Proc. SIGGRAPH)* 39.4 (July 8, 2020). DOI: 10/gg8xc6 2–4, 7, 8.

[RS20] RUDOLF, DANIEL and SPRUNGK, BJÖRN. "On a Metropolis-Hastings importance sampling estimator". *Electronic Journal of Statistics* 14.1 (2020), 857–889. DOI: 10.1214/20-EJS1680 3.

[SHJD22] SCHÜSSLER, VINCENT, HANIKA, JOHANNES, JUNG, ALISA, and DACHSBACHER, CARSTEN. "Path Guiding with Vertex Triplet Distributions". *CGF (Proc. EGSR)* 41.4 (2022). DOI: 10.1111/cgf.14582 3.

[SK21] SCHUSTER, INGMAR and KLEBANOV, ILJA. "Markov Chain Importance Sampling—A Highly Efficient Estimator for MCMC". *Journal of Computational and Graphical Statistics* 30.2 (2021), 260–268. DOI: 10.1080/10618600.2020.1826953 3.

[SKW*17] SCHIED, CHRISTOPH, KAPLANYAN, ANTON, WYMAN, CHRIS, et al. "Spatiotemporal Variance-Guided Filtering: Real-Time Reconstruction for Path-Traced Global Illumination". *Proc. HPG*. 2017, 2:1–2:12. DOI: 10/ggd8dg 1, 9.

[SLK*22] SAWHNEY, ROHAN, LIN, DAQI, KETTUNEN, MARKUS, et al. *Decorrelating ReSTIR Samplers via MCMC Mutations*. 2022. arXiv: 2211.00166 [cs.GR] 3.

[SSA05] STARK, MICHAEL, SHIRLEY, PETER, and ASHIKHMIN, MICHAEL. "Generation of Stratified Samples for B-Spline Pixel Filtering". *Journal of Graphics Tools* 10.1 (2005), 39–48. DOI: 10.1080/2151237X.2005.10129186 5.

[SW86] SWENDSEN, ROBERT and WANG, JIAN-SHEN. "Replica Monte Carlo Simulation of Spin-Glasses". *Physical Review Letters* 57.21 (1986), 2607–2610. DOI: 10.1103/PhysRevLett.57.2607 3.

[TCE05] TALBOT, JUSTIN F., CLINE, DAVID, and EGBERT, PARRIS. "Importance Resampling for Global Illumination". *Proceedings of the Sixteenth Eurographics Conference on Rendering Techniques*. EGSR '05. Konstanz, Germany: Eurographics Association, 2005, 139–146. ISBN: 3905673231 3.

[VG97] VEACH, ERIC and GUIBAS, LEONIDAS J. "Metropolis Light Transport". *Annual Conference Series (Proc. SIGGRAPH)*. Vol. 31. Aug. 1997, 65–76. DOI: 10/bkjqj4 3, 5.

[VHH*19] VORBA, JIŘÍ, HANIKA, JOHANNES, HERHOLZ, SEBASTIAN, et al. "Path Guiding in Production". *ACM SIGGRAPH 2019 Courses*. 2019, 18:1–18:77. DOI: 10.1145/3305366.3328091 3.

[VKŠ*14] VORBA, JIŘÍ, KARLÍK, ONDŘEJ, ŠIK, MARTIN, et al. "On-line Learning of Parametric Mixture Models for Light Transport Simulation". *ACM TOG (Proc. SIGGRAPH)* 33.4 (Aug. 2014) 2–4.

[WGGH20] WEST, REX, GEORGIEV, ILIYAN, GRUSON, ADRIEN, and HACHISUKA, TOSHIYA. "Continuous Multiple Importance Sampling". *ACM TOG (Proc. SIGGRAPH)* 39.4 (Aug. 2020). DOI: 10.1145/3386569.3392436 3, 4, 15.

[WND*14] WILKIE, ALEXANDER, NAWAZ, SEHERA, DROSKE, MARC, et al. "Hero Wavelength Spectral Sampling". *CGF (Proc. EGSR)* 33.4 (July 2014), 123–131 3.

[Yuk19] YUKSEL, CEM. "Stochastic Lightcuts". *Proc. HPG – Short Papers*. 2019. DOI: 10/ggjcgw 3.

[ZGJ20] ZELTNER, TIZIAN, GEORGIEV, ILIYAN, and JAKOB, WENZEL. "Specular Manifold Sampling for Rendering High-Frequency Caustics and Glints". *ACM TOG (Proc. SIGGRAPH)* 39.4 (July 8, 2020). DOI: 10/gg8xc8 3, 4.

[ZZ18] ZHENG, QUAN and ZWICKER, MATTHIAS. "Learning to Importance Sample in Primary Sample Space". *CoRR* abs/1808.07840 (2018). arXiv: 1808.07840 3.

Appendix A: Unbiasedness of the SMIS Estimator

Our derivation for unbiasedness closely relates to the proof found in the SMIS paper [WGGH20, Appendix B]. The sampling technique controlled by the Markov chain process is denoted t_i and x_i are the samples generated by this technique. In SMIS, the derivations start:

$$E[\langle I\rangle_{SMIS}] = E[\langle I\rangle_{SMIS}]_{(t_1,x_1),\dots,(t_n,x_n)} \tag{10}$$

$$= E[E[\langle I\rangle_{SMIS}]_{x_1,\dots,x_n}]_{t_1,\dots,t_n} \tag{11}$$

which is valid in our case because, as indicated by the arrows, the x_i depend on their respective t_i, but not the other way around: the t_i do not depend on any x. They are instead advanced by their respective Markov chain process, i.e. there is a dependency of t_{i+1} on t_i. The independence of the t_i among each other is not required to perform the step between eq. (10) and eq. (11), as long as the x_i are independent between one another. This is also employed by previous work [MELC15, Sec. 5.1], which explicitly states the importance of the independence of the mixture parameters t of the samples x. They also note two requirements for a consistent estimator. The first one is that the proposal densities $p(x|t)$ (in our notation) are *all* heavier-tailed than the target distribution. This is a simple way of expressing the need to cover the whole domain where the target is non-zero. In our case, every stochastic mixture always includes a defensive sampling lobe (BSDF or cosine hemisphere) [Hes95]. Their second requirement is fulfilled by using MIS with the balance heuristic (deterministic mixture MIS in their terms).

We want to point out that the dependency of t_i on $t_j, j < i$ inside one pixel is weak in our case. The subgroup shuffle will eliminate any dependency completely. A small amount is reintroduced by the reuse of a neighborhood of states in a 91×91 window around a pixel. In the case of 8-TSMIS this means we will randomly select 8 out of a set of 8281 candidates, keeping the probability of collisions very low.

$$eq.\ (11) = E\left[E\left[\sum_{i=1}^{n} w(x_i,t_i)\frac{f(x_i)}{p(x_i|t_i)}\right]_{x_1,\dots,x_n}\right]_{t_1,\dots,t_n} \tag{12}$$

$$= E\left[\underbrace{\sum_{i=1}^{n}\int_{\mathcal{X}} w(x,t_i)\frac{f(x)}{\cancel{p(x|t_i)}}\cancel{p(x|t_i)}\,\mathrm{d}x}_{=I,\text{ discrete MIS}}\right]_{t_1,\dots,t_n} = I. \tag{13}$$

This last step reduces the problem to discrete MIS using the stochastically chosen techniques t_i. Note that it also shows that already the inner expression is the sought-for integral, taking the expectation around it does not change the value, i.e. there is no interplay between different sets of t_i to correct the outcome, in particular independence of t_i is not required. The one requirement it imposes is that every stochastic mixture (set of t_i) covers the full domain where $f(x) > 0$, which we achieve by defensive sampling.