GEOGRAPHICAL INFORMATION IN VIRTUAL ENVIRONMENTS

Jos van Loo[a] and Joost van Lawick van Pabst[b †]

[a]Delft University of Technology, Delft, The Netherlands
[b]TNO Physics and Electronics Laboratory, The Hague, The Netherlands

ABSTRACT

We studied the combination of a Geographical Information System (GIS) and a Virtual Environment (VE). The goal was to establish a bi-directional link between a GIS and a virtual environment. The first step was to combine three types of data to build the 3D world and store it into the GIS: a digital elevation map, a photo image and GIS data of a particular area of the city of Rotterdam.

The GIS and virtual environment are communicating through an intermediate layer. Based on the position of the user in the virtual environment and the type of interaction, this layer constructs a query that is send to the GIS. In return, the GIS sends the results of the query back to this layer, that transforms it into correct virtual primitives for presentation to the user. The virtual environment presents changes in the GIS by polling the database frequently. This allows multiple users interacting with the same GIS at the same time, seeing the changes other users make.

Several widgets in the virtual environment enable the user to interact with the GIS. It can be concluded that interacting with GIS-data in a virtual environment enhances the understanding of the data.

Keywords: Geographic Information Systems, Virtual Environments, Interaction

1. INTRODUCTION

A Geographical Information System (GIS) is an important tool in many fields of analysis of spatial data[1]. The possibilities to present and manipulate geographical data and their relations will offer substantial opportunities to aid geographical decision support and planning. Visualisation presents the data, but it also offers facilities to manipulate data. Traditionally 2D-GIS systems present data in maps, in which analysis and manipulation are possible.

Nowadays, more and more attention is paid to the third dimension, which has consequences for the GIS and the visualisation. The data must be gained and stored in three dimensions, which involves more sophisticated algorithms to store and query the data. The visualisation of 3D environments requires rendering algorithms and sophisticated hardware to perform these algorithms in real time.

Interaction with spatial data using 2D representation of the 3D data on screen or desktop, has some restrictions because spatial data is difficult to interpret in 2D. Also the standard user interface metaphor (windows, mouse, pop-up menu's) is not useful when a sense of immersiveness is desirable. Virtual Environments (VE) offer a natural way to represent the 3D data and their relations. Also interaction with 3D data is more intuitive in a VE.

We studied the combination of a Geographical Information System (GIS) and a Virtual Environment (VE). The goal was to establish an interactive bi-directional link between a GIS and a VE. This allows users to navigate in the VE and simultaneously interact with the GIS. This raises two important questions: "How is the interactive bi-directional link between the GIS and the VE established?" and "Which interaction techniques are best suited to the task?".

2. CONSTRUCTING THE VIRTUAL ENVIRONMENT

The first step was to combine three types of data to build the 3D world and store it into the GIS: a digital elevation map, a photo image and GIS data of a particular area of the city of Rotterdam. The digital elevation data is defined by a 728x708 regular grid. The dataset was triangulated for visualisation purposes, resulting into about one million triangles. These large quantities induced three performance problems: database search time, network transport time and real time visualisation of the triangles. Decimation of the triangles reduced the number of triangles to 5% of the original. This large reduction was

Part of the IS&T/SPIE Conference on Visual Data Exploration and Analysis V • San Jose, California, USA • January 1998
SPIE Vol. 3298 • 0277-786X/98/$10.00

possible without losing detail because in this case, the elevation map consisted mainly of rectangular objects. We have adapted the algorithm of Schroeder[2] for this purpose. The triangles are stored in tristrip format in the database. Tristrips not only speed up the visualisation process, but also require less storage space in the database. Hence the database search-time is reduced and a less amount of data is transported through the network. Table 1 shows the difference in performance using the regular grid and the tristrip format.

Dataformat	data retrieval (sec)	framerate (Hz)
regular grid	409	2-5
tristrip	40	12-15

Table 1 Performance comparison dataformats

The basic 3D model obtained more realism by mapping the photo image of the area on the elevation map. Furthermore, we have modelled the 3D GIS data by hand, because it was not available at the time. The 3D GIS data consists of 3D rectangular blocks. Each block represents an object in the environment, with the following characteristics: object type, object name and object neighbours.

3. NAVIGATING THROUGH THE 3D MODEL

We have implemented two methods for navigating through the model. The fly-and-walk-through method enables users to move instantly through the 3D model in the virtual environment. We have used a generic controller for the input devices, which offers the user the possibility to use either a Spacemouse or a Flybox to navigate. The point-and-fly approach shows the user a 2D global map (top view) of the area. The user can pick a location on this map, and is flown to this point in the virtual environment. The user can also specify multiple locations (points) on the 2D global map, resulting in a flight through the 3D virtual environment along the path defined by these points. Figure 1 shows selections of several locations (blue), the path is interpolated between these locations. The arrow (yellow) indicates the line-of-sight.

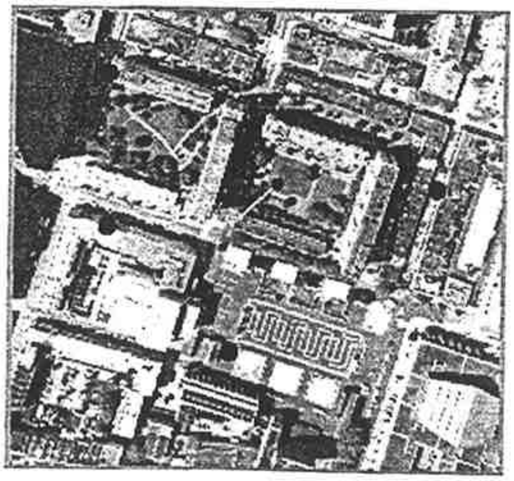

Figure 1Point-and-fly.

4. ESTABLISHING A BI-DIRECTIONAL LINK BETWEEN THE GIS AND VE

The GIS and VE are communicating through an intermediate layer. This layer (Gis2Ve97) is depicted in Figure 2. Using an intermediate layer, produces a loosely coupled link between the GIS and VE which makes for example replacing the GIS with another one, relatively easy. The GIS2VE97 architecture is based on the GIS2VE95 architecture[3]. Users of GIS2VE97 perform actions in a VE that evoke GIS transactions. These transactions return data, that is presented in the VE to the user. This means that user actions in the VE must be translated into transactions in the GIS. This translation is performed by the *Query Generator*. Execution of a query (e.g. a selection of an object) in the GIS, results in a selection of the data. The *Result Interpretator* translates this selection into geometric primitives that are presented in the VE. The *Tracker Controller* provides user position and orientation in the VE. The core of GIS2VE97 is the *Event Handler*, which monitors all user actions and GIS events, and takes proper actions. The *Rule base* is used for e.g. help functionality, but has not been implemented.

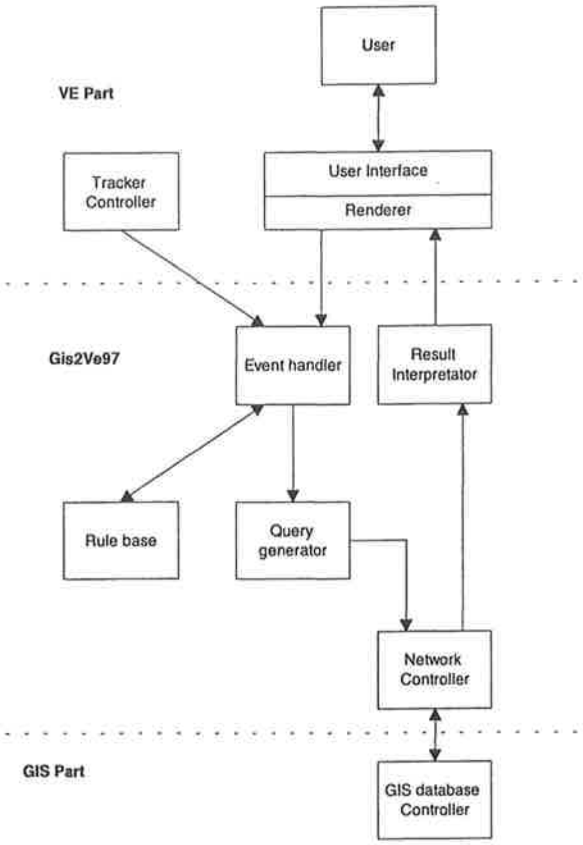

Figure 2 GIS2VE97 Architecture

Figure 3 shows the dataflow of GIS2VE97 in more detail. The task of the *Query Generator* is to construct a SQL-query that can be executed in the GIS. To perform this task, it needs three types of input data:

1. Position and orientation of the user. (E.g. used for distance measurement)
2. Position and orientation of objects selected by the user. (E.g. used for object property retrieval)
3. Widget type, used to identify the type of user action. Widgets are discussed in more detail below.

The result of the SQL-query is returned to *Result Interpretator*. Like the *Query Generator* it uses the same three types of input data mentioned above to construct a specific data representation in the VE. We have used coloured blocks, 3D text and numbers and audio to present the results.

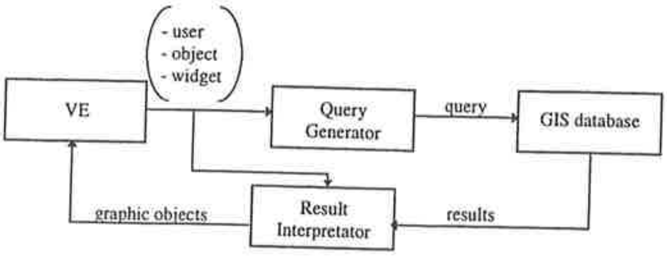

Figure 3 GIS2VE97 dataflow

The GIS supports a wide range of functions for selection of data and spatial calculations. The ultimate goal is to make all functions available to the user in the VE. Because of the wide range, it is necessary to classify these functions into coherent classes. The first class contains functions applicable to one object in the GIS, the second class contains functions that are applicable on two or more objects. This classification is summarized in Table 2. The number between the brackets stands for the number of objects the functions are applied to.

Class	Functions	Query result
Directional relations (1)	Above, Behind, Below, Front, Left, Right, North_of, South_of, East_of, West_of	multi-object
Transformation (1)	Scale, Translate, Orientate	single-object
Object properties (1)	Area, Center, Contents	text, number, boolean
Topologic relation (2)	ContainedBy, Contains, Cross, Disjoint, Equal, Inside, Intersect, Overlap, Topology, Touch	boolean
Metrics (2)	Distance, MaxDistance, MinDistance	number

Table 2 Function classification

5. INTERACTION TECHNIQUES

Widgets are used to encapsulate the look and feel of interaction techniques used in interactive applications. A number of widgets were constructed for interaction with the GIS to examine which were best suited.

First, a 2D menu widget was made. Each menu item represents a group of specific building types and by activating one of them, all buildings of that specific type are highlighted in a specific colour in the virtual environment. The 2D menu widget is presented right in front of the user in the virtual environment. Menu items can be selected by moving a 3D arrow and pointing towards one of them. When pointed correctly, the item is highlighted and selected by pressing a button on the control stick of the Flybox. Pointing correctly at the 2D menu proved to be a difficult job in the 3D environment.

Next, an alternative 2D menu was used (Figure 4a). The items can be selected by simply moving the control stick of the Flybox and each item is highlighted for a few seconds in sequence. When the desired item is reached, it can be selected by pressing the button on the control stick. This resulted in a much better control of the 2D menu widget because pointing was no longer needed.

The 2D menu was good for a small set of possible selections. For larger sets, a sundial menu widget[4] was implemented (Figure 4b). Here, the menu items are placed in a circle in front of the user in the virtual environment. Items are selected in the same manner as the 2D menu widget. By making the sundial widget semi-transparent, the user is able to keep an eye on the situation in the virtual environment behind the sundial widget. This widget proved to be easy to use.

The distance measure widget was added to get clues about dimensions in the environment (Figure 5a). This widget is presented as a 3D arrow that can be pointed into a direction in the environment. By pressing the button on the control stick, the arrow stretches along a straight line to the nearest obstacle and presents the distance (from the users position) in meters in front of the user in the virtual environment.

A pointing widget, a 3D arrow, was made to select 3D objects in the virtual environment (Figure 5b). This arrow initially points straight into the viewing direction of user. When moving the users position and orientation in the environment, objects were highlighted when the arrow was pointing towards them. It was also possible to change the orientation of the arrow directly but this turned out to be less intuitive. Once an object is highlighted, it can be selected by pressing a button on the Flybox. The information about the object (we only labelled names) is presented in textformat in the virtual environment in front of the user and is also presented by an audiosignal (we recorded the information).

The Sequential 3D and the Sundial menu implement functions of the Object Properties class (Table 2). The distance measure widget implements functionality of the Metrics class. The pointing widget is used to implement functions from the Directional relations and the Object properties class, depending on the context.

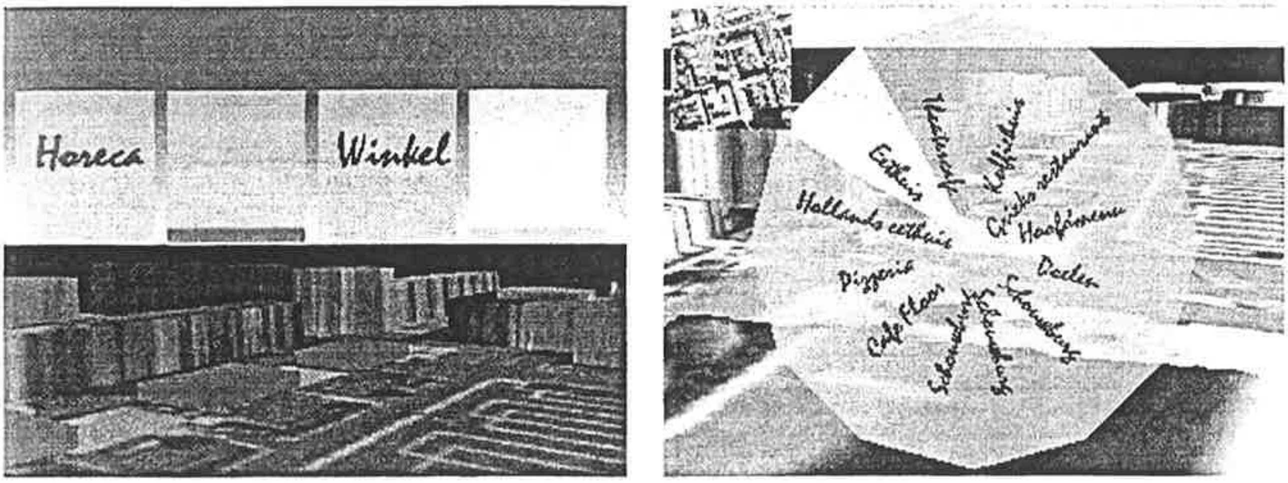

Figure 4a Sequential 3D menu. 3b Sundial menu

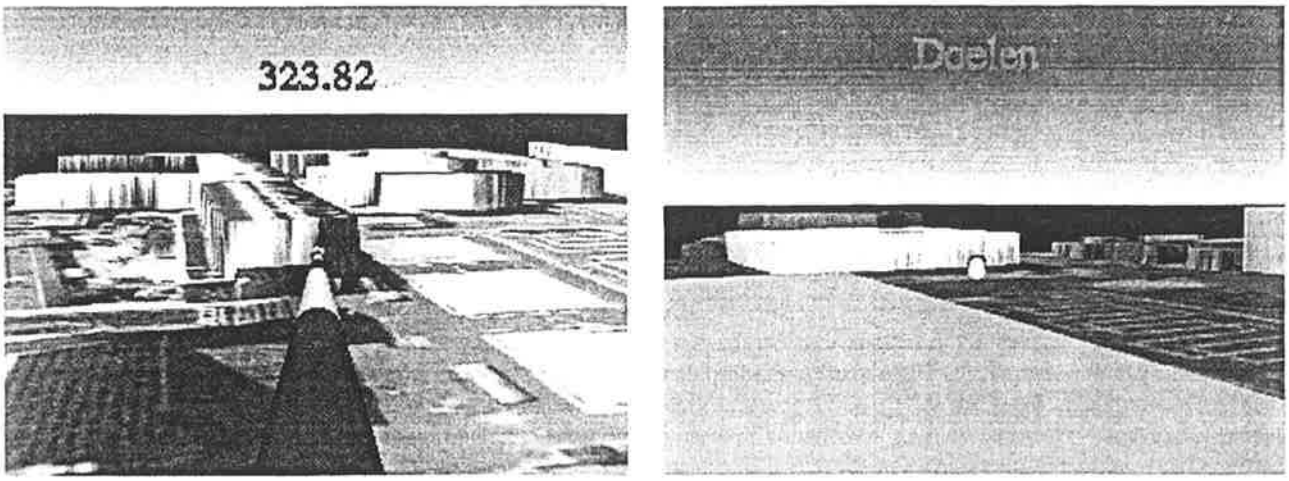

Figure 5a Distance measure widget. 5b Pointing widget

6. CHANGING THE 3D ENVIRONMENT

Three more widgets were implemented to add, move and delete 3D objects in the VE (Figure 6a,b). A number of predefined 3D models are presented to the user in a sundial menu widget. Selecting one of them, places the object in the VE in front of the user, on the surface. The control stick can be used to scale the object. Then, the object can be moved to the desired location in the environment. Object are also added to the GIS along with their locations.

Finally, a deleting widget was constructed to remove 3D objects from the environment and the GIS. This widget is represented as a pair of scissors in the virtual environment. Selection of an object with this widget, removes the object from the virtual environment and the GIS.

These three widgets implement functions of the Transformation class (Table 2).

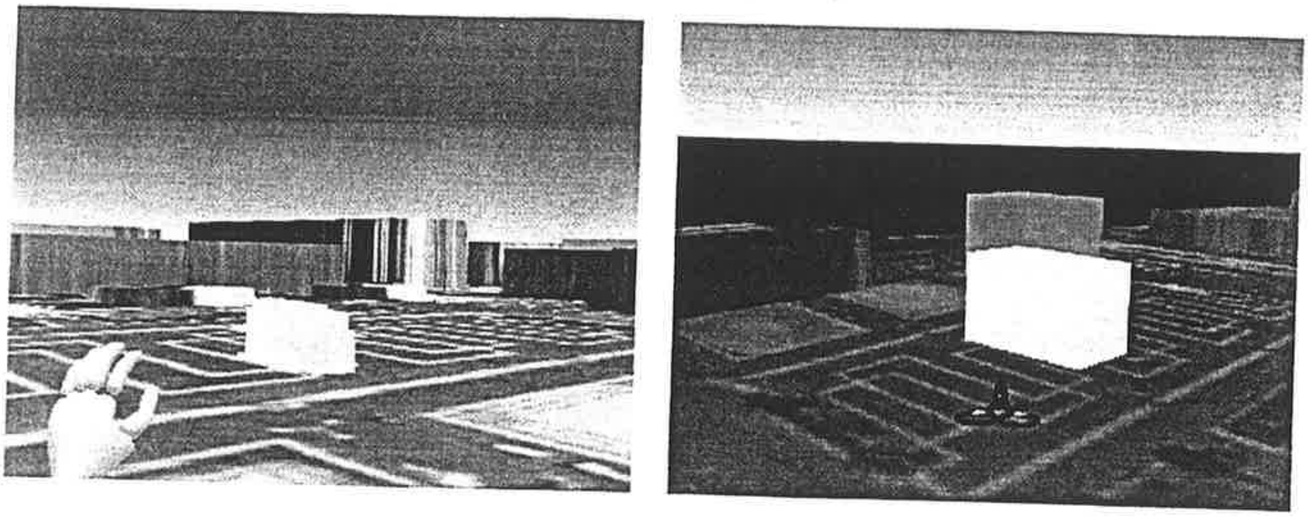

Figure 6a Moving 3D object widget. 6b Deleting 3D object widget

7. IMPLEMENTATION

We have used Informix Illustra as the GIS database and the SGI Iris Performer library to build and visualise the VE. Illustra runs on a Sun Sparcstation 2 and the VE on a SGI Onyx or Octane. The GIS and VE exchange information through Internet. Illustra offers an API for accessing the database and its functionality. GIS2VE97 is implemented in C++.

From Table 2, we have implemented one function of each class except the Topologic relation class.

The VE can be displayed in several ways. For single user purposes the environment is displayed using a desktop monitor, with or without stereo shutter glasses. A parabolic projection screen is used for presentation purposes to an audience.

GIS2VE97 polls the GIS frequently, to monitor changes in the data. This allows multiple users interacting with the same GIS at the same time, seeing the changes other users make.

8. CONCLUSIONS

It can be concluded that the visualisation of, the navigation through and the interaction with the GIS using a VE enhances the understanding of the abstract 3D GIS data. The bi-directional interactive link was implemented successfully. The navigation and selection widgets have proved to be very useful. Especially the sundial menu widget was very intuitive. Also audio feedback is a good way of presenting information in a VE. Because of the lack of textures on the sidewalls of the buildings, the user has no sense of proportions of the actual dimensions of the environment. The distance measuring widget addressed this shortcoming.

The prototype was designed for small environments and small objects. Tests with larger objects not only showed a decrease in system performance, but also overshoot the mark to give the user more insight in the 3D GIS data. Smart caching techniques are needed to solve the performance problem. The polling mechanism caused some overhead on CPU time and network and should be replaced by an event-driven mechanism.

9. REFERENCES

1. P.A. Burrough, "*Principles of Geographical Information Systems for Land Resources Assesment.*" Monographs on Soil and Resources Survey No 12. Oxford University Press, Oxford, 1986
2. W.J. Schroeder, J.A. Zarge, and W.E. Lorenson, "*Decimation of Triangle Meshes*", Computer Graphics, Vol. 26, Nr. 2, (Proc. Siggraph), July 1992.
3. G.J. Jense, H.C. Donkers, "*Dynamic Management of Geographic Data in a Virtual Environment*", Proceeding of 3rd Eurographics Workshop on Virtual Environments, Monte Carlo (Monaco), February 1996.
4. S. Bryson, "*Approaches to the Successful Design and Implementation of VR Applications*", 1994, URL: http://www.nas.nasa.gov/~bryson/papers/application.design.ps

† Further author information:
J.v.L.: Email: josl@and.nl; Telephone: +31 10 2759242
J.v.L.v.P.: Email: vanLawick@fel.tno.nl; Telephone: +31 70 3740283