Study on Visualization of Virtual City Model Based on Internet

JIN Baoxuan BIAN Fuling ZUO Xiaoqing WANG Fangxiong

ABSTRACT With the rapid development of computer graphics, distributed-computing and Internet, it is possible to achieve Internet-based virtual city. This paper dwells on the method of the terrain and its feature modeling and complex entity modeling in the virtual city. Then, discusses the method for Internet-based virtual city 3D visualization and the design of the Browser/Server architecture of the system of virtual city in the network environment. Finally, Java and Java 3D are used to show an experiment example, and the related conclusion about Internet-based virtual city 3D displaying and the client-side interactive operation is given.

KEYWORDS distributed-computing; virtual city modeling; 3D visualization; Java 3D

CLC NUMBER P208

Introduction

The rapid development of network technique makes a great development of the applications domain of GIS in recent years. The service of GIS tends to network step by step. In the past, the GIS applications through the network were just limited to 2D forms, the 3D Visualization of city was also just limited to a stand-alone computer or the inner of small LAN, and the requests to 3D spatial information was not so many. With the rapid development of the computer technique, graphics technique, virtual reality technique and Internet, it is possible to achieve Internet-based virtual city\(^1\). At present, the 3D visualization has been widely applied to those domains including city planning, architectural design, geologic prospecting, weather forecast, environment monitoring, and mobile communication and so on. After the concept “Digital Earth” was brought forward, cyber city and cyber block have become a development strategy of developed country. Obviously, the applications of virtual city in the Internet is more expansive.

Because of the limitation of network bandwidth, graphics technique and virtual reality technique, achieving virtual city in the Internet is still in a study phase.

This paper begins with discussing the method for geometry modeling of virtual city in the Internet, while mainly discusses how to model large scale terrain and key features. Then this paper dwells on the method for Internet-based virtual city visualization and gives an experiment example to illustrate the key-technique of Internet-based virtual city applications.

1 Method for virtual city modeling

In virtual city, the terrain is complex and the feature is various. The model classification mainly includes two kinds, namely, terrain model and feature model. Feature model mainly takes account of models of those key features such as building and road. As far as modeling technique, it mainly includes geometry configuration modeling and texture mapping model-
ing\textsuperscript{[2]}. For it is applied to the Internet environment, when we design a data model, we must consider the aspects such as the data transmission capacity in the network, multi-resolution expression, the 3D modeling data format of open standard etc.

1.1 3D modeling of terrain

Terrain is a foremost geographic object and a base of 3D space in virtual city. With respect to data structure, terrain can be divided into GRID, TIN and GRID-TIN. In factual applications, GRID and TIN are usually adopted, and each of them has its advantages and shortcomings. The modeling based on GRID is often used to deal with overall data of flat region, but is not suitable for fragmentized terrains and cragged regions. On the contrary, TIN is more suitable for those regions whose surface changes sharply, and it has variable resolution. However, the data storage and operating of TIN are more complicated than that of GRID\textsuperscript{[3]}.

Considering the fact of multi-resolution expression and partition of large scale terrain and the data transmission in network environment, this paper adopts the GRID model.

1.1.1 Terrain partition and spatial index

Since the area of a city is large and its terrain is complicated, considering computer’s performance and real-time data transmission in network environment, we must divide the whole city into some blocks and make a distributed-storing. The city is divided into $m \times n$ blocks (Fig. 1). Each block must be made into many multi-resolution GRID models respectively\textsuperscript{[4]}.

In 3D city displaying and walkthrough, the view scope of loading data is very finite. In order to load a given block quickly and dynamically, we must create spatial index to all blocks. There are many kinds of 3D search data structures, such as R-trees, quad trees, etc.

1.1.2 Simplifying technique of terrain block

At present, LOD technique has been applied mainly to terrain simplification. There are various kinds of algorithms for LOD, but the representative algorithms may be summed up as: real-time continuous LOD\textsuperscript{[5]}, Hoppe’s progressive meshes\textsuperscript{[6]}, Duachaineau\textsuperscript{[7]} and others’ ROAM. In the real-time continuous LOD algorithm, the authors adopted a quadtree structure to describe the terrain fragment, set up terrain height field recursively and set up node estimating function through screen error qualification. This algorithm is simple and effective. In Hoppe’s progressive meshes, scene details are expressed by adding triangles to grid and expanded this algorithm to form view-dependent progressive mesh. In the expanded algorithm, the authors divided terrain into equal terrain blocks, implemented PM rule in single block, but did not simplify triangles between blocks. Therefore, this method affects the efficiency of simplifying. In ROAM algorithm, Duachaineau adopted bintree to organize terrain, utilized combination queue and partition queue to complete the combination and partition operation of grid, set up the LOD model of terrain recursively. Though the algorithm is complicated, its efficiency is high.

The virtual terrain scene experiment in this paper adopts a real-time LOD algorithm based on binary tree and view-dependent. The algorithm is based on the viewpoint taking account of terrain features at the same time, which can decrease the number of triangles greatly and reduce the data transmission capacity in the network. Fig. 2 shows a LOD model generated by this algorithm.

1.2 3D modeling of building

Building, an indispensable geographic object in virtual city, is the most key feature. Its model-
ing efficiency will directly affect 3D displaying and interactive operation. No matter how complicated a building is, it must be made of some simple objects, so layer-combined model for building was used. Firstly, each storey of the building was split into roof and wall, then the roof and the wall are triangulated, respectively, finally they are divided into triangle structures easy to be stored. Fig. 3 shows the structure of triangle partition of building.

1.3 3D modeling of irregular features obtained

Irregular cultures, such as road, water area, have clear edge features. We can look them as a closed region. We adopt TIN to triangulate the interior of region, and get a constraint triangle network model. Those complicated cultures, such as road, can be obtained in the multilevel 3D road model on the basis of the degree of relation. At the same time, by the 3D road model a multi-resolution LOD model can be build up. In the whole 3D city model, triangular network, which is made from irregular features (road, water area), is integrated into a terrain model and makes up of TIN + GRID mixed structure model.

1.4 3D modeling of complicated 3D entity

In virtual city, complicated point features, such as wires-pole, streetlamp, tree, barrier, etc., are very hard to model by conventional means. We need other commercial software, such as 3DMAX, AUTOCAD etc. Firstly, we use their modeling tools to make 3D models, then make data conversion via standard external data exchanging format (.3ds, .dxf). The only work is to access those files and add 3D scene.

2 Visualization of virtual city model based on Internet

2.1 Framework of virtual city on Internet

The mainly function of Internet-based virtual city is to achieve 3D city information comprehension published on network, to provide users with a real-time, dynamic, interactive, virtual city environment, and to make itself reach a real information intercommunicating and data sharing. This system adopted B/S multilayer ar-
chitecture, namely, client-side, WEB server, application server and database server.

Client-side is an interface that provides information displaying and data operating for users and realizes remote users' interaction. In virtual city, client-side is mainly for displaying 3D scene, performing scene dynamic walkthrough, querying spatial and attribute information and analyzing spatial information simply. Its expression form will be Java Applet embed Web pages or Application running on the client-side.

Web Server provides WWW service and Web-based application services, such as log managing, session managing, server-side status managing etc. There are some implementation methods. Taking account of the compatibility of Java3D technique of client-side, the authors mainly adopt JSP + Servlets as server-side-based development technique.

Application Server, an enterprise-based application, is usually installed in those containers required by enterprise-based application, and run in the background as a server. It is mainly composed of Servlet engine and EJB component fulfilling key functions such as spatial data storage, 3D modeling and visualization, spatial query and spatial analysis in virtual city.

Database server provides the store of spatial data which contains vector, DEM, image data. Fig. 4 shows the framework of a virtual city system.

![Framework of virtual city system](image)

### 2.2 3D dynamic displaying technique of client-side based on Java3D

In virtual reality modeling, those languages, such as OpenGL, VRML, Direct3D, are adopted frequently. Since OpenGL has the advantage of across-platform, it has been widely used to develop 3D applications. However, OpenGL API functions are too much and too complicated. Because VRML has the characteristic of easy program mining and convenient network transferring, it has been more widely used in the network. But the function of VRML is not very powerful. Being a Microsoft 3D graphical program API, Direct3D is complicated, which is often used for the development of 3D games. Direct3D requires developers to have high C++ basis, so it is hard to be applied universally. Java3D, a SUN 3D graphical program API, provides developers with creating, control, render of 3D entity, and makes the developers work relatively simply. Because of the across-platform features of Java, codes are designed only once but can run everywhere. At the same time, Java3D can be integrated with Internet easily. Therefore, Java3D has been extensively used in the network. Java3D API is an interface that is defined by SUN and adopted to perform 3D displaying. 3D technique is a lower-level displaying technique and Java3D provides a upper-level Java-based interface, which packs OpenGL and DirectX into Java interface. The novel design makes 3D technique more simple and can be put into J2SE, J2EE architecture. All these features assure the strong expendability of Java3D technique.
For the above reasons, considering that server-side adopts SUN's JSP and Servlet technique, the authors adopt Java3D technique to accomplish 3D display of client-side. Data structure of Java3D is the same as that of OpenGL that can adopt data structure of scene map, namely, tree-structure is composed of some directive and asymmetric graphics. It can be shown in Reference [11].

Java3D scene includes entire description of the whole scene. The lowest-layer node is virtual world. There is only one node in each scene map. Each position node has one or more sub-map nodes. In order to form 3D application condition, we must create required shapes, present external appearance and geometry information of those shapes, and put them on the appropriate position. These shapes and their positions are created on sub-nodes. Besides, after the 3D scene is placed, we set a specific observation place and parameter (i.e. line of sign and angle of view). After finishing the above-mentioned job, the formation of 3D scene has been created. For the virtual city based on network, we use the same method to arrange it. Topmost-layer is virtual city module, and below it there are two embranchments, namely, terrain model and view model. Other surface feature models are under the terrain model. The scene structure can be seen from Fig. 5.

2.3 Data management tactics of Server-side based on EJBs

Server-side mainly includes three parts, namely, Web Server, Application Server and Database Server. They can be distributed on the same Server or many Servers. The description of

the function architecture of Server-side can be seen from Fig. 6.

![Diagram](attachment:image.png)

Fig. 6 Framework of server functionality

Because of the numerous data type and large data set of virtual city, to accomplish 3D model displaying and dynamic interactive operating in the Internet environment, we must have a set of reasonable management methods and access mechanism of data. In this paper, Server-side mainly adopts SUN J2EE system frame to construct the application of virtual city. J2EE is the standard intermediate component architecture, whose purpose is to simplify and specify the development and arrangement of distributed enterprise application system. The hard core of enterprise computing accomplished in the Server-side, EJB is a Server-side component that accomplishes enterprise computing in J2EE. It is obvious that adopting EJB technique to implement integrative data management tactics of Server-side can greatly improve the running efficiency, transportability, security and reusability of system. This technique uses standard Web container and EJB container to achieve component registering, seeking, calling and destroying. The authors achieve the following work for virtual city via the EJB component designed Server-side: spatial data (vector data, DEM data, image data) managing, 3D city model constructing, spatial references establishing, data compressing, database querying and spatial analysis etc. EJB components include session and entity components. The main component and function of designing can be seen in Fig. 7.
3 Experiment

The authors have developed a virtual city application system based on Browser/Server architecture, then use Java (jdk1.4.1) and Java3D (Java3D 1.3 for OpenGL) to realize the visualization of virtual city model in Client-side. Next, the authors adopt Tomcat4.1 as Web Server and JBoss3.2 as Application Server (EJB containers) to achieve the management of background data and the construction of 3D model. This system realizes Web-embedded 3D city browsing, Client-side random walkthrough, building attribute querying and 3D model constructing and so forth. Fig. 8 shows the 3D scene of a virtual block, including fifty-six buildings and involving the texture of one hundred different surfaces. Fig. 9 shows the 3D scene of a virtual terrain. We mainly used GRID to construct terrain model and realize those functions such as 3D coordinate querying, spatial distance measuring etc. And the terrain model construction and simplifying is accomplished in Server-side.

4 Conclusions

3D city modeling is the core of the virtual city system, it is very important that we use a scientific, reasonable, efficient method for modeling in the virtual city applications. This paper mainly discusses the method of terrain and key features modeling in detail. Considering the modeling efficiency and the data transmission capacity, we use continuous multi-resolution LOD model based on regular grid to express terrain and build different descriptions of detail, multi-level models and spatial reference for key features. The data managing tactics of EJB components based on J2EE system architecture are used for Server-side spatial data managing, and the integrative managing of multidimensional spatial data is realized. Last, the authors use Java and Java3D to develop the virtual city application experimental system based on the Browser/Server multi-level architecture.

There are various features, complicated terrain
and large data sets in virtual city. Because the field involving virtual city is very extensive and complicated, many theories and techniques need to be studied and solved.

REFERENCES
1 Tao V, Wang Q K (2002) GIServices-based 3D Internet GIS:GeoEye 3D. Acta Geodaetica Cartographica Sinica, 31(1):17-21 (in Chinese)
2 Wan G, Cheng G, You X (2002) Research on geometry modeling of terrain features in virtual city. Acta Geodaetica Cartographica Sinica, 31(1):60-65 (in Chinese)
3 Li Z L, Zhu Q (2000) Digital elevation model. Wuhan, Wuhan University Press. (in Chinese)
4 Gong J H, Lin H (2001) Virtual geographic environments—a geographic perspective on online virtual reality. Beijing: High Education Press. (in Chinese)
5 Lindstrom P, Koller D (1996) Real-time, continuous level of detail rendering of height fields. Siggraph, (8):109-118
6 Hoppe H (1997) View-dependent refinement of progressive meshes. Siggraph, (8):189-198
7 Duchaineau M (1997) Roaming terrain; real-time, optimally adapting meshes. The Conference on Visualization, (10);81-88
8 Yang B S, Li Q Q, Mei B Y (2000) Study on the visualization of 3D city model. Acta Geodaetica Cartographica Sinica, 29(2):149-154 (in Chinese)
9 Zhu Q, Li X F, Zhang Y T (2002) The design and implementation of web-based CCGIS browser plug-in. Acta Geodaetica Cartographica Sinica, 31(1):22-27 (in Chinese)
10 Cheng J Y, Zhou L S (2002) Java3D-based virtual reality modeling method. Application Research of Computers, (5):33-35 (in Chinese)
11 Dou Z H, Liu P (2002) Java3D programming practice—the 3D carton on internet. Beijing: Tsinghua University Press, 1-16. (in Chinese)

Notes to Contributors
Contributions are welcomed on one of the following subjects or in related areas:
- GIS
- GPS
- RS
- Cartology
- Geodynamic
- Geo-surveying
- Photogrammetry
- Graphics
- Physical geo-surveying
- Engineering surveying
- Mapping apparatus
- Photography

Paper submitted in the electronic text on a diskette should be sent along with two printed copies. The main text should be preceded by the abstract of no more than 300 words, followed by key words. Full references should be listed in the order of the citations in the text under the heading "References", guided by standard publication format. The name of the fund and project series number for articles of funded projects should also be given.

The authors assume sole responsibility for their dissertations. Canned theses are not allowed. Once the paper is submitted, the editors are authorized to make necessary literate processing. Authors who receive no notice in three months after submission may contact us for inquiry.