MULTI-REPRESENTATION GEOGRAPHIC DATA ORGANIZATION METHOD DEDICATED FOR VECTOR-BASED WEBGIS

Qian Xinlin[a*], Zhu Xinyan[a]

[a]State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, 430079, P.R. China - Qianxinlin@gmail.com

WG IV/5

KEY WORDS: Multi-Representation, Data Structure, Vector Data, Zoom Quadtree, Vector Pyramid, WebGIS, WFS

ABSTRACT:

The paper proposed a Quadtree-based vector data structure which supports multi-representation of geographic data: Zoom Quadtree, it aims to meet the requirement of vector-based WebGIS in efficient retrieval of geographic vector data at different scales. According to the characteristics of map display in WebGIS that maps mostly rendered on the electronic display device, this paper introduced a concept of virtual screen, we divided vector data into tiles which use the size of regions of Quadtree-node as basic unit, and these tiles are fundamental data containers which were efficient and convenient in response to client's window query, and these tiles also are visible snapshot of geographic data in different scale which can be directly displayed in the virtual screen with decent density. The zoom quadtree structure assured that window block data with any scale can be obtained in limited number of Disk I/Os, after discussing and analysing existed multi-scale models and structures, we gave the principle and technique in create and maintain Zoom Quadtree. Issues like dynamic update of vector data and the possibility usage of these structures in VGI –integrated application are also discussed.

1. INTRODUCTION

1.1 Requirement of vector-based WebGIS

With the rapid development of Internet and Web, WebGIS became an important aspect of GIS study, WebGIS provides the most efficient and low-cost way in dissemination and sharing of geographic information for public and specific groups(Kingston, Carver et al. 2000; Anderson and Moreno-Sanchez 2003; Dragicevic 2004). But most existing systems of WebGIS today were implemented in raster-based model that the map displayed on client is rasterized into images before it transmitted to users. Such as HTML viewer of ArcIMS and the commercial online mapping products of Map request, Google maps, Yahoo Maps and Microsoft Virtual earth. The problems of this kind of WebGIS is lower flexibility and less functionality compared to desktop GIS software and few semantic information of geographical data can be queried interactively, These issues are expected to be resolved in vector-based WebGIS(Peng and Zhang 2004), moreover, online data creation which are essential to VGI(Goodchild 2007) will benefit much more from the vector-based WebGIS, so the subject of vector-based WebGIS is worth studying in GIS disciplines.

For vector-based WebGIS, geographic vector data instead of pre-rasterized image should be transmitted to users quickly and efficiently. So it impose a very high level demand on the geographic data retrieval process, if the client's request cannot be answered within a reasonable time, then the usability of WebGIS is inferior, Especially for WebGIS on the Internet, because the extensiveness of user groups may result in a large number of requests, to complete these simultaneous geographic data request with high-performance is a central issue of vector-based WebGIS. On the other side, WebGIS map will be displayed in different scales under users' interactive operation like zoom in/out to make it easier for user to visit geographic information in different areas or find their own areas of interest; multi-scale support for geographic vector data is another key characteristic for spatial database.

In order to provide the geographic vector data service in high performance for vector-based WebGIS, the paper proposed a Quadtree-based method in spatial vector data organization and storage, it can support multi-representation of geographic data and ensure the performance of window block data retrieval, at the same time can support high-performance geographic data insertions and updates. The basic idea of Zoom Quadtree is regional and semantic clustering of geographic data. That is, based on Quadtree, feature data stored in tiles in accordance with the feature's geographical location and extend as well as the importance value that indicates the level of the feature and computed from the semantic information. So the window block data retrieval can be accomplished in given number of Disk I/O. The rest of this paper is organized as follows, Section 2 give a brief overview of related works on the issues of vector-based WebGIS and multi-scale spatial database. Section 3 presents the Zoom Quadtree and its construction process, Section 3 also gives describe text for searching operation as well as insertion and deletion in Zoom Quadtree. Section 4 draw the conclusions, mainly discuss the applicability of methods in variant environment and the problems which may arise in practical application. The last part also covers the next step needs to be done along the way on this work (including consideration of temporal dimension integration, geographical data synchronization automatically update etc.).

* Qian Xinlin: PhD candidate, major in Photogrammetry and Remote Sensing, research interests include spatial database and GIS.

2. RELATED WORKS

2.1 Vector Transmission

In the issues of vector data network transmission, many relevant scholars put forward their own solution, the majority of these study stem from the idea of progressive-transmission vector data to improve the efficiency, Yang(Yang, Purves et al. 2007) made vector data compression based on the clustering point algorithm and transmit vector data progressively on the internet. Bertolotto(Michela and Max 1999)proposed progressive transfer model for vector data, it formalized the process of data simplification into several elementary operations, with this model, the vector data consistency and topology information can be retained. Yang(Yang, Wong et al. 2005) presents a series of measures and techniques to improve the performance of WebGIS, including data compression, client and server cache, space index on the data, computer clusters and multi-threading. Buttenfield(Barbara 2002) uses Strip Tree to organize the data storage and handle progressive transmission of vector data, ZU-KUAN WEI(ZU-KUAN WEI 1999) give a idea of vector data block division and network transmission, C. Paiva(Anselmo, Elvis Rodrigues da et al. 2004) made another data division schema through the recursive partition on data space and use data transmission based on data blocks.

2.2 Multi-scale Spatial Database

There are many research results about organization and storage of Multi-scale vector data from academics and scholars in related area, most research focused on multi-scale model of the database structure and solution of data storage and retrieval, Ting-Hua Ai reviews the multi-scale spatial database construction approaches and proposed multi-scale vector data storage method based on model of incremental cumulative on the primary-scale data. ooSterom(Peter van 1994) proposed Reactive trees and BLG tree as fundamental structure for Construction of spatial data after the process of cartographic generalization, Horhammer(Mike et al. 1999) proposed BANG-file based multi-scale data storage and retrieval methods, particularly focus on data clustering not only in their Geographical location but also consider their scale as an extra dimension, Chan(Chan and Chow 2002) proposed multi-scale R-tree structure based on Hilbert R tree, use PR-file alike data storage approach to store multi-scale Data.

2.3 Brief Review

Study results listed above respectively do the multi-scale research from different points of view, the former start the research from practical requirement of the network transmission of geographic data. These are mainly motivated by web and internet applications and GIS service. The latter research is earlier in time sequence. Mainly study the basic structure of the database to consider how to effectively carry out a multi-scale solution of geographic data management and storage. In today’s public-oriented WebGIS, spatial database needs not only to cope with lots of concurrent user requests, but require for near real-time response, the existing research results can not satisfy these requirements fully, applications ask for the Multi-scale geographic data retrieval capability on one hand, on the other hand require Retrieval of data in the server-side to minimize disk I / O to achieve performance. The existing commercial databases cannot support the multi-scale geographic data properly, utilize existing multi-scale data storage strategy on the top of commercial DBMS will introduce an additional layer for data manipulation, these will lead to the inevitable loss of performance.

3. ZOOM QUADTREE

Zoom Quadtree is a kind of quadtree(Hanan 1984) with additional advanced features. Each node of Zoom quadtree is responded to one or more block on disk. The feature data that belongs to this node is stored in the block, which node every feature should belong to is explained in detail in this section. All features are stored in the Zoom Quadtree nodes hierarchically, besides the original feature data, feature data that derived from the descent node using generalisation operations such as simplification, selection, aggregation, displacement, exaggeration etc also stored in the node. The elaborate descriptions of the Zoom Quadtree are given below.

3.1 Screen display and Virtual Screen

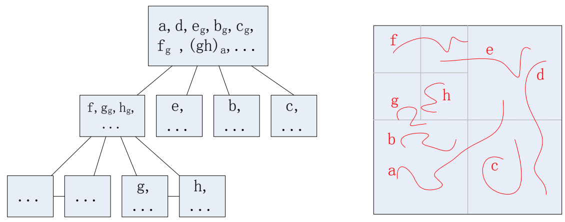
The primary characteristics of vector-based WebGIS is that the geospatial information is mostly rendered in the user's graphic display devices (CRT, LCD, projector, or large LED screen), It is quite different compared to traditional paper maps. As the computer terminals are interactive and it can take advantage of more advanced graphic techniques available (Kraak 2004), in order to achieve the purpose of conveying spatial knowledge, Characteristics of vector map displayed on the screen with different size should be concerned.

There are researches on information measurement and entropy of the map, Li Zhilin(Li and Huang 2002) developed some kinds of basic quantitative description of the map and used different quantitative indicators to represent the information amount of map and distribution of the symbols and the geographical phenomenon. There are also early studies for the map contents and map load, but lacks in establishing a direct numeric relation between geographic vector data volume and amount of information in map. classical conclusions in cartography is that appropriate value of the map load is 12 mm^2/cm^2, and the maximum map load should not exceed 15 mm^2/cm^2, there are also some empirical formulas, such as on the map every 100 cm^2 the number of residents selected during cartographic generalisation process shall not be more than 110. Based on the above-mentioned results and conclusions, this paper present a rough relation between size of the vector data and map information, based on this relation we associate the node of quadtree with map display.

After surveys about parameters and functions of mainstream display device, this paper introduced the concept of a virtual screen, resolution of the virtual screen is 2048 * 2048, Dot Pitch is 0.25 mm, equivalent to 100 DPI, the physical size is about 0.5 m * 0.5m, table 1 gives the details of the virtual screen, compare to display devices in real world, it is bigger and have higher-resolution. That is the information displayed on virtual screen is more abundant than real-world display device, so one frame vector data of virtual screen is enough for real display screen to draw. We presume single layer map displayed in virtual screen with fine aesthetics require vector data no more than 10,000 vertex. The number is for polyline and polygon features. For point feature, this number will be reduced to 500-800, because the point feature have a large number of annotations text, too many annotation text will degrade the readability of map. Each node in Zoom quadtree is corresponding to this virtual screen, each node itself is a geographic data snapshot view on its scale and position.

Parameter	Virtual Screen	
	mm	Inches
Width	500	19.7
Height	500	19.7
Pitch	0.25	0.01
Area	0.25 m^2	
Resolution	2048 * 2048	
Corresponding vector data size	10,000 vertex	

Table 1. Specification of virtual screen



a, b, c, ⋯, g denote the original feature data,
Subscript denote that feature is derived from original features
Subscript g donote the feature after simplification operator,
(gh) with subscript a denote the aggregation of feature g and feature h

Figure 1. Diagram of Zoom Quadtree

3.2 Zoom Quadtree Hierarchy

The Zoom Quadtree is based on traditional region Quadtree, use geographical coordinates for spatial reference, and partition the whole world space recursively into the same size subregions with no overlap to each other. low level node is fully contained by high level nodes(the direction is that root is high and leaf is low), each Quadtree node is related to one or more disk blocks, actual geographical data that use single feature as basic unit store in these blocks. Only geometry information of features will be stored in the node. Attribute and semantic information are kept separately from the node. Unique identifier is used to join the geometry and attribute information.

To determine which node one feature should be included in, it need to determine the leaf node level. According to the geometric extent of the specific feature, select the maximum extent value of all dimensions, use this maximum value to determine which level it should belong to, calculate the centre point of feature MBR, the node which this point falls into is the node which should store the feature.

3.3 Zoom Quadtree Operations

According to meta-data of specific data sets, we can figure out the best-scale of the data sets, draw map in the virtual screen under this scale, the geographic extent corresponding to virtual screen is the basic tiles size used to divide the dataset. We build the zoom quadtree from bottom-up; features falls into quadtree node fully will be stored in that node. Features whose extent beyond the region of current level node will stored in higher-level node and so on.

3.3.1 Zoom Quadtree Building There are two phases in building Zoom quadtree.

1. Features partition to node.
2. Consolidation on Zoom Quadtree using cartographic generalisation.

The zoom quadtree building process starts from the tree's lowest level. Features in datasets are partitioned to node using the bottom-up principle, if some features are beyond the extent of node, then they will be processed in a higher level, this recursive process continues until every feature is contained by one certain node of the zoom quadtree.

After features in datasets are allocated to certain nodes, the zoom quadtree became to a sparse quadtree, we say sparse because many nodes except for lowest node in quadtree only contain very few features or even empty. At this moment, the tree is just a traditional quadtree structure. We need to consolidate it to convert it to a Zoom quadtree.

Features in nodes should be processed using simplification operator. Features actually stored in the manner of important vertex prior to the less important ones like the store method used in PR-file(Bruno and Peter 1990), generally, the features contained in higher level node is more complex and bigger than those in lower level node, the simplification process can produce the multi-representation storage for complex and big feature. Every node in quadtree except leaf node has four child nodes. Features contained in these child nodes will be processed using generalization operator like selection, simplification, aggregation, displacement, exaggeration etc, the generalized data will keep a copy in the parent node, this kind of data are called derived data because it is produced through the generalization operator. The feature data from the original dataset is call original data. The process of derived data generation is also bottom-up and recursive. Figure 1 gives a visual describe for the Zoom quadtree. The generalization will base on the features' geometric information and semantic information. Derived data and original data are vertically linked implicitly by their global unique identification. Lowest level nodes in Zoom quadtree only contain original data. Any other nodes contain both original and derived data. These two kinds of data comprise the snapshot of the data in corresponding scale. The constraint of each node is that the snapshot contains vertex less than 10,000.

3.3.2 Zoom quadtree retrieval Data retrieval in zoom quadtree is up-down and straightforward. When handle a window query from the client, firstly, we determine the client's display scale and calculated which level in the quadtree will be the target level. After this, according to the query window, we traversal the tree up down until the target level is meet.

Each node encountered in process of up-down traversal will be examined to check if any origin feature data contained in this node intersect with the query window, if it does, the intersect part of the feature will be added to the result set. When the target level node is meet, all data including original and derived feature data in the target node will added to the result set.

For each window query, *n* disk block will be accessed, here *n* denote the number of hierarchy level from root to target node of the zoom quadtree.

3.3.3 Zoom Quadtree updates Dynamic update to zoom quadtree includes two steps:

1. Create / Edit / Delete original feature data.
2. Propagate update event upward along the tree to synchronically update the derived data.

When new features are created, we will find which node should contain the new features according to the principles introduced in the Zoom Quadtree building section and insert feature to it. If the node is bigger than 10,000 vertex, we will rearrange the node by delete some derived data. If the node is lowest node in the zoom quadtree, the node will be split into four child nodes with each containing corresponding feature from the parent. Parent removes features that have been allocated to the child and recalculate the derived data from child data. The update event will also propagate to ancestor node so higher level node can adjust the snapshot by recalculate related derived data to reflect the data change.

Feature deletion will cause the target node remove the original feature data and make its ancestor node to check the derived data that related the deleted feature and recalculate it.

Data edit process is similar to the data creation except that no new feature insert. When the extent of the feature changed, feature should move from one node to another, the process is composed by two sub-processes, delete old one and create new one. These two processes are already explained above.

4. CONCLUSIONS

The proposed Zoom Quadtree in this paper is aimed to resolve two main problems in the area of vector-based WebGIS; the first is efficient data retrieval especially for window query, the second is multi-representation of geographic data. Zoom quadtree uses the thought of geographic data clustering by their spatial location and semantic hierarchy information to improve the performance in window query data retrieval. At the same time, this paper introduces the concept of virtual screen and associate the quadtree node with the virtual screen and propose a hypothesis in map display that no more than 10,000 vertex per screen can result in a decent map. We use this concept and hypothesis to solve the problem of data multi-representations. Although there are data redundancies in the zoom quadtree, but it is our choice based on space-time strategy. The Zoom Quadtree here is used as containers to store the feature data and as structure to maintain a snapshot view of data at the same time.

We hope that this data structure can bring available and effective solution for vector-based WebGIS and it can provide the technical basis for the today's development of GIS, online VGI and the sharing and dissemination of space knowledge in public.

REFERENCE

Anderson, G. and R. Moreno-Sanchez (2003). "Building Web-Based Spatial Information Solutions around Open Specifications and Open Source Software." Transactions in GIS 7(4).

Anselmo, C. P., S. Elvis Rodrigues da, et al. (2004). A Multiresolution Approach for Internet GIS Applications. Proceedings of the Database and Expert Systems Applications, 15th International Workshop, IEEE Computer Society.

Barbara, B. (2002). Transmitting Vector Geospatial Data across the Internet. Proceedings of the Second International Conference on Geographic Information Science, Springer-Verlag.

Bruno, B. and W. Peter (1990). Spatial Priority Search: An Access Technique for Scaleless Maps, Albert-Ludwigs University at Freiburg.

Chan, E. P. F. and K. K. W. Chow (2002). "On multi-scale display of geometric objects." Data and Knowledge Engineering 40: 91-119.

Dragicevic, S. (2004). "The potential of Web-based GIS." Journal of Geographical Systems 6(2): 79-81.

Goodchild, M. (2007). "Citizens as sensors: the world of volunteered geography." GeoJournal 69(4): 211-221.

Hanan, S. (1984). "The Quadtree and Related Hierarchical Data Structures." ACM Comput. Surv. 16(2): 187-260.

Kingston, R., S. Carver, et al. (2000). "Web-based public participation geographical information systems: an aid to local environmental decision-making." Computers, Environment and Urban Systems 24(2): 109-125.

Kraak, M.-J. (2004). "The role of the map in a Web-GIS environment." Journal of Geographical Systems 6(2): 83-93.

Li, Z. and P. Huang (2002). "Quantitative measures for spatial information of maps." International Journal of Geographical Information Science 16: 699-709.

Michela, B. and J. E. Max (1999). Progressive vector transmission. Proceedings of the 7th ACM international symposium on Advances in geographic information systems. Kansas City, Missouri, United States, ACM.

Mike, H., rhammer, et al. (1999). Spatial Indexing with a Scale Dimension. Proceedings of the 6th International Symposium on Advances in Spatial Databases, Springer-Verlag.

Peng, Z.-R. and C. Zhang (2004). "The roles of geography markup language (GML), scalable vector graphics (SVG), and Web feature service (WFS) specifications in the development of Internet geographic information systems (GIS)." Journal of Geographical Systems 6(2): 95-116.

Peter van, O. (1994). Reactive data structures for geographic information systems, Oxford University Press, Inc.

Yang, B., R. Purves, et al. (2007). "Efficient transmission of vector data over the Internet." International Journal of Geographical Information Science 21(2): 215 - 237.

Yang, C., D. W. Wong, et al. (2005). "Performance-improving techniques in web-based GIS." International Journal of Geographical Information Science 19(3): 319-342.

ZU-KUAN WEI, Y.-H. O., JAE-DONG LEE, JAE-HONG KIM, DONG-SUN PARK, YOUNG-GEOL LEE, HAE-YOUNG BAE (1999). "Efficient Spatial Data Transmission in Web-Based GIS." Workshop on Web Information and Data Management.

ACKNOWLEDGEMENTS

The authors are supported by National Key Technology R&D Program(2006BAB10B03), Western Mapping Project of China, Hubei Natural Science Fund for Innovation Groups Projects (2006ABC010).