IN4310 Seminar Computer Graphics ReSTIR - Implementation Report

William Narchi (5046122)
w.narchi-1@student.tudelft.nl
Delft University of Technology
Delft, The Netherlands

1 Motivation and Overview

The provided implementation consists of a simplified implementation of the **Reservoir-based Spatio-temporal Importance Resampling (ReSTIR)** algorithm built on top of the software ray tracer final project for the course CSE2215 Computer Graphics, wherein all adjustable parameters - as described in the paper - are adjustable. This implementation serves to showcase how the ReSTIR algorithm operates and the effects which these various parameters exhibit.

1.1 Ray Tracer Specifics

The base ray tracer is mostly unchanged from the one used for the CSE2215 final project. It makes use of a Phong shading model with diffuse and specular components but with the addition of a quadratic falloff (i.e. a geometry term containing inverse squared distance).

Lights are represented as distinct entities and can be either points, line segments, or quadrilaterals, though only quadrilaterals are used in the test scene. Line segments and quadrilaterals are sampled uniformly across their length/surface area.

1.2 Parameter Overview

The components of the algorithm that can be adjusted or toggled can be seen in fig. 1. The functions of these parameters are described in the paper, the paper presentation, and the implementation presentation and so are omitted for brevity.

Figure 1. Configurable parameters of the ReSTIR algorithm.

, ,
.

1.3 Test Scene Design

In order to make up for the large render times that would be seen with most scenes, a test scene was designed in order to showcase the ReSTIR light sampling logic while keeping render times reasonable. This test scene is based on the eponymous Cornell Box and is affectionately dubbed the 'Cornell Nightclub' [1]. The scene is effectively a larger Cornell Box with several more boxes in the scene. A render of this scene can be seen in fig. 2.

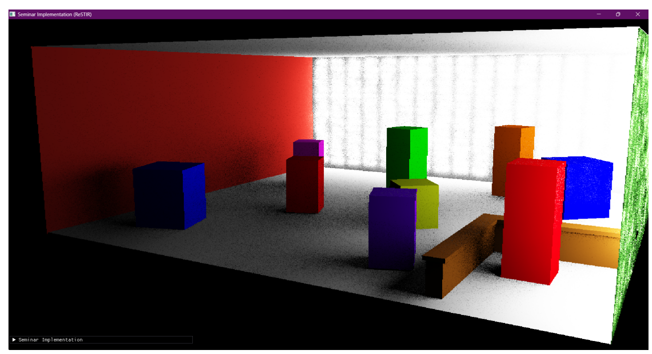


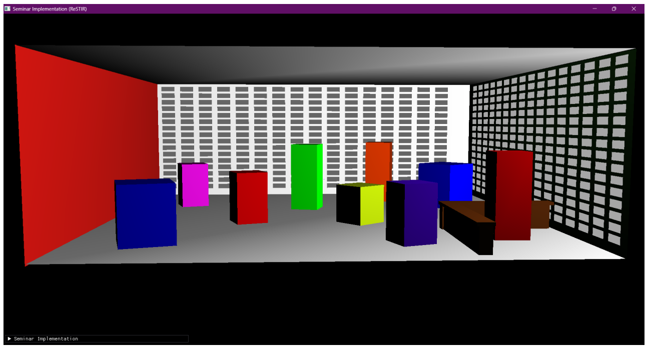
Figure 2. A rendering of the Cornell Nightclub test scene made with the provided implementation.

Where the test scene differs most is in the lighting arrangement as shown in fig. 3. The original Cornell Box features a single area light at the top of the box, whereas the Cornell Nightclub features 512 uniformly arranged rectangular area lights. The back wall contains 256 of them and each one has as color $(0.4, 0.4, 0.4)$; the right wall contains the other 256 of them and each one has as color $(0.65, 0.65, 0.65)$.

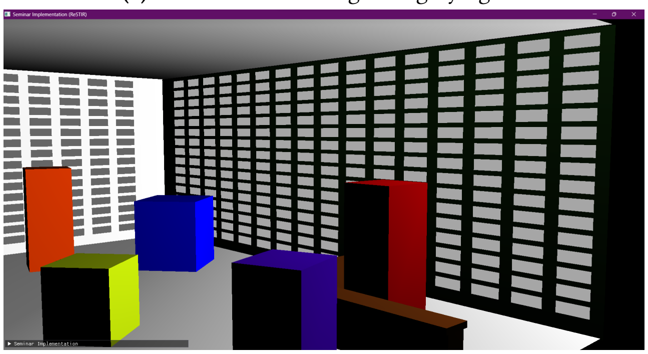
2 Limitations

The provided implementation has a few key limitations compared to the algorithm described in the original paper:

- **No consideration of motion vectors**
  In order to map from pixels in the current frame to their corresponding temporal predecessors in the prior frame, motion vectors are utilised in the original algorithm. This is not done in the provided implementation; the pixel at the exact same location (i.e. having the same screen-space coordinates) in the prior frame is resampled for temporal resampling. This does not



(a) Back wall containing dark grey lights.



(b) Right wall containing light grey lights.

Figure 3. Rasterized view of the Cornell Nightclub scene showcasing lighting arrangement.

impact demonstrative power much due to the implementation's framerates not being interactive.

- **Even sampling of all light sources**
In order to select initial light samples, the original algorithm selects lights with a probability proportional to their emissivity; this acts as a reasonably good heuristic for initial candidates and is accomplished via alias tables [2].[1] The provided implementation instead samples all light sources in the scene evenly in order to simplify light selection without incurring performance overhead; this is alleviated by the choice of test scene, as the lights are spatially distributed in a fairly regular manner and are of similar intensities.
- **Spatial neighbourhood shape**
The original paper selects spatial neighbours in a circle centered on the canonical pixel with an adjustable radius. The provided implementation instead selects neighbours in a square centered on the canonical pixel with an adjustable side length.

[1] More specifically, the choice of light is done in this method and then the point that is sampled on the light is uniformly chosen. Further, if an environment map is provided, 25% of all light samples are importance sampled from this map; the provided implementation does not make use of environment-based lighting and does not carry out this step.

3 Noteworthy Observations

Throughout testing, two noteworthy observations were made regarding parameters whose effects were not showcased in the original paper. The first of which is regarding the value of M_{clamp}, which limits the strength of temporal samples and which is set to 20 in the original paper. Figure 4 showcases how extreme values of this parameter influence results. A higher value encourages storing samples which have survived several iterations. In this scene, this leads to quicker convergence, though in more dynamic scenarios involving camera, object, and light source movement, this would discourage new, potentially more valuable samples in favour of stale temporal samples.

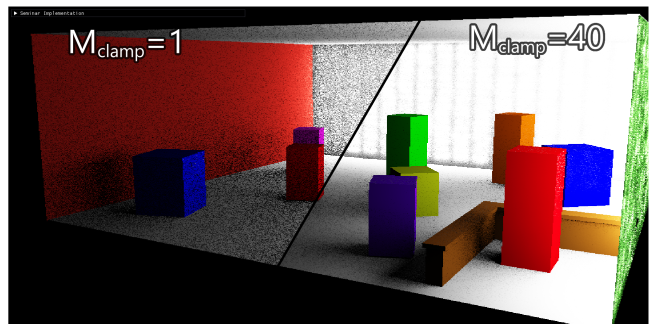


Figure 4. Comparison of temporal M_{clamp} for values 1 and 40 after 10 iterations and without spatial resampling.

The second is regarding the size of the neighbourhood in which spatial neighbours are resampled. Figure 5 shows that a smaller neighbourhood in this scene gives better results as closer neighbours are likely to share similar lighting distributions. However, this also results in blotches due to the high correlation in lighting distribution between neighbours. In scenes with more complex geometry, a larger neighbourhood might be more desirable as it is less likely that closer pixels are more correlated as in the Cornell Nightclub.

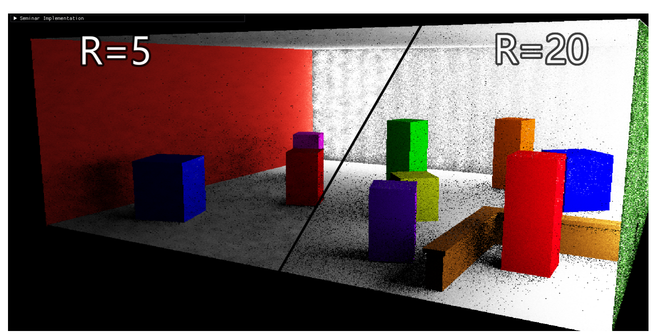


Figure 5. Comparison of spatial neighbourhood sizes for edge lengths 20 and 5.

References

[1] Simon Niedenthal. 2002. Learning from the Cornell box. *Leonardo* 35, 3 (2002), 249–254.

[2] Alastair J Walker. 1974. New fast method for generating discrete random numbers with arbitrary frequency distributions. *Electronics Letters* 8, 10 (1974), 127–128.