MODELING AND VISUALIZING 3D URBAN ENVIRONMENT VIA INTERNET FOR URBAN PLANNING AND MONITORING

Guoqing Zhou[a], Zhenyu Tan[a], Penggen Cheng[b], and Weirong. Chen[a]

[a]Old Dominion University, Kaufman Hall, Rm. 214, Norfolk, VA 23529, USA-gzhou@odu.edu
[b]Department of Surveying Engineering, East China Institute of Technology, Fuzhou, Jianxi 344000

ISPRS Commission II/6

KEYWORDS: Three-dimensional, GIS, city, modeling, visualization, internet/web

ABSTRACT:

Geographical Information System (GIS) technology has been an important tool in supporting urban planning and management for efficient accessibility to more, better, and timely information. This paper presents our research results of visualizing 3D city model in a web explorer environment under the supports of digital building model (DBM), digital terrain model (DTM), large-scale true urban orthoimage, and actual building sidewall pictures. A system has initially been designed and developed under a client/server architecture, which integrates Internet GIS with multi-tier web application for creation of a 3D city model. The front-end client provides a user interface for setting properties of the 3D model, while the backend server processes the parameters and generates the 3D model, which is then sent back to the VRML-enabled WWW browser for display and navigation. The technology for display and navigation of the 3D model is actually a Java JSP/Servlet technology built under multi-tier web architecture. The data accessing and processing of the technology is through Java programming language to link the displayer with the database where the required GIS spatial data are stored. The results from our experiment demonstrated that this developed architecture is able to display and visualize 3D city data in web environment for public access. Using the sidewall pictures taken from real city buildings, we refill the white walls of urban buildings into a real appearance as seen in our real world city. The player provides several click buttons to change the effect and position of visualization, including rotation, zoom, pan, seek etc. The users can easily to view details of specific building, the entire city in a moving 3 dimensions, to locate a specific building, and so on.

1. INTRODUCTION

The urban GIS plays an important role in city planning, communication system design, micro-climate control and simulation, tourism, etc. The demand for visualization of urban model and realistic presentation of the real world has been increased dramatically today. Internet and World Wide Web (WWW) have been widely applied in delivering, processing, and publishing geospatial data. Web-based GIS, which permits users to access, retrieve, display, and analyze geospatial data over the web, has got a wide variety of applications. In the recent years, web-based three-dimensional (3D) GIS for visualizing geospatial data has attracted many researchers. The operation uses the web as a delivery mechanism, deals with the transformation of multi-dimensional geospatial data, information, and knowledge into an effective 3D visual form (Lin *et al.,* 1999; Belfore *et al*., 2003; 2002; 2001; Blaser et al., 2000; Kreuseler, 2000). MacEachren (1997) used VRML 2.0 to model geospatial data, and Java to develop an interface to interact with the VRML world. Hibbard (1998) designed and developed the VisAD system, which enables many users to implement the visualization of a shared set of numerical data and computation sources. Lin *et al.* (1999) and Huang *et al.* (1999) developed a called GeoVR system, which is a client/server architecture to enable the interactive creation of a 3D scene and VRML model from 2D spatial data by integrating internet GIS and HTML. Rohrer and Swing (1997) suggested the use of traditional approaches for visualization because of web-based inherent limitation in, such as, the narrow channel of the Internet. Verbree et al. (1999) presented an interaction in virtual world views between 3D GIS and virtual reality (VR).

This paper presents our research results on development of a web-based urban 3D visualization. Using downtown Denver CO as protocol, the orthoimage is employed as a bottom layer, and objects are superimposed. The system has been designed and implemented under a client/server architecture, which integrates Internet GIS with multi-tier web application for creation of a 3D city model. The front-end client provides a user interface for setting properties of the 3D model, and the back-end server processes the parameters and generates a 3D city model, which is then sent back to the VRML-enabled WWW browser for display and navigation. The display and navigation of the 3D city model is implemented using a Java JSP/Servlet technology built under a multi-tier web architecture. The data processing is implemented through Java programming language, and the data accessing is carried out via linking the displayer to the database, in which the required GIS spatial data are stored.

2. ARCHITECTURE OF WEB-BASED URBAN 3D GIS

The architecture of web-based 3D city visualization is designed in Figure 1. The architecture consists of three-tiered modules. The first tier is to display layers that includes both the web browser and the web server, which is responsible for assembling the data into a presentable format. The second tier is the application layer. It consists of scripts and programs. The third tier provides the second tier with the data needed. The architecture operates as follows:

(1) Collect data from the user (the first tier),

(2) Send a request to the web server,
(3) Run the requested server program (the second and third tiers),
(4) Package up the data to be presented in the web browser, and
(5) Send it back to the browser for displaying (the first tier).

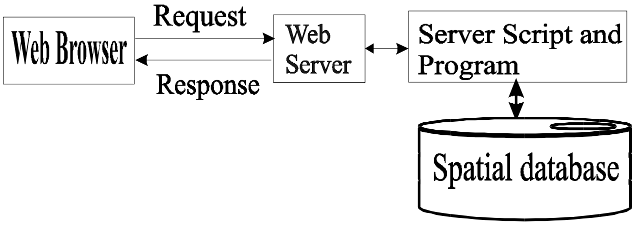


Fig. 1. Three-tier client/server architecture for web-based 3D city data visualization

Virtual Reality Modeling Language (VRML) has become a popular tool for building a 3D representation on client side. A VRML application is typically a hierarchical organization of geometries, sensors, light sources, script methods, and grouping primitives assembled in a meaningful fashion (Web3D Inc., 1998; Brutzman, 1998; Nadeau, 1999). There are various primitives, termed *nodes*, are available in VRML. These nodes are capable of directly visualizing a complicated scene, thus it is very useful for visualizing urban 3D models. The geometry of the primitives has been defined with shape nodes in VRML. The sensors are used to associate interactive capabilities. The Scripts enable users to implement many specific functions. Information is passed between nodes by events whose routes are either explicitly declared or established within Script nodes (Belfore *et al.*, 2002). VRML has many desirable features to meet users' needs for describing 3D objects on the WWW for GIS applications (Reddy *et al.*, 1999). A VRML model can be viewed by any web browser that has a VRML plug-in pre-installed. VRML "worlds" can depict realistic or otherworldly places, which can contain objects that link to other documents or VRML worlds on the web.

Java web technology is a powerful web development toolkit. Java technology is further improved by supporting servlet technology. Servlets provide the capabilities of cgi-scripts, and are superior in execution efficiency, stability and security. Through the Java Development Kit (JDK) API, servlets can connect to server databases.

Servlets are software methods that run on the server and can be called up as URLs from the Internet (Belfore et al., 2002). Servlets handle the client requests and dispatch them to the server. With servlets as a middleware, it can build a multi-tier web application. A servlet extends the capabilities of servers that host applications accessed via a request-response programming model. Servlets can respond to any type of requests, and are commonly used to extend the applications hosted by Web servers. JSP (JavaServer Pages) technology acts as a top layer above Java servlets on the servlet engine. It allows developers to combine HTML codes with server-side scripting to create dynamic pages.

2.1 Client Side Architecture

The basic architecture of client side is depicted in Figure 2. This application uses JSP pages to provide a front-end interface from client side. The users interact with the server for their choices via an interface. The requests from users are directed to the server through the Servlet middleware, and the server runs on the Web server.

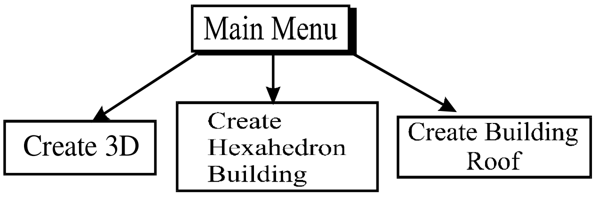


Fig. 2. Three-menu option on client side front end

The front end is comprised of three options. The first choice is for creating 3D VRML model directly from the data stored in the database. The server part of this application will access the data files, automatically load the data into the server, and then automatically build a VRML model based on data files. After that, the server sends the VRML model back to the client end for displaying it on the browser. The second choice is for creating single Hexahedron building through point coordinate data input. This part allows users to build a new building or add a building to an existing model when the users input the coordinates of the corners of a building. The third choice is for creating roof objects through parameter input. It enables users to add a roof object to an existing building (VRML model). The user first chooses what kind of roof he or she is interested in, and then inputs corresponding data and parameters. For example, when a user inputs the file name with the VRML extension and parameters describing building roofs, the roof object will be automatically created and added to the building (VRML model) that the user has specified via this file name.

2.2 Server Side Architecture

Our server side architecture is illustrated in Figure 3. The server side would provide actual functions, such as, read/write files access on the server, create VRML models, and generate content dynamically. Thus, the web server would serve all client requests for generating either static or dynamic contents. Several software technologies and development packages have been employed in this application. These include the Java^{TM} Development Kit JDK version 1.4, the Java Servlet Development Kit JSDK version 2.2, the Apache Web Server v 4.0.0, and the Cosmo Player version 2.1.1 browser plug-in. The application uses Java object-oriented design, which will connect and access the database files, process the user input data, and generate a VRML model.

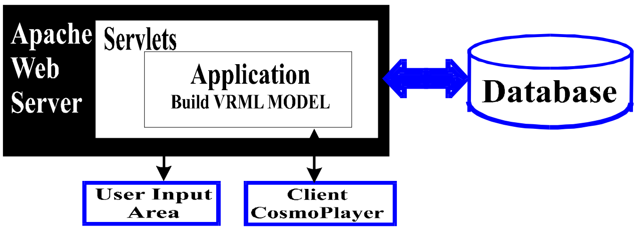


Fig. 3. Server Side Architecture

3. URBAN 3D MODEL

3.1 Urban 3D Data Structure and Model

Visualization of 3D urban requires a better 3D urban model. Different applications may have different data types and manipulation functions. The geometrical information to be operated to any targeted urban, generally includes two types of data: vector data and raster data. An appropriate data model should not only represents the geometrical information, e.g., shapes, lengths, areas, etc. but also implicitly or explicitly describe the topological relationship between geometrical objects, such as adjacency relations, link relations, positional relations (Wang *et al.*, 1998; Zhou et al., 2000; Zlatanova, 2000). For our model, the urban object is understood as three types of data sets (Figure 4):

(1) Digital terrain model (DTM),

(2) Original images and orthoimages, in our data structure, texture images are taken as attributes, they are stored in an independent database.

(3) Spatial objects, such as buildings, roads, waterways.

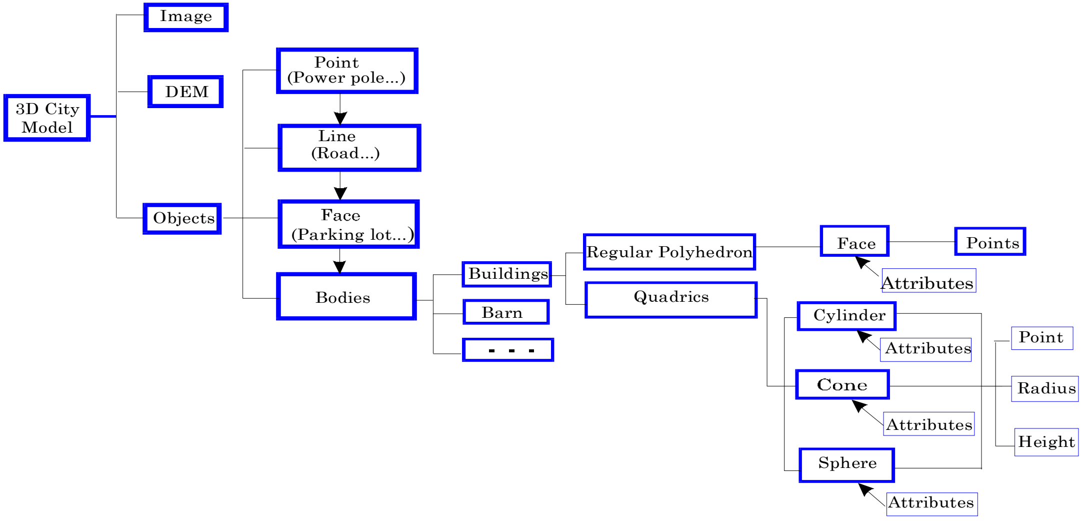


Fig. 4. A 3D urban concept model

The most important spatial objects in urban areas are buildings. There are four different geometric types of objects (Wang *et al.*, 1998).

(1) *Point objects:* which are zero-dimensional objects that have a position but no spatial extension, e.g., power poles, wells, etc.;

(2) *Line objects:* which are one-dimensional objects that made by connecting two points, e.g., power lines, telephone lines, etc.;

(3) *Face objects:* which are two-dimensional objects with area and perimeter characteristics, such as parking lots, grass fields, etc.; and

(4) *Body objects:* which are three-dimensional objects, such as buildings, barns, etc.

3.2 Implementation in a Relational Database

The data structure and model are implemented by a relational database technology (Figure 5). Each type of object, shown above, is defined as a table. The 2D tables are stored in a form of rows and columns, in which each row is identified by a key value. Each row stores the information of one instance of an entity, while each column describes an attribute of the entity. For example, a building table includes Building IDs, Roof IDs, Wall IDs, and attributes (see Figure 5). The Building ID is an identification code for a building object. A building is made from a combination of a roof and a wall. The Roof ID points to a Roof table. A roof table has two terms, Component ID and Component Type. It means that a building roof is made from a Component, whose type is either a regular Polyhedron or a Quadric. A Quadric is one of the three primitives: a Cylinder, a Cone and a Sphere. These are three basic types of Quadrics in the VRML model. A regular polyhedron is made from Face objects. This can be seen in the Polyhedron table, with a Face ID term, which points to a Face table. A face table has two terms; they are Texture ID and Point ID. Similarly, they point to Texture tables and Point tables. Just like Roof objects, a Wall object is also made from a Component. Its table relation is just the same as that of the Roof object.

4. IMPLEMENTATION THROUGH REAL DATA

4.1 Data Sets

- **Aerial Imagery**

The test area is located in downtown Denver, Colorado. In this experimental field, six aerial images were collected on April 17, 2000 using RC 30 aerial camera with a focal length of 153.022 mm at a flying height of 1650 m above the ground area. The six aerial photographs are formatted along two flight strips. The aerial photographs were originally recorded in film and later scanned into digital form at a pixel resolution of 25 µm. The endlap of the images is about 65% along strip, and sidelap is about 30%. Figure 6 shows one of images, DV1119, whose center is located in the downtown area where numerous tall buildings are situated.

The six original images, in combination with DSM, and exterior orientation parameters, are used for generation of urban TRUE orthoimage. A detailed description for urban TRUE orthoimage generation can be found in Zhou et al. (2003). The generated orthoimage is used for base map, on which the buildings (VRML model) to be built will
be superimposed.

- **Digital Surface Model (DSM)**

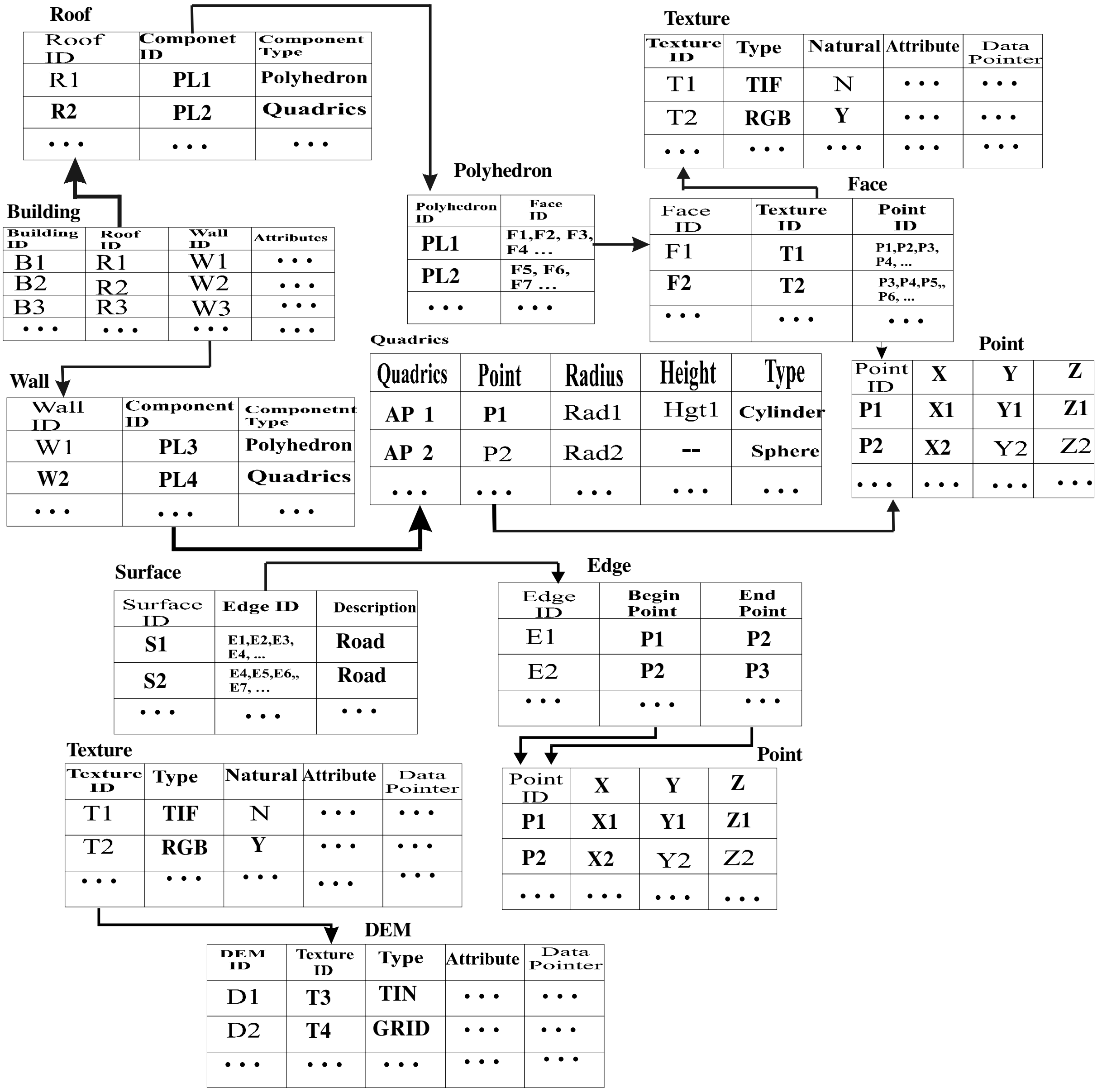
The digital surface model (DSM) available in this study area covers Downtown, Denver, CO, which is the central part of the downtown area where a number of high buildings exist. We choose a small part of the imaging area with relatively better DSM for our experiment.

- **Digital Building Data (DBD)**

The 3D coordinates of each building were measured from DSM using EARDAS/IMAGINE v. 6.0. The data were complied into vector data form and stored a relational database as described above.

- **Texture Image Data (TID)**

Over 200 texture images were captured by Snoy DSC-V1 Cyber-shot® digital camera. The different texture images from real world walls represent different texture patterns, different construction materials (e.g., bricks, glass, vynal), different ages of walls, different building styles, e.g., American, European, Chinese, Mexican, etc. All of these images were compressed into JPEG with a size of 640 x 480 pixel^2, and were stored in a texture image database.



Roof

Roof ID	Componet ID	Component Type
R1	PL1	Polyhedron
R2	PL2	Quadrics
• • •	• • •	• • •

Texture

Texture ID	Type	Natural	Attribute	Data Pointer
T1	TIF	N	• • •	• • •
T2	RGB	Y	• • •	• • •
• • •	• • •	• • •	• • •	• • •

Polyhedron

Polyhedron ID	Face ID
PL1	F1,F2, F3, F4 ...
PL2	F5, F6, F7 ...
• • •	• • •

Building

Building ID	Roof ID	Wall ID	Attributes
B1	R1	W1	• • •
B2	R2	W2	• • •
B3	R3	W3	• • •
• • •	• • •	• • •	• • •

Face

Face ID	Texture ID	Point ID
F1	T1	P1,P2,P3, P4, ...
F2	T2	P3,P4,P5,, P6, ...
• • •	• • •	• • •

Quadrics

Quadrics	Point	Radius	Height	Type
AP 1	P1	Rad1	Hgt1	Cylinder
AP 2	P2	Rad2	--	Sphere
• • •	• • •	• • •	• • •	• • •

Point

Point ID	X	Y	Z
P1	X1	Y1	Z1
P2	X2	Y2	Z2
• • •	• • •	• • •	• • •

Wall

Wall ID	Component ID	Componetnt Type
W1	PL3	Polyhedron
W2	PL4	Quadrics
• • •	• • •	• • •

Surface

Surface ID	Edge ID	Description
S1	E1,E2,E3, E4, ...	Road
S2	E4,E5,E6,, E7, ...	Road
• • •	• • •	• • •

Edge

Edge ID	Begin Point	End Point
E1	P1	P2
E2	P2	P3
• • •	• • •	• • •

Texture

Texture ID	Type	Natural	Attribute	Data Pointer
T1	TIF	N	• • •	• • •
T2	RGB	Y	• • •	• • •
• • •	• • •	• • •	• • •	• • •

Point

Point ID	X	Y	Z
P1	X1	Y1	Z1
P2	X2	Y2	Z2
• • •	• • •	• • •	• • •

DEM

DEM ID	Texture ID	Type	Attribute	Data Pointer
D1	T3	TIN	• • •	• • •
D2	T4	GRID	• • •	• • •
• • •	• • •	• • •	• • •	• • •

Fig. 5. One rational database model for a city 3D mode

4.2 Development of Graphic User Interface (GUI)

We have developed a graphic user interface (GUI), which allows a customer to create buildings, roofs, and wall textures, to customize buildings.

- **Customizing Buildings**

A called *3D Model Builder* has been developed at client side (see Figure 8). A user is only required to input the corner coordinates and two heights of a building from client side, and to send his/her request to server side. The server will visualize the building by the VRML model. On the other hand, our spatial database stores building data, and users can take advantage of menus to select the data files from the database to build a building that he/she likes.

- **Customizing Roofs**

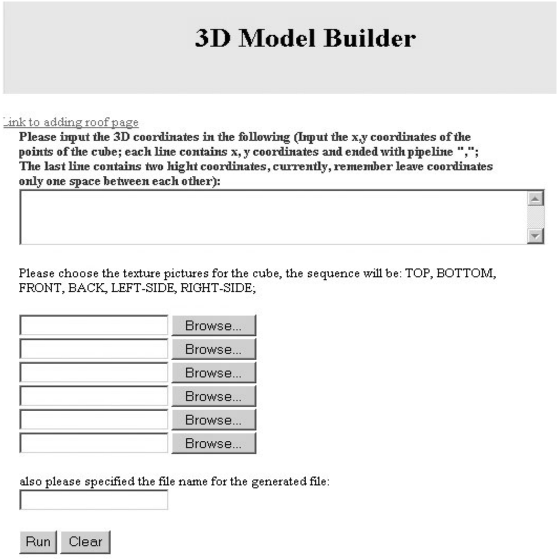
The client side is also designed with a GUI (see Fig. 9) for users to add various roofs to an existing building implemented in the step above. Users are first required to choose what type of roof they are interested in, and then input corresponding parameters. Recently, we have designed nine types of roofs described in Section 4.2. These roofs are stored in the roof database. Users can call the roof ID, corresponding to the VRML model ID, the specified roof will be automatically added to the VRML model that users specified.

- **Customizing Textures**

The client side also has a designed GUI for users to stick texture image onto the wall of an existing building created by the VRML model. A texture image database, in which 200 real wall texture images are stored, has been established, and various texture images are displayed in the interface. Users first choose a texture image with its identification (ID) that they are interested in, and then input the texture image ID, and a Building ID and a wall ID, the texture image will be automatically stick to the corresponding wall.

4.3 Visualization

This server/client-based system has been implemented. This system has functionality on pan, zoom, rotation, and visibility analysis features. The texture data have been used for wall features. We used this system to visualize Denver, CO, in which 70 buildings have been created via client side (Fig. 10). A client can pan, zoom in/out, and rotate the 3D model. With the texture integration in 3D city modeling, it makes the 3D visualization a much more real 3D representation.



3D Model Builder

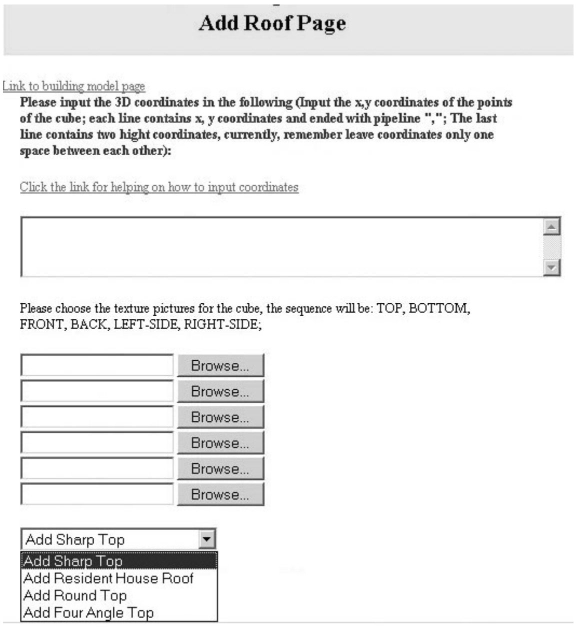
Link to adding roof page

Please input the 3D coordinates in the following (Input the x,y coordinates of the points of the cube; each line contains x, y coordinates and ended with pipeline ","; The last line contains two hight coordinates, currently, remember leave coordinates only one space between each other):

Please choose the texture pictures for the cube, the sequence will be: TOP, BOTTOM, FRONT, BACK, LEFT-SIDE, RIGHT-SIDE;

also please specified the file name for the generated file:

Fig. 8. Client front interface for generating building objects



Add Roof Page

Link to building model page

Please input the 3D coordinates in the following (Input the x,y coordinates of the points of the cube; each line contains x, y coordinates and ended with pipeline ","; The last line contains two hight coordinates, currently, remember leave coordinates only one space between each other):

Click the link for helping on how to input coordinates

Please choose the texture pictures for the cube, the sequence will be: TOP, BOTTOM, FRONT, BACK, LEFT-SIDE, RIGHT-SIDE;

Fig. 9. Client front end for making roof objects

5. CONCLUSIONS

This paper reports our research results of a project progress. Based on the developed data structure, we implemented data models using a relational database. Also, we built an automatic tool for creating buildings and roofs, and sticking textures. By storing 3D primitives into urban GIS databases, it is much beneficial for VRML programming when these primitives are retrieved from the GIS database. These 3D primitives appear at a very high frequency in a 3D city model. The results from our experiment demonstrated that this developed system is able to display and visualize 3D city data in a web environment for public accesses. Using the sidewall pictures taken from real city buildings, we refilled the white walls of urban buildings for a real appearance as seen in our real world cities. The player provides several click buttons to enhance the effect and position of visualization, including rotation, zoom, pan, seek, etc. The users can easily view details of a specific building, or an entire city in a moving 3D, or locate a specific building, and so on. This technique will bring current 3D GIS city modeling into a new era of development. Future work will implement further details of the 3D city model, such as adding database-querying ability, adding other geospatial objects, such as roads, rivers, grasses, etc.

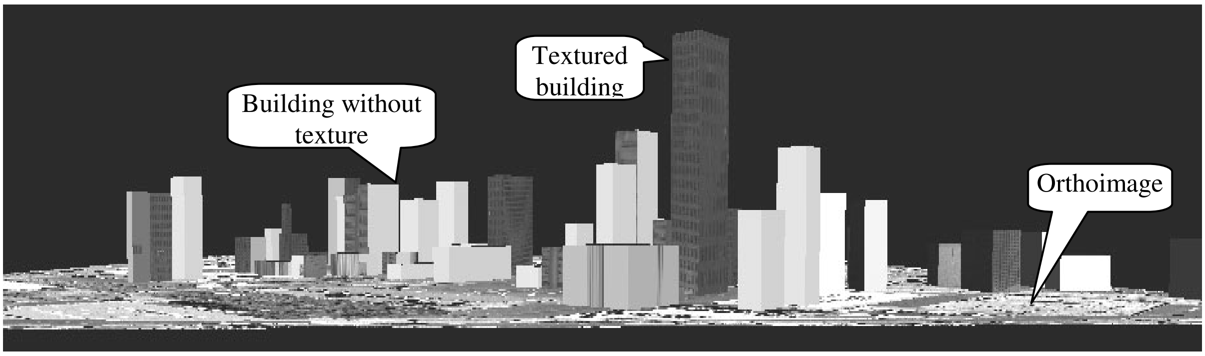


Fig. 10. A city model created by our system

ACKNOWLEDGEMENT

The project was funded by the US National Science Foundation (NSF) under contract number NSF 0131893. We would like to thank the project administrators at the City and County of Denver for granting permission to use their data. We especially thank Wolfgang Schickler at Analytical Surveying Inc. for their kind helps with the delivery and checking of the aerial images and digital surface model of downtown Colorado, CO.

REFERENCES

Belfore, A.L., II, 2001. An architecture for constructing large VRML worlds, *Transactions of the Society for Modeling and Simulation*, vol. 18, no. 1, March, pp. 24-40.

Belfore, A.L., II, 2002. An architecture supporting live updates and dynamic content in VRML based virtual worlds, *Int. Symposium on Military, Government and Aerospace Simulation (MGA 2002)*, April, San Diego, California, pp. 138-143.

Belfore, A.L., II, Roland Mielke, and Krishna Kunam, 2003. A framework for creating VRML visualizations from discrete event simulations, *in Proceedings of the International Symposium on Collaborative Technologies and Systems (CTS'03)*, January 2003, Orlando, Florida, CD-Rom.

Blaser Andreas, M. Sester, and M. Egenhofer, 2000. Visualization in an early stage of the problem-solving process in GIS, *Computer and Geosciences*, vol. 26, pp. 57-66.

Brutzman, D. 1998. The virtual reality modeling language and Java, *Communications of the ACM*, vol. 41, June, pp. 57--64.

Bruzzone, A.G., and G. Berrino, 1999. Modelling of urban services by VRML & Java, in *Proceedings of the 1999 Int. Conf. on Web-Based Modeling & Simulation*, San Francisco, CA, January, pp. 34--38.

Hibbard, B., 1998. VisAD: connecting people to computations and people to people, *IEEE Computer Graphics*, vol. 3, no. 32, pp.10-12.

Huang, B., Lin, H., 1999. GeoVR: a web-based tool for virtual reality presentation from 2D GIS data, *Computers & Geosciences*, vol. 25, pp. 1167-1175.

Kreuseler Matthias, 2000. Visualization of geographically related multidimensional data in virtual 3D scenes, *Computers and Geosciences,* vol. 26, pp. 101-108.

Lin. H., Gong, J. and Wang F., 1999. Web-based three-dimensional geo-referenced visualization. *Computers & Geosciences,* vol. 25, pp. 1177-1185.

MacEachren, M.A., Kraak, M.-J, 1997. Exploratory cartographic visualizatin: advancing the agenda. Computers and Geosciences, vol. 23, no.4, pp. 335-343.

Nadeau, D.R., 1999. Tutorial: Building virtual worlds with VRML, *IEEE Computer Graphics & Applications*, vol. 19, March/April, pp. 18-29.

Reddy, M., Y. Leclerc, L. Iverson, and N. Bletter, 1999. TerraVision II: Visualizing massive terrain databases in VRML, *IEEE Computer Graphics & Applications*, vol. 19, March/April, pp. 30--38.

Reddy, M., L. Iverson, and Y. G. Leclerc, 2000. Under the hood of GeoVRML 1.0, in *Proceedings of the Fifth Symposium on the Virtual Reality Modeling Language VRML2000*, Monterey, CA, February, pp. 23-38.

Rohrer, R.M., Swing, E., 1997. Web-based information visualization, *IEEE Computer Graphics and Applications*, July/August, pp. 53-59.

The Web3D Consortium, Incorporated, The Virtual Reality Modeling Language, http://www.web3d.org/Specifications/VRML97/index.html, 1998.

Verbree E., G. V., Maren, R. Germs, F. Jansen, and M. Kraak, 1999. Interaction in Virtual World views – linking 3D GIS with VR, *Computers and Geosciences,* vol. 13, no. 4, pp. 385-396.

Wang, X.H. and Gruen, A., 1998, A 3D City Model Data Structure and Its Implementation in a Relational Database, *The Proceedings of Spatial Information Science, Technology and Its Application*, Wuhan, China, pp. 429-436.

Zhou G., Ming Xie and J. Gong, 2000. Design and implementation of Attribute Database Management System for GIS (GeoStar), *Int. Journal of Geographic Information Science,* vol. 6, no. 2, pp. 170-180.

Zlatanova, Siyka, 2000. 3D GIS for Urban Development, ISBN: 90-6164-178-0, ITC Dissertation Number 69. Doktor der technischen Wissenschaften an der Technischen Universitate Graz.