Faster data structures and graphics hardware techniques for high performance rendering

a popular science summary

Per Ganestam
Department of Computer Science
Lund University

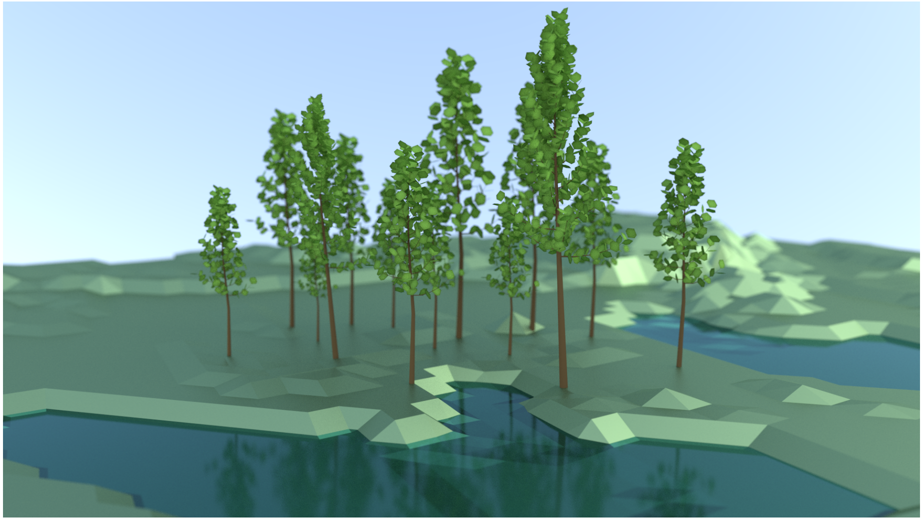

Figure 1: Path traced grove with ponds. Although a simple scene, this rendering displays a multitude of effects used in computer graphics. A glossy ground and water with reflections and refractions accompanied by diffuse inter-reflections. The leaves interact with light via subsurface scattering and the sky color is determined by a physically based sky model. The camera model creates a depth of field effect based on its aperture settings.

Computer generated imagery is used in a wide range of disciplines, all with different requirements. For instance, real-time applications such as games have completely different restrictions and demands than offline rendering for feature films. A game has to render quickly on limited resources, yet present visually adequate images. Film and effects rendering may not have strict time requirements but still need to render efficiently utilizing huge render systems with hundreds or even thousands of CPU cores.

A typical scene in computer graphics consists of millions of small triangles joined together to represent complex objects. Figure 1 displays a render of a scene presenting a few different effects commonly used in computer graphics. The image is generated using an offline rendering algorithm known as *path tracing*.

This thesis work presents two papers investigating and experimenting with analytical hardware models, feed-back systems and power consumption for real-time rendering applications. A project on producing computationally demanding reflections and refractions in real-time using a hybrid rendering approach is also presented. Two projects considering data structures used for the popular rendering techniques *ray tracing* and path tracing, often used in film rendering, are also presented.

In real-time rendering, with limited time and hardware resources, it is important to always 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 energy source, 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 method uses techniques with the real-time algorithm *rasterization* to approximate reflections and refractions in the distance but guarantees high quality effects nearby the viewer by the means of ray tracing.

The fourth and fifth paper present improvements in construction time and quality of a type of data structure often used to accelerate ray tracing. The data structure is called Bounding Volume Hierarchy (BVH) and is commonly a binary tree representing 3-dimensional data in a space partitioning hierarchy. Building BVHs quicker reduces rendering time in offline rendering and also brings ray tracing a step closer towards a feasible real-time approach. However, it is not necessarily beneficial to build BVHs quickly if they lack quality in terms of rendering performance.

Bonsai, presented in the fourth paper, constructs BVHs on CPUs faster than contemporary competing algorithms and produces BVHs of a very high quality. In some cases with superior rendering performance compared to competing BVH algorithms.

Following Bonsai, the fifth paper presents an algorithm that refines BVH construction by allowing triangles to be arbitrarily split. Although splitting triangles increases build times, 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.