Adaptive multi-view path tracing

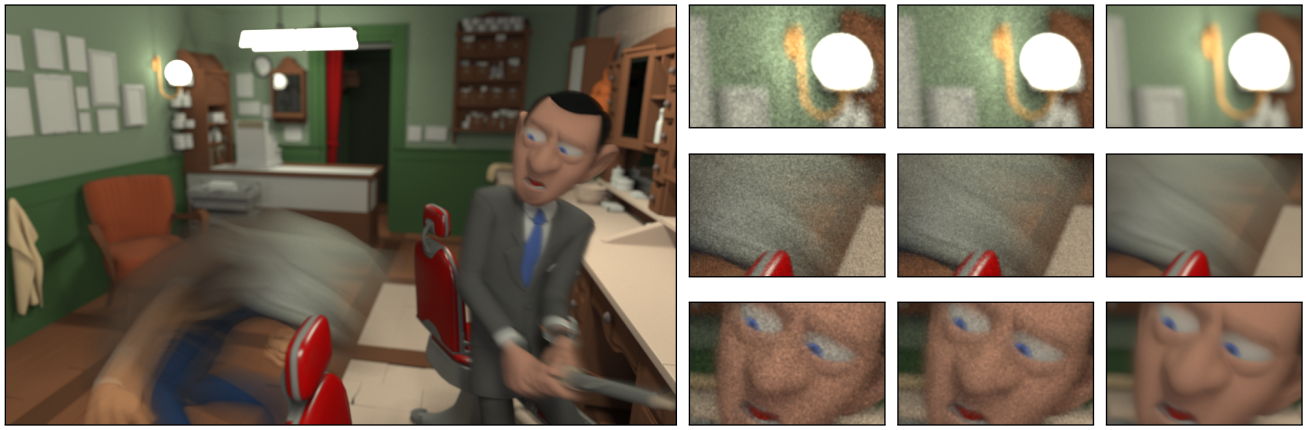
Basile Fraboni, Jean-Claude Iehl, Vincent Nivoliers, Guillaume Bouchard

Eurographics Symposium on Rendering (DL-only Track) (2019)
T. Boubekeur and P. Sen (Editors)

Adaptive multi-view path tracing

Basile Fraboni[1,3] and Jean-Claude Iehl[2] and Vincent Nivoliers[2] and Guillaume Bouchard[3]

[1]INSA de Lyon, CNRS, LIRIS, UMR5205, France
[2]Université de Lyon, CNRS, Université Lyon 1, LIRIS, UMR5205, France
[3]Mercenaries Engineering



a) Path tracing b) Ours - equal time c) Reference

Figure 1: *We compare a frame of the 100 frame barber shop animation sequence, rendered with independent path tracing in 40 seconds per frame (a), with our multi-view path tracing solution in the same time (b) and computed in 2 hours for a single reference frame (c). Our algorithm exploits consistency between neighboring views, and yields faster convergence than traditional path tracing.*

Abstract

Rendering photo-realistic image sequences using path tracing and Monte Carlo integration often requires sampling a large number of paths to get converged results. In the context of rendering multiple views or animated sequences, such sampling can be highly redundant. Several methods have been developed to share sampled paths between spatially or temporarily similar views. However, such sharing is challenging since it can lead to bias in the final images. Our contribution is a Monte Carlo sampling technique which generates paths, taking into account several cameras. First, we sample the scene from all the cameras to generate hit points. Then, an importance sampling technique generates bouncing directions which are shared by a subset of cameras. This set of hit points and bouncing directions is then used within a regular path tracing solution. For animated scenes, paths remain valid for a fixed time only, but sharing can still occur between cameras as long as their exposure time intervals overlap. We show that our technique generates less noise than regular path tracing and does not introduce noticeable bias.

CCS Concepts

• ***Computing methodologies*** → ***Computer graphics; Ray tracing;*** *Visibility;*

1. Introduction

Rendering noise-free images with path tracing often requires heavy computations and sampling thousands of paths through each pixel. Rendering a sequence of images is even more intensive despite the high similarity of consecutive frames. Reducing temporal noise also requires more samples to remove flickering. Several authors have proposed different ways of reusing a single path to compute several pixels. These methods have achieved interesting speedups, but they all share the same limitation. Paths are sampled starting from one camera and they are transformed to contribute to another

camera. As we shall see, this transformation can increase the variance of the Monte Carlo estimation of light transport.

Our paper proposes a different approach to this problem. We make the key observation that light transport can be estimated at any point in space and time and that many of these points are observed by more than one camera. Contrary to previous works that construct an entire path for a given camera, we progressively build a path that significantly contributes to several cameras. To do so, we first cast a ray from a camera, and then importance sample a reflected direction at the intersected point which gives significant energy to several cameras. Our importance sampling technique restricts the magnitude of the transformation and thus controls the variance of the observations by only selecting a subset of the cameras observing the point. The rest of the light transport is then estimated with standard path tracing.

Decoupling observations and transport estimation also makes our definition of a camera fairly general: a camera observes a region of space and time. A moving camera in an image sequence creates an observation per frame. A stereo/VR camera makes 2 observations, etc. Our approach can be used in very different settings: stereo-pair/VR video, camera array/light field rendering, etc.

2. Previous work

Exploiting the similarity between coherent viewpoints has been investigated in the past. We briefly describe existing approaches reusing paths and integrating in the gradient domain.

2.1. Space-time coherence

Taking advantage of space-time coherence by using the same samples in multiple renderings is a widely used technique for interactive and real time applications. In the context of offline Monte Carlo integration, several techniques have been developed.

Adelson and Hodges [AH93, AH95] directly exploit spatio-temporal coherence in the context of ray tracing. Their idea is to reuse samples by projecting them from one camera to another instead of sampling each camera separately. Samples on diffuse materials are reprojected, glossy and specular reflections are then simulated for each camera. These early approaches dramatically reduce the rendering time of nearby frames, like stereoscopic images or consecutive frames in animation sequences, but their efficiency decrease with the amount of glossy materials.

Bekaert et al. [BSH02] introduce the concept of path re-use to compute a single frame. The image is divided in small square windows and pixels of each window share sampled paths. This first solution however shows highly correlated and structured noise before reaching convergence. A correction of these artifacts is given in [XS07] by using less structured pixel neighborhoods and by shuffling reused contributions. This technique has recently been extended to gradient domain rendering [BPE17]. Massively increasing the number of samples by reusing calculations leads to a better domain exploration, at the cost of correlation and structured artefacts. Our technique works on a different setting: instead of sharing pieces of paths generated for neighboring pixels on a single image, we sample and share full paths between different viewpoints. In this setting path re-use doesn't show structured noise.

Havran et al. [HDMS03] discuss reprojection data structures and algorithms to efficiently render multiple animation frames over a time segment. They use clamping to remove negligible reprojected sample contributions which would otherwise increase the variance. Although this technique is appropriate for diffuse materials, re-evaluating and clamping the brdf to reproject samples can lead to a poor importance function for glossy materials.

Feliu et al. [FSSK06] improve on [HDMS03] and introduce the normalization integral involved in the projection from one camera to another. The authors also introduce multiple importance sampling to further reduce variance while reprojecting samples to several cameras. Even though the normalization constant and its integration are fully derived in the paper, the authors use a simple approximation. They achieve better results when rendering glossy materials.

Henrich et al. [HBGM11] propose a hybrid architecture mixing CPU rendering with a simple and efficient GPU reprojection of path vertices on the screen. Artifacts created by this simplified reprojection are then reduced by filtering and downsampling. We use a more sophisticated technique to selectively reproject samples, and to account for the Jacobian of the reprojection, thereby limiting the amount of variance and removing the need to post process our results.

Our method is similar to these approaches in that we re-use samples for similar viewpoints. However, our key idea is the selection of the set of cameras which will share a sample path *before generating the full path*. This allows us to take all these cameras into account in the generation of the full path. Our paths are then combined using multiple importance sampling to reduce the variance in the result.

Zimmer et al. [ZRJ*15] propose a complex post-processing pipeline to jointly denoise and upscale animation sequences, both spatially and temporally. Images are decomposed into diffuse and specular components. Diffuse paths are re-used (through motion interpolation) while specular paths are re-evaluated with a temporal manifold exploration technique and then interpolated. We present a simple variance reduction technique which can be used to accelerate the computation of an animation sequence (same variance but faster), or to reduce noise (same time but lower variance). Our technique can also be used in different settings such as light field rendering or stereo/VR rendering.

2.2. Gradient domain rendering

Gradient domain rendering is a recent approach computing both a coarse image and its gradient, and using Poisson reconstruction to obtain the final image. This approach has been successfully applied to Metropolis light transport [LKL*13], path tracing [KMA*15] and bidirectional path tracing [MKA*15]. Paths are reused between neighboring pixels to compute the gradients. The correlation between the paths improves the variance in the gradient computations. This approach has been further extended to animated sequences [MKD*16], computing temporal gradients

as well to reduce flickering artifacts between consecutive frames. Paths are reused in a different manner: an original path is generated to provide a new sample for the image integral. The path is then deterministically shifted to a neighboring pixel or frame. Finite differences are finally used to estimate gradient values. Our method is not incompatible with gradient domain methods and merely allows to get more paths using reprojections, and gradient domain solutions could benefit from our approach. We also demonstrate the efficiency of our method in the context of motion blur with fast moving objects. In such a context, to our knowledge, no efficient methods exist to compute the gradients. Since motion blur is mostly due to the discontinuities in the visibility of the scene, such gradients could be difficult to estimate.

3. Background

We focus on improving the classical path-tracing algorithm. This algorithm computes realistic images by simulating light transport throughout a scene via Monte Carlo integration. Each rendered pixel j of a camera k is computed as the integral over the light reaching the pixel [Vea98] and the exposure interval $[t_0\, t_1]$ of the camera:

$$I_j^k = \int_{t_0^k}^{t_1^k} \int_{\Omega(t)} f_j(\bar{X}) d\mu(\bar{X}) dt \tag{1}$$

where $\Omega(t)$ is the union of all light paths $\bar{X}$ of finite lengths at time t, f_j is the light contribution carried through the path and μ is the product of the measures of the differential area elements at each path vertex [VG97]. Note that f_j contains in particular the spatio-temporal pixel reconstruction filter, and will be 0 for a path not reaching the pixel. A path $\bar{X}$ is a finite sequence of points on the scene. For our purposes, we will decompose such a path in two parts $\bar{X} = \{xy, \bar{z}\}$ where xy is a prefix which connects a camera vertex x and a visible scene point y, and $\bar{z}$ is a suffix path, which starts at a point z and carries the radiance flux to y from z (Figure 2).

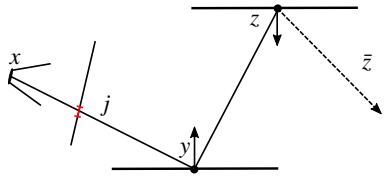


Figure 2: *An example path in the traditional configuration*

Path tracing approximates the integral Equation 1 by accumulating contributions from a set of sampled paths from the camera. A point x is generated on the exit lens and a ray is cast from x in the scene to find the first hit on the scene y. A suffix $\bar{z}$ starting from y is then generated using a stochastic process to generate a succession of bounce directions and points on the scene. The contributions f_j of the paths $\bar{X}_i = \{xy, \bar{z}\}$ for pixel j are then merged using their probability distribution functions (pdf) p :

$$\hat{I}_j^k = \frac{1}{n} \sum_i^n \frac{f_j(\bar{X}_i)}{p(\bar{X}_i)} \quad \text{with} \quad p(\bar{X}_i) = \underbrace{p(xy)p(z|xy)}_{\text{observation}} p(\bar{z}) \tag{2}$$

This formulation highlights the fact that the light transport can be decoupled from the observation. Given a suffix $\bar{z}$, its pdf $p(\bar{z})$ and the radiance $L(z \to y)$ reaching y, computing a contribution for pixel j only requires the evaluation of the visibility, brdf and camera importance function.

4. Multi-view rendering

Our goal is to reduce the variance of multiple path-tracing generated images of the same scene. We share sampled paths between multiple integrals at a fixed time. Thus virtually increasing the number of samples per image without increasing much the cost per generated path. Each camera is described by a set of parameters (e.g. position, orientation, exposure interval, lens, focal distance) and corresponds to one integral. In the case of a camera rendering an animated sequence, each frame of the sequence is considered as a unique camera. Our method is applicable in the context of surface rendering, but does not consider the case of volume rendering and participating media.

We propose a two-step importance sampling technique for multi-view path sampling. Each path is generated as follows :

1. **Prefix step :**
   - a time t is sampled ;
   - the camera ℓ and desired pixel i for the sample is importance sampled and a hit point y_ℓ is found on the scene ;
   - if the point is not purely specular, a subset of similar cameras that could have sampled this point is determined (subsubsection 4.2.4);
2. **Suffix step :**
   - a bounce direction is importance sampled using the subset of similar cameras ;
   - regular path tracing is used to generate the rest of the path ;
   - the path contribution is merged with the integral of each selected camera using multiple importance sampling and the balance heuristic.

4.1. Sampling

4.1.1. Camera sampling

We use the thin lens camera model. Generating a path for a camera ℓ therefore requires two points : a point on the film and a point on the lens. Once the pixel i requiring additional paths is determined, a film point x_i is generated in the pixel, and a lens point x_ℓ is generated on the lens disk.

4.1.2. Time sampling

For static scenes, time is irrelevant. Each frame of each camera is therefore rendered as an independent observer, and any path can be reused. For animated scenes, paths are only reused at fixed time. Indeed reusing paths between different time points in such a case introduces bias, even for paths only hitting static objects, since moving objects could change the visibility between the points, or change the illumination received by static objects. For this reason, paths are sampled at a precise time, and only reused if the exposure

interval of the observer contains that time. We therefore introduce no bias due to changes in visibility and illumination since the path is only used for a fixed time.

4.2. Multi-view integration

Once an initial sample is generated and a corresponding hit on the scene is found, a set of compatible cameras for sharing the path suffix is determined. The initial path prefix is transformed using a deterministic shift mapping, for each camera opened at the time of the path. The associated change in density is then evaluated through the Jacobian of the transformation. A selection probability is finally derived for the observer depending on both this Jacobian and the brdf at the hit point y.

4.2.1. Transforming the prefix of a path

For the generation of a prefix from a thin lens camera ℓ, both a point x_i on a pixel i and a point x_ℓ on the lens are sampled, thus obtaining a unique ray direction and a hit y_ℓ. Given another camera k, a point x_k on its lens is required to determine a point x_j on its film to reuse the path suffix. We define this point deterministically as

$$x_k = \frac{r_k}{r_\ell} x_\ell \tag{3}$$

where r_k and r_ℓ are the respective lens radii of cameras k and ℓ. This transformation does not introduce bias in the integral for k since x_k uniformly samples the lens of k. Given y_ℓ and x_k the point x_j on the film of k is uniquely defined. We denote $T_{\ell\to k}$ the transformation of a path $\bar{X}_\ell$ for observer ℓ to a path $\bar{X}_k$ for observer k (Figure 3):

$$T_{\ell\to k} : \{x_\ell y_\ell, \bar{z}\} \mapsto \{x_k y_\ell, \bar{z}\}\,. \tag{4}$$

By construction, $T_{\ell\to\ell}$ is the identity, and $T_{k\to\ell} = T_{\ell\to k}^{-1}$.

Every transformed path cannot or should not be reused. We therefore perform the following tests:

- check whether x_j is in the rectangle of the image k;
- run a Russian roulette on our selection probability (Eq. 13, 17);
- check the visibility on the segment $[x_k, y_\ell]$.

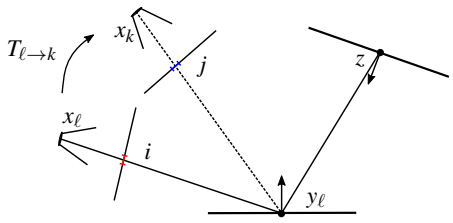


Figure 3: *Transformation from camera ℓ to camera k*

4.2.2. Transforming densities

Every observer of a primary hit point y is an importance strategy to sample both y and suffix paths $\bar{z}$. Let us consider an observer k and a pixel j for this observer. The integral for this pixel at a time t is

$$I_j^k(t) = \int_{\Omega(t)} f_j(\bar{X}) d\mu(\bar{X}). \tag{5}$$

Remember that the path contribution function f_j contains the pixel filter and will be 0 for paths not reaching j. We will omit t in the following since every path reuse is done at fixed t. Given a second observer ℓ, this integral can be split in two parts depending on whether the path could have been provided by observer ℓ or not. If Ω_ℓ is the set of paths reaching observer ℓ and $\Omega_{\ell\to k} = T_{\ell\to k}(\Omega_\ell)$, we can write

$$I_j^k = \int_{\Omega_{\ell\to k}} f_j(\bar{X}) d\mu(\bar{X}) + \int_{\Omega\setminus\Omega_{\ell\to k}} f_j(\bar{X}) d\mu(\bar{X}). \tag{6}$$

For the part of the integral over $\Omega_{\ell\to k}$ a second strategy exists to compute the integral using paths from observer ℓ. This corresponds to the following change of variables :

$$\int_{\Omega_{\ell\to k}} f_j(\bar{X}) d\mu(\bar{X}) = \int_{\Omega_\ell} f_j(T_{\ell\to k}(\bar{X}_\ell)) |T'_{\ell\to k}| d\mu(\bar{X}_\ell) \tag{7}$$

where $|T'_{\ell\to k}|$ is the determinant of the Jacobian of $T_{\ell\to k}$. As detailed in Appendix A the pdf associated with this integral for Monte Carlo estimation is

$$p_{\ell\to k}(\bar{X}_\ell) = \frac{p_\ell(\bar{X}_\ell)|T'_{\ell\to k}|V_k(T_{\ell\to k}(\bar{X}_\ell))}{K_{\ell\to k}} \tag{8}$$

$$K_{\ell\to k} = \int_{\Omega_\ell} p_\ell(\bar{X}_\ell)|T'_{\ell\to k}|V_k(T_{\ell\to k}(\bar{X}_\ell)) d\mu(\bar{X}_\ell) \tag{9}$$

where V_k checks the visibility of the transformed path.

The Jacobian determinant of the transformation can be efficiently computed as a special case of [JM12] [LKL*13], since the primary hit point y is assumed non specular and the suffix path is identical for each viewers:

$$\begin{aligned} |T'_{\ell\to k}| &= \left|\frac{\partial x_k}{\partial x_\ell}\right| \left|\frac{\partial y}{\partial x_i}\right| \left|\frac{\partial y}{\partial x_j}\right|^{-1} \\ &= \frac{r_k^2}{r_\ell^2} \frac{g(x_\ell, y)}{g(x_\ell, x_i)} \frac{g(x_k, x_j)}{g(x_k, y)} \end{aligned} \tag{10}$$

where x_i and x_j are respectively the film points at pixel i and j, and $g(a,b) = \frac{|\vec{\omega_{ab}}\cdot\vec{n_b}|}{\|b-a\|^2}$. For thin lens cameras, the Jacobian determinant is then:

$$|T'_{\ell\to k}| = \frac{r_k^2}{r_\ell^2} \frac{\cos\theta_y^\ell}{\|y-x_\ell\|^2} \frac{d_\ell^2}{\cos^3\theta_{x_i}} \frac{\|y-x_k\|^2}{\cos\theta_y^k} \frac{\cos^3\theta_{x_j}}{d_k^2} \tag{11}$$

where d is the distance between the camera origin and the film plane (cf. Figure 4) and r is the lens radius of the camera.

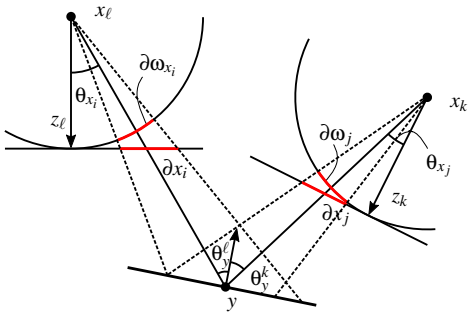


Figure 4: *Geometric configuration for the Jacobian determinant*

Given a hit generated from any observer, we first determine the set of observers who could have generated the same hit. Then for every observer in this set, we compute the pdf from the other observers and add the contribution of the path using multiple importance sampling. This strategy however may cause problems because importance functions associated with other observers can be very poor and slow down or prevent the convergence. We therefore add two selection probabilities on other observers in the multiple importance sampling framework to dampen the poor importance strategies.

4.2.3. Similar importance selection: Jacobian

The transformation of a path depends on the visibility and implies a change in density. Both the visibility and the Jacobian determinant appear in the pdf and in the normalization term. Two problems arise.

First, the evaluation of the normalization term of a pdf is an integral over the whole film of the observer potentially providing a path for the pixel. Computing multiple importance sampling weights requires comparing the pdfs of all the different strategies :

$$w_{\ell\to k}(\bar{X}_\ell) = \frac{p_{\ell\to k}(\bar{X}_\ell)}{\sum_j p_{j\to k}(T_{\ell\to j}(\bar{X}_\ell))}. \tag{12}$$

We would therefore have to evaluate $K_{\ell\to k}$ (Equation 9) for each observer involved, which is a considerable overhead.

Second, the Monte Carlo estimation of integrals suffers from bias if a subdomain is insufficiently sampled due to a finite number of samples. We observe such situations in areas where large changes in density occur. This is due to the inefficiency of some cameras to generate samples in such areas. Although no visibility problems occur, no sample is generated on one observer because the probability of generating one is small with respect to the sample budget. This bias results in strong artifacts in motion blur (cf. Figure 5).

To overcome these limitations, we ensure that the reused samples have an importance similar to that of native samples. We perform a stochastic selection (Russian roulette) of the prefixes using the Jacobian determinant as a similarity probability.

$$p_{\text{jacobian}}(\bar{X}_\ell, \bar{X}_k) = \begin{cases} \left|\frac{T'_{\ell\to k}}{T'_{\ell\to k}}\right|^{-1} & \text{if } \left|T'_{\ell\to k}\right| > 1 \\ \text{otherwise} \end{cases} \tag{13}$$

The Russian roulette accepts native samples with a probability of 1 ($\left|T'_{\ell\to\ell}\right| = 1$) and rejects samples with high or nearly zero Jacobian determinants. We show in Figure 5 that this selection strongly reduces the relative standard deviation of the normalization terms. We therefore consider the normalization terms as constants independent of the observer, and cancel them in multiple importance sampling weights without introducing noticeable bias.

4.2.4. Similar importance selection: material

For the integral to be correct, the contributions of the paths must be computed using the exact brdf with the right observer. Importance sampling is used to accelerate the convergence rate of the integral, by sampling paths proportionally to their contribution to the final integral. Using paths from other observers changes the importance function used. On glossy materials, this can slow down or prevent the convergence. We therefore use a second term in our selection probability for observers to ensure that their importance functions are compatible.

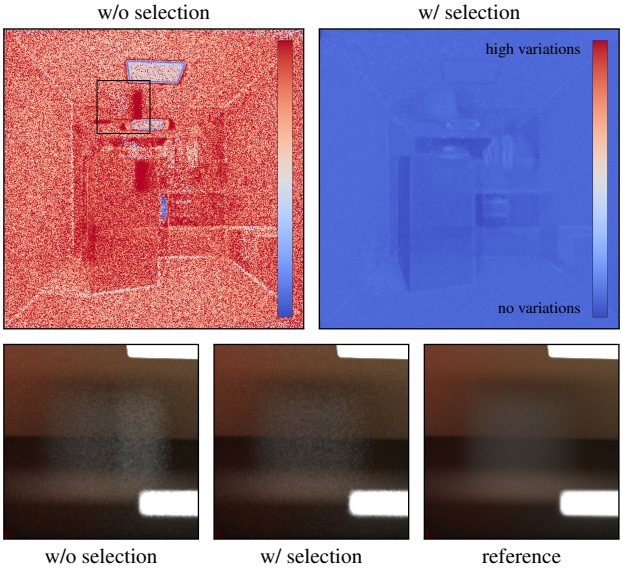


Figure 5: *Top row shows the relative standard deviation of the normalization constants (Equation 9). Performing the Jacobian selection strongly reduces their variations (right). Bottom row highlights a detail of the animated Cornell Box. From left to right, the biased result without selection due to the large variations of the constants, our result with the selection, and the reference.*

We thus propose a heuristic to select a subset of similar observers regarding the material parameters. Since the glossy importance functions $f_r(\cdot \to y \to x)$, denoted ρ, are distributions over the hemisphere we can use distance metrics to evaluate their similarity. Several distances between distributions exists, but most of them requires a numerical integration (e.g. Kullback-Leibler, Hellinger, Wasserstein) and are not always normalized. Precomputing distance tables for all material variations of a scene is costly and hardly feasible. We therefore use the total variation (TV) distance which can be evaluated on-the-fly for microfacet distributions. The TV distance δ is the maximum difference between the pdf for the same random variable.

$$\delta(\rho_\ell, \rho_k) = \sup_{\omega\in\mathcal{S}} |\rho_\ell(\omega) - \rho_k(\omega)| \tag{14}$$

The microfacet theory describes glossy lobes concentrated around the mirror direction for a given incident direction [CT82]. The mirror direction corresponds to the maximum intensity of a lobe and hence the maximum pdf value. The TV distance between two lobes is then computed by comparing the absolute difference of the pdf values regarding the peak of the initial distribution.

$$\delta(\rho_\ell, \rho_k) = |\rho_\ell(\omega_\ell) - \rho_k(\omega_\ell)| \quad \text{with} \quad \omega_\ell = \text{argmax}(\rho_\ell) \tag{15}$$

We can normalize the TV distance by the maximum of each distribution leading to a simpler formula. Thus, a TV distance close to zero means that the distributions are similar ; on the contrary, a

value close to one indicates a high dissimilarity.

$$\widetilde{\delta}(\rho_\ell, \rho_k) = \left| \frac{\rho_\ell(\omega_\ell)}{\rho_\ell(\omega_\ell)} - \frac{\rho_k(\omega_\ell)}{\rho_k(\omega_k)} \right| = \left| 1 - \frac{\rho_k(\omega_\ell)}{\rho_k(\omega_k)} \right| \tag{16}$$

As a consequence, we select similar cameras by comparing each visible camera k to the initial camera ℓ that sampled y on the scene (cf. Figure 6).

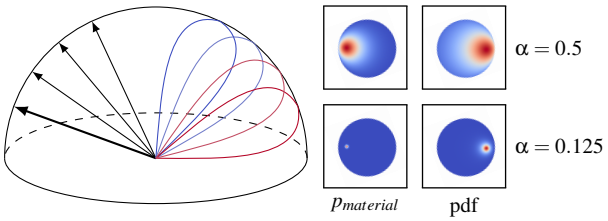


Figure 6: *Given an initial observation direction (thick arrow), the probability for selecting other prefix directions decrease with their distance to the initial distribution. We show the distribution of the probability (cf. Equation 17) over the hemisphere (left), and the pdf of the initial direction for GGX (right).*

In this way, we ensure to add a substantial contribution to the initial camera and minimize the inadequate importance sampling for the others. Since the dissimilarity of glossy distributions increases with the roughness (or shininess), we found that elevating the distance to a power leads to a better fit to the reference. Figure 7 shows convergence plots of the estimators for several roughnesses using 12 cameras. The probability for selecting a camera k similar to camera ℓ finally reads:

$$p_{\text{material}}(\bar{X}_\ell, \bar{X}_k) = \left(1 - \widetilde{\delta}(\rho_\ell, \rho_k)\right)^{\frac{1}{\alpha}} \tag{17}$$

where α is the roughness of the surface at y. We perform the selection before testing the visibility of the prefix by coupling the Jacobian selection and the brdf selection in the same Russian Roulette step. Figure 8 shows that using this additional selection probability indeed improves the result by reducing the variance. Hence the pdf of a prefix is multiplied with the selection probabilities. After selecting the subset of similar viewpoints we uniformly sample the mixture of their lobes to continue the path. The pdf of the sampled brdf direction is then:

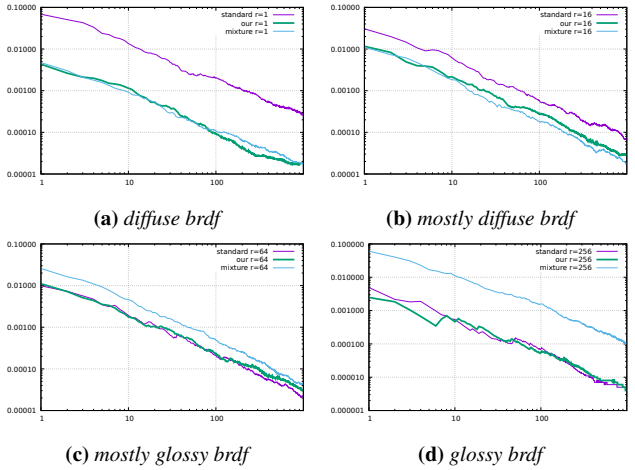
$$p(\omega) = \sum_j p(\omega | x_j y). \tag{18}$$

The path suffix is finally constructed using the classical recursive path tracing.

4.3. Adaptive sampling of the image space

Sharing paths between cameras leads to inhomogeneous sample density over the film due to occlusions and our selective path reuse. This is a classical drawback when reusing paths as pixels lacking samples cannot be predicted. For this reason we use a simple multi-pass adaptive strategy to fill in under-sampled areas (Figure 9).

We first proceed with a pilot iteration which distributes samples uniformly over all pixels. We then compute an error estimation



(a) *diffuse brdf* **(b)** *mostly diffuse brdf*

(c) *mostly glossy brdf* **(d)** *glossy brdf*

Figure 7: *RMSE of the integration of the GGX distribution for one direction among 12 observation directions with varying roughnesses. We compare the standard integration (purple), the multi-view integration by sampling a mixture of all distributions (light blue), and the multi-view integration by first selecting a subset of similar directions and then sampling the mixture (green). The RMSE with the reference shows that our selection performs equivalently to the best strategy. It is interesting to note that the mixture without selection increases the variance of the estimators when alpha decreases.*

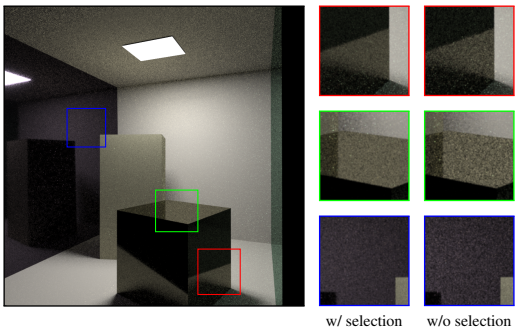


Figure 8: *At equal sample count, using our material selection (left) results in less noise on glossy materials than no selection (right).*

based on the χ^2 divergence [RFS03] for each pixel. We evaluate a total variation noise estimate of the error gradient, similar to the metric proposed by Heitz et al. [HHM18] (cf. Section 4.2). Finally, we use this noise estimate to redistribute samples during the next iteration. This scheme is iterated until a time limit or a target error is reached.

The described noise estimation highlights noisy areas and high gradients, which have been proved correlated with the variance [MKD*16]. The adaptive scheme both redistributes samples in under sampled areas, making them imperceptible, and slightly reduces the noise in areas where error is important. However, any adaptive strategy which focuses on under-sampled areas can be used. The error estimate simply drives the maximum number of

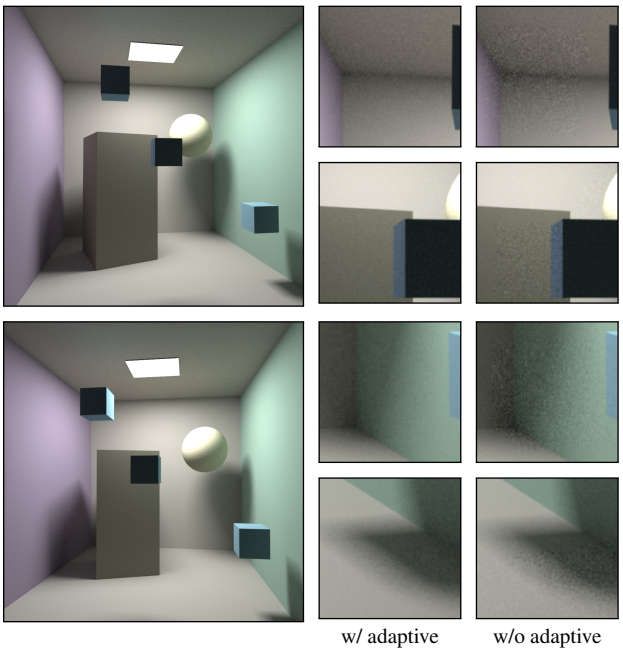


Figure 9: *We rendered 10 images, camera 1 (top left) and 9 camera 2 (bottom left), using our algorithm with and without adaptive sampling. Areas where visibility is not shared exhibit some high level noise in the case of uniform sampling. Adaptive sampling redistributes samples in such areas, making them imperceptible. Right insets show a comparison at equal sample count.*

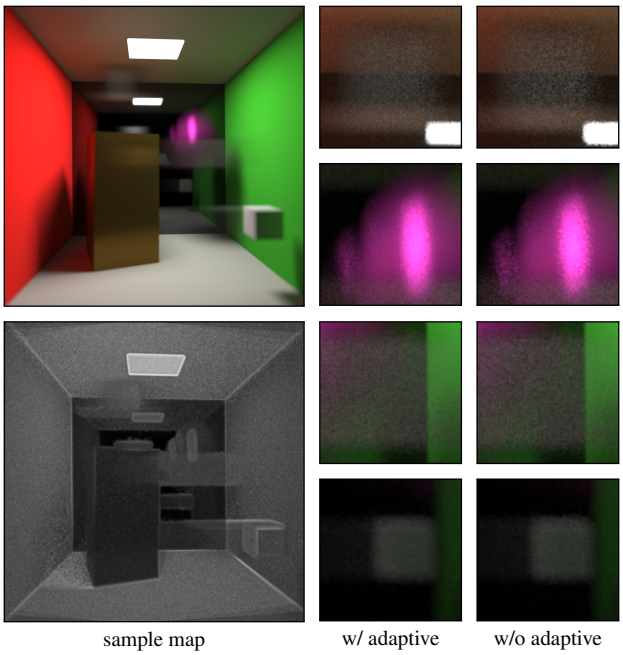


Figure 10: *A Cornell box rendered with or without adaptive refinement and the associated sample map. Our noise estimate focuses well on penumbrae, speckles, indirect lightning, motion blur and gradients. For a given sample budget (256 in this case), the adaptive refinement exhibits less noise on complex areas.*

samples allocated to the pixel. Slow converging, high error pixels will get more samples than fast converging, low error pixels. Figure 10 presents the results of using the aforementioned noise estimation and Figure 11 shows the evolution of the RMSE for a given samples budget.

4.4. Final reconstruction

Each camera selected to use the path contribution adds the contribution with the following MIS weight:

$$w_{\ell\to k}(\bar{X}_\ell) = \frac{n_\ell p_{\ell\to k}(\bar{X}_\ell) p_{\text{jacobian}}(\bar{X}_\ell,\bar{X}_k) p_{\text{material}}(\bar{X}_\ell,\bar{X}_k)}{\sum_j n_j p_{j\to k}(\bar{X}_j) p_{\text{jacobian}}(\bar{X}_j,\bar{X}_k) p_{\text{material}}(\bar{X}_j,\bar{X}_k)} \tag{19}$$

where n_ℓ and n_j are the number of samples generated on the respective pixels by our adaptive sampling strategy. With our approximation that the normalization term $K_{\ell\to k}$ in Equation 9 does not depend on ℓ, $p_{\ell\to k}$ can be computed as:

$$p_{\ell\to k}(\bar{X}_\ell) = p_\ell(\bar{X}_\ell)|T'_{\ell\to k}|V_k(T_{\ell\to k}(\bar{X}_\ell)). \tag{20}$$

5. Results & discussion

We implemented our method into a prototype CPU path tracer supporting multi-threading in C++. All scenes were rendered on a dual Intel(R) Xeon(R) CPU E5-2640 v3 @ 2.60GHz, with 64Gb of memory running Linux. We built on top of Intel(R) Embree kernels and use the Spatio-Temporal BVH [WAB17] to load multiple

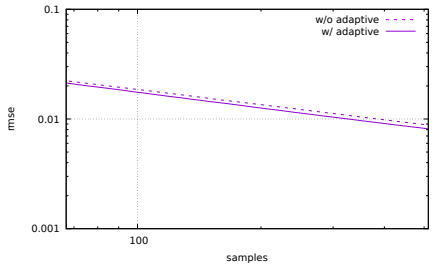
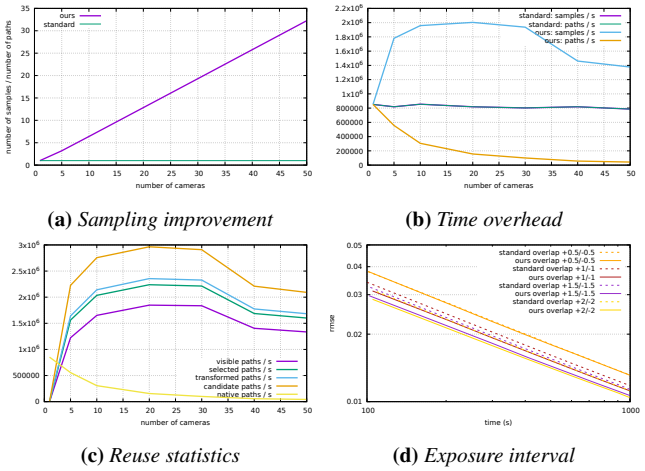


Figure 11: *Evolution of the RMSE with the reference on the animated Cornell sequence with or without adaptive sampling.*

animation segments at once. All test materials are modeled with a glossy lobe with GGX roughness varying from 0.5 to 0.001.

We showed the influence of the similarity selection on a test scene, a specially crafted Cornell Box, with fast moving cubes and glossy materials. This sequence is complex to render since cameras may see tangent geometry while cubes are moving, which implies large variations of the Jacobian and temporal visibility changes.

We rendered a 100 frames barbershop animation sequence with 100 cameras. The cameras shutter intervals overlap to highlight the noise reduction in motion blur, which is reconstructed using a simple triangle filter. Our method compares favorably with equal time path tracing, and achieves a good reduction in noise level with no



(a) *Sampling improvement* **(b)** *Time overhead*

(c) *Reuse statistics* **(d)** *Exposure interval*

Figure 12: *Overhead induced by path reuse*

other artefact. Please compare interactively both results in our supplemental material.

Various statistics on the barbershop sequence are depicted on Figure 12. Figure 12a illustrates the ratio between the number of image contributions and the number of paths. This is the average number of reused paths which increases linearly with the number of observers. Please note that our prototype path tracer doesn't use coherent visibility requests and the performance illustrated in Figure 12b may not be representative of a more optimized renderer. Visibility requests and observer selection eventually cost more than building a single path. This cost is scene and renderer dependent (ray request, suffix path length, material evaluation and sampling, etc). Figure 12c illustrate the effects of the selection. Simple geometric tests filter candidate paths down to selected paths and the last visibility test filters out half the total candidates. Figure 12d compares the RMSE of the teaser image computed with different exposure intervals. These intervals provide candidate paths to select and reuse. The best results are achieved with a large exposure interval, as expected. Without temporal overlap our technique performs exactly like a path tracer in this degenerate case.

5.1. Limitations

Considering multiple observers has an overhead. Asymptotically, the overhead depends quadratically on the number of observers. Evaluating the selection probabilities is required for every observer on every path prefix. We then trace a visibility ray for each selected observer. For a large number of observers, this exhaustive observer loop will need to be replaced by a fixed sampling of the observers coupled with a proximity cache to benefit from coherency.

The efficiency of our algorithm also depends on the whole scene setup:

- the proximity of points of view: our similarity selection will not share paths between distant cameras (both spatially and directionally) ;
- the scene materials: on glossy scenes many paths will not be shared ;
- the scene complexity, which directly affects the cost of visibility queries ;
- the path length which changes the cost of tracing a path, and therefore decreases the relative overhead of our method.

We use a selection probability such that the importance of reused paths is similar to that of the initial one. This directly affects the number of samples we can reuse at a given time. In cases when very few or no projections occur, our method may produce poorer results than the classical one since we have an overhead, to finally obtain the same result. This is the case on the Cornell video in the supplemental, on the highly specular back wall.

Our similarity selection of brdf lobes is derived in an empirical manner and a more formal work is needed to assess its quality. A material selection proportional to a statistical measure, such as the moments of the integral could further improve the convergence of our estimators [SHSK18].

6. Conclusions and perspectives

We have discussed an adaptive and multi-view path tracing technique. The main contribution is an importance sampling heuristic to robustly reuse paths. Compared to previous methods, our heuristic robustly identifies and reuses similar paths with a better control over the variance of the result. Our approach builds on standard path tracing and leaves the overall pipeline unchanged. This heuristic is however derived in an ad-hoc manner and a more formal study of the brdf similarity ad its impact on the variance of the result is necessary.

Combining our simple heuristic with the different path reuse techniques is also an interesting research area.

References

[AH93] ADELSON S. J., HODGES L. F.: Stereoscopic ray-tracing. *The Visual Computer 10*, 3 (Mar 1993), 127–144. 2

[AH95] ADELSON S. J., HODGES L. F.: Generating exact ray-traced animation frames by reprojection. *IEEE Computer Graphics and Applications 15* (1995), 43–52. 2

[BPE17] BAUSZAT P., PETITJEAN V., EISEMANN E.: Gradient-domain path reusing. *ACM Trans. Graph. (Proc. of SIGGRAPH Asia) 36*, 6 (November 2017). 2

[BSH02] BEKAERT P., SBERT M., HALTON J.: Accelerating path tracing by re-using paths. In *Proceedings of the 13th Eurographics Workshop on Rendering* (2002), EGRW '02, Eurographics Association, pp. 125–134. 2

[CT82] COOK R. L., TORRANCE K. E.: A reflectance model for computer graphics. *ACM Transactions on Graphics (TOG) 1*, 1 (1982), 7–24. 5

[FSSK06] FELIU A. M., SBERT M., SZIRMAY-KALOS L.: Reusing frames in camera animation. In *Journal of Winter School of Computer Graphics, Vol. 14* (January 2006), WSCG '06. 2

[HBGM11] HENRICH N., BAERZ J., GROSCH T., MÜLLER S.: Accelerating path tracing by eye-path reprojection. In *International Congress on Graphics and Virtual Reality (GRVR)* (2011). 2

[HDMS03] HAVRAN V., DAMEZ C., MYSZKOWSKI K., SEIDEL H.-P.: An efficient spatio-temporal architecture for animation rendering. In *ACM SIGGRAPH 2003 Sketches & Applications* (2003), SIGGRAPH '03, ACM, pp. 1–1. 2

[HHM18] HEITZ E., HILL S., MCGUIRE M.: Combining analytic direct illumination and stochastic shadows. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (2018), ACM, p. 2. 6

[JM12] JAKOB W., MARSCHNER S.: Manifold exploration: a markov chain monte carlo technique for rendering scenes with difficult specular transport. *ACM Transactions on Graphics (TOG) 31*, 4 (2012), 58. 4

[KMA*15] KETTUNEN M., MANZI M., AITTALA M., LEHTINEN J., DURAND F., ZWICKER M.: Gradient-domain path tracing. *ACM Trans. Graph. 34*, 4 (July 2015), 123:1–123:13. 2

[LKL*13] LEHTINEN J., KARRAS T., LAINE S., AITTALA M., DURAND F., AILA T.: Gradient-domain metropolis light transport. *ACM Trans. Graph. 32*, 4 (July 2013), 95:1–95:12. 2, 4

[MKA*15] MANZI M., KETTUNEN M., AITTALA M., LEHTINEN J., DURAND F., ZWICKER M.: Gradient-domain bidirectional path tracing. In *Eurographics Symposium on Rendering 2015* (2015). VK: Lehtinen, J.; RUIS; HICT. 2

[MKD*16] MANZI M., KETTUNEN M., DURAND F., ZWICKER M., LEHTINEN J.: Temporal gradient-domain path tracing. *ACM Trans. Graph. 35*, 6 (Nov. 2016), 246:1–246:9. 2, 6

[RFS03] RIGAU J., FEIXAS M., SBERT M.: Refinement criteria based on f-divergences. In *Proceedings of the 14th Eurographics Workshop on Rendering* (Aire-la-Ville, Switzerland, Switzerland, 2003), EGRW '03, Eurographics Association, pp. 260–269. 6

[SHSK18] SBERT M., HAVRAN V., SZIRMAY-KALOS L.: Multiple importance sampling revisited: breaking the bounds. *EURASIP Journal on Advances in Signal Processing 2018*, 1 (Feb 2018), 15. 8

[Vea98] VEACH E.: *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, Stanford, CA, USA, 1998. AAI9837162. 3

[VG95] VEACH E., GUIBAS L. J.: Optimally combining sampling techniques for monte carlo rendering. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques* (1995), SIGGRAPH '95, ACM, pp. 419–428. 9

[VG97] VEACH E., GUIBAS L. J.: Metropolis light transport. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques* (1997), ACM, pp. 65–76. 3

[WAB17] WOOP S., ÁFRA A. T., BENTHIN C.: Stbvh: A spatial-temporal bvh for efficient multi-segment motion blur. In *Proceedings of High Performance Graphics* (New York, NY, USA, 2017), HPG '17, ACM, pp. 8:1–8:8. 7

[XS07] XU Q., SBERT M.: A new way to re-using paths. In *Computational Science and Its Applications* (2007), ICCSA 2007, pp. 741–750. 2

[ZRJ*15] ZIMMER H., ROUSSELLE F., JAKOB W., WANG O., ADLER D., JAROSZ W., SORKINE-HORNUNG O., SORKINE-HORNUNG A.: Path-space motion estimation and decomposition for robust animation filtering. *Computer Graphics Forum 34*, 4 (2015), 131–142. 2

Appendix A: General Monte Carlo multiple integration

Let f_A and f_B be two functions integrated over a measure space Ω with given measure μ. Each integrand is non zero over a subset of Ω that we respectively denote Ω_A and Ω_B. Using Monte Carlo integration, these integrals can be estimated by sampling random variables in the sub-domains associated with each integral. For f_A,

$$I_A = \int_{\Omega_A} f_A(x)d\mu(x) \longrightarrow \hat{I}_A = \frac{1}{n}\sum_i^n \frac{f_A(x_i)}{p_A(x_i)} \tag{21}$$

The variance of this estimation can be improved using importance sampling, with a probability density functions (pdf) $p_A \propto f_A$ (respectively $p_B \propto f_B$). Suppose now that we have an injective map T such that Ω_A and $T(\Omega_B)$ overlap (Figure 13). The integration can then be divided in two parts:

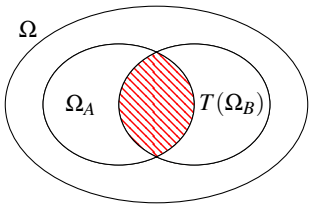


Figure 13: *Overlapping region of the integration domains (in red), which we exploit to increase the sampling of both f_A and f_B.*

$$I_A = \underbrace{\int_{\Omega_A \setminus T(\Omega_B)} f_A(x)d\mu(x)}_{\text{1 strategy}} + \underbrace{\int_{\Omega_A \cap T(\Omega_B)} f_A(x)d\mu(x)}_{\text{2 strategies}} \tag{22}$$

A change of variable in the second term gives

$$\int_{\Omega_A \cap \Omega_B} f_A(x)d\mu(x) = \int_{\Omega_A \cap \Omega_B} f_A(T(x_B))|T'|d\mu(x_B) \tag{23}$$

where $|T'|$ is the jacobian determinant of T. The pdf of such transformed samples is:

$$p_{B\to A}(x_B) = \frac{p_B(x_B)|T'|V_A(T(x_B))}{K_{B\to A}} \tag{24}$$

$$K_{B\to A} = \int_{\Omega_B} p_B(x_B)|T'|V_A(T(x_B))d\mu(x_B) \tag{25}$$

where $p_B(x_B)$ is the pdf of the sample, $V_A(T(x_B))$ is a visibility function which is zero when applied to a point out of Ω_A and K is a normalization constant. This constant is not trivial and requires a numerical integration as well.

Two Monte Carlo estimators can now be used to integrate over $\Omega_A \cap T(\Omega_B)$: sampling can be done either from Ω_A or from Ω_B using T and a visibility test. A robust strategy to weight both estimators and reduce the variance is the balance heuristic [VG95]:

$$w_{B\to A}(x_i) = \frac{p_{B\to A}(x_i)}{p_{B\to A}(x_i) + p_A(x_i)}$$
$$w_{A\to A}(x_i) = \frac{p_A(x_i)}{p_{B\to A}(x_i) + p_A(x_i)}. \tag{26}$$

Finally, the Monte Carlo estimators of $\hat{I}_A$ is:

$$\hat{I}_A = \frac{1}{n_A}\sum_j^{n_A} w_{A\to A}(x_j)\frac{f_A(x_j)}{p_A(x_j)} + \frac{1}{n_B}\sum_j^{n_B} w_{B\to A}(x_j)\frac{f_A(x_j)}{p_{B\to A}(x_j)} \tag{27}$$

and similarly for $\hat{I}_B$. The weight $w_{B\to A}$ equals zero when samples can not be generated from domain Ω_B and $w_{A\to A}$ is 1 in this case.

Both functions can now be jointly integrated with Monte Carlo estimators and samples can be shared where domains overlap. This approach generalizes well to N integrands and N associated importance functions.