Rhodes University

Submitted in partial fulfilment
of the requirements of the degree of

Bachelor of Science (Honours)

Creating and Optimizing a Sky Tessellation Algorithm for the SKA: Literature Review

Antonio Bradley Peters

supervised by
Prof. Karen Bradshaw
Prof. Denis Pollney

project originated by
Prof. Oleg Smirnov

April 21, 2016

1 Introduction

1.1 The SKA

1.2 Image Capturing

1.3 Errors in the Image

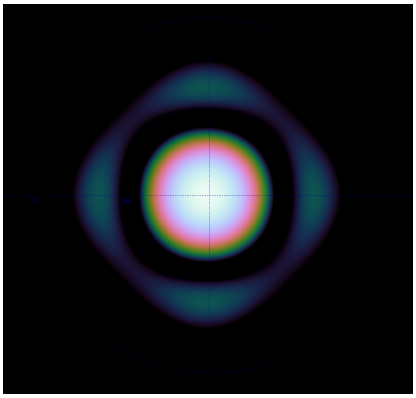

Figure 1: Primary Beam Focus Pattern[5]

2 Models and Algorithms

2.1 Tesselations

2.2 Voronoi Diagrams

A Voronoi Diagram is a partitioning of a space S by a set of points. Given n points the the space, $P = \{p_1, p_2, ..., p_n\}, P \subset S$, is partitioned into n regions, known as Voronoi Regions or Voronoi Cell, where every location, $s \in S_i, 0 \leq i \leq n-1$ in a region, $S_i \subset S$, is closest to a single point, $p_i \in P$, in terms of the space's distance measurement operation, d [3]. An example of a Voronoi Diagram can be seen in Figure 2.2.

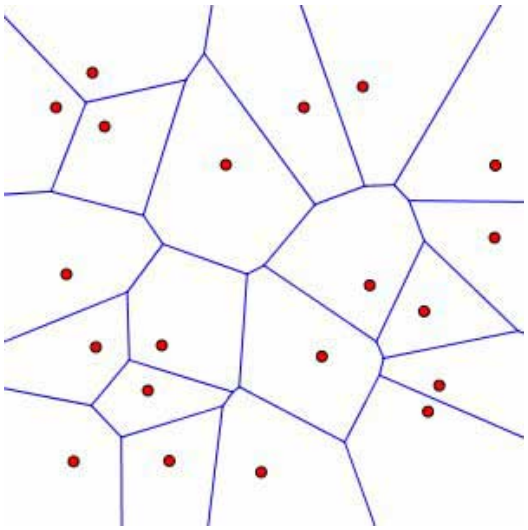

Figure 2: Voronoi Diagram[1]

2.3 Tesselation Generation Algorithms

2.3.1 Fortunes' Algorithm

2.3.2 K-Means Algorithm

3 GPU Architecture and Concepts

3.1 Parallelism

One of the main means of optimizing processing is through parallelism. The two main forms of parallelism are task and data parallelism. Task parallelism can be seen as running multiple processes concurrently where communication between the processes is explicitly defined to avoid race conditions. Data parallelism is the distribution of a data set over a number of identical processes each of which performs operations on a unique subset of the data. Race conditions occur when parallel processing streams access data or perform operations out of the intended order, leading to errors or incorrect output being produced. A combination of task and data parallelism can lead to an ideal speed-up, but both have their limits depending on the task and the data being operated on [6].

The increased need for parallelism came in about 2005, when CPU frequency peaked at 4GHz due to heat dissipation issues. However, Moore's Law still holds, and is still expected to hold until 2025; that is, that the number of transistors for a computer will double every two years. This leads to a problem where the speed at which an operation is done cannot be increased (due to the frequency limit), but the number of concurrent operations can still increase. This means that the only way to speed up an operation is to change it from a sequential to a parallel process [4].

3.2 GPU Optimization

GPU's were originally designed for rendering pixels and vectors in games. They were especially designed for this since CPU's are optimized to run sequential instructions as fast as possible; whereas pixel and vector calculations are inherently parallel. With NVIDIA's release of CUDA in 2006, general purpose GPU (GPGPU) programming became common place as a way to accelerate data processing through data parallelism and task parallelism through the simultaneous execution of similar task [2].

The power of GPU's come from its architecture which is optimized for a special case of SIMD (Single Instruction Multiple Data) processing known as SMIT (Single Instruction Multiple Threads). SIMD allows a central processor to distribute a set of instructions to multiple simple processors which then act on the data simultaneously. SIMT is more generalized as each thread of the GPU can perform different tasks given the same set of instructions. This is due to the way in which the GPU handles branching at the thread level. By exploiting these processes, and this instructional architecture, some instructions can be computed in faster time than that of a CPU [7]

4 Software

4.1 CUDA

4.2 C

4.3 Python

4.4 Anaconda

5 Related Works

5.1 Naieve Method

5.2 Standard Voronoi Model

6 Summary

References

[1] Furthest point voronoi diagrams: Voronoi diagrams. `http://www.ams.org/featurecolumn/images/august2006/diagramintro.1.jpg`. Accessed on: 26/02/2016.

[2] NVIDIA. CUDA. `http://www.nvidia.com/object/cuda_home_new.html`. Accessed on: 22/02/2016.

[3] Sugihara Okabe, Boots and Chiu. *Spatial tessellations: concepts and applications of Voronoi diagrams*, volume 501. John Wiley & Sons, 2009.

[4] Krishna Rajan. *Informatics for materials science and engineering: data-driven discovery for accelerated experimentation and application*. Butterworth-Heinemann, 2013.

[5] Oleg Smirnov. Personal Communication. SKA Chair at Rhodes Centre for Radio Astronomy Techniques & Technologies.

[6] O'hallaron Subhlok, Stichnoth and Gross. Exploiting task and data parallelism on a multicomputer. In *ACM SIGPLAN Notices*, volume 28, pages 13–22. ACM, 1993.

[7] Richard Vuduc and Jee Choi. A brief history and introduction to gpgpu. In *Modern Accelerator Technologies for Geographic Information Science*, pages 9–23. Springer, 2013.