OUT-OF-CORE SIMPLIFICATION WITH APPEARANCE PRESERVATION FOR COMPUTER GAME APPLICATIONS

TAN KIM HEOK

UNIVERSITI TEKNOLOGI MALAYSIA

OUT-OF-CORE SIMPLIFICATION WITH APPEARANCE PRESERVATION FOR COMPUTER GAME APPLICATIONS

TAN KIM HEOK

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

JANUARY 2006

To my beloved parents and family

ACKNOWLEDGEMENTS

I am most grateful for the guidance and support from my supervisor, Associate Professor Daut Daman. During these years of doing research together, he has been an outstanding advisor. I thank him for motivating and inspiring me to pursue good and relevant research.

I would like to thank Dr. Siti Mariyam for evaluating my first assessment proposal and giving her precious opinions to me. Her insightful comments have made me more confident in continuing my research.

Without an assortment of polygonal models, there would have been nothing to test my algorithms on. Therefore, I wish to thank Stanford graphics group and Cyberware organization for providing several polygonal datasets available.

Special thanks are dedicated to Peter Lindstrom, Hugues Hoppe, Low Kok-Lim and Jarek Rossignac for sharing their valuable suggestions and great ideas. At the same time, they have also answered some of the ambiguities that I faced.

I would like to express my greatest gratitude to my family for their unconditional love by supporting and understanding me during my master study.

I would like to send my gratefulness to Leong Chung Ern, Lai Chui Yen, Kenix Chong, Tan Boon Ee, Angeline Ang and Jonas Wong, who have morally supported and encouraged so that I completed my master on time. Lastly, I would like to express my appreciation to those who have assisted me directly or indirectly in making this thesis a success.

ABSTRACT

Drastic growth in computer simulations' complexity and 3D scanning technology has boosted the size of geometry data sets. Before this, conventional (in-core) simplification techniques are sufficient in data reduction to accelerate graphics rendering. However, powerful graphics workstation also unable to load or even generates the smooth rendering of these extremely large data. In this thesis, out-of-core simplification algorithm is introduced to overcome the limitation of conventional technique. Meanwhile, preservation on surface attributes such as normals, colors and textures, which essential to bring out the beauty of 3D object, are also discussed. The first process is to convert the input data into a memory efficient format. Next, datasets are organized in an octree structure and later partitioned meshes are kept in secondary memory (hard disk). Subsequently, submeshes are simplified using a new variation of vertex clustering technique. In order to maintain the surface attributes, a proposed vertex clustering technique that collapses all triangles in every leaf node using the generalized quadric error metrics is introduced. Unlike any other vertex clustering methods, the knowledge of neighbourhood between nodes is unnecessary and the node simplification is performed independently. This simplification is executed recursively until a desired levels of detail is achieved. During run-time, the visible mesh is rendered based on the distance criterion by extracting the required data from the previously generated octree structure. The evaluated experiments show that the simplification is greatly controlled by octree's subdivision level and end node size. The finer the octree, thus the finer mesh will be generated. Overall, the proposed algorithm is capable in simplifying large datasets with pleasant quality and relatively fast. The system is run efficiently on low cost personal computer with small memory footprint.

ABSTRAK

Perkembangan drastik dalam simulasi komputer dan teknologi pengimbasan 3D telah meningkatkan saiz data geometri. Sebelum ini, teknik simplifikasi tradisional (*in-core*) mampu mengurangkan saiz data untuk mempercepatkan visualisasi grafik. Walau bagaimanapun, stesen kerja grafik yang berspesifikasi tinggi juga tidak mampu memuat data yang terlalu besar ini apatah lagi menjana visualisasi yang licin. Dalam thesis ini, algoritma simplifikasi *out-of-core* telah diperkenalkan untuk mengatasi kekurangan teknik tradisional. Sementara ini, pengekalan ciri-ciri permukaaan seperti normal, warna dan tekstur yang menunjukkan kecantikan objek 3D telah dibincangkan. Proses pertama ialah menukarkan data input kepada format yang ramah memori. Kemudian, data disusun dalam struktur octree dan data yang siap dibahagikan disimpan dalam memori sekunder (cakera keras). Selepas ini, permukaan bagi setiap nod diringkaskan dengan teknik pengumpulan verteks. Untuk mengekalkan attribut-attribut permukaan, teknik pengumpulan vertex yang menggantikan segitiga-segitiga dalam setiap nod dengan menggunakan kaedah "generalized quadric error metrics" dicadangkan. Berbeza dengan teknik-teknik lain, pengetahuan antara nod jiran tidak diperlukan dan simplifikasi nod dilakukan secara individu. Proses simplifikasi ini dijalankan secara rekursif sehingga bilangan resolusi yang dikehendaki dicapai. Semasa perlaksaan sistem, permukaan yang boleh dinampak divisualisasikan berdasarkan aspek jarak dengan mengekstrak data berkenaan dari struktur octree yang dihasilkan. Eksperimen yang dianalisa menunjukkan bahawa simplifikasi banyak dikawal oleh aras pembahagian octree dan saiz nod akhir. Semakin banyak octree dibahagikan, semakin tinggi resolusi permukaan yang dihasilkan. Secara keseluruhan, algoritma cadangan adalah berpotensi dalam simplifikasi data besar dengan kualiti yang memuaskan dan agak cepat. Sistem ini telah dilaksanakan secara effektif pada komputer berkos rendah dengan penggunaan memori yang kecil.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
	DECLARATION	ii
	DEDICATION	iii
	ACKNOWLEDGEMENTS	iv
	ABSTRACT	v
	ABSTRAK	vi
	TABLE OF CONTENTS	vii
	LIST OF TABLES	xi
	LIST OF FIGURES	xiii
	LIST OF SYMBOLS	xv
	LIST OF APPENDICES	xvii
1	**INTRODUCTION**	**1**
	1.1 Introduction	1
	1.2 Background Research	2
	1.3 Motivations	5
	1.4 Problem Statement	6
	1.5 Purpose	6
	1.6 Objectives	6
	1.6 Research Scope	7
2	**LITERATURE REVIEW**	**8**
	2.1 Introduction	8
	2.2 Level of Detail Framework	9

	2.2.1	Discrete Level of Detail	9
	2.2.2	Continuous Level of Detail	10
	2.2.3	View-Dependent Level of Detail	11
2.3		Level of Detail Management	11
2.4		Level of Detail Generation	15
	2.4.1	Geometry Simplification	16
	2.4.1.1	Vertex Clustering	16
	2.4.1.2	Face Clustering	17
	2.4.1.3	Vertex Removal	18
	2.4.1.4	Edge Collapse	19
	2.4.1.5	Vertex Pair Contraction	21
	2.4.2	Topology Simplification	21
2.5		Metrics for Simplification and Quality Evaluation	23
	2.5.1	Geometry-Based Metrics	23
	2.5.1.1	Quadric Error Metrics	23
	2.5.1.2	Vertex-Vertex vs Vertex-Plane vs Vertex-Surface vs Surface-Surface Distance	24
	2.5.2	Attribute Error Metrics	25
2.6		External Memory Algorithms	25
	2.6.1	Computational Model	27
	2.6.2	Batched Computations	27
	2.6.2.1	External Merge Sort	27
	2.6.2.2	Out-of-Core Pointer Dereferencing	28
	2.6.2.3	The Meta-Cell Technique	29
	2.6.3	On-Line Computations	30
2.7		Out-of-Core Approaches	31
	2.7.1	Spatial Clustering	32
	2.7.2	Surface Segmentation	34
	2.7.3	Streaming	36
	2.7.4	Comparison	38
2.8		Analysis on Out-of-Core Approach	39

	2.8.1	Advantages and Disadvantages	39
	2.8.2	Comparison: In-core and Out-of-Core Approach	39
	2.8.3	Performance Comparison: Existing Simplification Techniques	41
	2.9	Appearance Attribute Preservation	46
	2.10	Summary	48
3	**METHODOLOGY**		**51**
	3.1	Introduction	51
	3.2	Algorithm Overview	52
	3.3	Data Processing	54
	3.31	Data File Structure	55
	3.4	Proposed Octree Construction	57
	3.5	Proposed Out-of-Core Simplification	59
	3.5.1	New Vertex Clustering	60
	3.5.2	Generalized Quadric Error Metrics	61
	3.5.2	Simplification on Internal Nodes	63
	3.6	Run Time Rendering	64
	3.7	Summary	65
4	**IMPLEMENTATION**		**66**
	4.1	Introduction	66
	4.2	Data Processing	67
	4.2.1	Input Data Reading	67
	4.2.2	Triangle Soup Mesh Generation	68
	4.2.2.1	Calculation of Triangle's Dimension and Sorting Index	69
	4.2.2.2	External Sorting on Sorting Index	71
	4.2.2.3	Dereferencing on Sorting Indices	72
	4.3	Octree Construction	73
	4.4	Out-of-Core Simplification	76

		4.4.1 Simplification on Leaf Nodes	76
		4.4.2 Simplification on Internal Nodes	76
	4.5	Run Time Rendering	77
	4.6	Summary	78
5	**RESULTS AND DISCUSSIONS**		**79**
	5.1	Introduction	79
	5.2	Octree Construction Analysis	80
	5.3	Out-of-Core Simplification Analysis	81
		5.3.1 Proposed Out-of-Core Simplification Analysis	82
		5.3.2 Relationships between Simplification and Octree Construction	85
		5.3.3 Surface-Preserving Simplification Analysis	89
		5.3.4 Comparison on Out-of-Core Simplifications	91
	5.4	Summary	93
6	**CONCLUSIONS**		**94**
	6.1	Summary	94
	6.2	Summary of Contributions	95
	6.3	Future Works	97
REFERENCES			99
Appendices A - B			112

LIST OF TABLES

TABLE NO.	TITLE	PAGE
2.1	Comparison between 3 types of out-of-core simplification	38
2.2	Comparison between in-core approach and out-of-core approach	40
2.3	Simplification results of running QSlim, memoryless simplification (Mless) and OOCS. All results were gathered on a 195 MHz R 10000 SGI Origin with 4GB of RAM and a standard SCSI disk drive. (Lindstrom, 2000)	41
2.4	Run-time performance of OOCS and OOCSx. Blade were computed on Linux PC with 512MB RAM and two 800 MHz Pentium III processor. Statue and fluid are simplified on SGI Onyx2 with forth-eight 250MHz R10000 processor and 15.5GB RAM. (Lindstrom and Silva, 2001)	43
2.5	Simplification on Happy Buddha using four different codes (Cignoni *et al.*, 2003c)	43
2.6	Reduction and performance rates for four standard models using a 1.8GHz Pentium IV PC with 512MB main memory (Borodin *et al.*, 2001)	43
3.1	Space Requirement (Garland and Heckbert, 1998)	63
4.1	Record’s dimension and sorting index for each data type	70

5.1	Octree generation duration	80
5.2	Simplification times and errors of simplified Buddha models	84
5.3	Multiresolution simplification on Buddha in different octree structures	87
5.4	Simplification on Buddha model using OOCS (Lindstrom, 2000b) and the proposed method	92

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE
2.1	Level of detail generation classification	15
2.2	Local simplification operations (Lindstrom, 2003a)	18
2.3	Spatial Clustering process (Lindstrom, 2003a)	33
2.4	Block-based simplification using uniform surface segmentation and edge collapse (Hoppe, 1998b)	35
2.5	Streaming properties (Lindstrom, 2003a)	37
3.1	Framework overview	52
3.2	Main structure of PLY file	56
3.3	PLY file example	57
3.4	New vertex clustering	61
3.5	Orthonormal vector e_1 and e_2 for origin vertex *p* for triangle *T* (Garland and Heckbert, 1998)	62
4.1	Data Structure for different type of vertex element	67
4.2	Finding record’s dimension and sorting index	70
4.3	Dereferencing process	72
4.4	Sub nodes in an octree node	73
4.5	Data member and parameters in *COctree* class	73
5.1	Octree building time versus (a) maximum number of triangles and (b) octree subdivision depth	81
5.2	Out-of-core simplification of Buddha model in different resolutions (T = number of triangles)	83
5.3	Multiresolution of Buddha model in different distances	85

5.4	Sequential simplification on Buddha model, using maximum number of triangles = 1000, Octree's depth = 8	86
5.5	Simplification time versus leaf node's maximum number of triangles	87
5.6	Two different texture preserving simplifications on Buddha model	90
5.7	Simplified Buddha with 90K triangles	91

LIST OF SYMBOLS

max_record	Maximum number of triangle to process at one time
A	Symmetric matrix
b	Vector
B	Size of data item
c	Scalar coefficient
center	Center vertex of node
D^2	Squared distance
e	Orthonormal unit vector
k	Maximum number of disk block that can fit in main memory
K	10^3
L	Current list
LOD	Level of detail
M	Available main memory size
MB	Megabytes
N	Input data size
OOCS	Out-of-core simplification
PLY	PLY data file format
tri_id	Triangle index
T	Number of triangle
T_{in}	Number of input triangle
T_{out}	Number of output triangle
vert_id	Vertex index
V	Vertex
V_T	Optimal vertex for a triangle
V_N	Optimal vertex for a node

width	Maximum width of node
Δ	Processing time

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
A	Results on Dragon model	112
B	Results on Blade model	118

CHAPTER 1

INTRODUCTION

1.1 Introduction

A 3D interactive graphics application is an extremely computational demanding paradigm, requiring the simulation and display of a virtual environment at interactive frame rates. It is significant in real time game environment. Even with the use of powerful graphics workstations, a moderately complex virtual environment can involve a vast amount of computation, inducing a noticeable lag into the system. This lag can detrimentally affect the visual effect and may therefore severely compromise the diffusion of the quality of graphics application.

Therefore, a lot of techniques have been proposed to overcome the delay of the display. It includes motion prediction, fixed update rate, visibility culling, frameless rendering, Galilean antialiasing, level of detail, world subdivision or even employing parallelism. Researches have been done and recovered that the fixed update rates and level of detail technique are the only solutions which enable the application program the balance the load of the system in real-time (Reddy, 1997). Of these solutions, concentration is focused on the notion of level of detail.

Since the mid nineteen-seventies, programmers have used Level of Detail (LOD) management to improve the performance and quality of their graphics systems. The LOD approach involves maintaining a set of representations of each polygonal object, each with varying levels of triangle resolution. During the execution of the animation, object deemed to be less important is displayed with a low-resolution representation. Where as object of higher importance is displayed with higher level of triangle resolution.

The drastic growth in scanning technology and high realism computer simulation complexity has lead to the increase of dataset size. Only super computer or powerful graphics workstation are capable to handle these massive datasets. For this reason, problem of dealing with meshes that are apparently larger than the available main memory exists. Data, which has hundreds, million of polygons are impossible to fit in any available main memory in desktop personal computer. Because of this memory shortage, conventional simplification methods, which typically require reading and storing the entire model in main memory, cannot be used anymore. Hence, out-of-core approaches are gaining its attention widely.

As commonly known, graphics applications always desire high realism scene yet smooth scene rendering. Smooth rendering can be achieved by reducing the number of polygons to a suitable level of detail using the simplification technique. It saves milliseconds of execution time that help to improve performance. However, in order to obtain a nice simplified mesh, surface attributes other than geometry information are essential to be preserved as well. Eye catching surface appearance certainly will increase the beauty of the scene effectively.

1.2 Background Research

Traditionally, polygonal models have been used in computer graphics extensively. Till this moment, large variety of applications is using this fundamental primitive to represent three dimensional objects. Besides, many graphics hardware and software rendering systems support this data structure. In addition, all virtual environment systems employ polygon renderers as their graphics engine.

In reality, many computational demanding systems desire smooth rendering of these polygonal meshes. To optimize the speed and quality of graphics rendering, level of details has been used widely to reduce the complexity of the polygonal mesh using level of detail technique. In short, a process which takes an original polygon description of a three dimensional object and creates another such description, retaining the general shape and appearance of the original model, but containing fewer polygons.

Recent advances in scanning technology, simulation complexity and storage capacity have lead to an explosion in the availability and complexity of polygonal models, which often consist of millions of polygons. Because of the memory shortage in dealing with meshes that are significantly larger than available main memory, conventional methods, which typically require reading and storing the entire model in main memory during simplification process, cannot solve the dilemma anymore. Thus, out-of-core approaches are introduced consequently.

Out-of-core algorithms are also known as external algorithms or secondary memory algorithms. Out-of-core algorithms keep the bulk of the data on disk, and keep in main memory (or so called in-core) only the part of the data that's being processed. Lindstrom (Lindstrom, 2000a) is the pioneer in out-of-core simplification field. He created a simplification method; called OoCS which is independent of input mesh size. However, the output size of the mesh must be smaller than the available main memory. Later on, other researchers carried out similar approaches.

Especially in out-of-core simplification, a large number of research have been done on level of detail's construction and management for use in interactive graphics applications, mostly in medical visualization, flight simulators, terrain visualization systems, computer aided design and computer games. For instance, simplification is used broadly in medical and scientific visualization. It always involves a lot of processing on high resolution three dimensional data sets. The data is mainly produced by those high technology scanners, such as CT or MRI scanners. The simplification process may need to extract volumetric data at different density levels. If the accuracy is not that critical, one may only process on its isosurfaces. Anyhow, these data simplification require a lot of processing time and it is mainly run daily on supercomputers worldwide.

Graphics applications, which demand high accuracy in simplification development is critical. It is essential to maintain the high quality and good frame rates at the same time. For example, medical visualization and terrain visualization is crucial in maintaining a good visual fidelity. Anyway, in many real time systems, the quality of data visualization has to be degraded in order to retain superior rendering time. For instance, an excellent frame rate is vital in game environment without doubt. Thus, the quality of simplified model has to be sacrificed sometimes.

Rendering the large models at interactive frame rates is essential in many areas, includes entertainment, training, simulation and urban planning. Out-of-core techniques are required to display large models at interactive frame rates using low memory machines. Hence, it needs new solution or further improvement such as prefetching, geomorphing, appearance preservation, parallelization, visibility pre-computing, geometry caching, image-based rendering, and etc.

To avoid the last minute data fetching when needed, prefetching, visibility pre-computing and geometry caching are imperative. Although the changes from frame to frame are regularly small, however, they are occasionally large, so, prefetching technique is needed. This technique predicts or speculates which part of the model are likely to become visible in the next few frames then prefetch them from disk ahead of time. Correa *et al.* (2002) showed that prefetching can be based on from-point visibility algorithms. Visibility can be pre-computed using from-region visibility or from-point visibility. Whilst geometry caching exploits the coherence between frames, thus keeping geometry cache in main memory and update the cache as the viewing parameters changes.

Above and beyond, parallelization and image-based rendering enhance the frame rates as well. Parallelization (Correa *et al.*, 2002) uses a few processors to run different tasks at the same time. Or, it can make use of multithreading concept in a single-processor machine too. On the other hand, image-based rendering techniques (Wilson and Monacha, 2003) such as texture-mapped impostors can be used to accelerate the rendering process. These texture-mapped impostors are generated either in a preprocessing step or at runtime (but not every frame). These techniques are suitable for outdoor models.

The geomorphing and surface preserving are potential in pleasant scene rendering. An unfortunate side effect of rendering with dynamic levels of detail is the sudden visual ‘pop’ that occurs when triangles are inserted or removed from the mesh. Geomorphing allows smooth transitions between the approximations (Levenberg, 20002; Erikson, 2000; Hoppe, 1998a). In virtual environments or three dimensional game engines, the surface attributes play an important role to make the object looks attractive. Therefore, these surface attributes like colors and textures should be maintained after simplification process.

1.3 Motivations

Why do we care about visualization of large datasets? Due to the advances in scanning technology and complexity of computer simulation, the size of the datasets grows rapidly these years. The data is vital because it has application in many areas, such as computer design and engineering, visualization of medical data, modeling and simulation of weapons, exploration of oil and gas, virtual training and many more.

These massive data can only be rendered on high end computer system. If it is needed to run on personal computer, it may be an impossible mission, or even it can, the output is jagged or ungraceful. Therefore, to run it on expensive high end graphics machine, it is very cost ineffective and not user friendly. There is a need to display the data in low cost PC with high quality output.

Surface attributes, for example, normal, curvature, color and texture values are important to make the rendered objects looks attractive. It can increase the realism of a virtual environment. It shows the details of an object, such as its illumination, lighting effect and material attributes. Without it, the rendered scene will become dull and bored.

1.4 Problem Statement

The datasets are getting enormous in size. However, even the well implemented in-core methods no more able to simplify these massive datasets. This is mainly because in-core approach loads the whole full resolution mesh into main memory during simplification process. Besides, we cannot keep relying on high end graphics machine as it is expensive and not everyone has the chance to use it. Therefore, when the datasets bigger than the main memory, the datasets cannot be simplified.

Geometry aspects like vertex position always retained after simplification process whether in in-core simplification or out-of-core simplification. However, work in preserving the surface appearance, e.g. surface normal, curvature and color or even texture attributes in the original mesh is not common in out-of-core simplification approach. The lost surface attributes will greatly reduce the realism of virtual environment.

1.5 Purpose

To render the massive datasets in 3D real-time environment and preserve its surface appearance during simplification process using commodity personal computer.

1.6 Objectives

1. To develop an out-of-core simplification technique.
2. To preserve the surface attributes on the out-of-core model based on error metrics.

1.7 Research Scope

a) Only triangular polygonal mesh is considered, other data representation is not investigated here.

b) The simplification is for only static polygonal objects, dynamic object is not covered here.

c) Only vertex positions, normals, colors and texture coordinates are preserved after simplification process.

d) Application is run on commodity personal computer. Commodity in this content means low cost PC with not more than 2GB RAM and not any kind of SGI machine.

e) Graphics card of the PC is assumed capable in handling real time rendering.

f) Only simple level of detail management system is applied by using distance criterion.

g) Secondary memory used here is the hard disk, other secondary memory devices are not investigated its cons and pros.

h) The size of the datasets mustn't larger than the size of secondary memory owned by the machine.

i) No other enhancement techniques such as geomorphing, prefetching, caching, parallelization and no disk usage reduction are investigated.

REFERENCES

Aggarwal, A., and Vitter, J. S. (1988). The Input/Output Complexity of Sorting and Related Problems. *Communications of the ACM*. 31(9):1116–1127.

Alliez, P. and Schmitt, F. (1999). Mesh Approximation using a Volume-Based Metric. *Proceeding of Pacific Graphics ’99 Conference Proceedings*. 1999. 292-301.

Bajaj, C., and Schikore, D. (1996). Error-bounded Reduction of Triangle Meshes with Multivariate Data. *SPIE.* 2656: 34-45.

Bayer, R. and McCreight, E. (1972). Organization of Large Ordered Indexes. *Acta Inform.*, 1:173–189.

Bernardini, F., Rushmeier, H., Martin, IM, Mittleman, J. and Taubin, G. (2002). Building a Digital Model of Michelangelo’s Florentine Pietà. *Computer Graphics and Applications*. 22(1): 59-67.

Borodin, P., Guthe, M. and Klein, R. (2003). Out-of-Core Simplification with Guaranteed Error Tolerance. *Vision, Modeling and Visualization 2003*. 19-21 November 2003. Munich, Germany: 309-316.

Brodsky, D. and Watson, B. (2000). Model Simplification though Refinement. In: Mackenzie, I.S. and Stewart, J. eds. *Graphics Interface 2000*. Montreal, Canada: Morgan Kaufmann Publishers. 211-228.

Brown, R., Pham, B. and Maeder, A. (2003a). Visual Importance-biased Image Synthesis Animation. *Proceedings of the 1st international conference on Computer graphics and interactive techniques in Austalasia and South East Asia*. 2003. New York, USA: ACM Press. 63-70.

Brown, R., Cooper, L. and Pham, B. (2003b). Visual Attention-Based Polygon Level of Detail Management. *Proceedings of the 1st international conference on Computer graphics and interactive techniques in Austalasia and South East Asia*. 2003. New York, USA: ACM Press. 55-62.

Cater, K. (2002). Selective Quality Rendering by Exploiting Inattentional Blindness: Looking but not Seeing. *Proceedings of the ACM symposium on Virtual reality software and technology*. 2002. New York, USA: ACM Press. 17-24.

Cater, K., Chalmers, A. and Dalton, C. (2003). Varying Rendering Fidelity by Exploiting Human Change Blindness. *Proceedings of the 1st international conference on Computer graphics and interactive techniques in Austalasia and South East Asia*. 2003. New York, USA: ACM Press. 39-46.

Certain, A., Popovic, J., DeRose, T., Duchamp, T., Salesin, D. and Stuetzle, W. (1996). Interactive Multiresolution Surface Viewing. *Proceedings of SIGGRAPH 96*. 91-98.

Chhugani, J. and Kumar, S. (2003). Budget Sampling of Parametric Surface Patches. *Proceedings of the 2003 symposium on Interactive 3D graphics*. 2003. New York, USA: ACM Press. 131-138.

Chiang, Y. J., Farias, R., Silva, C. and Wei, B. (2001). A Unified Infrastructure for Parallel Out-Of-Core Isosurface Extraction and Volume Rendering of Unstructured Grids'. *Proc. IEEE Symposium on Parallel and Large-Data Visualization and Graphics*. 59–66.

Chiang, Y. J., Goodrich, M. T., Grove, E. F., Tamassia, R., Vengroff, D. E. and J. S. Vitter. (2000). External-Memory Graph Algorithms. *Proc. ACM-SIAM Symp. on Discrete Algorithms*. 139–149.

Chiang, Y. J. and Silva, C. T. (1997). I/O Optimal Isosurface Extraction. *IEEE Visualization 97.* 293–300.

Chiang, Y. J. and Silva, C. T. (1999). External Memory Techniques for Isosurface Extraction in Scientific Visualization. *External Memory Algorithms and Visualization, DIMACS Series*. 50:247–277.

Chiang, Y. J., Silva, C. T. and Schroeder, W. J. (1998). Interactive Out-Of-Core Isosurface Extraction. *IEEE Visualization 98*. Oct. 1998. 167–174,.

Choudhury, P. and Watson, B. (2002). *Completely Adaptive Simplification of Massive Meshes.* Northwestern University: Tech. Report CS-02-09, 2002.

Chrislip, C. A. and Ehlert Jr., J. F. (1995). *Level of Detail Models for Dismounted Infantry*. Naval Posgraduate School, Monterey, CA.: Master's Thesis.

Cignoni, P., Ganovelli, F., Gobbetti, E., Marton, F., Federico, P., and Scopigno, R. (2003a). BDAM - Batched Dynamic Adaptive Meshes for High Performance Terrain Visualization. *Computer Graphics Forum*. 22(3): 505-514.

Cignoni, P., Ganovelli, F., Gobbetti, E., Marton, F., Federico, P., and Scopigno, R. (2003b). Planet-Sized Batched Dynamic Adaptive Meshes (P-BDAM). *Proceedings IEEE Visualization*. October 2003. Italy: IEEE Computer Society Press, 147-155.

Cignoni, P., Montani, C., Rocchini, C. and Scopigno, R. (2003c). External Memory Management and Simplification of Huge Meshes. *Visualization and Computer Graphics, IEEE Transactions*. 9(4): 525-537.

Cignoni, P., Rocchini, C., Scopigno, R. (1998). Metro: Measuring Error on Simplified Surfaces. *Computer Graphics Forum*. 17(2): 37-54.

Cohen, J., Olano, M. and Manocha, D. (1998). Appearance-Preserving Simplification. *Proceedings of SIGGRAPH 98 in Computer Graphics Annual Conference, ACM SIGGRAPH*. 19-24 July 1998. Orlando, Florida: ACM SIGGRAPH, ?.

Cohen, J., Varshney, A., Manocha, D., Turk, G., Weber, H., Agarwal, P. Brooks, Jr. F. P. and Wright, W. (1996). Simplification envelopes. In: Rishmeier, H. ed. *Proceedings of SIGGRAPH 96*. New Orleans, Louisianna: Addison Wesley. 119-128.

Cohen, J., Luebke, D., Duca, N. and Schubert, B. (2003). *GLOD: Level of Detail for The Masses*. The Johns Hopkins University and the University of Virginia: Technical Report.

Comer, D. (1979). The Ubiquitous B-Tree. *ACM Comput. Surv.* 11:121–137.

Correa, W. T, Klosowski, J.T and Silva, C. T. (2002). *iWalk: Interactive Out-of-Core Rendering of Large Models*. Princeton University, 2002: Technical Report TR-653-02.

Danovaro, E., Floriani, L. D., Lee, M. and Samet, H. (2002). Multiresolution Tetrahedra Meshes: an Analysis and a Comparison. *Shape Modeling International 2002 (SMI'02)*. 17-22 May 2002. Banff, Canada: IEEE, 83.

DeHaemer, J. M. and Zyda, M. J. (1991) Simplification of Objects Rendered by Polygonal Approximations. *Computers & Graphics*. 15(2):175-184.

DeRose, T.D., Lounsbery, M. and Warren, J. (1993). Multiresolution Analysis for Surfaces of Arbitrary Topology Type. Department of Computer Science, University of Washington: Technical Report TR 93-10-05.

Edelsbrunner, H. (1983). A New Approach to Rectangle Intersections, Part I. *Internat. J. Comput. Math.* 13: 209–219.

El-Sana, J. and Chiang, Y. J. (2000). External Memory View-Dependent Simplification. *Computer Graphics Forum*. 19(3): 139–150.

El-Sana, J. and Varshney, A. (1997). Controlled Simplification of Genus for Polygonal Models. In: Yagel, R. and Hagen, H. eds. *IEEE Visualization '97*. Phoenix, Arizona: IEEE. 403-412.

El-Sana, J. and Varshney, A. (1998). Topology Simplification for Polygonal Virtual Environments. *IEEE Transaction on Visualization and Computer Graphics*. 4(2): 133-144.

El-Sana, J. and Varshney, A. (1999). Generalized View-Dependent Simplification. *Computer Graphics Forum*. 18(3): 83–94.

Erikson, C. (1996). *Polygonal Simplification: An Overview*. Department of Computer Science, University of North Carolina, Chapel Hill, NC: UNC Technical Report No. TR96-016.

Erikson, C. (2000). *Hierarchical Levels of Detail to Accelerate the Rendering of Large Static and Dynamic Polygonal Environments*. University of North Carolina at Chapel Hill: Ph.D. Thesis.

Erikson, C. and Manocha, D. (1998). *Simplification Culling of Static and Dynamic Scene Graphs*. Department of Computer Science, University of North Carolina at Chapel Hill: Technical Report TR98-009.

Erikson, C. and Manocha, D. (1999). GAPS: General and Automatic Polygonal Simplification. In: Hodgins, J. and Foley, J. D. eds. *ACM symposium on Interactive 3D Graphics*. Atlanta, Georgia: ACM SIGGRAPH. 79-88.

Erikson, C., Monacha, D. and Baxter, W. V. III. (2001). HLODs for Faster Display of Large Static and Dynamic Environments. *2001 ACM Symposium on Interactive 3D Graphics*. 111-120.

Farias, R. and Silva, C. T. (2001). Out-Of-Core Rendering of Large, Unstructured Grids. *IEEE Computer Graphics & Applications*. 21(4): 42–51.

Fei, G. Z., Guo, B. N., Wu, E. H., Cai, K. Y. (2002). An Adaptive Sampling Scheme for Out-of-Core Simplification. *Computer Graphics Forum*. 21(2): 111-119.

Franc, M. and Skala, V. (2001). Parallel Triangular Mesh Decimation without Sorting. 17th *Spring Conference on Computer Graphics (SCCG '01)*. April 2001. Budmerice,Slovakia: IEEE, 22-29.

Funkhouser, T. A. and Sequin, C. H. (1993). Adaptive Display Algorithm for Interactive Frame Rates During Visualization of Complex Virtual Environment. *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*. 1993. New York, USA: ACM Press. 247-254.

Garland, M. (1999). *Quadric–Based Polygonal Surface Simplification*. Carnegie Mellon University: Ph.D. Thesis.

Garland, M. and Heckbert, P. S. (1997). Surface Simplification Using Quadric Error Metrics. In: Whitted, T. ed. *Proceedings of SIGGRAPH 97*. Los Angeles, California: ACM Press. 209-216.

Garland, M. and Heckbert, P.S. (1998). Simplifying Surface with Color and Texture Using Quadric Error Metrics. *IEEE Visualization ’98 Conference Proceeding*. IEEE, 263-270.

Garland, M. and Shaffer, E. (2002). A Multiphase Approach to Efficient Surface Simplification. *Visualization ’02*. Washington, DC, USA: IEEE Computer Society, 117-124.

Gieng, T. S., Hamann, B. Joy, K. I., Schlussmann, G. L. And Trotts, I. J. (1997). Smooth Hierarchical Surface Traingulations, In: Yagel, R. and Hagen, H. eds. *IEEE Visualization ’97*. Phoenix, Arizona: IEEE. 379-386.

Guthe, M., Borodin, P. and Klein, R. (2003). Efficient View-Dependent Out-of-Core Visualization. *Proceeding of The 4th International Conference on Virtual Reality and Its Application in Industry (VRAI’2003)*. October 2003.

Haibin, W. and Quiqi, R. (2000). Fast Rendering of Complex Virtual Environment. Proceedings of ICSP2000. 1395-1398.

Hamann, B. (1994). A data Reduction Scheme for Triangulated Surfaces. *Computer Aided Geometric Design*. 11(20): 197-214.

Hoppe, H. (1996). Progressive Mesh. In: Rushmeier, H. ed. *Proceeding of SIGGRAPH 96. Computer Graphics Proceedings, Annual Conference Series*. New Orleans, Louisiana: Addison Wesley. 99-108.

Hoppe, H. (1997). View-dependent Refinement of Progressive Meshes. *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques.* August 1997. ACM: 189-198

Hoppe, H. (1998a). Efficient Implementation of Progressive Meshes. *Computer Graphics*. 22(1): 27-36.

Hoppe, H. (1998b). Smooth View-Dependent Level-of-Detail Control and Its Application to Terrain Rendering. *IEEE Visualization ’98.* October 1998. North Carolina: IEEE, 25-42.

Hoppe, H. (1999). New Quadric Metric for Simplifying Meshes with Appearance Attributes. *IEEE Visualization ’99*. October 1999. San Francisco, California: IEEE, 59-66.

Hoppe, H., DeRose, T., Duchamp, T., McDonald, J. and Stuetzle, W. (1993). Mesh Optimization. In: Kajiya, J. T. ed. *Proceeding of SIGGRAPH 93*. Anaheim, California: Addison Wesley. 19-26.

Hughes, M., Lastra, A. A. and Saxe, E. (1996). Simplification of Global-Illumination Meshes. *Proceedings of Eurographics '96, Computer Graphics Forum.* 15(3): 339-345.

Isenburg, M. and Gumhold, S. (2003). Out-of-Core Compression for Gigantic Polygon Meshes. *ACM Transaction on Graphics*. 22(3): 935-942.

Isenburg, M., Gumhold, S. and Snoeyink,J. (2003a). Processing Sequence: A New Paradigm for Out-of-Core Processing on Large Meshes.

Isenburg, M., Lindstrom, P., Gumhold, S. and Snoeyink, J. (2003b). Large Mesh Simplification using Processing Sequences. *Proceedings of Visualization 2003*. 19-24 October 2003. Seattle, Washington: IEEE, 465-472.

Jacob, C. E., Finkelstein, A. and Salesn, D. H. (1995). Fast Multiresolution Image Querying. In: Cook, R. ed. *Proceeding of SIGGRAPH 95, Computer Graphics Proceedings, Annual Conference Series*. Los Angeles, California: Addison Wesley. 277-286.

Kalvin, A. D. and Taylor, R. H. (1994). *Superfaces: Polygonal Mesh Simplification with Bounded Error*. IBM Research Division, T. J. Watson Research CEntre, Yorktown Heights, NY: Technical Report RC 19828 (#87702).

Kalvin, A.D. and Taylor, R. H. (1996). Superfaces: Polygonal Mesh Simplification with Bounded Error. *IEEE Computer Graphics & Applications*. 16(3):64-77.

Klein, R., Liebich, G. ad Straber, W. (1996). Mesh Reduction With Error Control. In: Yagel, R. and Nielson, G. M. eds. *IEEE Visualization ’96*. San Francisco, Carlifornia: IEEE. 311-318.

Levenberg, J. (2002). Fast View-Dependent Level-of-Detail Rendering Using Cached Geometry. *IEEE Visualization 2002*. October 2002. Boston, MA, USA: IEEE. 259-265.

Linderman, J., (2000). *rsort* and fixcut man page, Apr 1996. (revised June 2000).

Lindstrom, P. (2000a). *Model Simplification using Image and Geometry-Based Metrics*. Georgia Institute of Technology: Ph.D Thesis.

Lindstrom, P. (2000b). Out-of-Core Simplification of Large polygonal Models. *ACM SIGGRAPH 2000*. July 2000. 259-262.

Lindstrom, P. (2003a). *Out-of-Core Surface Simplification*. California: University of California, Davis*, lectures on "Multiresolution Methods" February 2003.*

Lindstrom, P. (2003b). Out-of-Core Construction and Visualization of Multiresolution Surfaces. *Symposium on Interactive 3D Graphics.* April 2003. ACM, 93-102, 239.

Lindstrom, P. (2003c). Out-of-core Construction and Visualization of Multiresolution Surfaces. Proceedings of the 2003 Symposium on Interactive 3D Graphics. 28-30 April 2003. Monterey, California: ACM Press,

Lindstrom, P., Koller, D., Ribarsky, W., Hughes, L. F., Faust, N. and Turner, G. (1996). Real-Time, Continuous Level of Detail Rendering of Height Fields. *ACM SIGGRAGH 96*. 109-118.

Lindstrom, P., and Pascucci, V. (2001). Visualization of Large Terrains Made Easy. *IEEE Visualization 2001*. October 2001. San Diego, California: IEEE, 363–370.

Lindstrom, P. and Silva, C. (2001). A Memory Insensitive Technique for Large Model Simplification. *IEEE Visualization 2001*. October 2001. IEEE, 121-126.

Lindstrom, P. and Turk, G. (1998). Fast and Memory Efficient Polygonal Simplification. In: Ebert, D., Hagen, H. and Rushmeier, H. eds. *IEEE Visualization ’98*. Research Triangle Park, North Carolina: IEEE. 279-286.

Lindstrom, P. and Turk, G. (1999). Evaluation of Memoryless Simplification. *IEEE Transactions on Visualization and Computer Graphics*. 5(2): 98-115.

Liu, Y. J., Yuen, M. F. and Tang, K. (2003). Manifold-Guaranteed Out-of-Core Simplification of Large Meshes with Controlled Topological Type. *The Visual Computer (2003)*. 19(7-8): 565-580.

Low, K. L. and Tan T. S. (1997). Model Simplification using vertex-clustering. In: Cohen, M. and Zeltzer, D. eds. *1997 ACM Symposium on Interactive 3D Graphics*. Phode Island: ACM SIGGRAPH. 75-82.

Luebke, D. and Erikson, C. (1997). View-Dependent Simplification of Arbitrary Polygonal Environments. *ACM SIGGRAPH 97*. 199–208.

Nooruddin, F. S. and Turk, G. (2003). Simplification and Repair of Polygonal Models using Volumetric Techniques. *IEEE Transactions on Visualization and Computer Graphics*. 9(2): 191-205.

Ohshima, T., Yamamoto, H. and Tamura, H. (1996). Gaze-Directed Adaptive Rendering for Interacting with Virtual Space. *Proceedings of the IEEE Virtual Reality Annual International Symposium (VRAIS)*. 30 March – 03 April 1996. IEEE, 103-110.

Prince, C. (2000). *Progressive Meshes for Large Models of Arbitrary Topology*. University of Washington: Master’s Thesis.

Reddy, M. (1995). Musings on Volumetric Level of Detail for Virtual Environments. *Virtual Reality: Research, Development and Application*. 1(1): 49-56.

Reddy, M. (1997). *Perceptually Modulated Level of Detail for Virtual Environment.* University of Edinburgh: Ph.D Thesis.

Ronfard, R. and Rossignac, J. (1996). Full-Range Approximations for Triangulated Polyhedra. *Computer Graphics Forum*. 15(3): 67-76;

Rossignac, J. and Borrel, P. (1992). *Multi-resolution 3D Approximations for Rendering Complex Scenes*. NY: Technical Report RC 17697.

Rossignac, J. and Borrel, P. (1993). Multi-resolution 3d Approximations for Rendering Complex Scenes. In: Falciendo, B. and Kunii, T. L. eds. *Modeling in Computer Graphics*. Springer-Verlag. 455-465.

Rushmeier, H., Larson, G. W., Piatko, C., Sanders, P. and Rust, B. (1995). Comparing Real and synthetic Images: Some Ides about Metrics. In: Hanrahan, P. and Purgathofer, W. eds. *Eurographics Rendering Workshop 1995*. Dublin, Ireland: Springer-Verleg. 82-91.

Schmitt, F. J. M., Barsky, B. A. And Du, W. H. (1986). An Adaptive Subdivision Method for Surface-Fitting From Sample Data. *Computer Graphics*. 20(4): 179-188.

Schroeder, W. J. (1997). A Topology Modifying Progressive Decimation Algorithm. In: Yagel, R. and Hagen, H. eds. *IEEE Visualization '97*. Phoenix, Arizona: IEEE. 205-212.

Schroeder, W. J., Zarge, J. A. And Lorensen W. E. (1992). Decimation of Triangle Meshes. In: Catmull, E. E. ed. *Computer Graphics (Proceeding of SIGGRAPH 92)*. Chicago: Illinois. 65-70.

Shaffer, E. and Garland M. (2001). Efficient Simplification of Massive Meshes. *12th IEEE Visualization 2001 Conference (VIS 2001)*. 24-26 October 2001. San Diego, CA: IEEE.

Silva, C.T., Chiang, Y. J., El-Sana, J. and Lindstrom, P. (2002). Out-Of-Core Algorithms for Scientific Visualization and Computer Graphics. *IEEE Visualization Conference 2002.* October 2002. Boston, MA: IEEE, 217-224.

Southern, R., Marais, P. and Blake, E. (2001). Generic Memoryless Polygonal simplification. *ACM Afrigraph.* ACM, 7-15.

Trotts, I. J., Hamann, B., Joy, K. I. (1999). Simplification of Tetrahedra Meshes with Error Bounds. *IEEE Transaction on Visualization and Computer Graphics*. 5(3): 224-237.

Varadhan, G. and Manocha, D. (2002). Out-of-Core Rendering of Massive Geometric Environments. *IEEE Visualization 2002*. Oct 27 – Nov 1. Boston: IEEE, 69-76.

Vince, J. (1993). Virtual Reality Techniques in Flight Simulation. In A.A. Earnshaw, M.A. Gigante and H.Jones eds. *Virtual Reality Systems*. Academic Press Ltd.

Watson, B., Friedman, A. and McGaffey, A. (2000). Using Naming Time to Evaluate Quality Predictors for Model Simplification. *Proceeding of the CHI 2000 Conference on Human Factors in Computing Systems*. April 2000. The Hague, The Netherlands: Addison Wesley. 113-120.

Watson, B., Walker, N. and Hodges, L.F. (1995). A User Study Evaluating Level of Detail Degradation in the Periphery of Head-Mounted Displays. *Proceedings of the FIVE '95 Conference*. UK: University of London, 203-212.

Wernecke, J. (1993). *The Inventor Mentor: Programming Object-Oriented 3D Graphics with Open Inventor (TM)*. Release 2, 1st Edison. Boston, MA, USA: Addison-Wesley.

Wilson, A. and Manocha, D. (2003). Simplifying Complex Environments Using Incremental Textured Depth Meshes. *ACM Transactions on Graphics (TOG)*. July 2003. New York, USA: ACM Press. 678-688.

Wu, J. and Kobbelt, L. (2003). A Stream Algorithm for the Decimation of Massive Meshes. *Graphics Interface '03*. 185-192.

Xia, J. C., El-Sana, J. and Varshney, A. (1997). Adaptive Real-Time Level-of-Detail-Based Rendering for Polygonal Models. *IEEE Transactions on Visualization and Computer Graphics.* 3(2): 171-183.

Zach, C, Mantler, S. and Karner, K. (2002). Time-critical Rendering of Discrete and Continuous Levels of Detail**.** Proceedings of the ACM symposium on Virtual reality software and technology. 2002. New York, USA: ACM Press.