A Better Light Candidate Generation Algorithm for ReSTIR Ray Tracing Using an Acceleration Structure to Identify Relevant Lights

Rafayel Gardishyan
R.V.Gardishyan@student.tudelft.nl

Christoph Peters
C.J.P.Peters@tudelft.nl

Michael Weinmann
M.Weinmann@tudelft.nl

Elmar Eisemann
E.Eisemann@tudelft.nl

TU Delft

1 Background

$$L_o(y,\omega_0) = \int_\Omega f_r(y,\omega_i \rightarrow \omega_0) L_i(y,\omega_i)\cos\theta_i \, d\omega_i \quad (1)$$

The rendering equation, as seen in Equation (1), is the bottleneck of real-time ray tracing. Computing the integral, is a hard task, which is why we employ Monte Carlo techniques to estimate the final image sample by sample. This is illustrated in Fig. 1a.

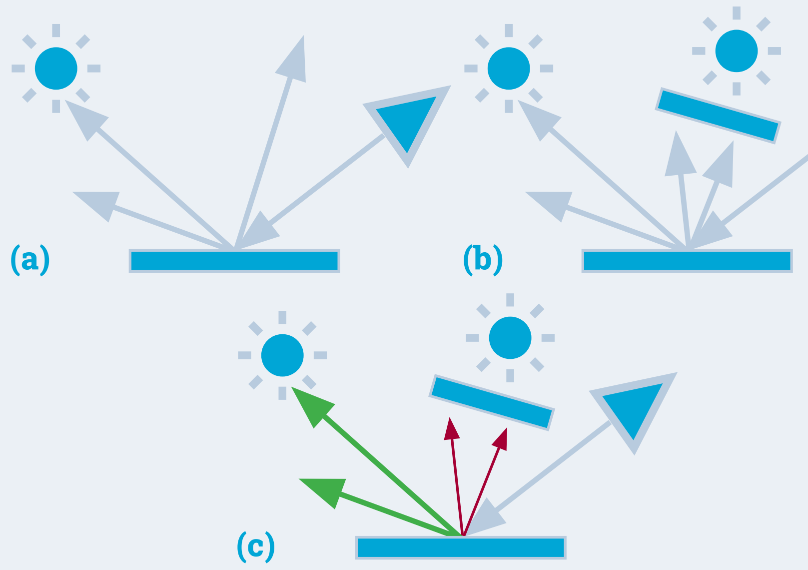

Fig. 1. Different sampling techniques. (a) Simple Monte Carlo, (b) Importance Sampling, (c) Resampled Importance Sampling

In complex scenes, with many light sources, sampling uniform directions leads to a slow convergence time, and thus a high computational cost, which is undesirable in real-time constraints.

To address this, multiple techniques, building on each other have been introduced:

- Importance Sampling: Samples in directions of light sources.
- Resampled Importance Sampling (RIS): Sample from an easy to sample distribution and reweight the samples.
- ReSTIR [Bitterli et al. 2020]: applies reservoir sampling [Wyman 2021] to RIS and applies spatial and remporal reuse to multiply the amount of samples considered.

This process is illustrated in Fig. 2.

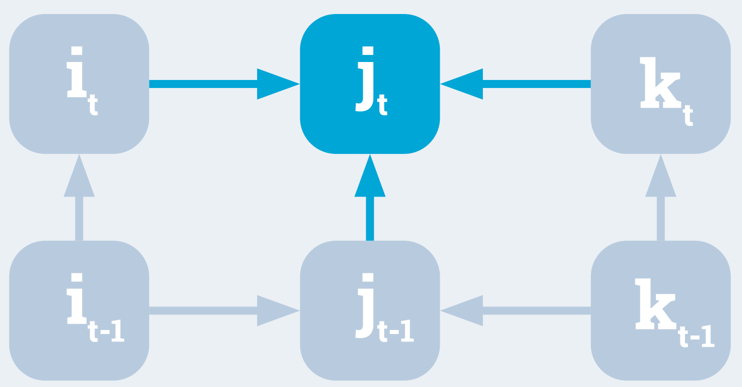

Fig. 2. Reuse of samples in ReSTIR. A pixel j at time t receives the samples from its neighbours and itself, from time t-1.

2 Problem Statement

While ReSTIR is very effective in scenes with multiple lights, it starts to lose it's effectiveness when there are thousands of emmissive triangles illuminating the scene, especially when the light sources have a very local volume of influence.

A contributor to this issue is the candidate generation mechanism of ReSTIR, which is often chosen to be a uniform random distribution over all emitters in the scene. This results in light sources, which have no influence for a certain shading point having a similar chance to be selected as the light sources contributing the most.

To address this, we propose integrating an acceleration structure, like a bounding-box volume hierarchy (BVH) in the candidate selection process in order to speed up convergence.

3 Proposed Algorithm

We propose building a BVH containing the volumes of influence of all light emitters. During the initial candidate generation of ReSTIR, we select light sources not uniformly by random, but by first performing a point query to identify the light sources that are relevant for the specific shading point (Illustrated in Fig. 3). Afterwards, we select one light source, uniformly at random, from the identified relevant emitters.

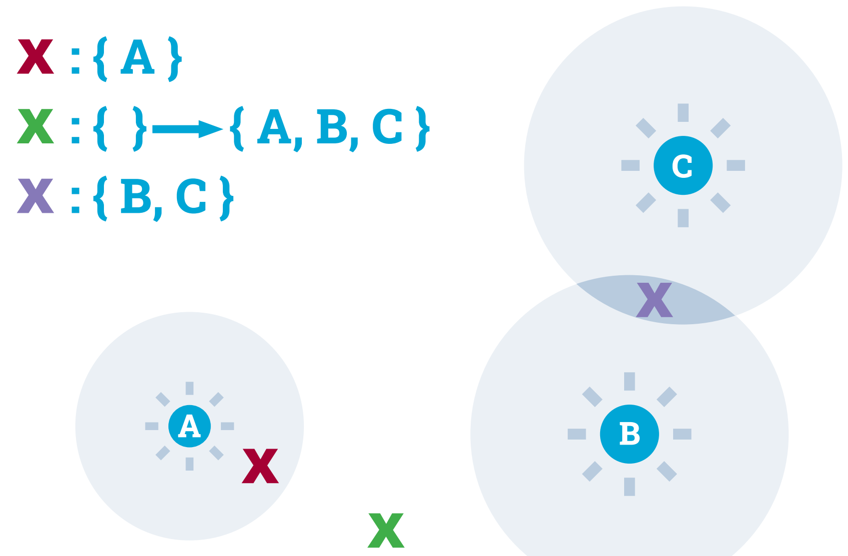

Fig. 3. Illustration of the workings of our algorithm. For each shading point X, it is shown which light sources are considered. The green shading point does not intersect any light sources influence area, thus we fall back on uniform random over all light sources.

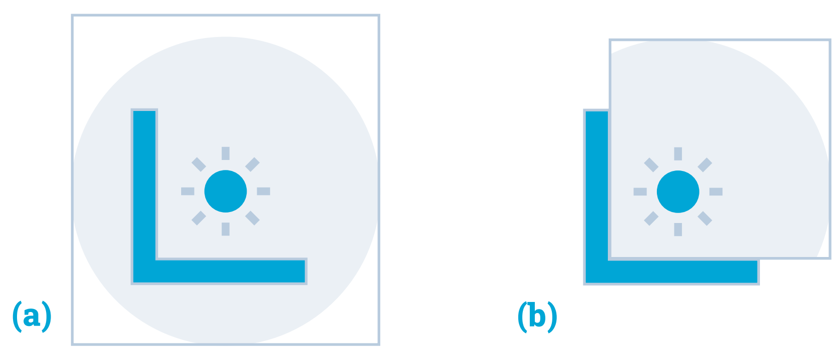

Fig. 4. (a) Simple bounding box construction strategy. (b) Visibility-aware bounding box construction strategy.

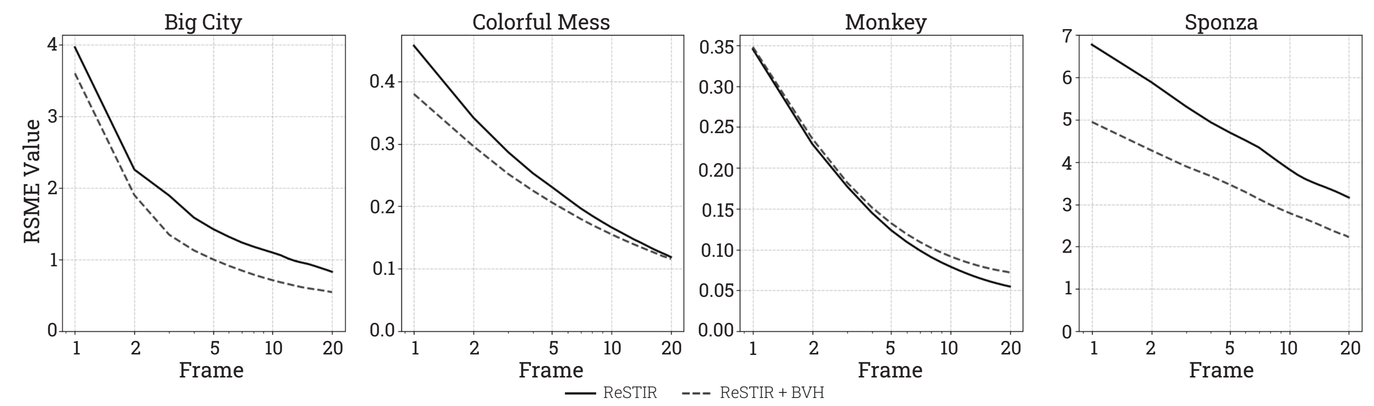

Fig. 5. Evolution of error (RMSE) in our scenes over frames rendered (accumulated). We compare ReSTIR [Bitterli et al. 2020] and our approach for 20 frames. Rendered with M = 32, k = 8 neighbours for spatial reuse in a radius of 5.

In order to build a BVH that is fit better to the scene, we experimented with visibility-aware bounding box construction. To this end, we would shoot rays in random directions from the light source, and record the farthest distance travelled in each of the axes of the bounding box. We would then adjust the the bounding box values to be `min(previous, ray_hit)`. This is illustrated in Fig. 4.

This strategy did not improve the performance of our technique, but would add additional cost to the BVH construction time. For this reason, we have decided to only compare the simple BVH construction strategy to uniform candidate generation.

4 Experimental Setup

We rendered 4 scenes:

- Night Cityscape - Complex; Many Lights
- Sponza (©crytek) - Complex, Few Lights
- Colorful Mess - Simple; Many Lights
- Monkey (©Blender) - Simple; Few Lights

Each scene was rendered, using RIS, ReSTIR and our method, for 20 frames. Error is measured using Root Mean Squared Error (RMSE).

BVH construction and render times were recorded as well.

The rendering engine used was custom built uing the following tooling:

- C++
- OpenMP (paralellisation)
- Intel® Embree (ray intersections)

The implementation of ReSTIR was built acoording to the paper by [Bitterli et al. 2020].

5 Results

We have found that, with the parameters of the BVH tuned correctly:

- BVH based candidate generation results in faster convergence (Fig. 5).
- In scenes with a few light sources, the method performs similarly to ReSTIR (Fig. 5).
- Intersecting the BVH introduces a small overhead (Fig. 6).

In Fig. 7 it is apparent, that our method results in a better visual fidelity from the first frame on.

6 Limitations

Below are some limitations and suggestions for future research.

- BVH Construction overhead (Can be mitigated partially by a 2-level acceleration structure)
- Unoptimised renderer, thus worse quality results
- Parameter optimisation

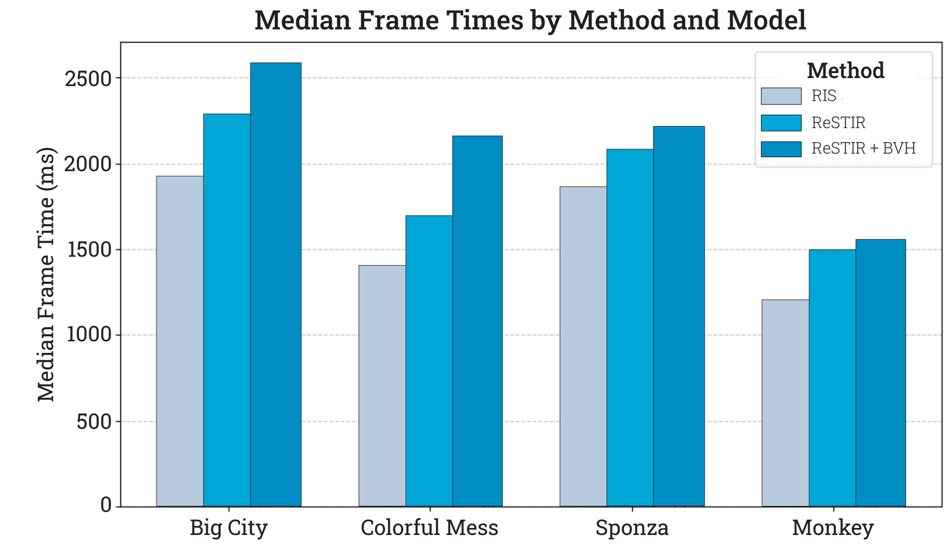

Fig. 6. Median frame times across different scenes. N=20 for each method. Frames are rendered with a resolution of 1920 x 1080 pixels. M=32, k=8 neighbours for spatial reuse in a radius of 5, 1 sample per pixel.

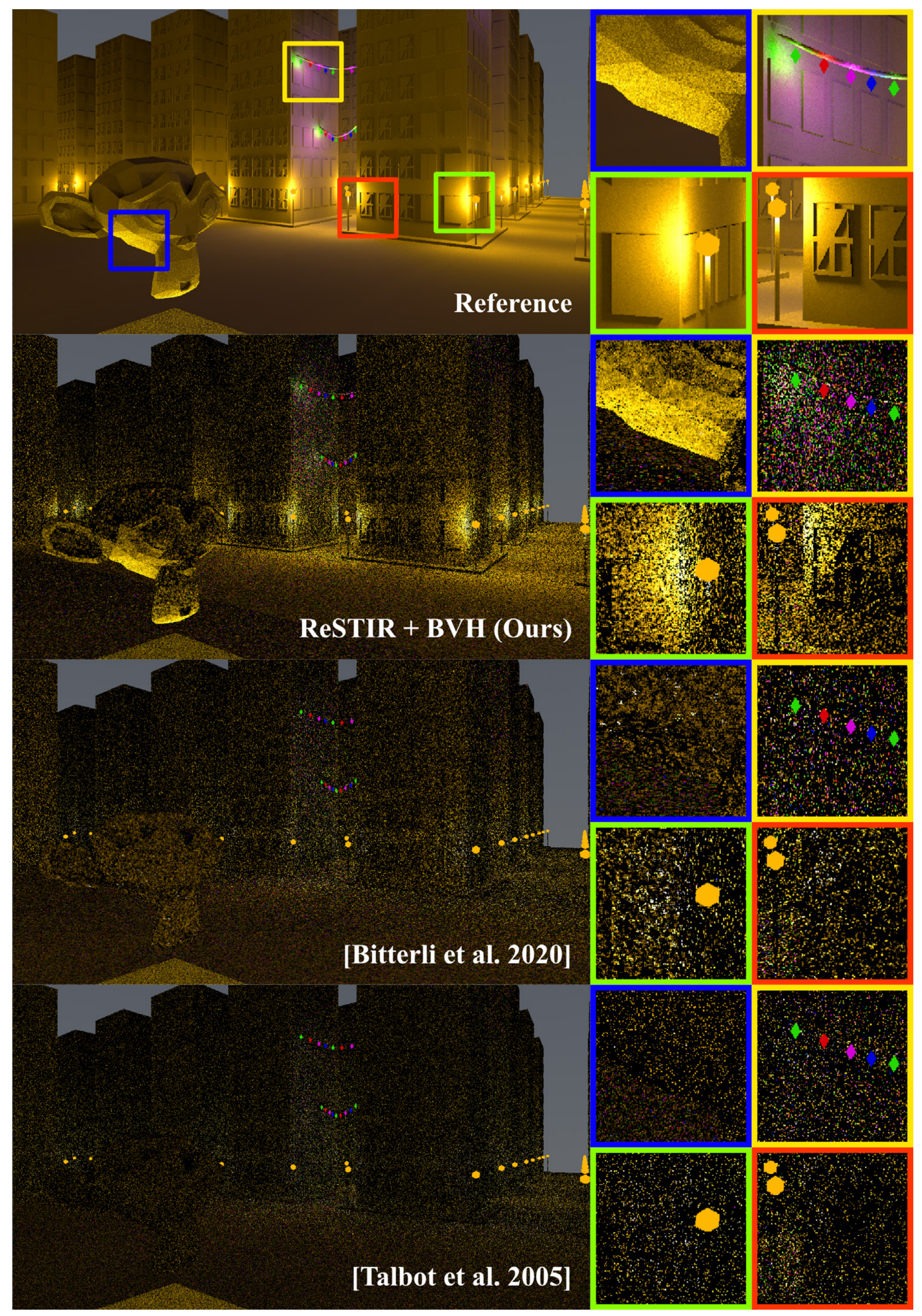

Fig. 7. The Night Cityscape scene. First frame comparison. Rendered with M = 32, k = 8 neighbours for spatial reuse in a radius of 5, 1 sample per pixel.

References

Bitterli, B. et al. 2020. Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. DOI: 10.1145/3386569.3392481

Wyman, C. 2021. Weighted reservoir sampling: randomly sampling streams. DOI: 10.1007/978-1-4842-7185-8_22