

## Research on the Key Technology of Massive Vector Graphic Displaying

Jun Wu

Electronic Engineering Department  
Naval University of Engineering  
Wuhan, China  
guangguzi9605108@163.com

Dan Xu

Electronic Engineering Department  
Naval University of Engineering  
Wuhan, China  
xudan0303@163.com

**Abstract**—The paper introduces a new method which can load TOPO elements more quickly. After analyzing the problem of the old TOPO component, this paper gives the solution containing method of data loading management and display control. The realization is presented at the end of the paper.

**Keywords**—TOPO component; GIS; data load management; mesh;

### I. INTRODUCTION

The old TOPO component has some problems in massive objects displaying. The biggest problem is that the consumption of the memory will increase and the displaying will take longer time with the increasing number of the display objects. We lead the map displaying technology of GIS (Geographic Information System) into the TOPO component.

### II. PROBLEMS

#### A. Data loading problem

According to TOPO coordinates range of the viewport, the grid data will be loaded. In order to reduce the count of loading data, we can expand the scope of the loading data when moving the graphic. To be exact, we expand on a row and a column. When loading the data, we also need to determine whether the data has been loaded, in order to reduce the amount of data loading [1]. When displaying, all the mesh data in the viewports range should be drawn, even if the data is not in the viewport, because it could avoid the time-consuming judgment.

When selecting the objects according to the coordinates and querying the TOPO objects, we must calculate the selected grid according to the coordinates firstly and then judge whether this grid contains any object. Because point element has a certain width and height, it is necessary to judge the mesh boundary when selecting the surrounding mesh objects [2]. In accordance with the ID to find the TOPO object, first read ID of the mesh, and then select the TOPO data from this mesh.

#### B. Layered Filtering Problem

The elements are stored in each layer vertically, and each layer corresponds to the scaling map, which can reach a maximum balance value at the range of unit viewport when map zooming, filter and display at a certain level to ease the pressure of displaying and memory.

#### C. Displaying Problem

In order to avoid delaying time in the process of displaying, rendering a view must be completed in 100 milliseconds (10 frames per second). If completed in 50 milliseconds, the display interface will be very smoothly. Through the efficiency test of Java, 300,000 elements' simple traversal (do nothing, only traversal) need 50 milliseconds. The TOPO package is needed to manage 100,000 TOPO elements, in order to complete drawing within 100 milliseconds, the data must be partitioned, which only processing TOPO data in viewport [3].

### III. SOLUTION

#### A. Data Loading Management

- Mesh (grid)

A rectangular area is split to series of small two-dimensional rectangles, and the length and the width of each small rectangle is the same. From the left top corner to the right bottom, each rectangle is named by a 32 bit integer number. The higher 16 bits represent the X direction, the other lower 16 bits represent the direction of Y. In order to calculate more rapidly, the length and width of Mesh are the same value, and is 2 to N power. So we can know all directions of the serial number by coordinates displacement (to the negative coordinate first absolute, then displace, lastly minus 1).

Calculation formula of TOPO Point coordinates and Mesh serial number

```
short meshx = Abs(ViewPoint.x) >> N
if (ViewPoint.x < 0) {
  meshx = -1 * meshx - 1;
}
short meshy = Abs(ViewPoint.y) >> N
if (ViewPoint.y < 0) {
  meshy = -1 * meshy - 1;
}
int meshID = ((meshx) << 16) + (meshy & 0xFFFF)
```

- the data load increment

The scope of the Mesh covering the viewport is firstly determined. The scope of loaded data is determined by left top corner and the scale. The Mesh number can be calculated by the corresponding TOPO coordinates at left top corner of viewport, the left top corner of loading range of Mesh is that the upper left corner X and Y values of Mesh minus 1. Then calculate Mesh number by TOPO coordinates of the right

bottom corner, the right bottom corner of loading range of Mesh is that the  $X + 1$  and  $Y + 1$ . The acquired rectangular area from the left top corner to the right bottom is the scope of the Mesh data [4].

Judging whether the viewport coverage of Mesh exist in the module of data management. If exist, do nothing. If not, call the data loading module to load the Mesh data.

Judging whether data in the module of data management is the data in viewport coverage of Mesh, if not, delete the Mesh data.

The loading area is shown below. Vertical and horizontal cross lines represents the mesh line, a lattice represents a mesh. The Red areas represent the loaded area.

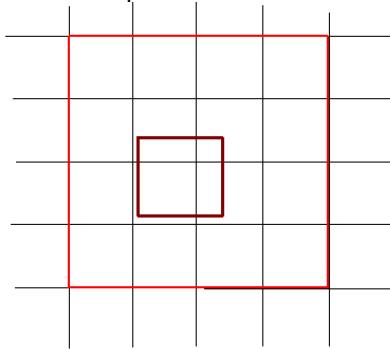


Figure 1. Loaded data area.

### B. The TOPO element display control

- The TOPO elements

When drawing TOPO elements, all TOPO displayed data from the data management module. The number of displayed data is small, and the saving efficiency can be ignored, but this increases the complexity of code; so there is no need to consider whether the data is in the current viewport.

All drawing of TOPO elements are completed through the draw interface itself. The TOPO element drawing structure is controlled by the elements of the member variable. Before drawing, set the control interface of drawing provided by the object and the element can be self mapping according to user requirements. Whatever the plotting position, shape, picture, linear, rendering methods, are all set by setting interface. In order to provide advanced display effect, set the Render object when drawing method and use Render provided for drawing, this drawing is finished in draw.

- The TOPO hierarchical control

Simulating the GIS map, the proportion and the layer correspond to each other, in order to control displaying elements in each layer.

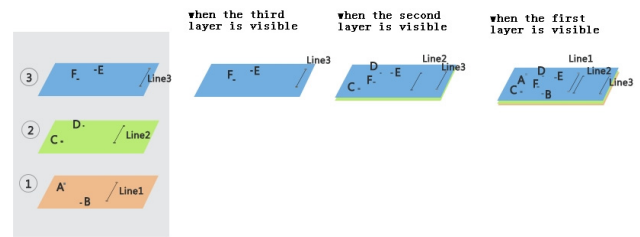


Figure 2. The layers of the data.

Through setting layers of topological elements, it can control the displaying. For some layers, it is special. For example, the line required global display, we put it on top of the other; for ordinary lines, if the data are too many, we set the default displaying layer's constraint, shows no general line or filter displaying line to achieve the requirements of finite viewport elements [5].

For large scale, when the view shrinks to a certain extent, it should be in accordance with brief presentation of the TOPO element, so that the user can see the overall structure. Large scale specific value can be set by user, judge TOPO element drawing. Briefly the way including but not limited to the following:

- 1) Omit drawing certain content, such as text etc;
- 2) skeleton displaying, for example, not to display pictures but only shape; width is set to the minimum;
- 3) No complicated rendering drawing operation;

### C. Display control

- The viewport

The TOPO control is a rectangular window for the outside presentation of topology information. The length and width of the viewport are defined by user and can be adjusted according to the needs of users. The origin of the viewport is the left top corner, to right or down is the positive direction. Viewport coordinates become logical coordinates range (0,0) to (viewport viewport width, height)

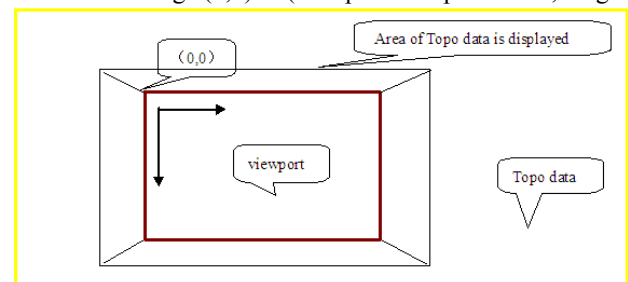


Figure 3. The relationship of viewport coordinate and topo data.

- Coordinate mapping mode

In order to display TOPO element data correctly in the viewport, coordinate of TOPO element data must be changed into logical coordinate. When TOPO diagram mobile or zoom, viewport coordinates will be changed, the origin of (0,0) will correspond to the location (View Point) of the TOPO coordinates. The size of TOPO coordinate

represented each unit of the viewport (pixel) is scale, the TOPO coordinates (TOPO Point) and logical coordinates (Logic Point)'s simple mapping formula blows:

$$\text{LogicPoint.x} = \text{TOPOPoint.x} / \text{scale} + \text{ViewPoint.x}$$

$$\text{LogicPoint.y} = \text{TOPOPoint.y} / \text{scale} + \text{ViewPoint.y}$$

- Moving of the viewpoint

TOPO coordinates range may be larger than the range viewport coordinates, in order to view all TOPO elements, you will need to move the viewport according to user specified manner, such as the mouse to drag the viewport, set the display coordinates, the scroll bar operation and so on. No matter which way are mobile, TOPO coordinates directly change the viewport origin coordinates representing (Viewpoint), and then re read the data and refresh the viewport display [6].

- Zooming of the viewport

Viewport zoom (zoom in and zoom out) refers to the viewport, fixed point TOPO element for visual range adjustment (scale), according to the coordinate mapping mode description, as can be seen, increasing the value of scale, is to expand the range of TOPO elements, also is to reduce operation; reduce scale value vice. Viewport zoom in addition to change the value of TOPO coordinates, the viewport origin coordinate representative (ViewPoint).

#### IV. CONCLUSION

Due to the TOPO graphic element management of large scale, if the element uses the observation mode, memory consumption is huge, which is not conducive to the persistent data. All the drawing elements this level by reference to save graph, call the corresponding method to directly provide for the notification of property changes, FHCCanvas in FHTOPO through the management of the observer pattern to notify event message.

A bird's eye view to show the global topological thumbnail, which provides basic camera switching, change operation, users can view to monitor the whole topology.

A bird's eye view of the topology map scaling to a certain size of display, arbitrary topological changes to elements will lead to a bird's eye view of the change, and drawing program at present is partial rendering, the viewport outside the region are not drawn, and the bird's eye view on global so need to draw the global view, so if keep distortion will bring significant overhead for rendering, because of the particularity of bird's eye view, so for a bird's eye view rendering details can be properly, meet the real-time requirement:

- The sample chart (the lower left corner of the eye):

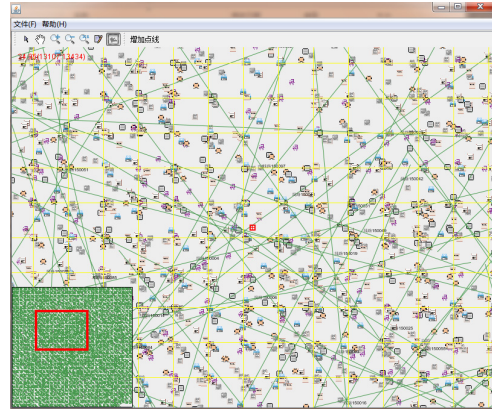


Figure 4. The sample of the application.

- The rubber band effect

For the topological graph elements of the editor, if we change the object directly in accordance with the pixel changing, it will often bring unnecessary overhead, and general operation, intermediate process actually do not need, such as user mobile elements. In fact we need to know the final position of the element in the process of moving elements. The user need only know that the trajectory which is rough. The rubber band effect is processed in the double buffer basis, which can save unnecessary operations consume resources.

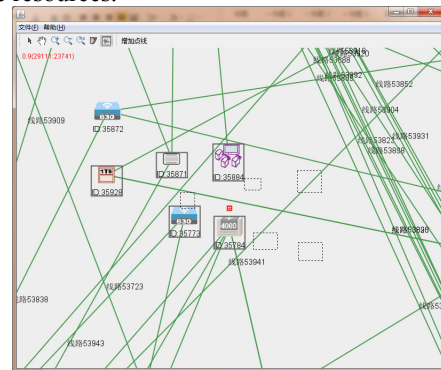


Figure 5. The sample of the rubber band effect

The dashed box as an epitome of a series of moving point, in drag to the process, other topological elements in the region class need not to be drawn, draw related elements can only move, users can know the track, also know the final operation, when the user releases the mouse really will be the corresponding element move to the specified location, so as to avoid multiple rendering costs.

#### REFERENCES

- [1] Brakken, Rick; Ait-Laoussine, Taj; Steffel, Stephen J.; Sikes, David, "Integrating customer load and GIS data for improved distribution planning & operations" Proceedings of the IEEE Power Engineering Society Transmission and Distribution Conference, p 1056-1063, 2006
- [2] Brakken, Rick; Ait-Laoussine, Taj; Sikes, David; Steffel, Steve "Integrating customer load and GIS data for improved distribution

- planning & operations” Proceedings of the IEEE Power Engineering Society Transmission and Distribution Conference, p 8-14, 2006
- [3] Aji, Ablimit; Sun, Xiling; Vo, Hoang; Liu, Qioaling; Lee, Rubao; Zhang, Xiaodong; Saltz, Joel; Wang, Fusheng “Demonstration of Hadoop-GIS: A spatial data warehousing system over MapReduce” Proceedings of the ACM International Symposium on Advances in Geographic Information Systems, p 518-521, 2013
- [4] Masashi; Kamei, Katsuyuki; Tamada, Takashi; Goto, Yukio; Seo, Kazuo “Dynamic load-balancing and control for server-side rendering on Ajax-GIS” IEEJ Transactions on Electronics, Information and Systems, v 133, n 4, p 891-898, 2013
- [5] Chen, Jun Wang, Huajun; Lu, Hanyu “Research for GIS raster data storage and display under the frameWork of private cloud” 2012 International Conference on Computer Science and Electronics Engineering, ICCSEE 2012, v 2, p 449-452, 2012
- [6] Buayai, Kittavit Ongsakul, Weerakorn; Nadarajah, Mithulananthan; Paurmchai, Koollachat “GIS based distribution load flow for better planning and operation” Proceedings of the 2011 International Conference and Utility Exhibition on Power and Energy Systems: Issues and Prospects for Asia, ICUE 2011