Product Importance Sampling for Light Transport Path Guiding

Sebastian Herholz[1]   Oskar Elek[2]   Jiří Vorba[2,3]   Hendrik Lensch[1]   Jaroslav Křivánek[2]

[1]Tübingen University   [2]Charles University Prague   [3]Weta Digital

Figure 1: *Sampling quality for the* KITCHENETTE *scene containing numerous anisotropic BRDFs. Our product sampling produces a visibly smoother image compared to Vorba et al. [VKŠ*14] at 512 samples per pixel.*

Abstract

The efficiency of Monte Carlo algorithms for light transport simulation is directly related to their ability to importance-sample the product of the illumination and reflectance in the rendering equation. Since the optimal sampling strategy would require knowledge about the transport solution itself, importance sampling most often follows only one of the known factors – BRDF or an approximation of the incident illumination. To address this issue, we propose to represent the illumination and the reflectance factors by the Gaussian mixture model (GMM), which we fit by using a combination of weighted expectation maximization and non-linear optimization methods. The GMM representation then allows us to obtain the resulting product distribution for importance sampling on-the-fly at each scene point. For its efficient evaluation and sampling we preform an up-front adaptive decimation of both factor mixtures. In comparison to state-of-the-art sampling methods, we show that our product importance sampling can lead to significantly better convergence in scenes with complex illumination and reflectance.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—; I.3.3 [Computer Graphics]: Picture/Image Generation—;

1. Introduction

Importance sampling is an essential component of efficient Monte Carlo light transport simulations. Estimating the illumination at a scene location involves sampling the integrand of the rendering equation [Kaj86], defined by the product of the incident radiance and the cosine-weighted reflectance function (BRDF). However, since the incident radiance is unknown upfront—as it corresponds to the transport solution itself—the traditional approach to importance sampling is to distribute samples proportionally to the BRDF [PH10].

On the other hand, multiple approaches— [Jen95, HP02], or more recently [VKŠ*14]—have shown that sampling according to even a rough estimate of the directional distribution of incident radiance can result in significant performance increases, especially in scenes with strongly varying illumination. However there is still large potential for improvement, since they do not sample proportionally to the full integrand, i.e., the product of radiance and BRDF.

In this paper we present a *product importance sampling* technique for estimating indirect illumination (Sec. 4). Our main goal is to find a good approximation to the illumination integrand as a whole, and use this approximation as a sampling distribution during rendering. With an accurate approximation, this approach ensures virtually optimal estimator convergence, while keeping the solution unbiased.

We propose to represent the distribution factors—BRDF and incident radiance—by two separately obtained *Gaussian mixtures* (see Fig. 2). For obtaining the BRDF factor, we design a pre-processing optimization step where all applicable BRDFs in the rendered scene are fitted for a densely sampled set of incident directions (Sec. 4.1). To estimate the incident radiance factor, we adapt the on-line learning framework of Vorba et al. [VKŠ*14] (Sec. 4). This in turn enables analytically calculating the product distribution as another Gaussian mixture that can be directly used for sampling. Given that the resulting product mixture has a quadratic number of components with respect to the input mixtures, we propose using an adaptive
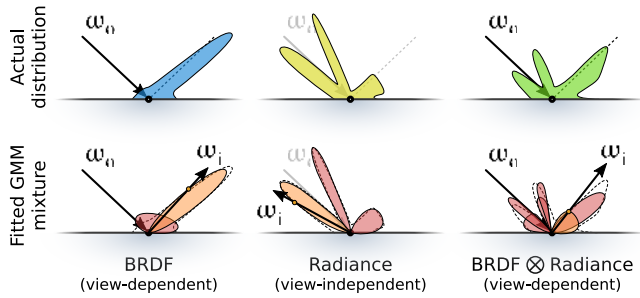
Figure 2: *Illustration of GMM fitting and our product approximation. Top row: target distributions that our method fits to. Bottom row: fitted GMMs and resulting product. While state-of-the-art methods (such as Vorba et al. [VKŠ*14]) multiple-importance-sample only the BRDF and/or incident radiance factors, our method samples proportionally to the full product (right). Therefore, unlike previous work, we cannot miss features of the integrand – provided that the factors are fitted accurately.*

decimation strategy to separately reduce the number of BRDF and radiance components (Sec. 4.2).

When used in rendering, the resulting method provides significant efficiency and sampling improvements, especially for scenes with strong glossy transport or difficult illumination (Fig. 1, Sec. 6). We also demonstrate that a slow, generic non-linear optimization, traditionally used to fit Gaussian kernels to BRDFs, can be side-stepped by a substantially more efficient expectation-maximization approach [DLR77, Bis06].

2. Related Work

This section gives an overview of related work on importance sampling in Monte Carlo (MC) light transport and the use of Gaussians in rendering.

Monte Carlo light transport. MC methods for light transport aim to solve the rendering equation [Kaj86] (Eq. 1) at a given point in the scene. They approach this problem by stochastically sampling and evaluating the reflected incident illumination, tracing paths from the sensor [Kaj86] or light source [Arv86]. Unlike our caching-based approach, these MC methods do not take advantage of any precomputed illumination approximation to improve sampling.

Markov-chain MC approaches [VG97, Vea98, KSKAC02] use Metropolis sampling to explore the neighborhood of high-throughput paths, once found. They attempt to achieve globally optimal path sampling by mutating entire paths, but face the difficult problem of balancing global exploration and local exploitation of the path space, which can lead to uneven convergence behavior. In contrast, our approach uses appropriate—as close to optimal as possible—local sampling decisions at each intersection point, and does not suffer from this issue.

BRDF and illumination importance sampling. Paths with high contribution can be found by importance-sampling the product of surface BRDF and incoming illumination. The BRDF can be either

represented by an analytic model or by measured data. The advantage of the former representation is that for most models an analytic way for importance sampling exists [WMLT07, HD]. Since the size and dimensionality of measured BRDFs can be impractical, several methods have been presented to either factorize the data [LRR04] or fit them to analytical models to enable direct importance sampling. Given that our method creates Gaussian mixture fits for the BRDFs, it has the potential to represent both analytic and measured models (albeit our results use only analytic ones).

There is a significant body of works on importance-sampling of the incoming illumination [Jen95, LW95, HP02, PH10]. We follow the direction of Vorba et al. [VKŠ*14], which shows a great potential to improve the convergence of different MC-based integrators by using Gaussian mixtures to represent incoming illumination. Still, the methods in this category are limited by the fact that they only sample the reflectance and the illumination separately. Multiple importance sampling (MIS) [Vea98] can be used to weigh the contributions from both estimates based on their probability [VKŠ*14], but this still does not correspond to sampling the product directly, as is the case in our method.

Product importance sampling for direct illumination. Several works have addressed product importance sampling for *direct illumination* calculation. Unlike our setting, in this case the illumination is known upfront since it is represented either by an environment map or by a cloud of point lights [WÅ]. Some works use a projection of the illumination and reflectance into a functional basis such as wavelets [CJAMJ05, CAM] or spherical harmonics [JCJ09]; hierarchical sample warping is then used to sample from the product. Importance re-sampling methods [TCE05, BGH05, WÅ] are based on the assumption that generating direct illumination and BRDF samples is cheap relatively to visibility sampling. These algorithms thus propose N samples from both domains, and use the product information from them to draw $M \ll N$ final samples. All these methods are limited to sampling the product distribution only for direct illumination, while our presented method targets sampling from the product distribution of *indirect illumination* and BRDF.

Gaussian representations in rendering. Since the product of Gaussians can be integrated in closed form, Gaussian representations of individual factors of the rendering equation are useful to directly approximate the illumination integral. In real-time rendering, representations based on spherical Gaussians [WRG*09, XSD*13] can be used to approximate the direct illumination of microfacet-based materials due to distant environment maps, or point lights. The Gaussian approximation for the reflectance of several microfacet BRDF models can be derived in a closed form. Wang et al. [XCM*14] additionally provide support for local inter-reflections. In the context of volume rendering, Gaussian mixtures were used by Jakob et al. [JRJ11] to efficiently store and evaluate the (spatial) photon density inside a medium.

However, as the above methods use approximations for estimating the illumination directly, they produce biased results. In contrast, we only utilize the Gaussian approximation to guide importance sampling, and therefore do not suffer from any bias.

3. Background

This section briefly covers the notation and several theoretical concepts we build on: the rendering equation and its MC sampling, Gaussian mixture models, and their fitting.

Rendering equation (RE). The solution to global illumination on surfaces (i. e., in the absence of participating media) is described by the rendering equation [Kaj86]:

$$L(\mathbf{x}, \omega_{o}) = L_{E}(\mathbf{x}, \omega_{o}) + L_{R}(\mathbf{x}, \omega_{o})$$
$$L_{R} = \int_{\Omega} \rho(\mathbf{x}, \omega_{o}, \omega_{i}) \cdot L(\mathbf{x}, \omega_{i}) \cdot \cos\theta \, d\omega_{i}, \qquad (1)$$

where L_E is surface emittance and ρ is the bi-directional scattering distribution function (BRDF) on the upper-hemispherical domain Ω. An MC estimator stochastically samples the RE. For instance, path tracing [Kaj86] incrementally builds paths from the camera – at each interaction point $\mathbf{x}$ it estimates L_R by importance-sampling a single direction ω_i according to a hemispherical probability density function (PDF) $p : \Omega \mapsto \mathbb{R}$, and continues recursively:

$$\widehat{L}_{R} = \frac{\rho(\mathbf{x}, \omega_{o}, \omega_{i}) \cdot L(\mathbf{x}, \omega_{i}) \cdot \cos\theta}{p(\omega_{i})}. \qquad (2)$$

The variance of such a stochastic estimator corresponds to the ability of p to approximate the integrand; the ideal case of p matching the integrand would result in an estimator with minimal variance possible (see Apx. C for how we determine this). The primary contribution of our method—finding p as close to optimal as possible—is described in Sec. 4.

Gaussian mixture model (GMM). A d-dimensional GMM is defined as a convex combination of K d-dimensional Gaussian components [Bis06]:

$$\mathcal{G}(\mathbf{y}|\Theta) = \sum_{k=1}^{K} \pi_{k} \cdot \mathcal{N}(\mathbf{y}|\mu_{k}, \Sigma_{k}), \qquad (3)$$

where the means μ_k, symmetric positive-definite covariance matrices Σ_k, and mixing weights π_k are aggregated in the parameter vector Θ. For $\mathcal{G}$ to remain a valid PDF the mixing weights must be non-negative and satisfy $\sum_{k}^{K} \pi_{k} = 1$. We opt for the most general anisotropic variant of the components $\mathcal{N}$ in order to represent anisotropic distributions frequently encountered in both BRDFs and radiance distributions.

Given a good initial estimate, the GMM can be efficiently fitted to discrete data by the expectation maximization (EM) algorithm [DLR77]. This yields a GMM-represented density estimate on the data. By considering them as sample multiplicity, particle weights can be incorporated into the estimate (cf. [VKŠ*14] and references therein). In general, many variants of EM exist, although virtually all of them suffer from getting stuck in local maxima [Bis06]. Generic non-linear optimization methods can sometimes yield better fits, but at the cost of significantly increased computational effort.

4. Product Importance Sampling

On the high level, our approach strives to obtain a sampling PDF p for choosing a scattering direction that is as close as possible to the integrand of the rendering equation (Eq. 1). To achieve this (also

see Fig. 2, bottom), we first compute approximations to the product factors defined as

$$p_{\rho}(\omega_{o}|\omega_{i}, \mathbf{x}) \propto \rho(\mathbf{x}, \omega_{i}, \omega_{o}) \qquad (4)$$
$$p_{L}(\omega_{o}|\mathbf{x}) \propto L(\mathbf{x}, \omega_{o}) \cos\theta \qquad (5)$$

and then compute the resulting PDF as their product:

$$p \propto p_{\otimes} = p_{\rho} \otimes p_{L}. \qquad (6)$$

The remainder of this section describes in detail how we obtain the factor mixtures and finally evaluate the product. Please note that for sake of a better intuition about the sampling decisions, from this point on we use a swapped notation for the directions ω_i and ω_o, in comparison to Eqs. 1 and 2.
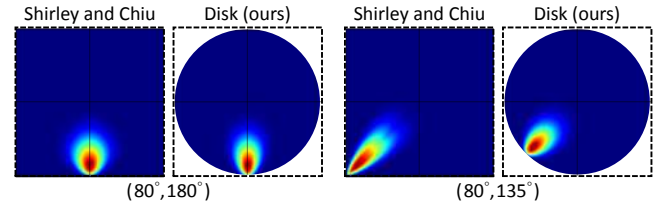
Figure 3: *Comparison of the 2D mappings for the PDF of a rough microfacet BRDF ($\alpha = 0.3$) at two incident directions. Unlike Shirley and Chiu, the disk mapping is shape-preserving under azimuthal rotations.*

GMM representation. As already sketched above, we represent the approximate sampling distributions from Eqs. 4 and 5 analytically using bi-variate Gaussian mixtures (Eq. 3). This enables us to take full advantage of the GM model: readily available fitting methods, closed-form sampling, trivial normalization, the ability to represent anisotropic distributions, and simple alignment of mixtures with different reference frames, which then enables a direct calculation of the product mixture (Sec. 4.2). Due to these reasons we opted for this representation over other possible models, such as anisotropic spherical Gaussians or von-Mises-Fisher distribution.

However, p_ρ and p_L are hemispherical distributions. To represent them in terms of the GMM, the hemispherical domain is mapped to a 2D Euclidean disk space using a low-distortion, area-preserving projective transformation (Fig. 3):

$$\mathbf{y} = \frac{\omega_{xy}}{\|\omega_{xy}\|\sqrt{1 - \omega_{z}}} \in \Omega \mapsto [-1, 1]^{2}. \qquad (7)$$

All computational steps of the algorithm—fitting, product calculation, and sampling—are then preformed in this domain.

Compared to the low-distortion map by Shirley and Chiu [SC97] used, e. g., by Vorba et al. [VKŠ*14], the disk mapping has the advantage of not distorting the mapping under azimuthal rotations (Fig. 3). We use this fact for the alignment of the BRDF and illumination mixtures during the product calculation, since they generally occupy two different reference frames. This can be performed by an azimuthal rotation around the local vertical axis, expressed by a 2×2 matrix $\mathbf{R}$: the parameters μ_k and Σ_k for each Gaussian component of one of the mixtures are transformed to $\mu'_{k} = \mathbf{R}\mu_{k}$ and $\Sigma'_{k} = \mathbf{R}\Sigma_{k}\mathbf{R}^{T}$.

Algorithmic overview. The pipeline of our method consists of three distinct steps:

1. *Pre-processing* (Sec. 4.1): for each material in the scene and a discrete set of incident directions ω_i, a GMM fit of the corresponding BRDF lobe is computed and stored in a BRDF cache. This results in a database of BRDF factor mixtures p_ρ (Eq. 4).

2. *Training*: over a number of passes, batches of bi-directional particles (i. e., photons and importons) are traced through the scene, being guided by the trained and cached distributions of all preceding passes. A dense illumination cache is built adaptively on-the-fly – each cache record stores a GMM representation of the incident radiance factor p_L (Eq. 5) valid for a small neighborhood. The representation is fitted by the EM algorithm – upon creation, each mixture is initialized from a KNN-estimate of local particle density, and then made more accurate by including traced particles from all subsequent training passes. This step follows the work of Vorba et al. [VKŠ*14]; please refer to the original paper for a detailed exposition.

3. *Rendering* (Sec. 4.2): paths from the sensor are guided through the scene by importance-sampling the product mixture $p_\otimes$ (Eq. 6), which is computed on-the-fly for every sampling decision from p_ρ and p_L by fetching the corresponding GMM caches.

4.1. BRDF Fitting

To be able to calculate the product between the illumination given in the GMM form, and the BRDF, a GMM representation of the latter is needed. This section describes how to obtain the BRDF GMM representation for all possible incident directions ω_i.

To our current knowledge no generic solution exists to represent an arbitrary BRDF by a mixture of Gaussians. For glossy BRDFs, such as Phong or microfacet models, Wang et al. [WRG*09] and Xu et al. [XSD*13] present a closed-form solution to represent normal distribution functions (NDFs) by a mixture of (anisotropic) spherical Gaussians. This NDF representation is then warped into the corresponding BRDF representation by rotating the mean and adjusting the covariance of the mixture components. However, the Fresnel and shadowing terms are considered to be constant over each mixture component and are only evaluated at their means. The warping of the Gaussian components also does not bound the spherical Gaussians to the upper hemisphere, allowing samples colliding with the surface to be created for certain configurations.

We strive to obtain as accurate and efficient product approximation as possible in this context. Consequently, the BRDF Gaussian mixture needs to encapsulate every single component defining it (Fresnel term, shadowing term, NDF, and hemispheric bound). We therefore trade the ability to derive the GMM representation in a closed form against a pre-processing step, which caches accurately fitted GMM representations of all BRDFs contained in the scene.

The following two sub-sections describe two different methods of fitting arbitrary BRDFs to the GMM. Since a BRDF can be a composite of a diffuse and a glossy part, we fit both parts separately and combine them after the fitting. A comparison of the quality of these fits is then presented in Secs. 4.1.2 and 6. The structure of the employed BRDF caches is explained in more detail in Sec. 4.1.3.

4.1.1. Weighted Expectation Maximiztion

Expectation maximization (EM) is used in the context of machine learning and classification to fit the GMM to distributions of observed data points. The EM algorithm maximizes the log-likelihood of the GMM parameter vector Θ given the observed dataset. Since EM fits to the density of the observed data instead of their associated values, Vorba et al. [VKŠ*14] extended it by adding weights to each observed sample representing the the observed function value.

To fit a GMM with K components to the glossy component of a BRDF a set of N samples is drawn from the BRDF using its standard sampling method. Weighted EM is then used to include the contribution of the samples weighted by $\rho(\omega)/p(\omega)$. This weighting also causes invalid samples not to influence the EM fit of the GMM (such as samples from the lower hemisphere).

Since the maximization of the log-likelihood of a GMM is a non-convex optimization problem, a good initialization of the GMM is essential for preventing the EM from converging to a sub-optimal local maximum. Instead of initializing the mixture parameters randomly, we found that using a quasi Monte-Carlo sampler to draw K samples using the BRDF PDF for the means, setting the component weights to $\pi_k = \frac{1}{K}$, and the covariance matrices to diagonal with $\Sigma_{ii} = 0.0125$ typically leads to near-optimal EM fits.

4.1.2. Non-linear Optimization

While the weighted EM algorithm is influenced by the BRDF values at the sampled data points, it is still only fitting the (weighted) PDF distribution of the BSDF, which does not correspond to the general curve-fitting problem we face. If a more accurate representation is needed and longer pre-processing time (e. g., one-to-two orders of magnitude) is available, a more sophisticated method such as non-linear optimization based on QR factorization—as implemented in the Ceres [AMO16] framework—can be used.

In the case of an ideal importance sampling the ratio between the BRDF $\rho(\omega)$ and its PDF $p(\omega)$ would be constant for every sample ω and the variance of these ratios would vanish (cf. Apx. C). Consequently, to fit as-optimal-as-possible GMM for importance sampling we choose the following objective function:

$$\sum_{j}^{N} \left[1 - \frac{\hat{\rho}(\omega_j)}{\mathcal{G}(\mathbf{y}_j | \Theta)} \right]^2 . \tag{8}$$

The parameters ω_j are the N directional samples drawn from the BRDF and $\mathbf{y}_j$ are the corresponding 2D positions in the GMM space (Eq. 7). The BRDF values for each sample are normalized ($\hat{\rho}(\omega_j) = \frac{\rho(\omega_j)}{\int \rho(\omega) \, d\omega}$, where $\int \rho(\omega) \, d\omega \approx \frac{1}{N} \sum_{j}^{N} \frac{\rho(\omega_j)}{p(\omega_j)}$) to represent a valid sampling PDF. Our fitted GMM matches this optimal PDF when the ratio between the normalized BRDF and the GMM is 1 at every sampling point.

To be able to robustly fit the GMM parameters Θ a re-parametrization needs to be done to ensure that the bounding conditions for a valid Gaussian mixture hold. The following constraints have to be considered: $\pi_k \geq 0$, $\Sigma_{xx} > 0$, $\Sigma_{yy} > 0$ and $|\Sigma| > 0$, where the parameters Σ_{xx} and Σ_{yy} are the diagonal entries of the 2D covariance matrix. To enforce these, we perform the following re-parametrization of the parameters π_k, μ_k and Σ_k of each Gaussian

component with respect to the optimization parameters w_k, m_k, G_k:

$$\pi_k = w_k^2 \qquad \Sigma_{xx_k} = G_{xx_k}^2 + \epsilon$$

$$\mu_{x_k} = m_{x_k} \qquad \Sigma_{xy_k} = G_{xy_k}$$

$$\mu_{y_k} = m_{y_k} \qquad \Sigma_{yy_k} = \frac{\sqrt{G_{xy_k}^2 - G_{t_k}^2}}{\sqrt{G_{xx_k}^2}} + \epsilon$$

To prevent the optimization from generating a Gaussian with a zero covariance matrix a regularization term of $\epsilon = 10^{-4}$ is added to the diagonal of Σ. Similarly as for the weighted EM, the non-linear optimization also needs a good initialization to avoid the convergence to non-optimal local minima. Therefore, even if the weighted EM does not converge to a perfect fit, it serves as a good initialization for the non-linear optimization and we use it as such.
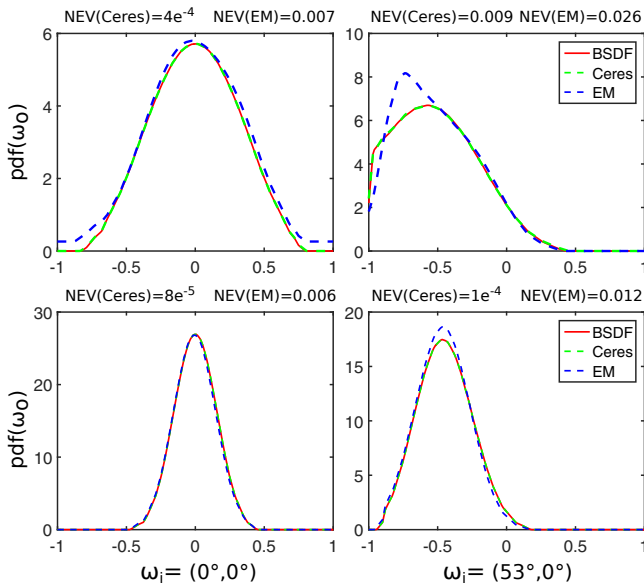

Figure 4: *1D slices of the mapped GMM PDF fits using weighted EM* (blue) *and Ceres* (green). *The fits represent a normalized rough conductor BRDFs: (top)* $\alpha = 0.3$ *and (bottom)* $\alpha = 0.15$. *Shown for two incident elevation angles.*

Fig. 4 shows a sample result of the two presented fitting methods. The non-linear optimization can achieve a tighter fit to the actual BRDF than the weighted EM, which—especially near grazing angles—can cause the fitted lobe to be skewed towards the hemispherical boundary. It is also visible that this effect relates to the roughness of the BRDF. At lower roughness values the distortion of the BRDF towards the boundary is not so significant and weighted EM can still produce suitable fits. The quality of these fitting methods in terms of their ability to efficiently importance-sample the BRDF is further analyzed in Sec. 6.1.

4.1.3. Caching

The caches for the GMM representations of the BRDF lobes can be categorized into three different types: diffuse, isotropic and anisotropic. The diffuse BRDF lobe is invariant w.r.t. incoming
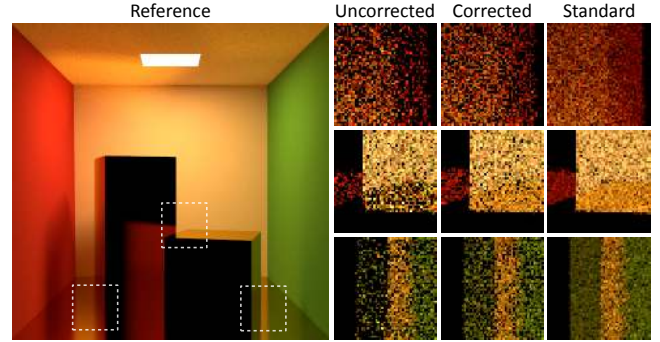
Figure 5: *Sample demonstration of our approximate translation for correcting GMM cache orientation (see Sec. 4.1.3). We compare BRDF sampling using our cached GM mixtures* without *and* with *the correction against the* standard *BRDF PDFs, in a basic path tracer (i.e., without product sampling and next event estimation) using 256 samples per pixel. Note that we purposely exaggerate the issue here by using a BRDF cache with only 16 records.*

directions, so only a single Gaussian mixture needs to be cached for each diffuse BRDF. Isotropic BRDFs are invariant to azimuthal rotations. To efficiently store these, only N GM mixtures for different elevation angles in $[0°, 90°]$ need to be stored. For an arbitrary incident direction ω_i the stored mixture with the closest elevation angle is used and rotated to the right azimuth using the formula from the beginning of Sec. 4. Finally, anisotropic BRDFs are not invariant to azimuthal rotations and therefore a cache entry for each hemispherical incident direction needs to be stored. To efficiently distribute and access the nearest cache positions, we use the spherical Fibonacci mapping [KISS15] to index M different cache positions on the upper hemisphere.

Because of the discrete nature of the BRDF cache, a record fetched for a query direction ω_i will generally not match it exactly. Given the inability to effectively interpolate and extrapolate GMM caches, we approximately correct this orientation mismatch by translating the mixture by a small offset in the 2D disk space. Note that a similar problem occurs when there is a mismatch between the principal axes of the illumination and BRDF caches. The translation offset corresponds to the difference between the projected reference and query directions. The effectiveness of this solution is shown in Fig. 5 for the case of BRDF cache correction.

4.2. Product Calculation and Sampling

Our method bases its sampling decisions on the product distribution, as sketched at the beginning of Sec. 4.

Since the BRDF mixture (and hence the product mixture as well) depends on the outgoing direction, it is necessary to calculate the product distribution every time a sampling decision is made. For a given incident direction ω_i and scene location $\mathbf{x}$, this procedure consists of the following steps (see Fig. 2, bottom, for an illustration):

1. Query the BRDF cache (Sec. 4.1) for the mixture p_ρ corresponding to ω_i and the material at $\mathbf{x}$.
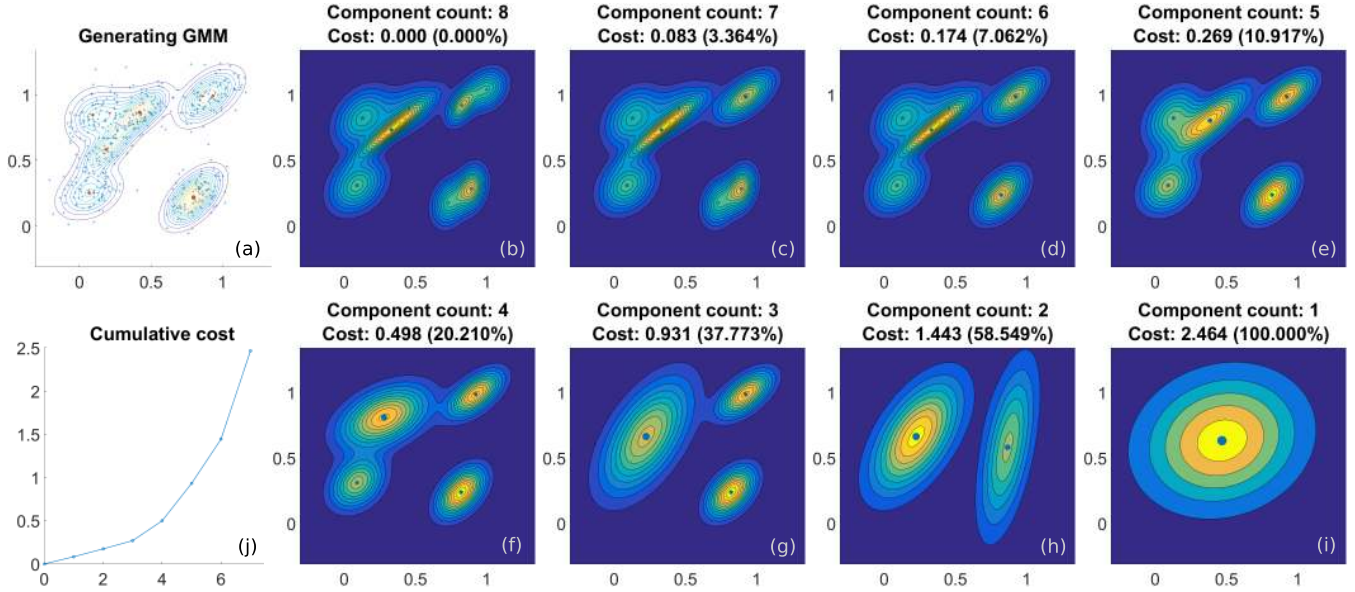
Figure 6: *Demonstration of the reduction algorithm for synthetic data generated from a random 8-component Gaussian mixture* (a). *We first perform an EM fit to the data* (b) *without any knowledge about the generating mixture, and then proceed to decimate it down to a single component* (c–i). *The cumulative reduction cost Eq. 11 is plotted in* (j) *with respect to the number of removed mixture components. In this case the reduction would stop at six components for our standard cost threshold* $C = 0.2$, *resulting in a mixture with minimal differences from the starting one.*

2. Query the illumination cache (Sec. 4, overview) for the mixture p_L at $\mathbf{x}$.
3. Compute the resulting product mixture $p_\otimes$ (Eq. 6).
4. Importance-sample $p_\otimes$ to obtain an outgoing direction ω_o to continue the path, and then evaluate $p_\otimes(\omega_o)$ to normalize the obtained path contribution (Eq. 2).

GMM product. Crucially to our approach, a product of two Gaussian mixtures is again a Gaussian mixture [Bro03]:

$$\mathcal{G}_1(\mathbf{y}) \cdot \mathcal{G}_2(\mathbf{y}) = (\mathcal{G}_1 \otimes \mathcal{G}_2)(\mathbf{y}) \equiv \mathcal{G}_{1,2}(\mathbf{y}), \qquad (9)$$

where we contract $\mathcal{G}(\Theta_1)$ to $\mathcal{G}_1$ and so on. The product mixture parameters $\Theta_{1,2}$ can be obtained analytically (we list the formulas in Apx. A).

However, the size of the resulting mixture is quadratic in respect to the factors, i. e., $K_{1,2} = K_1 K_2$. While sampling a Gaussian mixture is relatively cheap (based on the weights π_k only a single component is selected and sampled), the computation and evaluation of the product mixture each require $O(K_{1,2})$ evaluations of the bi-variate Gaussian distribution, corresponding to a relatively costly exponential function. The quadratic complexity of this core operation implies that it is more beneficial to decimate the number of components in the *factor mixtures* upfront, since each removed component decreases the product calculation costs linearly.

Mixture reduction. Instead of decimating the factor mixtures each time a product mixture is evaluated, we perform this operation only once for each factor mixture once its fitting has been finalized. For the BRDF mixtures this is the end of the pre-processing stage

(Sec. 4.1), while for the light mixtures this happens after the training stage (Sec. 4). Note that this means the training stage uses the full light mixtures, which is preferred since inaccuracies occurring during the training would negatively impact the entire rendering stage.

The naïve way to perfrom mixture decimation is based solely on the weights π_k: keep components with weights above a certain small quantile (i. e., $q = 0.1$) and trim the remaining ones. Unfortunately, this approach can unpredictably remove isolated components that might actually correspond to salient features in the solution. A preferred decimation method should therefore be based on *merging similar components*. Such a technique might still decrease the quality of the product sampling, but if done gracefully, will compensate for it with a higher resulting sampling rate.

For this purpose, we have adapted the GMM reduction algorithm by Runnalls [Run07]. This is a greedy algorithm that proceeds as follows:

1. Begin with a full mixture of K components.
2. For each pair of components $k_i, k_j \in 1..K$, compute the cost $c_{i,j}$ (Eq. 10) incurred by merging them.
3. Merge the two components with the lowest $c_{i,j}$, yielding a new mixture with $K' = K - 1$ components.
4. Finish if K' reaches a desired value, else set $K = K'$ and go to step 2.

Runnalls proposes to calculate the cost as an increase on the Kullback-Leibler (KL) divergence, corresponding to an amount of lost information expressed by the mixture. While KL divergence has no closed-form expression for the GMM, its upper bound is

easily computed and in practice tends to work well [CWPS11]:

$$c_{i,j} = \frac{\pi_i \pi_j}{2} \log |\Sigma_{i,j}| - \frac{\pi_i}{2} \log |\Sigma_i| - \frac{\pi_j}{2} \log |\Sigma_j|, \quad (10)$$

where $\Sigma_{i,j}$ is the covariance of the merged component (see Apx. B for the merging formulas). An illustrative example of this algorithm is shown in Fig. 6.

Reduction quality control. A shortcoming of Runnalls' algorithm is the inability to control the reduced mixture quality, since its termination is conditioned by the number of desired components. Instead, we wish for the reduction to be adaptive, i. e., to decimate mixtures in proportion to their component redundancy (especially because the EM algorithm frequently creates fits with such components). For this, we propose to calculate a *cumulative reduction cost $C_{K'}$* defined as

$$C_{K'} = \sum_{s=K}^{K'} \min_{i,j} c_{i,j}^s, \quad (11)$$

with $c_{i,j}^s$ being the KL cost of merging the components i and j in step s. In other words, Eq. 11 accumulates the reduction cost during the normal execution of the algorithm. We then stop the decimation as soon as $C_{K'}$ *would exceed* a user-defined cost threshold C in the next merging step (cf. Fig. 6). While such threshold is only empirical, we have experimentally found the value of $C = 0.2$ to be robust in all the presented scenes (Sec. 6).

Ideally, we would guide the reduction procedure by our target metric – the normalized estimator variance (Apx. C). However, in our case the estimator variance is a *ratio distribution* of two Gaussian mixtures, which has undefined mean and variance, and is notoriously difficult to estimate robustly. On the other hand, we have performed a numerical analysis indicating that the KL cost tends to be well correlated with the estimator variance and is therefore useful as a reduction measure. In addition, any threshold on the estimator variance would have to be determined empirically as well, since there is no direct way to be related to the global solution quality.

5. Implementation Details

We implemented our algorithm as a plug-in for the Mitsuba [Jak10] rendering framework. The approximation of the incoming illumination is based on the online-learning code provided by Vorba et al. [VKŠ*14].

BRDF Sampling Probability. The original sampling of Vorba et al. [VKŠ*14] corresponds to multiple importance sampling between the original BRDF and the incoming illumination GMM with a fixed probability of 0.5. This value prevents them from becoming biased (which can be caused by an inaccurate illumination approximation) or from strongly increasing the variance on glossy surfaces. Since our algorithm includes the knowledge of the BRDF via its product with the incoming radiance, we can safely lower this threshold to a marginal safety value of 0.1 (i. e., the product distribution will be used to make 90 % of sampling decisions).

Parallelization. In all our experiments we initially (before reduction) use 8 GMM components to fit both BRDFs and illumination. The GMM parameters are stored in SSE vectors with a width of 4,
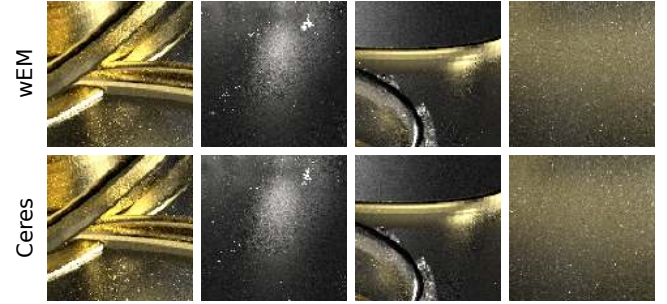

Figure 7: *Comparing the glossy* JEWELRY *scene rendered by our algorithm using different fitting methods for the BRDF caches.* Top: *weighted EM (*MSE = 0.00118*), bottom: Ceres (*MSE = 0.00109*).*

granting us a four-fold increase in the product evaluation and sampling performance. The parallelized code iterates over the BRDF components (since these tend to have a higher affinity to be reduced to less components given a certain reduction cost threshold), processing four illumination components at the same time. We allow the reduction of BRDF mixtures to freely vary between 1 and 8 components, while the reduction of illumination mixtures can reach only 4 components, or stay at 8 (since, due to our parallelization scheme, the intermediate illumination mixture sizes would not bring any additional speed-up).

6. Results

This sections presents some results of our algorithm. We first analyze the BRDF GMM fitting, comparing the quality of fits using weighted EM and non-linear optimization with CERES (Sec. 6.1). We then compare the presented product sampling algorithm to Vorba et al. [VKŠ*14] and other light transport algorithms, followed by a deeper analysis of individual aspects of our method (Sec. 6.2). All simulations were executed on consumer PCs running Linux on Intel Xeon E5-1620 processors (8 cores at 3.6 GHz) and 16 GB RAM.

6.1. BRDF Fitting

Sec. 4.1.2 showed that the non-linear optimization can produce better BRDF fits than the weighted EM algorithm, especially when it comes to materials with high roughness and at grazing angles (Fig. 4). However, if the roughness is not high then weighted EM still produces acceptable fits to the BRDF lobe. In our experiments we found out the there is no major difference between both methods when $\alpha \leq 0.2$. Therefore, to be closer to a practically realistic use-case, we used weighted EM for our main results presented in Sec. 6.2, because it provides a near-instant fitting (cf. Table 1). Fig. 7 compares the results of the JEWELRY scene, using weighted EM and Ceres for fitting the BRDF caches.

6.2. Product Sampling

Fig. 8 shows a comparison of our product sampling to standard methods: path tracing (PT), bi-directional path tracing (BDPT) and the illumination-guided path tracing (GPT) of Vorba et al. [VKŠ*14].
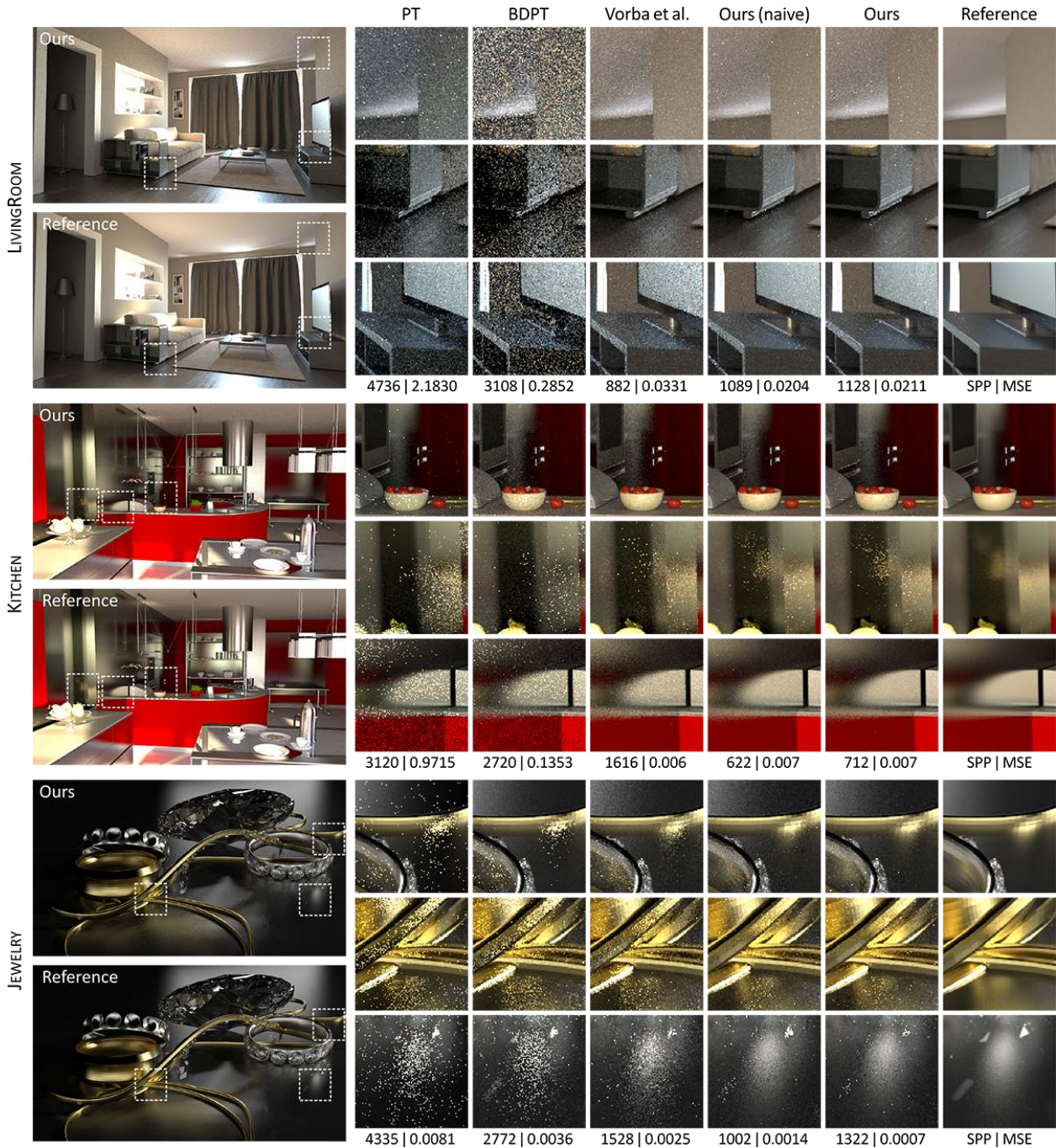
	PT	BDPT	Vorba et al.	Ours (naïve)	Ours	Reference
LIVINGROOM	4736 | 2.1830	3108 | 0.2852	882 | 0.0331	1089 | 0.0204	1128 | 0.0211	SPP | MSE
KITCHEN	3120 | 0.9715	2720 | 0.1353	1616 | 0.006	622 | 0.007	712 | 0.007	SPP | MSE
JEWELRY	4335 | 0.0081	2772 | 0.0036	1528 | 0.0025	1002 | 0.0014	1322 | 0.0007	SPP | MSE

Figure 8: *Equal-time (1 hour) results containing SPP and mean squared error (MSE) values for* LIVINGROOM, KITCHEN *and* JEWELRY *scenes. Two different versions of our product sampling are used:* "Ours (naïve)" *samples the full product while* "Ours" *includes the cache reduction. Detailed analysis is provided in Sec. 6.2.*

The render times for all algorithms were fixed to 1 hour; this includes the illumination cache training times (see Table 1) for the methods that require it (GPT and ours). We use identical illumination caches for GPT and our method, trained with 60M photons and 60M importons spread across 30 training passes. The timings for the BRDF fitting were however counted separately (we treat this step as pre-processing) and are listed in Table 1. For isotropic caches we use $N = 512$ incident elevation angles, and $M = 4069$ incident directions for the anisotropic caches. Reference images took approximately two weeks to render with vertex connection and

merging [GKDS12]. To compare the sampling quality directly, we also present one scene with equal sampling rates in Fig. 1.

Table 1: *Timings for the cache fitting stages (in minutes).*

Scene	Illumination	BRDF	
		wEM	Ceres
LIVINGROOM	14.0	0.31	25.5
KITCHEN	20.1	0.44	42.3
JEWELRY	6.1	0.04	6.3

Memory consumption and component reduction. Since the 2D GMMs can be stored compactly using 6 floats per Gaussian component, the memory needed to store the BRDF mixtures is negligible (<10 MB even for complex scenes like KITCHEN or LIVINGROOM). We therefore always reserve the space for $K = 8$ components and use the presented reduction algorithm (Sec. 4.2) only to speed up the computation and sampling of the product mixtures. Table 2 (left and middle) shows detailed statistics for the BRDF and illumination caching in our test scenes. On average the required number of reduced BRDF components is quite low, and interestingly, it varies for different incident directions. Namely, more components (up to 5) are needed at grazing angles, since the BRDF lobe is more distorted and skewed towards the hemispherical boundary. As for the illumination caches, we need about 25 % less memory than Vorba et al. [VKŠ*14]: on average about half of the illumination mixtures can be reduced from 8 to 4 components without negatively impacting the sampling.

Sampling overhead. To compare the computational cost of our product sampling to the illumination-only sampling [VKŠ*14] we measured the average overhead *per generated path segment* induced by our algorithm. Table 2 (right) summarizes these measurements. Note that on purely diffuse surfaces the BRDF is constant and its product with the illumination mixture is the illumination mixture itself. We therefore switch to pure illumination sampling on diffuse surfaces to avoid wasting computational effort. Interestingly, while the mixture reduction decreases the per-segment sampling time, it does not impact the overall rendering time significantly for our tested scenes. However, given that the product calculation has a quadratic complexity with respect to the factor mixtures, the relative importance of reducing their sizes would increase if we used more than 8 components to fit them.

Path length analysis. Examining the generated samples per pixel (SPP) for the presented scenes (Fig. 8), it is apparent that they are not directly proportional to the product-sampling overhead (Table 2, right). Our analysis shows that this is caused by the fact that the product sampling generates longer paths compared to the GPT of Vorba et al. [VKŠ*14]. Specifically, the average path lengths in each scene were (GPT / ours): 9.0 / 6.5 (LIVINGROOM), 4.9 / 9.8 (KITCHEN), 3.7 / 4.9 (JEWELRY), 6.1 / 17.2 (KITCHENETTE). The main reason is the used path termination scheme (see the discussion in Sec. 7), which culls *low-throughput* paths. Since our product sampling generates paths with higher and more consistent throughput, they tend to stay alive longer. One exception is the LIVINGROOM scene, where all BRDFs have a diffuse component and consequently the paths stay alive longer even for GPT – here we actually generate
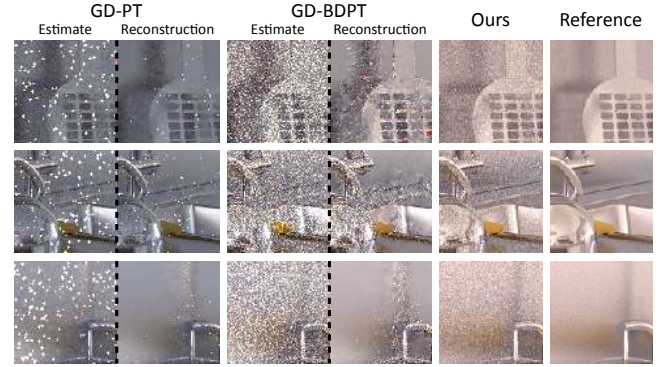

Figure 9: *Comparison of gradient-domain (GD) methods (PT and BDPT) to our product sampling, after 75 minutes of total rendering time. While incorporating image gradients into the reconstruction removes some variance, features not present in the base estimate still cannot be recovered. In contrast, our product sampling can successfully find even the more difficult paths and yields a much more even convergence.*

paths with lengths closer the mean contributing path length and hence also produce more SPP.

7. Discussion

Extension for other MC algorithms. Since our implementation treats the product sampling as a drop-in replacement for BRDF sampling, combining it with other tracing-based algorithms such as BDPT or gradient-domain PT / BDPT [KMA*,MKA*15] would be relatively straightforward. Vorba et al. [VKŠ*14] already implemented illumination-guided sampling for BDPT – while their guided sampling increased the convergence rate, it still struggles at sampling complex glossy-glossy interactions, which our product sampling handles well. Similarly, while gradient-domain methods can employ image gradients for better image reconstruction, they rely on the underlying baseline algorithms to reliably obtain the path contribution and therefore suffer from their inherent limitations. We demonstrate this in Fig. 9 on the KITCHENETTE scene, where significant path contributions come from multiple consecutive glossy-glossy interactions. Our algorithm increases the robustness of these methods and would directly produce better per-pixel radiance and gradient estimates, which in turn should lead to better reconstruction.

Spatially varying BRDFs. A current limitation of our approach is the handling of SVBRDFs. Due to the compactness of the BRDF G-MM caches, pre-sampling a large variety of different parameters for the glossy components of the SVBRDFs is still feasible. However, an ideal solution would be to develop a direct functional transform from the SVBRDF parameters to a GMM representation of the BRDF lobe. We see this as an interesting direction for future work, which could build for instance on [WÅ] or [XSD*13].

Russian roulette. By design, our product sampling always generates paths with significant throughput. The consequence of terminating paths using the weight-window RR with a low threshold (akin to

Table 2: Left *and* middle*: BRDF and illumination caching statistics for the scenes in Fig.* 8*.* Right*: Overhead of the product sampling relative to illumination-only sampling, without ('naïve') and with the reduction of the BRDF and illumination mixtures.*

Scene	BRDF caching				Illumination caching			Sampling overhead [%]	
	# BRDFs	# Caches	Avg. # comp.	Mem.	# Caches	# Reduced	Mem.	Naïve	Reduced
LIVINGROOM	41	15k	2.5	7.7 MB	82k	57 %	192.9 MB	10.8	7.1
KITCHEN	72	2.5k	1.8	10 MB	107k	62 %	236.9 MB	26.7	9.9
JEWELRY	6	1.5k	1.44	0.7 MB	16k	33 %	19.5 MB	16.5	−1.1

Vorba et al. [VKŠ*14], cf. Sec. 6.2) is that we tend to generate long paths, even if their eventual contribution is small. Unfortunately the low threshold is necessary to find some long but significant paths, and making its value higher typically increases the incidence of outlier noise ('fireflies'). We believe this issue could be addressed by employing an adaptive path termination strategy (such as [VK16]), which takes the relationship between the path throughput and its expected contribution into account.

Quality of illumination caches and MIS weight. The illumination caches can sometimes be fitted poorly, which is manifested by discontinuous patches of high variance (e. g., LIVINGROOM, left wall) or different low-variance convergence rates (e. g., JEWELRY, floor). This is a general limitation shared with Vorba et al. [VKŠ*14], and we believe it holds potential for further research. For instance, the iterative nature of their online training algorithm could possibly be combined with a component merging and splitting strategy, so that poorly fitted GMMs could be identified and corrected early. Another possible optimization is to integrate the MIS weight for sampling the BRDF into the illumination GMM by adding a special mixture component that would represent 'uncategorized' background radiance. This would prevent the EM algorithm from being distracted by background radiance or variance in the illumination samples.

Mixture reduction. The reduction algorithm adapted from Runnalls [Run07] is robust, and its simplicity helps to keep the computational costs low. Also the fact that it maintains the mean and variance of the entire mixture unchanged (Apx. B) is useful in preventing salient parts of the directional domain from becoming critically under-sampled. Still, more sophisticated decimation approaches could be examined [CWPS11], or even a more general algorithm could be employed by combining the reduction with the possibility of *splitting* the components, similar to the bottom-up building of GMMs [HH08, SH09].

8. Conclusion

Product importance sampling can significantly improve the convergence of Monte Carlo global illumination solvers, which we have demonstrated for a guided uni-directional path tracing. This is especially true for difficult scenes—for instance with complex illumination, difficult visibility, or glossy materials—where ignoring some terms of the rendering equation during importance sampling can lead to significant increases of variance. Our main effort in the future will be to improve the adaptivity of the algorithm to even more difficult conditions, but, equally importantly, to recognize and limit its use whenever it can lead to unnecessary overhead.

References

[AMO16] AGARWAL S., MIERLE K., OTHERS: Ceres solver. http://ceres-solver.org, 2016. 4

[Arv86] ARVO J.: Backward ray tracing. In *ACM SIGGRAPH Course Notes* (1986). 2

[BGH05] BURKE D., GHOSH A., HEIDRICH W.: Bidirectional importance sampling for direct illumination. In *Proc. of EGSR* (2005). 2

[Bis06] BISHOP C. M.: *Pattern recognition and machine learning*. Springer, 2006. 2, 3

[Bro03] BROMILEY P.: Products and convolutions of Gaussian distributions. http://www.tina-vision.net/docs/memos/2003-003.pdf, 2003. internal report. 6

[CAM] CLARBERG P., AKENINE-MÖLLER T.:Practical product importance sampling for direct illumination. *Comp. Graph. Forum 27*, 2. 2

[CJAMJ05] CLARBERG P., JAROSZ W., AKENINE-MÖLLER T., JENSEN H. W.: Wavelet importance sampling: efficiently evaluating products of complex functions. *ACM Trans. Graph. (proc. of SIGGRAPH) 24*, 3 (2005). 2

[CWPS11] CROUSE D., WILLETT P., PATTIPATI K., SVENSSON L.: A look at Gaussian mixture reduction algorithms. In *Proc. of Information Fusion* (July 2011). 7, 10

[DLR77] DEMPSTER A. P., LAIRD N. M., RUBIN D. B.: Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Ser. B*, 1 (1977). 2, 3

[GKDS12] GEORGIEV I., KŘIVÁNEK J., DAVIDOVIČ T., SLUSALLEK P.: Light transport simulation with vertex connection and merging. *ACM Trans. Graph. 31* (2012). 9

[HD] HEITZ E., D'EON E.: Importance sampling microfacet-based BSDFs using the distribution of visible normals. *Computer Graphics Forum 33*, 4. 2

[HH08] HUBER M. F., HANEBECK U. D.: Progressive Gaussian mixture reduction. In *Proc. of Information Fusion* (2008). 10

[HP02] HEY H., PURGATHOFER W.: Importance sampling with hemispherical particle footprints. In *Proc. of SCCG* (2002). 1, 2

[Jak10] JAKOB W.: Mitsuba renderer, 2010. http://www.mitsuba-renderer.org. 7

[JCJ09] JAROSZ W., CARR N. A., JENSEN H. W.: Importance sampling spherical harmonics. *Computer Graphics Forum (Proc. of Eurographics) 28*, 2 (2009). 2

[Jen95] JENSEN H. W.: Importance driven path tracing using the photon map. In *Rendering Techniques (proc. EGWR)* (1995). 1, 2

[JRJ11] JAKOB W., REGG C., JAROSZ W.: Progressive expectation-maximization for hierarchical volumetric photon mapping. *Computer Graphics Forum*, 4 (2011). 2

[Kaj86] KAJIYA J. T.: The rendering equation. *SIGGRAPH Comput. Graph.* (1986). 1, 2, 3

[KISS15] KEINERT B., INNMANN M., SÄNGER M., STAMMINGER M.: Spherical Fibonacci mapping. *ACM Trans. Graph. 34*, 6 (2015). 5

[KMA∗] KETTUNEN M., MANZI M., AITTALA M., LEHTINEN J., DURAND F., ZWICKER M.: Gradient-domain path tracing. *ACM Trans. Graph. 34*. 9

[KSKAC02] KELEMEN C., SZIRMAY-KALOS L., ANTAL G., CSONKA F.: A simple and robust mutation strategy for the Metropolis light transport algorithm. *Comp. Graph. Forum 21* (2002). 2

[LRR04] LAWRENCE J., RUSINKIEWICZ S., RAMAMOORTHI R.: Efficient BRDF importance sampling using a factored representation. In *Proc. of SIGGRAPH* (2004). 2

[LW95] LAFORTUNE E. P., WILLEMS Y. D.: A 5D tree to reduce the variance of Monte Carlo ray tracing. In *Rendering Techniques* (1995). 2

[MKA∗15] MANZI M., KETTUNEN M., AITTALA M., LEHTINEN J., DURAND F., ZWICKER M.: Gradient-domain bidirectional path tracing. In *Proc. of EGSR (Exp. Ideas & Impl.)* (2015). 9

[PH10] PHARR M., HUMPHREYS G.: *Physically based rendering: from theory to implementation*, 2nd ed. Morgan Kaufmann, 2010. 1, 2

[Run07] RUNNALLS A. R.: A Kullback-Leibler approach to Gaussian mixture reduction. *IEEE Trans. Aerosp. Electron. Syst* (2007). 6, 10, 11

[SC97] SHIRLEY P., CHIU K.: A low distortion map between disk and square. *J. Graph. Tools*, 3 (1997). 3

[SH09] SCHIEFERDECKER D., HUBER M.: Gaussian mixture reduction via clustering. In *Proc. of Information Fusion* (2009). 10

[TCE05] TALBOT J. F., CLINE D., EGBERT P.: Importance resampling for global illumination. In *Proc. EGSR* (2005). 2

[Vea98] VEACH E.: *Robust Monte Carlo methods for light transport simulation*. PhD thesis, Stanford University, 1998. 2

[VG97] VEACH E., GUIBAS L. J.: Metropolis light transport. In *Proc. of SIGGRAPH* (1997). 2

[VK16] VORBA J., KŘIVÁNEK J.: Adjoint-driven russian roulette and splitting in light transport simulation. *ACM Trans. Graph (Proc. of SIGGRAPH)* (2016). To be published. 10

[VKŠ∗14] VORBA J., KARLÍK O., ŠIK M., RITSCHEL T., KŘIVÁNEK J.: On-line learning of parametric mixture models for light transport simulation. *ACM Trans. Graph. (Proc. of SIGGRAPH) 33* (2014). 1, 2, 3, 4, 7, 9, 10

[WÅ] WANG R., ÅKERLUND O.: Bidirectional importance sampling for unstructured direct illumination. *Computer Graphics Forum (proc. of Eurographics) 28*, 2. 2, 9

[WMLT07] WALTER B., MARSCHNER S. R., LI H., TORRANCE K. E.: Microfacet models for refraction through rough surfaces. In *Proc. of EGSR* (2007). 2

[WRG∗09] WANG J., REN P., GONG M., SNYDER J., GUO B.: All-frequency rendering of dynamic, spatially-varying reflectance. In *Proc. of SIGGRAPH Asia* (2009). 2, 4

[XCM∗14] XU K., CAO Y.-P., MA L.-Q., DONG Z., WANG R., HU S.-M.: A practical algorithm for rendering interreflections with all-frequency BRDFs. *ACM Trans. Graph. 33*, 1 (2014). 2

[XSD∗13] XU K., SUN W.-L., DONG Z., ZHAO D.-Y., WU R.-D., HU S.-M.: Anisotropic spherical Gaussians. *ACM Trans. Graph. 32*, 6 (2013). 2, 4, 9

Appendix A: GMM Product Calculation

Given two GMM mixtures $\mathcal{G}_1$ and $\mathcal{G}_2$, the parameters $\Theta_{1,2}$ of the resulting product mixture (Eq. 9) can be calculated as follows:

$$K_{1,2} = K_1 K_2 \tag{12}$$

$$\pi_{k_{1,2}} = \frac{\int \pi_{k_1} \mathcal{N}_{k_1}(\mathbf{y}) \cdot \pi_{k_2} \mathcal{N}_{k_2}(\mathbf{y}) \, d\mathbf{y}}{\int \mathcal{G}_1(\mathbf{y}) \cdot \mathcal{G}_2(\mathbf{y}) \, d\mathbf{y}}$$

$$= \frac{\pi_{k_1} \pi_{k_2} \cdot \mathcal{N}(\mu_{k_1} | \mu_{k_2}, \Sigma_{k_1} + \Sigma_{k_2})}{\sum \sum \pi_{k_1} \pi_{k_2} \cdot \mathcal{N}(\mu_{k_1} | \mu_{k_2}, \Sigma_{k_1} + \Sigma_{k_2})} \tag{13}$$

$$\mu_{k_{1,2}} = \Sigma_{k_1}(\Sigma_{k_1} + \Sigma_{k_2})^{-1} \mu_{k_2} + \Sigma_{k_2}(\Sigma_{k_1} + \Sigma_{k_2})^{-1} \mu_{k_1} \tag{14}$$

$$\Sigma_{k_{1,2}} = \Sigma_{k_1}(\Sigma_{k_1} + \Sigma_{k_2})^{-1} \Sigma_{k_2} \tag{15}$$

where $k_{1,2} = K_1 k_1 + k_2$. We contract the notation similarly to Eq. 9.

Appendix B: Merging Gaussian Components

The parameters of the Gaussian component $\mathcal{G}_{1,2}$ created by merging components $\mathcal{G}_1$ and $\mathcal{G}_2$ are calculated as follows:

$$\pi_{1,2} = \pi_1 + \pi_2 \tag{16}$$

$$\mu_{1,2} = \bar{\pi}_1 \mu_1 + \bar{\pi}_2 \mu_2 \tag{17}$$

$$\Sigma_{1,2} = \bar{\pi}_1 \Sigma_1 + \bar{\pi}_2 \Sigma_2 + \bar{\pi}_1 \bar{\pi}_2 (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T \tag{18}$$

where $\bar{\pi}_i = \frac{\pi_i}{\pi_1 + \pi_2}$. This operation in fact preserves the first two moments of the merged components [Run07]. Consequently, the reduced mixture produced by the algorithm described in Sec. 4.2 retains the mean and variance of the full mixture, independently on the decimation magnitude.

Appendix C: Normalized Estimator Variance

To measure the quality of importance-sampling a given target function f using a PDF p, we introduce a discrepancy measure called the *normalized estimator variance*:

$$\text{NEV}_{f,p} = \frac{1}{N} \cdot \sum_n^N \left(\frac{f(\mathbf{y}_n)}{m_{f,p} \cdot p(\mathbf{y}_n)} - 1 \right)^2, \text{where} \tag{19}$$

$$m_{f,p} = \frac{1}{N} \cdot \sum_n^N \frac{f(\mathbf{y}_n)}{p(\mathbf{y}_n)} \tag{20}$$

As the name implies, the usefulness of Eq. 20 stems from the fact that it directly measures the variance of an estimator with N samples, given that p is a valid PDF for f:

$$m_{f,p} \approx \int f(\mathbf{y}) \, d\mathbf{y}. \tag{21}$$

Furthermore, it has the following properties:

- $\text{NEV}_{f,p} = 0$ iff f and p differ only by a constant factor.
- $\text{NEV}_{f,p} \to \infty$ for increasing discrepancy between f and p, in agreement with the fact that the variance of an estimator might not have a finite or even a defined value.
- $\text{NEV}_{f,p}$ is purposefully asymmetric, since, even if the case of both f and p being valid PDFs, they are not interchangeable for the purposes of sampling each other.