TU WIEN Informatics

Real-Time Ray Tracing in Quake III

Lukas Lipp
Visual Computing

TU Wien Informatics
Institute of Visual Computing and Human-Centered Technology
Research Unit of Computer Graphics
Supervisor: Associate Prof. Dipl.-Ing. Dipl.-Ing. Dr.techn. Michael Wimmer
Assistance: Dipl.-Ing. Christian Freude BSc
Contact: lukas.lipp@outlook.at

Motivation and Problem Statement

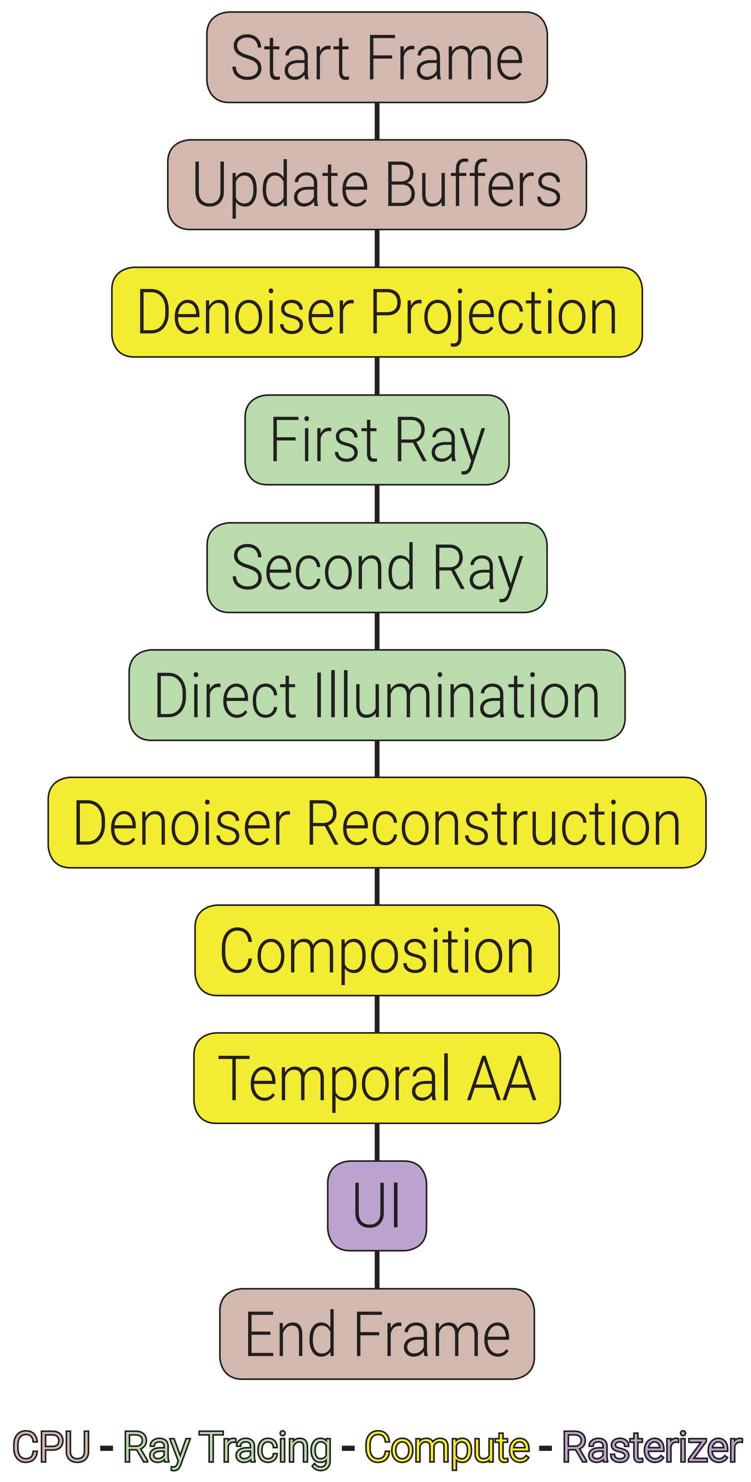
In 2018 NVIDIA released their new RTX GeForce graphics card generation, which are the first consumer-grade GPUs to offer hardware accelerated ray tracing. It is now the question to which extent this new hardware support allows the use of ray tracing in an established game engine. We will use distributed ray tracing (DRT) to render the entirety of Quake III's 3D game world. A naive ray-tracing implementation will, even with modern hardware acceleration through dedicated processors, be too computationally expensive.

Contribution

We discuss important aspects of modifying ray-tracing algorithms for real-time usage and implemented/analyzed the following optimizations:

- Random subsampling from a list of lights using random values from blue noise textures
- A light culling strategy using a pre-existing potentially visible set
- A denoiser called Adaptive Spatiotemporal Variance-Guided Filtering [1]

Implementation



This work consisted of the following steps:

1. Refactor Quake III
2. Implement a renderer based on Vulkan
3. Implement distributed ray tracing
4. Implement previously mentioned optimizations

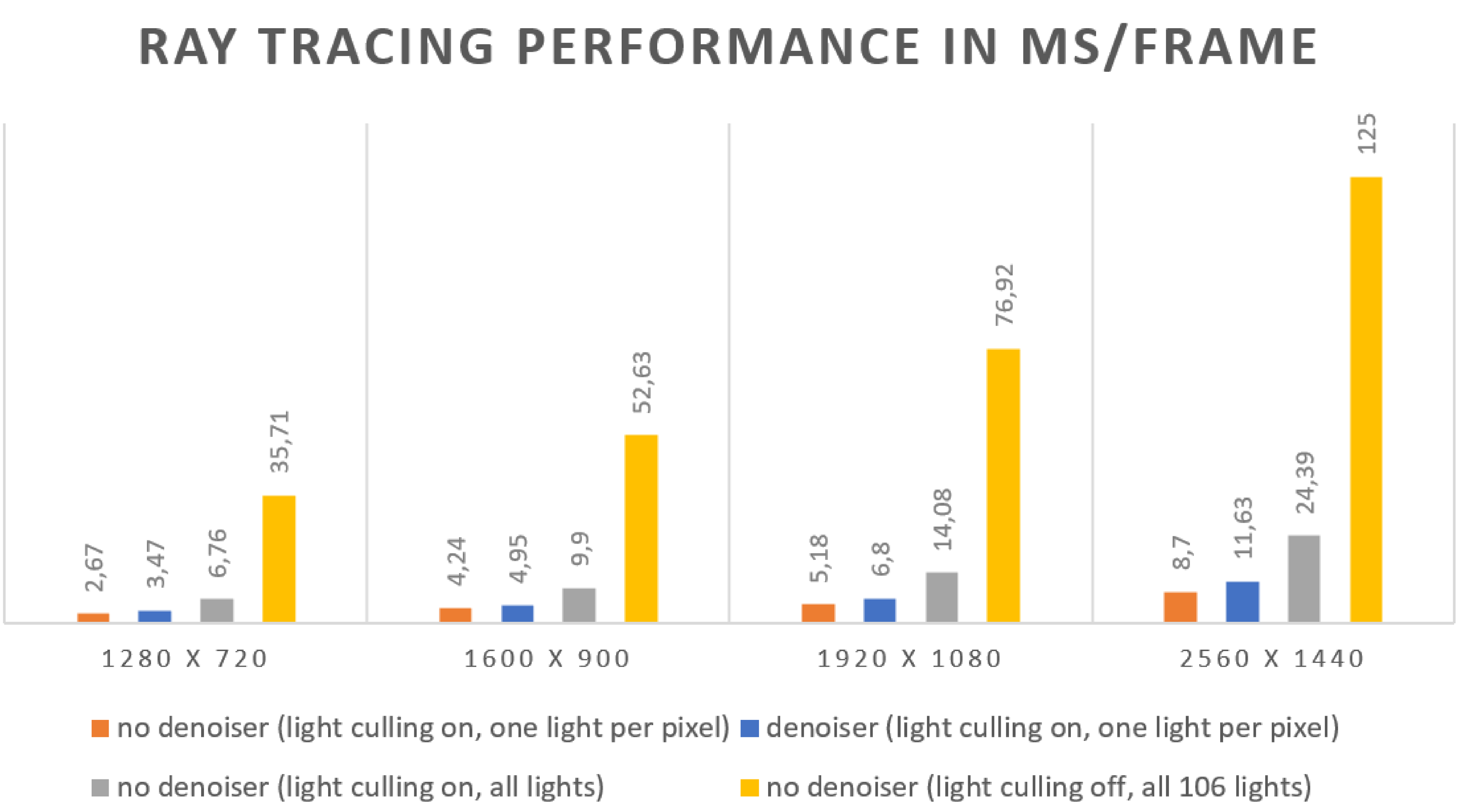
Follow the arrow to the right to see the results after each optimization. The graph on the left shows the required steps to produce one output frame.

Results

The GPU we used to produce these results was a GeForce RTX 2080 Ti running the Quake III level 'q3dm7'. The graph to the right shows the performance captured at different resolutions from the scene shown below.

We can tell from the graph that light culling does a great job at removing unimportant lights. Especially when all lights are used this results in a big performance gain. The denoiser combined with light subsampling (one light) also improves performance a lot, while still being able to produce a solid visual result.

Still, the most exciting results were when we combined the denoiser with light culling and subsampling of the lights. This strategy produced the best quality/performance ratio.



[1] Christoph Schied, Christoph Peters, and Carsten Dachsbacher. 2018. Gradient Estimation for Real-time Adaptive Temporal Filtering. Proc. ACM Comput. Graph. Interact. Tech. 1, 2, Article 24 (August 2018), 16 pages.