ReSTIR Subsurface Scattering for Real-Time Path Tracing - Supplemental Material

MIRCO WERNER, Karlsruhe Institute of Technology, Germany
VINCENT SCHÜSSLER, Karlsruhe Institute of Technology, Germany
CARSTEN DACHSBACHER, Karlsruhe Institute of Technology, Germany

CCS Concepts: • **Computing methodologies → Ray tracing**.

Additional Key Words and Phrases: subsurface scattering, diffusion profile, BSSRDF importance sampling, ReSTIR, real-time rendering

ACM Reference Format:
Mirco Werner, Vincent Schüßler, and Carsten Dachsbacher. 2024. ReSTIR Subsurface Scattering for Real-Time Path Tracing - Supplemental Material. *Proc. ACM Comput. Graph. Interact. Tech.* 7, 3, Article 1 (July 2024), 8 pages. https://doi.org/10.1145/3675372

1 SHIFT MAPPINGS AND JACOBIANS

In this section, we describe the two building blocks of our reconnection and delayed reconnection shifts. A shift mapping T maps a base path $\bar{\boldsymbol{x}} = [\boldsymbol{x}_0, \boldsymbol{x}_1, \dots]$ in domain Ω_i to the offset path $T(\bar{\boldsymbol{x}}) = \bar{\boldsymbol{y}} = [\boldsymbol{y}_0, \boldsymbol{y}_1, \dots]$ in domain Ω. The camera vertex $\boldsymbol{x}_0$ and the primary hit $\boldsymbol{x}_1$ are fixed for a given pixel and time, i.e. for a given domain. T generates the free vertices $\boldsymbol{y}_2, \boldsymbol{y}_3, \dots$ of the offset path based on the base path $\bar{\boldsymbol{x}}$. We denote $\boldsymbol{\omega}_k^{\boldsymbol{x}}$ as the unit vector from $\boldsymbol{x}_k$ to $\boldsymbol{x}_{k+1}$, $\theta_k^{\boldsymbol{x}}$ the angle between $\boldsymbol{\omega}_k^{\boldsymbol{x}}$ and the normal at $\boldsymbol{x}_{k+1}$, and $\boldsymbol{u}_k^{\boldsymbol{x}}$ the random numbers that lead from $\boldsymbol{x}_k$ to $\boldsymbol{x}_{k+1}$. Figure 1 shows an example for the notation.

Reconnection. The path space shift reconnection [Hua et al. 2019] reconnects the offset path to the base path at a vertex $\boldsymbol{x}_{k+1}$ and reuses all following vertices, i.e.

$$T([\boldsymbol{x_0}, \boldsymbol{x_1}, ..., \boldsymbol{x_k}, \boldsymbol{x_{k+1}}, \boldsymbol{x_{k+2}}, ...]) = [\boldsymbol{y_0}, \boldsymbol{y_1}, ..., \boldsymbol{y_k}, \boldsymbol{x_{k+1}}, \boldsymbol{x_{k+2}}, ...]. \tag{1}$$

Reconnection has the Jacobian [Kettunen et al. 2015]

$$\left|\frac{\partial \boldsymbol{\omega}_k^{\boldsymbol{y}}}{\partial \boldsymbol{\omega}_k^{\boldsymbol{x}}}\right| = \left|\frac{\cos\theta_k^{\boldsymbol{y}}}{\cos\theta_k^{\boldsymbol{x}}}\right| \frac{||\boldsymbol{x}_{k+1} - \boldsymbol{x}_k||^2}{||\boldsymbol{x}_{k+1} - \boldsymbol{y}_k||^2} \quad \text{or} \quad \left|\frac{\partial \boldsymbol{y}_{k+1}}{\partial \boldsymbol{x}_{k+1}}\right| = \left|\frac{\partial \boldsymbol{x}_{k+1}}{\partial \boldsymbol{x}_{k+1}}\right| = 1 \tag{2}$$

for solid angle parametrization or area measure, respectively.

Authors' addresses: Mirco Werner, mirco.werner@student.kit.edu, Karlsruhe Institute of Technology, Karlsruhe, Germany; Vincent Schüßler, vincent.schuessler@kit.edu, Karlsruhe Institute of Technology, Karlsruhe, Germany; Carsten Dachsbacher, dachsbacher@kit.edu, Karlsruhe Institute of Technology, Karlsruhe, Germany.

ACM 2577-6193/2024/7-ART1
https://doi.org/10.1145/3675372

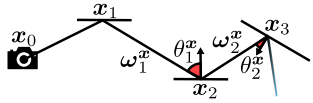

Fig. 1. Parametrization of the base path $\boldsymbol{x}$.

Random Replay. The PSS shift random replay [Hua et al. 2019] generates vertices on the offset path by copying the random numbers from the base path, i.e.

$$T([\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, ...]) = [\boldsymbol{y}_0, \boldsymbol{y}_1, \boldsymbol{y}_2, ...], \tag{3}$$

where $\boldsymbol{y}_2, \boldsymbol{y}_3, ...$ are generated with the random numbers used to generate $\boldsymbol{x}_2, \boldsymbol{x}_3, ...$, respectively. The Jacobian for generating $\boldsymbol{y}_{k+1}$ by random replay can be derived by transforming to PSS.

$$\left|\frac{\partial \omega_k^{\boldsymbol{y}}}{\partial \omega_k^{\boldsymbol{x}}}\right| = \left|\frac{\partial \omega_k^{\boldsymbol{y}}}{\partial \boldsymbol{u}_k^{\boldsymbol{y}}}\right| \left|\frac{\partial \boldsymbol{u}_k^{\boldsymbol{y}}}{\partial \boldsymbol{u}_k^{\boldsymbol{x}}}\right| \left|\frac{\partial \boldsymbol{u}_k^{\boldsymbol{x}}}{\partial \omega_k^{\boldsymbol{x}}}\right| \quad \text{or} \quad \left|\frac{\partial \boldsymbol{y}_{k+1}}{\partial \boldsymbol{x}_{k+1}}\right| = \left|\frac{\partial \boldsymbol{y}_{k+1}}{\partial \boldsymbol{u}_k^{\boldsymbol{y}}}\right| \left|\frac{\partial \boldsymbol{u}_k^{\boldsymbol{y}}}{\partial \boldsymbol{u}_k^{\boldsymbol{x}}}\right| \left|\frac{\partial \boldsymbol{u}_k^{\boldsymbol{x}}}{\partial \boldsymbol{x}_{k+1}}\right| \tag{4}$$

The transformation to PSS is given by [Kelemen et al. 2002]

$$d\boldsymbol{u}_k^{\boldsymbol{x}} = p_{\boldsymbol{x}_k}(\omega_k^{\boldsymbol{x}}) d\omega_k^{\boldsymbol{x}} \quad \text{or} \quad d\boldsymbol{u}_k^{\boldsymbol{x}} = p_{\boldsymbol{x}_k}(\boldsymbol{x}_{k+1}) d\boldsymbol{x}_{k+1}, \tag{5}$$

where p is the PDF in solid angle or area measure, respectively. Copying the random numbers $\boldsymbol{u}_k^{\boldsymbol{x}} = \boldsymbol{u}_k^{\boldsymbol{y}}$ leads to $\left|\partial \boldsymbol{u}_k^{\boldsymbol{y}} / \partial \boldsymbol{u}_k^{\boldsymbol{x}}\right| = 1$ and Eq. (4) reduces to

$$\left|\frac{\partial \omega_k^{\boldsymbol{y}}}{\partial \omega_k^{\boldsymbol{x}}}\right| = \frac{p_{\boldsymbol{x}_k}(\omega_k^{\boldsymbol{x}})}{p_{\boldsymbol{y}_k}(\omega_k^{\boldsymbol{y}})} \quad \text{or} \quad \left|\frac{\partial \boldsymbol{y}_{k+1}}{\partial \boldsymbol{x}_{k+1}}\right| = \frac{p_{\boldsymbol{x}_k}(\boldsymbol{x}_{k+1})}{p_{\boldsymbol{y}_k}(\boldsymbol{y}_{k+1})}, \tag{6}$$

the ratio of the two PDFs.

2 EXAMPLE FOR THE INVERTIBLE SAMPLING OF A MIXTURE DISTRIBUTION

In the main paper, we show in Appendix A that for a suitable inversion $R\colon \mathcal{P} \times \Gamma \to \mathcal{U}$ to a sampling scheme $S\colon \mathcal{U} \to \mathcal{P} \times \Gamma$ the Jacobians have to cancel

$$\left|J_S(R(\bar{\boldsymbol{x}}, \bar{\boldsymbol{\gamma}}))\right| \left|J_R(\bar{\boldsymbol{x}}, \bar{\boldsymbol{\gamma}})\right| = 1, \tag{7}$$

so that the path density and the UCW do not change. We ensure this by choosing R such that this property holds following Bitterli et al. [2017]. Here, we give an explicit example for sampling a mixture distribution and its inversion. In our example, we use a sampling scheme

$$\begin{aligned} S\colon \mathcal{U} &\to \mathcal{P} \times \Gamma \\ (u_1, u_2, u_3) &\mapsto (x, \gamma_1, \gamma_2) \end{aligned} \tag{8}$$

that is a mixture of two techniques with technique-specific sampling schemes

$$\begin{aligned} S_1, S_2\colon \mathcal{U}^* &\to \mathcal{P} \\ u_3 &\mapsto x \end{aligned} \tag{9}$$

with PDFs p_1, p_2 and weights $\alpha_1, \alpha_2 = 1 - \alpha_1$ for importance sampling x. Intuitively, γ_1 determines the discrete technique index (1 or 2) while γ_2 determines the random number u_2 in the continuous interval that maps to the specific technique ($[0, \alpha_1]$ or $[\alpha_1, 1]$).

Sampling. For sampling, we first select a technique index t and sample x using the respective technique-specific sampling scheme S_t:

$$t = \begin{cases} 1 & \text{if } u_2 < \alpha_1, \\ 2 & \text{otherwise,} \end{cases} \qquad x = S_t(u_3). \tag{10}$$

Although we never explicitly compute it in practice, the equations for γ_1, γ_2 are:

$$\gamma_1 = \begin{cases} \frac{u_1 \alpha_1 p_1(x)}{\alpha_1 p_1(x) + \alpha_2 p_2(x)} & \text{if } t = 1, \\ \frac{\alpha_1 p_1(x) + u_1 \alpha_2 p_2(x)}{\alpha_1 p_1(x) + \alpha_2 p_2(x)} & \text{if } t = 2, \end{cases} \qquad \gamma_2 = \begin{cases} \frac{u_2}{\alpha_1} & \text{if } t = 1, \\ \frac{u_2 - \alpha_1}{\alpha_2} & \text{if } t = 2. \end{cases} \tag{11}$$

The Jacobian of S is the reciprocal PDF of the sampling scheme

$$|J_S(u_1, u_2, u_3)| = \frac{1}{\alpha_1 p_1(x) + \alpha_2 p_2(x)}. \tag{12}$$

Inversion. For the inversion R, we again select the technique index t first

$$t \sim p(t \mid x) = \frac{\alpha_t p_t(x)}{\sum_{t' \in \{1,2\}} \alpha_{t'} p_{t'}(x)} \quad \Rightarrow \quad t = \begin{cases} 1 & \text{if } \gamma_1 < \frac{\alpha_1 p_1(x)}{\alpha_1 p_1(x) + \alpha_2 p_2(x)}, \\ 2 & \text{otherwise.} \end{cases} \tag{13}$$

Then, we invert x using the respective inverse technique-specific sampling scheme $R_t = S_t^{-1}$ and compute the remaining u_1 and u_2:

$$u_1 = \begin{cases} \gamma_1 \frac{\alpha_1 p_1(x) + \alpha_2 p_2(x)}{\alpha_1 p_1(x)} & \text{if } t = 1, \\ \left(\gamma_1 - \frac{\alpha_1 p_1(x)}{\alpha_1 p_1(x) + \alpha_2 p_2(x)}\right) \frac{\alpha_1 p_1(x) + \alpha_2 p_2(x)}{\alpha_2 p_2(x)} & \text{if } t = 2, \end{cases} \quad u_2 = \begin{cases} \alpha_1 \gamma_2 & \text{if } t = 1, \\ \alpha_1 + \alpha_2 \gamma_2 & \text{if } t = 2, \end{cases} \quad u_3 = R_t(x). \tag{14}$$

The Jacobian of R is then

$$|J_R(x, \gamma_1, \gamma_2)| = \left|\frac{\partial u_1}{\partial \gamma_1}\right| \left|\frac{\partial u_2}{\partial \gamma_2}\right| \left|\frac{\partial u_3}{\partial x}\right| = \frac{\alpha_1 p_1(x) + \alpha_2 p_2(x)}{\alpha_t p_t(x)} \cdot \alpha_t \cdot p_t(x) = \alpha_1 p_1(x) + \alpha_2 p_2(x). \tag{15}$$

Therefore, both Jacobians (Eq. (12) and Eq. (15)) do cancel in this case:

$$\left|J_S(R(\bar{\boldsymbol{x}}, \bar{\boldsymbol{\gamma}}))\right| \left|J_R(\bar{\boldsymbol{x}}, \bar{\boldsymbol{\gamma}})\right| = |J_S(u_1, u_2, u_3)| \, |J_R(x, \gamma_1, \gamma_2)| = \frac{\alpha_1 p_1(x) + \alpha_2 p_2(x)}{\alpha_1 p_1(x) + \alpha_2 p_2(x)} = 1. \tag{16}$$

This example can be generalized to a mixture of an arbitrary number of techniques [Bitterli et al. 2017].

3 SEQUENTIAL SHIFT FOR RESTIR WITH SURFACE REFLECTIONS

Similar to Section 4.3 in the main paper, where we explain the application of the sequential shift for ReSTIR SSS, we now give an example in the presumably more familiar context of surface reflections, i.e. BSDF sampling, to clarify the application of the probabilistic inversion. Therefore, we consider paths that connect the camera with the light source and contain only surface reflections at the vertices (cf. Fig. 1). We assume to have a BSDF with a diffuse lobe l_d and a metal lobe l_m with lobe-specific sampling techniques S_{l_d} and S_{l_m} and conditional PDFs $p_{\boldsymbol{x},l_d}$ and $p_{\boldsymbol{x},l_m}$ that may depend on the current vertex $\boldsymbol{x}$. The material at $\boldsymbol{x}$ specifies the probabilities $p_{\boldsymbol{x}}(l_d)$ and $p_{\boldsymbol{x}}(l_m)$ of selecting a certain lobe for sampling. The PDF for a sampled vertex $\boldsymbol{x}'$ given the vertex $\boldsymbol{x}$ is

$$p_{\boldsymbol{x}}(\boldsymbol{x}') = p_{\boldsymbol{x}}(l_d) p_{\boldsymbol{x},l_d}(\boldsymbol{x}') + p_{\boldsymbol{x}}(l_m) p_{\boldsymbol{x},l_m}(\boldsymbol{x}'). \tag{17}$$

BSDF sampling with multiple lobes is an example for sampling a mixture distribution, where multiple lobes can produce the same sample. Hence, $\bar{\boldsymbol{\gamma}} \in \Gamma$ determines the lobe (technique). During reconnection shift, we reconnect the offset path vertex $\boldsymbol{y}_k$ with the base path vertex $\boldsymbol{x}_{k+1}$ and need

to obtain matching random numbers that are cached in the reservoir for a subsequent random replay pass. We draw $\gamma_1 \in [0, 1]$ to select a possible lobe l based on the likelihood that it generated the sample:

$$l \sim p(l \mid \boldsymbol{x_{k+1}}) = \frac{p_{\boldsymbol{y}_k}(l) p_{\boldsymbol{y}_k, l}(\boldsymbol{x}_{k+1})}{\sum_{l' \in \{l_d, l_m\}} p_{\boldsymbol{y}_k}(l') p_{\boldsymbol{y}_k, l'}(\boldsymbol{x}_{k+1})} \tag{18}$$

After the lobe is fixed, either $S_{l_d}^{-1}$ or $S_{l_m}^{-1}$ has to be used to determine the random numbers $\boldsymbol{u}_{\text{sample}}$ for sampling $\boldsymbol{x}_{k+1}$ with lobe l. This, for example, can involve inverting cosine hemisphere sampling or inverting sampling of a microfacet model. Since $\bar{\boldsymbol{\gamma}}$ selects the lobe based on varying material properties, we need to find a matching random number u_{lobe} for selecting the lobe. We draw $\gamma_2 \in [0, 1]$ and set

$$u_{\text{lobe}} = \begin{cases} p_{\boldsymbol{x}}(l_d)\gamma_2 & \text{if } l = l_d, \\ p_{\boldsymbol{x}}(l_d) + p_{\boldsymbol{x}}(l_m)\gamma_2 & \text{if } l = l_m. \end{cases} \tag{19}$$

We store u_{lobe} and $\boldsymbol{u}_{\text{sample}}$ in the reservoir, which are accessed in a subsequent resampling pass with delayed reconnection during random replay. Instead of calculating and caching the random numbers directly after reconnection, it is possible to apply the probabilistic inversion directly during random replay, since changing $\bar{\boldsymbol{\gamma}}$ does not require modification of the UCW such that it can be sampled lazily when needed (cf. Section 4.2 in the main paper). This reduces memory overhead but can lead to more frequent computation of probabilistic inverses.

4 HYBRID AND SEQUENTIAL: (IMPLICIT) SHIFT SELECTION

In Fig. 2, we visualize whether the hybrid shift selects reconnection or delayed reconnection using its criterion and compare it to the sequential shift that can pick the most successful shift implicitly.

5 SEARCH RADIUS FOR NEIGHBOR RESAMPLING

Spatial reuse randomly selects pixels in a certain radius. On the one hand, if the radius is too small, correlation artifacts appear. On the other hand, if the radius is chosen too large, the acceptance rate drops, because samples are too far apart. ReSTIR DI [Bitterli et al. 2020] uses a constant 30 pixel search radius around the current pixel. In our case, we propose to search in a radius of $2d_{\max} = 2\max(\boldsymbol{d}.r, \max(\boldsymbol{d}.g, \boldsymbol{d}.b))$. As this distance is in world space, we project it to screen space. Reservoirs with valid samples store vertices that are usually less far apart as $d_{\max}$ since $\boldsymbol{d}$ is a scaled version of the mean free path. For reconnection shift, it is then only necessary to search in a radius of $2d_{\max}$. Of course, this assumption breaks when using random replay during delayed reconnection shift, since we do not reconnect through the translucent object. Generally, however, the geometry changes over the distance, making random replay for disk-based sampling less effective, such that our non-constant search radius can be helpful after all. Nevertheless, to avoid too low or too high values, we clamp the search radius between 5 and 30. This dynamic search radius ensures a smaller search radius for translucent objects that are far away and captures only a small region of the screen and vice versa. In practice, this often leads only to a small improvement.

6 UNBIASED CONVERGENCE

All shift mappings in ReSTIR SSS can be implemented without introducing any bias. While there is a slight and rather unnoticeable bias with our implementation of the sequential shift due to the delayed calculation of the intersection index (main paper Section 4.3), without this performance optimization the sequential shift is unbiased as well. We show unbiased convergence in spatiotemporal reuse

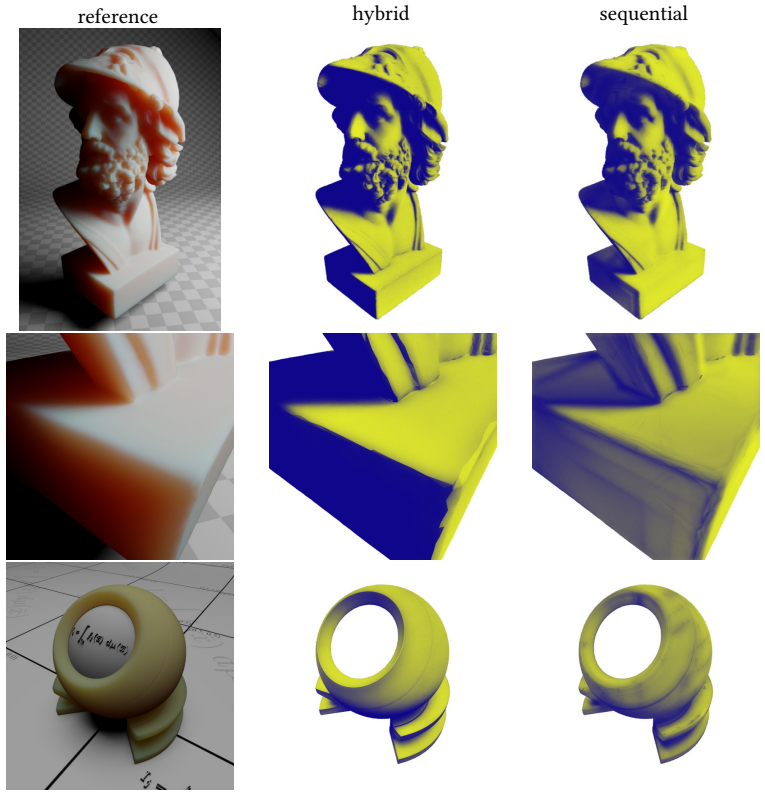

Fig. 2. Visualization of our hybrid and sequential shift during spatiotemporal reuse (averaged over multiple frames). *Hybrid*: Blue pixels indicate that the base path selects reconnection. Yellow pixels indicate delayed reconnection. *Sequential*: In each of the two spatial resampling passes, we record the fraction of shifted samples $T(X_i)$ that have a higher target function than the canonical sample X, i.e. $\hat{p}(T(X_i)) > \hat{p}(X)$. Blue pixels indicate that the pass with reconnection has a higher fraction and yields more improved samples compared to the pass with delayed reconnection. Yellow pixels indicate the opposite. This way, we can visualize, whether reconnection or delayed reconnection is implicitly chosen by the sequential shift. By comparing the visualizations of the hybrid and sequential shift, we can observe that the hybrid shift criterion often makes a binary decision per region while the sequential shift contains regions where both reconnection as well as delayed reconnection can be used to find improved samples.

when using the reconnection, delayed reconnection, and hybrid shift as well as the sequential shift with its rather unnoticeable bias in Fig. 3.

7 SPATIAL VERSUS SPATIOTEMPORAL REUSE

In Fig. 4, we compare spatial versus spatiotemporal reuse. While spatial reuse alone decreases noise little, reusing samples spatially and temporally reduces noise significantly in regions with visible SSS.

ACKNOWLEDGMENTS

Asian Dragon by XYZ RGB Inc. LTE Orb by Yasutoshi Mori. Ajax Bust by Jotero.

Fig. 3. Spatiotemporal reuse using our four shifts. All renderings eventually converge to the standard path-traced reference, but the color and temporal correlations remain for different periods. While reconnection, delayed reconnection, and hybrid shift do not introduce any bias, our implementation of the sequential shift is slightly biased due to performance reasons. However, this bias is rather unnoticeable.

Fig. 4. Scene with a highly scattering object and a single large light source. Comparison between standard path tracing and ReSTIR SSS with spatial and spatiotemporal reuse.

REFERENCES

Benedikt Bitterli, Wenzel Jakob, Jan Novák, and Wojciech Jarosz. 2017. Reversible Jump Metropolis Light Transport Using Inverse Mappings. *ACM Trans. Graph.* 37, 1, Article 1 (Oct. 2017). https://doi.org/10.1145/3132704

Benedikt Bitterli, Chris Wyman, Matt Pharr, Peter Shirley, Aaron Lefohn, and Wojciech Jarosz. 2020. Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. *ACM Trans. Graph.* 39, 4, Article 148 (Aug. 2020). https://doi.org/10.1145/3386569.3392481

Binh-Son Hua, Adrien Gruson, Victor Petitjean, Matthias Zwicker, Derek Nowrouzezahrai, Elmar Eisemann, and Toshiya Hachisuka. 2019. A Survey on Gradient-Domain Rendering. *Computer Graphics Forum* 38, 2 (2019), 455–472. https://doi.org/10.1111/cgf.13652

Csaba Kelemen, László Szirmay-Kalos, György Antal, and Ferenc Csonka. 2002. A Simple and Robust Mutation Strategy for the Metropolis Light Transport Algorithm. *Computer Graphics Forum* 21, 3 (2002), 531–540. https://doi.org/10.1111/1467-8659.t01-1-00703

Markus Kettunen, Marco Manzi, Miika Aittala, Jaakko Lehtinen, Frédo Durand, and Matthias Zwicker. 2015. Gradient-domain path tracing. *ACM Trans. Graph.* 34, 4, Article 123 (July 2015). https://doi.org/10.1145/2766997