VR-GIS: An Integrated Platform of VR Navigation and GIS Analysis for City/Region Simulation

Chunyong Ma[¤] Yongyang Qi[+] Yong Chen[§] Yong Han[*] Ge Chen[#]
Ocean University of China

Abstract

This paper introduces a virtual city oriented VR-GIS platform which synthesizes several latest information technologies including virtual reality, 3D geographical information system, remote sensing and multi-dimensional visualization. The platform is a seamless integration of VR functions and GIS analysis methods, which can be used to organize and present massive spatial data. It also supplies 3D spatial analysis functions, 3D visualization for spatial process and natural simulation, and serves as an engine platform for digital city.

Keywords: VR-GIS, urban simulation, massive data, natural simulation

1 Introduction

With the development of VR (Virtual Reality) technology and widely applications in various areas, the requirements to VR are also increasing rapidly. Users not only need to obtain the landscape geospatial data dynamically but also need to perform some analyses, calculations, managements and transfers based on data. As a result, a combination of geographic information system and virtual reality technology [Huang et al. 2001] which is named VRGIS (Virtual Reality Geographical Information System) has become a hot topic.

Urban Simulation is the application of VRGIS in urban planning, management, architecture, etc. Using advanced computer technology to construct digital city has attracted the attentions from many walks of life. By integrating the friendly interactive interface of Virtual Reality System and spatial analysis specialty of Geographical Information System, VRGIS is preferred in practical application, especially by the geography and urban planning workers.

Urban simulation is becoming widely noticed nowadays, and some simulation systems have been developed in this area. For example: ArcView3D Analyst, Imagine Virtual GIS, GeoMedia, etc. In fact, these systems do not focus on spatial analysis but revealing the geospatial data. Vega developed by Multigen-Paradigm has some advantages in VR simulation, but it based on single model date source, which exported from Creator developed by the same company. The VR-GIS platform supports steadily real-time navigation in virtual scenes which are constructed with massive, multi-resolution data from various sources. Especially, models exported from 3DS MAX could be used, so the software accepts more popular data sources than Vega. VR and GIS modules are integrated seamlessly in the platform. And high quality natural phenomenon simulation is supported. In a word, kinds of requirements for large-scale landscape simulation and city/region data management can be satisfied.

Using the platform, 3D urban landscape database with various data sources can be produced to implement spatial analysis and 3D visualization. It can respond timely to various external information and environments by making use of the latest decision-making assistant technology, and a better basis of scientific planning can be supplied[Ishida 2002].

[¤]e-mail: chunyongma@126.com
[+]e-mail: qiyongyang123@tom.com
[§]e-mail: chenyong@ouc.edu.cn
[*]e-mail: chinahanyong@126.com
[#]e-mail: gechen@public.qd.sd.cn

VRCAI 2008, Singapore, December 8 –9, 2008.
© 2008 ACM 978-1-60558-335-8/08/0012 $5.00

2 Related Work

GIS technology has brought great convenience in urban management. However, it is still a two-dimensional technology which is essentially based on the abstract symbols. With the development of VR, it appears an inevitable trend that Digital City, VR and GIS will be integrated. Nowadays, there are various VR simulation platforms which own different characteristics. Multigen-Paradigm Company in the U.S. has developed a series of Multigen Creator software which is advanced in polygon modeling, vector modeling, and real-time optimization for OpenFlight data formats. It has close connection with the subsequent real-time simulation software Vega, and has a leading position in the world about visual navigation, simulation training, urban navigation, interactive games, scientific visualization and other real-time simulation fields. However, there are still some imperfections about visual effects, massive data management, high-speed rendering of massive three-dimensional data, spatial analysis. Tree C Company in Netherlands has developed a soft VR4MAX which achieves data support for mainstream modeling software 3DS MAX. It has excellent visual effects, and supports the 3D glasses, head tracker and other hardware equipment. But its weak point is missing the combination with GIS technology, especially with applications of Digital City. There is a large gap to achieve the design and management of urban simulation.

In the early 1990s, Koller and others had an integration research about VR and GIS, and put forward the concept of

VRGIS [Koller et al. 1995]. VRGIS is based on VR technology as a front-end interaction with users and supports GIS spatial data storage, processing, query and analysis functions of the system from bottom. It has several characteristics as following: ①The true performance of the spatial data, such as terrain and environment, can be represented precisely. And it is also applicable to the past and the future things which may not exist. ②Users can observe, immerse and have real-time interaction from any point of view. They can move inside and outside freely in the chosen geographical zone (the geographical scope). ③ Based on three-dimensional space which has a database of basic GIS functions (such as query, spatial analysis, etc.)

Three-dimensional digital city usually has massive data and diversiform formats, including 3D models, grid data, vector data and property data, etc. Taking the project of "Digital Qingdao" for example, the amount of model data is nearly 20G, the amount of DOM data is nearly 150G, and the amount of DEM data is nearly 2G. Thus the unity of massive data management and efficient three-dimensional rendering is the difficulty that VR-GIS Platform needs to overcome.

With the development of VR technology and computer hardware technology, massive data management and three-dimensional visualization technologies have been greatly improved. In VR field, massive data management and accelerated rendering technology include the following areas of study: architecture design based on out-of-core, optimization to accelerate the three-dimensional rendering, I/O completion ports with multi-thread technology, large and high-speed terrain rendering technology. However, most of current commercial platforms have no timely application of these research results. That makes the combination of VR and GIS not in place and reduces the speed of city digitization. The advantages of VR-GIS Platform are the integration of various up-to-date VR technologies, which can solve the problems in three-dimension visualization of massive data, realize the seamless integration of VR and GIS technologies and provide a strong technical support for the city digitization.

3 Platform Overview

VR-GIS platform designation uses object-oriented project database, whose data storage and management are based on object-oriented nodes. The system structure of VR-GIS is laid out in figure 1.

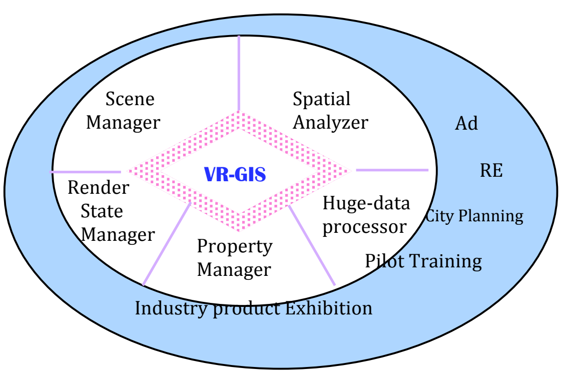

Figure 1 System Configuration and Functions

3.1 Software Architecture

The VR-GIS platform regards everything in scene as objects and organizes all kinds of objects with the hierarchy architecture, which is divided as core level, middle level and extended level. The core level is responsible for the organization and management of data and also integrating two render modes: in memory or out-of-core. The middle level supports the interface by encapsulating the core level and exporting the accessing interface for the extended level. The extended level implements extended function to fulfill the demand on application layer. With design patterns such as factory, visitor and singleton, the platform has good flexibility and extensibility. The hierarchy model is laid out in figure 2.

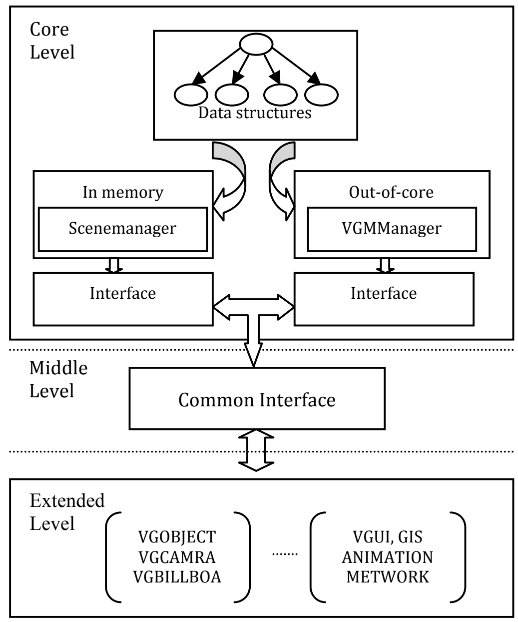

Figure 2 Software Architecture

4 Key Technology

Both VR and GIS systems need massive data, which can be provided from various sources. In the platform, DEM and DOM are used to construct the virtual terrain, and a 3DS MAX plug-in is developed to implement the combination of VR-GIS and 3DS MAX. Because the virtual city needs massive data and computes too much, VR-GIS platform introduces several optimization and acceleration technologies. To build the virtual scene, natural environment is simulated based on GPU. Lastly, VR and GIS modules are integrated seamlessly in the platform.

4.1 Support of Massive Geospatial Data

Virtual city simulation encounters the problem about massive data. When we say massive data, it means that the data is dozens of GB or even more, and too large to load in the memory at one time, it always include various data sources and millions of triangle faces in models of a city or region. The VR-GIS platform is designed to support massive data. Data transfer methods used in VR-GIS platform make the ability of present large-scale scenes perform well.

4.1.1 The Real-time Render of Massive 3D Model

Since the models in virtual city always include millions of triangle polygons, which make the models become much finer and complicated. Higher demands in memory capacity, processing speed, drawing speed and transfer efficiency appear. In order to perform real-time rendering of massive data and considering current PC configuration, the following problems should be solved.

First, the data is too massive to be loaded in the memory completely. Second, the overfull computing workload of CPU and GPU cause slack rendering.

To solving the first problem, an interactive rendering system is proposed for Out-of-Core dataset [Ahn et al. 2006]. The internal memory only includes the data which need to be rendered. With the efficient algorithm of data schedule, these data can be updated real time. Many technologies are used to increase the speed of I/O, such as the file system driver development and the use of Hilbert space-filling curve [Moon et al. 2001]. The former enhances the control of file access and the later guarantees the consistent of the data access and file arrangement. Thus, the continuity of the data access can be guaranteed to give an excellent presentation depending on the viewer's perspective.

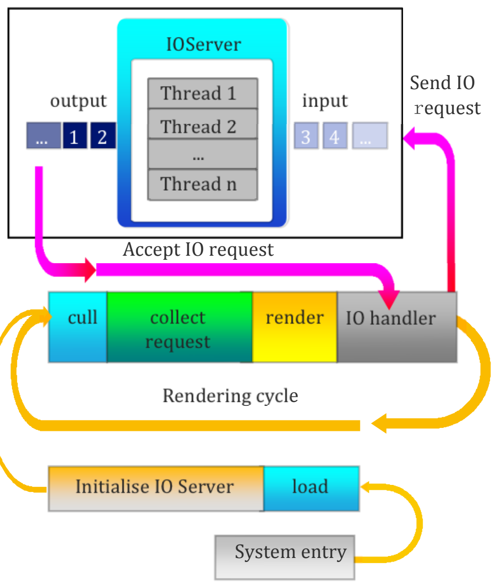

Figure 3 Asynchronous I/O Scheduling Algorithm

VR-GIS platform uses I/O completion ports, which provide an efficient threading model for processing multiple asynchronous I/O requests on a multiprocessor system. When a process creates an I/O completion port, the system creates an associated queue object for the requests whose sole purpose is to service these requests. Processes that handle many concurrent asynchronous I/O requests can do this more quickly and efficiently by using I/O completion ports in conjunction with a pre-allocated thread pool than by creating threads at the time they receive an I/O request.

The scheduling algorithm is shown in figure3. When we enter VR-GIS platform, IOServer shown in the figure3 is initialized. At the time, IOServer creates several I/O threads and an I/O completion port named "CompletionPort" with the CreateIOCompletionPort function. When 3D files are imported, their handles would be associated with the port "CompletionPort", and the memory-resident information including 3D models' boundary box, file properties are fetched.

So how does the IOServer work? When a file handle is associated with the completion port "CompletionPort", the status block which is passed in will not be updated until the packet is removed from the completion port. The I/O threads create when IOServer initializes using the GetQueuedCompletionStatus function to wait for a completion packet to be queued to the I/O completion port. Threads that block their execution on an I/O completion port are released in last-in-first-out (LIFO) order, and the next completion packet is pulled from the I/O completion port's FIFO queue for that thread.

The struct "request" inherited from the struct "OVERLAPPED" packs the I/O data, including file brief information, data size and buffer. The main thread takes charge of the system rendering cycle. Following the viewpoint moving, the frustum will be updated. Some models which haven't stay in the memory but should be rendered in current frame would be packed as the I/O requests. And the models which should be deleted from memory are culled in this frame. At the end of one frame main thread will send the requests to the IOServer, and receive some requests dispatched by the IOServer which contain the needed data for next frame.

To solving the second problem, we introduce several optimization technologies of visualization including texture mapping, automatic level-of-detail [Peng and Kuo 2005], occlusion culling and frustum culling with Quad-tree as shown in figure 4.

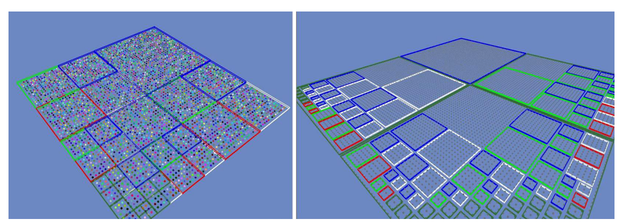

Figure 4 Quad-tree Scene Graph

In VR-GIS platform, all types of entities including models, billboard trees, fountains and cameras are all inherited from the base class "Node". They are organized and managed by Quad-tree combining with AABB (Axis-Aligned Bounding Boxes).

Before constructing the scene tree, we calculate the scene's boundary box. Firstly, split the father node with symmetric mode. Then, dispatch all "Node" included in the father node to his children based on their AABB. If one "Node" is included by two or more children, it will be dispatched to the child who includes its center point, and the boundary of this child would adjust to its AABB. And the tree node would stop splitting if its triangles number less than a threshold. So the tree will be build to organize all "Node" through the recursive splitting.

Before rendering scene, VR-GIS platform process frustum culling with Quad-tree. If a tree node's boundary box doesn't intersect with frustum completely, the traverse will be stopped. So the culling is more effective than linear traverse mode, especially when the amount of "Node" is very large.

4.1.2 GPGPU Based Large Scale Terrain Rendering

General terrain LOD algorithms use a hierarchy of mesh refinement operations to adapt the surface tessellation. But VR-GIS platform adopts the main idea of the GPU-Based Geometrical Clipmap [Losasso and Hoppe 2004], which caches the terrain in a set of nested regular grids centered about the viewer as shown in figure 5.

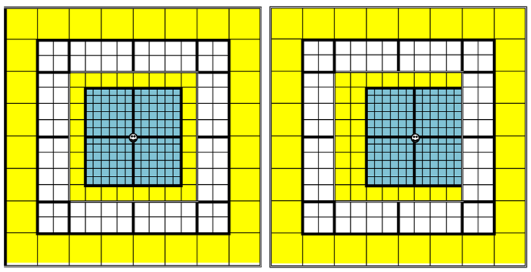

Figure 5 Nested Grid Structure

The emphasis focused on simplification of the detail, more effective data structures and elevation data are updated based on GPGPU technology. To resolve the problem of different step size between two successive clipmap rings, the ring strip lies at the boundary of coarser level. When viewpoint updates, the coarser level firstly adjusts itself inside as shown in figure5.

After preprocessing the elevation data of each level, we can compress the elevation and residual data as a set of images. And every layer can be divided into 4 or 12 blocks and a strip. Because the blocks are invariable and regular, they are stored in graphics memory. When we render the nested grids in VBO technology, their corresponding elevation will be fetched real-timely in vertex Shader, with the technology of VTF (Vertex Texture Fetch) for NVIDIA or R2VB (Render to Vertex Buffer) for ATI graphics card.

To avoid cracks, the platform adjusts the elevation data with the residual data to assure that vertices of the coarser level are coincident with those of the finer level at the interlevel boundary. High resolution terrain image is used and high quality rendering of large scale terrain performs well.

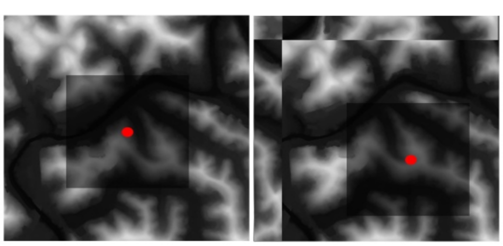

Figure 6 Elevation Image Update

We preprocess terrain elevation image data into a pyramid of multilevel, and divide each level into many blocks then code them. As the viewpoint moving, elevation image will process a toroidal update, which takes chunk as unit, as shown in figure 6.

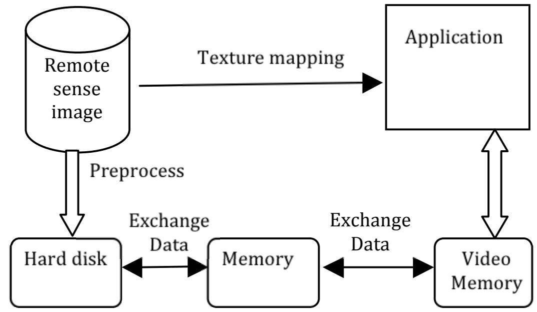

Figure 7 Remote Sense Image Process

The process of remote sense image is illustrated in figure 7. Firstly, it will be preprocessed into a pyramid of multilevel corresponding to the elevation pyramid mode and be saved on hard disk. In the process of terrain rendering, terrain images are loaded to memory to generate textures based on the viewpoint and the pyramid mode. High resolution images are chose near the viewpoint, and low resolution images are loaded in the far region. At the same time, some textures in memory or display memory, which don't need to be rendered, will be deleted. Also in this process, the IOCP technology is used to improve the efficiency of texture I/O.

Figure 8 shows the terrains generated with 16384× 16384 height map of Alaska, United Status.

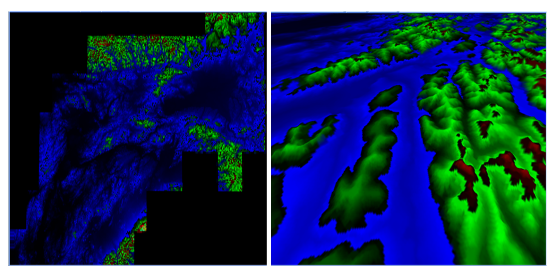

Figure 8 Terrains in VR-GIS

4.2 GPU-based Hardware Acceleration and Natural Simulation

With the powerful vector operation ability, advanced hardware architecture and high speed I/O system (high video memory bandwidth), GPU lightens the burden of CPU. Characterized with parallel processing and dense computing capability, GPU is always considered as a SIMD parallel machine or a flow processor, which can be used in massive data processing. On the basis of GPU occlusion query [Aila and Miettinen 2004], the system puts forward an optimal hierarchy occlusion culling algorithm for complex 3D scenes. The occlusion query and rendering visible nodes are executed alternately, so the waiting time caused by the delay of the occlusion query is reduced effectively. Using advanced VBO (Vertex Buffer Object) technology, the high frequent accessed vertices data are stored in video memory, so the transmission of the data between internal memory and video memory is reduced and the performance of rendering pipeline is improved.

VR-GIS platform can simulate kinds of natural phenomena, including rain, snow, fog, flowing water, reflection of glass and metal, etc. All the simulations are based on GPU shaders programming which have the flexibility to perform amazing vertex-by-vertex and pixel-by-pixel effects, and combined with the parallel-processor performance to use shaders in interactive graphics. Take water simulation as an example, the surface distortion is controlled by normal map Shader and the reflection can be implemented with cube mapping in the pixel Shader.

4.3 Seamless Integration of VR and GIS

Data sharing and function unification are the main goals of the integration of VR and GIS. Data of VR and GIS are uniformly abstracted as node objects and organized by node objects in the platform. Each node stores the geospatial information and the index of its property information. By abstracting VR data and GIS data uniformly and inheriting their traits, the platform provides uniform external access interface and makes nodes have their own render and access mode. In this way, unification and sharing of VR and GIS data are achieved.

As to the function unification, 3DGIS analysis functions are developed based on the uniform geospatial data. Common interface provided by middle level is used to access the geospatial data (points, lines, faces, vectors, etc) as an object. Combining the data's 3D traits, GIS analysis algorithm is applied in the 3D virtual scene.

The GIS analysis functions in the platform includes: measurement of the point, lines, faces and volumes, flooding analysis, earthwork measurement, muter-invisibility analysis, contour analysis, shadow analysis, path analysis, etc.

5 Implementation and Application

The oriented city/region simulation VR-GIS platform is developed based on OpenGL and C++, which integrates VR and GIS seamlessly and supports massive data. Based on VR-GIS platform, we have developed some applications, mainly including: "A Virtual Interactive Planning and Management System for the National Park of Leshan Giant Buddha", "A 3D Virtual Interactive Planning and Management System for the 2008 Olympic Regatta in Qingdao", "A Multi-dimensional Interactive Simulation System for the Planning and Management of the International Cruise Terminal of Shanghai Port", "A 3D Virtual Interactive Urban Planning and Management System for the Huangshan Road Area of Hefei City", etc.

We have tested the platform on a PC with a 3.0 GHz Pentium4 CPU, 1 GB system memory, and an GeForce 8600 GT GPU with 512MB video memory by using data of Leshan city which about 5G including 48sq.km. DEM, DOM and model files exported from 3DS Max. The experiment can run 25fps averagely. Figure 9 proves the platform and our experiment, and figure 10 shows the result of measure a door distance. In this experiment we can move our viewpoint freely through controlling keyboard and mouse, and easily get the distance by click two points on the render region after choose the function button of "Distance Measure". The message box shows the door's height is 2.95m.

The virtual scene of 2008 Olympic regatta in Qingdao rendered in VR-GIS is shown in figure 11, and the virtual Hefei city is shown in figure 12.

Figure 9 The Virtual Leshan Buddha Spectacle in VR-GIS platform

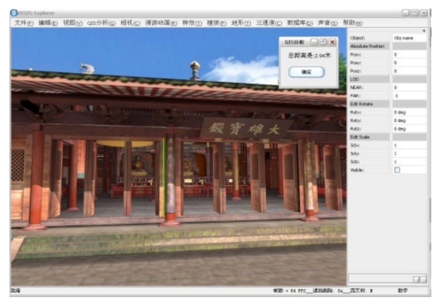

Figure 10 Measurement of Distance in VR-GIS platform

Figure 11 the Virtual Scene of 2008 Olympic Regatta in Qingdao Rendered in VR-GIS Platform

Figure12 the Virtual Scene of Hefei City in VR-GIS Platform

6 Conclusion and Perspectives

A type of software named VR-GIS is developed, in which the integration of VR and GIS is researched. Massive data shared by the two systems (VR and GIS) is supported in the platform. It can meet the requirements for virtual city construction, GIS analysis in virtual environment, and natural simulation based on GPU.

The future work will focus on the use of agent in VR-GIS platform, the way to build virtual city automatically, and making the process more intelligently.

7 References

Ahn, M., Guskov, P., and Lee, P. 2006. Out-of-core remeshing of large polygonal meshes, *IEEE Transactionson Visualization and Computer Graphics*, 12, 5, 1221-1228.

Aila T. and Miettinen V. 2004. Dpvs: An occlusion culling system for massive dynamic environments. *IEEE Computer graphics and Applications* 24, 2, 86–97.

Asirvatham, A., and Hoppe, H. 2005. Terrain Rendering Using GPU-Based Geometry Clipmaps. *GPU Gems 2*, 27-45.

Cai, K., Liu, P., Wang, W., Sun, W, and Wu, P. 2006. Progressive out-of-core compression based on multi-level adaptive octree . In *Proceedings of the 2006 ACM international conference on Virtual reality continuum and its applications*. 83-89.

Faust, N., Ribarsky, W., Jiang, T. Y, and Wasilewski, T. 2000. Real-Time global data model for the digital Earth, In *Proceedings of the International Conference on Discrete Global Grids.*

Huang, B., Jiang, B., and Li, H. 2001. An integration of GIS, virtual reality and the Internet for visualization, analysis and exploration of spatial data. *International Journal of Geographical Information Science*, 15, 5, 439-456.

Ishida, T. 2002. Digital City Kyoto: Social Information Infrastructure for Everyday Life. *Communications of the ACM*, 45, 7, 76-81.

Koller, D., Lindstrom, P., Ribarsky, W. 1995. Virtual GIS: a Real-time 3D Geographic Information System Visualization. *Visualization 1995, Proceedings, IEEE Conference*, 94-100, 443

Losasso, F., and Hoppe, H. 2004. Geometry clipmaps: Terrain rendering using nested regular grids. *ACM Transactions on Graphics*, 23, 3, 769-776.

Moon, B., Jagadish, H. v., Faloutsos, C., and Saltz. J. H. 2001. Analysis of the clustering properties of the hilbert space-filling curve. *IEEE TKDE*, 13, 1, 124-141.

Peng, J., and Kuo, C. J. 2005. Geometry-guided progressive lossless 3D mesh coding with octree (OT) decomposition. *ACM Transactions on Graphics*, 24, 3, 609-616.

Roth, M., Voss, G., and Reiners, D. 2004. Multi-threading and Clustering for Scene Graph Systems. *Computer and Graphics*, 28, 1, 63-66.

Vitter, J. S. 2001. External Memory Algorithms and Data Structures: Dealing with massive data. *ACM Computing Surveys*, 33, 2, 209-27