Supplemental Document for ReSTIR BDPT

TREVOR HEDSTROM, University of California San Diego, USA
MARKUS KETTUNEN, NVIDIA, Finland
DAQI LIN, NVIDIA, USA
CHRIS WYMAN, NVIDIA, USA
TZU-MAO LI, University of California San Diego, USA

ACM Reference Format:
Trevor Hedstrom, Markus Kettunen, Daqi Lin, Chris Wyman, and Tzu-Mao Li. 2025. Supplemental Document for ReSTIR BDPT. *ACM Trans. Graph.* 0, 0, Article 0 (2025), 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 RECURSIVE MIS WEIGHTS FOR RECONNECTION

In this section, we derive a formula for Georgiev et al.; van Antwerpen's [2012; 2011] recursive MIS weights that is more suitable for our ReSTIR BDPT algorithm. We first prove a non-recursive form for Equation 6; that will be Equation 7. Then, we present new cached quantities accessible during our path sampling and reuse (Section 1.4), and give Equation 6 a new expression using these quantities (Section 1.5).

1.1 Notation

The quantity $\overrightarrow{p}_i$ denotes the area-measure probability density of sampling the vertex x_i from the vertex x_{i-1} while tracing from the camera. Similarly, the quantity $\overleftarrow{p}_i$ denotes the area-measure probability density of sampling vertex x_i from the vertex x_{i+1} while tracing from the light. The area-measure densities are calculated using the geometry terms as

$$\overrightarrow{p}_i = \overrightarrow{p}^{\sigma}_i \cdot \overrightarrow{g}_i \tag{1}$$

$$\overleftarrow{p}_i = \overleftarrow{p}^{\sigma}_i \cdot \overleftarrow{g}_i \,, \tag{2}$$

where

$$\overrightarrow{p}^{\sigma}_i = \begin{cases} p^{\sigma}(x_{i-2} \to x_{i-1} \to x_i) & x_{i-1} \text{ nondelta} \\ p^{\text{delta}}(x_{i-1} \to x_i) & \text{otherwise} \end{cases} \tag{3}$$

$$\overleftarrow{p}^{\sigma}_i = \begin{cases} p^{\sigma}(x_{i+2} \to x_{i+1} \to x_i) & x_{i+1} \text{ nondelta} \\ p^{\text{delta}}(x_{i+1} \to x_i) & \text{otherwise.} \end{cases} \tag{4}$$

Here, $p^{\sigma}(x \to x' \to x'')$ is the solid-angle probability density of sampling the vertex x'' from x', following Veach's three-point notation [Veach 1997], and p^{delta} is the probability of selecting the delta component of the material, or 1 for a delta-only material. This formulation allows us to manipulate the $\overrightarrow{p}_i$ and $\overleftarrow{p}_i$ quantities without separate cases for delta PDFs, since $\overrightarrow{p}^{\sigma}_i$ and $\overleftarrow{p}^{\sigma}_i$ do not contain deltas.

1.2 Recursive MIS weights

The recursive MIS algorithm computes recursive *partial MIS quantities* at each subpath vertex. For camera subpaths, these quantities are computed with

$$d^{\text{p}}_i = [x_{i-1} \text{ nondelta}] \left(\frac{1}{\overrightarrow{p_i}}\right)^{\beta} , \tag{5}$$

$$d^{\text{VC}}_i = \left(\frac{\overleftarrow{g_{i-1}}}{\overrightarrow{p_i}}\right)^{\beta} \left([x_{i-1} \text{ nondelta}]\, d^{\text{p}}_{i-1} + \left(\overleftarrow{p}^{\sigma}_{i-2}\right)^{\beta} d^{\text{VC}}_{i-1}\right). \tag{6}$$

where the bracket notation evaluates to 1 if the expression inside the bracket is true, or 0 otherwise. These quantities are computed and stored with the camera subpath $\bar{z}$. The full MIS weight $\omega_{s,t}$ is then recovered using the recursive quantities from the camera and light subpath endpoints y_{s-1} and z_{t-1} as described in Section 6.1.

After reconnecting to the reconnection vertex z_r, we can compute d^{p}_r and d^{VC}_r using Equation 5 and Equation 6. In order to compute the full MIS weight, we require d^{p}_{t-1} and d^{VC}_{t-1} from the end of the camera subpath, which can be an arbitrary number of vertices after the reconnection vertex z_r. Calculating d^{VC}_{t-1} using Equation 6 requires visiting all vertices between z_r and z_{t-1}. Instead, we derive an algorithm to compute the recursive quantity d^{VC}_{t-1} from d^{VC}_r, without visiting vertices between z_r and z_{t-1}.

1.3 Non-recursive d^{VC}_{t-1}

We first prove a non-recursive expression for d^{VC}_{t-1}. We define

$$\widetilde{d}^{\text{VC}}_{r+n} = \left(\frac{\overleftarrow{g}_{r+n-1}}{\overrightarrow{p}_{r+n}}\right)^{\beta} \left(\sum_{i=r}^{r+n-1} [x_i \text{ nondelta}]\, d^{\text{p}}_i \prod_{j=i}^{r+n-2} \left(\frac{\overleftarrow{p}_j}{\overrightarrow{p}_{j+1}}\right)^{\beta} + \left(\overleftarrow{p}^{\sigma}_{r-1}\right)^{\beta} d^{\text{VC}}_r \prod_{i=r}^{r+n-2} \left(\frac{\overleftarrow{p}_i}{\overrightarrow{p}_{i+1}}\right)^{\beta} \right), \tag{7}$$

where $n = t-1-r$ counts the number of camera subpath vertices after the reconnection vertex, and prove by induction that $\widetilde{d}^{\text{VC}}_{r+n} = d^{\text{VC}}_{r+n}$ for $n \geq 1$.

1.3.1 *Proof.*

Authors' addresses: Trevor Hedstrom, University of California San Diego, USA, tjhedstr@ucsd.edu; Markus Kettunen, NVIDIA, Finland, mkettunen@nvidia.com; Daqi Lin, NVIDIA, USA, daqil@nvidia.com; Chris Wyman, NVIDIA, USA, chris.wyman@acm.org; Tzu-Mao Li, University of California San Diego, USA, tzli@ucsd.edu.

0730-0301/2025/0-ART0 $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Case $n = 1$.

$$\begin{aligned}
\widetilde{d}^{\mathrm{VC}}_{r+1} &= \left(\frac{\overleftarrow{g}_r}{\overrightarrow{p}_{r+1}}\right)^\beta \left(\begin{array}{l} \displaystyle\sum_{i=r}^{r} [x_i \text{ nondelta}]\, d^{\mathrm{p}}_i \prod_{j=i}^{r-1} \left(\frac{\overleftarrow{p}_j}{\overrightarrow{p}_{j+1}}\right)^\beta \\ + \left(\overleftarrow{p}^{\sigma}_{r-1}\right)^\beta d^{\mathrm{VC}}_r \cancel{\displaystyle\prod_{i=r}^{r-1} \left(\frac{\overleftarrow{p}_i}{\overrightarrow{p}_{i+1}}\right)^\beta} \end{array}\right) \\
&= \left(\frac{\overleftarrow{g}_r}{\overrightarrow{p}_{r+1}}\right)^\beta \left(\begin{array}{l} [x_r \text{ nondelta}]\, d^{\mathrm{p}}_r \cancel{\displaystyle\prod_{j=r}^{r-1} \left(\frac{\overleftarrow{p}_j}{\overrightarrow{p}_{j+1}}\right)^\beta} \\ + \left(\overleftarrow{p}^{\sigma}_{r-1}\right)^\beta d^{\mathrm{VC}}_r \end{array}\right) \\
&= \left(\frac{\overleftarrow{g}_r}{\overrightarrow{p}_{r+1}}\right)^\beta \left([x_r \text{ nondelta}]\, d^{\mathrm{p}}_r + \left(\overleftarrow{p}^{\sigma}_{r-1}\right)^\beta d^{\mathrm{VC}}_r\right) \\
&= d^{\mathrm{VC}}_{r+1}.
\end{aligned}$$

Case $n = k + 1$. We start with the non-recursive form of d^{VC}_{r+k+1},

$$d^{\mathrm{VC}}_{r+k+1} = \left(\frac{\overleftarrow{g}_{r+k}}{\overrightarrow{p}_{r+k+1}}\right)^\beta \left([x_{r+k} \text{ nondelta}]\, d^{\mathrm{p}}_{r+k} + \left(\overleftarrow{p}^{\sigma}_{r+k-1}\right)^\beta d^{\mathrm{VC}}_{r+k}\right).$$

Then, assuming $d^{\mathrm{VC}}_{r+k} = \widetilde{d}^{\mathrm{VC}}_{r+k}$,

$$d^{\mathrm{VC}}_{r+k+1} = \left(\frac{\overleftarrow{g}_{r+k}}{\overrightarrow{p}_{r+k+1}}\right)^\beta \left(\begin{array}{l} [x_{r+k} \text{ nondelta}]\, d^{\mathrm{p}}_{r+k} \\ + \left(\overleftarrow{p}^{\sigma}_{r+k-1}\right)^\beta \left(\dfrac{\overleftarrow{g}_{r+k-1}}{\overrightarrow{p}_{r+k}}\right)^\beta \cdot \\ \left(\begin{array}{l} \displaystyle\sum_{i=r}^{r+k-1} [x_i \text{ nondelta}]\, d^{\mathrm{p}}_i \prod_{j=i}^{r+k-2} \left(\frac{\overleftarrow{p}_j}{\overrightarrow{p}_{j+1}}\right)^\beta \\ + \left(\overleftarrow{p}^{\sigma}_{r-1}\right)^\beta d^{\mathrm{VC}}_r \displaystyle\prod_{i=r}^{r+k-2} \left(\frac{\overleftarrow{p}_i}{\overrightarrow{p}_{i+1}}\right)^\beta \end{array}\right) \end{array}\right).$$

We simplify $\overleftarrow{p}^{\sigma}_{r+k-1} \cdot \overleftarrow{g}_{r+k-1} = \overleftarrow{p}_{r+k-1}$ by Equation 2 to get

$$= \left(\frac{\overleftarrow{g}_{r+k}}{\overrightarrow{p}_{r+k+1}}\right)^\beta \left(\begin{array}{l} [x_{r+k} \text{ nondelta}]\, d^{\mathrm{p}}_{r+k} \\ + \left(\dfrac{\overleftarrow{p}_{r+k-1}}{\overrightarrow{p}_{r+k}}\right)^\beta \displaystyle\sum_{i=r}^{r+k-1} [x_i \text{ nondelta}]\, d^{\mathrm{p}}_i \prod_{j=i}^{r+k-2} \left(\frac{\overleftarrow{p}_j}{\overrightarrow{p}_{j+1}}\right)^\beta \\ + \left(\dfrac{\overleftarrow{p}_{r+k-1}}{\overrightarrow{p}_{r+k}}\right)^\beta \left(\overleftarrow{p}^{\sigma}_{r-1}\right)^\beta d^{\mathrm{VC}}_r \displaystyle\prod_{i=r}^{r+k-2} \left(\frac{\overleftarrow{p}_i}{\overrightarrow{p}_{i+1}}\right)^\beta \end{array}\right).$$

We now move the ratio $\left(\overleftarrow{p}_{r+k-1}/\overrightarrow{p}_{r+k}\right)^\beta$ into the products, which increases their upper bound to $r + k - 1$:

$$= \left(\frac{\overleftarrow{g}_{r+k}}{\overrightarrow{p}_{r+k+1}}\right)^\beta \left(\begin{array}{l} [x_{r+k} \text{ nondelta}]\, d^{\mathrm{p}}_{r+k} \\ + \displaystyle\sum_{i=r}^{r+k-1} [x_i \text{ nondelta}]\, d^{\mathrm{p}}_i \prod_{j=i}^{r+k-1} \left(\frac{\overleftarrow{p}_j}{\overrightarrow{p}_{j+1}}\right)^\beta \\ + \left(\overleftarrow{p}^{\sigma}_{r-1}\right)^\beta d^{\mathrm{VC}}_r \displaystyle\prod_{i=r}^{r+k-1} \left(\frac{\overleftarrow{p}_i}{\overrightarrow{p}_{i+1}}\right)^\beta \end{array}\right).$$

Next, we move the term $[x_{r+k} \text{ nondelta}]\, d^{\mathrm{p}}_{r+k}$ into the sum, which increases its upper bound to $r + k$. This is possible because the product is empty for $j = r + k$, yielding

$$\begin{aligned}
&= \left(\frac{\overleftarrow{g}_{r+k}}{\overrightarrow{p}_{r+k+1}}\right)^\beta \left(\begin{array}{l} \displaystyle\sum_{i=r}^{r+k} [x_i \text{ nondelta}]\, d^{\mathrm{p}}_i \prod_{j=i}^{r+k-1} \left(\frac{\overleftarrow{p}_j}{\overrightarrow{p}_{j+1}}\right)^\beta \\ + \left(\overleftarrow{p}^{\sigma}_{r-1}\right)^\beta d^{\mathrm{VC}}_r \displaystyle\prod_{i=r}^{r+k-1} \left(\frac{\overleftarrow{p}_i}{\overrightarrow{p}_{i+1}}\right)^\beta \end{array}\right) \\
&= \widetilde{d}^{\mathrm{VC}}_{r+k+1}.
\end{aligned}$$

1.4 Additional quantities

Our algorithm works by computing and storing these extra quantities when the camera subpath is first sampled:

$$\bar{\gamma} = \left(\frac{\overleftarrow{g}_{t-2}}{\overrightarrow{p}_{t-1}}\right)^\beta, \tag{8}$$

$$\bar{\lambda}^{\mathrm{VC}} = \left(\frac{\overleftarrow{p}_r}{\overrightarrow{g}_{r+1}}\right)^\beta \prod_{i=r+1}^{t-3} \left(\frac{\overleftarrow{p}_i}{\overrightarrow{p}_{i+1}}\right)^\beta, \tag{9}$$

$$\bar{\lambda}^{\mathrm{P}} = [x_{r+1} \text{ nondelta}] \prod_{i=r+1}^{t-3} \left(\frac{\overleftarrow{p}_i}{\overrightarrow{p}_{i+1}}\right)^\beta, \tag{10}$$

$$\bar{\sigma} = \sum_{i=r+2}^{t-2} [x_i \text{ nondelta}]\, d^{\mathrm{p}}_i \prod_{j=i}^{t-3} \left(\frac{\overleftarrow{p}_j}{\overrightarrow{p}_{j+1}}\right)^\beta. \tag{11}$$

Next, we give a new formula for d^{VC}_{t-1} using these cached quantities.

1.5 Computing d^{VC}_{t-1}

Case $r = t - 1$. If the reconnection vertex is the last vertex on the camera subpath, then Equation 6 can be evaluated directly.

Case $r = t - 2$. If the reconnection vertex is the second-to-last vertex on the camera subpath, we rewrite Equation 6 using Equation 1 to separate the geometry term in the denominator:

$$d^{\mathrm{VC}}_{t-1} = \left(\frac{\overleftarrow{g}_r}{\overrightarrow{p}^{\sigma}_{r+1}\overrightarrow{g}_{r+1}}\right)^\beta \left(d^{\mathrm{p}}_r + \left(\overleftarrow{p}^{\sigma}_{r-1}\right)^\beta d^{\mathrm{VC}}_r\right), \tag{12}$$

In this case, $[x_r \text{ nondelta}]$ is always 1 since x_r must be nondelta for reconnection to occur. The quantities $\overrightarrow{p}^{\sigma}_{r+1}$, d^{p}_r, and $\overleftarrow{p}^{\sigma}_{r-1}$ are computed during reconnection, and the remaining quantities are cached during initial sampling.

Case $r < t - 2$. If the reconnection vertex is before the second-to-last vertex on the camera subpath, we rewrite Equation 6 using eqs. (8) to (11)

$$d^{\mathrm{VC}}_{t-1} = \bar{\gamma} \left(\begin{array}{l} \left(\dfrac{1}{\overrightarrow{p}^{\sigma}_{r+1}}\right)^\beta \bar{\lambda}^{\mathrm{VC}} \left(d^{\mathrm{p}}_r + \left(\overleftarrow{p}^{\sigma}_{r-1}\right)^\beta d^{\mathrm{VC}}_r\right) \\ + \left(\dfrac{1}{\overrightarrow{p}^{\sigma}_{r+1}}\right)^\beta \left(\dfrac{1}{\overrightarrow{g}_{r+1}}\right)^\beta \bar{\lambda}^{\mathrm{P}} + \bar{\sigma} \end{array}\right), \tag{13}$$

which we prove equal to $\widetilde{d}^{\mathrm{VC}}_{t-1}$ in the following.

1.5.1 Proof. We start by substituting $\bar{\lambda}^{\mathrm{VC}}$ from Equation 9 into Equation 13:

$$d^{\mathrm{VC}}_{t-1} = \bar{y}\left(\begin{array}{l}\left(\frac{1}{\overrightarrow{p}^{\sigma}_{r+1}}\right)^{\beta}\left[\left(\frac{\overleftarrow{p}_r}{\overrightarrow{g}_{r+1}}\right)^{\beta}\prod_{i=r+1}^{t-3}\left(\frac{\overleftarrow{p}_i}{\overrightarrow{p}_{i+1}}\right)^{\beta}\right]\left(d^{\mathrm{P}}_r + \left(\overleftarrow{p}^{\sigma}_{r-1}\right)^{\beta} d^{\mathrm{VC}}_r\right)\\ + \left(\frac{1}{\overrightarrow{p}^{\sigma}_{r+1}}\right)^{\beta}\left(\frac{1}{\overrightarrow{g}_{r+1}}\right)^{\beta}\bar{\lambda}^{\mathrm{P}} + \bar{\sigma}\end{array}\right).$$

Combining factors and substituting $\overrightarrow{p}^{\sigma}_{r+1}\cdot\overrightarrow{g}_{r+1} = \overrightarrow{p}_{r+1}$ by Equation 1 yields

$$= \bar{y}\left(\begin{array}{l}\left(\frac{\overleftarrow{p}_r}{\overrightarrow{p}_{r+1}}\right)^{\beta}\left(\prod_{i=r+1}^{t-3}\left(\frac{\overleftarrow{p}_i}{\overrightarrow{p}_{i+1}}\right)^{\beta}\right)\left(d^{\mathrm{P}}_r + \left(\overleftarrow{p}^{\sigma}_{r-1}\right)^{\beta} d^{\mathrm{VC}}_r\right)\\ + \frac{1}{\left(\overrightarrow{p}_{r+1}\right)^{\beta}}\bar{\lambda}^{\mathrm{P}} + \bar{\sigma}\end{array}\right).$$

We merge the ratio $\left(\frac{\overleftarrow{p}_r}{\overrightarrow{p}_{r+1}}\right)^{\beta}$ into the product as $i = r$, giving

$$= \bar{y}\left(\begin{array}{l}\left(\prod_{i=r}^{t-3}\left(\frac{\overleftarrow{p}_i}{\overrightarrow{p}_{i+1}}\right)^{\beta}\right)\left(d^{\mathrm{P}}_r + \left(\overleftarrow{p}^{\sigma}_{r-1}\right)^{\beta} d^{\mathrm{VC}}_r\right)\\ + \frac{1}{\left(\overrightarrow{p}_{r+1}\right)^{\beta}}\bar{\lambda}^{\mathrm{P}} + \bar{\sigma}\end{array}\right).$$

Expanding the parentheses now yields

$$= \bar{y}\left(\begin{array}{l}d^{\mathrm{P}}_r\prod_{i=r}^{t-3}\left(\frac{\overleftarrow{p}_i}{\overrightarrow{p}_{i+1}}\right)^{\beta} + \left(\overleftarrow{p}^{\sigma}_{r-1}\right)^{\beta} d^{\mathrm{VC}}_r\prod_{i=r}^{t-3}\left(\frac{\overleftarrow{p}_i}{\overrightarrow{p}_{i+1}}\right)^{\beta}\\ + \frac{1}{\left(\overrightarrow{p}_{r+1}\right)^{\beta}}\bar{\lambda}^{\mathrm{P}} + \bar{\sigma}\end{array}\right).$$

We now substitute $\bar{y}$, $\bar{\lambda}^{\mathrm{P}}$, and $\bar{\sigma}$ (Equations 8, 10, 11) to reach

$$= \left(\frac{\overleftarrow{g}_{t-2}}{\overrightarrow{p}_{t-1}}\right)^{\beta}\left(\begin{array}{l}d^{\mathrm{P}}_r\prod_{i=r}^{t-3}\left(\frac{\overleftarrow{p}_i}{\overrightarrow{p}_{i+1}}\right)^{\beta} + \left(\overleftarrow{p}^{\sigma}_{r-1}\right)^{\beta} d^{\mathrm{VC}}_r\prod_{i=r}^{t-3}\left(\frac{\overleftarrow{p}_i}{\overrightarrow{p}_{i+1}}\right)^{\beta}\\ + \frac{1}{\left(\overrightarrow{p}_{r+1}\right)^{\beta}}[x_{r+1}\text{ nondelta}]\prod_{i=r+1}^{t-3}\left(\frac{\overleftarrow{p}_i}{\overrightarrow{p}_{i+1}}\right)^{\beta}\\ + \sum_{i=r+2}^{t-2}[x_i\text{ nondelta}]\, d^{\mathrm{P}}_i\prod_{j=i}^{t-3}\left(\frac{\overleftarrow{p}_j}{\overrightarrow{p}_{j+1}}\right)^{\beta}\end{array}\right).$$

Since the reconnection vertex x_r and its predecessor x_{r-1} must be non-delta for reconnection to occur, we have $[x_r\text{ nondelta}] = 1$, and the first term becomes

$$d^{\mathrm{P}}_r\prod_{i=r}^{t-3}\left(\frac{\overleftarrow{p}_i}{\overrightarrow{p}_{i+1}}\right)^{\beta} = [x_r\text{ nondelta}]\, d^{\mathrm{P}}_r\prod_{j=r}^{t-3}\left(\frac{\overleftarrow{p}_j}{\overrightarrow{p}_{j+1}}\right)^{\beta},$$

i.e., term $i = r$ in the bottom sum. Similarly, we have by Equation 5

$$\frac{1}{\left(\overrightarrow{p}_{r+1}\right)^{\beta}} = [x_r\text{ nondelta}]\frac{1}{\left(\overrightarrow{p}_{r+1}\right)^{\beta}} = d^{\mathrm{P}}_{r+1},$$

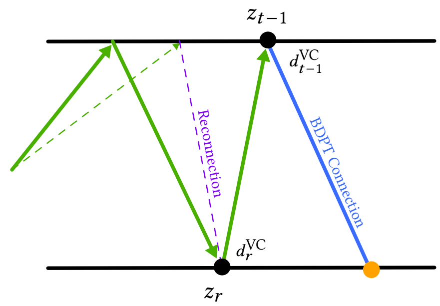

Fig. 1. If $r = t-2$, we only need to advance d^{VC}_r by one bounce to recover d^{VC}_{t-1}.

so the middle term becomes term $i = r+1$ in the bottom sum. Hence, we reach

$$d^{\mathrm{VC}}_{t-1} = \left(\frac{\overleftarrow{g}_{t-2}}{\overrightarrow{p}_{t-1}}\right)^{\beta}\left(\begin{array}{l}\sum_{i=r}^{t-2}[x_i\text{ nondelta}]\, d^{\mathrm{P}}_i\prod_{j=i}^{t-3}\left(\frac{\overleftarrow{p}_j}{\overrightarrow{p}_{j+1}}\right)^{\beta}\\ + \left(\overleftarrow{p}^{\sigma}_{r-1}\right)^{\beta} d^{\mathrm{VC}}_r\prod_{i=r}^{t-3}\left(\frac{\overleftarrow{p}_i}{\overrightarrow{p}_{i+1}}\right)^{\beta}\end{array}\right)$$
$$= \widetilde{d}^{\mathrm{VC}}_{r+n}\text{ with } n = t-1-r.$$

2 BOUNDS ON ERROR FROM TECHNIQUE MIS REUSE

In the following, we show that the relative bias from reusing old samples' technique MIS weights is bounded by their relative error.

In our proposed method, we select an extended path $\hat{Y} = (Y, \tau)$ with GRIS by resampling from M candidates $\hat{X}_1, \ldots, \hat{X}_M$, with $\hat{X}_i = (X_i, \tau_i)$, according to resampling weights w_i. We then select a single candidate $\hat{X}_z$ and shift it into the target domain with shift map T_z:

$$\hat{Y} = T_z(\hat{X}_z), \tag{14}$$

mapping X_z from its domain Ω_z to the target Ω with the shift mapping corresponding to its sampling technique τ_z, retaining the sampling technique:

$$T_z(\hat{X}_z) = (T_{z,\tau_z}(X_z), \tau_z). \tag{15}$$

Once $\hat{Y}$ is selected, the integral I is estimated using the MIS-weighted GRIS estimator $\langle I\rangle$:

$$\langle I\rangle = \omega_\tau(Y) f(Y) W_{\hat{Y}}. \tag{16}$$

If we modify the right side of Equation 16 to instead use the unshifted candidate's MIS weight $\omega_\tau(X_z)$, we form a new estimator

$$\begin{aligned}\langle I_{\text{biased}}\rangle &= \omega_\tau(X_z) f(Y) W_{\hat{Y}}\\ &= \omega_\tau(T^{-1}_{z,\tau}(Y)) f(Y) W_{\hat{Y}}.\end{aligned} \tag{17}$$

We first define the contribution error as the technique MIS weight error scaled by the integrand:

$$f^{\Delta}_i(\hat{Y}) = f(Y)\left|\omega_\tau(Y) - \omega_\tau\left(T^{-1}_{i,\tau}(Y)\right)\right|. \tag{18}$$

Now, we write the bias as the expectation of the difference of Equation 16 and Equation 17:

$$\begin{aligned}|\text{Bias}| &= \left|\mathbb{E}\left[\langle I\rangle - \langle I_{\text{biased}}\rangle\right]\right| \\ &= \left|\mathbb{E}\left[\omega_\tau(Y)f(Y)W_{\hat{Y}} - \omega_\tau\left(T^{-1}_{z,\tau}(Y)\right)f(Y)W_{\hat{Y}}\right]\right| \\ &\leq \mathbb{E}\left[\left|\omega_\tau(Y) - \omega_\tau\left(T^{-1}_{z,\tau}(Y)\right)\right| f(Y)W_{\hat{Y}}\right] \\ &= \mathbb{E}\left[f^{\Delta}_z(\hat{Y})W_{\hat{Y}}\right].\end{aligned} \tag{19}$$

We write this as an expectation over the possible choices $T_i(\hat{X}_i)$ for $\hat{Y}$, noting that the selection probability $P_s(i) = 0$ if $\hat{X}_i$ cannot be shifted with T_i. With

$$W_{\hat{Y}} = \frac{1}{\hat{p}(\hat{Y})}\sum_{i=1}^{M} w_i, \tag{20}$$

we reach $|\text{Bias}|$

$$\leq \sum_{i=1}^{M}\mathbb{E}\left[f^{\Delta}_i(T_i(\hat{X}_i))\left(\frac{1}{\hat{p}(T_i(\hat{X}_i))}\sum_{j=1}^{M} w_j\right)[P_s(i) > 0]P_s(i)\right] \tag{21}$$

$$\begin{aligned}&= \sum_{i=1}^{M}\mathbb{E}\left[f^{\Delta}_i(T_i(\hat{X}_i))\left(\frac{1}{\hat{p}(T_i(\hat{X}_i))}\cancel{\sum_{j=1}^{M} w_j}\right)[P_s(i) > 0]\frac{w_i}{\cancel{\sum_{j=1}^{M} w_j}}\right] \\ &= \sum_{i=1}^{M}\mathbb{E}\left[[\hat{p}(T_i(\hat{X}_i)) > 0]f^{\Delta}_i(T_i(\hat{X}_i))\frac{w_i}{\hat{p}(T_i(\hat{X}_i))}\right] \\ &= \sum_{i=1}^{M}\mathbb{E}\left[[\hat{p}(T_i(\hat{X}_i)) > 0]f^{\Delta}_i(T_i(\hat{X}_i))\frac{m_i(T_i(\hat{X}_i))\cancel{\hat{p}(T_i(\hat{X}_i))}W_{\hat{X}_i}\left|\frac{\partial T_i}{\partial \hat{X}_i}\right|}{\cancel{\hat{p}(T_i(\hat{X}_i))}}\right].\end{aligned}$$

On line 3, the bracket $[\hat{p}(T_i(\hat{X}_i)) > 0]$ first requires that $T_i(\hat{X}_i)$ is defined, i.e., $\hat{X}_i \in \text{Dom}(T_i)$; we leave this implicit for brevity. All summands are now of form $\mathbb{E}\left[g(\hat{X}_i)W_{\hat{X}_i}\right]$, and by the definition of unbiased contribution weights, we write

$$\begin{aligned}&= \sum_{i=1}^{M}\int_{\text{supp}\,\hat{X}_i}[\hat{p}(T_i(\hat{x}))>0]f^{\Delta}_i(T_i(\hat{x}))m_i(T_i(\hat{x}))\left|\frac{\partial T_i}{\partial \hat{x}}\right|\mathrm{d}\hat{x} \\ &= \sum_{i=1}^{M}\int_{\text{supp}\,\hat{X}_i\cap\text{Dom}(T_i)}[\hat{p}(T_i(\hat{x})) > 0]f^{\Delta}_i(T_i(\hat{x}))m_i(T_i(\hat{x}))\left|\frac{\partial T_i}{\partial \hat{x}}\right|\mathrm{d}\hat{x}.\end{aligned} \tag{22}$$

The change of variables $\hat{y} = T_i(\hat{x}_i)$ now yields

$$= \sum_{i=1}^{M}\int_{T_i(\text{supp}\,\hat{X}_i\cap\text{Dom}(T_i))}[\hat{p}(\hat{y}) > 0]f^{\Delta}_i(\hat{y})m_i(\hat{y})\,\mathrm{d}\hat{y}, \tag{23}$$

which by exchanging the domain and the bracket is equivalent with

$$= \sum_{i=1}^{M}\int_{\text{supp}\,\hat{p}}[\hat{y} \in T_i(\text{supp}\,\hat{X}_i \cap \text{Dom}(T_i))]f^{\Delta}_i(\hat{y})m_i(\hat{y})\,\mathrm{d}\hat{y}. \tag{24}$$

We now drop the bracket, since the resampling MIS weights m_i require $m_i(\hat{y}) = 0$ if there exist no $\hat{x} \in \text{supp}\,\hat{X}_i$ such that $\hat{y} = T_i(\hat{x})$. We reach

$$= \sum_{i=1}^{M}\int_{\text{supp}\,\hat{p}} f^{\Delta}_i(\hat{y})m_i(\hat{y})\,\mathrm{d}\hat{y}. \tag{25}$$

Let us now assume the relative technique MIS weight error is smaller than some arbitrary ϵ_ω:

$$\frac{\left|\omega_\tau(\bar{y}) - \omega_\tau\left(T^{-1}_{i,\tau}(\bar{y})\right)\right|}{\omega_\tau(\bar{y})} \leq \epsilon_\omega, \tag{26}$$

where $\hat{y} = (\bar{y}, \tau)$. Continuing from Equation 25, we reach the inequality

$$\begin{aligned}|\text{Bias}| &\leq \sum_{i=1}^{M}\int_{\text{supp}\,\hat{p}} f^{\Delta}_i(\hat{y})m_i(\hat{y})\,\mathrm{d}\hat{y} \\ &= \sum_{i=1}^{M}\int_{\text{supp}\,\hat{p}}\left(f(\bar{y})\left|\omega_\tau(\bar{y}) - \omega_\tau\left(T^{-1}_{i,\tau}(\bar{y})\right)\right|\right)m_i(\hat{y})\,\mathrm{d}\hat{y} \\ &\leq \sum_{i=1}^{M}\int_{\text{supp}\,\hat{p}}\epsilon_\omega\;\omega_\tau(\bar{y})\;f(\bar{y})m_i(\hat{y})\,\mathrm{d}\hat{y} \\ &= \epsilon_\omega\int_{\text{supp}\,\hat{p}}\omega_\tau(\bar{y})\;f(\bar{y})\left(\sum_{i=1}^{M} m_i(\hat{y})\right)\mathrm{d}\hat{y}.\end{aligned}$$

Since the original estimator is unbiased, the MIS weights sum to 1, and we reach

$$\begin{aligned}&= \epsilon_\omega\int_{\text{supp}\,\hat{p}}\omega_\tau(\bar{y})\;f(\bar{y})\,\mathrm{d}\hat{y} \\ &= \epsilon_\omega\,\mathbb{E}\left[\langle I\rangle\right] \\ &= \epsilon_\omega I,\end{aligned}$$

i.e.,

$$|\text{Bias}| \leq \epsilon_\omega\; I,$$

the relative bias from reusing technique MIS weights is bounded by their relative error.

3 DATA STRUCTURES

The Light Vertex Cache is a simple array of LightSubpathVertex structs:

```
struct LightSubpathVertex {
    SceneVertex   y_0
    PathVertex    y_{s-1}
    float3        throughput
    float         subpathPdf
    uint          subpathSeed
    uint          subpathId
    uint16_t      numVertices
    uint16_t      numBounces
    float         d^VC
    float         d^P
}
```

In our implementation, the PathVertex struct is 48 bytes as it contains information needed to fetch all shading and geometry data, while the SceneVertex struct is only 16 bytes as it only contains information needed to fetch geometry data. The full LightSubpathVertex struct is 112 bytes (including padding for alignment).

Our per-pixel path reservior struct is shown below. Here, the MisEvalData struct is 84 bytes, containing all information required

```
struct PathReservoir {
    float          UCW
    float          confidenceWeight  // c_i
    uint           cameraSubpathSeed
    uint           cameraSubpathId
    uint8_t        bounces
    uint8_t        prefixBounces  // Bounces before x_r
    uint8_t        prefixDiffuseBounces
    uint8_t        flags  // Caustic or non-caustic
    float3         pathF  // f(x̄)
    float          misWeight  // ω_τ
    float          prefixPdf  // p(x_0 → ... → x_r)
    PathVertex     x_r  // Reconnection vertex
    float3         suffixF    // f(x_r → ... → x_{s+t-1})
    float          suffixPdf  // p(x_r → ... → x_{s+t-1})
    float          reconnectionCos  // |n_r · (x_{r-1} - x_r) / ||x_{r-1} - x_r|| |
    float          reconnectionDist  // ||x_{r-1} - x_r||
    float3         dirOut  // (x_{r+1} - x_r) / ||x_{r+1} - x_r||
    SceneVertex    y_0
    uint           lightPathSeed
    uint           lightPathId
    float          lightPathPdf  // p(y_0 → ... → y_{s-1})
    uint16_t       lightPathVertices
    uint16_t       lightPathDiffuseBounces
    MisEvalData    cachedMisData
}
```

to compute $\bar{w}_{s-1}$ and $\bar{w}_{t-1}$, which are used to compute the full MIS weight ω_τ (Equation 33 in the main text).

REFERENCES

Iliyan Georgiev, Jaroslav Křivánek, Tomáš Davidovič, and Philipp Slusallek. 2012. Light Transport Simulation with Vertex Connection and Merging. *ACM Transactions on Graphics (TOG)* 31, 6, Article 192 (Nov. 2012), 10 pages. https://doi.org/10.1145/2366145.2366211

D. van Antwerpen. 2011. Recursive MIS Computation for Streaming BDPT on the GPU. (2011).

Eric Veach. 1997. *Robust Monte Carlo Methods for Light Transport Simulation*. Ph. D. Dissertation.