VIRTUAL ENVIRONMENTS IN PLANNING AFFAIRS
Getting closer to geographic data, a better way!

Mohammed Abdul Mannan, Bogdahn Juergen.

Dept. of Geomatics, Computer Sciences and Mathematics, Stuttgart University of Applied Sciences,
Schellingstrasse 24, 70174 Stuttgart, Germany. - {abdulmannan, juergen.bogdahn}@web.de

KEY WORDS: City model, GIS, Planning, Programming, Web based, Visualization, Virtual Reality.

ABSTRACT:

Urban environmental elements and planning involve a complex data and phenomenon. It is required to have a better understanding of concerned geographic data to make a decision or plan for the applications like urban planning, infrastructure management, and visibility analysis and tourism promotion. Current practice in any public or private sector planning context at any spatial scale involves better visualization techniques. In many planning issues, GIS is combined with other several computer tools to enhance the planning process and decision-making. The rapid advancement in science and technology had opened wide options for a change and development of current methods and concepts. Computer data visualizations, not only help laypersons, who often have trouble in understanding or interpreting complex data, but they also can help experts with its various capabilities. High performance visualization tools provides dynamic work sphere to combine interactive urban environmental elements with internet based visualizations (web 3D) and Virtual reality. This study is based on such visualizations. The main focus of the research is to display the geographic data comprising of spatial as well as the non-spatial data in these virtual environments. The first part of project, named as web 3D project enables online internet access to 3D urban models including geometry and thematic attribute data. The second part of project is focused on virtual reality, which let the user immerse in the data, that is you are surrounded by the landscape and you can walk, fly around the city. One can visualize the existing scenario in C.A.V.E or H.M.D and can test future natural and built environments. And also a program is developed to access the attribute information like name of building, usage etc. for the objects in virtual reality, which will display in runtime while navigating the scene.

1. INTRODUCTION

The city is usually the centre of education, trade, industry, cultural life, government, etc. It offers a variety of opportunities, which always has attracted people from the villages. The increase in population inevitably leads to the increase in the both complexity of tasks that have to be planned and the information that has to be processed, like the research study area Stuttgart city in Germany. In many planning affairs, the need for 3D geo-referenced information is extremely high. For example: Issues related to 3D interaction between newly designed elements (buildings, roads, etc.) and existing infrastructure, environmental impact analysis, transportation networks, tourism and shopping information, real estate market, management and preservation of cultural sites, etc.

For planning of such cities and maintaining public areas and buildings, it's required to maintain the variety of data and respond to the increased demand for urban information. There are many deficiencies to handle and planning of cities in the existing information systems as also identified by many authors (Bodum et al 1998, Tempfli 1998). Most of the problems discussed refer to the representation and analysis of spatial objects.

The commonly established systems dealing with spatial information were originally designed to create, edit and display small 3D graphics models, show weaknesses in the 3D visualization of large models (Zlatanova, 2000).

Recent developments (power processors, graphics accelerators and standard hardware-implemented graphics libraries) have opened up new horizons for 3D visualization, like Web 3D and Virtual Reality (VR). Visualization on Web electronic market is considered as a new information revolution and tremendous amounts of money and manpower have been invested in Web-based commerce systems. A system is developed based on existing GIS (shape file) and Virtual model (VRML model of Stuttgart city) that links up with to visualize spatial and non-spatial data and will help user to improve the planning process. The user walks around, in the virtual environment and can point to objects in the scene and ask for information from GIS database.

Another domain of project is named as virtual reality (VR). Sophisticated devices or conventional equipment allow complete immersion with the model, simulating actions and observing reactions similar to those in the real world. VR offers new and exciting opportunities to visualize 3DGIS data. Users can walk through 3D environments; see newly planned buildings and experience changes in the landscape. In most cases, however, interaction with the data is limited to viewing. These visualizations have two conceptual specifics that result in insufficient means for GIS analysis. Most of them are not designed to deal with semantic information and spatial analysis is hardly in focus.

A program is developed using Tcl programming language extends Lightning VR software (Blach et al 1998) to access and display semantic information in such environment. This paper shows the concept and tools used to shape this idea. Various aspects, which have lead to the basic set-up of a Web 3D/VR-system, are discussed in two separate sections. First, we describe the need, tools and methodology under each section. And final conclusions were included from both the Virtual environments.

2. WEB 3D

An important factor in a 3D GIS is remote access to the data. Recent developments on the Web, i.e. VRML make possible visualization and interaction with 3D models. The exploration of the 3D worlds in the VR browsers ensures a certain level of virtual reality techniques. Existing VRML, HTML and other Web standards and software modules allow the development of GUI with limited effort, relying on some operations provided already by browsers. Therefore VRML and HTML will be employed as a front-end engine to the 3D GIS model. This project was started in order to visualize the 3D city model of Stuttgart and find a way to connect the objects in the 3D scene with their attributes in the land registry dataset. The motivation was to give the user the ability to interact with the 3D scene not only by moving around, but also to have access to the attributes information by clicking on the objects. The given 3D dataset only consists of geometry information so that an algorithm had to be invented, which connects the geometry with the attributes. In the test application the conversion from the given ASCII format to VRML and the connection between the VRML objects and the attribute data is done in one step.

This part is considered to be the Web3D part. Although the test application is a desktop application, it is possible to realize the concept on a website as well. The VRML viewer can be add in web pages and the application part could be realized as a Java Applet, which interacts with the viewer.

2.1 Tools

The chosen developing environment is Visual C++ 6.0 and for VRML visualization, the Cortona Viewer by Parallel Graphics. Parallel Graphics supports developers by the Cortona SDK (User guide), which allows you to add the Cortona VRML Viewer into your C++ projects as an ActiveX component. There are also some event handler classes to handle mouse events, and other interface functions so you can completely interact with the viewer. To access the given land registry information in shape format the C-library "shapelib" is used (Warmerdam, 2002), which can access the geometry in the shp-file as well as the attributes in the dbf-file.

2.2 Description of the 3D objects

The dataset delivered by the city administration Stuttgart, consists of several files in ASCII format storing the geometry of the buildings. These buildings are created of several parts. The 3-dimensional geometry of the roof is taken out of aerial stereo photos by semi-automatic analysis. The floor is created by intersection of the 2-dimensional outline of the building taken from the land registry and the digital terrain model. This outline should have been connected with the attributes but this connection is not supported in the ASCII files, maybe because the focus was only on visualization matters. To create the walls of the building, the 3D floor outline is connected with the roof geometry. A representative point is set inside the geometry, which is used for connection between 3D objects and the attributes in the shape-file.

2.3 Visualization

The 3D geometry dataset consists of several ASCII files, which are consisting of point tables. Besides the x, y, z-coordinates each point has got the attributes houseindex, wallindex, typeofarea (roof, wall, floor). With the help of these attributes it is possible to create an IndexedFaceSet-Node (IFS). The used fields are shown in Figure 1. This VRML-Node seems to be the best solution to describe the building geometry, because it can describe complex objects, which consist of several areas, very easily.

```
IndexedFaceSet{
exposedField  SFNode color NULL
exposedField  SFNode coord NULL
field    MFInt32 coordIndex []
field    MFInt32 colorIndex []
}
```

Figure 1. Part of VRML Specification '97

The used fields allow avoiding saving redundant points, which makes the VRML-File smaller than the corresponding ASCII-File. In the coord –field the points of the geometry are saved by the x, y, z coordinates, and in the coordIndex-field all points describing an area are listed by their index in the coord-field. So it is possible to check if a new point is already in the coord-field, if it is the point doesn't have to be saved again, only the index of the previously saved point has to be used.

The algorithm to convert the ASCII information into VRML syntax only has to go through the ASCII table and read the coordinates and the attributes to decide when to create a new building (IFS-Node), a new area or when to change the color of the created area to color roof and walls differently.

To be able to identify the objects, what is essential for the connection, you have to give each IFS-Node a unique identifier. In VRML it is possible to give a Node a name called DEF-Name. In the ASCII file the index for the buildings starts always with one, so if this index is used and the 3D scene consists of more than one ASCII file there is no unique identifier for the buildings. Therefore a combination of filename and index is used to form a kind of primary key for each building, which is also used for the connection later on.

2.4 Connection to attribute data

The connection is done by a geometrical test. Each building of the ASCII file has got a representative point inside its geometry. The location of this point in the 2D land registry is checked and if the point lies inside of a land registry building outline, the dataset connected to this outline is considered to be connected to the 3D building. This connection is saved in a text file (see Table.1), to avoid doing this test each time a model is loaded.

DEF-Name	DBF-Index	Representative3D

Table 1. Connection Table

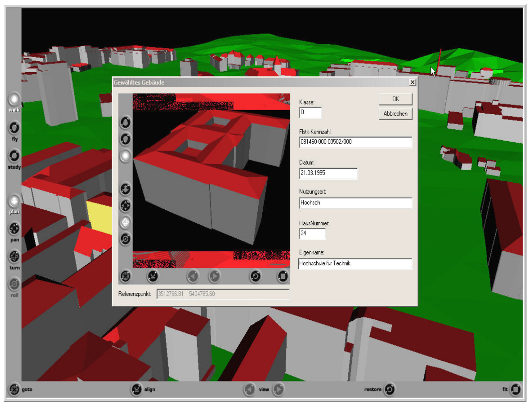
In this table ("connection table") the DEF-Name of the 3D object and the corresponding dbf index of the land registry dataset are saved. The representative coordinates are saved in addition for query or analyzing matters.

The concept of the test is fairly simple but there are some more difficult points in programming, which will not be discussed in this paper. To test if the representative is inside of the polygonal outline, it has to be connected with a very far point by a line. Now the intersection points of this line with the outline are counted. Even number of intersections means the point is outside, an odd number of points means point is inside. This test works also if the geometry has got void areas inside (Finley, 1998).

2.5 Select a building

To select a building the pick()-method of the Cortona Viewer is used. This pick() method works with the "ray" concept. When the user clicks into the scene the coordinates of the pointer are taken and a ray is sent through this position. The first object hit by the ray is the result of the pick() method. A pointer to this object is returned and you have access to the object and its fields. Now it is possible to read the DEF-Name of the returned object and find the corresponding land registry dataset in the connection table. With the dbf index the attribute information can be read from the dbf dataset and be displayed in a dialog (see Fig.2), the scene itself or used in any other way, for example, written to a file.

This method enables the user to use the 3D scene as an interface to the attribute data. So you can navigate through the scene and get information about any object, which has connection information. Though the connection is realized on a system working with files, it is possible to get geometry and attribute information from a database to do the connection. In this case a connection table is not necessary, because the DEF-Name of the 3D object can be add as a column to the database table. So it is possible to find object attributes by a SQL query. The connection to the database could be realized by ODBC in case of Visual C++, for example.



Gewähltes Gebäude

Klasse: 0

Flstk-Kennzahl: 081460-000-00502/000

Datum: 21.03.1995

Nutzungsart: Hochsch

HausNummer: 24

Eigenname: Hochschule für Technik

Referenzpunkt: 3512786.81 5404785.60

Figure 2. Selection and attribute display of the building (© Stadtmessungsamt Stuttgart)

2.6 Surface Visualization

The surface information is given as a TIN in DXF format. This format contains much information, which is not needed for visualization, like layer numbers, line colors, etc. Therefore it was parsed into a text file, which contains only the triangle information.

To visualize the surface a rectangle is calculated around the area with buildings. Then all triangles of the TIN, which are inside of this rectangle are selected and converted into an IFS- Node.
We are working on a concept to create IFS-Nodes for each parcel. At the moment, the surface is only one object so the connection to parcel attributes is not possible. To get one object for one parcel, which would be selectable, it is necessary to intersect the 2D parcel outline with the TIN to get the parts of the parcel for each triangle, as shown in figure 3.

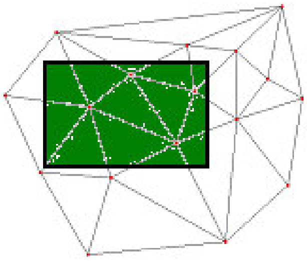


Figure 3. Clipping of TIN based on parcel polygon

These "result" parts (polygons) can be transferred into an IFS-Node. To visualize the surface for the buildings in scene, all parcel objects for the "building"-rectangle have to be created and visualized in the scene. In this way you can get a surface consisting of selectable parcel objects. The outline of the parcel is taken from the land registry dataset and it is connected to attribute data, so the index of the attribute data record can be used as DEF-Name for the created IFS-Node. In that way you have a connection between 3D object and attributes of the land registry.

To select a parcel object the pick() method is used again, and it will return the result object, and with the dbf index stored in the DEF-Name field the attributes can be read and displayed.

2.7 Query example

To show a kind of query or thematic map a small query example was realized. The query functionality can be extended with useful functions, but this test software only wants to show what is possible.

In this query example the user can color the building walls according to the use of the building, so when the user navigates through the scene, he can differentiate buildings with their colors for example university buildings in blue, offices in green etc, (see Fig.4).

To color the buildings a text file was created containing a table, which connects a use with a RGB triple (use-color table).
To color all buildings in the scene the connection table is used to find the corresponding data records to the buildings, the use of the object is read from the data record, the corresponding color is read from the use-color table and a VRML Viewer interface function is used to color the building with this color.

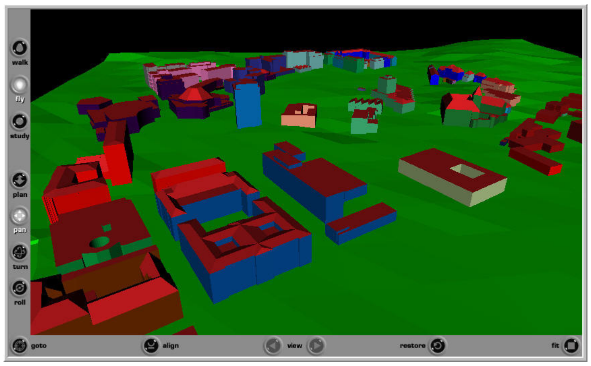


Figure 4. Thematic query based on USE from attribute table. (© Stadtmessungsamt Stuttgart)

3. VIRTUAL REALITY

The second part of the project is focussed on Virtual Reality. Virtual Reality is an immersive experience in which participants wear tracked, head-mounted displays to visualize, and listen to 3-D sounds and are free to explore and interact within a 3-D world. This part of project is aimed to allow user to interact with the world and display the attributes as text, while the user is still navigating around in immersive environment like C.A.V.E or Responsive Workbench.

VR is multi-modal that defines the more hardware oriented interface definition as the combination and cooperation of various input and output channels, which distinguishes the interface channels more from the point of media instead of the sensory perception. Dynamic behaviour introduces the notion of time into virtual environments. A highly interactive system has to deal with time in general and specifically with variable time frames and their synchronization. Responsive virtual environments should operate in real-time, that is, the response time and update rate of the system is high enough so that the boundary between user and virtual environment, the so-called interface, seems to vanish. This property is one of the major differences between VR systems and other 3D systems.

3.1 Tools

Whether they're images, 3-D sounds, or tactile vibrations, all aspects of VR must be coordinated and precisely delivered otherwise confusion will result. The computer system and external hardware that supply effectors (HMD or C.A.V.E) with the necessary sensory information comprise the reality engine (Pimentel and Teixeira 1993).

This project has been tested in the C.A.V.E at Fraunhofer Research Institite for Industrial Engineering IAO in Stuttgart using the application software "Lightning VR" developed by them. Its core component is a database called object pool. The interactivity and the behaviour can be introduced with functional objects, so-called scripts and communication channels, so-called routes. These communication channels define application specific event propagation and therefore the interactivity (Blach et al 1998).

Tcl/Tk programming language is used for interaction with lightning, as it is easy to integrate into other systems (Ousterhout, 1993).

3.2 Input Data

Input geometry data for this application is the same VRML model converted from the ASCII file in the above web3D project. The model consists of buildings and terrain information of Stuttgart city. Each building has unique name defined to the node that is describing the geometry. Lightning VR uses LibVRML97 for reading and displaying VRML files, which is a portable C++ class library (Chris Morley, 2002). This library can access the properties like position, orientation, scale, centre of bounding box etc. of the world files in scene.

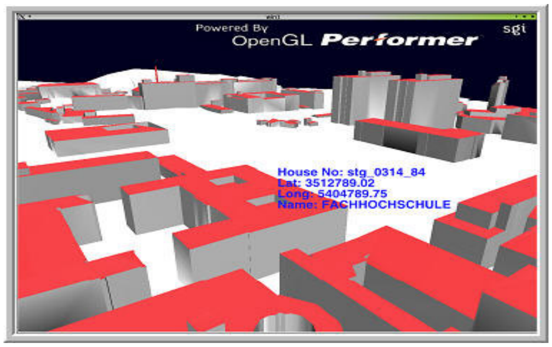
The attribute information concerned to the objects, which was purposefully created and saved as text file called "connection table" in the first part of project. This connection table has the information like Definition name, x, y, z, House name etc. Here definition name will act as primary key in connecting spatial and non-spatial data.

3.3 Selection of building

The input device can be a 3D mouse or glove etc, for the identification of objects in the virtual environment. The properties like position, orientation, scale etc, of input device is calculated by motion sensor in Lightning. In order to make input device a selecting tool, a ray has to be made active using ltSelectRay. It's a library under Lightning server, which is invisible. It can sense the objects hit and returns field values like first object hit, position & orientation, count of hit objects, etc, to the application. It is quite similar to pick() method of Cortona in Web3D project. The position and orientation values of ray were now routed (borrowed) from motion sensors properties, to make the input instrument act as a selecting tool. Now this ray (cursor) can be used to sense any object in virtual world. While moving the input device over building, it senses the object, recognizes it and sends the definition (name) to NodeHitOut property of SelectRay. This definition name will be used as primary key to match with the concerned attribute data that is stored in the text file.

3.4 Connection to attribute data

The attribute data of the model is available as separate text file. The program written in Tcl will read the text file line by line and check if the definition name from the ray, matches with first field of the text file. Once matched, the whole line of the text file is converted to text element using ltText property of Lightning and displayed over the object, which you pointed in runtime scene situation, as shown in figure 5.



House No: stg_0314_84
Lat: 3512789.02
Long: 5404789.75
Name: FACHHOCHSCHULE

Figure 5. Selection and attribute display of the building (© Stadtmessungsamt Stuttgart)

3.5 Implementation

Perception is related to the degree of realism experienced by the viewer. The basic concept of VR is the direct coupling of the head position and viewing direction of the user with the viewing position used to generate the image on the display. The head position of the user is constantly tracked and fed into the display algorithm to calculate at each moment a correct stereoscopic and perspective display of the scene. This strong coupling of the current eye position of the user and the image offered by the display system, gives the user the illusion of 'immersiveness'. The ultimate VR-systems completely immerse the user's personal viewpoint inside the virtual world. A major distinction of VR-systems is the mode with which they interface to the user. Current systems range from true immersive to nonimmersive displays. These involvements allow the planners a better platform for decision-making and also help laypersons to understand the complex geographic data, as if they see in reality.

The users manipulate what is perceived to be "real" objects in the same manner as they would manipulate them in the real world, as opposed to the typing, pointing and clicking, one traditionally uses to manipulate objects when you interact in other computer environments. For example, to move an object in a VR environment, you may grab the object with your hands, lift it as you normally lift objects in the real world, and put it down wherever you want it inside the virtual environment. Now the program lets you see the attribute data as well.

4. CONCLUSIONS

Whether the challenge is to pick a location for a new facility or decide how to construct it, GIS is important in the decision-making process. By combining 3-D GIS, integrated with applications like Web3D to share information over Internet and get input from the public will enhance planning process. When combined with immersive environments, results are superior to desktop systems for characteristics such as 'intuitive working', 'real-time interaction' and 'full-scale modelling in a 3D immersive environment'.

3D information systems with a connection to attribute data could be more helpful in addition to geometry visualization. For example in the field of underground supply lines, which have a complex structure. In a large area of network, supply lines are layered one above the other and so the importance of z coordinate exists. Planners can visualize the network more closer may be to find space for new lines or to know details about with access to attribute data.

The major advantage of Web 3D concept is it can create the database connection between the 2D shape file and the 3D VRML model by means geometry and representative point (A point inside of geometry). So, the creation of 3D VRML model can be from any source for example aerial photos. whilst the VRML model should be georeferenced common to the 2D shape file. The project can be further improved by providing functionalities of online updating of attribute data and other thematic queries based on attribute data as well.

The concept of connecting and display of attribute and 3D geometry for VR visualizations were successful with direct use of text file created in web 3D part. And a lot of improvements can be done on the basis of this idea to make it more useful for planning applications. The VR application can be further enhanced to access shape files based on same concept above, enabling to perform basic functionalities as of GIS software.

4.1 References and/or Selected Bibliography

References from Books:
Pimentel, Ken and Teixeira, Kevin 1993. *Virtual reality through the new looking glass.* Windcrest Books, USA, pp. 66-86

Stuart, Rory 1996. *The design of virtual environments.* McGraw-Hill Inc. USA, pp. 97-125

Zlatonova, Siyka 2000. Phd thesis, 3d GIS for Urban development, Thesis ICGV, GrazUT, Austria and ITC, The Netherlands, ITC Dissertation Series No69, http://www.gdmc.nl/zlatanova/PhDthesis/pdf/content.html

References from Other Literature:
Bodum, L., I. Afman and J. Smith, 1998. Spatial planning moves out of the flatlands, in: *Proceedings of AGILE*, Enschede, The Netherlands.

Ousterhout, J., 1993. Tcl and the Tk Toolkit (Addison-Wesley, Reading, Massachusetts).

Tempfli, K., 1998. Urban 3D topologic data and texture by digital photogrammetry, in:*Proceeding of ISPRS*, March-April, Tempa, Florida, USA, CD-ROM.

User Guide, Cortona SDK 4.1
http://www.parallelgraphics.com/developer/

References from websites:
Blach, R., Landauer, J. Rösch, A., and Simon, A., 1998. A Highly Flexible Virtual Reality System
http://vr.iao.fhg.de/papers/fgcs98.pdf (accessed Dec 2003)

Finley, Darel. R., 1998. Point-In-Polygon Algorithm.
http://www.alienryderflex.com/polygon/ (accessed Aug 2003)

Morley, Chris 2002. Tool: libVRML97
http://www.vermontel.net/~cmorley/vrml.htm.
(accessed Sep 2003)

VRML '97 Specification.
http://www.web3d.org/x3d/specifications/vrml/ISO_IEC_14772-All/index.html. (accessed Jan 2003)

Warmerdam, Frank 2002. Tool: Shapelib v 1.26.
http://shapelib.maptools.org/ (accessed May 2003)

4.2 Acknowledgements

We would like to express heartfelt gratitude to Prof. Dr. Volker Coors at Stuttgart University of Applied Sciences for the continuous support, guidance and motivation throughout the research. Our sincere thanks to City administration department (Stadtmessungsamt), Stuttgart for providing the necessary data, without which, the idea wouldn't have proper fuel. We would like to extend our thanks to Fraunhofer Research Institute for industrial training IAO in Stuttgart, for providing software and hardware for the project part of Virtual reality, especially Mr. Frank Haselberger for his invaluable support.