ABSTRACT

Computer generated imagery is used in a wide range of disciplines, each with different requirements. As an example, real-time applications such as computer games have completely different restrictions and demands than offline rendering of feature films. A game has to render quickly using only limited resources, yet present visually adequate images. Film and visual effects rendering may not have strict time requirements but are still required to render efficiently utilizing huge render systems with hundreds or even thousands of CPU cores. In real-time rendering, with limited time and hardware resources, it is always important to produce as high rendering quality as possible given the constraints available. The first paper in this thesis presents an analytical hardware model together with a feed-back system that guarantees the highest level of image quality subject to a limited time budget. As graphics processing units grow more powerful, power consumption becomes a critical issue. Smaller handheld devices have only a limited source of energy, their battery, and both small devices and high-end hardware are required to minimize energy consumption not to overheat. The second paper presents experiments and analysis which consider power usage across a range of real-time rendering algorithms and shadow algorithms executed on high-end, integrated and handheld hardware. Computing accurate reflections and refractions effects has long been considered available only in offline rendering where time isn’t a constraint. The third paper presents a hybrid approach, utilizing the speed of real-time rendering algorithms and hardware with the quality of offline methods to render high quality reflections and refractions in real-time. The fourth and fifth paper present improvements in construction time and quality of Bounding Volume Hierarchies (BVH). Building BVHs faster reduces rendering time in offline rendering and brings ray tracing a step closer towards a feasible real-time approach. Bonsai, presented in the fourth paper, constructs BVHs on CPUs faster than contemporary competing algorithms and produces BVHs of a very high quality. Following Bonsai, the fifth paper presents an algorithm that refines BVH construction by allowing triangles to be split. Although splitting triangles increases construction time, it generally allows for higher quality BVHs. The fifth paper introduces a triangle splitting BVH construction approach that builds BVHs with quality on a par with an earlier high quality splitting algorithm. However, the method presented in paper five is several times faster in construction time.