ReSTIR GI: Path Resampling for Real-Time Path Tracing

High-Performance Graphics 2021
Date: 2021.12.08.
Presenter: Chanhyeok Lee

English subtitle available

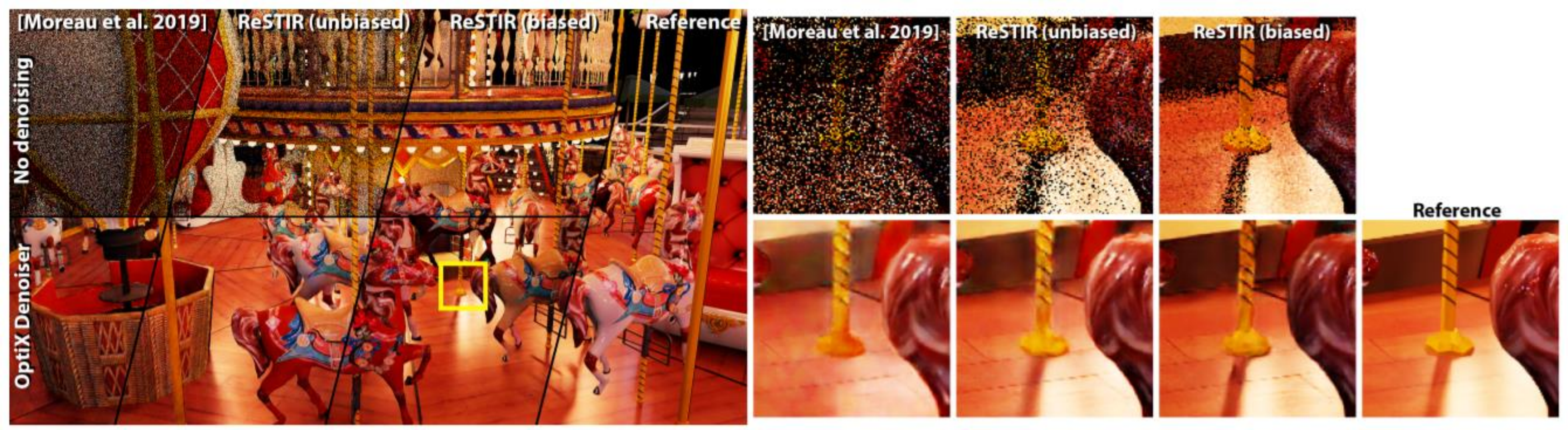
Contents

- Introduction
- Original ReSTIR algorithm
- Improvements of ReSTIR GI
- Results
- Conclusion

Introduction

- SPP is limited in a real-time path tracing application
- Denosing is required after rendering with small SPP.
- Maximize the quality of rendered images with limited number of samples

Introduction



Bitterli, Benedikt, Chris Wyman, Matt Pharr, Peter Shirley, Aaron Lefohn, and Wojciech Jarosz. "Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting." *ACM Transactions on Graphics (TOG)* 39, no. 4 (2020): 148-1.

Resampled Importance Sampling

- IS: $$\langle L\rangle_{\text{is}}^{N} = \frac{1}{N}\sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)} \approx L.$$

- MIS: $$\langle L\rangle_{\text{mis}}^{M,N} = \sum_{s=1}^{M} \frac{1}{N_s}\sum_{i=1}^{N_s} w_s(x_i)\frac{f(x_i)}{p_s(x_i)}.$$

- RIS:

$$\langle L\rangle_{\text{ris}}^{1,M} = \frac{f(y)}{\hat{p}(y)}\cdot\left(\frac{1}{M}\sum_{j=1}^{M} \mathrm{w}(x_j)\right)$$

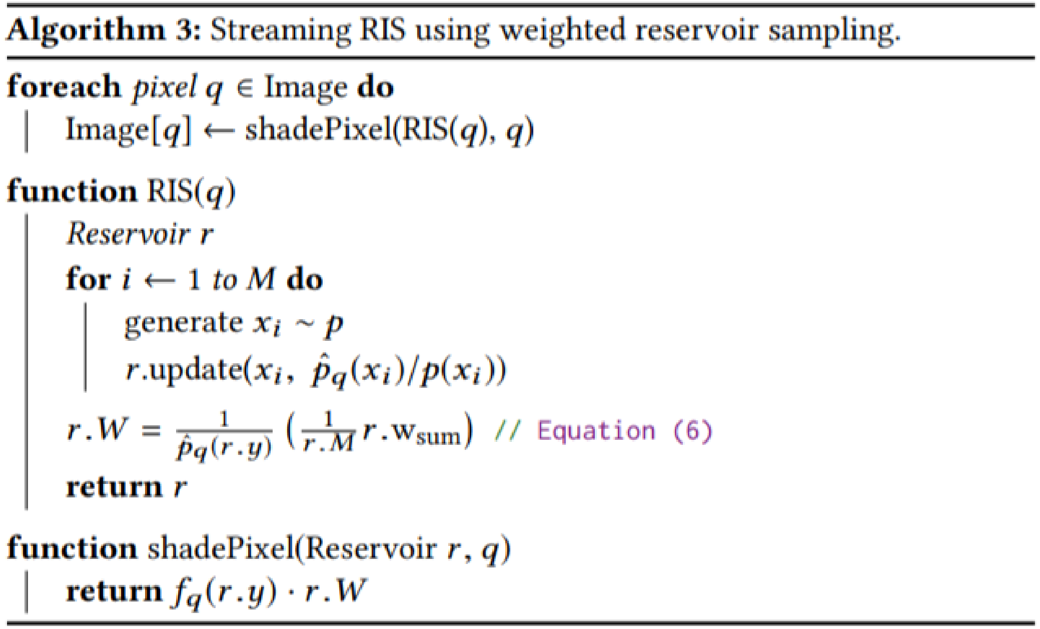
$$\langle L\rangle_{\text{ris}}^{N,M} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{f(y_i)}{\hat{p}(y_i)}\cdot\left(\frac{1}{M}\sum_{j=1}^{M} \mathrm{w}(x_{ij})\right)\right)$$

$$p(z \mid \mathbf{x}) = \frac{\mathrm{w}(x_z)}{\sum_{i=1}^{M} \mathrm{w}(x_i)} \quad \text{with} \quad \mathrm{w}(x) = \frac{\hat{p}(x)}{p(x)},$$

Weighted Reservoir Sampling

- Sampling probability: $P_i = \frac{\mathrm{w}(x_i)}{\sum_{j=1}^{M} \mathrm{w}(x_j)}.$
- Probability of new element: $\frac{\mathrm{w}(x_{m+1})}{\sum_{j=1}^{m+1} \mathrm{w}(x_j)},$
- Probability updating rule: $\frac{\mathrm{w}(x_i)}{\sum_{j=1}^{m} \mathrm{w}(x_j)} \left(1 - \frac{\mathrm{w}(x_{m+1})}{\sum_{j=1}^{m+1} \mathrm{w}(x_j)}\right) = \frac{\mathrm{w}(x_i)}{\sum_{j=1}^{m+1} \mathrm{w}(x_j)},$

Streaming RIS using reservoir sampling

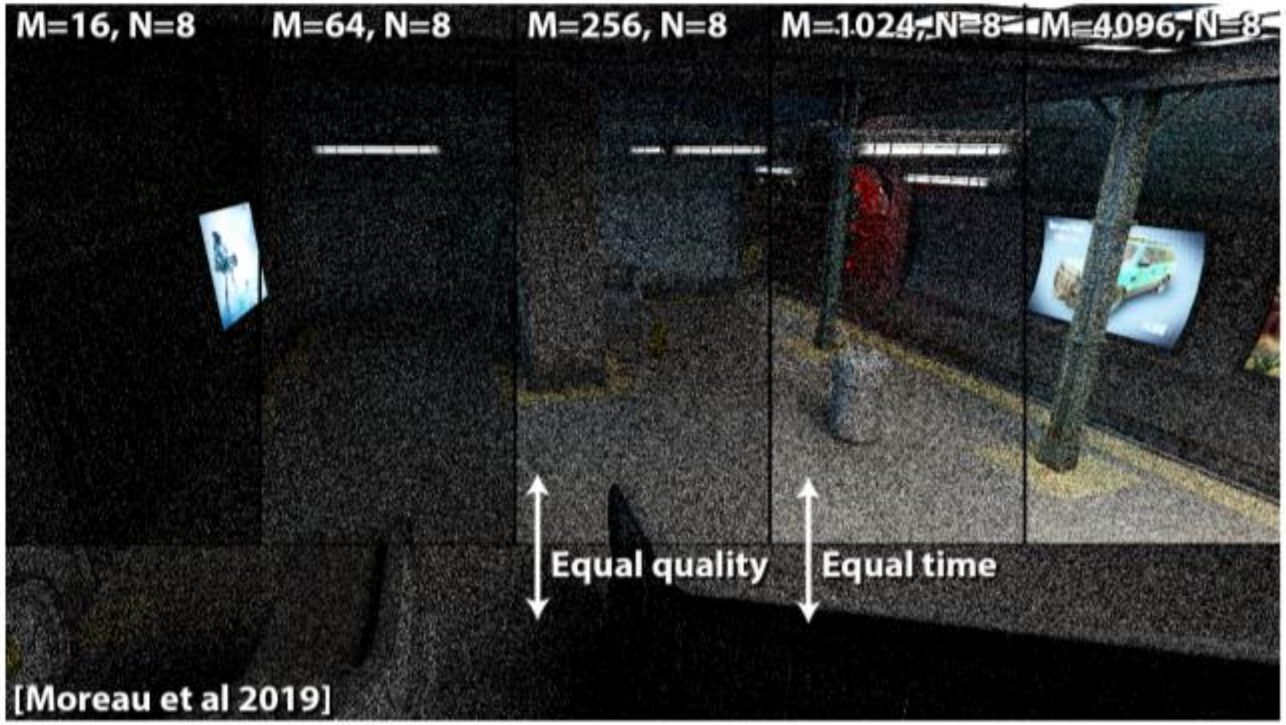


Algorithm 3: Streaming RIS using weighted reservoir sampling.

```
foreach pixel $q \in$ Image do
    Image[$q$] $\leftarrow$ shadePixel(RIS($q$), $q$)

function RIS($q$)
    Reservoir $r$
    for $i \leftarrow 1$ to $M$ do
        generate $x_i \sim p$
        $r$.update($x_i$, $\hat{p}_q(x_i)/p(x_i)$)
    $r.W = \frac{1}{\hat{p}_q(r.y)} \left( \frac{1}{r.M} r.\mathrm{w_{sum}} \right)$   // Equation (6)
    return $r$

function shadePixel(Reservoir $r$, $q$)
    return $f_q(r.y) \cdot r.W$
```



[Moreau et al 2019]

Bitterli, Benedikt, Chris Wyman, Matt Pharr, Peter Shirley, Aaron Lefohn, and Wojciech Jarosz. "Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting." *ACM Transactions on Graphics (TOG)* 39, no. 4 (2020): 148-1.

Spatiotemporal Reuse

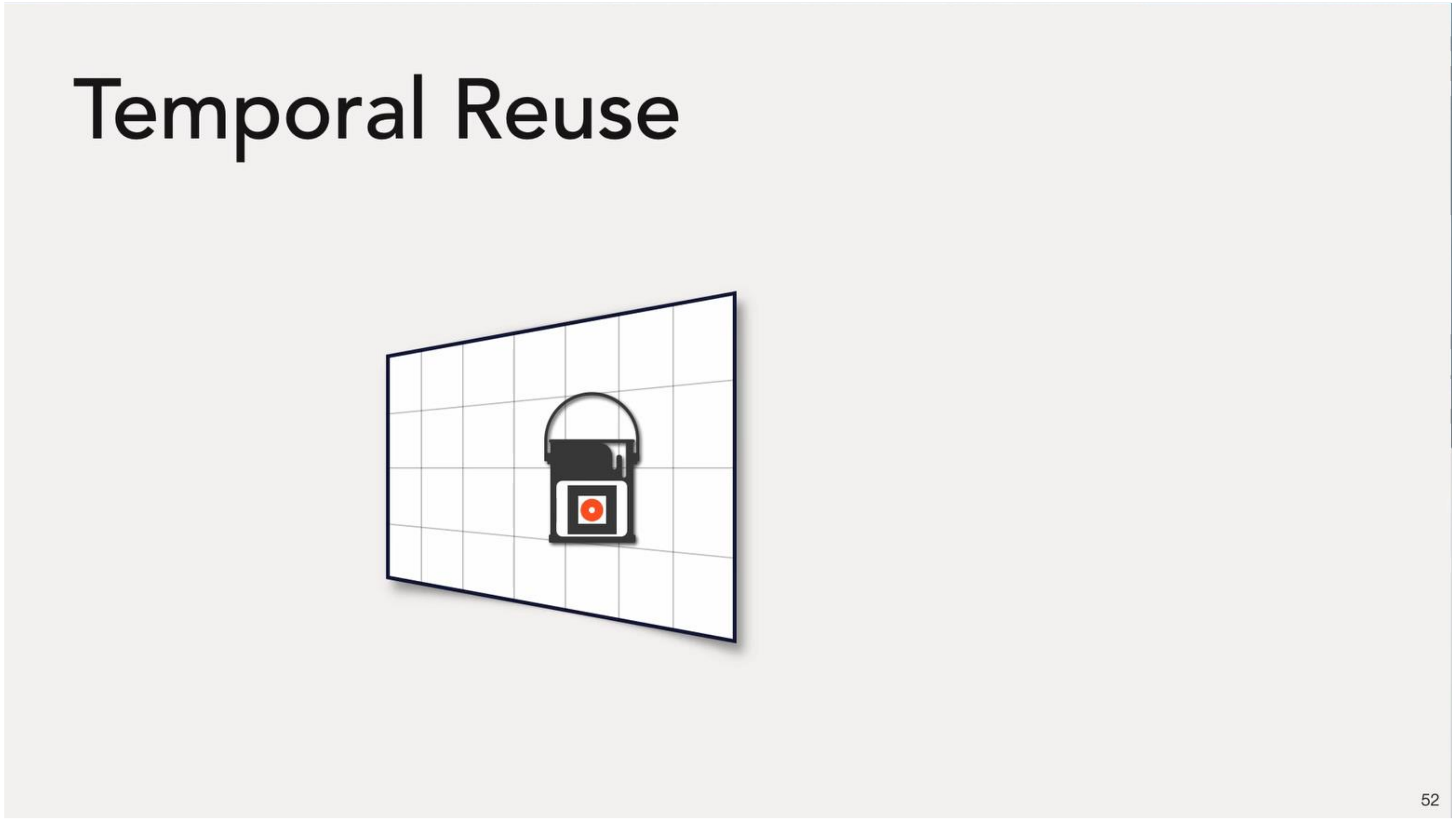
- Spatial Reuse

Le Petit Coin
MENU

Bitterli, Benedikt, Chris Wyman, Matt Pharr, Peter Shirley, Aaron Lefohn, and Wojciech Jarosz. "Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting." *ACM Transactions on Graphics (TOG)* 39, no. 4 (2020): 148-1.

Spatiotemporal Reuse

- Temporal Reuse



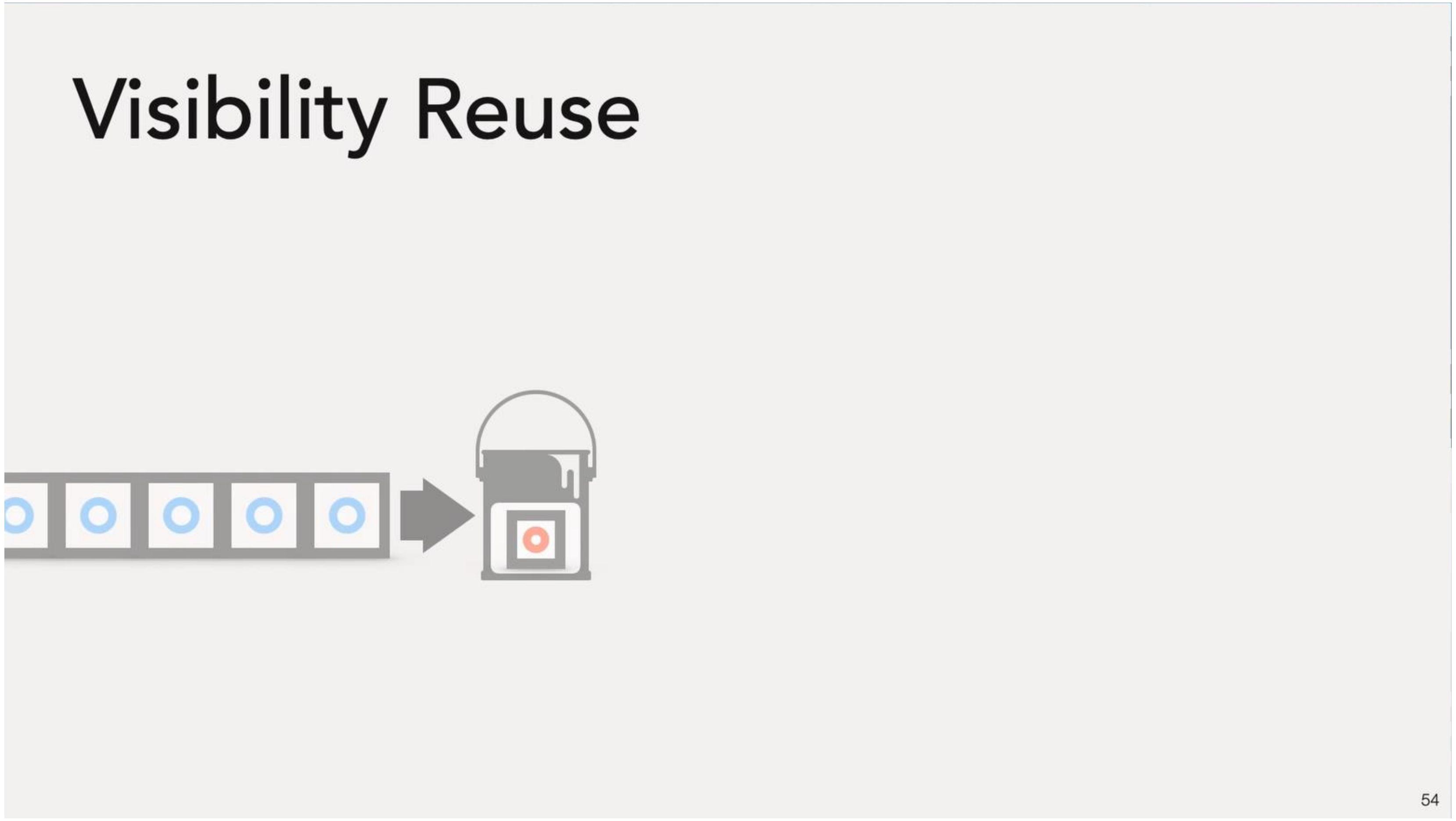
Temporal Reuse

52

Bitterli, Benedikt, Chris Wyman, Matt Pharr, Peter Shirley, Aaron Lefohn, and Wojciech Jarosz. "Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting." *ACM Transactions on Graphics (TOG)* 39, no. 4 (2020): 148-1.

Spatiotemporal Reuse

- Visibility Reuse

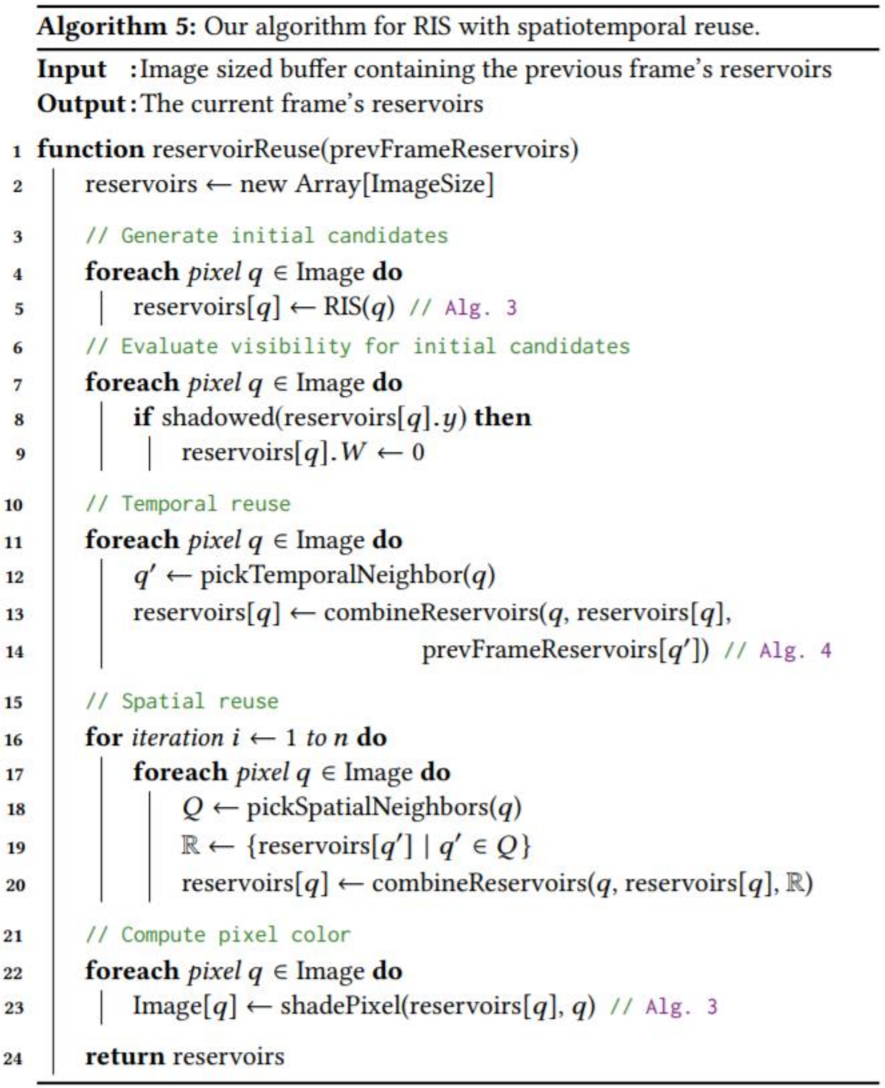


Visibility Reuse

54

Bitterli, Benedikt, Chris Wyman, Matt Pharr, Peter Shirley, Aaron Lefohn, and Wojciech Jarosz. "Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting." *ACM Transactions on Graphics (TOG)* 39, no. 4 (2020): 148-1.

Streaming RIS with Spatiotemporal Reuse

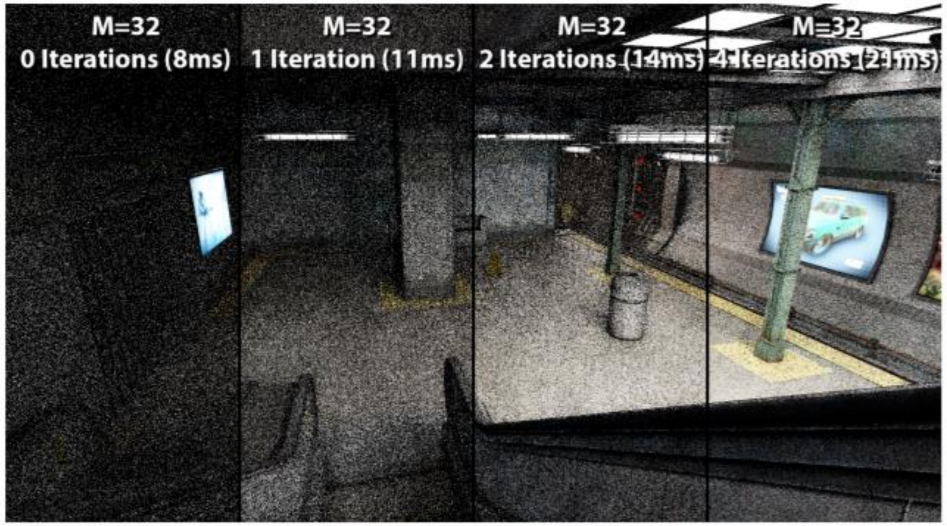


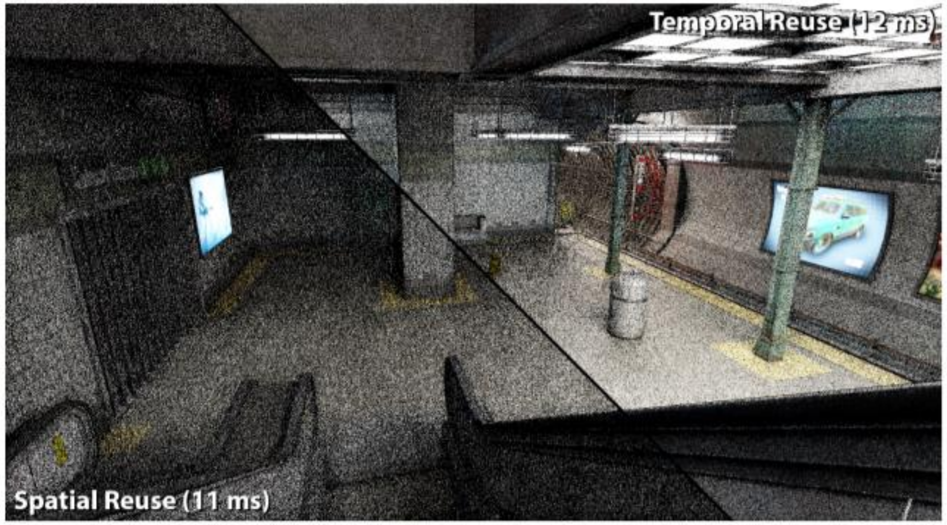
Algorithm 5: Our algorithm for RIS with spatiotemporal reuse.

Input : Image sized buffer containing the previous frame's reservoirs

Output : The current frame's reservoirs

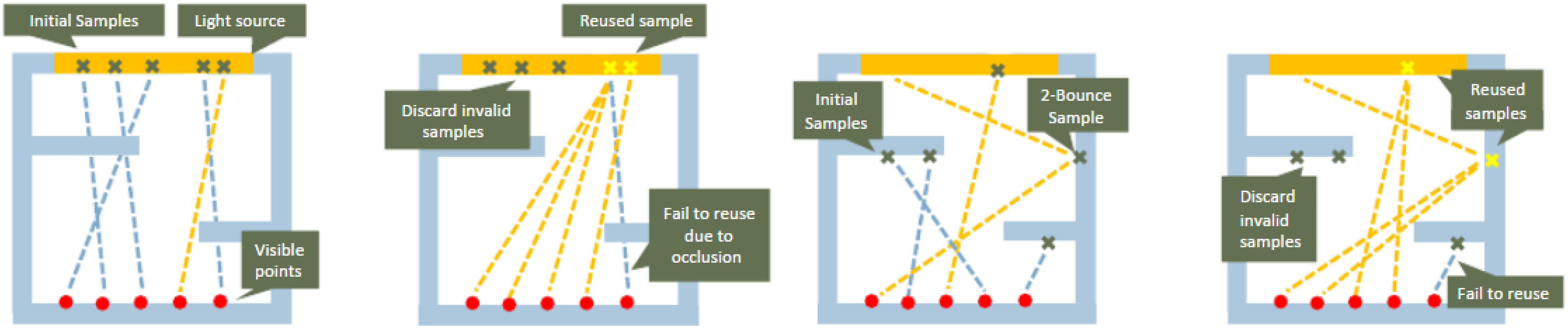
```
function reservoirReuse(prevFrameReservoirs)
    reservoirs ← new Array[ImageSize]
    // Generate initial candidates
    foreach pixel $q$ ∈ Image do
        reservoirs[$q$] ← RIS($q$) // Alg. 3
    // Evaluate visibility for initial candidates
    foreach pixel $q$ ∈ Image do
        if shadowed(reservoirs[$q$].$y$) then
            reservoirs[$q$].$W$ ← 0
    // Temporal reuse
    foreach pixel $q$ ∈ Image do
        $q'$ ← pickTemporalNeighbor($q$)
        reservoirs[$q$] ← combineReservoirs($q$, reservoirs[$q$],
                                  prevFrameReservoirs[$q'$]) // Alg. 4
    // Spatial reuse
    for iteration $i$ ← 1 to $n$ do
        foreach pixel $q$ ∈ Image do
            $Q$ ← pickSpatialNeighbors($q$)
            $\mathbb{R}$ ← {reservoirs[$q'$] | $q'$ ∈ $Q$}
            reservoirs[$q$] ← combineReservoirs($q$, reservoirs[$q$], $\mathbb{R}$)
    // Compute pixel color
    foreach pixel $q$ ∈ Image do
        Image[$q$] ← shadePixel(reservoirs[$q$], $q$) // Alg. 3
    return reservoirs
```





Bitterli, Benedikt, Chris Wyman, Matt Pharr, Peter Shirley, Aaron Lefohn, and Wojciech Jarosz. "Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting." *ACM Transactions on Graphics (TOG)* 39, no. 4 (2020): 148-1.

ReSTIR GI



Initial Sampling
(a)

Reuse Samples
(b)

Initial Sampling
(c)

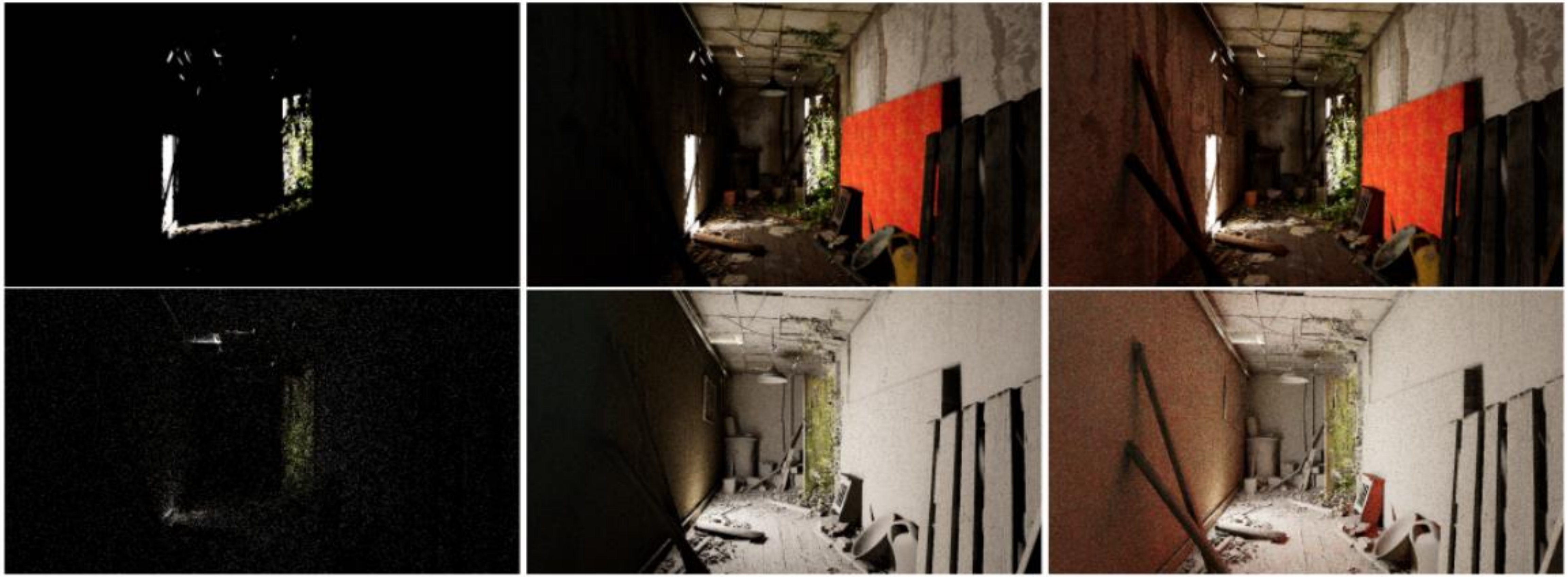
Reuse Samples
(d)

ReSTIR

ReSTIR GI

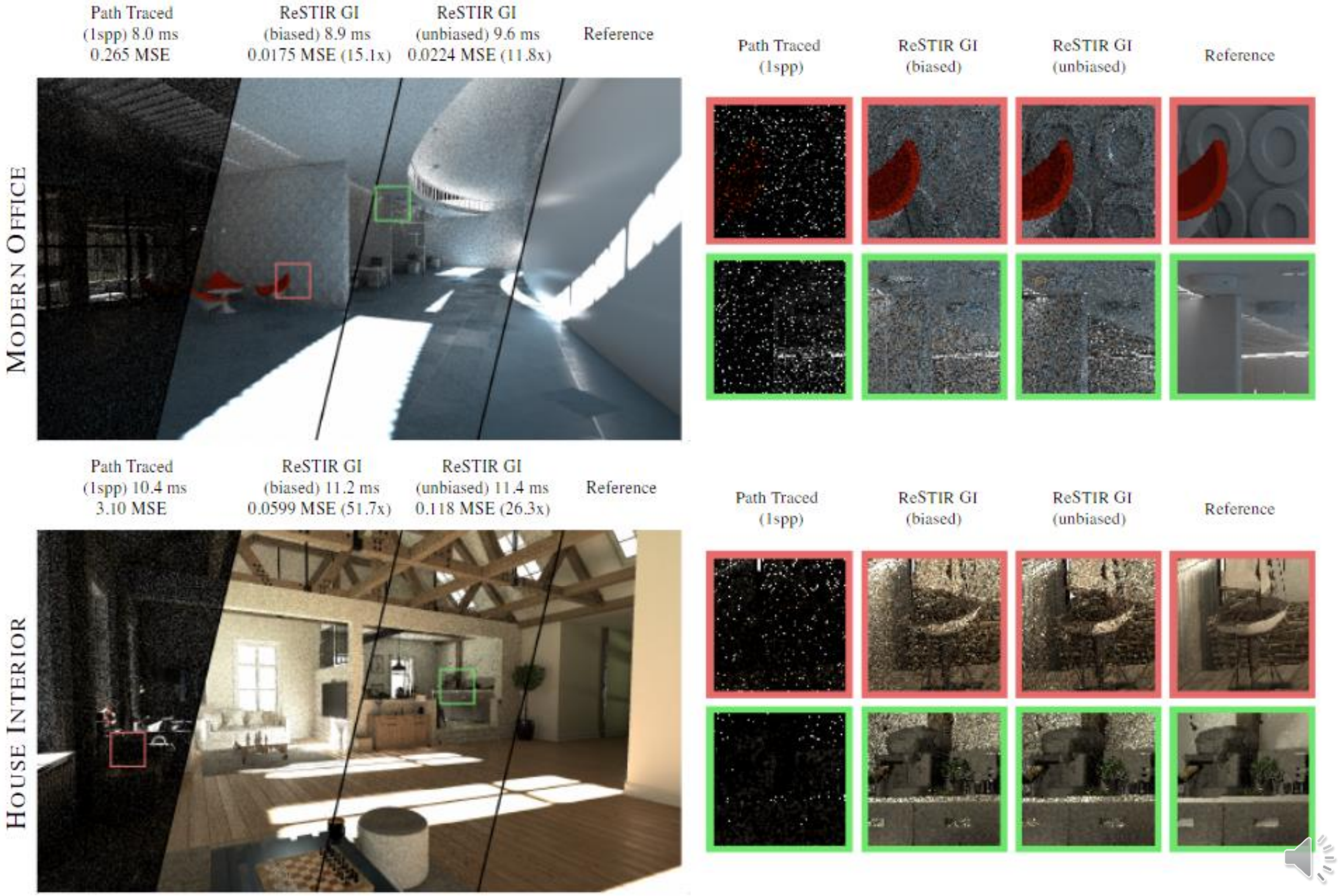
Results

- Multiple bounces



Results

- Comparisons to other methods


Path Traced (1spp) 8.0 ms 0.265 MSE
ReSTIR GI (biased) 8.9 ms 0.0175 MSE (15.1x)
ReSTIR GI (unbiased) 9.6 ms 0.0224 MSE (11.8x)
Reference
Path Traced (1spp)
ReSTIR GI (biased)
ReSTIR GI (unbiased)
Reference
Modern Office
Path Traced (1spp) 10.4 ms 3.10 MSE
ReSTIR GI (biased) 11.2 ms 0.0599 MSE (51.7x)
ReSTIR GI (unbiased) 11.4 ms 0.118 MSE (26.3x)
Reference
Path Traced (1spp)
ReSTIR GI (biased)
ReSTIR GI (unbiased)
Reference
House Interior

Conclusion and Limitations

- Fast and quality path tracing algorithm with global illumination up to 166 times clear than the previous one (wrt. MSE)
- Still slow in some applications
- With a sudden change of a scene, reused samples may be useless
- Spatial reuse is not so helpful with glossy surfaces
- Temporal reuse may cause artifacts at a denoising phase.

Conclusion and Limitations

- Fast and quality path tracing algorithm with global illumination up to 166 times clear than the previous one (wrt. MSE)
- Still slow in some applications
- With a sudden change of a scene, reused samples may be useless
- Spatial reuse is not so helpful with glossy surfaces
- Temporal reuse may cause artifacts at a denoising phase.

Thank you