A Raytracing Pipeline for the Radiance GPU

Shashank Anand, Yongye Zhu

1 Problem Definition

Radiance, a proposed open source heterogeneous GPU, is currently under development at the SLICE lab. A key feature of the GPU is decoupled accelerator orchestration: allowing special purpose units such as ray tracing cores and matrix multiplication units to be decoupled from the SIMT core, thereby allowing them to operate asynchronously while freeing up the SIMT cores to handle other instructions. The Radiance programming model enables programmers to interface with these decoupled accelerators. Radiance is inspired by Vortex [1], a RISC-V open source GP-GPU.

Latency from raytracing at high samples-per-pixel (SPP) is infeasible for real-time applications, and real-time raytracing is achieved by rendering at very low SPP and denoising the resulting image [2, 3]. ReSTIR GI [4] is a novel method for rendering better images at very low SPP (1 spp), improving the quality of the denoised image. ReSTIR GI is also supported in Unreal Engine 5[5]. Intel Open Image Denoise (OIDN) [6] is an open source denoising library, with pre-trained networks.

For the final course project, we aim to create a state of the art end-to-end neural rendering pipeline as a proof-of-concept for the applicability of our GPU architecture. We will implement a ReSTIR GI augmented path tracer and denoise the resulting noisy image with OIDN. We will port this pipeline to both the Vortex RISC-V programming model and the Radiance programming model, and evaluate the improvements in utilization and performance.

We note that the project is very ambitious and as a result, we will break it down into several checkpoints. We also highlight that the Radiance functional simulator is currently under development.

2 Goals

The ReSTIR GI implementation requires a substantial rewrite of the path tracer framework. Additionally we will also implement the path tracer on a traditional GPU (CUDA) for correctness reasons. We note that having advanced BSDF modelling would be great for sampling using MIS (the recommended method for ReSTIR GI), and that being able to ray trace a stream of images by moving the camera would help better utilize temporal sampling for ReSTIR GI.

2.1 Must Have

1. ReSTIR GI CPU implementation on top of HW3 Path Tracer infra.
2. ReSTIR GI CPU + OIDN integration and denoised results
3. ReSTIR GI GPU implementation
4. ReSTIR GI Vortex port + image rendered on Vortex functional sim
5. OIDN Vortex port + image denoised on Vortex functional sim

2.2 Stretch Goals

1. ReSTIR GI Radiance port
2. OIDN Radiance port

3. Advanced BSDF modeling for multiple importance sampling [7]

4. Ray tracing a sequence of images by rotating the camera along one or more axes

5. *GLTF support (?). Investigate if GLTF to Collada conversion is possible (example via Blender)*

3 Schedule

1. ReSTIR GI CPU implementation (mostly done).

2. ReSTIR GI + OIDN integration (under development, 3 days)

3. ReSTIR GI GPU implementation (1 week)

4. ReSTIR GI Vortex port (∼ 1 week)

5. ODIN Vortex port (∼0.5 week)

6. Stretch goal 1. and 2. (∼1.5 - 2 weeks)

7. Stretch goals 3-5 (∼1 week, if time permits)

4 Resources

The primary reference will the be the original ReSTIR GI paper and the OIDN library. No special hardware or software requirements. We will build the pathtracer on top of the HW3 framework.

References

[1] B. Tine, K. P. Yalamarthy, F. Elsabbagh, and K. Hyesoon, "Vortex: Extending the risc-v isa for gpgpu and 3d-graphics," in *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO '21, (New York, NY, USA), p. 754–766, Association for Computing Machinery, 2021.

[2] A. GPUOpen, "Neural supersampling and denoising for real-time path tracing," 2025. https://gpuopen.com/learn/neural_supersampling_and_denoising_for_real-time_path_tracing/.

[3] A. Scardigli, L. Cavigelli, and L. K. Müller, "Rl-based stateful neural adaptive sampling and denoising for real-time path tracing," in *Advances in Neural Information Processing Systems* (A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, eds.), vol. 36, pp. 66390–66407, Curran Associates, Inc., 2023.

[4] Y. Ouyang, S. Liu, M. Kettunen, M. Pharr, and J. Pantaleoni, "ReSTIR GI: Path Resampling for Real-Time Path Tracing," *Computer Graphics Forum*, 2021.

[5] gamegpu.com, "Nvidia updates unreal engine 5." https://en.gamegpu.com/game/nvidia-updates-unreal-engine-5-new-support-restir-gi-for-realistic-lighting.

[6] A. T. Áfra, "Intel® Open Image Denoise," 2025. https://www.openimagedenoise.org.

[7] E. Veach, *Robust Monte Carlo Methods for Light Transport Simulation.* PhD thesis, Stanford University, 1997. Chapter 9: Multiple Importance Sampling. Accessed: 2025-04-03.