LARGE-SCALE VECTOR DATA DISPLAYING FOR INTERACTIVE MANIPULATION IN 3D LANDSCAPE MAP

M. Sun[a,*], G.L. Lv[a,b], C. Lei[c]

[a]Institute of RS&GIS, Peking University, Yiheyuan Road, Beijing, China - sunmin@pku.edu.cn
[b]Institute of Survey and Mapping, Central South University, Changsha, China
[c]Shenzhen Geotechnical Investigation & Surveying Institute Co., Ltd, Shenzhen, China

Commission IV, WG IV/4

KEY WORDS: 3D Landscape Map, Large-scale Vector Data, Manipulation, Displaying

ABSTRACT:

Displaying large-scale 3D vector data in landscape map is very important, as 3D vector data can provide many important information, such as: precise geographic boundaries, areas, 3D text marks, different attributes identity, precise path and many important invisible information in real world (e.g.: underground things) and etc. In this paper, we present a technique that is capable of efficiently and precisely displaying 2D vector data in 3D landscape map. In order to release CPU from heavy computation, texture-based approach is selected to render 2D vector data on 3D terrain blocks in browsing mode. Geometry-based approach is selected in interactive manipulation mode or precise expression of vector data is required. We use static terrain mesh (STM) instead of dynamic terrain mesh on the base of SOAR algorithm, to improve efficiency of 3D geometry primitives creation from 2D vector data in real-time, and to support interactive manipulation of 3D vector object. In addition, several scales of 2D vector data map are used to build a pyramid of vector data. These strategies made large-scale vector data displaying easy in 3D landscape map.

1. INTORDUCTION

The majority of the research works on 3D landscape map emphasize the reconstruction of the real view, which use luxuriant models to express buildings and vegetation in one virtual scene, however, geospatial data (include vector data and its attribute data) expression is generally ignored (Stephen M. 2004). The importance of expression of geospatial data is that invisible data in real view can be browsed, manipulated and analyzed in digital virtual scene. This is also the main purpose of utilization of geographic information system (GIS). In other words, the application of GIS relies on geospatial data. Therefore, if the expression and visualization of geospatial data in 3D landscape map is ignored, then the result utilization would only limit to landscape browsing.

Some research works have been done on how to visualize vector data on digital terrain model or in 3D landscape map. They generally can be divided into two classes of method: class of raster method, which project vector data onto one image that is used as a texture map on DEM, and class of vector method which convert 2D geometry primitives into 3D geometry primitives, two classes have different drawback. Generally, raster method can't avoid aliasing and memory cost problem, e.g. when the terrain mesh close enough to the viewpoint, texture has to be zoomed in, even 1 pixel precise can't satisfy the expression of vector data. Vector method generally can't avoid z buffer artifacts problem, as vector data in GIS are mainly expressed on 2D plane, when they were converted into 3D space, line segment and polygon plane cannot match 3D terrain mesh surface in details.

In our approach, we use two classes of method respectively in different levels of detail. For the fast visualization of large terrain scene, a pyramid of 3D terrain model is built with level-of-detail (LOD) and indexed with quadtree, blocks of DEM are expressed with static terrain mesh. While one corresponding pyramid of large-scale vector data is built with several scales of map data, vector data blocks in the pyramid are converted into textures in pre-processing or rendered with P-buffer technique in real-time if hardware capability is permitted.

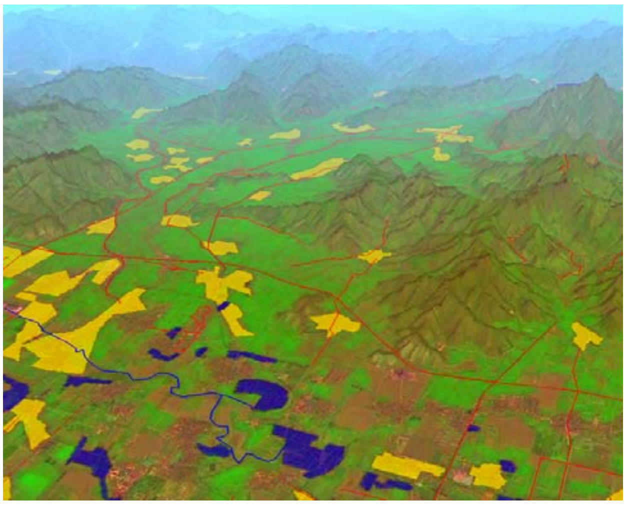


Figure 1. Visualization of vector data in Pinggu area (Beijing)

When textures of vector data have not stretched large than their original size in rendering process, vector data can be expressed precisely. In our approach, we switch to geometry-based method if a block of DEM move close to the viewpoint or one vector object needs interactive manipulation. 3D geometry primitives are created from static terrain mesh intersects with vertical planes that pass line segments of vector objects.

The advantage of our approach including satisfy large-scale vector data rendering in real time, support interactive manipulation and edit to vector objects on the 3D terrain surface.

The paper is structured as follows: section 2 discusses related work. Section 3 explains DEM and vector data management. Section 4 explains our approach of large-scale vector data rendering. Section 5 explains techniques for interactive manipulation and vector object edit. Section 6 gives results and discussion.

2. RELATED WORK

For large-scale 3D terrain model display, some technologies are come into mature, typical works have been done by Mark (1997), Lindstrom (2001), and Ulrich (2002). Lindstrom presented SOAR algorithm and Mark presented ROAM algorithm, both become famous algorithm for terrain mesh simplification, Ulrich presented Chunked LOD method made large-scale terrain visualization more easy. The idea of these methods is to realize real-time LOD producing, i.e. the 3D terrain mesh dynamically refreshes itself globally or locally by different precision triangle set.

In order to resolve 3D vector display on 3D terrain mesh that produced by these simplify methods, Wartell et al. (2003) presented an algorithm to render polylines on terrain mesh by adapting polylines to the current state of the multiresolution terrain mesh in real time, which resulting in expensive computational costs when massive vector data need to be rendered. Schneider et al. (2005) presented a hybrid technique which composes of two parts, the first part is a texture-based approach and the second part is geometry-based approach. In their work, multiresolution 3D geometry is created by preprocessing step, and associated with quadtree nodes for load and display. The main drawback of this method is that interactive manipulation to the vector data is not supported, and not suited for large-scale vector data set.

In addition, Schneider and Klein (2007) present a method that is based on the stencil shadow volume algorithm and allows high-quality real-time overlay of vector data on terrain. The method render vector data using a screen-space algorithm, so can avoid aliasing artifacts and its performance is almost independent of terrain geometry. Although interactive editing and manipulation of the vector data is possible with this method, independence of vector data rendering from terrain geometry would introduce difficulty for the actual performance.

Besides these geometry-based approaches, some texture-based approaches were presented. One typical work presented by Kersting et al. (2002), they proposed to render vector data using P-buffer provide by OpenGL, one on-demaind texture pyramid is produced in real time by project vector data onto a certain plane. Although P-buffer technology can be performed very fast, but its size is limited and vector data traverse in each frame need expensive computation.

In addition, some works also present different methods towards adding GIS function in 3D landscape map (Dollner 2000), or add vector performance into 3DGIS (Stephen et.al, 2008), but much works are remained to fulfill.

3. DEM AND VECTOR DATA MANAGEMENT

Data management is important for display geospatial data efficiently. Large-scale terrain displaying method generally adapt pyramid structure to organize DEM and matched images (figure 2), as DEM and images data are raster format, it's easy to create pyramid structure. However, vector data are generally captured and processed with certain scales (e.g: 1: 500, 1: 1000, 1:2000, 1:5000, etc.). A set of vector data with random scale need generation from neighbour scale, but generation algorithm itself still keeps as one big problem if high quality of the random scales vector data is required.

In order to match the DEM data levels from the pyramid structure, corresponding scales of vector data are required. In our approach, we use dynamic generation method to satisfy this requirement. One simple method is to make dense points of vector objects to be sparse depend its distance from the viewpoint. In our approach, we adopt method presented by Sun M. et al. 2007.

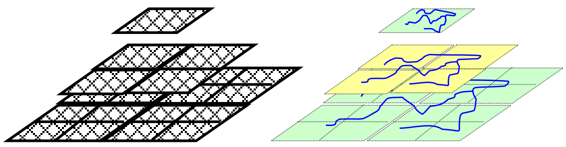


Figure 2. DEM (left) and vector data (right) management with pyramid structure (dash line means vector data is just indexed but not divided into quadtree blocks physically).

4. RENDER LARGE-SCALE VECTOR DATA

4.1 Texture-based Approach

In our approach, static texture pyramid is adapted, which is created by project vector data onto one plane. The advantages of static texture pyramid is that just need one time data pre-processing, and easy to be down scaled at various resolutions to build one corresponding pyramid of DEM. The main drawback of this kind of approaches is that the accuracy of vector data displaying is limited by texture size, or need huge memory cost to improve real-time rendering of large-scale vector data in high accuracy (Kersting 2002). One alternative way is to produce the overlay texture when the vector data is loading if the hardware capability is permitted.

In order to supply a gap that arise from the drawback of texture-based method, and support interactive manipulation and edit functions to vector data, we propose one fast geometry-based approach.

4.2 Fast Geometry-based Approach

Wartell et al. presented a geometry-based approach that based on dynamic LOD terrain model (Wartell 2003), so expensive computation cost is required in real-time rendering (Schneider 2007). In our approach, we develop static triangle mesh (STM) to express DEM blocks on the base of SOAR algorithm presented by Lindstrom. STM is created by simplify grid DEM with one given threshold of error when grid DEM is down scaled at various resolutions. STM has a merit can support low computation cost. Although expression error is introduced into STM, it can be controlled by assign one possible lowest value to the threshold, and would not affect the utilization of DEM in certain degrees.

As one pyramid structure is used for loading DEM data in real time depend on viewpoint, so at each frame just a few blocks are refreshed, this make smooth rendering is feasible when display card has a good capabilities (e.g. 1G RAM).

Generally, the most computational time of geometry-based approach spend on triangles searching and intersection calculation. As STM is composed of right-angled triangle set, so geometry-based approach can be improved in great degrees.

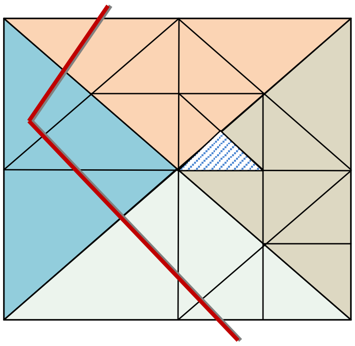


Figure 3. Static Terrain Mesh (STM) is composed of right-angled triangles, which support efficient searching and computation of intersect with random line segment.

4.2.1 Coding the triangles in STM: One STM block is built by split two triangles into two binary trees, it is easy to express triangles' position and direction in STM block with a set of binary code, so easy to locate one triangle by the binary tree code. E.g. the code of one triangle (filled with bias) in figure 3 can be expressed with code 1R2L3L4R5L, R esxpress right child, L express left child.

4.2.2 Intersect random line segment with right-angled triangle: There are eight directions of right-angled triangles exist in STM (see figure 3). It's easy to determine one triangle direction by its level and the fact that it is right or left child of a parent. However, one random line segment intersection with triangles has only four cases: respectively with vertical side, horizontal side, 45 degree longest side and 135 degree longest side. The intersection algorithm of the four special cases can be coded and executed efficiently.

4.2.3 3D polygon geometry primitives: the creation of 3D polygon geometry primitives is complex as it needs to determine it's inside surface that is usually not one plane. In order to keep coincident with terrain surface, the aim of 3D polygon geometry creation in our approach is to find triangles of STM block is inside and intersect with 2D vector polygon.

It's easy to determine intersected triangles with given line segment position and code of triangles in STM. When several vertices of the polygon are inside one triangle, the intersected part can be simply triangulated into a fan of triangles (figure 4), as all these vertices are in one plane.

In order to determine inside triangles, we use horizontal scan lines to confirm whether one triangle is inside a polygon, e.g: if one triangle intersect with a scan line but not intersect with polygon boundary, then it is inside the polygon. The interval of scan line can use value that a bit smaller than the horizontal side of the lowest level triangle (figure 4), scan line method can process cases of concave and convex polygon.

The 3D geometry primitive of one polygon is a 3D mesh, which is composed of all inside triangles and triangulated part of intersections. In order to avoid z-buffer artifacts, one small offset value should be added to z part of each vertex.

For large-scale vector data, only vector objects that close enough to the viewpoint are rendered with geometry-based method, hence to improve render speed.

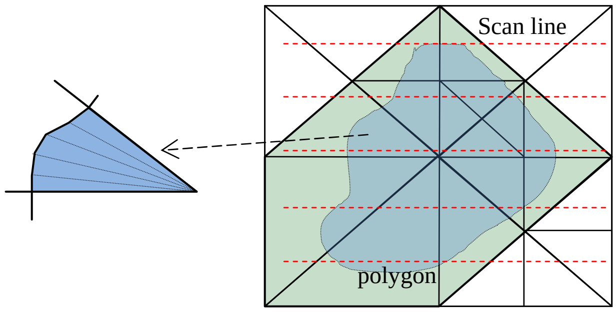


Figure 4. Creation of 3D geometry primitives of 2D polygon

5. INTERACTIVE MANIPULATION

5.1 Picking Operation

When vector data is rendering by texture-based technique, picking operation can be realized by project picked terrain point onto the plane of 2D vector data to find selected vector objects, this idea was proposed by Kresting et al. (2002). When 3D geometry primitives are created, picking operation can be performed with similar approach by project 3D geometry and picked 3D terrain point onto one 2D plane, as the computation in 2D plane is more simple and efficient than in 3D space.

5.2 Manipulation of Vector Objects

Manipulation of vector objects including edit vertex, move, rotate or resize one object, and change properties of one object, and etc. As 3D geometry primitives are independent of STM blocks, so the manipulation only performs on 3D vector object. When one 3D vector object is moved, rotated or resized, the old vector data is removed, and the new vector data is produced with section 4b introduced approach. If one vertex of polyline or polygon is edited, then the 3D geometry is refreshed by the same approach.

6. RESULTS AND DISCUSSION

Rendering vector data can enhance many invisible information in 3D landscape map, and made 3D landscape map more close to a 3DGIS. However, large-scale vector data generally need to be rendered in 3DGIS. Both pyramid structure and index of quadtree can improve large-scale vector data of multiple scales to be load easily in real time. In order to overcome expensive computation of real time intersection of terrain mesh with vector line segments, we built STM, as it has a special structure, so 3D geometry primitives can be created in preprocess or in real time browsing process, and interactive manipulation can be realized easily.

Consider the compatibility of the system to 2DGIS, 3D geometry primitives are created in real time. In order to avoid massive data computation, only these vector objects that close to the viewpoint are processed and switch to 3D vector displaying, once 3D geometry primitives are created for one 2D vector object, then it is recorded and saved until some edit is required.

In figure 5 some results obtained with our method are shown. It is capable of visualizing line and polygon features. Local objects that are rendered in raster mode switch to vector mode can be implemented without effects on real-time smooth browsing.

7. CONCLUSION

In our approach, we render vector data precisely and efficiently with a hybrid technique in 3D landscape map. Texture-based technique is used to improve large-scale vector data rendering in areas that locate far from viewpoint. Geometry-based technique is used at near viewpoint and when interactive manipulation is required. For fast 3D geometry primitives produce in real time rendering, DEM pyramid is built with blocks of STM, which is indexed with binary tree and composed of right-angled triangles that can support efficient computation.

ACKNOWLEDGEMENT

We would like to thank MOST (Prj. No: 2007AA12Z234) and NSFC (Prj. No: 40620130438) of china for their financial support to this work.

REFERENCES

Stephen M. Landscape Visualization: Progress and Prospects, ESRI 2004 UC Proc, 2004. URL: http://gis.esri.com/library/userconf/proc04/docs/pap1647.pdf (accessed 5 May. 2008)

Mark D, Murray W, et al. Roaming Terrain: Real-Time Optimally Adapting Meshing. In: *Proc. of the IEEE Conf. on Visualization 1997*. 1997. 81-88.

Ulrich T. Rendering massive terrains using chunked level of detail control. In: *Proc. of the ACM SIGGRAPH 2001*. 2002.

Lindstrom, P., Pascucci, V. Visualization of Large Terrains Made Easy. In: *Proceedings of IEEE Visualization 2001*, 363-370, 2001.

Wartell Z., Kang E., Wasilewski T., et al. Rendering vector data over global, multi-resolution 3D terrain. In: *Proceedings on the Symposium on Data Visualization*, volume 40, pages 213–222, 2003.

Schneider M., Guthe M., and Klein. R. Real-time rendering of complex vector data on 3d terrain models. In: *Proceedings of The 11th International Conference on Virtual Systems and Multimedia*, pages 573–582, 2005.

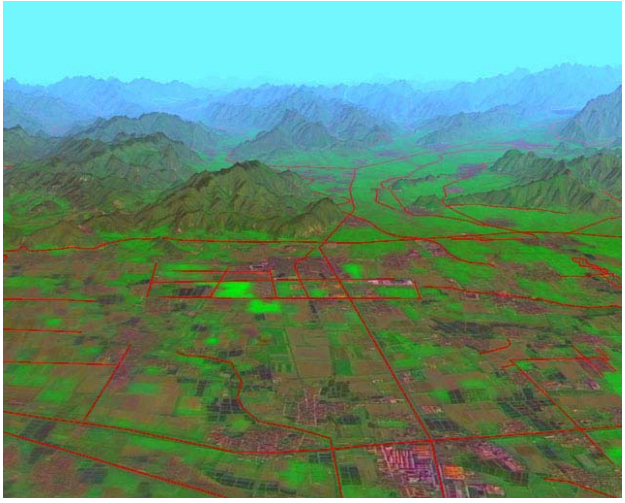
Schneider M., Klein. R. Efficient and Accurate Rendering of Vector Data on Virtual Landscapes, 2007

Kersting O., Dollner J. Interactive 3D Visualization of vector data in gis. In *Proceedings of the 10th ACM International Symposium on Advances in GIS*, 2002.

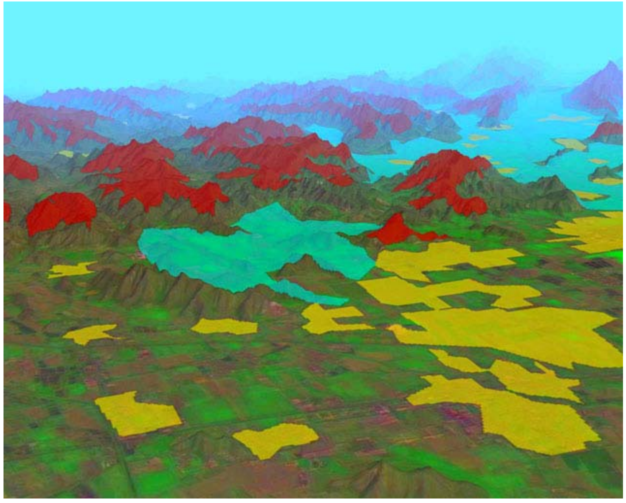
Dollner J., Hinrichs K. An object-oriented approach for integrating 3D visualization systems and GIS, Computers & Geosciences, (26) 2000, 67-76

Stephen B., Jacqueline L., and Whalley, Multilayer hybrid visualizations to support 3D GIS, URL: www.sciencedirect.com (accessed: Mar.1 2008)

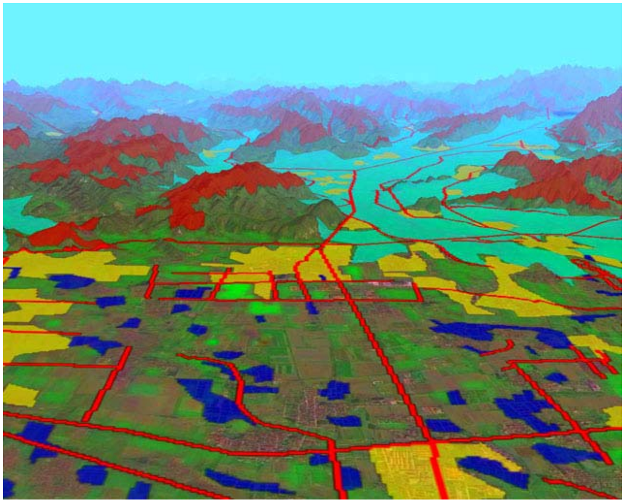
Sun M., et al, 3D Visualization Method of Large-scale Vector Data for Operation, in: *Proc. of the Fourth International Conference on Cooperative Design, Visualization and Engineering*, 2007, Shanghai



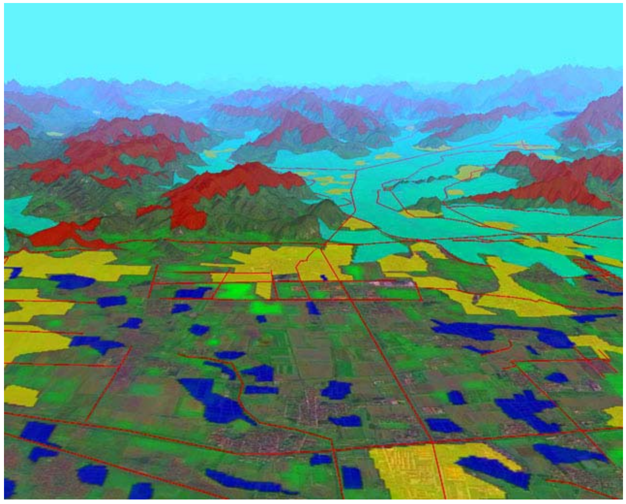
a) Line objects



b) Polygon objects



c) Vector objects rendered with texture-based approach



d) Line objects switch to be rendered with geometry-based approach

Figure 5. Line and polygon objects are overlayed on terrain with our hybrid system.