ReSTIR Subsurface Scattering for Real-Time Path Tracing

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

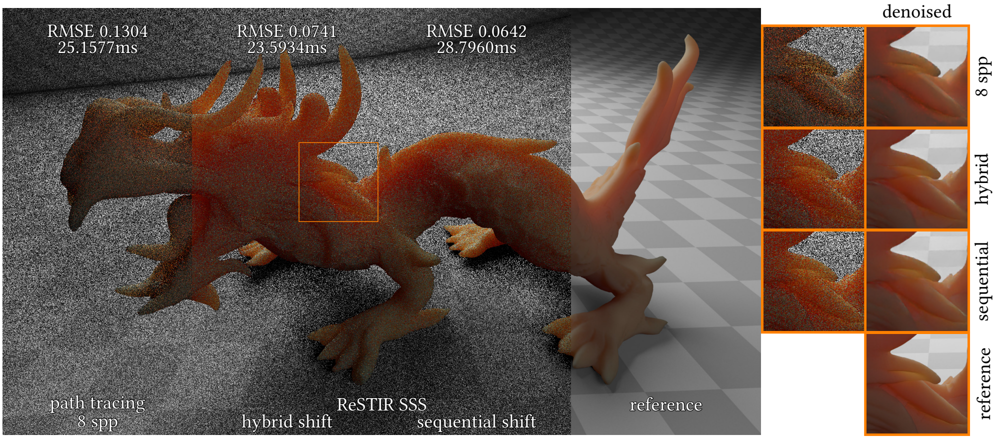


Fig. 1. A dragon with a material that exhibits noticeable subsurface scattering. We apply ReSTIR for subsurface scattering using our hybrid and sequential shift in real-time path tracing to significantly reduce noise and denoising artifacts in regions with visible scattered light.

Subsurface scattering is an important visual cue and in real-time rendering it is often approximated using screen-space algorithms. Path tracing with the diffusion approximation can easily overcome the limitations of these algorithms, but increases image noise. We improve its efficiency by applying reservoir-based spatio-temporal importance resampling (ReSTIR) to subsurface light transport paths. For this, we adopt BSSRDF importance sampling for generating candidates. Further, spatiotemporal reuse requires shifting paths between domains. We observe that different image regions benefit most from either reconnecting through the translucent object (*reconnection* shift), or one vertex later (*delayed reconnection* shift). We first introduce a local subsurface scattering specific criterion for a *hybrid* shift that deterministically selects one of the two shifts for a path. Due to the locality, it cannot always choose the most efficient shift, e.g. near shadow boundaries. Therefore, we additionally propose a novel *sequential* shift to combine multiple shift mappings: We execute subsequent resampling passes, each one using a different shift, which does not require to deterministically choose a shift for a path. Instead, resampling can pick the most successful shift implicitly. Our method achieves real-time performance and significantly reduces noise and denoising artifacts in regions with visible subsurface scattering compared to standard path tracing with equal render time.

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

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

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. *Proc. ACM Comput. Graph. Interact. Tech.* 7, 3, Article 1 (July 2024), 19 pages. https://doi.org/10.1145/3675372

1 INTRODUCTION

Subsurface scattering (SSS) is an important visual cue to capture in rendering which occurs in translucent materials such as marble, human skin, leaves, wax, milk, juices, or fruits. Light scattering through such objects results in a smoother appearance and slight color bleeding.

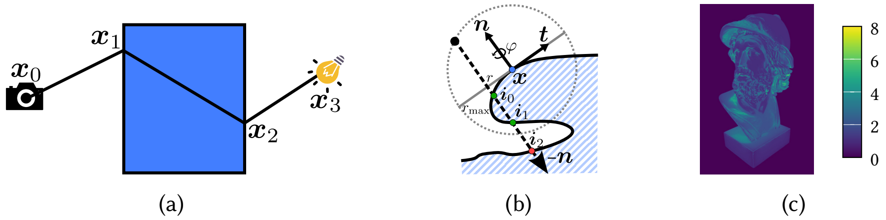
Simulating the entire subsurface light transport with a volumetric path tracer is often still infeasible in real-time. As an approximation, we can describe the transport using a bidirectional surface scattering reflectance distribution function (BSSRDF) that uses a diffusion profile [Jensen et al. 2001] to render visually plausible SSS. Instead of performing the actual random walks with scattering and absorption to compute the light transport, the diffusion profile approximates the fraction of light scattered through an object based on the distance and orientation between an entry and exit position. Current real-time screen space methods typically also use the diffusion approximation, however, computing SSS in a post-processing pass suffers from the limited screen space information. We avoid this limitation and compute diffusion using BSSRDF importance sampling (BSSRDF IS) [King et al. 2013] in a path tracer which inherently has access to world space information. However, the integration over entry points on the surface and computing SSS leads to high variance. Our goal is to decrease this noise significantly and make rendering SSS feasible in real-time path tracing.

In recent years, reservoir-based spatio-temporal importance resampling (ReSTIR) [Bitterli et al. 2020] helped making path tracing applicable in real-time rendering. ReSTIR reuses samples spatially across pixels and temporally between frames to achieve faster convergence for direct and indirect lighting as well as volumetric scattering in media such as smoke [Bitterli et al. 2020; Lin et al. 2022, 2021; Ouyang et al. 2021]. In this paper, we present *ReSTIR SSS* and show how to apply ReSTIR to rendering SSS using BSSRDF IS and to exploit the spatiotemporal reuse of paths that scatter much light and are difficult and expensive to find. Existing shift mappings, namely reconnection and random replay, that allow reusing paths in other domains, each work best only in certain regions of a translucent object. For surface reflection there exist roughness-based heuristics, however, for SSS we need to define a different combination of both shifts to achieve the best result.

In summary, our paper makes the following contributions:

- we show how ReSTIR can be applied to rendering SSS with BSSRDF IS (Section 3),
- we design a deterministic criterion for a *hybrid* shift to combine the advantages of existing shift mappings (Section 4.1),
- we derive a *sequential* shift that allows executing subsequent resampling passes with different shifts that removes the need of selecting only a single shift deterministically per sample (Sections 4.2 and 4.3).

In the next section, we describe how SSS using a diffusion profile is integrated into a path tracer (Section 2.1), recapitulate ReSTIR (Section 2.2), and review related work.



(a) (b) (c)

Fig. 2. (a) The direct illumination case for SSS. Light is scattered through the translucent object (blue) between the entry position $\boldsymbol{x}_2$ and the exit position $\boldsymbol{x}_1$. (b) To sample an entry position for a given exit position $\boldsymbol{x}$ ($= \boldsymbol{x}_1$), we use BSSRDF IS [King et al. 2013]. The diffusion profile is non-zero only inside a sphere (dotted circle) with radius r_{max}. The tangent $\boldsymbol{t}$ and bitangent at $\boldsymbol{x}$ span a disk (solid gray line) in this sphere. BSSRDF IS generates a point on this disk by importance sampling the diffusion profile to obtain a radius r, and sampling the angle φ uniformly. For projecting this point to the geometric surface, it traces a ray perpendicular to the disk. Intersections within the bounds of r_{max}, i.e. $\boldsymbol{i}_0, \boldsymbol{i}_1$ ($= \boldsymbol{x}_2$), are valid sampled points on the object, while intersections outside, i.e. $\boldsymbol{i}_2$, are ignored. (c) Average number of intersections with BSSRDF IS at primary hit.

2 BACKGROUND AND RELATED WORK

2.1 Subsurface Scattering (SSS)

Subsurface light transport in translucent materials is summarized by the bidirectional surface scattering reflectance distribution function (BSSRDF) $S(\boldsymbol{\omega}_o, \boldsymbol{x}_o, \boldsymbol{\omega}_i, \boldsymbol{x}_i)$. It describes the fraction of the light entering the object at $\boldsymbol{x}_i$ from direction $\boldsymbol{\omega}_i$ that exits at $\boldsymbol{x}_o$ in direction $\boldsymbol{\omega}_o$ [Jensen et al. 2001]. In contrast to using the bidirectional scattering distribution function (BSDF) in the surface rendering equation [Kajiya 1986], employing the BSSRDF is computationally more expensive and leads to higher variance in a Monte-Carlo estimator since it requires integrating over the surface A to account for all light that is scattered through the translucent object:

$$L(\boldsymbol{x}_o, \boldsymbol{\omega}_o) = \int_A \int_\Omega S(\boldsymbol{\omega}_o, \boldsymbol{x}_o, \boldsymbol{\omega}_i, \boldsymbol{x}_i) L(\boldsymbol{x}_i, \boldsymbol{\omega}_i) \cos\theta_i d\boldsymbol{\omega}_i d\boldsymbol{x}_i \tag{1}$$

The BSSRDF is usually simplified as the product of two directional Fresnel transmission terms F_t, the radially symmetric one-dimensional diffusion profile $R_{\boldsymbol{d}}$, and a constant C [Christensen 2015; Jensen et al. 2001]:

$$S(\boldsymbol{\omega}_o, \boldsymbol{x}_o, \boldsymbol{\omega}_i, \boldsymbol{x}_i) = CF_t(\boldsymbol{\omega}_o, \boldsymbol{x}_o) R_{\boldsymbol{d}}(||\boldsymbol{x}_o - \boldsymbol{x}_i||) F_t(\boldsymbol{\omega}_i, \boldsymbol{x}_i) \tag{2}$$

For the diffusion profile, we choose Burley's normalized diffusion [Christensen 2015], which approximates both single and multiple scattering well:

$$R_{\boldsymbol{d}}(r) = \boldsymbol{a} \frac{e^{-r/\boldsymbol{d}} + e^{-r/(3\boldsymbol{d})}}{8\pi \boldsymbol{d} r}, \tag{3}$$

where $\boldsymbol{a}$ is the surface albedo, r the distance between the entry and exit point, and $\boldsymbol{d} = \boldsymbol{l}/s$ a scaled version of the wavelength-dependent mean free path $\boldsymbol{l}$ with a scaling factor s. For importance sampling, the PDF and CDF of $R_{\boldsymbol{d}}$ are given by [Christensen 2015; Golubev 2019]

$$p(r) = \frac{e^{-r/d} + e^{-r/(3d)}}{8\pi d r} \quad \text{and} \quad P(r) = 1 - 0.25 e^{-r/d} - 0.75 e^{-r/(3d)}. \tag{4}$$

The BSSRDF and the diffusion profile are used during path tracing as follows (Fig. 2a): When a ray hits a translucent object, we sample an entry point on the surface of the object. For this, we employ BSSRDF importance sampling (BSSRDF IS) [King et al. 2013], which first samples a point on a disk and then uses ray tracing to project it onto the surface (see Fig. 2b). Since the mean free path $\boldsymbol{d}$ is wavelength-dependent, one channel of $\boldsymbol{d}$ is chosen uniformly to sample the radius r and to determine the maximum distance $r_{\max}$. Tracing a ray along the surface normal works well for planar geometry, however, points on curved geometry have a low or zero probability of being found this way. Hence, the tangent and bitangent can be chosen as the projection axis for the ray direction as well. One of the generated intersections is chosen as the entry position path vertex and the path tracer continues as usual. The throughput of the path between the entry and exit position is given by Eq. (2) and the PDF by Eq. (4) times the probability of choosing the channel and the axis. To reduce variance, multiple importance sampling (MIS) for the three possible axes and the three channels is applied [King et al. 2013; Pharr et al. 2016].

For our purposes, we restrict SSS to be applied at the primary hit, i.e. exit position $\boldsymbol{x}_1$ and entry position $\boldsymbol{x}_2$. When the path tracer happens to hit a translucent object after the primary hit, we revert to a simple surface reflection using a BSDF, e.g. a BSDF with integrated SSS to mimic some of the aspects of actual SSS [Burley 2015]. This increases performance and prohibits code execution divergence on the GPU. Also, we model only scattering between vertices of the same object.

2.1.1 Related Work. The formulation of SSS as a diffusion process has been introduced by Jensen et al. [2001]. Diffusion profiles are either explicitly physically-based [D'Eon and Irving 2011; Donner and Jensen 2005; Habel et al. 2013], or designed to match Monte Carlo references well [Christensen 2015]. Integration of diffusion in path tracing requires importance sampling of the BSSRDF [Christensen et al. 2012; King et al. 2013; Walter et al. 2012] and provides inherent access to world space information. Irradiance information can be cached in a point cloud and filtered during rendering [Jensen and Buhler 2002]. The effects of SSS can also be approximated as a post-processing pass by convolving incoming irradiance with the diffusion profile in either texture space [d'Eon et al. 2007; Green 2004] or screen space [Golubev 2018; Jimenez et al. 2009, 2015]. Both approaches suffer from limited information. As the intensity of SSS effects varies across an object, adaptive sampling [Xie et al. 2020; Zwicker et al. 2015] redistributes the sampling budget to use more samples for complex pixels. We refer to Liang et al. [2024] for a detailed overview of diffusion profiles and rendering with BSSRDFs. SSS can also be approximated using light propagation volumes [Børlum et al. 2011]. Instead of approximations, volumetric path tracing renders SSS by simulating the full volumetric light transport [Burley 2015; Novák et al. 2018]. Long random walks that are prone to high variance can be guided toward the closest surface or nearest light source [Křivánek and d'Eon 2014; Meng et al. 2016]. Still, many long paths have to be traced through the object to reduce variance sufficiently, which is typically too expensive for real-time rendering.

2.2 Reservoir-Based Spatio-Temporal Importance Resampling (ReSTIR)

2.2.1 Generalized Resampled Importance Sampling (GRIS). GRIS [Lin et al. 2022] provides an importance sampling estimator for the integral

$$I = \int_{\Omega} f(x)dx, \tag{5}$$

for any real-valued function $f : \Omega \rightarrow \mathbb{R}$. Let $\hat{p}$ be a desired target function, i.e. $\hat{p} \approx f$, with $\operatorname{supp}(f) \subseteq \operatorname{supp}(\hat{p})$ that can be evaluated but has no practical sampling algorithm. As inputs, GRIS takes $M \geq 1$ samples $X_1, \ldots, X_M$ from domains $\Omega_1, \ldots, \Omega_M$ with possibly intractable PDFs $p_1(X_1), \ldots, p_M(X_M)$ but known unbiased contribution weights (UCWs) $W_{X_1}, \ldots, W_{X_M}$ that unbiasedly estimate $\mathbb{E}\left[W_{X_i}|X_i\right] = 1/p_i(X_i)$. It then randomly selects one of the samples, shifted into the target integration domain $Y = Y_s = T_s(X_s) \in \Omega$, proportional to assigned resampling weights w_i. As $M \rightarrow \infty$, the distribution of Y approaches $\hat{p}$, i.e. $Y \sim \hat{p}/\int_{\Omega}\hat{p}$. The one sample GRIS estimator for

Algorithm 1: WRS

```
struct Reservoir
    $Y \leftarrow \emptyset$
    $W_Y \leftarrow 0$
    $w_{\text{sum}} \leftarrow 0$
    $c \leftarrow 0$

    function update($X_i$, $w_i$, $c_i$)
        $w_{\text{sum}} \leftarrow w_{\text{sum}} + w_i$
        $c \leftarrow c + c_i$
        if rand() $\leq w_i / w_{sum}$ then
            $Y \leftarrow X_i$
```

Algorithm 2: GRIS

```
Input: Samples $X_1 \dots X_M$ in domains $\Omega_1 \dots \Omega_M$, confidence weights
        $c_1 \dots c_M$, and unbiased contribution weights $W_{X_1} \dots W_{X_M}$
Output: Combined reservoir with sample $Y$ in target domain $\Omega$

Reservoir $r$

for $i = 1 \dots M$ do
    $Y_i \leftarrow T_i(X_i)$ // shift sample from $\Omega_i$ into target domain $\Omega$
    $w_i \leftarrow m_i(Y_i)\hat{p}(Y_i)W_{X_i} \left|T_i^{'}(X_i)\right|$
    $r$.update($Y_i$, $w_i$, $c_i$)

if $r.Y \neq \emptyset$ then
    $r.W_Y \leftarrow \frac{1}{\hat{p}(r.Y)} r.w_{\text{sum}}$

$r.c \leftarrow$ min($r.c, c_{cap}$)

return $r$
```

I (Eq. (5)) is

$$\hat{I} = f(Y)\frac{1}{\hat{p}(Y)}\sum_{i=1}^{M} \underbrace{m_i(T_i(X_i))\hat{p}(T_i(X_i))W_{X_i}\left|T_i^{'}(X_i)\right|}_{=:w_i} = f(Y)\frac{1}{\hat{p}(Y)}\sum_{i=1}^{M} w_i = f(Y)W_Y. \quad (6)$$

Since the X_i are allowed to come from different domains Ω_i, they need to be shifted into the target domain Ω using a shift mapping T_i that shifts samples from domain Ω_i to Ω. When applying shift mappings, the densities p_i of the samples are changed. As a consequence, the UCWs change as well: $W_{Y_i} = W_{X_i}|T_i^{'}(X_i)|$. In general, the individual $T_i(X_i)$ do not cover $\hat{p}$, i.e. $\text{supp}(\hat{p}) \not\subseteq \text{supp}(T_i(X_i))$, such that proper resampling MIS weights $m_i(T_i(X_i))$ become necessary. This way, to ensure unbiasedness, it is only required that the union of all $T_i(X_i)$ covers $\hat{p}$, which can be guaranteed by adding one canonical sample $X_1 \in \Omega_1 = \Omega$. The resampling MIS weights can be calculated with the generalized balance heuristic [Lin et al. 2022] using the target function $\hat{p}$ as a proxy for the (possibly) intractable PDF p

$$m_i(T_i(X_i)) = \frac{\hat{p}_{\leftarrow i}(T_i(X_i))}{\sum_{j=1}^{M}\hat{p}_{\leftarrow j}(T_i(X_i))}, \text{ where } \hat{p}_{\leftarrow j}(T_i(X_i)) = \hat{p}_j\left(T_j^{-1}(T_i(X_i))\right)\left|T_j^{-1'}(T_i(X_i))\right|, \quad (7)$$

where $\hat{p}_j$ is the target function evaluated in domain Ω_j. This requires $O(M^2)$ operations. Alternatively, pairwise MIS [Bitterli 2021; Lin et al. 2022] can be used to calculate a pairwise generalized balance heuristic between the shifted sample $T_i(X_i)$ and the canonical sample $T_1(X_1) = X_1$ in $O(M)$.

2.2.2 Reservoir-Based Spatio-Temporal Importance Resampling (ReSTIR). ReSTIR [Bitterli et al. 2020; Lin et al. 2022] transforms GRIS into a streaming algorithm by using weighted reservoir sampling (WRS) [Chao 1982]. While processing a stream of samples X_i, a reservoir only stores the currently selected sample Y, the current sum of weights w_{sum}, and a confidence weight c that encodes the effective sample count, i.e. the number of samples processed. After processing all samples, the UCW W_Y can be calculated using the information stored in the reservoir. Algorithms 1 and 2 show WRS and streaming GRIS. Additionally, ReSTIR chains GRIS passes for spatiotemporal reuse. GRIS passes can be chained by using a set of reservoirs $r_1, \dots, r_M$ with their respective selected samples $r_1.Y, \dots, r_M.Y$ and UCWs $r_1.W_Y, \dots, r_M.W_Y$ as inputs. The resulting reservoir combines the inputs and has an increased effective sample count of $c = r_1.c + \cdots + r_M.c$. The different effective sample counts of reservoirs should be respected in the resampling MIS weights during resampling, which leads to the generalized balance heuristic with confidence weights:

$$m_i(T_i(X_i)) = \frac{c_i \hat{p}_{\leftarrow i}(T_i(X_i))}{\sum_{j=1}^{M} c_j \hat{p}_{\leftarrow j}(T_i(X_i))} \tag{8}$$

In the context of path tracing, the goal is to integrate the measurement contribution function f. A given pixel and time (frame) defines the domain Ω, such that f and $\hat{p}$ depend on Ω. ReSTIR allows reusing light paths by sharing across pixels and between frames. This amortizes costs and increases the effective sample count per pixel. Therefore, each pixel stores a reservoir. In each frame, ReSTIR commonly performs the following tasks for each pixel:

1. Candidate generation: Generate M candidate samples (paths) $X_1, \ldots, X_M$ using (potentially suboptimal) importance sampling schemes with PDFs $p_i(X_i)$ and computable UCWs $W_{X_i} = 1/p_i(X_i)$. Select one initial sample using GRIS and store it in the reservoir.
2. Temporal reuse: Find the matching pixel in the previous frame (e.g. using motion vectors) and combine the reservoir of the last frame with the current reservoir using GRIS.
3. Spatial reuse: Randomly select several neighbor pixels and combine their reservoirs with the current reservoir using GRIS.
4. Shade: Shade the current pixel using the selected sample stored in the reservoir: $\hat{I} = f(Y)W_Y$.

Since the samples during candidate generation originate from the same source domain, the identity shift $T_i(X_i) = X_i$ with $\left|T_i^{'}(X_i)\right| = 1$ is used. In contrast, spatiotemporal reuse shares samples between different pixels and frames and therefore requires proper shift mappings between domains. To ensure unbiased convergence, the confidence weights have to be capped. Otherwise, temporal reuse causes $c \longrightarrow \infty$, new initial samples will have little influence on the selected sample, and correlation artifacts will occur.

2.2.3 Related Work. GRIS [Lin et al. 2022] generalizes resampled importance sampling (RIS) [Talbot et al. 2005] to support samples from different source domains. After the initial introduction of ReSTIR [Bitterli et al. 2020] for direct illumination only, several papers have been published to extend ReSTIR to longer paths for global illumination [Lin et al. 2022, 2021; Ouyang et al. 2021]. Recent work extends ReSTIR to conditional probability spaces [Kettunen et al. 2023] and interleaves ReSTIR with Markov Chain Monte Carlo mutations for decorrelation [Sawhney et al. 2024]. Shift mappings originate from gradient-domain rendering [Kettunen et al. 2015; Lehtinen et al. 2013], including reconnection [Lehtinen et al. 2013], random replay [Hua et al. 2019; Kettunen et al. 2015; Manzi et al. 2016], manifold exploration [Lehtinen et al. 2013], and half-vector copying [Kettunen et al. 2015]. Generally, a single shift mapping does not work well for all cases, which requires the combination of shifts [Gruson et al. 2018; Hua et al. 2017; Kettunen et al. 2015].

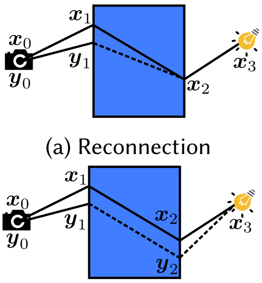
3 RESTIR FOR SUBSURFACE SCATTERING (RESTIR SSS)

In this section, we explain our application of ReSTIR to objects that are noticeably translucent and affected by SSS. We focus on the aspects that are new or different in our case compared to the existing ReSTIR and GRIS theory. For clarity, our explanation and subsequent evaluation focus on the direct illumination case for SSS (Fig. 2a), where we have light paths $\bar{x} = [\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3]$. The camera vertex $\boldsymbol{x}_0$ and the primary hit $\boldsymbol{x}_1$ on the object define the domain Ω. The measurement contribution function $f(\bar{x})$ describes the transported light from the light source at $\boldsymbol{x}_3$ that is scattered through the object $\boldsymbol{x}_2 \longrightarrow \boldsymbol{x}_1$ towards $\boldsymbol{x}_0$. We set $\hat{p}(x) = \hat{p}_\Omega(x) = f(\bar{x})$.

We show how to adopt BSSRDF IS in the ReSTIR framework in Section 3.1 to generate candidate paths in the first step of ReSTIR SSS. The subsequent spatiotemporal reuse then shares paths between pixels and frames by shifting them between domains. We adapt two commonly used

Algorithm 3: Candidate sample

```
struct Sample
    Vertex $x_2$ // vertex on translucent object
    Vertex $x_3$ // vertex on light source
    uint axis // index for projection axis
    uint channel // index for mean free path channel
    float $u_{\text{radius}}$ // random number for radius
    float $u_{\text{angle}}$ // random number for angle
    uint intId // intersection index
    float $J$ // cached part of Jacobian
```



(a) Reconnection

(b) Delayed reconnection

Fig. 3. Our ReSTIR SSS reconnection shift (a) reconnects the offset path $\boldsymbol{y}$ to the base path $\boldsymbol{x}$ through the object. Our ReSTIR SSS delayed reconnection shift (b) uses random replay to generate an entry point $\boldsymbol{y}_2$ similar to the base path before reconnecting to the base path one vertex later.

shift mappings for ReSTIR SSS in Section 3.2. Since they do not always work well individually, we introduce techniques to combine them in Section 4.

3.1 Candidate Generation

Given $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$, BSSRDF IS and light source sampling are used to obtain $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$ to generate a complete candidate path. Candidate generation generates M candidate paths X_i and uses GRIS to pick one initial path Y. During BSSRDF IS, multiple intersections are often generated for each sampled disk point with the object, as shown in Fig. 2c. Normally, one of the intersections is randomly picked which introduces a PMF of $1/N_i$, where N_i is the number of found intersections. This leads to UCWs $W_{X_i} = \frac{1}{p(X_i)\frac{1}{N_i}}$ and

$$W_Y = \frac{1}{\hat{p}(Y)} \sum_{i=1}^{M} \frac{1}{M} \hat{p}(X_i) \frac{1}{p(X_i)\frac{1}{N_i}}. \tag{9}$$

As all intersections need to be listed during the projection via ray tracing to determine N_i anyway, we propose to increase the effective sample count at a low cost by taking all found intersections into account. We observe that the back part of Eq. (9) is a one sample estimator of the weighted sum over all intersections, i.e. $\sum_{j=1}^{N_i} \frac{\hat{p}(X_{ij})}{p(X_{ij})} \approx \frac{\hat{p}(X_i)}{p(X_i)} \frac{1}{\frac{1}{N_i}}$. Therefore, we rewrite Eq. (9) as

$$W_Y = \frac{1}{\hat{p}(Y)} \sum_{i=1}^{M} \sum_{j=1}^{N_i} \frac{1}{M} \hat{p}(X_{ij}) \frac{1}{p(X_{ij})}, \tag{10}$$

where $W_{X_{ij}} = 1/p(X_{ij})$. This has the advantage that the effective sample count is increased to $\sum_{i=1}^{M} N_i \geq M$. Since it is not necessary to know the number of intersections N_i in Eq. (10), the individual intersections can still be generated and added to the reservoir in a streaming fashion. In both cases, the stored confidence weight in the reservoir after candidate generation is $c = M$.

We show the structure of a candidate path in Algorithm 3. In addition to the sampled path, we require information from BSSRDF IS, which we later use for shifting paths (Section 3.2). The PDF p of the candidate path consists of the PDF for BSSRDF IS $p_{\text{SSS},\boldsymbol{x}_1}(\boldsymbol{x}_2)$ times the PDF for sampling the light source. We cache $p_{\text{SSS},\boldsymbol{x}_1}(\boldsymbol{x}_2)$ as J in the candidate sample. J is part of the shift Jacobian.

3.2 Shift Mappings

Spatiotemporal reuse requires shifting paths between domains. In this section, we describe the application of two common shift mappings T to map a base path $\bar{x} = [\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3]$ to an offset path $T(\bar{x}) = \bar{\boldsymbol{y}} = [\boldsymbol{y}_0, \boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_3]$ in the context of ReSTIR SSS and state their respective Jacobians $\left|T^{'}(\bar{x})\right|$ that account for the change in the densities of the paths (cf. Section 2.2.1 and Algorithm 2). A more general introduction to the building blocks of the two shifts independent from our subsurface case and a derivation of the Jacobians can be found in the supplemental material.

3.2.1 Reconnection Shift. The reconnection shift [Kettunen et al. 2015] reconnects the offset path directly to the first free vertex of the base path ($\boldsymbol{x}_0$ and $\boldsymbol{x}_1$ are always fixed for a given pixel and time), i.e. $T([\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3]) = [\boldsymbol{y}_0, \boldsymbol{y}_1, \boldsymbol{x}_2, \boldsymbol{x}_3]$. This means, we reconnect through the translucent object (Fig. 3a), which can help to reuse a long scattering path that carries light from an illuminated region into a shadowed region. BSSRDF IS is performed in area measure, hence the Jacobian of the shift is $\left|T^{'}(\bar{x})\right| = 1$ [Hua et al. 2019]. Reconnecting through the object does not require tracing a ray to stay unbiased, since the visibility inside the object is solely determined by the diffusion profile. This makes the shift fast. In contrast, for surface reflection, reconnection can lead to occlusion between $\boldsymbol{y}_1 \leftrightarrow \boldsymbol{x}_2$. This visibility has to be tested to avoid biasedness [Ouyang et al. 2021].

3.2.2 Delayed Reconnection Shift. Instead of reconnecting directly to the base path at the first free vertex, reconnection can be delayed to one vertex later (Fig. 3b), i.e. $T([\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3]) = [\boldsymbol{y}_0, \boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{x}_3]$. Delayed reconnection shift first uses random replay [Hua et al. 2019] to sample a vertex $\boldsymbol{y}_2$ on the object using the same random numbers as the base path for BSSRDF IS. This generally leads to a similar but shifted version of the sample $\boldsymbol{x}_2$, which can be advantageous for re-creating short scattering paths. Random replay is followed by reconnection to reconnect the offset path to the first vertex after the object. The Jacobian for the shift mapping is the product of the Jacobians of the random replay and the reconnection step. The Jacobian for this is [Hua et al. 2019]

$$|T'(\bar{x})| = \frac{p_{\mathrm{SSS},\boldsymbol{x}_1}(\boldsymbol{x}_2)}{p_{\mathrm{SSS},\boldsymbol{y}_1}(\boldsymbol{y}_2)} \left|\frac{\cos\theta_2^{\boldsymbol{y}}}{\cos\theta_2^{\boldsymbol{x}}}\right| \frac{||\boldsymbol{x}_3 - \boldsymbol{x}_2||^2}{||\boldsymbol{x}_3 - \boldsymbol{y}_2||^2} \quad \text{or} \quad |T'(\bar{x})| = \frac{p_{\mathrm{SSS},\boldsymbol{x}_1}(\boldsymbol{x}_2)}{p_{\mathrm{SSS},\boldsymbol{y}_1}(\boldsymbol{y}_2)}, \tag{11}$$

depending on whether the light source $\boldsymbol{x}_3$ is sampled in solid angle or area measure, respectively. $\theta_2^{\boldsymbol{x}}$ denotes the angle between the normalized direction from $\boldsymbol{x}_3$ to $\boldsymbol{x}_2$ and the normal at $\boldsymbol{x}_3$. We cache $p_{\mathrm{SSS},\boldsymbol{x}_1}(\boldsymbol{x}_2)$ (part of the Jacobian) during candidate generation (cf. Section 3.1 and Algorithm 3) to avoid re-calculation. For the next execution of delayed reconnection, we update the cached part of the Jacobian of the offset path to $p_{\mathrm{SSS},\boldsymbol{y}_1}(\boldsymbol{x}_2)$. To ensure bijectivity, a sample also stores the index j of the intersection as an identifier (cf. Section 3.1 and Algorithm 3). The offset path chooses the same j-th intersection found by BSSRDF IS with the copied random numbers. If the number of intersections is less than j, the shift fails. This idea of reusing the intersection index is similar to ReSTIR PT's [Lin et al. 2022] extension of the path space with a lobe index. Compared to the simpler reconnection shift (Section 3.2.1), delayed reconnection requires tracing a ray for the projection during BSSRDF IS. If the target function $\hat{p}$ contains visibility between $\boldsymbol{x}_2 \leftrightarrow \boldsymbol{x}_3$, another visibility check is necessary to ensure that $\boldsymbol{y}_2 \leftrightarrow \boldsymbol{x}_3$ is not occluded.

4 COMBINING SHIFTS

Depending on the situation, reconnection or delayed reconnection yields better results (cf. Section 6): Reconnection works better in transitions between illuminated and shadowed regions, where the effects of SSS are more dominant. Delayed reconnection works better for surfaces that are directly

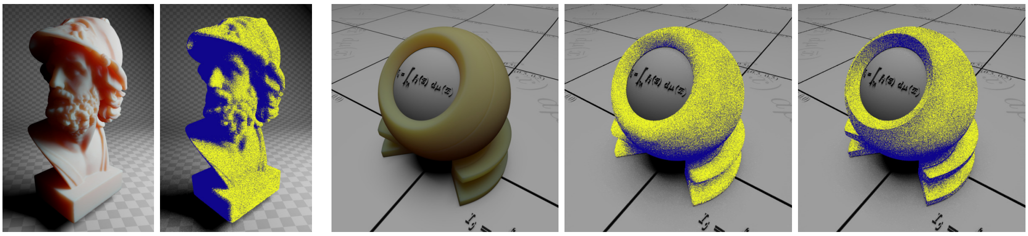


Fig. 4. Visualization of our hybrid shift. Blue pixels indicate that the base path selects reconnection. Yellow pixels indicate delayed reconnection. Our criterion separates the illuminated and the shadowed regions. The second image from the right shows the selection of the shifts based on the distance only, without normal orientation. Adding the orientation helps with finding more regions with visible SSS (right).

illuminated by a light source, where SSS generally only leads to a smoothing effect and distances between the entry and exit point are rather small.

In this section, we therefore investigate how both shifts can be combined. First, we design a hybrid shift (Section 4.1) by introducing a deterministic SSS specific criterion to decide which shift should be used for a specific sample. After that, we introduce a novel approach to allow sequentially executed resampling passes to use different shifts such that there is no need to deterministically decide on a single shift for a specific sample (Section 4.2). We then apply this theory to our SSS case (Section 4.3).

4.1 Hybrid Shift for ReSTIR SSS

ReSTIR PT [Lin et al. 2022] designs a hybrid shift that delays reconnection until both vertices $\boldsymbol{y}_k$ and $\boldsymbol{x}_{k+1}$ of the offset path (generated up to $\boldsymbol{y}_k$ with random replay) and the base path are diffuse. This is determined by a roughness-based criterion. The first vertex $\boldsymbol{x}_{k+1}$ on the base path that satisfies the criterion for $\boldsymbol{x}_k$ and $\boldsymbol{x}_{k+1}$ is used for reconnection. To ensure a deterministic and bijective shift, the offset path vertices $\boldsymbol{y}_k$ and $\boldsymbol{x}_{k+1}$ have to satisfy the criterion as well.

Like the hybrid shift in ReSTIR PT, our hybrid shift also deterministically decides whether $\boldsymbol{x}_2$ or $\boldsymbol{x}_3$ is used for reconnection, i.e. whether reconnection or delayed reconnection shift is used for a specific sample. We design our criterion based on the observations from before (Section 4) to distinguish the two regions, where either one of the shifts is superior. Typically, for regions where the subsurface contribution dominates direct light, the distance between $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ is rather large. It is also possible that the distance is small but the orientation of the surface normals $\boldsymbol{n}_{\boldsymbol{x}_1}$ and $\boldsymbol{n}_{\boldsymbol{x}_2}$ is significantly different, which indicates a potentially shadowed $\boldsymbol{x}_1$ but unshadowed $\boldsymbol{x}_2$. We choose the reconnection shift using thresholds κ_{distance} and $\kappa_{\text{orientation}}$ (cf. Section 5.1) if

$$||\boldsymbol{x}_2 - \boldsymbol{x}_1|| \geq \kappa_{\text{distance}} \quad \text{or} \quad \boldsymbol{n}_{\boldsymbol{x}_2} \cdot \boldsymbol{n}_{\boldsymbol{x}_1} \leq \kappa_{\text{orientation}}.$$

We visualize which shift the base path selects based on the introduced criterion in Fig. 4.

4.2 Sequential Shift

Hybrid shift sometimes fails if the base and offset path cannot agree on a vertex for reconnection. Additionally, the criterion might not always be a perfect fit for all scenarios, e.g. directly at shadow boundaries. As a consequence, we now investigate the idea of using different shift mappings in sequentially executed resampling passes. For example, a first resampling pass can use reconnection shift for all samples. Then, a second pass gets the result of the first pass as its input and uses delayed reconnection shift for all samples. This way, both pixels where reconnection performs better as well as pixels where delayed reconnection works better have a chance to find more useful

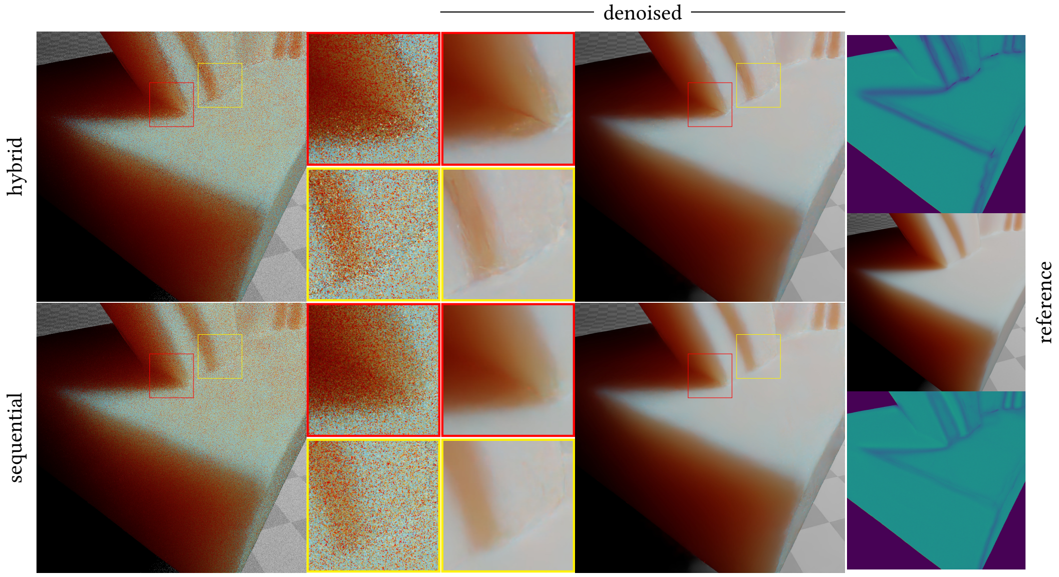


Fig. 5. Comparison between hybrid and sequential shift. Sequential shift leads to a more uniform sampling at shadow and geometry edges, where the criterion of the hybrid shifts is suboptimal. The visualization in the right column shows the fraction of shifted samples $T(X_i)$ that have a higher target function than the canonical sample X, i.e. $\hat{p}(T(X_i)) > \hat{p}(X)$, averaged over multiple frames. Brighter regions indicate a higher fraction. Sequential shift yields more improved samples.

samples. We will refer to this shift as *sequential shift*. A visual comparison between the hybrid and sequential shift can be found in Fig. 5. The alternating use of both shift mappings does however require non-trivial changes to the shifts, due to their difference in domain.

Our shift mappings operate both in path space $\mathcal{P}$ (reconnection) and primary sample space (PSS) $\mathcal{U}$ (random replay used in delayed reconnection). For a candidate path, both coordinates $\bar{\boldsymbol{x}} \in \mathcal{P}$ and $\bar{\boldsymbol{u}} \in \mathcal{U}$ are available, since our sampling scheme is a function that maps random numbers $\bar{\boldsymbol{u}}$ to $\bar{\boldsymbol{x}}$. After resampling and changing $\bar{\boldsymbol{x}}$ to $\bar{\boldsymbol{y}}$, we require *inversion* to obtain matching random numbers $\bar{\boldsymbol{v}}$, in order to later apply random replay using $\bar{\boldsymbol{y}}$ as a base path. Without further modifications, this is not always possible, since discrete sampling decisions lead to a mismatch in dimensionality between $\mathcal{P}$ and $\mathcal{U}$, i.e. there is no bijection. In particular, this problem occurs in our case when sampling a mixture distribution, i.e. when we choose an axis and channel in BSSRDF IS.

Previous work [Bitterli et al. 2017; Otsu et al. 2017; Pantaleoni 2017] solves this problem in the context of Metropolis light transport using *probabilistic inversion*. They extend $\mathcal{P}$ with extra dimensions $\Gamma = [0,1]^m$ such that there exists a bijection between $\mathcal{P} \times \Gamma$ and $\mathcal{U}$. Intuitively, $\bar{\boldsymbol{\gamma}} \in \Gamma$ decides to which of many possible $\bar{\boldsymbol{u}}$ we map from $\bar{\boldsymbol{x}}$. They further show that since the value of $\bar{\boldsymbol{\gamma}}$ does not affect the measurement integral, we can sample it randomly before inversion.

Probabilistic inversion has been applied in the context of ReSTIR to obtain random numbers to apply Metropolis-Hastings mutations in primary sample space (PSS) in a separate pass [Sawhney et al. 2024]. In the following, we explain how probabilistic inversion fits directly into the ReSTIR framework. We maintain the uniform distribution of $\bar{\boldsymbol{\gamma}}$, therefore changing $\bar{\boldsymbol{\gamma}}$ does not require a modification of the UCW (see Appendix A). This allows us to employ probabilistic inversion for ReSTIR in shift mappings, i.e. we do not need to store a specific value of $\bar{\boldsymbol{\gamma}}$ and instead sample it lazily when needed. To keep shift mappings deterministic, we sample a uniform random value for

Algorithm 4: Probabilistic inverse BSSRDF IS

```
Input: $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ and random number $\gamma$
Output: Axis, channel and random numbers $u_{\text{radius}}$, $u_{\text{angle}}$ to sample $\boldsymbol{x}_2$ given $\boldsymbol{x}_1$
// sample using $\gamma$, proportionally to selection probabilities $p_{\text{axis}}$, $p_{\text{channel}}$ and PDFs $p_{\text{SSS},\boldsymbol{x}_1,\cdot,\cdot}$
$axis, channel \leftarrow a, c \sim p(a, c \mid \boldsymbol{x}_2) = \frac{p_{\text{axis}}(a) p_{\text{channel}}(c) p_{\text{SSS},\boldsymbol{x}_1,a,c}(\boldsymbol{x}_2)}{\sum_{a'=1}^{3} \sum_{c'=1}^{3} p_{\text{axis}}(a') p_{\text{channel}}(c') p_{\text{SSS},\boldsymbol{x}_1,a',c'}(\boldsymbol{x}_2)}$
$\boldsymbol{p}_x, \boldsymbol{p}_y, \boldsymbol{p}_z \leftarrow$ projectionAxis($axis$) // orthonormal basis with $\boldsymbol{p}_z$ along the projection axis
$d \leftarrow \boldsymbol{d}[channel]$
$\boldsymbol{l} \leftarrow \boldsymbol{x}_2 - \boldsymbol{x}_1$
$l_x \leftarrow \boldsymbol{l} \cdot \boldsymbol{p}_x, l_y \leftarrow \boldsymbol{l} \cdot \boldsymbol{p}_y$ // projection on the disk
$r \leftarrow \sqrt{l_x^2 + l_y^2}$
$u_{radius} \leftarrow 1 - 0.25e^{-r/d} - 0.75e^{-r/(3d)}$ // Eq. (4), invert radius
$\varphi \leftarrow$ atan2$(l_y/r, l_x/r)$ // = atan2$(\sin\varphi, \cos\varphi) \in [0, 2\pi]$
$u_{angle} \leftarrow \varphi/(2\pi)$ // invert angle
return $axis$, $channel$, $u_{radius}$, $u_{angle}$
```

$\bar{\boldsymbol{y}} \in [0, 1]^m$ of the base path, right before applying the shift. In practice, the shift fails if the shifted result cannot be inverted due to zero sampling probability. The symmetry of the reconnection shift is guaranteed during a resampling pass, i.e. $T_{j\to i}(T_{i\to j}(\bar{x})) = \bar{x}$, since we know that $\bar{x}$ did not have zero probability of being sampled in the first place, so inversion is possible. Of course, there is always a chance that the inversion cannot be computed due to numerical errors and the shift fails.

4.3 Sequential Shift for ReSTIR SSS

We now apply the sequential shift (Section 4.2) to ReSTIR SSS. BSSRDF IS is an example for sampling a mixture distribution, where multiple combinations of projection axis index and channel of the mean free path can produce the same sample. Hence, $\bar{\boldsymbol{y}} \in \Gamma$ determines the axis and channel. We will now refer to a specific combination of axis index and channel as a *technique*. During reconnection shift, we reconnect the offset path vertex $\boldsymbol{y}_1$ with the base path vertex $\boldsymbol{x}_2$ and need to obtain matching random numbers that are cached in the reservoir for a subsequent random replay pass. We draw $\gamma \in [0, 1]$ to select a possible technique based on the likelihood that it generated the sample. Once the technique is fixed, the radius and the angle used during BSSRDF IS can be calculated and inverted to obtain the random numbers u_{radius} and u_{angle}. Since $\bar{\boldsymbol{y}}$ maps to the technique independently from the geometry and material, there is no need to invert $\bar{\boldsymbol{y}}$ or the technique to obtain random numbers in $\mathcal{U}$. Instead, we can store the technique in the reservoir and use it directly for random replay when constructing an offset path. Algorithm 4 shows the complete probabilistic inverse BSSRDF IS procedure.

For random replay, the intersection index is required (cf. Section 3.2.2). Like the random numbers, this index can change when reconnecting. This means, we have to re-calculate it by tracing a ray from $\boldsymbol{y}_1$ to $\boldsymbol{x}_2$. Since this operation is expensive, we want to minimize the number of re-calculations as much as possible. In theory, we only have to determine the new intersection index once for the final sample selected by GRIS after the resampling pass with reconnection shift finished, since this information is not required to calculate the resampling (MIS) weights and since the intersection can always be found when the inversion does not fail. In practice, however, tracing the ray from $\boldsymbol{y}_1$ to $\boldsymbol{x}_2$ introduces slight numerical errors, such that no intersection can be found with $\boldsymbol{x}_2$ or in a reasonably small neighborhood after all. This, for example, can happen if the point is on the edge of a triangle which itself is at the edge of the object and the ray just misses the edge. To stay unbiased, this requires determining the intersection index during inversion and failing the shift directly when the intersection cannot be found (cf. Section 4.2). As a practical solution for avoiding this overhead, we set the intersection index to an unknown state when the intersection cannot be found for the selected sample after the resampling pass. This tells random replay in a subsequent pass to select a random intersection instead. This results in a slight, rather unnoticeable bias, which we assume is acceptable for real-time applications given the performance gain by tracing fewer rays.

5 IMPLEMENTATION DETAILS

We implemented ReSTIR SSS with all introduced shifts in Vulkan and provide its source code[1].

5.1 Parameters

For real-time rendering, we use $M = 1$ for candidate generation and 4 neighbors for spatial reuse. For the sequential shift, the temporal pass uses reconnection, and we execute two consecutive spatial passes that use 2 neighbor samples for the first pass with reconnection and 2 for the following pass with delayed reconnection. Spatial reuse employs the defensive form of pairwise MIS [Lin et al. 2022] to calculate resampling MIS weights in a streaming fashion. As temporal reuse only combines the reservoirs of the last and the current frame, we choose the generalized balance heuristic (Eq. (7)) for the resampling MIS weights. We cap the confidence weight of the reservoir of the last frame to a maximum of 20× the confidence weight of the reservoir of the current frame (Algorithm 2), as we mostly rely on temporal reuse for real-time rendering. While this leads to unbiased results when accumulating several frames, we encourage applicants to decrease this multiplier to avoid long-lasting correlations in the case of a static camera and scene or when doing offline rendering. Our hybrid shift criterion uses $\kappa_{\text{distance}} = 2d_{\max} = 2\max(\boldsymbol{d}.r, \max(\boldsymbol{d}.g, \boldsymbol{d}.b))$, which is twice the maximum scaled mean free path, and $\kappa_{\text{orientation}} = 0.5\ (60^\circ)$.

5.2 Render Time Optimizations

Since ReSTIR (SSS) reads and writes reservoir in each frame with random memory accesses for reading neighbors during spatial reuse, our application is currently memory-bound. Therefore, we reduce VRAM costs by compressing samples that are stored in reservoirs before writing to the VRAM similarly to G-buffer compression methods [Burns and Hunt 2013; Schied and Dachsbacher 2015]. To optimize the performance of the spatial pass, a streaming version of pairwise MIS is used to avoid loading all neighbor reservoirs at once and spilling registers. Additionally, we unroll the loop for iterating over neighbors to increase performance. To decrease memory access costs during candidate generation, we sample only a single new light source for a candidate and reuse the same light source across all found intersections. Of course, further optimizations, like using only a subset of lights in each frame as proposed for ReSTIR DI [Bitterli et al. 2020; Wyman and Panteleev 2022], can be applied here as well.

5.3 Global Illumination and Integration with ReSTIR GI/PT

We explained our technique for the direct illumination case only. Following ReSTIR PT [Lin et al. 2022], it is naturally possible to trace full paths and only store a single reconnection vertex, a seed for the random number generator, as well as the incoming radiance. To allow for the sequential shift, multiple vertices on the path can be stored along with their random numbers. Since there are no sharp details preserved after the light scattered through the object, we assume that querying a form of radiance cache [Dachsbacher et al. 2014; Majercik et al. 2022; Müller et al. 2021] at the sampled entry point can be beneficial allowing faster candidate generation and resampling passes.

[1] Project repository: https://github.com/MircoWerner/ReSTIR-SSS.git

ReSTIR SSS is a specialized technique to render paths with SSS. Using it as separate passes next to a technique for surface reflections like ReSTIR GI or ReSTIR PT leads to significant performance and memory overhead. The two techniques can be combined by letting shifts fail that try to reuse a subsurface sample on a different object. As most objects have a multi-layered material including translucent objects, shifts between an SSS lobe and a surface reflection lobe are prohibited. Similar to ReSTIR PT, this requires storing the lobe indices. In the supplemental material, we show how surface reflections can employ sequential shift as well.

6 EVALUATION

We evaluate our technique and shift mappings in different scenes on an NVIDIA RTX 3070 and compare to standard path tracing. Reported errors are calculated on the translucent objects only using unbiased converged references. Render times include the passes for ReSTIR SSS (candidate generation, spatial and temporal reuse, final shading) or standard path tracing. We do not include G-buffer costs and the costs of rendering non-translucent objects. For spatiotemporal reuse, the temporal history is recorded by moving the camera for 16 frames along a predefined short trajectory and applying candidate generation and spatiotemporal reuse once per frame. We additionally evaluate the influence of our method on denoising using the OptiX denoiser [NVIDIA Research 2017].

Results. Fig. 6 shows equal time comparisons of ReSTIR SSS with spatiotemporal reuse and the four possible shift mappings versus standard path tracing with and without applying the denoiser on an object that is highly affected by SSS. ReSTIR SSS does not introduce bias regardless of the shift and eventually converges to the reference (see supplemental material). While there is a slight bias with our implementation of the sequential shift due to the delayed calculation of the intersection index (Section 4.3), without this performance optimization the sequential shift is unbiased as well. Reconnection shift significantly reduces noise in shadowed regions with scattered light. However, it leads to strong correlation artifacts in illuminated regions, since it mainly reuses longer paths that transport only the dominant scattering color that is defined by the diffusion profile. Delayed reconnection shift outperforms reconnection shift in illuminated regions but finds an insufficient number of long scattering paths in shadowed regions. This leads to visible artifacts like blotches and discontinuities after denoising that are visually irritating and disagree with the general smoothing effect caused by SSS. Similar denoising artifacts can be seen for standard path tracing which only rarely finds long subsurface light transport paths from illuminated to shadowed regions. In contrast, our new hybrid and sequential shifts sample both regions well. This leads to a superior denoising quality compared to standard path tracing. In Fig. 7, we show the same scene but with many small light sources, which increases noise. Fig. 8 shows our technique with spatiotemporal reuse with hybrid and sequential shift for various mean free paths of different materials. As only one light source is used and the mean free paths are rather short, SSS is less noticeable in most regions, and the benefit of our method with direct lighting is comparatively small. In this case, BSSRDF IS searches for points in a relatively small sphere, which already decreases variance substantially. This effect is also clearly visible in Fig. 9, where we compare the same scene multiple times with the mean free path scaled to varying degrees. With increasing mean free paths, our technique outperforms standard path tracing in regions with noticeable SSS effects.

Correlation. Due to the spatiotemporal nature of ReSTIR, correlation artifacts between pixels occur. Although hybrid and sequential shift minimize these artifacts compared to (delayed) reconnection, the correlation is still visible by a change in the color tone that is noticeable at first. However, applying the denoiser re-creates the color tone of the reference (e.g. Fig. 8). Correlation can be traded for noise by reducing the influence of temporal resampling [Lin et al. 2022].

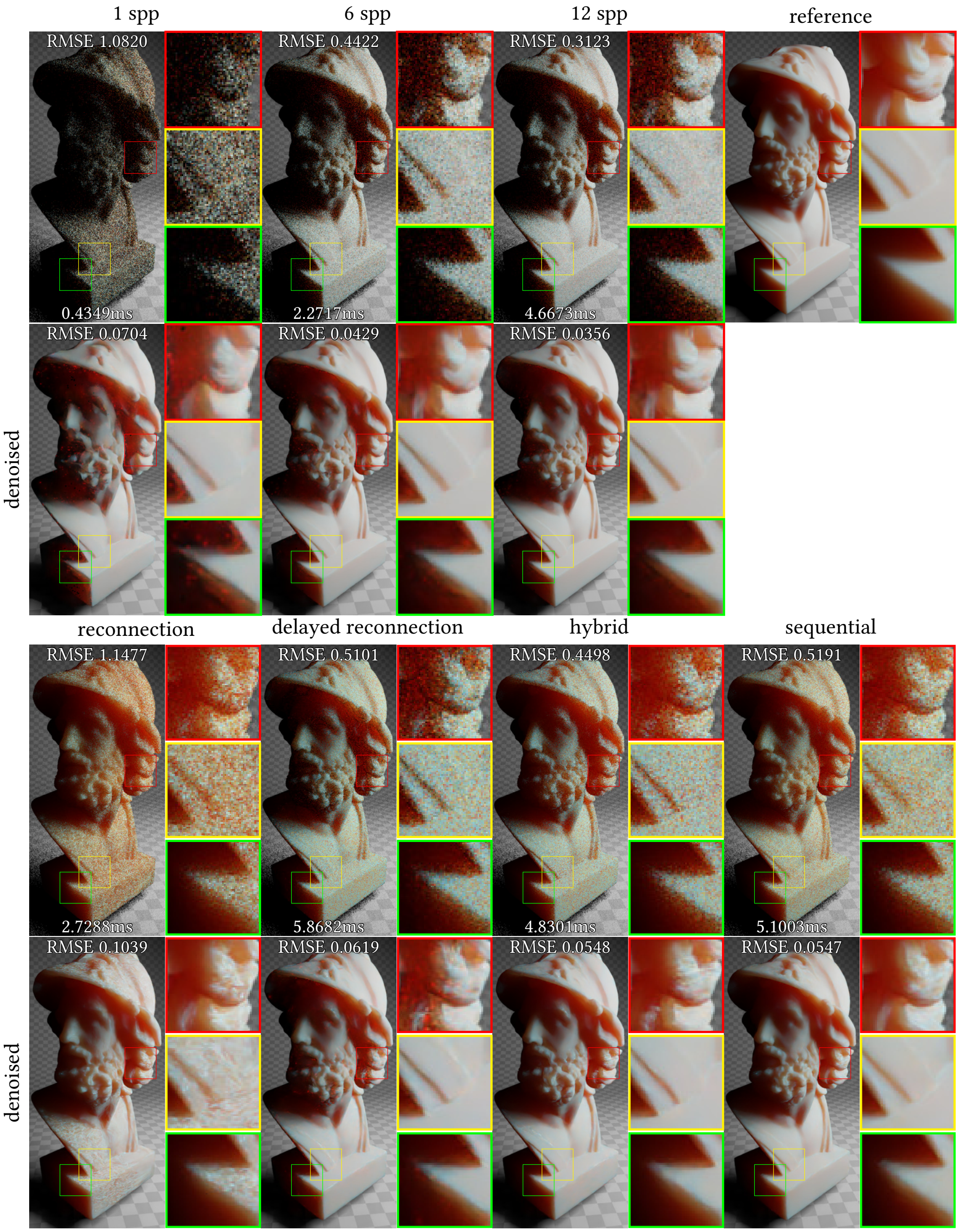


Fig. 6. Scene with a highly scattering object and a single large light source. Spatiotemporal reuse reduces noise significantly in regions with distinctly visible SSS compared to standard path tracing. Since there are both shadowed and illuminated regions that show SSS to varying degrees, reconnection shift and delayed reconnection shift only work well in one of the two regions at a time and lead to correlation artifacts in the other region. Hybrid and sequential shift sample both regions well and avoid most blotches and discontinuities after denoising.

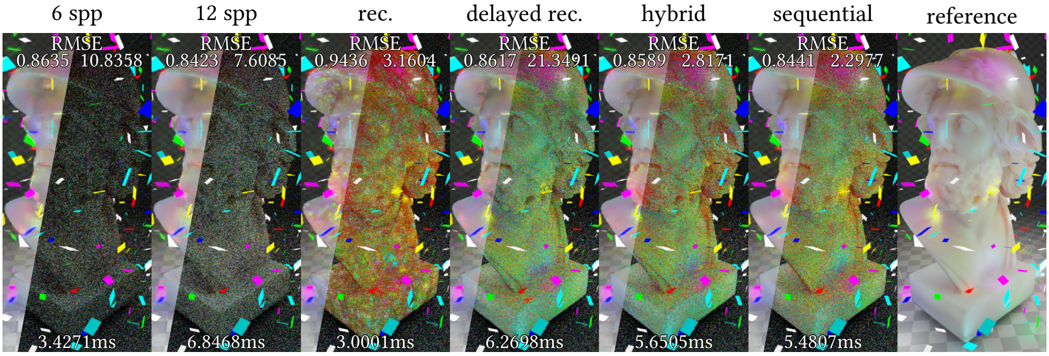


Fig. 7. Scene with a highly scattering object and many small light sources. Spatiotemporal reuse reduces noise significantly compared to standard path tracing. The denoised versions of the poorly sampled images from path tracing appear blurry. Hybrid and sequential shift lead to sharper denoised results.

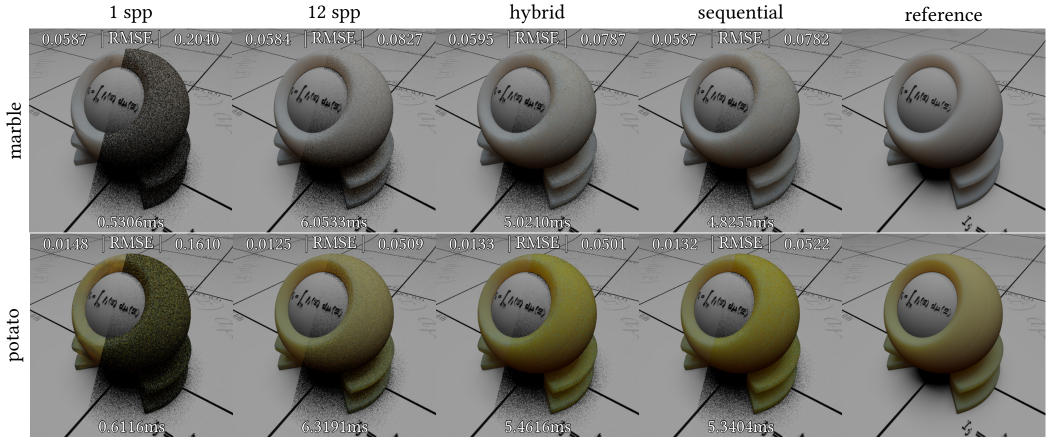


Fig. 8. Materials with small mean free paths in scenes with only one dominant light source benefit little from our introduced technique in the direct illumination case. Hybrid and sequential shift seem to change the color tone during spatiotemporal reuse, however, applying the denoiser re-creates the look of the reference image. Mean free paths and surface albedos are taken from Jensen et al. [2001].

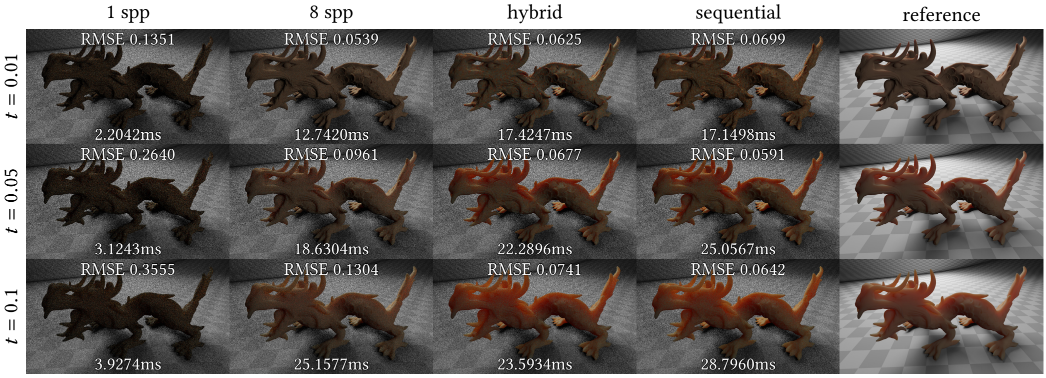


Fig. 9. Comparison of differently scaled mean free paths td. Our technique becomes more effective with longer mean free path. Standard path tracing fails to capture the red tint of longer scattering in equal time.

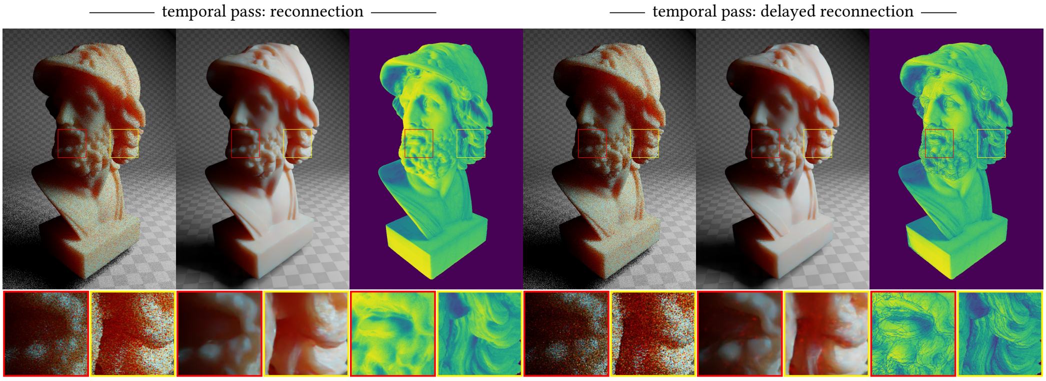

temporal pass: reconnection
temporal pass: delayed reconnection

Fig. 10. Sequential shift: Comparison between the use of reconnection and delayed reconnection in the temporal pass. Camera jitter and re-projection with motion vectors introduce a position mismatch on a subpixel level during temporal reuse. The change in geometry leads to differently shifted samples with delayed reconnection (random replay) compared to the last frame. This additional randomization generally leads to less uniformly distributed samples. The heatmap visualizes whether the shifted sample from the last frame $T(X_i)$ has a higher target function than the current sample X, i.e. $\hat{p}(T(X_i)) > \hat{p}(X)$, averaged over multiple frames. Brighter regions indicate that the shifted sample is more likely to have a higher target function. Using delayed reconnection for temporal reuse yields less improved samples in regions with higher geometric complexity.

Render Times. Our technique is currently memory-bound due to frequent reading and writing of reservoirs in all passes (cf. Section 5.2). As a consequence, it is possible to take a comparatively large number of samples per pixel in standard path tracing that is mainly limited by ray tracing performance. We assume that our technique becomes even more effective in the future, with increased memory performance, larger cache sizes, and higher register counts, when the ray tracing performance becomes the limiting factor as well.

Hybrid Versus Sequential Shift. Since our criterion for the hybrid shift is only based on local information of the base path and has to be satisfied for the offset path as well for symmetry reasons, finding the most optimal shift frequently fails at boundaries between the two regions we want to separate. We show this behavior in Fig. 5. By applying both shifts, the sequential shift implicitly picks the most successful one for a path and allows a more uniform sampling along the boundaries of the regions. Although we delay the calculation of the intersection index after the resampling pass with reconnection during the sequential shift, which improves performance, the second spatial resampling pass introduces additional memory accesses. As ReSTIR SSS is currently memory-bound, the sequential shift generally requires more time than the hybrid shift, although both use a total of four neighbors for spatial reuse.

Sequential Shift: Selection of Shifts. The sequential shift does not initially restrict whether a resampling pass uses reconnection or delayed reconnection. For temporal reuse, reconnection should be employed (see Fig. 10). We experimented with the order of reconnection and delayed reconnection in the two spatial passes but did not find either order to be clearly superior. As the error was slightly smaller, we decided to use reconnection first and delayed reconnection second.

7 CONCLUSION AND FUTURE WORK

We have introduced ReSTIR SSS that allows unbiased reuse of subsurface light transport paths. It minimizes the cost of generating points on the object with BSSRDF importance sampling by sharing them across pixels and between frames. Our introduced hybrid and sequential shifts combine reconnection and delayed reconnection that individually either decrease noise in regions with visible scattered light or in illuminated regions. This way, our combined shifts significantly reduce error in regions with noticeable scattered light while avoiding the occurrence of strong correlation artifacts in illuminated areas. Our sequential shift allows shifting a path with both reconnection and delayed reconnection by executing subsequent resampling passes, where each pass uses a different shift. In contrast to the hybrid shift, there is no need to deterministically decide on one shift for a given path. We believe that this ability to perform random replay on a path that has been previously constructed using reconnection is not only applicable to ReSTIR SSS, but might improve ReSTIR in general, e.g. for surface reflections, to decrease the number of failed shifts and in cases where no or only a suboptimal criterion for a hybrid shift can be defined.

ACKNOWLEDGMENTS

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

REFERENCES

Benedikt Bitterli. 2021. *Correlations and reuse for fast and accurate physically based light transport.* Ph. D. Dissertation. Dartmouth College.

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

Jesper Børlum, Brian Bunch Christensen, Thomas Kim Kjeldsen, Peter Trier Mikkelsen, Karsten Østergaard Noe, Jens Rimestad, and Jesper Mosegaard. 2011. SSLPV: Subsurface Light Propagation Volumes. In *Proceedings of the ACM SIGGRAPH Symposium on High Performance Graphics* (Vancouver, British Columbia, Canada) *(HPG '11)*. Association for Computing Machinery, New York, NY, USA, 7–14. https://doi.org/10.1145/2018323.2018325

Brent Burley. 2015. Extending the Disney BRDF to a BSDF with Integrated Subsurface Scattering. In *SIGGRAPH Courses: Physically Based Shading in Theory and Practice.* https://doi.org/10.1145/2776880.2787670

Christopher A. Burns and Warren A. Hunt. 2013. The Visibility Buffer: A Cache-Friendly Approach to Deferred Shading. *JCGT* 2, 2 (12 Aug. 2013), 55–69. http://jcgt.org/published/0002/02/04/

Min-Te Chao. 1982. A general purpose unequal probability sampling plan. *Biometrika* 69, 3 (Dec. 1982), 653–656. https://doi.org/10.1093/biomet/69.3.653

Per H. Christensen. 2015. An approximate reflectance profile for efficient subsurface scattering. In *SIGGRAPH Talks.* ACM, Article 25. https://doi.org/10.1145/2775280.2792555

Per H. Christensen, George Harker, Jonathan Shade, Brenden Schubert, and Dana Batali. 2012. Multiresolution radiosity caching for global illumination in movies. In *SIGGRAPH Talks.* ACM, Article 47. https://doi.org/10.1145/2343045.2343108

Carsten Dachsbacher, Jaroslav Křivánek, Miloš Hašan, Adam Arbree, Bruce Walter, and Jan Novák. 2014. Scalable Realistic Rendering with Many-Light Methods. *Computer Graphics Forum* 33, 1 (2014), 88–104. https://doi.org/10.1111/cgf.12256

Eugene D'Eon and Geoffrey Irving. 2011. A quantized-diffusion model for rendering translucent materials. *ACM Trans. Graph.* 30, 4, Article 56 (July 2011). https://doi.org/10.1145/2010324.1964951

Eugene d'Eon, David Luebke, and Eric Enderton. 2007. Efficient rendering of human skin. In *Rendering Techniques (Proc. EGSR).* Eurographics Association, 147–157. https://doi.org/10.2312/EGWR/EGSR07/147-157

Craig Donner and Henrik Wann Jensen. 2005. Light diffusion in multi-layered translucent materials. *ACM Trans. Graph.* 24, 3 (July 2005), 1032–1039. https://doi.org/10.1145/1073204.1073308

Evgenii Golubev. 2018. Efficient screen-space subsurface scattering using Burley's normalized diffusion in real-time. *SIGGRAPH Courses: Advances in Real-Time Rendering in Games Course* (2018). https://doi.org/10.1145/3214834.3264541

Evgenii Golubev. 2019. Sampling Burley's Normalized Diffusion Profiles. zero-radiance.github.io [Online; posted 2019-12-16].

Simon Green. 2004. Real-Time Approximations to Subsurface Scattering. In *GPU Gems: Programming Techniques, Tips and Tricks for Real-Time Graphics.* Pearson Higher Education, Chapter 16.

Adrien Gruson, Binh-Son Hua, Nicolas Vibert, Derek Nowrouzezahrai, and Toshiya Hachisuka. 2018. Gradient-domain volumetric photon density estimation. *ACM Trans. Graph.* 37, 4, Article 82 (July 2018). https://doi.org/10.1145/3197517.3201363

Ralf Habel, Per H. Christensen, and Wojciech Jarosz. 2013. Photon Beam Diffusion: A Hybrid Monte Carlo Method for Subsurface Scattering. *Computer Graphics Forum (Proc. EGSR)* 32, 4 (2013), 27–37. https://doi.org/10.1111/cgf.12148

Binh-Son Hua, Adrien Gruson, Derek Nowrouzezahrai, and Toshiya Hachisuka. 2017. Gradient-Domain Photon Density Estimation. *Computer Graphics Forum* (2017). https://doi.org/10.1111/cgf.13104

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

Henrik Wann Jensen and Juan Buhler. 2002. A rapid hierarchical rendering technique for translucent materials. *ACM Trans. Graph.* 21, 3 (July 2002), 576–581. https://doi.org/10.1145/566654.566619

Henrik Wann Jensen, Stephen R. Marschner, Marc Levoy, and Pat Hanrahan. 2001. A practical model for subsurface light transport. In *Annual Conference Series (Proc. SIGGRAPH)*. ACM, 511–518. https://doi.org/10.1145/383259.383319

Jorge Jimenez, Veronica Sundstedt, and Diego Gutierrez. 2009. Screen-space perceptual rendering of human skin. *ACM Trans. Appl. Percept.* 6, 4, Article 23 (Oct. 2009). https://doi.org/10.1145/1609967.1609970

Jorge Jimenez, Károly Zsolnai, Adrian Jarabo, Christian Freude, Thomas Auzinger, Xian-Chun Wu, Javier von der Pahlen, Michael Wimmer, and Diego Gutierrez. 2015. Separable Subsurface Scattering. *Computer Graphics Forum* 34, 6 (2015), 188–197. https://doi.org/10.1111/cgf.12529

James T. Kajiya. 1986. The Rendering Equation. *Computer Graphics (Proceedings of SIGGRAPH)* 20, 4 (Aug. 1986), 143–150. https://doi.org/10/cvf53j

Markus Kettunen, Daqi Lin, Ravi Ramamoorthi, Thomas Bashford-Rogers, and Chris Wyman. 2023. Conditional Resampled Importance Sampling and ReSTIR. In *SIGGRAPH Asia Conference Papers*. ACM, Article 91. https://doi.org/10.1145/3610548.3618245

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

Alan King, Christopher Kulla, Alejandro Conty, and Marcos Fajardo. 2013. BSSRDF importance sampling. In *SIGGRAPH Talks*. ACM, Article 48. https://doi.org/10.1145/2504459.2504520

Jaroslav Křivánek and Eugene d'Eon. 2014. A zero-variance-based sampling scheme for Monte Carlo subsurface scattering. In *SIGGRAPH Talks*. ACM, Article 66. https://doi.org/10.1145/2614106.2614138

Jaakko Lehtinen, Tero Karras, Samuli Laine, Miika Aittala, Frédo Durand, and Timo Aila. 2013. Gradient-domain metropolis light transport. *ACM Trans. Graph.* 32, 4, Article 95 (July 2013). https://doi.org/10.1145/2461912.2461943

Shiyu Liang, Yang Gao, Chonghao Hu, Peng Zhou, Aimin Hao, Lili Wang, and Hong Qin. 2024. State of the Art in Efficient Translucent Material Rendering with BSSRDF. *Computer Graphics Forum* 43, 1 (2024). https://doi.org/10.1111/cgf.14998

Daqi Lin, Markus Kettunen, Benedikt Bitterli, Jacopo Pantaleoni, Cem Yuksel, and Chris Wyman. 2022. Generalized resampled importance sampling: foundations of ReSTIR. *ACM Trans. Graph.* 41, 4, Article 75 (July 2022). https://doi.org/10.1145/3528223.3530158

Daqi Lin, Chris Wyman, and Cem Yuksel. 2021. Fast volume rendering with spatiotemporal reservoir resampling. *ACM Trans. Graph.* 40, 6, Article 279 (Dec. 2021). https://doi.org/10.1145/3478513.3480499

Zander Majercik, Thomas Müller, Alexander Keller, Derek Nowrouzezahrai, and Morgan McGuire. 2022. Dynamic Diffuse Global Illumination Resampling. *Computer Graphics Forum* 41, 1 (2022), 158–171. https://doi.org/10.1111/cgf.14427

Marco Manzi, Markus Kettunen, Frédo Durand, Matthias Zwicker, and Jaakko Lehtinen. 2016. Temporal gradient-domain path tracing. *ACM Trans. Graph.* 35, 6, Article 246 (Dec. 2016). https://doi.org/10.1145/2980179.2980256

Johannes Meng, Johannes Hanika, and Carsten Dachsbacher. 2016. Improving the Dwivedi Sampling Scheme. *Computer Graphics Forum* 35, 4 (2016), 37–44. https://doi.org/10.1111/cgf.12947

Thomas Müller, Fabrice Rousselle, Jan Novák, and Alexander Keller. 2021. Real-time neural radiance caching for path tracing. *ACM Trans. Graph.* 40, 4, Article 36 (July 2021). https://doi.org/10.1145/3450626.3459812

Jan Novák, Iliyan Georgiev, Johannes Hanika, and Wojciech Jarosz. 2018. Monte Carlo Methods for Volumetric Light Transport Simulation. *Computer Graphics Forum* 37, 2 (2018), 551–576. https://doi.org/10.1111/cgf.13383

NVIDIA Research. 2017. *NVIDIA OptiX™AI-Accelerated Denoiser.* https://developer.nvidia.com/optix-denoiser

Hisanari Otsu, Anton S. Kaplanyan, Johannes Hanika, Carsten Dachsbacher, and Toshiya Hachisuka. 2017. Fusing state spaces for markov chain Monte Carlo rendering. *ACM Trans. Graph.* 36, 4, Article 74 (July 2017). https://doi.org/10.1145/3072959.3073691

Y. Ouyang, S. Liu, M. Kettunen, M. Pharr, and J. Pantaleoni. 2021. ReSTIR GI: Path Resampling for Real-Time Path Tracing. *Computer Graphics Forum* 40, 8 (2021), 17–29. https://doi.org/10.1111/cgf.14378

Jacopo Pantaleoni. 2017. Charted metropolis light transport. *ACM Trans. Graph.* 36, 4, Article 75 (July 2017). https://doi.org/10.1145/3072959.3073677

Matt Pharr, Wenzel Jakob, and Greg Humphreys. 2016. *Physically Based Rendering: From Theory to Implementation* (3rd ed.). Morgan Kaufmann Publishers Inc.

Rohan Sawhney, Daqi Lin, Markus Kettunen, Benedikt Bitterli, Ravi Ramamoorthi, Chris Wyman, and Matt Pharr. 2024. Decorrelating ReSTIR Samplers via MCMC Mutations. *ACM Trans. Graph.* 43, 1, Article 10 (Jan. 2024). https://doi.org/10.1145/3629166

Christoph Schied and Carsten Dachsbacher. 2015. Deferred attribute interpolation for memory-efficient deferred shading. In *Proc. High-Performance Graphics*. ACM, 43–49. https://doi.org/10.1145/2790060.2790066

Justin Talbot, David Cline, and Parris Egbert. 2005. Importance Resampling for Global Illumination. In *Proc. EGSR*. Eurographics Association. https://doi.org/10.2312/EGWR/EGSR05/139-146

Bruce Walter, Pramook Khungurn, and Kavita Bala. 2012. Bidirectional lightcuts. *ACM Trans. Graph.* 31, 4, Article 59 (July 2012). https://doi.org/10.1145/2185520.2185555

Chris Wyman and Alexey Panteleev. 2022. Rearchitecting spatiotemporal resampling for production. In *Proc. High-Performance Graphics*. Eurographics Association, 23–41. https://doi.org/10.2312/hpg.20211281

Tiantian Xie, Marc Olano, Brian Karis, and Krzysztof Narkowicz. 2020. Real-time subsurface scattering with single pass variance-guided adaptive importance sampling. *Proc. ACM Comput. Graph. Interact. Tech.* 3, 1, Article 3 (May 2020). https://doi.org/10.1145/3384536

Matthias Zwicker, Wojciech Jarosz, Jaakko Lehtinen, Bochang Moon, Ravi Ramamoorthi, Fabrice Rousselle, Pradeep Sen, Cyril Soler, and Sung-Eui Yoon. 2015. Recent Advances in Adaptive Sampling and Reconstruction for Monte Carlo Rendering. *Computer Graphics Forum (Proc. Eurographics)* 34, 2 (May 2015), 667–681. https://doi.org/10/f7k6kj

A INVERSE MAPPINGS IN RESTIR

Measurement Integral with Extended Path Space. As long as $\bar{\boldsymbol{y}}$ is distributed uniformly in $\Gamma = [0,1]^m$, we can trivially add it to the integration domain. This keeps contribution weights W unbiased, i.e. it preserves the correctness of ReSTIR without any modifications:

$$\mathbb{E}\left[f(\bar{\boldsymbol{x}})W\right] = \int_{\mathcal{P}} f(\bar{\boldsymbol{x}})\,\mathrm{d}\bar{\boldsymbol{x}} = \int_{\mathcal{P}} \underbrace{\int_{\Gamma} \mathrm{d}\bar{\boldsymbol{y}}}_{=1}\; f(\bar{\boldsymbol{x}})\,\mathrm{d}\bar{\boldsymbol{x}} = \int_{\mathcal{P}\times\Gamma} \underbrace{f(\bar{\boldsymbol{x}},\bar{\boldsymbol{y}})}_{:=f(\bar{\boldsymbol{x}})}\,\mathrm{d}(\bar{\boldsymbol{x}},\bar{\boldsymbol{y}}) = \mathbb{E}\left[f(\bar{\boldsymbol{x}},\bar{\boldsymbol{y}})W\right] \tag{12}$$

Canceling the Jacobian. We formalize the mapping from and to PSS using the *sampling scheme* $S\colon \mathcal{U} \to \mathcal{P}\times\Gamma$ and its *inversion* $R\colon \mathcal{P}\times\Gamma \to \mathcal{U}$. The sampling scheme S is usually given by our candidate generation, while we would like to derive a suitable inversion R. R should not change the path density, i.e. the PSS integral should behave as if we had started out in PSS. This means the value of the integral of inversion followed by sampling (application of S) should match the path space integral:

$$\begin{aligned}\int_{\mathcal{P}\times\Gamma} f(\bar{\boldsymbol{x}},\bar{\boldsymbol{y}})\,\mathrm{d}(\bar{\boldsymbol{x}},\bar{\boldsymbol{y}}) &= \int_{\mathcal{U}} f(S(\bar{\boldsymbol{u}}))\,\left|J_S(\bar{\boldsymbol{u}})\right|\,\mathrm{d}\bar{\boldsymbol{u}} \\ &= \int_{\mathcal{P}\times\Gamma} f(S(R(\bar{\boldsymbol{x}},\bar{\boldsymbol{y}})))\left|J_S(R(\bar{\boldsymbol{x}},\bar{\boldsymbol{y}}))\right|\left|J_R(\bar{\boldsymbol{x}},\bar{\boldsymbol{y}})\right|\,\mathrm{d}(\bar{\boldsymbol{x}},\bar{\boldsymbol{y}})\end{aligned} \tag{13}$$

This is the case when the Jacobians in the last line cancel each other:

$$\left|J_S(R(\bar{\boldsymbol{x}},\bar{\boldsymbol{y}}))\right|\left|J_R(\bar{\boldsymbol{x}},\bar{\boldsymbol{y}})\right| = 1 \tag{14}$$

We choose R such that this property holds following Bitterli et al. [2017] in Sections 4.2 and 4.3 and then can perform integration in PSS as usual. Also see the supplemental material, where we give an explicit example for sampling a mixture distribution and its inversion as well as an example of how to apply the sequential shift for ReSTIR in the context of surface reflections.