Raytracing supported by AI (Abstract)

Felix Fuhl, 4AHITN 2020/21

Raytracing is a much talked about topic these days especially in combination with artificial intelligence. Especially in games, Raytracing is a very effective feature as it creates a much more realistic and lifelike experience through novel Raytracing and computation. This work is based to a large extend on internet sources and an interview with the director of Real Time Raytracing from Nvidia.

Raytracing as the name tells is a technique to follow rays from the source according to the laws of physics while they propagate through space, in particular Raytracing has a great advantage because it can represent shadows and reflective surfaces much more lifelike than other algorithms. The technology works by an algorithm that always uses a primary object to calculate the light rays reflected from it. Of course, this technology has its drawbacks, the main one being the vast power consumption, but that's where artificial intelligence comes in. This takes some of the computation away from Raytracing so the algorithm (and consequently the computer) doesn't have as much data to process. DLSS (Deep Learning Super Sampling) is a project from Nvidia that is supposed to combine these two technologies. Although DLSS is still in its infancy and faces many challenges, it gives us a glimpse of the future of Raytracing in any program.

Since Raytracing is based on very power-hungry algorithms, a satisfactory user experience is only possible if powerful computers are used, which means additional capital, expenditures and operation (power consumption).

The goal of many computer manufacturers is to create special CPUs, GPUs or other hardware parts that can cope with these lines and also have the lowest possible power consumption.

The goal of graphics card and game manufacturers is to create new algorithms with significantly reduced computing power requirements, which nevertheless model reality as precisely as possible.

Raytracing in combination with artificial intelligence is a promising approach to save resources or to use existing resources more efficiently. This is particularly important when using performance-hungry programs on less powerful computers, and it is particularly good when a less powerful computer has to execute a performance-hungry program. DSLL is a good example of the approach taken for this, but there are still major challenges to be mastered in order to make these hybrid algorithms suitable for the mass market.