VECTOR GEOSPATIAL DATA COMPRESSION FOR WEBGIS AND CARTOGRAPHIC GENERALIZATION

Li Qingyuan
The Key Laboratory of Coal Resource, Ministry of Education
Institute Road Ding 11, Beijing, China 100083
Liqy@casm.ac.cn

KEY WORDS: WebGIS, vector geospatial data, data compression, cartography generalization

ABSTRACT

This paper introduces the author's work on the compression of vector geospatial data intended for transport on the Internet. WebGIS vector data format is shifting to ASCII code such as XML. In some applications, however, the technique of vector map data compression is still needed. The author proposes four innovations regarding WebGIS vector spatial data compression. (1) Not only integer, but also short data can be used to replace traditional double or float data of WebGIS. (2) A formula to calculated width of 'virtual screen' is proposed, (the formula is 'vituralScreenWidth = dispWidthPixel*maxEnlargeRatio / stepWidth'). The 'virtual screen' is used to map the original double or float geographic coordinate data to short data. In this formula, if stepWidth equals 1, the compression has no loss in precision. If stepWidth is larger than 1, the compression is less precise. (3) Three ways of dealing with the special case of long segments are proposed in the paper. The first is 'long segment add points', which fits in with having only few long segments. The second is 'long-short segments grouping', the third is 'respective deal with on layer', the second and third ways fit in the case many long segments, such as a traffic route map or street block map. After these operations a vector data map may be compressed to 1/5 or less than the original data. And (4) when the 'vituralScreenWidth' is given different values, it can form a 'hierarchy' of geospatial data versions, somewhat like cartographic generalization, but having the problem of feature selection.

1 INTRODUCTION

At present, geospatial data transmission mainly relies upon the raster data model. Although the raster model proves efficient for a number of reasons, there are many reasons to use vector data in Internet GIS (Buttenfield, 1999). The vector geospatial data format of Internet GIS or WebGIS is shifting from binary code to the ASCII code of GML/SVG. Vector geospatial file will become very big. Despite the fact that the Internet bandwidth is becoming wider and wider, (in many cities the Internet can even run real time movies) but in many applications, bandwidth is still a bottleneck. For example, a handheld wireless device or in rural dial-up, bandwidth is still insufficient to map applications. Another case; data owners hope that a visitor could view the map but not be able to take the data without approval. Therefore, compressed binary vector geospatial data, which is self-encrypted, is still needed. The technique of compression and encryption to vector geospatial data will not stop developing as Internet's bandwidth continues increasing, and GML/SVG being accepted.

Vector data compression also be called reduce storage. It had been studied many years ago in computer graphics and GIS. It has many widely accepted algorithms. Among them, the most adopted algorithm is 'Douglas Line Reduction Algorithm' (Douglas and Peucker, 1973). In computer graphics, integer and short data are often used as coordinates of graphics to quicken the operation. Some commercial GIS systems (e.g., SDE and SYSTEM 9) use integer store geospatial coordinates too. But, most GIS systems still use double or float data to store geographic coordinate. Storing chain ('offset' or 'increment') of arc is also a common technique to reduce storage in computer graphics and GIS (Goodchild and Douglas, 990).

In Internet, people often feel that operation to map is slow. It lets WebGIS developers again pay attention to vector data compression. Li et al (2000) suggest two methods to do it: (1) mapping double or float coordinates to integer coordinates, (2) 'storing chain' to arc points. Though, these are not new, but used in WebGIS, really effective. The author believes that for geospatial coordinates, not only integer, but also short data could be used in WebGIS applications.

'Virtual Screen', similar to "virtual frame buffer" in the textbook of Foley, van Dam, 1992, is needed for mapping data from double or float to integer or short. How to determine the size of 'virtual screen'? The author gives a formula to determine it.

When byte data as offset is used in this technique, as suggested by Li Q., there will be a problem of the 'long segment'. 'Long segment' is that the offset between two adjacent points is longer than 127 units. How to deal with these 'long segments'? The author suggests three methods, 'Long segment add point', 'Long-short segments grouping' and 'respective deal with on layers', to deal with it.

Vector geospatial data compression is somewhat similar to cartographic generalization. Except 'Douglas Line Reduction Algorithm', there are many other methods, such as 'Nature principle' (Li Z. and Openshaw S. 1992, 1993), 'Hierarchical coordinate system' (Dutton G., Buttenfield B.P., 1993, Dutton G. 1997, 1998, 1999), 'Neural network' (Thomas and Roberts, 1994), 'Delaunay Triangulation' (van der Poorten and Jones, 2001) and so on, have been used in cartographic generalization. The author also connects the compression with cartographic generalization. If the 'vituralScreenWidth' is given different values, it can form 'hierarchical' map data, somewhat similar to 'Nature principle', but the algorithm is simpler than it.

2 REPLACE DOUBLE WITH SHORT

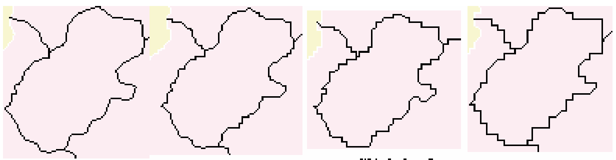
The author believes that in storing and transporting spatial vector coordinate data for WebGIS, not only integer but also short data can be used to replace traditional float or double data. We know that the range of short is -32768~+32767, or 0~65536. If map coordinates are transformed (mapped) to short data, could we lose map precision? Let us analyze this problem. When users view a map, they generally first overview the whole map, then they enlarge it to see a more detailed view. If the size of computer screen is 800*600 pixels, short data can allow a map of 800*600 be enlarged to 80 times without any precision loss.

From another point of view many spatial data of WebGIS are digitized from traditional paper maps. If a paper map's width is 1000 mm, 1 mm being divided to 10 map units (digitized precision is only so), the total map width is 10000 units. If width of a paper map is 6000mm(rarely so large map), the map total width is 60000 units, it is still smaller than short data range 65536. Obviously, for displaying a single map, short data can meet most WebGIS needs. When an area is made up of thousands of maps, we can add an offset in header of every map. In practice, the map data can be stored and transported as short. When it reaches client, offset data are added and transformed into integers. Even for multi-maps displaying, zooming in and panning, in this way there will be no problem.

3 'VIRTUAL SCREEN' SIZE

Geospatial coordinates, which are expressed by float or double data in the original geographical (or map) reference frame, are not simply be cut to short points, but are transformed to short points, which is expressed by short data in a 'virtual screen' reference frame. The algorithm is similar to drawing on a computer screen, but it is drawing on a virtual screen (actually it is stored to an array of arc). It can be seen that the ratio of reduced points in 'filtering compression' to arc points is in inverse ratio to the size of 'virtual screen'. The smaller of 'virtual screen', the more points are reduced, the higher of compression ratio is and the more precision lost and vice versa. If 'virtual screen' is large enough, 'filtering compression' can't cut down on any point, then the compression ratio of 'filtering compression' is zero. How to determinate the size of 'virtual screen'? Is client screen map displaying size (such as 600) set? Or is maximum range of short data (-32768--+32767 i.e. 65535) set? Within will do.

In the first case (600), ratio of data compression is very high. But, when the map is enlarged, obvious saw-teeth (or steps) will appear in the arc. In the second case (65535), if the range of map displaying is 600*600 in client computer screen, when the map is enlarged to 100 times, no saw-tooth or step appears in the arc, i.e. no precision is lost. Generally, the system has some limitations, which couldn't allow user to enlarge without limits, when enlarged to a fixed ratio, one can't continue enlarging or the system will load maps of the next grade. In most cases, this maximum enlargement ratio is far smaller than 100. When the maximum enlargement ratio is far smaller than 100 (such as 16), if virtual screen size is set to 65535(i.e. range: -32768 ~ +32767), there is a great waste of 'precision'. In other words, there are too many points that are transported to client and will never be used. In the case of slow transport, time will be wasted. It is necessary to balance speed and precision. Sometimes it is needed to reduce precision to reduce waiting time, i.e. it is accepted that arcs have some saw-teeth or jagged line in the maximum enlarge ratio. The pixel is the basic unit of the map displaying in a screen. The curve or inclined beeline, on enlarging to pixel grade, saw-teeth will appear in arcs. When the width of a step is one or two pixels, it is not easy to be distinguished by naked eye; when the width of step reaches 3 pixels, the naked eye can distinguish it; when reached to 4 pixels, the step or saw-teeth are very obvious even by naked eye. Figure 1 is a detail of a 1:4,000,000 map in maximum enlargement ratio of 16 times (When larger than 16 times, 1:1,000,000 maps will be loaded). In the enlargement ratio of 16 times, if stepWidth is 1, there are no saw-tooth; else if step is 2, there are no obvious saw-teeth; else if stepWidth is 3, there are saw-teeth appearing in arcs; else if stepWidth is 4, the saw-teeth of arc are very obvious.



Smooth case of different setpWidth (Figure 1)

When enlarged to maximum ratio, the next grade (1:1000000) map will be loaded immediately, if the client could bear with the step of 3 pixels in maximum enlargement ratio (when the enlargement ratio smaller than 16, the step width of saw-tooth is smaller than 3 pixels). By 'filtering compression', the number of the map arc's point may be reduced 36.2%. It will reduce 1/3 of waiting time. So, the saw-teeth have its value. The author proposes a formula to calculate virtual screen width as follows:

virtualScreenWidth =dispWidthPixel*maxEnlargeRatio/stepWidth (1)

Where: virtualScreenWidth: virtual screen width of mapping

dispWidthPixel: client screen displaying width pixels .

maxEnlargeRatio : maximum enlargement ratio.

stepWidth: number of pixels of saw-tooth step width in maximum enlargement ratio

Calculating the formula to derive virtualScreenHeight is similar to the formula given above

4 FILTERING COMPRESSION

When the size of 'virtual screen' is determined, the original points can be transformed to the 'virtual screen' reference frame. In the transformation, 'filtering compression' can be done. The theory base of 'filtering compression' is that in original map data, the points of arc are very dense, when transforming them to 'virtual screen', it is so dense that several following points in one arc fall to one pixel of the 'virtual screen'. In one arc, only one of these points, which fall to the same pixel, is needed, the others are redundant, since they have no contribution to increase fine within the maximum enlargement ratio. The redundant points should be eliminated. 'Filtering compression' is to eliminate these redundant points. The method of 'filtering compression' is very simple. When transforming an arc to virtual screen only the points that differ from last accepted point in x, y can be accepted. Filtering is an effective way to compress vector arc data. 'Filtering Compression' has been used in many systems of GIS software. The author only emphasizes that in the formula (1), if the stepWidth is 1, the filtering has no loss in precision, else the stepWidth larger than 1, the filtering has loss in precision. People can make a selection in the level of refinement and speed.

This is an example. A map of China, the total number of points of the all arcs, which includes the arcs of polygon boundary and the arcs of line feature, is 598483. If the display range of a map is 600 pixels in the clients system, the maximum enlargement ratio for the clients is 16, in the maximum enlargement ratio, stepWidth of arc saw-teeth is given 1,2,3,4, calculated the width of the virtual screen of the on the formula (1) are 9600, 4800, 3200 and 2400. The numbers of the accepted points are 594561, 485234, 382087 and 313924. The point which are eliminated are 3922, 113249, 216396 and 284559. The compression ratios by the different stepWidth in filtering are 0.7%, 18.9%, 36.2% and 47.5%, as show in the table2. The map displaying effect is show in Figure 1 above.

Step Width	Virtual Screen Width	Left point number	Reduce point number	Compress ratio
0.4	24000	595977	2506	0.4%
0.8	12000	595335	3148	0.5%
1	9600	594561	3922	0.7%
2	4800	485234	113249	18.9%
3	3200	382087	216396	36.2%
4	2400	313924	284559	47.5%
8	1200	186424	412059	68.9%
16	600	106651	491832	82.2%
32	300	60891	537592	98.6%

Table1. The Relation of step width, virtual screen width, Left point number, Reduce point number and compress ratio

5 STORING CHAINS AND DEALING WITH LONG SEGMENTS

Li et al (2000) suggest using byte data as offset of continuous points to replace coordinate points in arc (i.e. storing chain of arcs, see Goodchild and Douglas 1990). It is based on the fact that after transformation to 'virtual screen', the distances of two connected points in the arcs are not large. Most of them are smaller than 127. The basic thought of 'storing chain' is that for an arc, begin point (x, y) needs to be recorded, the other following points only the offsets in two connected points are recorded, i.e. except begin point, two bytes can express a point in the arcs. In this way, it can reduce data quantity to 1/2 (short to byte) or 1/4 (integer to byte) after reduced 1/2 by double transformation into integer. Because most of map data are from arc (including polygon boundary and line), 'storing chain' is a very effective way to reduce data quantity. But the long segment problem should be noticed in using 'storing chain', i.e., some offset value in two following points of an arc may be more than 127 pixel (or less than –127 pixel), such as either traffic route or polygon position in map edge. These long segments may be only a little number, but if not taking especially protected method, it may induce map displaying into a mess-up. The author proposes three ways to solve this problem:

5.1 Long Segment Add Points

First way is 'long segment add points', i.e. adding some points into a long segment make the long segment become to several short segments, where every segment is shorter than 127. It may be needed to increase some points, but the long segments are few and far between. So, in transformation into 'virtual screen', 'storing chain' and 'long segment add point', the data increment is far less than the data reduction.

5.2 Long-Short Segments Grouping

The second way is 'long-short segments grouping'. If there are many long segments, such as in the traffic route, too many long segments add points will make 'storing chains' less effective. In this case, the author suggests another way 'long-short segments grouping', i.e. dividing an arc into short segment group and long segment group, for short segments using 'storing chain'; for long segments no 'storing chains', still recording coordinates. The data structure is designed in JAVA as follows.

```
Class ARC {
     Short              numOfSegmentGroups;
    SegmentGroup[ ]   segmentGroups; }

Class SegmentGroup {
    byte          isLongSegment;   // 0: false  ,1:true
    shortPoint    beginPoint;      // arc's begin point
    bytePoint     offsets[];       // isLongSegment==1,offsets=null
```

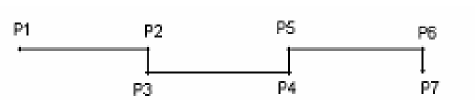


Figure 2. A long segment adjacent a short segment

```
shortPoint   points[];           // isLongSegment==0,points=null }
```

The worst case is the whole arc is a long segment adjacent a short segment, showing in Figure 2. In the worst case, 'long-short segments grouping' will spend more memory than common arc structure. Even though, it still can work. In most cases, this data structure will work perfectly.

5.3 Respective Deal with on Layer

The third way is respective deal with on layer. For the layer, which most segments are shorter than 127, such as river, road, boundary of district, storing chains of arcs, a small quantity of long segment can be dealt by 'long segment add points'. For the layers, which there are many long segments, such as bus route, building, street block, don't store chains of arcs, still store every point's coordinates.

6 EFFECT OF COMPRESSION

In the example, original point data is float. The map is composed of 5 files, including arc points, arc index, topology, line, polygon and annotation. The total size of the file is 1363,401 bytes. The author regroups and compresses them ('short replace float', 'filter compression', 'storing chains' and 'long segment add points' are used), then writes them to a new file. The width of virtual screen, size of compressed file, compression ratio, and possible understanding for the width of 'virtual screen' (dispWidth* maxEnlargeRatio /stepWidth) are show in table2. (Width*Ratio/stepWidth can have other meaning, for example, 600*1/2 = 300*2/2 =900*1/3).

When the width of virtual screen range varies from 300 pixels to 24,000 pixels, the size of compressed files varies from 55,295 bytes to 284,517 bytes. The smaller the virtual screen range is, the smaller the compressed file size is. When the width of virtual screen range reaches to 4800, the size of compressed file increases very small. 4800 of virtual screen width means that if width of client screen display map is 600 pixels, enlargement ratio is 16 times, the step of saw-tooth in arc is 2 pixels, show in Figure3.

VirtualScreenWidth	FileSize	Compress Ratio	Width*Ratio/stepWidth
300	55,295	95.95%	600*1/2
600	82,061	93.99%	600*2/2
1200	119,659	91.22%	600*4/2
2400	185,461	86.40%	600*8/2
3000	213,551	83.29%	600*10/2
4800-	276,173	79.74%	600*16/2
9600	284,466	79.14%	600*36/2
12000	284,517	79.14%	600*40/2
24000	284773	79.13%	600*80/2
24000	284773	79.13%	600*80/2

Table 2. Relation of VirtualScreenWidth, FileSize, CompressRatio and Width*Ratio/stepWidth

Figure 3. virtualScreenWidth=4800, dispWidthPixel=600, enlargement ratio=16, step of saw-tooth in arc is two pixels

7 RELATIONSHIP WITH CARTOGRAPHIC GENERALIZATION

Here, the author considers that the compression of vector geospatial data has some analogous with cartographic generalization. When the virtualScreenWidth is given different values, hierarchical map data versions, which have different displaying precision, are gotten. It is somewhat similar in spirit to cartographic generalization, which is developed by Van Hoor (1985), Li and Openshaw (1992, 1993), Zhan and Buttenfield (1996) and Dutton (1997, 1998, 1999). Of course, cartographic generalization needs to consider selection of features in different levels. A line feature or a polygon feature, even though, is transformed into a very small 'virtual screen', reduced to a point at the least. In cartographic generalization, it may be deleted. So, the grading and selection to the feature should be obeyed the principle of cartographic generalization. The selected features may be extracted by transformed to a virtual screen. It somewhat similar to 'Nature principle', but the algorithm is simpler than it.

8 A FEW NOTES

These points should be noted.

(1) After the data are transported to clients, the first operation is comeback offset arc data to normal arc data. It lets the compression of data not disturb to displaying and inquiring.

(2) There is no precision loss in 'storing chain'. When virtual screen width set to too large, the number of segments in 'long segment add points' will increase.

(3) The compression methods of this paper can't eliminate the redundant points in the beeline arcs. Therefore, geospatial data, after being transformed to 'virtual screen' but before 'filtering compression', should be processed with Douglas or other algorithms to eliminate the redundant points in a beeline arcs.

(4) The compression has no effect to the data of point features, annotations, indexes and topology.

9 CONCLUSION

The author would like to emphasized these conclusions:

(1) The technique of compression to vector geospatial data will not stop developing as the Internet bandwidth increasing, and GML/SVG being accepted.

(2) To compress vector map data, the effective way is transforming traditional coordinates from double or float into short 'virtual screen', the width or height of virtual screen is calculated as: Width=dispWidthPixel*maxEnlargeRatio/stepWidth

In the process of transformation, 'filter compression' and 'storing chains of arc' are needed. If stepWidth<=1, there is no precision loss, if stepWidth>1, there is precision loss.

(3) To deal with long segment, three ways of 'long segment add points', 'long-short segments grouping' and 'respective deal with on layer'can be used. The first suits a few cases long segment. The second and third suits too many long segments.

(4) The Compression is somewhat similar in spirit to 'Nature principle' of cartographic generalization. The selected features may be extracted by transformed them to a 'virtual screen'.

ACKNOWLEDGEMENT

The research was funded by Natural Science Foundation of China (SES 49771059) and Visitor Scholar Foundation of Education Ministry Higher Education Key Laboratory of China. Prof. David H. Douglas assisted with the English. Liu Xiaodong assisted with the programming. Errors and omissions are those of the author.

REFERENCES

Buttenfield, B.P., 1999. Progressive Transmission of Vector Data on the Internet: A Cartographic Solution, Proceedings, 18th International Cartographic Conference, Ottawa, Canada, 16-19.

Douglas, D.H. and T.K. Peucker, 1973. Algorithms for the Reduction of the Number of Points Required to Represent a Digital Line or Its Caricature. Canadian Cartographer, 10(2): 112-122.

Dutton, G., 1997. Digital Map Generalization using a Hierarchical Coordinate System. Proc. Auto Carto 13. (Seattle, WA) Bethesda, MD: ACSM/ASPRS: 367-376.

Dutton, G., 1998. A Hierarchical Coordinate System for Geoprocessing and Cartography. Lecture Notes in Earth Science. Berlin: Springer: 230 p.

Dutton G., 1999. Scale, Sinuosity, and Point Selection in Digital Line Generation. Cartography and Geographic Information Science, 26(1): 33-53.

Dutton, G. and B.P. Buttenfield, 1993. Scale Change via Hierarchical Coarsening: Cartographic properties of Quaternary Triangular Meshes. Proc. 16th Int. Cartographic Conference. Koln, Germany: 847-862.

Foley, D., 1992. Principles of Interactive Computer Graphics.

Goodchild, M.F. and D.H. Douglas, 1990. NCGIA CORE CURRICULUM in GIS, www.geog.ubc.ca/courses/klink/gis.notes/ncgia/u31.html#unit31

Li, Q. and Yang, C. and Cheng, A., 2000. Research on Geographical Relational Database in WebGIS. Journal of Chinese Image and Graphics, 5(A,2): pp119-123.

Li, Z. and S. Openshaw, 1992. Algorithms for Automated Line Generalization Based on a Natural Principle of Objective Generalization. Int. J. GIS, 6(5): 373-389.

Li, Z. and S. Openshaw, 1993. A Nature Principle for the Objective Generalization of Digital Maps. Cartographic Information System, 20(1): 19-20.

Thomas, W. and W. Robert, 1994. Use of Neural Networks in Line Generalization. EGIS: 76-85.

van der Poorten, P.M. and C.B. Jones, 2001. Customisable Line Generalisation using Delaunay Triangulation, www.comp.glam.ac.uk/pages/staff/pmvander/

Van H.E., 1985. Generalization of Cartographic Database. Proc. Auto Carto 7. (Baltimore, MD) Bethesda, Maryland, ACSM/ASPRS: 532-540.

Zhan F.B. and Buttenfield B.P., 1996. Multi-scale Representation of a Digital Line. Cartography and Geographic Information System. 23(4): 204-228.