Visualization with Level of Detail at Multi-Scale in 3D GIS

Jun Li[1] Ning Jing[2] Maoyin Sun[3]

[1]*Department of Electronic Engineering, National University of Defense Technology*
Changsha, Hunan, China, 410073
E-mail: junli@nudt.edu.cn

[2] *Department of Electronic Engineering, Changsha Institute of Technology*
Changsha, Hunan, China, 410073
E-mail: ningjing@nudt.edu.cn

[3] *Department of Electronic Engineering, National University of Defense Technology*
Changsha, Hunan, China, 410073
E-mail: mysun@nudt.edu.cn

ABSTRACT Although 3D GIS becomes more and more important in many applications, such as urban planning, digital earth, there still exist many difficulties in database management, visualization, and etc. 3D GIS requires not only fast photo-realistic visualization, but also some generalization techniques such as selection, simplification at multi-scale. For example, at highest scale, all objects can be displayed. But at lower scale, only some important objects can be seen. Therefore, we should take account of two conditions about level of detail: generalization at different scale and different representation of detail at a certain scale. In this paper, by investigating level of detail in 3D GIS, we integrate generalization techniques with 3D visualization techniques on the basis of object-oriented database. In order to improve the performance of database access, we investigate the access structure based on modified reactive tree and analyze the detailed method of representing level of detail. This approach visualizes 3D object with object-oriented database and integrates generalization with 3D visualization techniques, 3D GIS is expected to achieve better performance of query and photo-realistic visualization at multi-scale.
KEY WORDS 3D GIS, level of detail, object-oriented database

1. Introduction

Geographic information system (GIS) have been experiencing a steady and fast development, we can model, store, process and display more and more geographic information. However many requirements still can't be fulfilled with traditional 2D GIS. In order to model and represent three dimensional (3D) spatial information, researches on three dimensional GIS (3D GIS) are booming up. For example, for urban planning, we need model, store and display some important 3D objects (e.g. buildings, trees and roads). In this digital city, user should be facilitated with spatial query, spatial analysis, fly-over visualization and so on.

Generally 3D GIS should represent different 3D data sets at multi-scales. At different scale, different geographic objects need be represented by generalization techniques: selection, simplification, aggregation, symbolization and etc. Thus different level of detail (LOD) is expressed. For example, at the biggest scale, all buildings should be displayed. While scale becomes smaller, some objects will be neglected and others will be displaced in the form of symbols.

It is very time-consuming for displaying all into database with high performance of data access. generalized objects in detail at a certain scale. Especially when some virtual reality functions such as fly-over and walk-through are needed to be implemented, the entire 3D scene can't be visualized real time along with constant move of viewpoint. Therefore it is necessary for 3D GIS to make use of 3D visualization techniques for LOD. For generalized geographic objects, different details will be displayed by size of objects, distance from viewpoint, visual angle and so on. This kind of visualization technique for LOD ensures fast and photo-realistic visualization of 3D scene. At the same time, because generalization techniques make use of management of database, 3D visualization integrated with generalization will exceed out of pure domain of computer graphics. Visualization for LOD can utilize the virtues of database (e.g. modeling, data sharing, distributed computation), and also implementation of spatial query and spatial analysis at multi-scale is guaranteed.

The most used generalization techniques include simplification, aggregation, symbolization, selection, exaggeration, typification and Classifica -tion. These generalization techniques store data These generalization techniques are directly related to visualization of geographic objects. There are different data representations at different scale for visualization with LOD.

3D visualization requires fast photo-realistic rendering for every 3D objects. In order to render 3D object in detail without depressing the quality of 3D scene, 3D objects should be divided into several LODs. Through selecting the LOD each visible geographic object lies in at a moment, each geographic object is rendered in a certain detail. It will decrease time consumption which is mainly caused by computation and rendering of detail data.

For visualization with LOD at multi-scale, 3D GIS should integrate generalization techniques with 3D visualization techniques, so that many functions such as spatial query, spatial analysis and fly-over are implemented together. The two kinds of techniques both represent LOD of 3D scene, but they focus on different objects. Generalization techniques focus on different object representation for entire scene at different scale regardless of visualization. 3D visualization techniques focus on rendering object in different detail. The integrated visualization with LOD should be based on object-oriented database, and proper data structure should be developed for fast data access.

This paper is organized as follows. In section 2, we introduce the previous work on visualization with LOD. In section 3, we describe the principles of integrated 3D visualization with LOD and data structure in object-oriented database. Based on extending reactive tree into 3D space, we modify the index algorithm and make it adapt to 3D space. Section 4 concludes the paper.

2. Related Work

Many researchers have investigated representation of LOD. Their achievements focus on two domains:

1)Computer graphics. Early In 1978, Clark proposed the principle that rendering of complex scene could be speeded up by adopting mechanism of LOD (Clark 1976). He presented that LOD of one geographic object is judged by the area of polygon formed by projection of the object into the viewport, compared to the area of the entire viewport. He also discussed how objects are rendered in different detail by distance from viewpoint. Now this kind of method is still applied broadly. Maciel proposed the concept of hierarchical LOD, and took account of more factors affecting judgement of LOD, such as visual angle (Maciel 1995). These representations of LOD focus on fast rendering of complex 3D scene. They don't make use of database management, so it is difficult to create data model.

2)GIS. Researches on this domain mainly focus on generalization techniques at multi-scale. Database management is used for modeling, storing and querying geographic objects. In 1991, Oosterom analyzed and concluded generalization techniques for LOD, and proposed reactive tree, which is a spatial index structure for LOD with high performance of data access based on R-tree (Oosterom 1993). Many generalization techniques (e.g. selection, symbolization) can be implemented successfully by reactive tree. The index structure for LOD has been used broadly in 2D GIS.

Many achievements have been made respectively in each of the two domains. But because there are few interconnections between the two domains, these achievements can't be applied successfully in many domains of interest, (e.g. Digital Earth, 3D GIS). By comparing of the two kinds of representations for LOD, we find out that the biggest discrimination between GIS and computer graphics lies in management of database (Kofler 1998). GIS applications are mostly based on relational database, object-oriented database or object-relational database. These database systems are provided with many functions, such as data security, multi-user access and incorporated query language. In the other way, applications of computer graphics are based on file system. The major reason is simple: data access is much faster.

Unlike other 3D applications, it is generally not possible to pre-load the entire scene in 3D GIS into main memory prior to visualization because of large amounts of data. Thus file system can't meet the requirement of 3D GIS applications. 3D GIS database should be built to manage large amounts of data (maybe up to hundreds of gigabytes). In 3D GIS database, many functions such as querying, analysis and remote access should be provided with. Currently there are some researches devoting to integrating database management with 3D visualization techniques. In 3D urban GIS developed by Stuttgart University and Rostock University in Germany, through distance from viewpoint, visual angle and other rules, geographic objects (e.g. buildings, trees and roads) can be rendered in different details with move of viewpoint (Koninger 1997). This system is on the basis of object-oriented database. Kofler made deep research on R-tree for visualizing large 3D GIS database (Kofler1996). Based on R-tree for fast data access in object-oriented database, geographic objects are organized into hierarchical tree by geographic position, and LOD is formed by containing and contained relation between minimal bounding box (MBB) of geographic object. Though these achievements make use of database

management
for 3D visualization, they don't consider merging any generalization techniques. This method limits its use in applications and not suited for 3D GIS applications at multi-scale. For our best knowledge, few papers discussed in detail the problems for visualization with LOD at multi-scale in 3D GIS.

3. Visualization for Level of Detail

3D GIS applications should represent LOD of the entire scene at multi-scale and render each visual geographic object in different details. Once we select a scale, 3D scene should be rendered fast without losing the quality. At the same time, spatial query and spatial analysis should be implemented with high performance at a certain scale. We built an experiment system, in which all geographic data are stored into object-oriented database 'jasmine' with proper data model. Based on object-oriented database, visualization with LOD is implemented initially in our experiment system. At different scale, selection, which is the most used generalization technique, is implemented to select only necessary geographic objects. Once the scale is fixed, selected geographic objects should be judged for respective LOD by volume of object, distance from viewpoint and visual angle. Then different geometric representation can be used for rendering different details.

3.1. Principle of Visualization with LOD in 3D GIS

In general, data representation of LOD can be divided into two methods:

A.Different LOD corresponds to different data structure, objects in every LODs are stored respectively by levels. The implementation of this method is simply, but it causes huge data redundancy, large space consumption and weak data consistency.

B.Only data at the highest LOD are stored. When other LOD is selected, the data are generalized by these data stored at highest LOD. Though the data structure is simple, it is difficult to realize generalization. Because it is much time-consuming, fast access method must be investigated, as well as optimized generalized techniques and many other techniques.

By characteristics of 3D space, visualization with LOD in 3D GIS should be divided into two conditions. The two methods of data representation mentioned above should be used respectively for the following two conditions:

1)Visualization with LOD at multi-scales. As in 2D GIS, 3D GIS requires different data representation at different scale. Because at different scale, different geographic objects are represented, there are different LOD corresponding to different scales. Generalization techniques should be used for data representation of different LODs at different scale, in order to decrease amounts of data and avoid data redundancy. Generally method B mentioned above is used for generalization. In this paper, we will only take account of selection, which is the most used generalization technique.

2)Visualization with LOD at a certain scale. With the development of computer graphics and virtual reality techniques, more and more fast visualization of complex scene is needed. Details of each 3D geographic object should be divided into several levels. For ensuring real-time rendering with high quality and decrease time of computation, method A is often used for storing respectively all data of each 3D object at each LOD into database.

Database is absolutely necessary for syn-thesizing the two conditions together in 3D GIS. The database is also required with proper data structure and fast data access method, in order to ensure high-efficient querying and updating of data in database. Now object-oriented model has been applied in GIS successfully. Taking account of more complex types of object and relations between objects in 3D GIS, object-oriented database is our best approach for data management. Based on object-oriented database, we integrate gene-ralization with 3D visualization for representing LOD in 3D GIS at multi-scale.

For generalization, we mainly implement fun-ction of selection, which is defined as holding or deletion of geographic object. We can delete too small or less important objects as scale becomes smaller. If each geographic object is given an important value, by selecting of important value, we can judge if an object should be deleted or ignored at a certain scale. In database, only data at the biggest scale are stored. If we want to deal with the scene at other scale, only those geographic objects whose important value is greater than a predefined value can be selected. Generalization techniques should be combined with fast data access method to improve the performance of spatial query and spatial analysis. Spatial index should be built by rule of LOD.

Because rendering of detail for 3D visualization is very time-consuming, it is impossible to render all visual geographic objects in perfect detail in the 3D scene. So different LOD of each geographic object is rendered by position of viewpoint. For example, imagine that you are flying in a helicopter towards a city. When you see the city for the first time, your are quite far away. To visualize the city from the viewpoint, only outline of building and tree are sufficient. We can displace these buildings and scribed in perfect detail, and the surface of these objects should be covered with texture image. For some ordinary objects, it is sufficient to display only their shape in detail. Those far away from viewpoint are displayed with only their MBB. In the process of fly-over, with the constant move of viewpoint, details of different objects change correspondingly. Through proper algorithm, different LOD will be selected for visualization. Though all data of every objects at each LOD are stored into object-oriented database in advance, the amount of data needed to draw the entire scene will stay approximately stable, and objects near the viewpoint will contribute most of the data.

We judge LOD of geographic objects by position of viewpoint. The factors affecting LOD of geographic objects are as follows:

1) Distance from viewpoint. Closer to the viewpoint, more details are discernible.

2) Visual angle from viewpoint. The LOD decreases with increased lateral distance.

3).Size of objects. The volume of object is bigger, the number of visible pixels should increase.

Thus we can project the MBB of each objects into the viewport. Comparing the area of polygon formed by projection of geographic object with different threshold values predefined, we can make sure which LOD the object lies in. If object is of greater volume, closer to viewpoint, visual angle is smaller, the area projected in the visual zone is greater. Thus more details will be drawn.

3.2. The Data Structure of Object-Oriented Database

Based on our previous work on object-oriented GIS, we classify the architecture of database into four levels (Jun 1999). As shown in figure 1, Scene level, Subject level, Geoobject level and Geometry level describe respectively the entire scene, subject layer, geographic object and detailed geometry data. A 3D scene (Scene level) should be divided into several layers based on subjects to be measured. Each subject layer (Subject level) includes many geographic objects (Geoobject level), and their geometry data are stored in Geometry level. Among the four levels, Geoobject level represents detailed attributes of geographic objects and relation between geographic objects. Because Geoobject level is directly related to visualization with LOD, we will focus on the data structure of this level.

The structure of Geoobject level is shown in figure 2. Class Geoobject is the base class for

Geometry level

Geoobject level

Subject level

Scene level

Figure 1 The four levels of object-oriented 3D GIS

trees with their MBB . As you are slowly comenearer, some outstanding objects should be rendered in most detail. Their shapes should be de-

representing attribute information of geographic objects at a certain subject layer. By the classification of geographic objects, there are five subclass of class Geoobject: Point, Line, Surface, Body, Complex-obj. The five subclasses describe respectively point-alike object, line-alike object, surface-alike object, body-alike object and complex object.

In the base class Geoobject, attribute imp_value represents important value of each geographic object. By imp_value, we can judge what objects are selected at a certain scale.

imp_value = (int) f (type, size, subject layer,…) (1)

In the above function, the important value is related to type of object, size of object, property of subject layer and so on. In order to divide geographic objects into different levels by important value in index tree and improve the efficiency of data access, the important value should be rounded off. At the same time, to decrease the height of index tree, we should make sure a continuous integral field of value is guaranteed by level of scale. We map important value into the field of value.

Attribute mbb describes minimal axes-parallel bounding box of geographic object. Attribute texture represents filename of textural image file of geographic object. They are related to the three LOD for visualization of geographic object. Attribute shape represents geometric description of geographic object. Detailed geometric data are stored in Geometry level.

3.3. Data Access Method for LOD

Each geographic object is stored into object-oriented database as object. In order to improve the performance of access of these objects, proper data structure should be built for all objects in database based on imp_value, so that only selected objects at a certain scale are operated. Furthermore, as a GIS, spatial query and spatial analysis are inherent to its application. For example, there are requirements that we want to find all objects intersecting a zone at a certain scale. This requirement demands to build 3D spatial index (e.g. R tree) for all objects. Therefore, it is necessary to investigate spatial index based on important value, that is 3D spatial index for LOD.

The reactive tree based on R tree, which is proposed by Oosterom in 1991, is a successful spatial index for LOD (Oosterom 1993). On the basis of the reactive tree, many generalization techniques (e.g. selection, simplification, symbolization) can be implemented high-efficiently. The reactive tree is applied successfully in 2D GIS. However in 3D GIS, the index tree based on R tree will encounter some problems, most of which are caused by greatly increased overlay among MBB of geographic objects in 3D space. These inherent problems will decrease the performance of index.

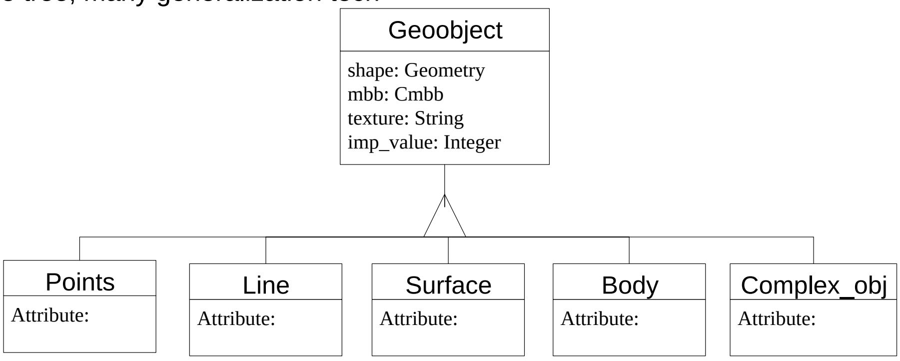

Figure2 The Structure of Geoobject Level

Based on achievements in high dimensional space made by some index method such as R* tree (Beckmann 1990) and X tree (Berchtold 1996), we extend the reactive tree to 3D space. Specifically we modify the searching algorithm of reactive tree, and make it adapt to 3D space.

Because an added axis z is considered in 3D space, overlaps among MBB of spatial objects increase greatly. However, in the algorithm of choosing subtree of the reactive tree, only method of least volume enlargement is adopted. Those subnode, whose MBB needs least volume enlargement to include the new body, are selected. The single method can’t avoid the effect brought by increased overlap. So we should modify the algorithm of choosing subtree to improve the performance of index.

Thus we introduce the method of least overlap enlargement in algorithm, which has been used successfully in R* tree. Those subnode, whose MBB needs least overlap enlargement to include the new body, are selected. Let $E_1, \ldots, E_p$ be the entries in the current node. We define overlap of entry E_k as follows:

But if only method of least overlap enlargement is adopted, the performance of index is also affected. Furthermore, equation above shows that

$$\text{overlap}(E_k) = \sum_{i=1, i \neq k}^{p} \text{volumn}(E_k.\text{MBR} \cap E_i.\text{MBR})$$

computation of overlap among objects is very time-co nsuming. Therefore, we should combine the two methods. Method of least overlap enlargement is used at the most desired node. At other nodes, we will still use method of least volume enlargement to choose subnode.

In most 3D GIS applications, we can find out that for those directory nodes lying in the upper level of index tree, there are few discrimination at axis z. However, for those object nodes lying in the lowest level of index tree, the discrimination at z axe is very important. For example, in 3D digital city, many upper MBBs containing many MBBs of objects have almost the same height. Whereas those geographic objects are perhaps doors and windows, and their position at axis z is apparently greatly different. Because of the increased overlay at axis z, the geographic objects should be considered first for decreasing overlay. So we draw a conclusion: There is the greatest desirability of least overlay for the object node. In the process of choosing subnode, we should use method of least overlay enlargement for those nodes pointing to object node, whereas method of least volume enlargement for other nodes. At the same time, we should know the truth that for very distant box the probability to yield the minimum overlap is very small. For reducing the CPU cost for computing least overlap, we can use method of least volume enlargement for filtering some subnode. Then least overlap is computed for these filtered subnodes.

Modified algorithm for choosing subnode is shown as follows:

1. Set N to the root.
2. If (imp_value of N = imp_value of new box)

    Return N

    Else

    {

    If (imp_value of subnode of N = imp_value of new box)

    {

    Sort the MBB in N in increasing order of their volume enlargement needed to include the new data box.

    Let A be the group of the first p entries.

    From the entries in A, considering

all entries in N, choosing the entry whose MBB needs least overlap enlargement.
Else
Choosing the entry whose MBB needs least volume enlargement.
}
3. Set N to be the child node pointed to by the child pointer of the chosen entry and repeat from step 2
}

3.4. Implementation of Visualization with LOD

Based on object-oriented database management system (DBMS), by using library of 3D graphics for interface of visualization, we implemented an experiment system for visualization with LOD. In the experiment system, all geographic data are stored into object-oriented database. The data model is organized by four levels mentioned above, which represent respectively scene, subject layer, geographic object and geometric data. The entire scene is divided into three scales. At each scale, some 3D objects are neglected by imp_value. Once the scale is fixed, selected objects will be rendered in 3D scene, and some operations of query are provided with. In the process of visualization, three kinds of different details of each visible 3D object are rendered with move of viewpoint.

In the experiment system, we focus on several problems as follows:

1) Selection of DBMS. Most 2D GIS are based on relational database. But there exists many unsolved problems in relational database applied in GIS: no user-defined data type; new index method can't be created by user; E-R Model can't meet requirement of GIS, weak extensibility, and etc. For 3D GIS, because of huge quantities of data, new data type and complex spatial relation, object-oriented database should be used. We make use of object-oriented DBMS Jasmine developed newly by CA for commercial use. Though now object-oriented DBMSs are of low efficiency, but we still use it
by taking into consideration developing trend of object-oriented database and easier modeling.

2)Selection of interface of visualization. Comparing several general interface of visualization: OpenGL, IRIS Performer, OpenGL Optimizer and Open Inventor, we select Open Inventor as interface of visualization in our experiment system (Wernecke 1993). Open Inventor is a graphic library based on OpenGL. Because it provides with perfect interactive interface and is integrated together with C++, it is perfectly suite for developing 3D visualization system.

3) Division of LOD of 3D objects. In database. Because all data at every LODs needs to be stored together, it will decrease the time consumed for rendering, but contrarily add storage space inevitably. So it is difficult to create and maintain database. Thus taking synthetically into consideration difficulties of creating database and actual requirement of high-quality scene, we divide LOD into three levels: The first level only displays MBB of geographic objects, and represents approximate shape and size of object. The second level displays detailed geometric shape of geographic object. The third level renders texture in the surface of geographic object as well as detailed geometric shape, so it is the most detailed level. With moving of viewpoint, selected visible geographic objects judge their LOD by comparing ratio, by which viewport is occupied by projected polygon, with two threshold values.

4. Conclusion

Through analyzing requirement of visualization with LOD at multi-scale in 3D GIS, this paper divides representation of LOD into two conditions: 1) generalization at multi-scale; 2) representation of different details of geographic objects at a certain scale. After integrating representation of LOD at the two conditions together in object-oriented database with proper access method for LOD adapting to 3D space, visualization with LOD at multi-scale in 3D GIS can be implemented successfully.

Based on this paper, our future work will focus on the following aspects:

1)By characteristics of 3D space, analysis of performance for our modified index method should be done;

2) Representation of LOD should be studied when scale changes smoothly;

3) The efficiency of access for database should be improved when more LOD data of 3D objects need be stored in advance;

4) How to visualize DTM data with LOD by object-oriented database techniques.

References

Beckmann N., Kriegel H.P., Schneider R., Seeger B., 1990, The R*-tree: An Efficient and Robust Access Method for Points and Rectangels, Proc. ACM SIGMOD Int. Conf. On Management of Data, Atlantic City, NJ, 1990, pp. 322-331.

Berchtold S., Keim D.A., Kriegel H.P., 1996, The X-tree: An Index Structure for High-Dimensional Data, Proceedings of the 22nd VLDB Conference.

Clark J.H., 1976, Hierarchical Geometric Models for Visible Surface Algorithm, Communications of the ACM, No. 10, pp. 547-554.

Jun Li, Ning Jing, Maoyin Sun, 1999, An Object-Oriented Data Model Integrating Field and Object, Proceedings of Geoinformatics'99 Conference and the International Symposium on the Geoinformatics and Socioinformatics, Michigan.

Kofler M., Gervauta M., Gruber M., 1998, The Styria Flyover – LOD management for huge textured DTM models, Proceedings of Computer Graphics International 98, Hannover.

Kofler M., Rehatschek H., Gruber M., 1996, A Database for a 3D GIS for Urban Environments Supporting Photo-Realistic Visualization, International Archives of Photogrammetry and Remote Sensing, Vol. XXXI, Part B2, Commission III, pp. 198-202, Vienna.

Koninger A. Bartel S., 1997, 3D-GIS For Urban Planning – Object Hierarchy, Methods And Interactivity, Proc. of JEC-GI, Vienna.

Maciel P.W.C., Shirley P., 1995, Visual Navigation of Large Environments Using Textured Clusters. ACM Siggraph Symposium on Interactive 3D Graphics, pp. 95-102.

Oosterom P.V., 1993, Reactive Data Structures for Geographic Information Systems, , Published by Oxford University Press.

Wernecke J., 1993, The Inventor Mentor, Published by Addison-Wesley.