Ray tracing with OptiX

Degree programme: BSc in Computer Science | Specialisation: Computer Perception and Virtual Reality
Thesis advisor: Prof. Marcus Hudritsch
Expert: Dr. Harald Studer (Optimo Medical AG)

The purpose of this bachelor thesis is to explore the current state of affairs regarding real time ray tracing. How close are we to real time graphics with ray tracing? This question is answered by building a rendering program. The Nvidia framework OptiX is used to run the calculations on the graphics card.

Nic Alexander Dorner

Rasterization

Up until today the rasterization rendering technique has been used to render real time computer graphics. The rasterization algorithms are very efficient in terms of computational effort. Modern graphics cards are specially designed to compute these algorithms extremely quickly in real time. However, many effects that are used in computer graphics have to be approximated. For example: shadows, reflections and global illumination. To simulate these effects in an image, the programmers and designers have to use tricks. Sadly, these tricks can not be perfect and the rendered image will not be photo realistic.

Ray tracing

If photo realism is desirable then another technique comes to mind: Ray tracing. The idea of it is very simple and easy to understand. In theory we follow the path of the light from the light source to the eye of the observer (most implementations start from the eye however). A light ray bounces off materials and therefore illuminates other objects. This has the advantage of enabling lighting effects that are not feasible with rasterization per default. However, ray tracing is computationally very expensive. In this thesis the framework OptiX from Nvidia was used to create a rendering program for 3D computer graphics that runs on the GPU.

Classic ray tracer

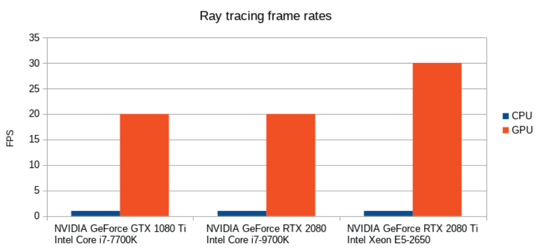
The classic ray tracing algorithm was used to create a renderer. The advantage of the classic ray tracing algorithm is that it is more efficient than other algorithms. A scene with some animations as well as reflections and refractions can be rendered in real time with a modern graphics card. Sadly, it is not a fully photo realistic implementation, meaning some real world effects can not be simulated.
With the finished rendering application a user can load a scene definition. The renderer displays an image from that description. The scene definition alone is sufficient to create the necessary lighting effects for the image to look authentic.

Path tracer

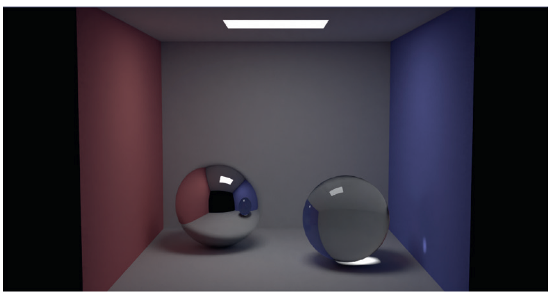
Another algorithm that was briefly explored is the path tracer. It uses a stochastic approach to render a scene and traces multiple samples that are combined in the end. It produces a noisy image at first but this can be dealt with by a denoiser. On the plus side, this algorithm is physically correct and enables more lighting effects than the classic ray tracer.

Result

The rendering application written in this thesis was used to compare the performance with that of a CPU based renderer. This showed that a much higher frame rate can be achieved by harnessing the parallel computing power of a graphics card.



Ray tracer fram rate comparison with tree graphics cards



Path tracer (1000 samples, gamma corrected, denoised)