Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting

Supplemental document

BENEDIKT BITTERLI, Dartmouth College
CHRIS WYMAN, NVIDIA
MATT PHARR, NVIDIA
PETER SHIRLEY, NVIDIA
AARON LEFOHN, NVIDIA
WOJCIECH JAROSZ, Dartmouth College

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

Additional Key Words and Phrases: Photorealistic rendering, resampled importance sampling, real-time rendering, reservoir sampling

ACM Reference Format:
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: Supplemental document. *ACM Trans. Graph.* 39, 4, Article 1 (July 2020), 2 pages. https://doi.org/10.1145/3386569.3392481

1 DISCUSSION OF TALBOT MIS

Both our work and the work of Talbot et al. propose ways to combine multiple sampling techniques in RIS using MIS. In the main paper, we discuss the differences between the two approaches from a theoretical perspective in Section 4.3. Here, we expand on this discussion and show empirical results demonstrating the benefit of our approach to MIS over that of Talbot et al. for the purpose of spatial or temporal sample reuse through RIS.

The need for MIS arises when reusing samples from neighboring pixels. In our rendering algorithm, spatial (Alg 5 main paper, lines 16-20) and temporal reuse (lines 11-14) occur after each pixel has already performed RIS (lines 4-5) to obtain its initial set of samples. Samples produced by RIS follow a PDF that is some (unknown) mixture of the candidate PDF p and the target distribution $\hat{p}_q$. Because the target distribution varies from pixel to pixel, the sample(s) stored in the reservoir follow a different PDF at each pixel after one or more uses of RIS. Combining (and reusing) samples from different pixels and therefore different PDFs can lead to variance, and this can be avoided when using MIS.

Authors' addresses: Benedikt Bitterli, Dartmouth College, benedikt.m.bitterli.gr@dartmouth.edu; Chris Wyman, NVIDIA, chris.wyman@acm.org; Matt Pharr, NVIDIA, matt.pharr@gmail.com; Peter Shirley, NVIDIA, ptrshrl@gmail.com; Aaron Lefohn, NVIDIA, 2788 San Tomas Expressway, Santa Clara, CA, 95051, alefohn@nvidia.com; Wojciech Jarosz, Dartmouth College, Department of Computer Science, 9 Maynard St. Hanover, NH, 03755, wojciech.k.jarosz@dartmouth.edu.

0730-0301/2020/7-ART1 $15.00
https://doi.org/10.1145/3386569.3392481

To compute its weighting factors, MIS requires knowing the PDF (or an approximation to it) of the input samples. We do not know the exact PDF of samples produced by RIS, and need to approximate their PDF instead. It's important to point out that this approximate PDF is only needed for computing the MIS weights, and the pixel estimator is still unbiased (as its PDF is computed differently). The approximate PDFs only affect the MIS weights, which may no longer be optimal if the approximation poor. For simplicity, we simply assume a sample x produced by RIS at pixel q follows PDF $\hat{p}_q(x)$. We additionally check if x is occluded at q, and set the PDF to zero in that case; pixel q would have discarded the sample otherwise (lines 7-9 in Alg. 5, main paper). This approximation is not without issues: For one, $\hat{p}$ is not normalized, which can skew MIS weights if the integral of $\hat{p}$ varies strongly between pixels. Additionally, this approximation is only reasonable if the number of candidates M is large enough such that samples are close to the target distribution. While this approximation worked well enough for our purposes, finding a better approximate PDF for MIS purposes is a fruitful direction for future work.

Similar to Alg. 6 in the main paper, we show our algorithm for RIS with MIS in Alg. 1; code relevant to MIS has been marked in blue. It proceeds similar to standard RIS, but uses a modified weighting term that uses the MIS weight m instead of the fraction of $1/M$ in standard RIS (line 9). Computing the MIS weight (6-9) requires a second loop over the k input pixels, and an equal number of evaluations of the approximate PDF (which includes tracing a shadow ray). In Alg. 2, we show the equivalent of applying the MIS of Talbot et al. to reservoir reuse. Instead of modifying the final weighting factor, Talbot et al. modify the selection weights of each sample by multiplying it with the MIS weight (line 4-7). Because there are as many sampling techniques as there are samples, this requires iterating k times for each of the k samples—a quadratic cost. This is especially costly, as evaluating the approximate PDF involves tracing a shadow ray.

1.1 Results

We implemented the MIS of Talbot et al. in our framework and ran similar empirical measurements as in the main paper. In Figure 1, we show how the RMAE evolves with increased render time for

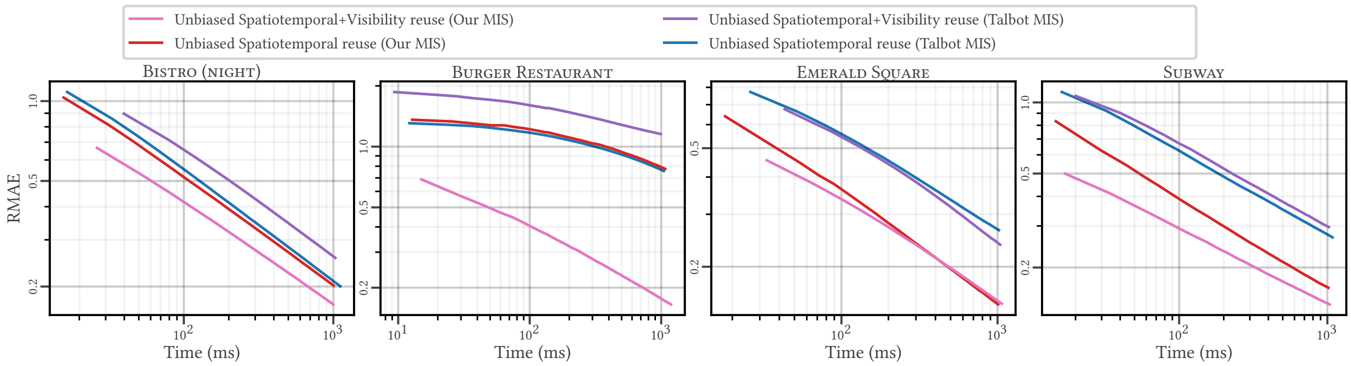

Fig. 1. The evolution of error (relative mean absolute error) in our scenes over render time. We compare unbiased forms of spatiotemporal reuse, with and without visibility reuse, with our proposed version of MIS and Talbot et al.'s version of MIS. When visibility reuse is enabled, Talbot et al.'s form of MIS is dramatically worse in terms of error over time, as the number of shadow rays it needs to trace increases quadratically as the number of neighboring pixels increases. Without visibility reuse, no shadow rays need to be traced when computing MIS weights. While the MIS of Talbot et al. may decrease noise slightly, it does not offset its additional cost (even without tracing additional rays).

Algorithm 1: Reservoir reuse using our proposed MIS.

Input : Reservoirs r_i and the pixels q_i they originated from.
Output: An unbiased combination of the input reservoirs.

function combineReservoirsMIS($q, r_1, r_2, \ldots, r_k, q_1, \ldots, q_k$)
  Reservoir s
  foreach $r \in \{r_1, \ldots, r_k\}$ **do**
    s.update($r.y,\ \hat{p}_q(r.y) \cdot r.W \cdot r.M$)
  $s.M \leftarrow r_1.M + r_2.M + \ldots + r_k.M$
  $p_{\text{sum}} \leftarrow 0$
  foreach $q_i \in \{q_1, \ldots, q_k\}$ **do**
    $p_{\text{sum}} \leftarrow p_{\text{sum}} + \hat{p}_{q_i}(s.y)$
  $m \leftarrow \frac{\hat{p}_{q_*}(s.y)}{p_{\text{sum}}}$ // q_*: Pixel that contributed $s.y$
  $s.W = \frac{1}{\hat{p}_q(s.y)}\left(m \cdot s.\text{w}_{\text{sum}}\right)$
  return s

Algorithm 2: Reservoir reuse, with MIS proposed by Talbot et al.

function combineReservoirsTalbotMIS($q, r_1, r_2, \ldots, r_k, q_1, \ldots, q_k$)
  Reservoir s
  foreach $r_i \in \{r_1, \ldots, r_k\}$ **do**
    $p_{\text{sum}} \leftarrow 0$
    foreach $q_j \in \{q_1, \ldots, q_k\}$ **do**
      $p_{\text{sum}} \leftarrow p_{\text{sum}} + \hat{p}_{q_j}(r_i.y)$
    $m \leftarrow \frac{\hat{p}_{q_i}(r_i.y)}{p_{\text{sum}}/k}$
    s.update($r_i.y,\ \hat{p}_q(r_i.y) \cdot r_i.W \cdot r_i.M \cdot m$)
  $s.M \leftarrow r_1.M + r_2.M + \ldots + r_k.M$
  $s.W = \frac{1}{\hat{p}_q(s.y)}\left(\frac{1}{s.M} \cdot s.\text{w}_{\text{sum}}\right)$
  return s

four different methods. We compare unbiased spatiotemporal reuse with and without visibility reuse, and combined with our proposed version of MIS or Talbot et al.'s version of MIS. When visibility reuse is enabled, Talbot et al.'s form of MIS is dramatically worse in terms of error over time. This is because the number of shadow rays it needs to trace increases quadratically as the number of neighboring pixels increases. Even with a low number of neighbors used ($k = 5$ in these experiments), the cost is pronounced. In addition, we also show plots without visibility reuse; in this case, no shadow rays need to be traced when evaluating MIS weights. This makes the MIS of Talbot et al. slightly more competitive; however, while it may decrease noise slightly compared to our MIS, this does not offset its additional cost (even without tracing additional rays). Even in this scenario, our version of MIS is as good or better as that of Talbot et al.

REFERENCES

Justin F. Talbot, David Cline, and Parris Egbert. 2005. Importance Resampling for Global Illumination. In *Proc. EGSR*. Eurographics Association, 139–146. https://doi.org/10/gfzsm2