



# World Space ReSTIR in Vulkan

---

Jichu Mao    Zhiyi Zhou  
CIS 5650 - Final Project - 2024 Fall

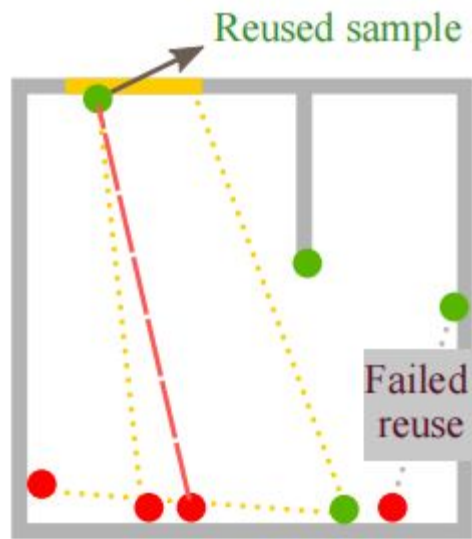
# Author

---

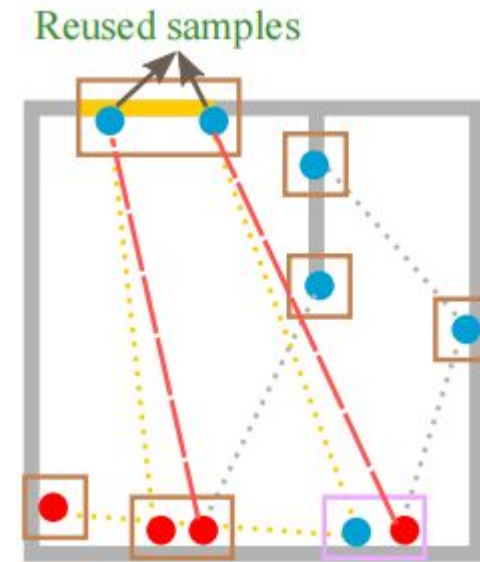
- **Zhiyi Zhou** (zhiyiz@seas.upenn.edu)  
[GitHub](#) [LinkedIn](#)
- **Jichu Mao** (jichumao@seas.upenn.edu)  
[GitHub](#) [LinkedIn](#)
- Please feel free to contact us via GitHub repository or by email.

# Overview

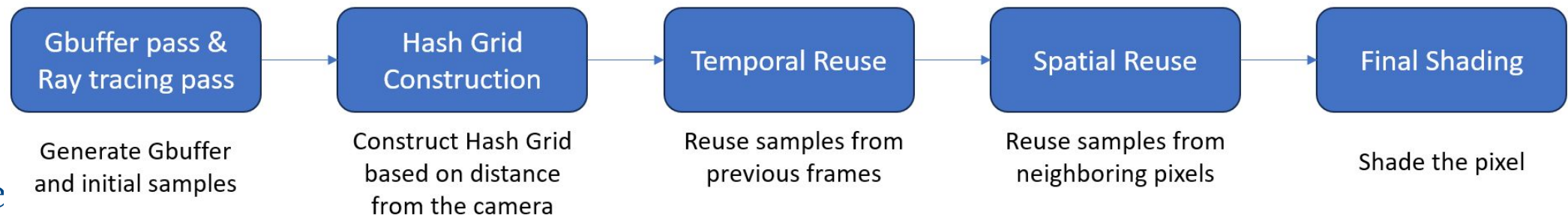
- **Goal:** Implement a real-time global illumination renderer based on world-space ReSTIR in Vulkan



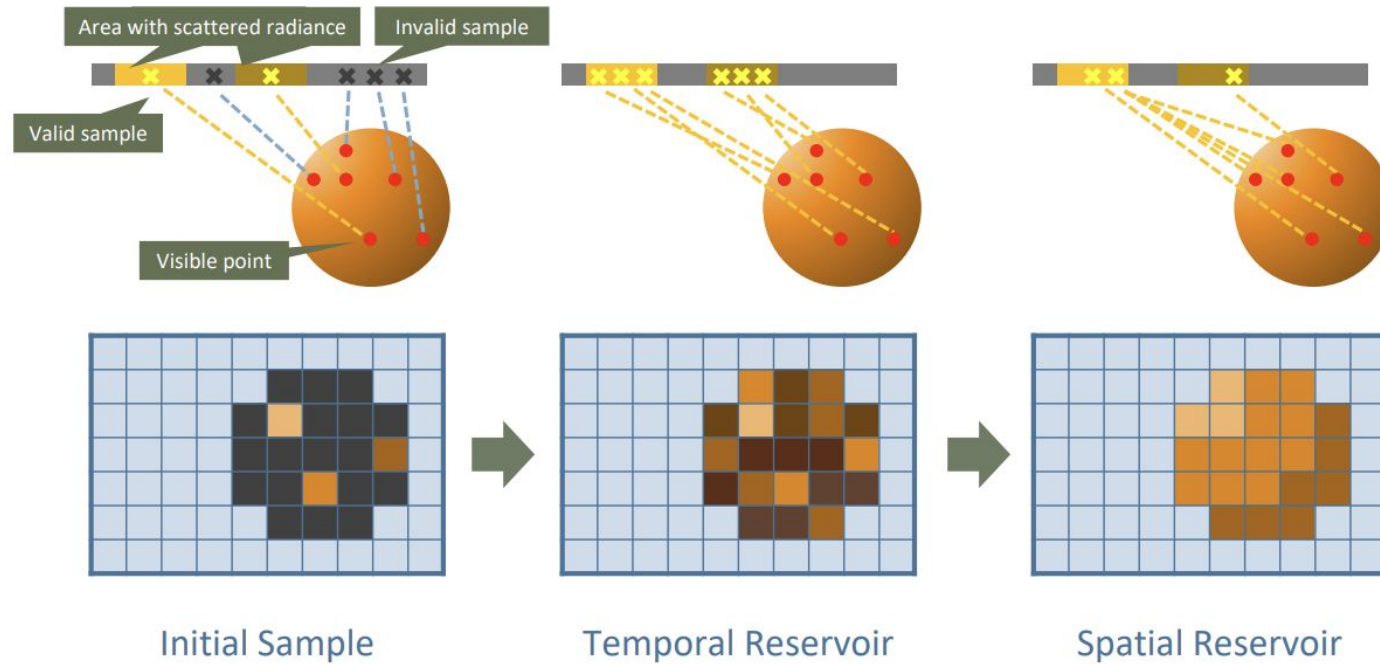
Pure ReSTIR



Ours(World Space ReSTIR)



# ReSTIR GI



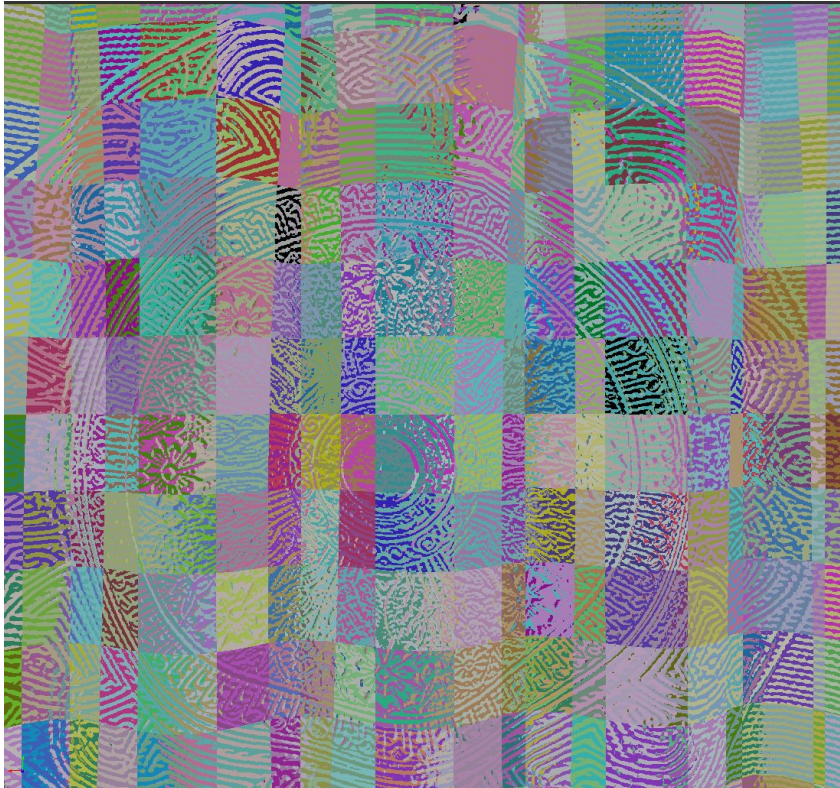
- The original **ReSTIR GI** (Reservoir-based Spatiotemporal Importance Resampling for Global Illumination) algorithm reuses samples in **screen space** to optimize rendering efficiency and improve performance.
- We are trying to use world space sample reuse to get higher quality reused sample.



# Hash Grid

$$\text{HashID} = \text{hash}(\text{position}, \text{normal})$$

Hash Grid



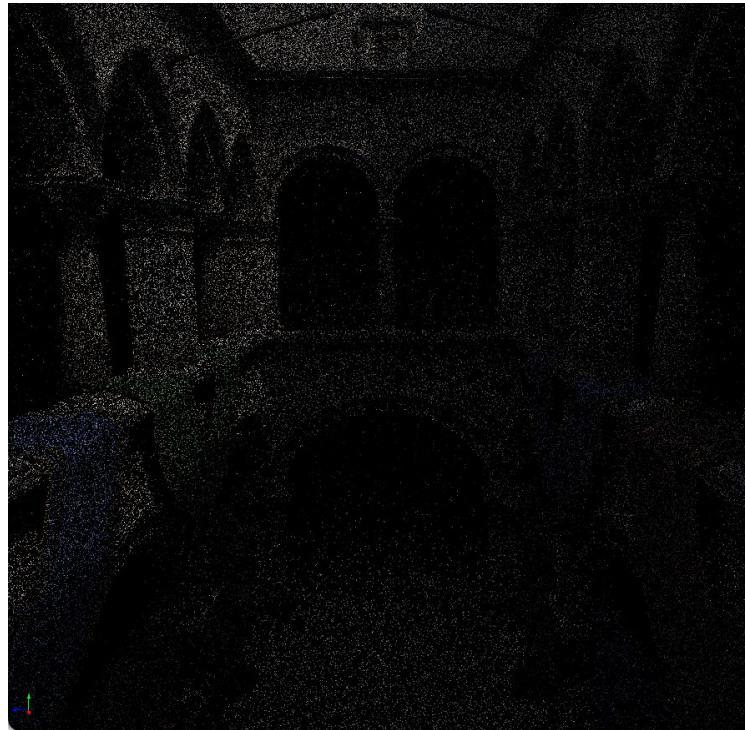
Normal





# Final Shading

- Ours(1 spp)
- Path Tracing(1 spp)
- Ground truth





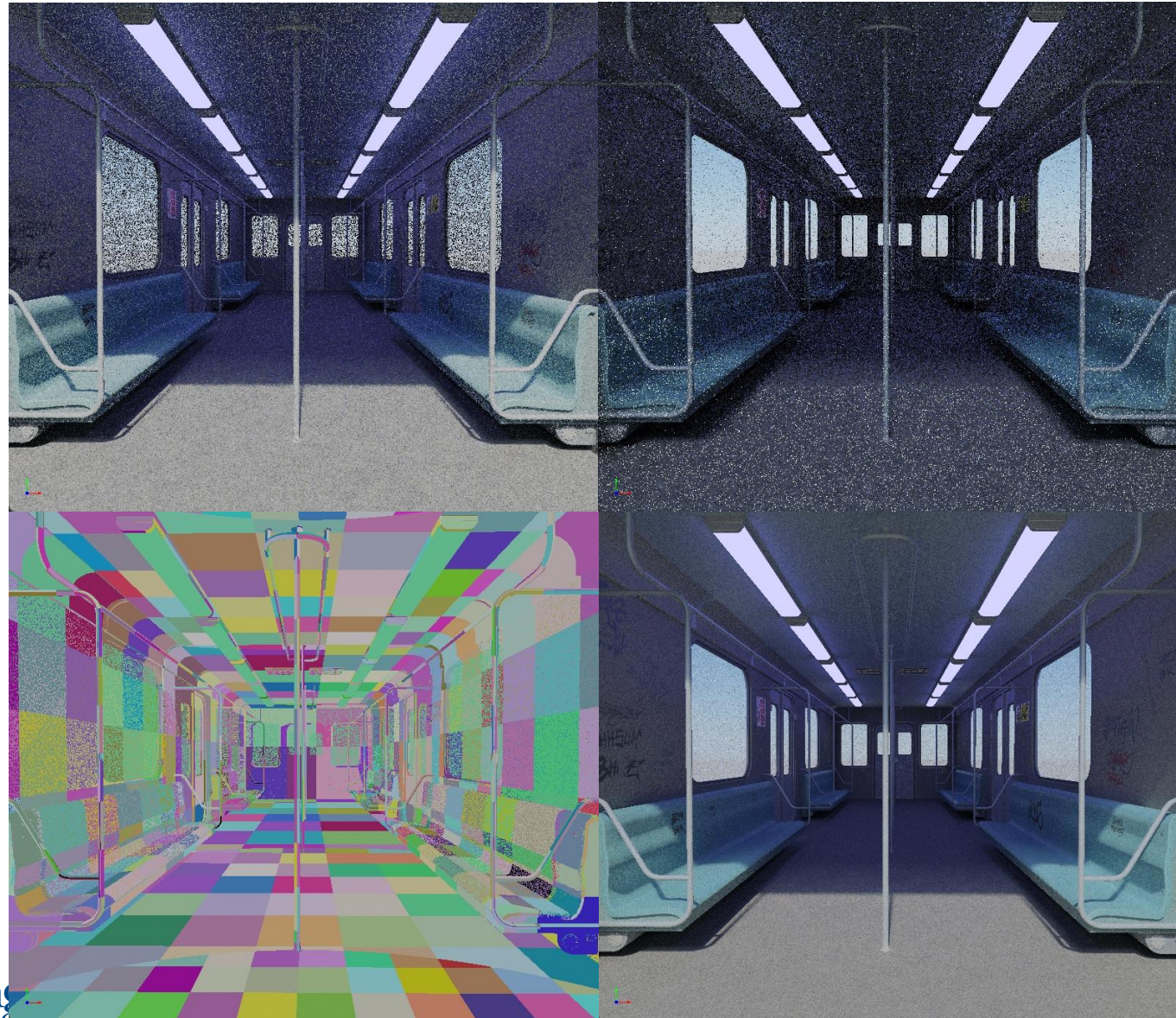
# Comparison

Ours  
(1 spp)

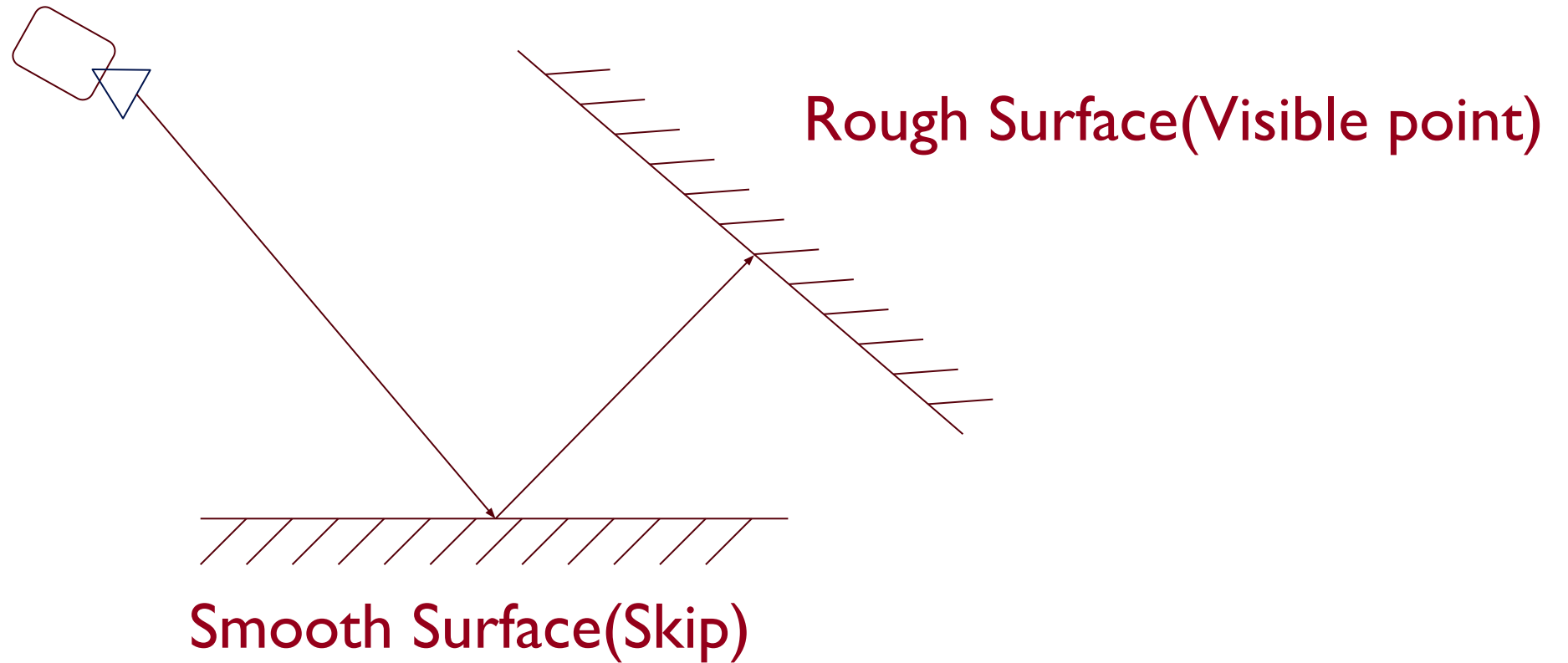
Path Tracing  
(1 spp)

Hash Grids

Ground truth

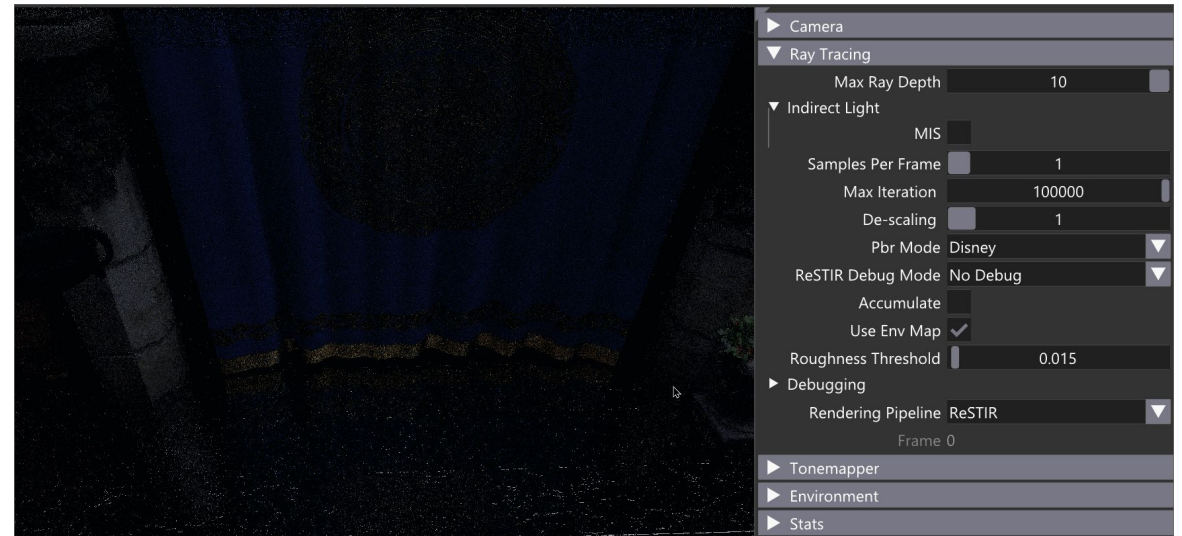
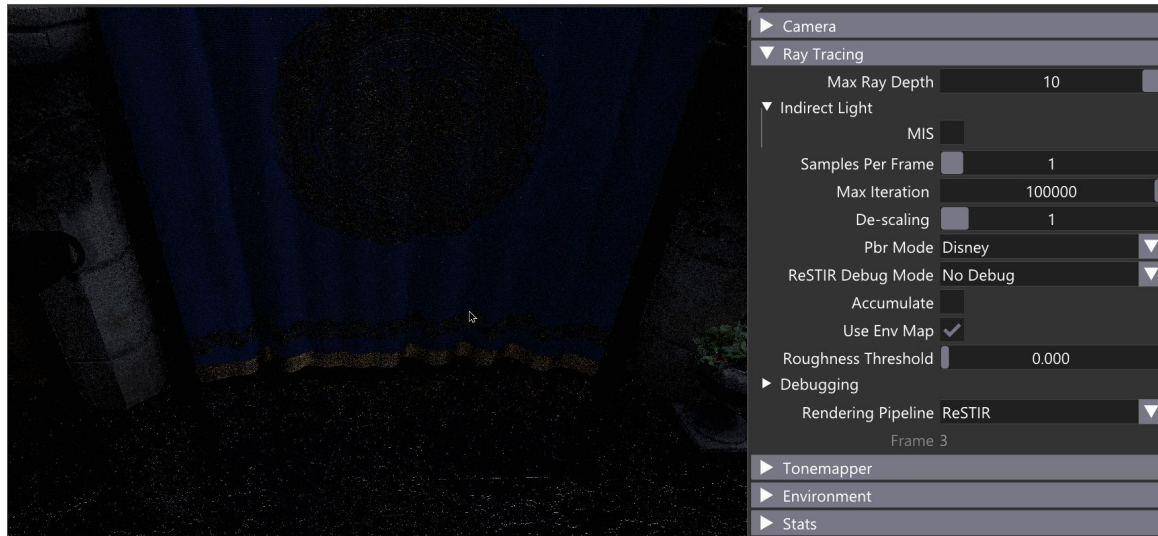


# Catch first rough vertex

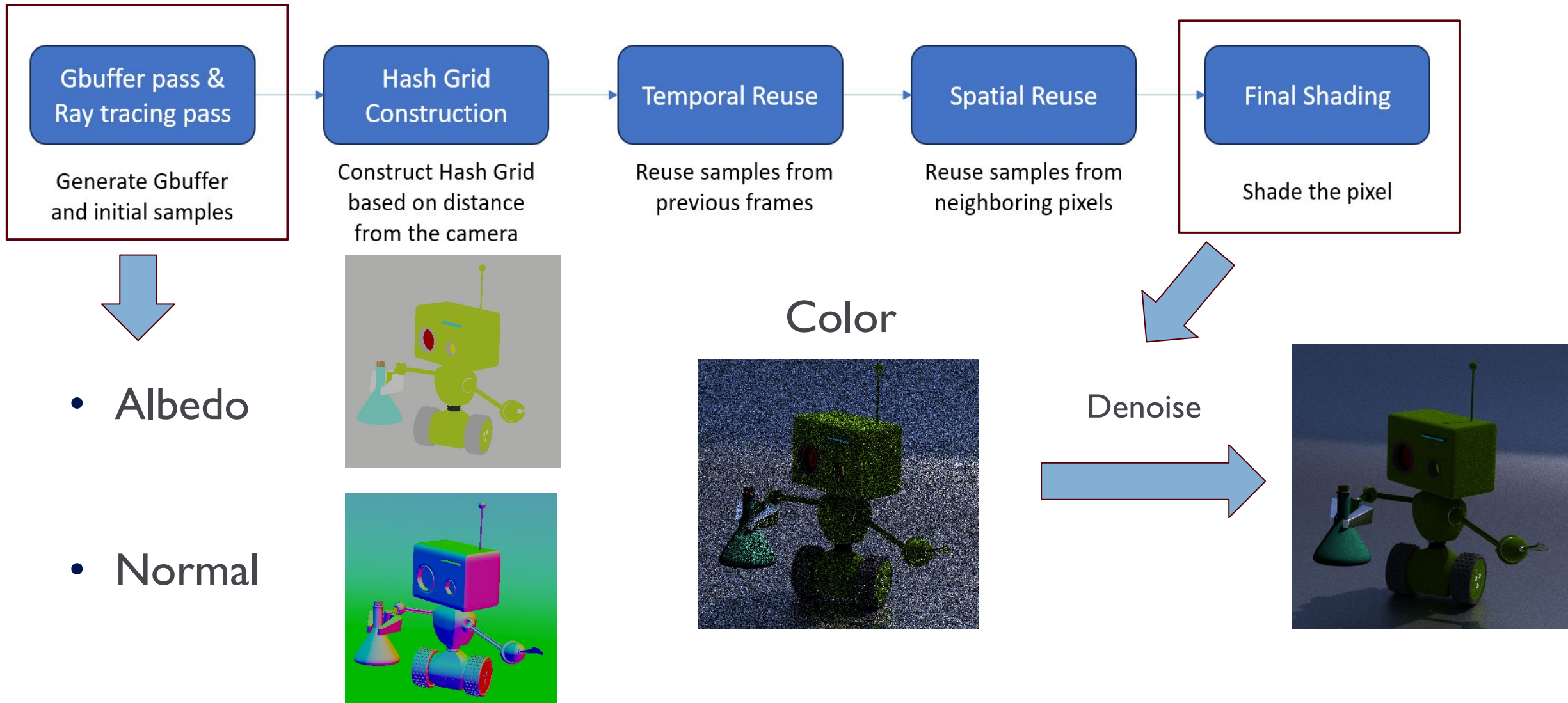




# Catch first rough vertex

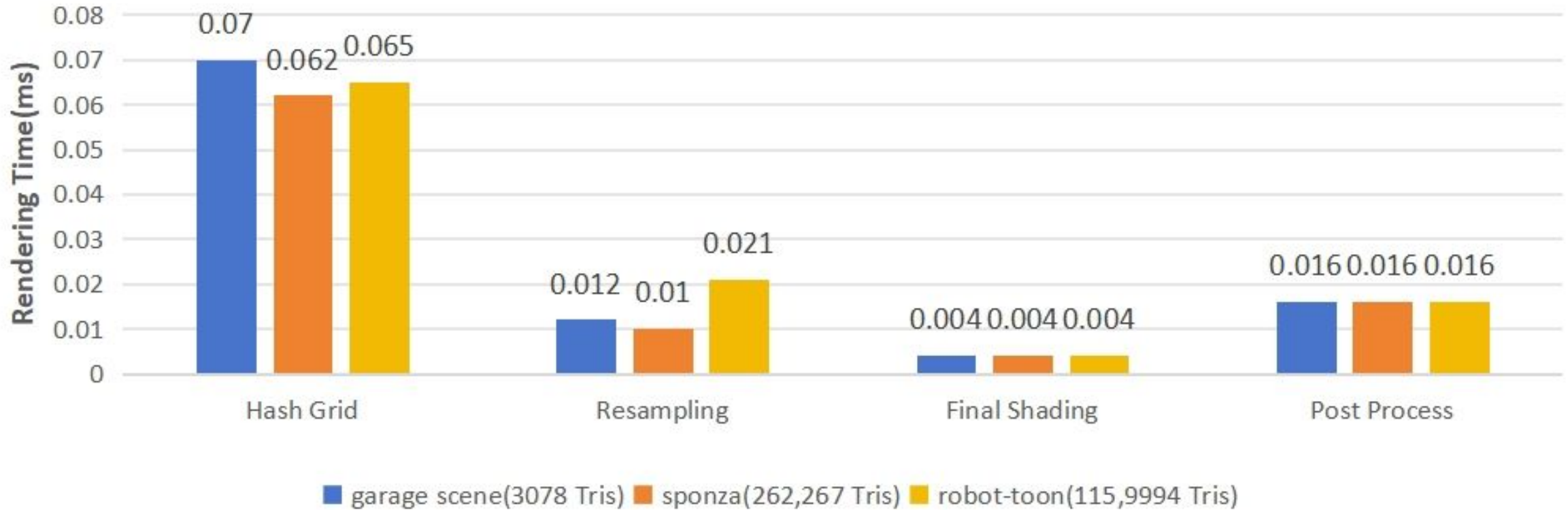


# Denoise Process



# Performance Analysis

Rendering Time Benchmarks with Different Scene  
(Lower is Better)





# Future Work

---

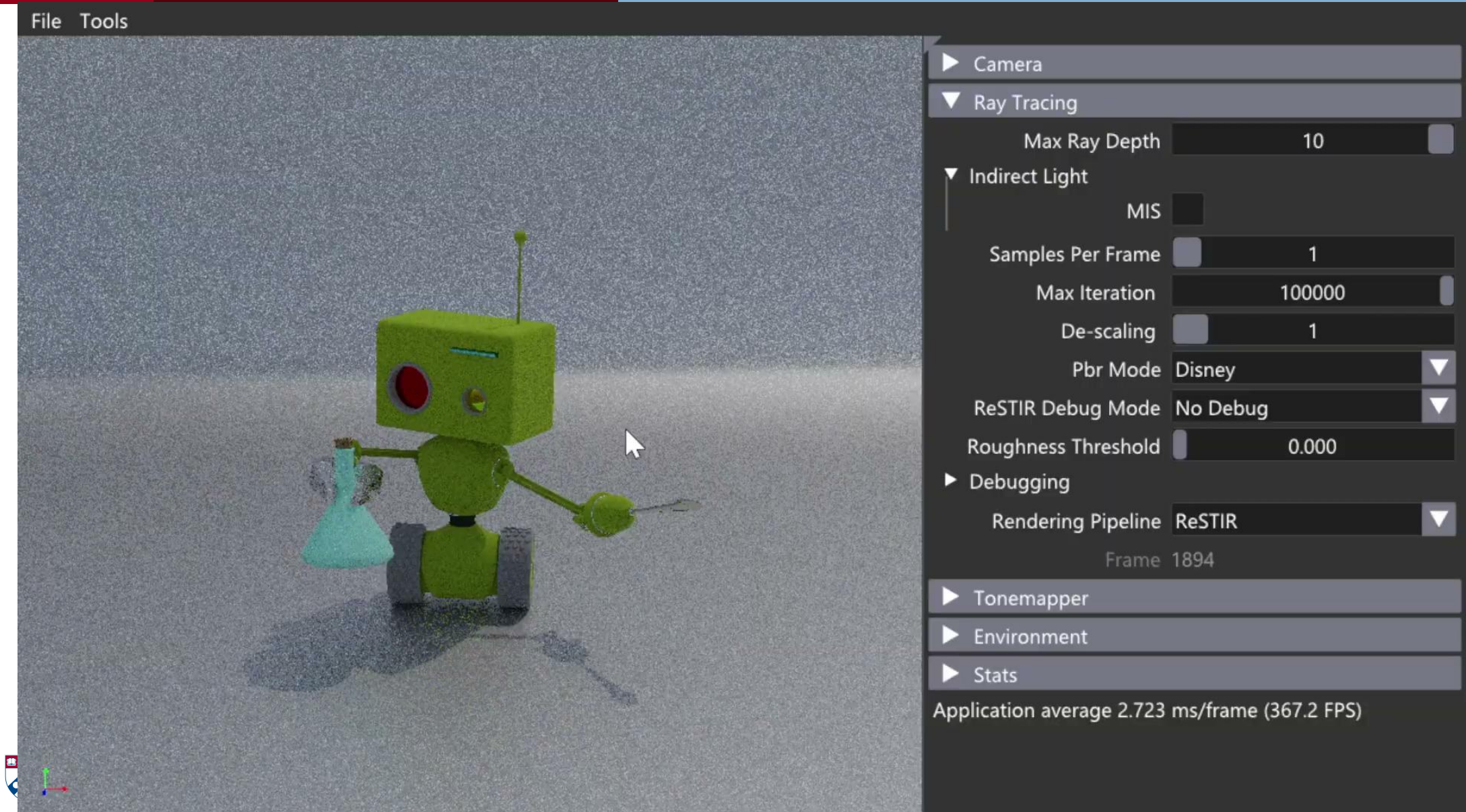
- Now use full screen resolution to do ray tracing, try to use lower resolution and combine with super resolution method like FSR
- Implement GPU based denoiser to improve the picture quality

# Credits

---

- Paper
- Nvidia's Vulkan raytrace renderer
- Vulkan ray tracing
- ray tracing tutorial
- subway scene

# Demo







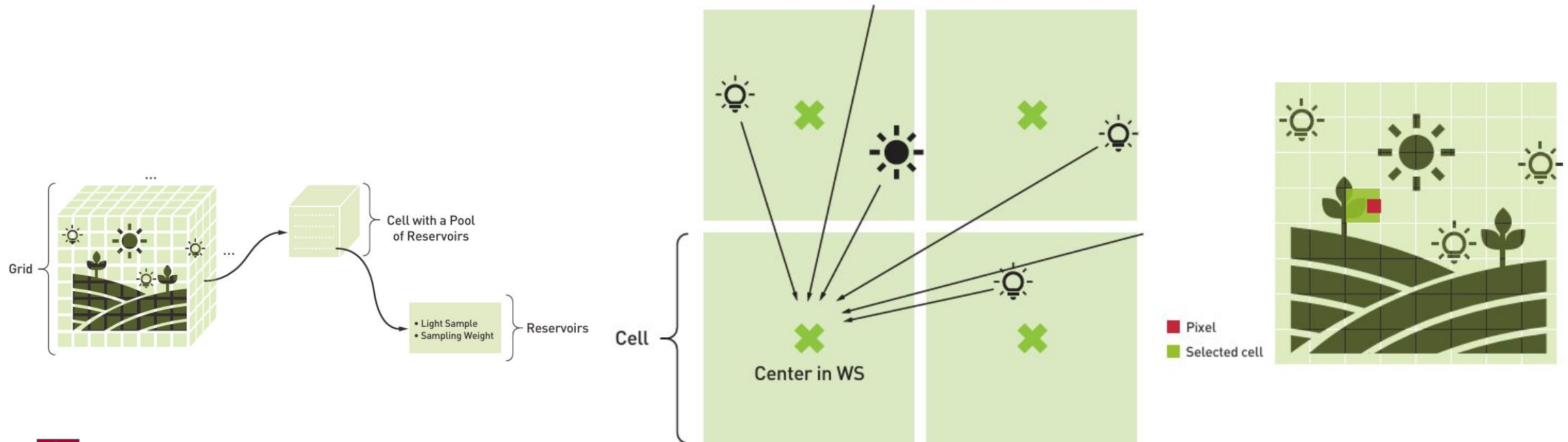
---

# Appendix

# Background

## What is GRID-BASED RESERVOIRS(ReGIR)?

ReGIR (Reservoir Grid Importance Resampling) is an algorithm designed for **efficiently rendering scenes with many light sources** in real-time ray tracing. It builds upon existing techniques such as ReSTIR (Reservoir Spatiotemporal Importance Resampling) and applies them to world-space sampling using a grid-based structure to optimize light sampling for secondary rays.



# Why this Project Matters?

- Original **ReSTIR GI** Operates in **screen space**. It optimizes global illumination sampling for primary rays and uses spatiotemporal resampling to manage indirect illumination. The reservoirs are tied to screen-space pixels. **ReGIR** Uses **world space**. It divides the scene into a 3D grid, with reservoirs distributed across these grid cells.
- We want to learn more about Vulkan as a modern graphics API and leverage hardware-accelerated ray tracing to achieve better performance.

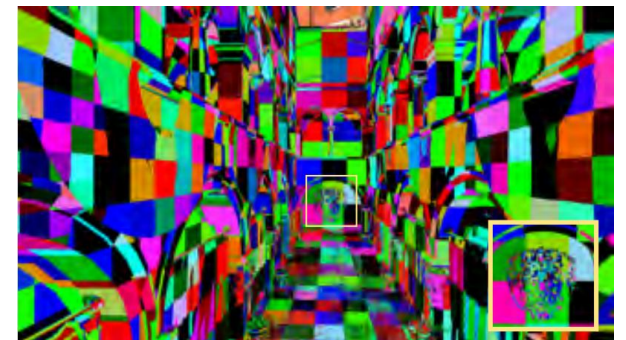
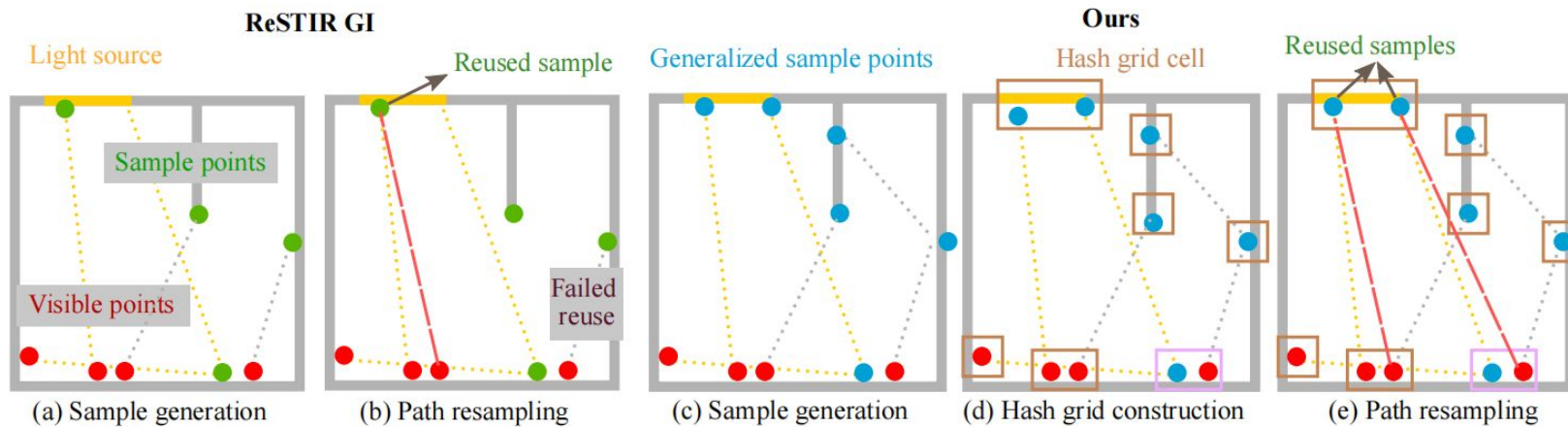
## Goal

- Implement **ReGIR** in Vulkan **to** optimize multi-light rendering and produce high-quality visuals under strict performance constraints.
- Optimization to the origin algorithm, e.g. adaptive grid resolution, time stability enhancement, efficient memory management



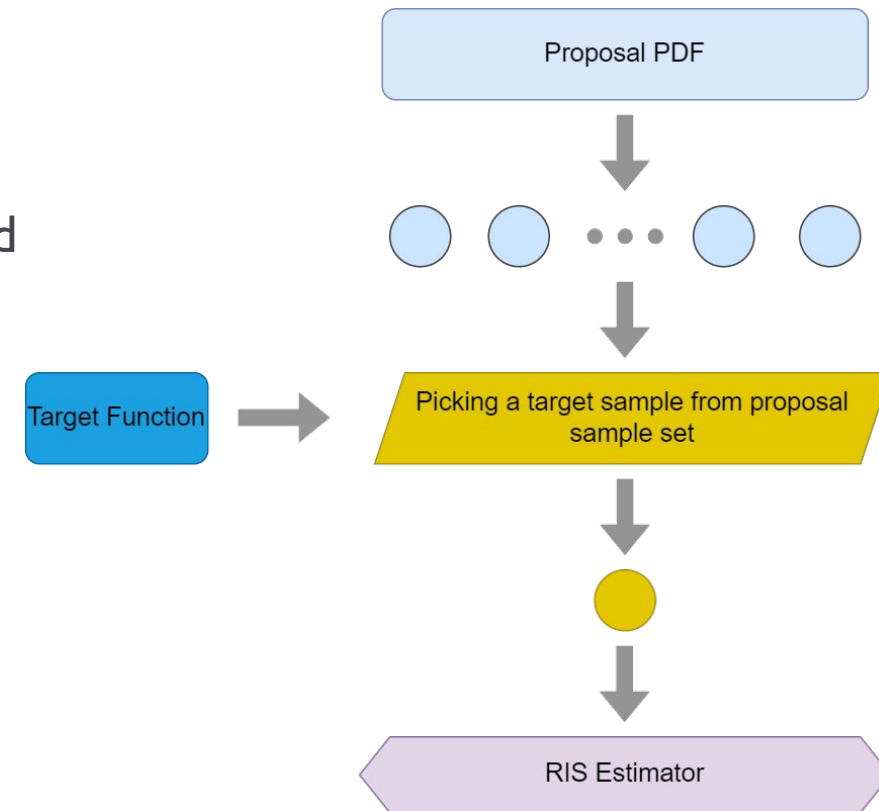
# Paper

- Pure **ReSTIR GI** Operates in **screen space**. It optimizes global illumination sampling for primary rays and uses spatiotemporal resampling to manage indirect illumination. The reservoirs are tied to **screen-space pixels**.
- We extend it to **world space**. It divides the scene into a **3D grid**, with reservoirs distributed across these grid cells.
- To construct the grid, we use a hash function based on the world position and surface normal to map these cells, allowing us to locate sample points with similar geometric properties.



# ReSTIR

- RIS: Resampled importance sampling. Draw  $n$  samples from a proposal distribution and select one based on a weighting function. This approach is progressively unbiased as  $n$  approaches infinity.
- WIS: Weighted Reservoir Sampling. The reservoir efficiently stores and manages samples without requiring all of them to remain in memory (we only need some statistical properties).
- Temporal / Spatial reuse: Reuse samples from neighboring pixels and previous frames, reservoir structure can help us to combine different reservoirs from different pixels and frames.

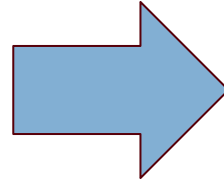


# Hash Grid Visualization

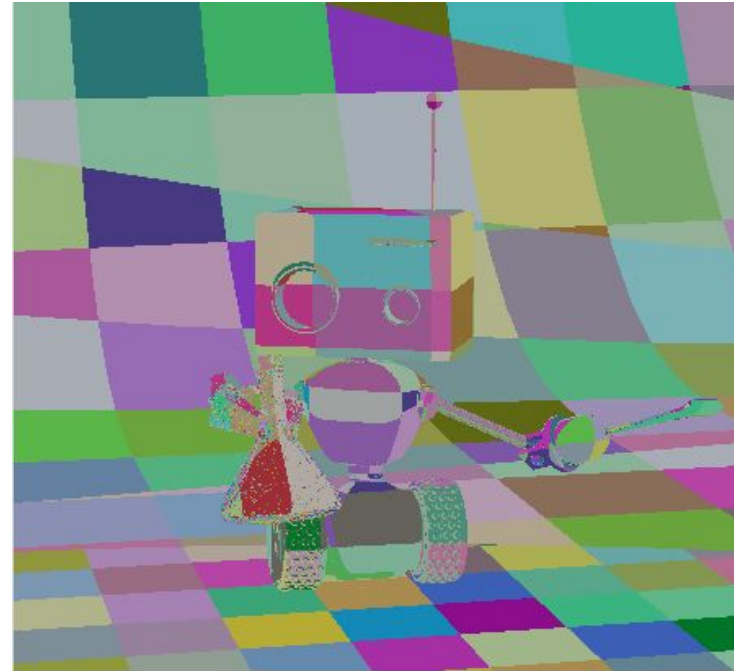
Undersized grid limit



Hash collisions



Optimized

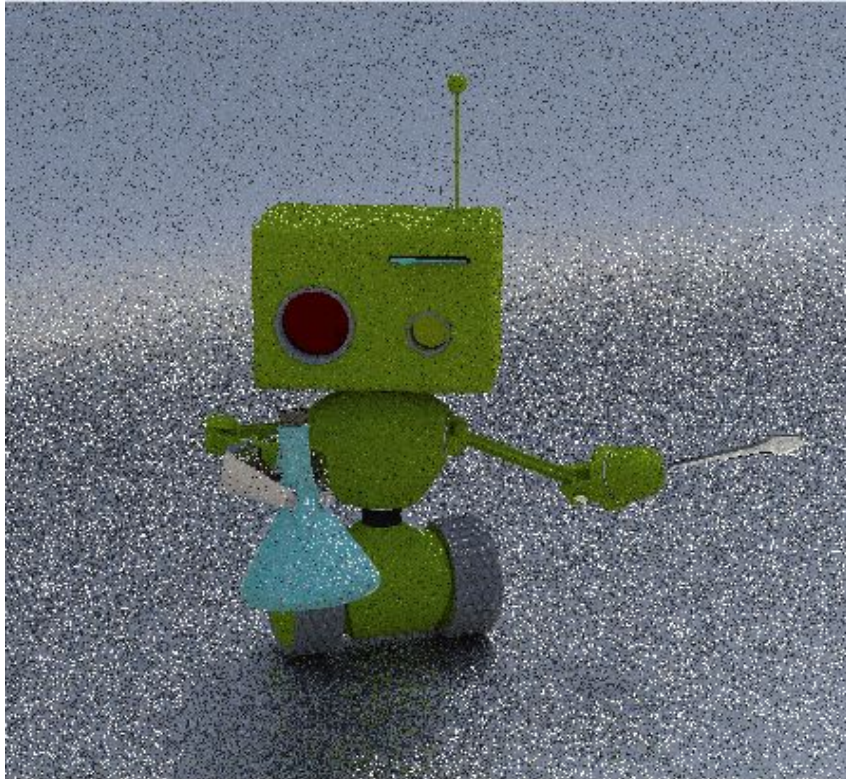


No collision

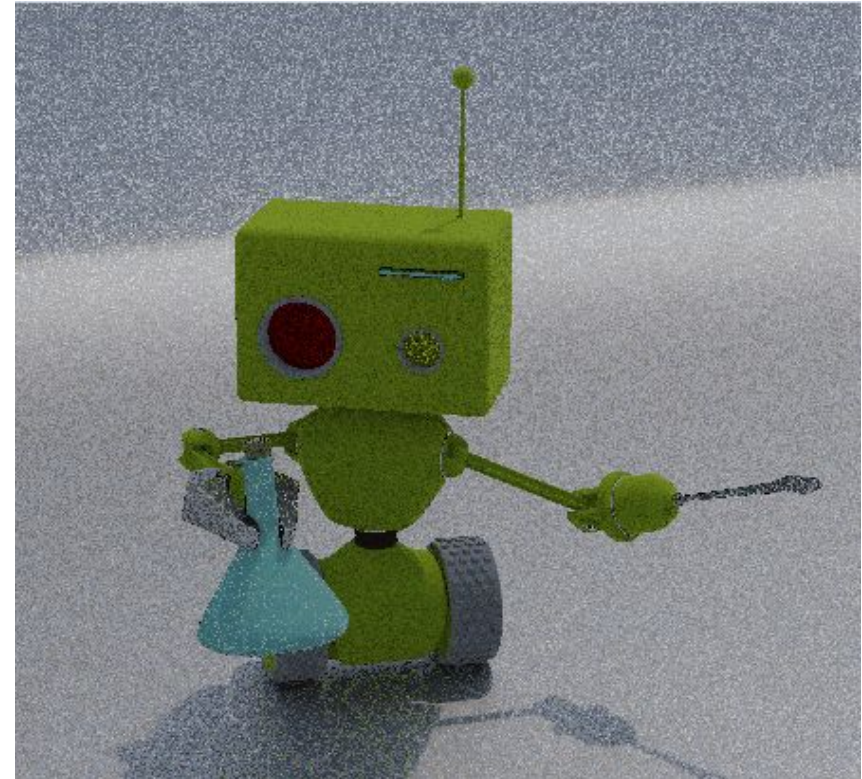


# Spatiotemporal Reuse

- Temporal Reuse



- Temporal & Spatial Reuse



# World Space ReSTIR DI

