SMARTER SAMPLING: ENHANCING RESTIR WITH NORMAL-AWARE RESERVOIR CACHING

We improve ReSTIR's early-frame quality by caching reservoirs in a normal-aware hash grid. This reduces noise and enhances lighting stability with moderate overhead. While effective in structured scenes, the method faces challenges with artifacts and memory in complex environments, highlighting areas for future optimization.

AUTHORS

Author: Vlad-Tudor Stefanescu (vstefanescu@tudelft.nl)
Responsible Professors: Michael Weinmann, Elmar Eisemann
Supervisor: Christoph Peters
Examiner: George Smaragdakis

AFFILIATIONS

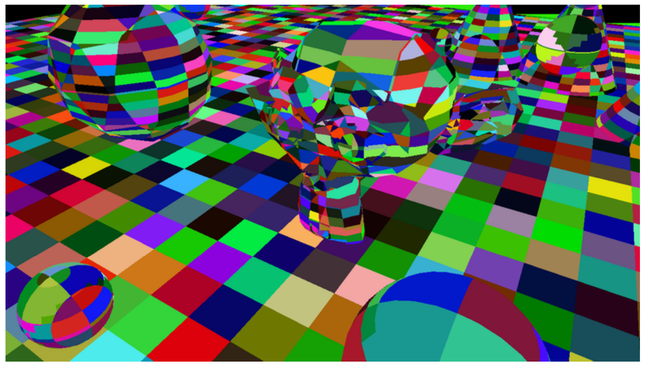
TU Delft

INTRODUCTION

Ray tracing produces photorealistic images by simulating light in 3D scenes but has long been too computationally demanding for real-time use. Advances in hardware and algorithms, notably ReSTIR [1], have improved its practicality by reusing lighting samples across space and time via reservoir sampling. However, ReSTIR suffers from high visual noise in early frames. We propose a modification that caches samples in a normal-aware hash grid of reservoirs, enabling access to higher-quality data from the start and potentially reducing noise with a 20–23% frame time overhead. This paper examines the impact of reservoir caching using a normal aware hash grid on image quality and performance in real-time ray-traced direct illumination.

METHODOLOGY

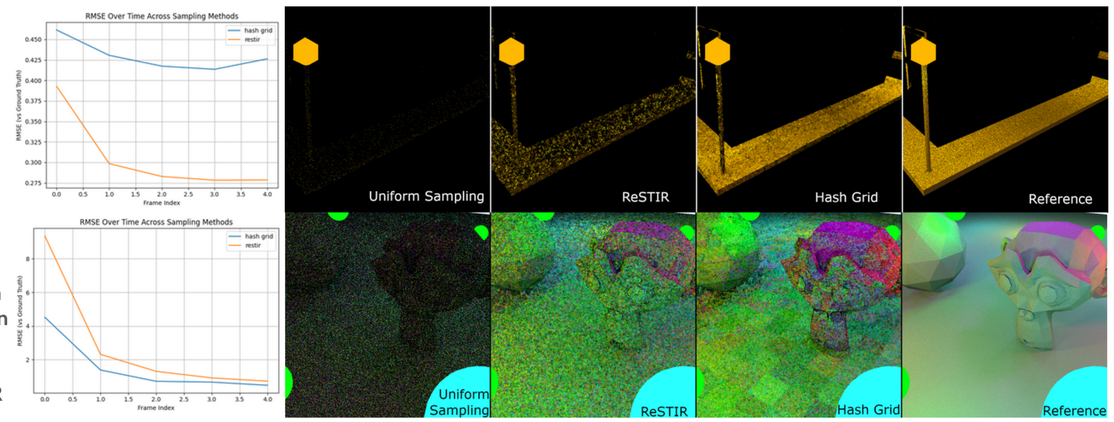
We implemented a CPU-based version of ReSTIR and extended it with a mechanism for caching precomputed reservoirs. This is achieved by discretizing the scene into a 3D grid, further subdividing each cell based on surface normals. The resulting spatial-directional cells are stored in a hash map, as illustrated in the figure below, each color representing a unique cell. These cells hold multiple reservoirs populated during a precomputation phase. During rendering, reservoirs from cells that closely match the position and normal of the shading point are merged into ReSTIR's screen-space reservoirs. We evaluate this approach against baseline ReSTIR in terms of image quality, visual appearance, memory usage, and frame time.



Normal aware hash grid cells, represented by different colors.

RESULTS

In the City Grid scene, the Hash Grid method visibly reduces noise and improves lighting stability compared to Uniform Sampling and base ReSTIR, especially in the first frame. Surfaces like sidewalks and lamp posts appear cleaner and more coherent, closely matching the reference render.
In the Monkeys scene, while the Hash Grid enhances geometry clarity over other methods, it introduces noticeable grid-like artifacts, especially on flat surfaces. These artifacts reduce realism and likely result from resolution limits in the grid structure.
RMSE analysis confirms these trends: Hash Grid converges faster than ReSTIR in early City Grid frames but offers less long-term advantage. In the Monkeys scene, ReSTIR maintains lower RMSE, aligning with the visual artifact issues seen in Hash Grid results.



Rendered results from the first frames of four methods, along with RMSE graphs over the initial five frames. The top row shows the City Grid scene; the bottom row shows the Monkeys scene.

DISCUSSION

Our results show that incorporating a normal-aware hash grid into ReSTIR improves early-frame quality by providing better initial samples. This enhancement leads to cleaner and more stable illumination, particularly in dense scenes like the City Grid, where early RMSE values are significantly lower.
However, this improvement comes with trade-offs. In the Monkeys scene, grid artifacts appear, revealing limitations in the spatial resolution and binning strategy. Additionally, the hash grid introduces substantial memory overhead in complex scenes and increases frame times by around 20–23%. These costs highlight the need for further optimization, both in grid structure and hash performance, to make the method more broadly practical.

CONCLUSIONS

This work shows that a normal-aware hash grid can enhance ReSTIR's direct illumination by improving initial sample quality, reducing early-frame noise, and increasing visual stability—especially in structured scenes. However, artifacts in complex scenes, memory overhead, and higher frame times reveal key limitations.
Future work will focus on dynamic updates to the grid during rendering for temporal reuse, optimizing hash functions, and exploring adaptive grids or octrees to reduce artifacts. Further improvements could also come from a GPU implementation, reservoir memory layout optimizations and better handling of camera motion to improve temporal stability.

RELATED LITERATURE

[1] B. Bitterli, C. Wyman, M. Pharr, P. Shirley, A. Lefohn, and W. Jarosz, 'Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting', ACM Trans. Graph., vol. 39, no. 4, Aug. 2020.