

AN EFFICIENT REAL-TIME TERRAIN DATA ORGANIZATION AND
VISUALIZATION ALGORITHM BASED ON ENHANCED
TRIANGLE-BASED LEVEL OF DETAIL TECHNIQUE

MUHAMAD NAJIB BIN ZAMRI

A thesis submitted in fulfilment of the
requirements for the award of the degree of
Master of Science (Computer Science)

Faculty of Computer Science and Information System
Universiti Teknologi Malaysia

MAY 2006

*To
my beloved father and mother, Zamri Ismail and Rashidah Shariff;
my brothers, Azainizam, Shahirman and Hanafi;
my sisters, Anira, Aziha and Aizan; and
my friends*

ACKNOWLEDGEMENT

First of all, thanks to Allah SWT, the Creator, for giving me the strength to complete this thesis. I would like to express my appreciation to the Ministry of Science, Technology and Innovation (MOSTI) for providing me the financial support to further my education to this level under National Science Fellowship (NSF) program. Obviously, without their help, this journey would not have allowed.

I would like to express my sincere gratitude to my main supervisor, Associate Professor Safie Mat Yatim and my co-supervisor, Mr. Noor Azam Md. Sheriff, for their guidance and support through my studies as well as their constructive criticism of my research. This thesis would not have been possible without them. I would like to thank lecturers that involved and gave their brilliant ideas and solutions for this research especially Associate Professor Sarudin Kari, Associate Professor Daut Daman, Mr. Nik Isrozaidi Nik Ismail and Mr. Mohd Shafry Mohd Rahim.

Furthermore, I would also like to acknowledge my fellow postgraduate students, colleagues and housemates who have made my lives so enjoyable. They are Irwandi Hipni, Yusrin, Hamimah, Mun San, Weng, Baldave, Sandilian, Kim Heok, Adlur, Fendi, Amri, Mohd Hafiz, Asyraf and Asrulakmar. Special thanks go to foreign researchers that help me in explaining the related technical processes and developing the prototype system. They are Stefan Röttger, Mark Duchaineau, Peter Lindstrom, Chris Cookson, ZhaoYoubing, Thatcher Ulrich and Ulf Assarsson.

Finally, I am grateful to my parents and my family for their support and patience during my stay at the Universiti Teknologi Malaysia.

ABSTRACT

The massive volume of data involved in the development of a real-time terrain visualization and navigation system is inevitable. Based upon the current off-the-shelf hardware capability, it is impossible to process the amount of data using the conventional approach. This is due to the fact that the amount of data to be processed exceeds the capacity that can be loaded into main memory. This problem is further compounded by other hardware constraints such as memory bus speed and data transfer bandwidth from the main memory to the graphics card. Consequently, this limitation has affected the total system performance. The triangle-based level of detail (LOD) technique has been developed in order to reduce the drawback but it still suffers from the main memory constraint and slow data loading. The purpose of this research is to design, develop and implement an algorithm for enhancing the rendering efficiency of triangle-based LOD technique. A prototype system has been developed using digital elevation data for testing purposes. The system was evaluated based on the following criteria, namely: data size, processing time for data partitioning, memory usage, loading time, frame rate, triangle count and geometric throughput. According to the results obtained during the pre-processing, the partitioning of digital elevation data into tiles has successfully reduced the data size although it required a longer processing time. During the run-time processing, the integration of dynamic tile loading scheme, view frustum culling and enhanced triangle-based LOD technique has produced encouraging results with significant overall improvement as compared to the techniques that have been developed earlier. The proposed algorithm in this thesis is very practical in developing interactive graphics applications, which consider real-time rendering as one of the important elements.

ABSTRAK

Data bersaiz besar yang terlibat di dalam pembangunan sistem visualisasi dan navigasi bentuk muka bumi secara masa nyata memang tidak dapat dielakkan. Berdasarkan kepada keupayaan perkakasan terkini yang berada di pasaran, adalah mustahil untuk memproses data tersebut menggunakan pendekatan biasa. Ini disebabkan oleh amaun data yang diproses melebihi kapasiti yang boleh dimuatkan ke dalam ingatan utama. Masalah ini ditambah pula dengan kekangan-kekangan perkakasan lain seperti kelajuan ingatan bas dan lebar jalur penghantaran data dari ingatan utama ke kad grafik. Kesannya, limitasi ini telah mempengaruhi keseluruhan prestasi sistem. Teknik aras keperincian berasaskan segi tiga telah dibangunkan dalam usaha untuk mengurangkan kelemahan tetapi masih perlu berhadapan dengan kekangan ruang ingatan utama dan pemuatan data yang perlahan. Tujuan penyelidikan ini adalah untuk mereka bentuk, membangun dan mengimplementasi sebuah algoritma bagi meningkatkan keberkesanan perenderan teknik aras keperincian berasaskan segi tiga. Satu sistem prototaip telah dibangunkan dengan menggunakan data ketinggian digital untuk tujuan pengujian. Sistem ini telah dinilai berdasarkan kepada kriteria-kriteria berikut: saiz data, masa pemprosesan bagi pemecahan data, penggunaan ruang ingatan, masa pemuatan, kadar kerangka, serta bilangan dan kadar penjanaan segi tiga. Berdasarkan kepada keputusan yang diperolehi semasa pra-pemprosesan, pemecahan data ketinggian digital kepada blok-blok data kecil telah berjaya mengurangkan saiz data walaupun proses ini memerlukan masa pemprosesan yang lebih panjang. Semasa pemprosesan masa larian, gabungan skema pemuatan blok-blok data kecil yang dinamik, teknik penghapusan data berasaskan sudut pandangan dan teknik aras keperincian berasaskan segi tiga yang dipertingkatkan telah menghasilkan keputusan yang memberangsangkan dengan peningkatan keseluruhan yang signifikan jika dibandingkan dengan teknik-teknik yang telah dibangunkan sebelum ini. Algoritma yang dicadangkan di dalam tesis ini adalah sangat praktikal di dalam membangunkan aplikasi-aplikasi grafik yang bersifat interaktif di mana mempertimbangkan perenderan masa nyata sebagai salah satu daripada elemen penting.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
	TITLE PAGE	i
	DECLARATION OF ORIGINALITY AND EXCLUSIVENESS	ii
	DEDICATION	iii
	ACKNOWLEDGEMENT	iv
	ABSTRACT	v
	ABSTRAK	vi
	TABLE OF CONTENTS	vii
	LIST OF TABLES	xii
	LIST OF FIGURES	xiii
	LIST OF ALGORITHMS	xvi
	LIST OF ABBREVIATIONS	xviii
	LIST OF APPENDICES	xx
1	INTRODUCTION	1
	1.1 Introduction	1
	1.2 Problem Background	5
	1.2.1 Massive Terrain Dataset	5
	1.2.2 Geometric Complexity	6
	1.2.3 Limited Capability of Triangle-Based LOD Techniques	8
	1.3 Problem Statements	8
	1.4 Goal	8

1.5	Objectives	9
1.6	Scopes	9
1.7	Organization of the Thesis	10
2	LITERATURE REVIEW	11
2.1	Introduction	11
2.2	Spatial Data Structures	12
2.2.1	Kd-Tree	13
2.2.2	Quadtree	15
2.2.3	R-Tree	16
2.2.4	Tiled Block	18
2.2.5	Comparative Studies	19
2.3	Visibility Culling Methods	20
2.3.1	Types of Culling Methods	21
2.3.1.1	Backface Culling	21
2.3.1.2	Small Feature Culling	22
2.3.1.3	Occlusion Culling	23
2.3.1.4	View Frustum Culling	24
2.3.1.5	Comparative Studies	25
2.3.2	Intersection Test Techniques	26
2.3.2.1	Axis Aligned Bounding Box	27
2.3.2.2	Oriented Bounding Box	27
2.3.2.3	Bounding Sphere	29
2.3.2.4	k-Discrete Orientation Polytope	30
2.3.2.5	Comparative studies	31
2.4	View-Dependent LOD Techniques	32
2.4.1	Terrain Representation	32
2.4.1.1	Regular Grid	32
2.4.1.2	Triangulated Irregular Network	33
2.4.1.3	Advantages and Disadvantages	34

2.4.2	Geometric Primitives	35
2.4.2.1	Independent Triangles	35
2.4.2.2	Triangle Fans	36
2.4.2.3	Triangle Strips	36
2.4.2.4	Comparative Studies	37
2.4.3	Previous Works	37
2.4.3.1	Real-time Optimally Adapting Meshes	38
2.4.3.2	Real-time Generation of Continuous LOD for Height Fields	43
2.4.3.3	View-Dependent Progressive Meshes	45
2.4.3.4	Out-of-Core Terrain Visualization	48
2.4.3.5	Comparative Studies	52
2.5	Summary	53
3	METHODOLOGY	55
3.1	Introduction	55
3.2	Partitioning Terrain Data	58
3.2.1	Extracting Data	58
3.2.2	Splitting Data into Tiles	59
3.3	Loading Terrain Tiles	63
3.4	Culling Invisible Terrain Tiles	64
3.4.1	Plane Extraction	64
3.4.2	Intersection Test	67
3.5	Simplifying Geometry of Visible Terrain Tiles	68
3.5.1	Distance Calculation	69
3.5.2	Subdivision Test	69
3.5.3	Generation of Triangle Fans	70
3.6	Summary	71

4	IMPLEMENTATION	72
4.1	Introduction	72
4.2	Pre-Processing	73
4.2.1	Reading a DEM File and Creating TXT Files	73
4.2.2	Reading TXT Files and Creating TILE Files	79
4.3	Run-time Processing	87
4.3.1	Creation of Classes	87
4.3.2	Initialization	89
4.3.3	Rendering	89
4.3.3.1	Configuring Information for Camera	90
4.3.3.2	Loading and Unloading Terrain Tiles	90
4.3.3.3	Calculating Frustum's Planes	92
4.3.3.4	Removing Unseen Terrain Tiles	93
4.3.3.5	Generating Quadtree Traversal	94
4.3.3.6	Generating Triangle Fans	94
4.3.4	Interaction	96
4.3.5	Shutdown	96
4.4	Summary	97
5	RESULTS, ANALYSIS AND DISCUSSIONS	98
5.1	Introduction	98
5.2	System Specifications	99
5.3	Testing Data	100
5.4	Screenshots from the Prototype System	101
5.5	Evaluations of the System	103
5.5.1	Pre-Processing	103

5.5.2	Run-time Processing	105
5.5.2.1	Memory Usage	106
5.5.2.2	Loading Time	108
5.5.2.3	Frame Rate	109
5.5.2.4	Triangle Count	111
5.5.2.5	Geometric Throughput	113
5.6	Comparison with Previous Techniques	114
5.6.1	Memory Usage	115
5.6.2	Loading Time	116
5.6.3	Frame Rate	117
5.6.4	Triangle Count	118
5.7	Summary	120
6	CONCLUSION	121
6.1	Conclusion	121
6.2	Recommendations	124
	REFERENCES	126
	APPENDICES A - C	133 - 139

LIST OF TABLES

TABLE NO.	TITLE	PAGE
1.1	Types of busses and the corresponding memory bandwidth and geometric throughput (Hill, 2002; Peng <i>et al.</i> , 2004)	7
2.1	Advantages and disadvantages of visibility culling methods	25
2.2	Comparison of intersection test techniques	31
2.3	Comparison among the geometric primitives	37
2.4	Comparison of view-dependent LOD techniques	53
3.1	Clipping planes with corresponding co-efficients	66
4.1	Data structure	75
4.2	Declaration of variables	80
5.1	Hardware specifications	99
5.2	Software specifications	99
5.3	Testing data	100
5.4	Environment settings for testing the proposed algorithm	101
5.5	The results of time required for converting DEM data and the corresponding file size for number of vertices per tile: 129 x 129 and 257 x 257	103
5.6	Number of generated tiles	104
5.7	Environment settings for comparison purpose	115

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE
1.1	Visualization of full resolution terrain model (number of vertices = 16,848,344,430 and number of triangles = 33,695,786,468)	6
2.1	Kd-tree	14
2.2	Quadtree	15
2.3	R-tree	17
2.4	Partitioning data into tiles	18
2.5	Tree structure for tiles	19
2.6	Vector angle between view direction and polygons normal	21
2.7	Small feature culling a) OFF, b) ON	22
2.8	Occlusion culling of occluded object	23
2.9	Camera's view frustum	24
2.10	The red objects lie completely outside the view- frustum and will be discarded from rendering when view frustum culling is performed	24
2.11	AABB of an object	27
2.12	OBB of a rotated object	28
2.13	BS of an object	29
2.14	Aircraft model and the corresponding k-dops a) Level 1, b) Level 2, c) Level 3, d) Level 4	30
2.15	Variations of grid representations	33
2.16	TIN representation	33
2.17	Independent triangles	35
2.18	Triangle fan	36

2.19	Triangle strip	36
2.20	Levels 0-5 of a triangle bintree	38
2.21	Split and merge operations on a bintree triangulation	39
2.22	Forced splitting process	39
2.23	Wedgies in ROAM implementation	40
2.24	A sample triangulation of a 9 x 9 height field a) quadtree matrix, b) the corresponding quadtree structure	43
2.25	Measuring surface roughness	44
2.26	Vertex split and edge collapse operations	45
2.27	Steps in pre-processing block-based simplification	46
2.28	Interleaved quadtree structure	49
2.29	Four levels of longest-edge bisection operation	50
3.1	Research framework	56
3.2	Overlapping tile	62
3.3	Dynamic load management	63
3.4	Six planes of view frustum	64
3.5	Plane-sphere intersection test for view frustum culling technique a) sample scene, b) intersection test, c) result after culling	68
3.6	Distance (l) and edge length (d)	69
3.7	Generation of triangle fans	70
3.8	Terrain tiles after applying view-dependent LOD technique	71
4.1	Menu for partitioning DEM data	73
4.2	Structure of a DEM file	74
4.3	First data point for DEM data a) West of the central meridian b) East of the central meridian	77
4.4	Parameters after extraction of important header information	78
4.5	The indices and the corresponding elevation values	79

4.6	List of generated TILE files	85
4.7	Structure in a TILE file	86
4.8	Class diagram	88
4.9	Sixteen possibilities of generating triangle fans	95
5.1	Small dataset a) wireframe, b) textured	102
5.2	Moderate dataset a) wireframe, b) textured	102
5.3	Large dataset a) wireframe, b) textured	102
5.4	Total time for converting DEM datasets to TILE files	104
5.5	Memory usage for the proposed system	106
5.6	The results of memory usage	107
5.7	The results of loading time	108
5.8	The results of average frame rate based on different FOV	109
5.9	The results of average frame rate based on different length of view frustum	110
5.10	The results of average triangle count based on different FOV	111
5.11	The results of average triangle count based on different length of view frustum	111
5.12	The results of frame rate versus triangle count	112
5.13	The results of average geometric throughput based on different FOV	113
5.14	The results of average geometric throughput based on different length of view frustum	114
5.15	Comparison based on memory usage	116
5.16	Comparison based on loading time	117
5.17	Comparison based on frame rate	118
5.18	Comparison based on triangle count	119

LIST OF ALGORITHMS

ALGORITHM NO.	TITLE	PAGE
2.1	Pseudo code for split queue	41
2.2	Pseudo code for merge queue	42
2.3	Pseudo code for selective refinement	47
2.4	Pseudo code for force vsplit	48
2.5	Pseudo code for mesh refinement (<i>mesh-refine</i>)	51
2.6	Pseudo code for sub mesh refinement (<i>submesh-refine</i>)	51
2.7	Pseudo code for generating triangle strips (<i>trianglestrip-append</i>)	52
4.1	Pseudo code for obtaining maximum number of vertices per side for original terrain data	81
4.2	Pseudo code for assigning new maximum number of vertices per side for terrain data	81
4.3	Pseudo code for determining the maximum number of vertices per side for tiles in vertical direction (V_{t,v_max})	82
4.4	Pseudo code for obtaining starting point and existence status for each tile	83
4.5	Pseudo code for partitioning data and creating TILE files	84
4.6	Pseudo code for obtaining starting coordinates of existing tile	85

4.7	Pseudo code for obtaining the current tile index based on camera position	91
4.8	Pseudo code for obtaining the terrain indices to be loaded	92
4.9	Pseudo code for detecting the visibility status and assigning flag for loaded tiles	93

LIST OF ABBREVIATIONS

ABBREVIATION	DESCRIPTION
2D	- Two-dimensional
3D	- Three-dimensional
AABB	- Axis Aligned Bounding Box
API	- Application Programming Interface
BS	- Bounding Sphere
CBIS	- Computer-based information system
CFD	- Computational fluid dynamics
CPU	- Central processing unit
DEM	- Digital Elevation Model
DSS	- Decision Support System
DTED	- Digital Terrain Elevation Data
ESS	- Executive Support System
FOV	- Field of view
FPS	- Frames per second
GIS	- Geographic Information System
GPU	- Graphics processing unit
GRASS	- Geographic Resources Analysis Support System
IS	- Information system
KMS	- Knowledge Management System
LOD	- Level of detail
MBB	- Minimal bounding box
MBR	- Minimal bounding rectangle
MIS	- Management Information System
NE	- Northeast

NW	-	Northwest
OAS	-	Office Automation System
OBB	-	Oriented Bounding Box
OS	-	Operating system
PC	-	Personal computer
RAPID	-	Rapid and Accurate Polygon Interference Detection
ROAM	-	Real-time Optimally Adapting Meshes
SE	-	Southeast
SW	-	Southwest
TIN	-	Triangulated irregular network
USGS	-	United States Geological Survey
UTM	-	Universal Transverse Mercator
VDPM	-	View-dependent progressive meshes
ViRGIS	-	Virtual Reality GIS

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
A	DEM Elements Logical Record Type A	133
B	DEM Elements Logical Record Type B	138
C	List of Published Paper Works	139

CHAPTER 1

INTRODUCTION

1.1 Introduction

Computer-based information system (CBIS) has changed the people works from traditional manual system to an automation system that provides more effective and systematic way in solving specific problems. In addition, the works can be done and completed much faster than the previous approach. Thus, time consumption can be reduced and managed efficiently. This is due to the rapid development of computer's technology over the last few years. Several types of CBISs have been developed and those systems have contributed to the evolution of information system. Among of them are Management Information System (MIS), Office Automation System (OAS), Executive Support System (ESS), Decision Support System (DSS), Knowledge Management System (KMS), Geographic Information System (GIS) and Expert System (Gadish, 2004).

Geographic information system (GIS) is one of the important developments of CBIS that combines digital map and geographic data; and facilitate assessments of related characteristics of people, property and environmental factors. In general, GIS is a set of computer tools for collecting, storing, retrieving, transforming and displaying data from the real world for a particular set of purposes or application domains (Burrough and McDonnell, 1998). There are two main elements in developing GIS applications: database and visualization. This system provides an opportunity to store both kinds of spatial data (location-based data) and non-spatial

data (attribute data) in a single system. In current GIS applications, these two types of data are often stored in two separate databases. However, these databases are linked or joined together later depending on the particular characteristic.

Computer graphics disciplines are exploited in visualizing the GIS data. It is useful for analyzing and verifying the related activities in terms of accuracy, efficiency and other factors. For this reason, GIS is becoming increasingly important to a growing number of activities in many sectors including urban planning, flood planning, oil exploration, evaluations of vegetation, soil, waterway and road patterns (Fritsch, 1996; Burrough and McDonnell, 1998).

Lacking in several aspects such as user interaction (limited to panning and zooming operations), presentation (points, lines and symbols) and interpretation in two-dimensional GIS (2D GIS) was lead to the development of higher dimensional GIS especially three-dimensional GIS (3D GIS), which is also known as Virtual GIS (Burrough and McDonnell, 1998). The use of colors and symbols representation in previous GIS is replaced with 3D photo-textured models, which improves the level of understanding of the system. Moreover, Virtual GIS has the ability to have detailed 3D views and provide real-time navigation whether fly-through or walkthrough for analyzing large volumes of spatial data and related information. At the same time, the operations used in 2D GIS can be integrated easily into the system (Noor Azam and Ezreen Elena, 2003).

Real-time terrain visualization is one of the subfield in GIS that attracts a number of researchers involved in this area. Normally, terrain and the other objects such as trees, buildings and roads are managed separately due to their different data representation. Speed and realism are two factors frequently considered in solving related terrain problems. At present, there are more than 800 commercial terrain visualization software have been developed by different vendors and developers as reported by United States Army Topographic Engineering Center (2005). Among the popular software are ESRI[®] ArcGIS, ERDAS[®] Imagine VirtualGIS, Geographic Resources Analysis Support System (GRASS) and Manifold System.

In this thesis, the research focuses on real-time 3D terrain data organization, visualization and on-the-fly navigation. There are two key issues have to be encountered in developing terrain-based applications in GIS field: spatial database management and fast interactive visualization (Kofler, 1998). These two obstacles take into account with the capabilities of current hardware technology that supports the system to be developed. Large amount of terrain dataset need to be handled efficiently in real-time environment with restricted main memory capacity. These include the management of geometric and texture data. At the same time, the interactive frame rate needs to be achieved consistently when facing with too many polygons generated for each frame. The detailed information on problem background is covered in Section 1.2.

There are many solutions have been proposed by a number of researchers in the last few years. Among of them are the implementation of hierarchical data structure, visibility culling methods and level of detail (LOD) techniques. Briefly, the most common types of data structures can be employed to manage data are Kd-tree (Bentley, 1975; Bittner *et al.*, 2001; Langbein *et al.*, 2003), quadtree (Samet, 1984; Röttger *et al.*, 1998; Pajarola, 1998), R-tree (Guttman, 1984; Kofler, 1998; Yu *et al.*, 1999; Zlatanova, 2000) and tiled block (VTP, 1999; Lindstrom *et al.*, 1996; Pajarola, 1998; Ulrich, 2002). The difference is depending on how the space (data) is divided in hierarchical manner (Gaede and Günther, 1998).

For visibility culling, there are four types of methods can be adopted: backface culling (Laurila, 2004), small feature culling (Burns and Osfield, 2001; Assarsson, 2001), occlusion culling (Stewart, 1997; Mortensen, 2000; Martens, 2003) and view frustum culling (Hoff, 1996; Assarsson and Möller, 2000). These are useful for removing the unneeded data from being processed. One of the important components in culling methods is the intersection test. The related research works include Axis Aligned Bounding Box (Beckmann *et al.*, 1990; Morley, 2000), Oriented Bounding Box (Gottschalk *et al.*, 1996; Assarsson and Möller, 2000), Bounding Sphere (Hubbard, 1995; Dunlop, 2001; Morley, 2000; Picco, 2003) and k-Discrete Orientation Polytope (Klosowski *et al.*, 1998).

LOD techniques are used to reduce the number of polygons and give details to the certain area in a scene. The examples of well-known techniques are Real-time Optimally Adapting Meshes (Duchaineau *et al.*, 1997), Real-time Generation of Continuous LOD (Röttger *et al.*, 1998), View-Dependent Progressive Meshes (Hoppe, 1998) and Out-of-core Terrain Visualization (Lindstrom and Pascucci, 2001). Although the LOD technique can reduce the problem complexity but lots of polygons that are out-of-sight still being rendered, thus deteriorating the system performance. Most of the abovementioned researchers have integrated their system with culling method at vertices or polygon level in order to solve the problem. However, lots of computations involved in implementing such method especially when facing with large amount of datasets.

In this thesis, a set of techniques has been designed and proposed for handling massive terrain dataset in interactive rate. The idea behind this research is to expand the capability of triangle-based LOD technique and improve the performance of real-time terrain visualization and navigation system. Triangle-based LOD technique has been chosen for this research because it has potential to remove lots of terrain geometry and capable to speed up the system at interactive rates without losing the visual appearance of visualization part (Duchaineau *et al.*, 1997; Hoppe, 1998; Pajarola, 1998; Röttger *et al.*, 1998; Lindstrom and Pascucci, 2001).

There are two separate phases involved, which are pre-processing and run-time processing. In first phase, the specific terrain data is converted to the internal data structure that is based on tile representation. The original terrain data is partitioned into several small tiles or patches. In run-time, several tiles are loaded depending on camera position. Then, these tiles are tested for determining the visibility status to remove the unseen block of data. Finally, the geometric simplification of visible terrain tiles is done which is based on triangle-based LOD technique in order to minimize the number of triangles to be processed and being rendered.

1.2 Problem Background

There are three challenging issues in developing terrain-based applications. These involve the massive terrain datasets, geometric complexity and capability of current triangle-based LOD techniques.

1.2.1 Massive Terrain Dataset

Various types of terrain datasets are widely being used in GIS applications, including Digital Elevation Model (DEM), Digital Terrain Elevation Data (DTED), satellite data and aerial photographs. Commonly, terrain consists of geometry and imagery data. Geometric data represents the location or coordinate of related terrain's vertices, while imagery data represents as a texture map for producing realistic 3D model. In fact, large amount of this terrain dataset have to be faced when dealing with GIS applications. It is not surprising if the size of dataset will occupy a number of Gigabytes or even Terabytes of storage (Gaede and Günther, 1998).

The use of this massive dataset will affect the capability of main memory to load and store data for visualization. A number of researchers have pointed out that it is impossible to hold the entire detailed dataset at once (Kofler, 1998; Pajarola, 1998). Besides that, accessing this data quickly enough to use it in real-time rendering is a difficult problem. It will limit the performance of the system during interactive visualization. This is because when the new frame is generated, all polygons will be displayed whether they are within the view frustum or not. It is time consuming to compute and render the unneeded data continuously in real-time (Hill, 2002).

1.2.2 Geometric Complexity

For any detailed terrain model, the number of triangles that must be rendered is prohibitively high. For instance, a 100 km x 100 km terrain sampled at 1-meter resolution and represented as a regular grid will contain roughly 20,000,000,000 triangles. At present, it is not be able to render anywhere near this full resolution data in real-time. According to the research that has been done by Hill (2002) in managing large terrains, rendering 20,000,000,000 triangles per second would not be feasible until 2010. Furthermore, rendering this many triangles at 60 frames per second would not be possible until 2016. Figure 1.1 shows an example of visualizing original terrain model with billion of vertices and polygons. The white line shows the triangles generated in wireframe mode and the black color is a background of a scene.

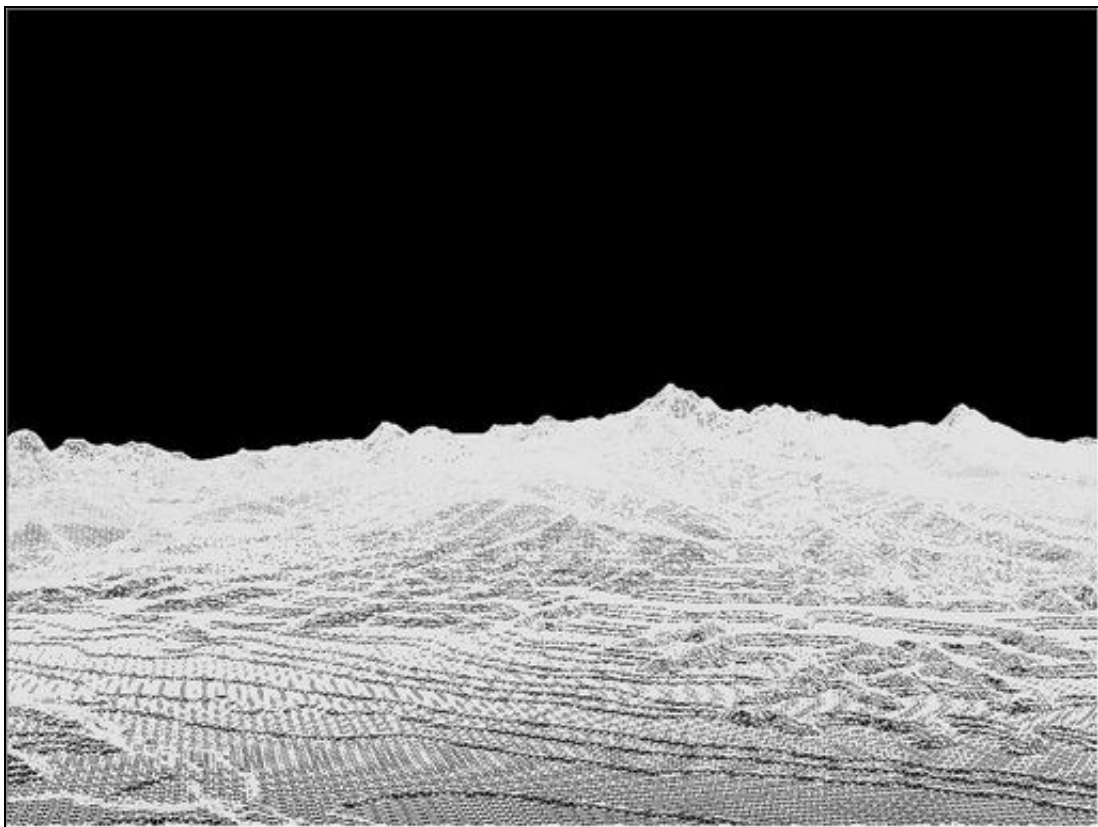


Figure 1.1 Visualization of full resolution terrain model (number of vertices = 16,848,344,430 and number of triangles = 33,695,786,468)

With current graphics hardware, all communications between the card and the CPU takes place over the memory bus. Presently, the primary bottleneck related to rendering performance is the memory bandwidth on the bus. The graphics hardware is capable of rendering substantially more geometry per second than can be read in over the bus. In order to render 60 million triangles per second, over 5 Gigabytes of data would need to be transferred to the card each second. According to the maximum memory bandwidth and million triangles per second can be transferred to the graphics card as shown in Table 1.1, it is impossible to achieve the maximum rendering speed directly. Moreover, this bandwidth is also used for transferring texture data to the card, so the 60 million triangles per second target are even further out of reach (Hoppe, 1999; Hill, 2002).

Table 1.1: Types of busses and the corresponding memory bandwidth and geometric throughput (Hill, 2002; Peng *et al.*, 2004)

Type of Bus	Bandwidth Limitation (MB/s)	Geometric Throughput (Million triangles per second)
PCI	133	1.385
PCI64	533	5.552
AGP 1x	266	2.771
AGP 2x	533	5.552
AGP 4x	1066	11.104
AGP 8x	2150	22.396
PCI Express	4300	44.792

1.2.3 Limited Capability of Triangle-Based LOD Techniques

In order to deal with so much dataset, level of detail (LOD) techniques need to be employed. Theoretically, terrain close to the viewer must be presented with more geometry than terrain that far away from the viewer; and rough terrain requires more geometry than flat terrain does. Frequently, triangle-based LOD algorithm is applied for simplifying the terrain geometry. Although this technique provides fast visualization through polygon simplification in real-time but it is not practical for very large terrain dataset because the amount of data to be loaded exceeds the data that can be stored into main memory. Even if the system is able to load the data, it is time expensive for waiting the loading process completed (Hill, 2002).

1.3 Problem Statements

According to the problems mentioned earlier, the following question needs to be solved in this research:

How to store and display large amount of terrain datasets efficiently for real-time visualization and navigation system with restricted capacity of current main memory and memory bus bandwidth?

1.4 Goal

The research goal is to expand the capabilities of triangle-based LOD technique by developing a set of techniques using hybrid method that will enhance the efficiency of real-time 3D terrain visualization and navigation for the development of Virtual GIS.

1.5 Objectives

In order to accomplish the main goal, several objectives have to be fulfilled.

These include:

- (i) To design and construct an efficient data structure that will be able to store arbitrary-sized terrain model.
- (ii) To develop an algorithm that will be able to provide fast data loading and reduce the memory usage in run-time.
- (iii) To develop a culling technique that will be able to remove large amount of terrain geometry.
- (iv) To integrate aforementioned data structure, loading algorithm and culling technique with the chosen triangle-based LOD technique into the real-time terrain visualization and navigation system.

1.6 Scopes

The research focuses on managing large-scale terrain dataset in real-time.

Several scopes need to be considered. These include:

- (i) The system manages static terrain dataset. It does not include the buildings, trees, rocks and other related objects.
- (ii) The system focuses on managing terrain geometry. Imagery terrain data is only used as a texture map for enhancing the visual appearance.
- (iii) Terrain datasets are the regular grid representation.
- (iv) Triangle-based LOD technique is based on view-dependent LOD technique.
- (v) The system is developed in real-time environment, which involves user interaction through computer's keyboard and mouse on desktop PC.

1.7 Organization of the Thesis

The thesis is organized into six chapters. Chapter 1 gives a general picture on related research fields, problem background and existing solutions involved in developing real-time terrain visualization and navigation in Virtual GIS. Besides that, the goal, objectives and scopes of this research have been specified.

Chapter 2 provides surveys on spatial data structures, visibility culling methods, intersection test techniques and view-dependent LOD techniques.

Chapter 3 presents the methodology of this research, which consists of four main steps. The explanations include partitioning original terrain data, loading tiles, culling invisible tiles and simplifying geometry of visible tiles.

Chapter 4 describes the implementation details of prototype system from pre-processing to run-time processing. For pre-processing, there are two pairs of steps have been explained, which are reading a DEM file and creating TXT files, and reading TXT files and creating TILE files. For run-time processing, the section has been divided into five subsections: creation of classes, initialization, rendering, interaction and shutdown procedure.

Chapter 5 presents and discusses the results obtained from the implementation phase. System specifications and testing data have been specified. Subsequently, the evaluations of the system and comparison among the existing technique have been explained in further details.

Chapter 6 summarizes the thesis and includes recommendations of future works that can be extended with the prototype system.

Moore's law for CPU. Thus, the implementation of real-time terrain system using this processing can be well-suited for the next few years.

(iv) Implementing parallel processing

The prototype system developed in this research can be expanded to the networking environment. By adopting the client-server communication, parallel processing can be done in order to obtain higher level of system performance. The use of load balancing algorithm can be used to distribute the job efficiently based on priority and burden of processing power.

REFERENCES

- Ahn, H.K., Mamoulis, N. and Wong H.M. (2001). *A Survey on Multidimensional Access Methods*. COMP630c - Spatial, Image and Multimedia Databases: Technical Report UU-CS-2001-14.
- Assarsson, U. (2001). *Detail Culling for Robotic Products*. ABB. <http://www.abb.com/global/abbzh/abbzh251.nsf!OpenDatabase&db=/global/seitp/seitp161.nsf&v=17ec2&e=us&c=76133777536BC959C12569CE006C6EE3>
- Assarsson, U. and Möller, T. (2000). Optimized View Frustum Culling Algorithms for Bounding Boxes. *Journal of Graphics Tools*. 5(1): 9-22.
- Beckmann, N., Kriegel, H.P., Schneider, R. and Seeger, B. (1990). The R*-tree: an efficient and robust access method for points and rectangles. *Proceedings of ACM SIGMOD International Conference on the Management of Data*. May 23-25. New York, USA: ACM Press, 322-331.
- Bentley, J.L. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM*. 18(9): 509-517.
- Bittner, J., Wonka, P. and Wimmer, M. (2001). Visibility Preprocessing for Urban Scene using Line Space Subdivision. *Proceedings of Pacific Graphics 2001*. October 16-18. Tokyo, Japan, 276-284.
- Blekken, P. and Lilleskog, T. (1997). *A Comparison of Different Algorithms for Terrain Rendering*. Spring semester project. CS Dept., Norwegian U. of Science and Technology. unpublished.
- Bradley, D. (2003). *Evaluation of Real-time Continuous Terrain Level of Detail Algorithms*. Charlton University: COM4905 Honours Project.
- Burns, D. and Osfield, R. (2001). *Introduction to the OpenSceneGraph*. OSG Community. <http://www.openscenegraph.com>
- Burrough, P.A. and McDonnell, R.A. (1998). *Principles of Geographical Information Systems*. New York: Oxford University Press Inc.

- CGL (2004). *Backface Culling*. CS488/688: Introduction to Interactive Computer Graphics. Computer Graphics Lab, University of Waterloo.
<http://medialab.di.unipi.it/web/IUM/waterloo/node66.html>
- DeFloriani, L., Magillo, P. and Puppo, E. (2000). VARIANT: A System for Terrain Modeling at Variable Resolution. *GeoInformatica*. 4(3), 287-315.
- Duchaineau, M., Wolinsky, Sigeti, D.E., Miller, M.C., Aldrich, C., and Mineev-Weinstein, M.B. (1997). ROAMing Terrain: Real-time optimally adapting meshes. *Proceedings of the 8th Conference on Visualization '97*. October 18-24. Los Alamitos, CA, USA: IEEE Computer Society Press, 81-88.
- Dunlop, R. (2001). *Collision Detection, Part 1: Using Bounding Spheres*. Microsoft DirectX MVP. http://www.mvps.org/directx/articles/using_bounding_spheres.htm
- Fagin, R., Nievergelt, J., Pippenger, N. and Strong, R. (1979). Extendible Hashing: A Fast Access Method for Dynamic Files. *ACM Transactions Database Systems*. 4(3): 315-344.
- Faloutsos, C. (1988). Gray Codes for Partial Match and Range Queries. *IEEE Transactions on Software Engineering*. 14(10): 1381-1393.
- Faloutsos, C. and Roseman, S. (1989). Fractals for Secondary Key Retrieval. *Proceedings of the 8th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*. March. New York, USA: ACM Press, 247-252.
- Fritsch, D. (1996). Three-Dimensional Geographic Information Systems – Status and Prospects. *Proceedings of International Archives of Photogrammetry and Remote Sensing*. July 12-18. Vienna, Austria: ISPRS, 215-221.
- Gadish, D. (2004). *Information System for Management*. CIS 500 Summer 2004: Slide show.
- Gaede, V. and Günther, O. (1998). Multidimensional Access Methods. *ACM Computing Surveys*. 30(2): 170-231.
- Gottschalk, S., Lin, M.C. and Manocha, D. (1996). OBBTree: A Hierarchical Structure for Rapid Interference Detection. *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*. New York, USA: ACM Press, 171-180.

- Gribb, G. and Hartmann, K. (2001). Fast Extraction of Viewing Frustum Planes from the World-View-Projection Matrix. unpublished.
- Guttman, A. (1984). R-trees: A Dynamic Index Structure for Spatial Searching. *Proceedings of SIGMOD '84*. Jun 18-21. Boston, MA: ACM Press, 47-57.
- Haines, E. (2001). *3D Algorithmic Optimizations*. California, USA: SIGGRAPH 2001 Course.
- Hawkins, K. and Astle, D. (2002). *OpenGL Game Programming*. Ohio, USA: Premier Press.
- He, Y. (2000). *Real-time Visualization of Dynamic Terrain for Ground Vehicle Simulation*. University of Iowa: Ph.D. Thesis.
- Hill, D. (2002). *An Efficient, Hardware-Accelerated, Level-of-Detail Rendering Technique for Large Terrains*. University of Toronto: M.Sc. Thesis.
- Hjaltason, G.R. and Samet, H. (1995). Ranking in Spatial Databases. *Proceedings of the 4th International Symposium on Advances in Spatial Databases*. August 6-9. London, UK: Springer-Verlag, 83-95.
- Hoff, K. (1996). A Fast Method for Culling of Oriented-Bounding Boxes (OBBs) Against a Perspective Viewing Frustum in Large Walkthrough Models. <http://www.cs.unc.edu/hoff/research/index.html>
- Hoppe, H. (1998). Smooth View Dependant Level-of-Detail Control and its Application to Terrain Rendering. *Proceedings of IEEE Visualization 1998*. July 29-31. Los Alamitos, CA, USA: IEEE Computer Society Press, 35-42.
- Hoppe, H. (1999). Optimization of mesh locality for transparent vertex caching. *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*. New York, USA: ACM Press, 269-276.
- Hubbard, P.M. (1995) Collision Detection for Interactive Graphics Applications. *IEEE Transactions on Visualization and Computer Graphics*. 1(3): 218-230.
- Jagadish, H.V. (1990). Linear Clustering of Objects with Multiple Attributes. *Proceedings of the ACM SIGMOD International Conference on Management of Data*. May 23-26. New York, USA: ACM Press, 332-342.
- Kim, S.H. and Wohn, K.Y. (2003). TERRAN: out-of-core Terrain Rendering for Real time Navigation, *Proceedings of EUROGRAPHICS 2003*. September 1-6. New York, USA: ACM Press.

- Klosowski, J.T., Held, M., Mitchell, S.B., Sowizral, H. and Zikan, K. (1998). Efficient Collision Detection Using Bounding Volume Hierarchies of k -DOPs. *IEEE Transaction on Visualization and Computer Graphics*. 4(1): 21-36.
- Kofler, M. (1998). *R-trees for Visualizing and Organizing Large 3D GIS Databases*. Technischen Universität Graz: Ph.D. Thesis.
- Kumler, M.P. (1994). An Intensive Comparison of Triangulated Irregular Networks (TINs) and Digital Elevation Models (DEMs). *Cartographica Monograph* 45. 31(2): 1-99.
- Langbein, M., Scheuermann, G. and Tricoche, X. (2003). An Efficient Point Location Method for Visualization in Large Unstructured Grids. *Proceedings of 8th International Fall Workshop on Vision, Modeling and Visualization 2003*. November 19-21. Munich, German: Aka GmbH, 27-35.
- Larson, P.A. (1980). Linear Hashing with Partial Expansions. *Proceedings of the 6th International Conference on Very Large Data Bases*. October 1-3. Washington, DC, USA: IEEE Computer Society, 224-232.
- Laurila, P. (2004). *Geometry Culling in 3D Engines*. GameDev.net, LLC.
<http://www.gamedev.net/reference/programming/features/culling/default.asp>
- Lindstrom, P. and Pascucci, V. (2001). Visualization of large terrains made easy. *Proceedings of IEEE Visualization 2001*. October 24-26. Washington, DC, USA: IEEE Computer Society Press, 363-371.
- Lindstrom, P., Koller, D., Ribarsky, W., Hodges, L.F., Bosch, A.O. and Faust, N. (1997). *An Integrated Global GIS and Visual Simulation System*. Georgia Institute of Technology: Technical Report GIT-GVU-97-07.
- Lindstrom, P., Koller, D., Ribarsky, W., Hodges, L.F., Turner, G.A. and Faust, N. (1996). Real-time, continuous level of detail rendering of height fields. *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*. New York, USA: ACM Press, 109-118.
- Litwin, W. (1980). Linear Hashing: A New Tool for File and Table Addressing. *Proceedings of the 6th International Conference on Very Large Data Bases*. October 1-3. Washington, DC, USA: IEEE Computer Society, 212-223.

- Luebke, D., Reddy, M., Cohen, J.D., Varshney, A., Watson, B. and Huebner, R. (2002). *Level of Detail For 3D Graphics*. San Francisco: Morgan Kaufmann Publishers.
- Martens, R. (2003). *Occlusion Culling for the Real-time Display of Complex 3D Models*. Limburgs Universitair Centrum, Belgium: M.Sc. Thesis.
- Mokbel, M.F., Aref, W.G. and Kamel, I. (2002). Performance of Multi-Dimensional Space-Filling Curves. *Proceedings of the 10th ACM International Symposium on Advances in Geographic Information Systems*. November 8-9. New York, USA: ACM Press, 149-154.
- Morley, M. (2000). *Frustum culling in OpenGL*. <http://www.markmorley.com>
- Mortensen, J. (2000). *Real-time rendering of height fields using LOD and occlusion culling*. University College London, UK: M.Sc. Thesis.
- Nievergelt, J., Hinterberger, H. and Sevcik, K. (1984). The Grid File: An Adaptable, Symmetric Multikey File Structure. *ACM Transactions Database Systems*. 9(1): 38-71.
- Noor Azam, M. S. and Ezreen Elena, A. (2003). 3D Virtual GIS Architecture. *Proceedings of Advanced Technology Congress 2003*. May 20-21. Putrajaya, Selangor: The Institute of Advanced Technology (ITMA).
- Ögren, A. (2000). *Continuous Level of Detail In Real-Time Terrain Rendering*. Umea University: M.Sc. Thesis.
- Orenstein, J.A. and Merrett, T.H. (1984). A Class of Data Structures for Associative Searching. *Proceedings of the 3rd ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*. April 2-4. New York, USA: ACM Press, 181-190.
- Pajarola, R. (1998). *Access to Large Scale Terrain and Image Databases in Geoinformation Systems*. Swiss Federal Institute of Technology (ETH) Zurich: Ph.D. Thesis.
- Peng, W., Petrovic, D. and Crawford, C. (2004). Handling Large Terrain Data in GIS. *Proceedings of the XXth ISPRS Congress – Commission IV*. XXXV(B4): 281-286.
- Picco, D. (2003). *Frustum Culling*. FLIPCODE.COM, INC.
http://www.flipcode.com/articles/article_frustumculling.shtml
- Polack, T. (2003). *Focus on 3D Terrain Programming*. Ohio, USA: Premier Press.

- Röttger, S., Heidrich, W., Slusallek, P. and Seidel, H.P. (1998). Real-Time Generation of Continuous Levels of Detail for Height Fields. *Proceedings of WSCG '98*. February 9-13. Plzen, Czech Republic: University of West Bohemia, 315-322.
- Samet, H. (1984). The Quadtree and Related Hierarchical Data Structure. *ACM Computing Survey*. 16(2): 187-260.
- Samet, H. (1990). *The Design and Analysis of Spatial Data Structures*. Reading, MA: Addison-Wesley.
- Shade, J.W. (2004). *View-Dependent Image-Based Techniques for Fast Rendering of Complex Environments*. University of Washington: Ph.D. Thesis.
- Shen, H.W. (2004). *Spatial Data Structures and Culling Techniques*. Slide Show - CIS 781.
- Sheng, J.M. and Liu Sheng, O.R. (1990). R-trees for large geographic information systems in a multi-user environment. *Proceedings of the 23rd Annual Hawaii International Conference on System Sciences*. January 2-5. Washington, DC, USA: IEEE Computer Society Press, 10-17.
- Stewart, A.J. (1997). Hierarchical Visibility in Terrains. *Proceedings of the Eurographics Workshop on Rendering Techniques '97*. June 16-18. London, UK: Springer-Verlag, 217-228.
- Tamminen, M. (1982). The Extendible Cell Method for Closest Point Problems. *BIT* 22. 27-41.
- Ulrich, T. (2002). Rendering Massive Terrains using Chunked Level of Detail Control. *Course Notes of ACM SIGGRAPH 2002*. July 21-26. Texas, USA: ACM Press, Course 35.
- United States Army Topographic Engineering Center (2005). *Survey of Terrain Visualization Software*. Alexandria (U.S.): Technical Report.
- USGS (1992). *Standards for Digital Elevation Models*. U.S. Department of the Interior, United States Geological Survey, National Mapping Division, USA: National Mapping Program Technical Instructions.
- VTP (1999). *Applying Ground Textures*. Virtual Terrain Project.
http://www.vterrain.org/Textures/ground_textures.html
- Woo, M., Neider, J., Davis, T. and Shreiner, D. (1999). *OpenGL Programming Guide*. Canada: Addison-Wesley Longman, Inc.

- Youbing, Z., Ji, Z., Jiaoyang, S. and Zhigeng, P. (2001). A Fast Algorithm for Large Scale Terrain Walkthrough. *Proceedings of the International Conference on CAD & Graphics*. August 20-22. Kunming, China.
- Yu, H., Mehrotra, S. Ho, S.S., Gregory, T.C. and Allen, S.D. (1999). Integration of SATURN System and VGIS. *Proceedings of the 3rd Symposium on the Federated Lab on Interactive and Advanced Display*. February 2-4. Aberdeen, MD: Army Research Lab, 59-63.
- Zlatanova, S. (2000). *3D GIS for Urban Development*. Graz University of Technology: Ph.D. Thesis.