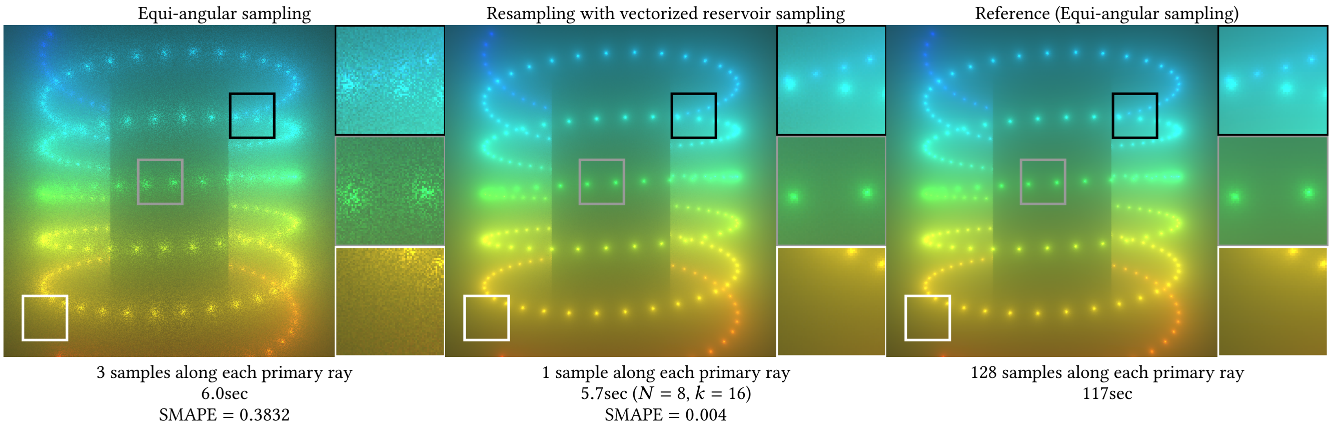
Vectorized Reservoir Sampling

Shinji Ogaki
ZOZO, Inc.
shinji.ogaki@gmail.com



Equi-angular sampling | Resampling with vectorized reservoir sampling | Reference (Equi-angular sampling)

3 samples along each primary ray
6.0sec
SMAPE = 0.3832

1 sample along each primary ray
5.7sec ($N = 8$, $k = 16$)
SMAPE = 0.004

128 samples along each primary ray
117sec

Figure 1: Single scattering of point light sources in a strongly forward scattering participating medium. For all images 256 primary rays are cast per pixel. Left: Equi-angular sampling [Kulla and Fajardo 2012]. Center: Resampling with vectorized reservoir sampling produces better results by generating samples from the product of the geometric term, phase function, and transmittance. N is the length of each substream and k is the number of substreams. Right: Reference.

ABSTRACT

Reservoir sampling is becoming an essential component of realtime rendering as it enables importance resampling with limited storage. Chao's weighted random sampling algorithm is a popular choice because of its simplicity. Although it is elegant, there is a fundamental issue that many random numbers must be generated to update reservoirs. To address this issue, we modify Chao's algorithm with sample warping. We apply sample warping in two different ways and compare them. We further vectorize the modified algorithm to make reservoir sampling more useful for CPU rendering and give a couple of practical examples.

CCS CONCEPTS

• **Computing methodologies → Rendering**.

KEYWORDS

rendering, importance resampling, reservoir sampling

ACM Reference Format:
Shinji Ogaki. 2021. Vectorized Reservoir Sampling. In *SIGGRAPH Asia 2021 Technical Communications (SA '21 Technical Communications), December 14–17, 2021, Tokyo, Japan.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3478512.3488602

SA '21 Technical Communications, December 14–17, 2021, Tokyo, Japan

ACM ISBN 978-1-4503-9073-6/21/12…$15.00
https://doi.org/10.1145/3478512.3488602

1 INTRODUCTION

The ReSTIR algorithm [Bitterli et al. 2020; Wyman and Panteleev 2021] reuses samples in space and time to effectively compute direct illumination from many lights. Recently, it was extended to handle global illumination [Ouyang et al. 2021] and participating media [Lin et al. 2021] at an interactive frame rate. All of these rely on Chao [1982]'s algorithm because of its simplicity. A problem common to weighted random sampling algorithms (e.g., [Chao 1982; Efraimidis and Spirakis 2006]) is that they require as many random numbers as the length of a data stream. Although Efraimidis [2015] significantly reduced required random numbers by applying the jumps technique to Chao's algorithm, many random numbers are still necessary for a long data stream. In this paper we show that using a single random number suffices to randomly select a sample from a moderate-length stream with the help of sample warping [McCool and Harwood 1997]. We apply sample warping in two different ways, explicitly and implicitly, and conclude explicit warping is better.

It may appear that the combination of resampling and reservoir sampling only brings significant benefits to realtime rendering. However, we show that the vectorized version of the proposed algorithm can be useful for CPU renderers by giving two examples. If it is possible to vectorize the target function (e.g., by using enoki [Jakob 2019]), our approach can be used as a substitution for complex sampling algorithms.

2 APPLYING SAMPLE WARPING

Chao's algorithm. Chao [1982]'s algorithm randomly selects one or more samples from a stream of samples of unknown length, depending on their weights. To simplify the following discussion, we will only consider a situation where only one sample is selected from a sequence of N samples (Alg. 1). The acceptance probability of the nth $(0 < n)$ sample with weight w_n is given as

$$\frac{w_n}{\sum_{i=1}^{N} w_i}.$$

Algorithm 1: Chao's algorithm

```
Input: a stream of N weights $\{w_1, w_2, \cdots, w_N\}$
Result: sample index $I$
Function Select($\{w_1, w_2, \cdots, w_N\}$):
    $I \leftarrow \emptyset, w_{sum} \leftarrow 0$
    for $i \leftarrow 1$ to $N$ do
        $w_{sum} \leftarrow w_{sum} + w_i$
        if random() $< w_i / w_{sum}$ then
            $I \leftarrow i$
    return $I$
```

Chao's algorithm with explicit warping. Applying sample warping [McCool and Harwood 1997] to Chao's algorithm is straightforward (Alg. 2). For simplicity, let $p_n = \frac{w_n}{\sum_{i=1}^{n} w_i}$ and $q_n = 1 - p_n$. With sample warping, the next (i.e. stretched) random number after acceptance is given by

$$\xi_{n+1} = \frac{\xi_n}{p_n}, \tag{1}$$

and similarly after rejection it is given by

$$\xi_{n+1} = \frac{\xi_n - p_n}{q_n}. \tag{2}$$

The $n + k$th $(0 \leq k)$ sample is accepted if

$$\xi_{n+k} < p_{n+k}. \tag{3}$$

Since we directly manipulate the random number, we call this method *explicit warping*. This approach is also used in the concurrent work by Zirr [2021].

Algorithm 2: Chao's algorithm with explicit warping

```
Input: a stream of N weights $\{w_1, w_2, \cdots, w_N\}$ and random
        number $\xi$
Result: sample index $I$
Function Select($\{w_1, w_2, \cdots, w_N\}, \xi$):
    $I \leftarrow \emptyset, w_{sum} \leftarrow 0$
    for $i \leftarrow 1$ to $N$ do
        $w_{sum} \leftarrow w_{sum} + w_i, p \leftarrow w_i / w_{sum}$
        if $\xi < p$ then
            $I \leftarrow i, \xi \leftarrow \xi / p$
        else
            $\xi \leftarrow (\xi - p)/(1 - p)$
    return $I$
```

Chao's algorithm with implicit warping. In explicit warping, the random number is stretched multiple times. Here we present an alternative method that does not modify the given random number at all. We call this method *implicit warping*. By recursively expanding ξ_{n+k} using Eq. (1) and (2), the inequality (3) can be rewritten as

$$\xi_n < \underbrace{\sum_{j=0}^{k-1} A_{n+j} \prod_{i=0}^{j-1} B_{n+i}}_{r \text{ in Alg. 3}} + p_{n+k} \underbrace{\prod_{i=0}^{k-1} B_{n+i}}_{c \text{ in Alg. 3}}$$

with two auxiliary sequences

$$A_n = \begin{cases} 0 & \text{if } n\text{th sample is accepted} \\ p_n & \text{if } n\text{th sample is rejected} \end{cases}$$

and

$$B_n = \begin{cases} p_n & \text{if } n\text{th sample is accepted} \\ q_n & \text{if } n\text{th sample is rejected.} \end{cases}$$

Therefore, the acceptance test of the n+kth sample can be performed using the nth random number ξ_n. To give a concrete example, the $n + 2$th sample is accepted after two consecutive rejections if

$$\xi_n < p_n + q_n p_{n+1} + q_n q_{n+1} p_{n+2}.$$

In the practical implementation (Alg. 3), we use two additional buffers: one for storing the first term of the right hand side, and another for the coefficient of p_{n+k}.

Algorithm 3: Chao's algorithm with implicit warping

```
  Input: a stream of N weights $\{w_1, w_2, \cdots, w_N\}$ and random
          number $\xi$
  Result: sample index $I$
1 Function Select($\{w_1, w_2, \cdots, w_N\}, \xi$):
2     $r \leftarrow 0, c \leftarrow 1, I \leftarrow \emptyset, w_{sum} \leftarrow 0$
3     for $i \leftarrow 1$ to $N$ do
4         $w_{sum} \leftarrow w_{sum} + w_i, p \leftarrow w_i / w_{sum}$
5         if $\xi < r + cp$ then
6             $I \leftarrow i, c \leftarrow cp$
7         else
8             $r \leftarrow r + cp, c \leftarrow c(1 - p)$
9     return $I$
```

Theoretically, implicit warping is equivalent to explicit warping, but it yields worse results when a stream becomes long. Since c decreases rapidly and monotonically, and p also tends to decrease, the floating point number r will soon stop changing unless a sample with a very large weight appears. Therefore, it is recommended to use explicit warping. While not practically useful, this implicit reformulation helps to reveal the connection to Efraimidis [2015]' jumps technique. Efraimidis only applies jumps to the situation where rejections occur and draws a fresh random number whenever a sample is accepted. In fact, the jumps technique is a partial use of implicit warping (line 8 in Alg. 3) and results in a limited reduction in calls of a random number generator. When selecting a single sample from a data stream of length N, Efraimidis's method reduces the number of calls from $O(N)$ to $O(log(N))$. On the other hand, Chao's algorithm with sample warping reduces the number to $O(1)$.

3 VECTORIZATION

One of our goals is to make reservoir sampling more useful for CPU offline rendering. To this end, we split a stream into multiple substreams. The reasons are twofold. First, splitting the stream can improve the efficiency by exploiting SIMD units. This can be easily done because only one random number is required to select a sample as shown in the previous section. Second, the use of short streams reduces numerical errors [Wyman and Panteleev 2021].

Let denote k be the number of SIMD lanes (e.g., $k = 8$ for AVX and $k = 16$ for AVX-512). We divide the input stream into equal-sized k-disjoint substreams such that the sum of weights of each substream is roughly equal. We use k independent reservoirs and they are updated simultaneously. Note that we can use the same random number for all k substreams because they do not overlap. In addition, stratification can be slightly improved since a single reservoir is associated with each substream. Once all the substreams have been scanned, a single reservoir is selected based on the sum of weights stored in each one using another random number. Thus we use two random numbers for the vectorized version.

In general, the calculation of the weights including BRDF evaluation and geometric term computation dominates the processing time in scanning substreams. The overhead of generating random numbers might be negligible for low cost generators (e.g., [Jarzynski and Olano 2020]). However, reducing the calls of a random number generator is important especially when a sophisticated low discrepancy sequence is used. Being able to use the same random number for all substreams means that it is not necessary to vectorize algorithms to generate such sequences.

4 APPLICATIONS

We briefly review importance resampling [Talbot et al. 2005]. Then, we show a couple of applications which benefit from the proposed algorithm.

The unbiased estimate of the integral $L = \int f(x)dx$ can be obtained with importance resampling as

$$\hat{L} = \frac{f(x')}{\hat{p}(x')} \frac{1}{N} \sum_{i=1}^{N} w(x_i), \tag{4}$$

where $w(x_i) = \frac{\hat{p}(x_i)}{p(x_i)}$. The set of samples $\{x_1, x_2, \cdots, x_N\}$ is generated from the source pdf $p(x)$, and x' is selected stochastically from the set according to the probability

$$\frac{w(x')}{\sum_{i=1}^{N} w(x_i)}.$$

The target function $\hat{p}$ should closely approximate the integrand f to be efficient. By letting the weight of x'

$$W(x') = \frac{1}{N\hat{p}(x')} \sum_{i=1}^{N} w(x_i),$$

the estimate of L is given by $\hat{L} = f(x')W(x')$. Bitterli et al. [2020] showed that this resampling can be implemented as a streaming algorithm using weighted reservoir sampling.

We implemented the proposed vectorized reservoir sampling in our research renderer, and all experiments were conducted using Intel®Core™i7-11800H.

Example 1. Single scattering. In the first example, we apply vectorized reservoir sampling to computing single scattering in a homogeneous participating medium. It is obtained by the integral

$$L = \int_{t_{min}}^{t_{max}} \underbrace{\underbrace{L_e \rho(\boldsymbol{\omega}_i, \boldsymbol{\omega}) e^{-\mu(t+|\boldsymbol{x}_l - \boldsymbol{x}_t|)} G(\boldsymbol{x}_l, \boldsymbol{x}_t)}_{\text{target function } \hat{p}(\boldsymbol{x}_t)} V(\boldsymbol{x}_l, \boldsymbol{x}_t)\, dt}_{\text{integrand } f},$$

where L_e is the emitted radiance, ρ the phase function including the scattering coefficient, $\boldsymbol{\omega}_i = (\boldsymbol{x}_l - \boldsymbol{x}_t)|\boldsymbol{x}_l - \boldsymbol{x}_t|^{-1}$, μ the attenuation coefficient, $\boldsymbol{x}_l$ the position of a light source, V the visibility function, and the geometric term is given as $G(\boldsymbol{x}_l, \boldsymbol{x}_t) = |\boldsymbol{x}_l - \boldsymbol{x}_t|^{-2}$. We compute the integral along the ray $\boldsymbol{x}_t = \boldsymbol{o} + \boldsymbol{\omega}t$, where $\boldsymbol{o}$ is the origin and $\boldsymbol{\omega}$ is the viewing direction. We placed 256 point light sources in a box filled with a homogeneous participating medium and used Schlick phase function with parameter 0.9 which exhibits a strongly forward scattering property. The source pdf is the uniform density $p(\boldsymbol{x}_t) = \frac{1}{t_{max} - t_{min}}$. Our method achieves a lower *symmetric mean absolute percentage error* (SMAPE) value than the equi-angular sampling [Kulla and Fajardo 2012] with a slightly less amount of time as shown in Fig. 1. The comparison and combining with the state-of-the-art method by Villeneuve et al. [2021] is our future work.

Example 2. Direct illumination. Next, we compute the direct illumination from a triangular light source arriving at a Lambertian surface. It can be computed by an integral over the triangle surface A:

$$L = \int_A \underbrace{\underbrace{L_e \rho(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) G(\boldsymbol{x}_l, \boldsymbol{x}_s)}_{\text{target function } \hat{p}(\boldsymbol{x}_l)} V(\boldsymbol{x}_l, \boldsymbol{x}_s)\, dA(\boldsymbol{x}_l)}_{\text{integrand } f},$$

where L_e is the emitted radiance, ρ the BRDF, and the geometric term is given as $G(\boldsymbol{x}_l, \boldsymbol{x}_s) = \langle -\boldsymbol{\omega}_i, \boldsymbol{n}_l \rangle \langle \boldsymbol{\omega}_i, \boldsymbol{n}_s \rangle |\boldsymbol{x}_l - \boldsymbol{x}_s|^{-2}$. The incoming direction is given as $\boldsymbol{\omega}_i = (\boldsymbol{x}_l - \boldsymbol{x}_s)|\boldsymbol{x}_l - \boldsymbol{x}_s|^{-1}$, and $\boldsymbol{\omega}_o$ is the outgoing direction, $\boldsymbol{n}_l$ the normal of the light source, and $\boldsymbol{n}_s$ the normal at the shading point $\boldsymbol{x}_s$. We use the source pdf $p(\boldsymbol{x}_l) = \frac{1}{A}$. Writing the estimate in the form of Eq. (4), we have

$$\hat{L} = V(\boldsymbol{x}_l', \boldsymbol{x}_s) \underbrace{\frac{A}{N} \sum_{j=1}^{N} L_e \rho(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) G(\boldsymbol{x}_{lj}, \boldsymbol{x}_s)}_{\text{unshadowed illumination } U} = V(\boldsymbol{x}_l', \boldsymbol{x}_s) U.$$

If ρ is the Lambertian BRDF (more generally, if it can be described by a linearly transformed cosine [Heitz et al. 2016]), we can compute the unshadowed illumination U by the boundary integration method. Thus $\hat{L}$ can be interpreted as the ratio estimator [Heitz et al. 2018]. Heitz et al. used solid angle sampling. Instead, we use $\hat{p}(\boldsymbol{x}_l) = L_e \rho(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) G(\boldsymbol{x}_l, \boldsymbol{x}_s)$ as the target function. Note that reservoir sampling is only used to pick a sample $\boldsymbol{x}_l'$ on the triangle. This formulation works without any intricate math and dramatically simplifies the implementation of projected solid angle sampling. Nevertheless, we achieved a performance comparable to the state-of-the-art method by Peters [2021] as shown in Fig. 2.

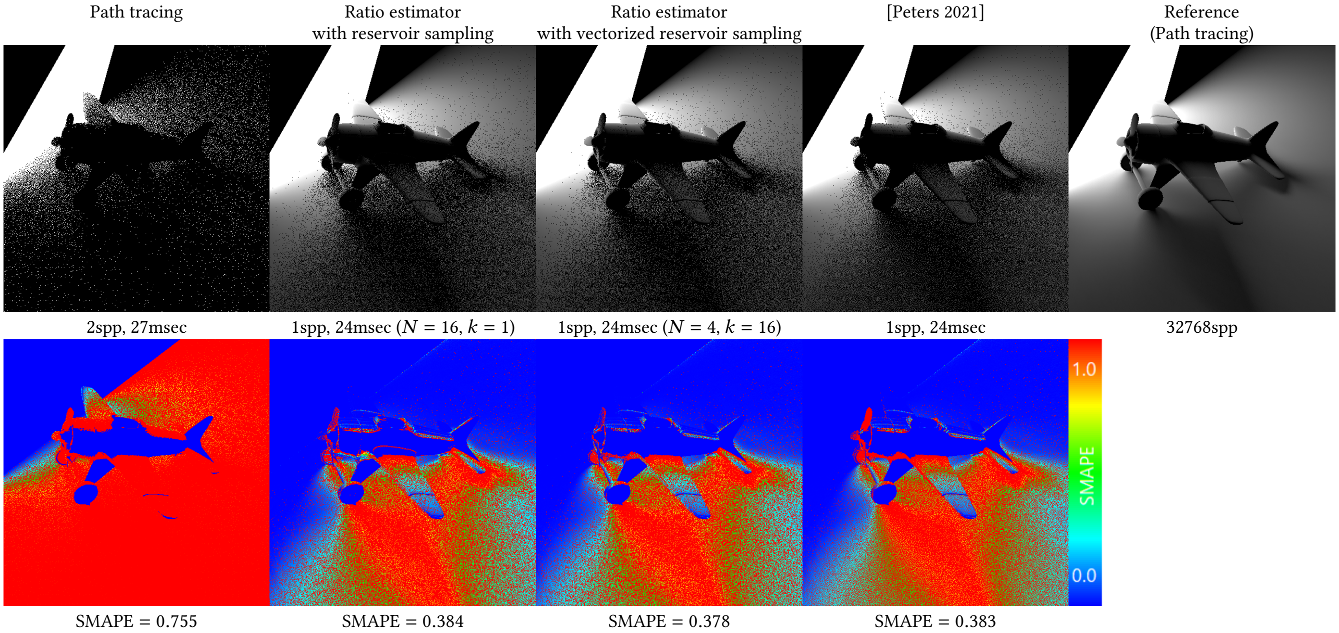

Path tracing
Ratio estimator with reservoir sampling
Ratio estimator with vectorized reservoir sampling
[Peters 2021]
Reference (Path tracing)
2spp, 27msec
1spp, 24msec ($N = 16$, $k = 1$)
1spp, 24msec ($N = 4$, $k = 16$)
1spp, 24msec
32768spp
1.0
SMAPE
0.0
SMAPE = 0.755
SMAPE = 0.384
SMAPE = 0.378
SMAPE = 0.383

Figure 2: A plane lit by a triangular light. Left: Path tracing with cosine-weighted hemispherical sampling. Center Left: Ratio estimator with reservoir sampling. Center: Ratio estimator with vectorized reservoir sampling. Center Right: The method by Peters [2021]. Right: Reference. N is the length of each substream and k is the number of substreams.

5 CONCLUSION AND FUTURE WORK

By combining sample warping with Chao's weighted random sampling, we showed that only a single random number is required to generate a sample. With the two examples we also showed that vectorized reservoir sampling can be a simpler alternative to the state-of-the-art sampling methods and potentially be very useful, especially for CPU renderers.

For future work, we would like to improve numerical robustness. A naive solution would be to split a long sequence into subsequences and change the scanning order in such a way that numerical errors are reduced. Although we did not use vectorized reservoir sampling in conjunction with multiple importance sampling in the two examples, it would be interesting to find an efficient way to evaluate pdfs for samples generated by other methods.

The code and additional results can be found in the supplemental material.

REFERENCES

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 (July 2020), 17 pages. https://doi.org/10.1145/3386569.3392481

M. T. Chao. 1982. A General Purpose Unequal Probability Sampling Plan. *Biometrika* 69, 3 (1982), 653–656. http://www.jstor.org/stable/2336002

Pavlos S. Efraimidis. 2015. *Weighted Random Sampling over Data Streams.* Springer International Publishing, Cham, 183–195. https://doi.org/10.1007/978-3-319-24024-4_12

Pavlos S. Efraimidis and Paul G. Spirakis. 2006. Weighted random sampling with a reservoir. *Inform. Process. Lett.* 97, 5 (2006), 181–185. https://doi.org/10.1016/j.ipl.2005.11.003

Eric Heitz, Jonathan Dupuy, Stephen Hill, and David Neubelt. 2016. Real-Time Polygonal-Light Shading with Linearly Transformed Cosines. *ACM Trans. Graph.* 35, 4, Article 41 (July 2016), 8 pages. https://doi.org/10.1145/2897824.2925895

Eric Heitz, Stephen Hill, and Morgan McGuire. 2018. Combining Analytic Direct Illumination and Stochastic Shadows. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (Montreal, Quebec, Canada) *(I3D '18)*. Association for Computing Machinery, New York, NY, USA, Article 2, 11 pages. https://doi.org/10.1145/3190834.3190852

Wenzel Jakob. 2019. Enoki: structured vectorization and differentiation on modern processor architectures. https://github.com/mitsuba-renderer/enoki.

Mark Jarzynski and Marc Olano. 2020. Hash Functions for GPU Rendering. *Journal of Computer Graphics Techniques (JCGT)* 9, 3 (17 October 2020), 20–38. http://jcgt.org/published/0009/03/02/

Christopher Kulla and Marcos Fajardo. 2012. Importance Sampling Techniques for Path Tracing in Participating Media. *Comput. Graph. Forum* 31, 4 (June 2012), 1519–1528. https://doi.org/10.1111/j.1467-8659.2012.03148.x

Daqi Lin, Chris Wyman, and Cem Yuksel. 2021. Fast Volume Rendering with Spatiotemporal Reservoir Resampling. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2021)* 40, 6, Article 278 (12 2021), 18 pages. https://doi.org/10.1145/3478513.3480499

Michael D. McCool and Peter K. Harwood. 1997. Probability trees. In *Proceedings of the Graphics Interface 1997 Conference, May 21-23, 1997, Kelowna, BC, Canada.* 37–46. http://graphicsinterface.org/wp-content/uploads/gi1997-5.pdf

Yaobin Ouyang, Shiqiu Liu, Markus Kettunen, Matt Pharr, and Jacopo Pantaleoni. 2021. ReSTIR GI: Path Resampling for Real-Time Path Tracing. *Computer Graphics Forum* (2021). https://doi.org/10.1111/cgf.14378

Christoph Peters. 2021. BRDF Importance Sampling for Polygonal Lights. *ACM Trans. Graph.* 40, 4, Article 140 (July 2021), 14 pages. https://doi.org/10.1145/3450626.3459672

Justin Talbot, David Cline, and Parris Egbert. 2005. Importance Resampling for Global Illumination. In *Eurographics Symposium on Rendering (2005)*, Kavita Bala and Philip Dutre (Eds.). The Eurographics Association. https://doi.org/10.2312/EGWR/EGSR05/139-146

Keven Villeneuve, Adrien Gruson, Iliyan Georgiev, and Derek Nowrouzezahrai. 2021. Practical Product Sampling for Single Scattering in Media. In *Eurographics Symposium on Rendering - DL-only Track*, Adrien Bousseau and Morgan McGuire (Eds.). The Eurographics Association. https://doi.org/10.2312/sr.20211290

Chris Wyman and Alexey Panteleev. 2021. Rearchitecting Spatiotemporal Resampling for Production. In *High-Performance Graphics - Symposium Papers*, Nikolaus Binder and Tobias Ritschel (Eds.). The Eurographics Association. https://doi.org/10.2312/hpg.20211281

Tobias Zirr. 2021. *Light Sampling in Quake 2 Using Subset Importance Sampling.* Apress, Berkeley, CA, 765–790. https://doi.org/10.1007/978-1-4842-7185-8_47