Handling Multiple Component Materials in Lightmetrica Version 3

Hisanari Otsu

December 23, 2019

1 Introduction

This document will discuss on the handling of multiple component materials in Lightmetrica Version 3. We will review current implementation based on Russian roulette, and explore the possibility to achieve the feature without it.

2 Problem Definition

BSDF. A *material* in Lightmetrica describes the property of the surface using *bidirectional scattering distribution function* (BSDF). `lm::Material` class provides interfaces for sampling directions, evaluating BSDF, etc. Given a point on the surface $\mathbf{x}$, we write a BSDF as $f_s(\mathbf{x}, \omega_i, \omega_o)$, where ω_i is the incoming direction and ω_o is the outgoing direction. In the following text, we assume the BSDF is defined given the point $\mathbf{x}$ and omit it from the notation depending on the context, e.g., $f_s(\omega_i, \omega_o) \equiv f_s(\mathbf{x}, \omega_i, \omega_o)$.

BSDF with Multiple Components. We will consider a BSDF being composed of *multiple components*. A multiple component BSDF is defined as a sum of different BSDFs:

$$f_s(\omega_i, \omega_o) := \sum_{k=1}^{K} f_s^{(k)}(\omega_i, \omega_o), \tag{1}$$

where $f_s^{(k)}$ being k-th BSDF. For instance, we can consider mixture of mixture of Cook-Torrance BRDF f_G and Lambertian BRDF f_D. In this case, $K = 2$ and Eq. 1 becomes

$$f_s(\omega_i, \omega_o) = f_D(\omega_o) + f_G(\omega_i, \omega_o).$$

Sampling Multiple Component BSDF. We often utilize importance sampling of BSDF. Assuming the surface on which $\mathbf{x}$ is lying is not emissive, according to scattering equation [1], the outgoing radiance L reads

$$\begin{aligned} L(\mathbf{x}, \omega_o) &= \int_{\mathcal{H}} L_i(\mathbf{x}, \omega) f_s(\mathbf{x}, \omega, \omega_o) |\mathbf{N} \cdot \omega| \, d\omega, \\ L(\omega_o) &\equiv \int_{\mathcal{H}} \underbrace{L_i(\omega) f_s(\omega, \omega_o) |\mathbf{N} \cdot \omega|}_{=:F(\omega, \omega_o)} d\omega \equiv \int_{\mathcal{H}} F(\omega, \omega_o) \, d\omega, \end{aligned} \tag{2}$$

where L_i is incoming radiance and $\mathbf{N}$ is angle between surface normal at $\mathbf{x}$. For clarification, in the following discussion, we use shortened symbols as in the second line of the equation. The integral in defined according to solid angle measure $d\omega$ over hemisphere domain $\mathcal{H}$.

We can use the probability density $p(\omega)$ which has roughly the same shape as $f_s \cdot |\mathbf{N} \cdot \omega|$ to importance sample the integrant. The single-sample Monte Carlo estimate of the integral is written as

$$L(\omega_o) \approx \hat{L} := \frac{F(\omega, \omega_o)}{p(\omega)}, \tag{3}$$

where $\omega \sim p(\cdot)$. Here we discuss only about single-sample estimate because in the implementation we use the sampled direction to construct a path.

Now let us consider the case with multiple component BSDF. Substituted by Eq. 1, the scattering equation becomes

$$L(\omega_o) = \int_{\mathcal{H}} L_i(\omega) \left[\sum_{k=1}^{K} f_s^{(k)}(\omega_i, \omega_o) \right] |\mathbf{N} \cdot \omega| \, d\omega. \tag{4}$$

We will consider the problem of estimating Eq. 4. Assume we already know each BSDF component $f_s^{(k)}$ has its own corresponding distributions $p_k(\omega)$.

3 Estimation with Russian Roulette

In this section we will explain the formulation on which the current implementation of multiple component material in Lightmetrica. Currently, the multiple component material is implemented as `Material_WavefrontObj` class.

Table 1: Correspondence between terms or operations in the formulation and the functions in Lightmetrica.

Term	Function	Explanation
$p_{\text{sel}}(k)$	`lm::Scene::pdf_comp()`	Evaluation of selection PMF
$p_k(\omega)$	`lm::Scene::pdf()`	Evaluation of PDF for direction sampling
$f_s^{(k)}(\omega, \omega_o)$	`lm::Scene::eval_contrb()`	Evaluation of BSDF

Estimation with Russian Roulette. We start from Eq. 4. The integrant of the equation is always positive irrespective to the inputs. Thus we can apply Tonelli's theorem to have

$$
\begin{aligned}
L(\omega_o) &= \sum_{k=1}^{K} \int_{\mathcal{H}} \underbrace{L_i(\omega) f_s^{(k)}(\omega, \omega_o) |\mathbf{N} \cdot \omega|}_{=:F^{(k)}(\omega,\omega_o)} d\omega \\
&\equiv \sum_{k=1}^{K} \underbrace{\int_{H} F^{(k)}(\omega, \omega_o) d\omega}_{=:L_k(\omega_o)} \equiv \sum_{k=1}^{K} L_k(\omega_o).
\end{aligned}
$$

To estimate the equation, we can thus separately estimate the summed integrals L_k instead. The current implementation utilizes Russian roulette to estimate it. That is, the estimate

$$
\hat{L}_{\text{RR}}(\omega_o) := \frac{1}{p_{\text{sel}}(k)} \hat{L}_k(\omega_o) \tag{5}
$$

p_{sel} is a PMF taking $k = 1, \ldots, K$ and $k \sim p_{\text{sel}}(\cdot)$. Note that $\sum_{k=1}^{K} p_{\text{sel}}(k) = 1$. Here, $\hat{L}_k$ is another estimate for L_k defined similarly as Eq. 3:

$$
\hat{L}_k(\omega_o) := \frac{F^{(k)}(\omega, \omega_o)}{p_k(\omega)},
$$

where $\omega \sim p_k(\cdot)$. The correspondence between terms or operations in the formulation and the functions in Lightmetrica is described in Table. 1.

Connection to MIS. Estimation with Russian roulette (Eq. 5) is closely related to multiple importance sampling (MIS). In fact, this is a special case of *one-sample model* [1, Sec. 9.2.4]. Using the same notations, the one-sample estimator [1, Eq. 9.15] is defined as

$$
\hat{L}_{1-\text{MIS}}(\omega_o) := \frac{w_k(\omega) F(\omega, \omega_o)}{p_{\text{sel}}(k) p_k(\omega)},
$$

where $k \sim p_{\text{sel}}(\cdot)$ and $\omega \sim p_k(\cdot)$. $w_k(\omega)$ is MIS weight having a condition that its sums over k is one for all ω. We can easily check $\hat{L}_{\text{RR}} = \hat{L}_{1-\text{MIS}}$ only if we define the weighting function as

$$w_k(\omega) := \begin{cases} \frac{F^{(k)}(\omega,\omega_o)}{F(\omega,\omega_o)} & p_k(\omega) > 0 \\ 0 & \text{otherwise.} \end{cases}$$

We can easily show that $\sum_{k=1}^{K} w_k(\omega) = 1$ only if $F(\omega,\omega_o) > 0$. Note that the condition for p_k is necessary to preserve the condition (W2) of the MIS weight [1, p.260]. Also, interestingly, $\hat{L}_{\text{RR}}$ is not optimal in a sense that there exists weighting function with which provably better variance is exhibited than any other weighting functions [1, Theorem 9.4].

Estimating L_k with MIS. We can use MIS to combine multiple different sampling strategies. Please be careful not to confuse with the use of MIS in the above discussion. Here we assume to combine two different sampling strategies (a) BSDF sampling, (b) light sampling, respectively having densities p_{BSDF} and p_{Light}. We want to apply MIS to estimate L_k using multi-sample estimate taking one sample for each strategy. That is, the estimate can be written as

$$\hat{L}_k^{\text{MIS}}(\omega_o) := \sum_{j \in \{\text{BSDF},\text{Light}\}} w_{\text{j}}(\omega_j) \frac{F^{(k)}(\omega,\omega_o)}{p_j(\omega_j)}, \tag{6}$$

where $\omega_j \sim p_j(\cdot)$ and w_{j} is MIS weight (balance heuristic) defined as

$$w_j(\omega_j) = \frac{p_j(\omega_j)}{p_{\text{BSDF}}(\omega_j) + p_{\text{Light}}(\omega_j)}.$$

Replacing $\hat{L}_k(\omega_o)$ with Eq. 6 in Eq. 5, we have a final estimate of L using MIS. Since MIS is applied to estimate L_k after the selection of k from $p_{\text{sel}}(k)$, the PDFs appeared in Eq. 6 and MIS weight do not contain $p_{\text{sel}}(k)$.

Implementation. To implement this approach, the renderer implementation needs to know which component index k currently being selected, since it is used to evaluate some terms in Eq. 6. In Lightmetrica, component and direction sampling are implemented in `lm::Scene::sample_ray()` function. The returned structure includes sampled component index `lm::RaySample::comp` and sampled direction `wo`.

Problem. A problem of this approach is the renderer need to manage component index in the process of sampling and evaluation. As a result, we occasionally need to handle the computation in an unintuitive manner. For instance, the next snippet taken from the implementation of `renderer::volpt` includes unexplained division by `pdf_sel`, which only can be explained with the above discussion.

```
const auto C = throughput / pdf_sel * Tr * fs * sL->weight;
```

An unintuitiveness comes from the fact that we need to estimate sub-problem (estimating L_k) rather than the original problem (estimating L). In the next section we will introduce an another approach to resolve this issue.

4 Consideration in Path Space

Path Space. The discussion in Sec. 3 mainly done based on recursive formulation of light transport (scatter equation, Eq. 2), but the problem can also be discussed with path integral formulation. According to the formulation, the pixel intensity can be represented by a single integral of the measurement contribution function $f(\bar{x})$ over the path space Ω:

$$I = \int_{\Omega} f(\bar{x})\, d\bar{x}, \tag{7}$$

where $\bar{x} = (\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_l) \in \Omega$ is the transport path of length l. The measurement contribution function $f(\bar{x})$ is defined as

$$f(\bar{x}) := W(\mathbf{x}_0, \mathbf{x}_1)\, L_e(\mathbf{x}_l, \mathbf{x}_{l-1})\, G(\mathbf{x}_0, \mathbf{x}_1) \prod_{i=1}^{l-1} f_s(\mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+1})\, G(\mathbf{x}_i, \mathbf{x}_{i+1}), \tag{8}$$

where G being geometry term defined by

$$G(\mathbf{x}, \mathbf{y}) := \frac{\left|\mathbf{N}_{\mathbf{x}} \cdot \omega_{\mathbf{x}\to\mathbf{y}}\right| \cdot \left|\mathbf{N}_{\mathbf{y}} \cdot \omega_{\mathbf{y}\to\mathbf{x}}\right|}{\|\mathbf{x} - \mathbf{y}\|^2}.$$

Here we want to substitute f_s in Eq. 8 for multiple component BSDF (Eq. 1). After some algebraic operations, Eq. 8 becomes (omitting some arguments):

$$f_j(\bar{x}) = \sum_{\mathbf{j} \in \mathcal{J}} \underbrace{\left[W \cdot L_e \cdot G \cdot \prod_{i=1}^{l-1} f_s^{(j_i)}(\mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+1})\, G(\mathbf{x}_i, \mathbf{x}_{i+1}) \right]}_{=: f^{(\mathbf{j})}(\bar{x})} \equiv \sum_{\mathbf{j} \in \mathcal{J}} f^{(\mathbf{j})}(\bar{x}),$$

where $\mathcal{J} \equiv \{1, \ldots, K\}^{l-1}$ is the index set and $\mathbf{j} \equiv (j_1, j_2, \ldots, j_{l-1}) \in \mathcal{J}$ is its element.

$f^{(\mathbf{j})}$ is the measurement contribution function defined for $\mathbf{j}$, replacing i-th occurrence of f_s (from the left) with $f_s^{(j_i)}$. Using this equation, the pixel intensity can be decomposed into a sum of integrals of $f^{(\mathbf{j})}$:

$$I = \sum_{\mathbf{j} \in \mathcal{J}} \underbrace{\int_{\Omega} f^{(\mathbf{j})}(\bar{x})\, d\bar{x}}_{=: I^{(\mathbf{j})}} \equiv \sum_{\mathbf{j} \in \mathcal{J}} I^{(\mathbf{j})}. \tag{9}$$

Now the problem of estimating I boils down to the problem of estimating a sum of $I^{(\mathbf{j})}$. Therefore we can apply a similar arguments to estimate the integral using Russian roulete as we did in Eq. 5.

Bidirectional Path Tracing. Here we will discuss about possible problems that we will encounter in the implementation of bidirectional path tracing (BDPT). The computation process of BDPT is composed of (a) sampling of eye/light subpaths, (b) constructing of full paths by vertex connections, (c) evaluation of contribution and MIS weights. Followed by the notation by Veach, we use the path sampling strategies indexed by a pair of the number of vertices (s, t) where s and t being the number of vertices of eye/light subpaths respectively. Here, we will only consider the path length of l. Thus $s + t = l + 1$.

We will start the discussion from the path integral (Eq. 7). BDPT uses $l + 2$ sampling strategies to sample the path of length l. We denote that path sampled using the densities $p_{s,t}(\cdot)$ corresponding the strategy (s, t) being $\bar{x}_{s,t}$. The multi-sample estimate of I to combine all strategies is written as

$$\hat{I} := \sum_{s,t} w_{s,t}\left(\bar{x}_{s,t}\right) \frac{f\left(\bar{x}_{s,t}\right)}{p_{s,t}\left(\bar{x}_{s,t}\right)}. \tag{10}$$

Note that the paths $\bar{x}_{s,t}$ can be efficiently sampled by the process (a) and (b). We denote the sampled eye/light subpaths in the process (a) being $\hat{x}_s^{\rightarrow} = \left(\mathbf{x}_0^{\rightarrow}, \mathbf{x}_1^{\rightarrow}, \ldots, \mathbf{x}_{s-1}^{\rightarrow}\right)$ and $\hat{x}_t^{\leftarrow} = \left(\mathbf{x}_0^{\leftarrow}, \mathbf{x}_1^{\leftarrow}, \ldots, \mathbf{x}_{t-1}^{\leftarrow}\right)$ respectively, starting from the endpoints (i.e., $\mathbf{x}_0^{\rightarrow}$ represents a point on sensor and $\mathbf{x}_0^{\leftarrow}$ represents a point on light). Since the path is sampled by connecting s-th vertex of eye subpath and t-th vertex (note that when $s = 0$ or $t = 0$ there's no connection), the sampled path can be written as

$$\bar{x}_{s,t} = \left(\hat{x}_s^{\rightarrow}, \text{reverse}\left(\hat{x}_t^{\leftarrow}\right)\right) = \left(\mathbf{x}_0^{\rightarrow}, \mathbf{x}_1^{\rightarrow}, \ldots, \mathbf{x}_{s-1}^{\rightarrow}, \mathbf{x}_{t-1}^{\leftarrow}, \ldots, \mathbf{x}_1^{\leftarrow}, \mathbf{x}_0^{\leftarrow}\right)$$

The PDF $p_{s,t}\left(\bar{x}_{s,t}\right)$ can be written as

$$p_{s,t}\left(\bar{x}_{s,t}\right)=p\left(\hat{x}_s^{\rightarrow}\right)p\left(\hat{x}_t^{\leftarrow}\right)=\left[p\left(\mathbf{x}_0^{\rightarrow}\right)\prod_{i=1}^{s-1}p\left(\mathbf{x}_i^{\rightarrow}\mid\mathbf{x}_{i-1}^{\rightarrow}\right)\right]\cdot\left[p\left(\mathbf{x}_0^{\leftarrow}\right)\prod_{i=1}^{t-1}p\left(\mathbf{x}_i^{\leftarrow}\mid\mathbf{x}_{i-1}^{\leftarrow}\right)\right].$$

Next we will discuss about when multiple component BSDF is considered. Assuming non-endpoint vertices lying on the surface with multi component BSDF (Eq. 1). First we will consider a sampling of subpath $\bar{x}$ (either eye subpath $\bar{x}^{\rightarrow}$ or light subpath $\bar{x}^{\leftarrow}$ with number of vertices $k=s$ or t). The sampling start from the initial vertex $\mathbf{x}_0\sim p(\cdot)$. Given a vertex $\mathbf{x}_i$ $(i=0,\ldots,k-1)$ we first sample the component index $j\sim p_{\text{sel}}(\cdot\mid\mathbf{x}_i)$ and the next vertex based on the selected PDF $\mathbf{x}_{i+1}\sim p^{(j)}(\cdot\mid\mathbf{x}_i)$. For simplicity we also sample the component index for the initial vertex.

We denote the vector of sampled component indices being $\mathbf{j}:=(j_1,j_2,\ldots,j_{k-1})$. By construction, $\mathbf{j}$ and $\bar{x}$ cannot be sampled separately. This means the PDF for $(\mathbf{j},\bar{x})$ pair is a joint PDF:

$$p(\mathbf{j},\bar{x})=p(\mathbf{x}_0)\,p_{\text{sel}}(j_0\mid\mathbf{x}_0)\cdot\prod_{i=1}^{k-1}\left[p_{\text{sel}}(j_i\mid\mathbf{x}_i)\,p^{(j_i)}(\mathbf{x}_i\mid\mathbf{x}_{i-1})\right].$$

Now let us consider the bidirectional case when the path is combined by the strategy (s,t). We denote the sampled indices for both directions being $\mathbf{j}^{\rightarrow}:=\left(j_1^{\rightarrow},\ldots,j_{s-1}^{\rightarrow}\right)$ and $\mathbf{j}^{\leftarrow}:=\left(j_1^{\leftarrow},\ldots,j_{t-1}^{\leftarrow}\right)$ respectively, and combined component indices being

$$\mathbf{j}_{s,t}:=\left(\mathbf{j}^{\rightarrow},\text{reverse}\left(\mathbf{j}^{\leftarrow}\right)\right)=\left(j_1^{\rightarrow},j_2^{\rightarrow},\ldots,j_{s-1}^{\rightarrow},j_{t-1}^{\leftarrow},\ldots,j_2^{\leftarrow},j_1^{\leftarrow}\right).$$

The joint PDF for combined $\left(\mathbf{j}_{s,t},\bar{x}_{s,t}\right)$ pair is written as

$$p_{s,t}\left(\mathbf{j}_{s,t},\bar{x}_{s,t}\right)=p\left(\mathbf{j}^{\rightarrow},\bar{x}_s^{\rightarrow}\right)p\left(\mathbf{j}^{\leftarrow},\bar{x}_t^{\leftarrow}\right).$$

Estimating $I^{(\mathbf{j})}$ Using Russian Roulette. From the fact that the component indices must be sampled in the same time as subpaths, at first it seems Eq. 9 cannot be estimated using Russian roulette because Eq. 5 assumed the component index must be sampled independently. But actually we can use Russian roulette even in this case by expanding multiple integral (according to product area measure) one by one applying Russian roulette in the same time. Although we don't provide the detailed calculation, the final estimate of $I^{(\mathbf{j})}$ is quite similar to Eq. 5:

$$I^{(\mathbf{j}_{s,t})}\approx\frac{f^{(\mathbf{j}_{s,t})}\left(\bar{x}_{s,t}\right)}{p_{s,t}\left(\mathbf{j}_{s,t},\bar{x}_{s,t}\right)},\ \text{where}\ \left(\mathbf{j}_{s,t},\bar{x}_{s,t}\right)\sim p_{s,t}\left(\mathbf{j}_{s,t},\bar{x}_{s,t}\right).$$

Problem with BDPT. BDPT generates full paths by connecting two vertices in eye and light subpaths. As we discussed above, the component selection must happen in the same time as subpath sampling. This implies the process generate $l + 2$ *pairs* of $\left(\mathbf{j}_{s,t}, \bar{x}_{s,t}\right)$. Each pair can be used to estimate $I^{(\mathbf{j})}$, but not usable to estimate other integrals $I^{(\mathbf{j}')}$ where $\mathbf{j} \neq \mathbf{j}'$. In the next section we will discuss on the approach to resolve this issue.

5 Estimation without Russian Roulette

Idea. The cause of difficulties discussed in Sec. 3 and 4 comes from the fact that Russian roulette forces us to estimate sub-problems with separated integrals rather than estimating the original integral. In this section, we will introduce an approach to directly estimate the original integral without Russian roulette. The idea is to use MIS in the extended path space with a pair of component indices and a path. In this section we will only discuss about the application to BDPT, but it is also applicable to the problem in Sec. 3.

Direct Application of MIS. We will start the derivation from the path integral. Using MIS weight, the integral can be decomposed into

$$I = \int_{\Omega} f(x) dx = \sum_{s,t} \underbrace{\int_{\Omega} w_{s,t}\left(\bar{x}\right) f\left(\bar{x}\right) d\bar{x}}_{:=I_{s,t}} = \sum_{s,t} I_{s,t}.$$

We sample $\left(\mathbf{j}_{s,t}, \bar{x}_{s,t}\right)$ pair from the joint density $\left(\mathbf{j}_{s,t}, \bar{x}_{s,t}\right) \sim p_{s,t}\left(\mathbf{j}_{s,t}, \bar{x}_{s,t}\right)$. Estimating $I_{s,t}$ using one-sample estimate (see Appendix), we have

$$I \approx \sum_{s,t} \frac{w\left(\mathbf{j}_{s,t}, \bar{x}_{s,t}\right) \cdot \left[w_{s,t}\left(\bar{x}_{s,t}\right) f\left(\bar{x}_{s,t}\right)\right]}{p_{s,t}\left(\mathbf{j}_{s,t}, \bar{x}_{s,t}\right)},$$

where $w\left(\mathbf{j}_{s,t}, \bar{x}_{s,t}\right)$ is fixed to the balance heuristic:

$$w\left(\mathbf{j}_{s,t}, \bar{x}_{s,t}\right) = \frac{p_{s,t}\left(\mathbf{j}_{s,t}, \bar{x}_{s,t}\right)}{\sum_{\mathbf{j}'_{s,t}} p_{s,t}\left(\mathbf{j}'_{s,t}, \bar{x}_{s,t}\right)} = \frac{p_{s,t}\left(\mathbf{j}_{s,t}, \bar{x}_{s,t}\right)}{p_{s,t}\left(\bar{x}_{s,t}\right)}.$$

Here, $p_{s,t}\left(\bar{x}_{s,t}\right)$ is the marginal PDF defined as

$$p_{s,t}\left(\bar{x}_{s,t}\right) = \sum_{\mathbf{j}_{s,t}} p_{s,t}\left(\mathbf{j}_{s,t}, \bar{x}_{s,t}\right). \tag{11}$$

We use this marginal PDF to construct the first MIS weight $w_{s,t}\left(\bar{x}\right)$. For instance, using balance heuristic:

$$w_{s,t}\left(\bar{x}_{s,t}\right) = \frac{p_{s,t}\left(\bar{x}_{s,t}\right)}{\sum_{s',t'} p_{s',t'}\left(\bar{x}_{s,t}\right)} = \frac{\sum_{\mathbf{j}_{s,t}} p_{s,t}\left(\mathbf{j}_{s,t}, \bar{x}_{s,t}\right)}{\sum_{s',t'} \sum_{\mathbf{j}_{s',t'}} p_{s',t'}\left(\mathbf{j}_{s',t'}, \bar{x}_{s,t}\right)}.$$

Substituted by these equations, the original estimate becomes

$$I \approx \sum_{s,t} \frac{p_{s,t}\left(\bar{x}_{s,t}\right)}{\sum_{s',t'} p_{s',t'}\left(\bar{x}_{s,t}\right)} \cdot \frac{f\left(\bar{x}_{s,t}\right)}{p_{s,t}\left(\bar{x}_{s,t}\right)}. \tag{12}$$

Note that Eq. 12 is quite similar to Eq. 10. The difference is the PDFs are replaced with marginalized PDFs with respect to the component indices.

Computing Marginal. Computing the marginal PDF (Eq. 11) seems intractable, but we can compute it using the following equalities:

$$p_{s,t}\left(\bar{x}_{s,t}\right) = \sum_{\mathbf{j}_s^{\rightarrow}, \mathbf{j}_t^{\leftarrow}} p\left(\mathbf{j}_s^{\rightarrow}, \bar{x}_s^{\rightarrow}\right) p\left(\mathbf{j}_t^{\leftarrow}, \bar{x}_t^{\leftarrow}\right) = \left[\sum_{\mathbf{j}_s^{\rightarrow}} p\left(\mathbf{j}_s^{\rightarrow}, \bar{x}_s^{\rightarrow}\right)\right] \cdot \left[\sum_{\mathbf{j}_t^{\leftarrow}} p\left(\mathbf{j}_t^{\leftarrow}, \bar{x}_t^{\leftarrow}\right)\right],$$

where

$$\sum_{\mathbf{j}_s^{\rightarrow}} p\left(\mathbf{j}_s^{\rightarrow}, \bar{x}_s^{\rightarrow}\right) = p\left(\mathbf{x}_0^{\rightarrow}\right) \cdot \prod_{i=1}^{s-1} p\left(\mathbf{x}_i^{\rightarrow} \mid \mathbf{x}_{i-1}^{\rightarrow}\right),$$
$$\sum_{\mathbf{j}_t^{\leftarrow}} p\left(\mathbf{j}_t^{\leftarrow}, \bar{x}_t^{\leftarrow}\right) = p\left(\mathbf{x}_0^{\leftarrow}\right) \cdot \prod_{i=1}^{t-1} p\left(\mathbf{x}_i^{\leftarrow} \mid \mathbf{x}_{i-1}^{\leftarrow}\right),$$

and ($\leftrightarrow$ is either $\rightarrow$ or $\leftarrow$):

$$p\left(\mathbf{x}_0^{\leftrightarrow}\right) = \sum_{j_0^{\rightarrow}} p\left(\mathbf{x}_0^{\leftrightarrow}\right) p_{\mathrm{sel}}\left(j_0^{\leftrightarrow} \mid \mathbf{x}_0^{\leftrightarrow}\right),$$
$$p\left(\mathbf{x}_i^{\leftrightarrow} \mid \mathbf{x}_{i-1}^{\leftrightarrow}\right) = \sum_{j_i^{\leftrightarrow}} p_{\mathrm{sel}}\left(j_i^{\leftrightarrow} \mid \mathbf{x}_i^{\leftrightarrow}\right) p^{\left(j_i^{\leftrightarrow}\right)}\left(\mathbf{x}_i^{\leftrightarrow} \mid \mathbf{x}_{i-1}^{\leftrightarrow}\right).$$

These equations imply the marginal PDF can be computed by a product of *local* marginal PDFs with respect to the component index of the current vertex. Aside from the marginal PDFs, the equations are same as path sampling PDF of standard BDPT. This means we can utilize the almost same code as standard BDPT to implement this approach.

Implementation. Since the computation of marginal PDF can be done locally and does not depends on any additional information (as is component index), the computation can utilize the existing interface of `lm::Material` class. From the discussion above, the users of the material interface do not need to know the existence of the multi component material. We can just use the interface as if it is a normal single component material. The applications discussed in Sec. 3 and 4 can be implemented without any special care of the underlying multiple component materials.

References

[1] Eric Veach. *Robust Monte Carlo methods for light transport simulation.* PhD thesis, Stanford University, USA, 1998. AAI9837162.

A Deriving One-/Multi-sample Estimate

In this appendix we will show the relationship between one-sample estimate and multi-sample estimate of MIS. We will start from the integral to be estimated $I = \int_\Omega f(x)dx$. We can assume that the support of f is Ω and $f(x) > 0$. Using the condition (W1) of MIS weight, we have $\sum_{k=1}^{K} w_k(x) = 1$ for all $x \in \Omega$.

$$I = \int_\Omega 1 \cdot f(x)\, d = \int_\Omega \left[\sum_{k=1}^{K} w_k(x)\right] f(x)\, dx.$$

Using the fact that $w_k(x) f(x) > 0$ for all x, we can apply Tonelli's theorem to exchange the integral and the sum:

$$= \sum_{k=1}^{K} \underbrace{\int_\Omega w_k(x) f(x)\, dx}_{=:I_k} \equiv \sum_{k=1}^{K} I_k. \tag{13}$$

Multi-sample Estimate. The multi-sample estimate can be derived by estimating $\{I_k\}$ using k different samples $\{x_k\}$ where $x_k \sim p_k(\cdot)$:

$$I_k \approx \hat{I}_k := \frac{w_k(x_k) f(x_k)}{p_k(x_k)} \Rightarrow I \approx \hat{I}_{\text{multi-MIS}} = \sum_{k=1}^{K} \frac{w_k(x_k) f(x_k)}{p_k(x_k)}.$$

One-sample Estimate. To derive the one-sample estimate, we first apply Russian roulette to Eq. 5. Denoting the PMF for selecting k-th strategy being $p_{\text{sel}}(k)$, the estimate can be written as

$$I \approx I_{\text{RR}} := \frac{I_k}{p_{\text{sel}}(k)}.$$

Next we take a sample x from the selected k-th strategy: $x \sim p_k(\cdot)$ and estimate I_k. Then the desired estimate can be obtained:

$$I_k \approx \hat{I}_k := \frac{w_k(x_k) f(x_k)}{p_k(x_k)} \Rightarrow I \approx \hat{I}_{\text{1-MIS}} = \frac{w_k(x_k) f(x_k)}{p_{\text{sel}}(k) p_k(x_k)}. \tag{14}$$

One-sample Estimate Using Joint Density. A natural extension of one-sample estimate is to use joint density $p(k, x)$ to sample a (k, x) pair:

$$I \approx \hat{I}_{\text{1-MIS-J}} = \frac{w_k(x) f(x)}{p(k, x)}.$$

The one-sample estimate is the case where $p(k, x) = p_{\text{sel}}(k) p_k(x)$. It is not obvious that $\hat{I}_{\text{1-MIS-J}}$ is unbiased for any weights. At least we can show $\hat{I}_{\text{1-MIS-J}}$ is unbiased when w_k is balance heuristic, by using marginalized density $\sum_{k=1}^{K} p(k, x)$:

$$I \approx \frac{f(x)}{\sum_{k'=1}^{K} p(k', x)} = \underbrace{\frac{p(k, x)}{\sum_{k'=1}^{K} p(k', x)}}_{w_k(x)} \cdot \frac{f(x)}{p(k, x)}.$$