UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

An Empirically Derived System for High-Speed Rendering

by Pierre (HR) Rautenbach

Submitted in partial fulfilment of the requirements for the degree

Doctor Philosophiae (Computer Science)

in the Faculty of Engineering, Built-Environment

and Information Technology

University of Pretoria

April 11th 2012

Abstract

This thesis focuses on 3D computer graphics and the continuous maximisation of rendering quality and performance. Its main focus is the critical analysis of numerous real-time rendering algorithms and the construction of an empirically derived system for the high-speed rendering of shader-based special effects, lighting effects, shadows, reflection and refraction, post-processing effects and the processing of physics. This critical analysis allows us to assess the relationship between rendering quality and performance. It also allows for the isolation of key algorithmic weaknesses and possible bottleneck areas.

Using this performance data, gathered during the analysis of various rendering algorithms, we are able to define a selection engine to control the real-time cycling of rendering algorithms and special effects groupings based on environmental conditions. Furthermore, as a proof of concept, to balance Central Processing Unit (CPU) and Graphic Processing Unit (GPU) load for and increased speed of execution, our selection system unifies the GPU and CPU as a single computational unit for physics processing and environmental mapping. This parallel computing system enables the CPU to process cube mapping computations while the GPU can be tasked with calculations traditionally handled solely by the CPU.

All analysed and benchmarked algorithms were implemented as part of a modular rendering engine. This engine offers conventional first-person perspective input control, mesh loading and support for shader model 4.0 shaders (via Microsoft's High Level Shader Language) for effects such as high dynamic range rendering (HDR), dynamic ambient lighting, volumetric fog, specular reflections, reflective and refractive water, realistic physics, particle effects, etc. The test engine also supports the dynamic placement, movement and elimination of light sources, meshes and spatial geometry.

Critical analysis was performed via scripted camera movement and object and light source additions – done not only to ensure consistent testing, but also to ease future validation and replication of results. This provided us with a scalable interactive testing environment as well as a complete solution for the rendering of computationally intensive 3D environments. As a full-fledged game engine, our rendering engine is amenable to first- and third-person shooter games, role playing games and 3D immersive environments.

Evaluation criteria (identified to access the relationship between rendering quality and performance), as mentioned, allows us to effectively cycle algorithms based on empirical results and to distribute specific processing (cube mapping and physics

processing) between the CPU and GPU, a unification that ensures the following: nearby effects are always of high-quality (where computational resources are available), distant effects are, under certain conditions, rendered at a lower quality and the frames per second rendering performance is always maximised.

The implication of our work is clear: unifying the CPU and GPU and dynamically cycling through the most appropriate algorithms based on ever-changing environmental conditions allow for maximised rendering quality and performance and shows that it is possible to render high-quality visual effects with realism, without overburdening scarce computational resources. Immersive rendering approaches used in conjunction with AI subsystems, game networking and logic, physics processing and other special effects (such as post-processing shader effects) are immensely processor intensive and can only be successfully implemented on high-end hardware. Only by cycling and distributing algorithms based on environmental conditions and through the exploitation of algorithmic strengths can high-quality real-time special effects and highly accurate calculations become as common as texture mapping. Furthermore, in a gaming context, players often spend an inordinate amount of time fine-tuning their graphics settings to achieve the perfect balance between rendering quality and frames-per-second performance. Using this system, however, ensures that performance vs. quality is always optimised, not only for the game as a whole but also for the current scene being rendered – some scenes might, for example, require more computational power than others, resulting in noticeable slowdowns, slowdowns not experienced thanks to our system's dynamic cycling of rendering algorithms and its proof of concept unification of the CPU and GPU.

Key words and phrases: Algorithms, Ambient Occlusion, Chromatic Dispersion, Displacement Mapping, Distributed Rendering, Depth-of-Field, Dynamic Algorithm Selection, Fuzzy Logic, High Dynamic Range Lighting, Instruction Set Utilisation, Light Maps, Normal Maps, Parralax Mapping, Particles, Physics, Reflection, Refraction, Shaders, Shadow Mapping, Soft Shadows, Spatial Subdivision, Specular Highlights, Stencil Shadow Volumes, Volumetric Materials.

Supervisor: Prof. Dr. D.G. Kourie
Department of Computer Science
Degree: Doctor Philosophiae

Acknowledgements

First of all, my eternal gratitude and thanks to my supervisor and mentor at the University of Pretoria, Prof. Dr. Derrick G. Kourie, for support, encouragement and for being a great sounding board and a ceaseless source of inspiration.

I would also like to thank Prof. Dr. Dr. Bruce W. Watson and Prof. Dr. Roelf van den Heever for all their guidance, inspiration and for believing.

Special thanks to some of the best students at the university (and some of the best software developers I know), Deon Pienaar, Kepler Engelbrecht, Chris Schulz, Tiaan Scheepers, Jaco Prinsloo and Morkel Theunissen. My appreciation for your friendship and for putting up with all my start-up ideas!

Last but not least I am grateful to my family for their love and encouragement.

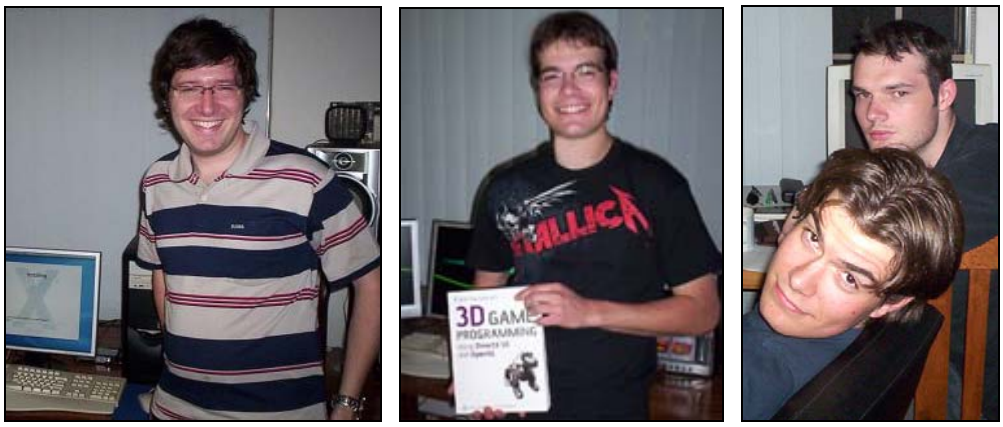
Preface
An Informal Personal History of this Thesis

This preface provides an historical account of the genesis and evolution of ideas in this study as well as how these ideas have flowed into one another over time. It is provided so that the reader clearly understands the scope of the thesis and is able to differentiate between its roots in earlier developments, and the new work undertaken to complete the thesis.

Development of the presented rendering engine, as a first-person-shooter game engine, commenced on the 3rd of March 2002. It started out as a silly little student project but has since lead to a textbook, a MSc dissertation and now this, a PhD thesis. When I was a second year electronic engineering student back in 2002, I teamed up with two fellow students with the idea that we, a couple of kids hailing from South Africa, could take on the mighty id Software and write a 3-D first-person-shooter featuring dynamic lighting, curved surfaces, shadows, bump mapping and a number of other technological advancements. These two friends shared my inextirpable passion for making games but also my burning desire to be a part of the next big software start-up. Deon Pienaar, coder extraordinaire and no holds barred genius and Kepler Engelbrecht, a electrical engineering student bent on becoming the next Shigeru Miyamoto (the creator of *Mario*, *Donkey Kong* and *The Legend of Zelda,* some of the most successful video game franchises of all time). As for yours truly, I had spent every waking moment since getting my first computer at the age of thirteen programming games ranging from crude text-adventures to 3-D simulations based on the tracing of light-paths between objects in an image plane. I should also mention that, in addition to my passion for 3-D graphics and gaming, my penchant for 80's metal and long hair (at the time, at least) didn't really hurt when it came to bonding with these guys. So, it was settled: I would hone my skills as graphics programmer and we would spend day and night chasing the dream of founding a company where we would be rock stars and where the easy money was only a game away. It sounds like a fantasy but it didn't feel too out of reach at the time. It was the early 2000s and anything seemed possible; the dot-com boom had just happened and all we had to do was emulate a couple of guys up in Texas who were continuously redefining the 3-D first-person-shooter genre through multimillion-dollar grossing titles like *Doom*, *Quake, Quake II* and the seminal *trequal*, *Quake III*. As they say, it was ON!

The next three years were spent programming, sleeping on pizza boxes (please note, I might be exaggerating) and missing out on a lot varsity life had to offer (socially, at least). But we did it and with Kepler dealing with network coding and the game logic, Deon working his magic with the cognitive AI implementation (amongst other things) and

yours truly implementing the renderer, we finally had the technology to create “the coolest technology demonstration ever.” The problem with this was that in order to showcase everything the engine was capable of, we suddenly needed artistically gifted individuals to put together a number of appropriate environments to emphasise the engine’s capabilities. We wanted detailed 3-D models of buildings arranged to form a coastal town set in an archipelago-type landscape. We wanted a beach, a forest, canyons, an old, explorable mineshaft. We wanted a rocket-propelled grenade launcher mounted nearby to fire at the player as he made his way through town. However, taking our artistic disposition into account, a compromise was eventually struck and it was decided that our tech demo would feature a basic coastal town with source-less mortar fire raining down on the player…nothing more, nothing less and things were more or less back on track. That said, this was, sadly, the beginning of the end – graduation came and real-life soon caught up with a vengeance! In short: Kepler has since joined SAAB and is currently doing a lot of low-level programming on defence projects while Deon remains dedicated to the dream, the two of us (along with some other friends and enemies of modern music) still chasing start-up stardom!



Deon, Pierre and Pierre and Kepler…

So, following that first four years and desperately trying to avoid the dark reality that is the 9-to-5, I had a truly great and empowering idea; why not write a 3-D game programming text book? I set out writing a simple “OpenGL and C++ in 21 days” sort of book, you know the thing, where all the basics would be presented in bottom-up fashion. Having completed the book, I couldn’t think of anyone better than one of my professors to show it to. Professor Judith Bishop, the author and co-author of 15 monographs and text books translated into German, Italian, Spanish, Polish and Russian was the obvious choice and strangely enough, everything worked out exactly as I had planned. Well, almost; my “3-D Programming Power Book” wasn’t a truly unique concept or really marketable but it was definitely a taste of things to come. In summary, I gave Prof Bishop a printed copy which she handed over to an editor friend of hers in London who, in turn, contacted me with an offer to write something truly unique – a textbook where two competing graphics/game programming technologies would be covered in a

seamless, parallel fashion. It was now June 2006 and by January 2007, I had signed on the dotted line of a writing contract.

Thinking back to 2007, I recall days going to the gym at 7:00 and writing from 9:00 to 18:00, seven days a week. It was just what I needed; I fancied myself a real writer and relished being in complete and utter control of my time (and life). And when it finally came out in October 2008, the 679 page book garnered reviews lauding the writing style as “a pleasure to read” and the concept as “excellent”. 3D Game Programming Using DirectX 10 and OpenGL (Rautenbach, 2008) was an instant hit, it was one-of-a-kind and it was in many ways a one-hit-wonder with even the worst reviews being positive. Most importantly, the book captured everything I knew and it served as a triumphant tour de force in what could easily have been a short-lived and disappointing game programming career.

The book caused me to spend a lot of time on shadow rendering (mostly stencil shadow volume optimisations and the development of a hybrid spatial subdivision approach). This research and ceaseless experimentation eventually culminated in the development of an empirically derived system for high-speed shadow rendering. This system was subsequently documented and submitted in partial fulfilment of the requirements for the degree Magister Scientiae (Computer Science), which was awarded *cum laude* on the 20th of April 2009. The presented thesis now builds on this past work, the core aim being the continuous maximisation of rendering quality and performance through the real-time cycling and distribution of algorithms, calculations and rendering approaches based on ever-changing environmental conditions.

Table of Contents