

Project Pre-proposal

| | | | |
|---------------|--|------------|--------|
| Surname | Jones | First name | Robert |
| Home Phone | | Work Phone | |
| Email address | rob@phantom-web.co.uk | | |

Application of Transmission Line Matrix Theory on Modern Day Graphics Processing Unit Hardware

Introduction

Transmission Line Matrix (TLM) theory allows for the simulation of wave propagation and allows for the modelling of this. To date this simulation has been performed on the CPU of a computer system however with the advent of programmable graphics processing units (GPUs) it is now possible to leverage the power of this hardware to remove the burden of many calculations from the CPU to the GPU which can perform many calculations in parallel and thus speed up operations.

This project builds upon the work carried out by Steve Harris and Krisitan Dor and will be using some of their work as a foundation.

I have chosen to proceed with this project as it allows me to pursue my own interests in computer programming, GPU programming and visualisation which have been a hobby of my own since 2000.

Project Aim

The purpose of this project is to investigate the potential for this off loading in TML simulations using and comparing various techniques with the over all aim of the project to produce results which show which method is potentially the best route to take when utilising modern hardware.

Project Deliverable

The final deliverable will be a piece of software which will be used to both benchmark and visualise the various methods used in the comparison. The software will have a GUI allowing for a change in the method of simulation as well as allowing for various settings to be tweaked in real time.

Project Objectives

- Investigate TLM theory
- Select and familiarise myself with the relevant 3D API and any support APIs
- Produce a basic framework to work with
- Adapt Kristian's application to fit within the required framework (with possible optimisations to allow for greater parallelism on the CPU)
- Produce a CPU/GPU hybrid implementation of the TML algorithm
- Produce a GPU only implementation of the TML algorithm
- Produce a GPU only implementation of the TML algorithm using next generation hardware (known as D3D10 hardware)
- Produce a 'close to the metal' (CTM) implementation of the TML algorithm using current hardware and/or next generation hardware
- Produce a 'final' application to host the various implementations of the algorithm

Note: The final two objectives are optional as the latter requires access to ATI's CTM Beta program (currently waiting on registration) and the former depends somewhat on hardware and software availability as it requires hardware which currently isn't in production and Windows Vista as well as the relevant software development tools.

The proposed methods of comparing the various methods will be to use a combination of some or all of the following metrics;

- Frames per second
- Image quality
- Number of vertices rendered/generated
- Memory usage
- Other resource usage
- CPU time required