CITY INFORMATION VISUALIZER USING 3-D MODEL AND COMPUTER GRAPHICS

ATSUKO KAGA[*1], YOSHIKI SHIMAZU[*2],
*[*1]Hankyu Corporation, Osaka Japan [*2]Tokyu Research Institute, Tokyo Japan*

AND

TOMOTAKA YAMAUCHI[*2] , HIRONOBU ISHIHARA[*1], TSUYOSHI SASADA[*3]
*[*3]Department of Environmental Engineering, Faculty of Engineering, Osaka University, Osaka Japan*

Abstract. 3-D models and computer graphics with its visual characteristics enables easier understanding of various information. Up until now 3-D models and computer graphics has not been used for the analysis of city information due to its high cost and the need for special techniques. Currently, we have discovered new technology in hyper medium based on network technology and lower costs. This paper focuses on the construction of an interactive and visual 3-D city information system, aiming at the 'idea processor' for research and analysis of city planning and market research. We have discovered the requirements necessary for the City Information Visualizer system. Using this technology we will construct the prototype system of the 3-D City Information Visualizer. This system is based on the personal computer and the Client/Server system. The system is then applied to practical city analysis. This paper presents the prototype system and its evaluation in a real project.

1. Introduction

City information is necessary to persuasively present the principle of land and building development to land developers and building owners. In order to understand the changing dynamics of a city, surveys and analyses of landowners, buildings, tenants, pedestrian traffic flow, etc... must be undertaken. This city information is obtained either qualitatively or quantitatively. Useful city information must be selected along with its

CAADRIA '98 : Proceedings of The Third Conference on Computer Aided Architectural Design Research in Asia.
eds. T. Sasada, S. Yamaguchi, M. Morozumi, A. Kaga, and R. Homma
April 22-24, 1998. Osaka University, Osaka, Japan. Pp. 193-201

analysis methods and its presentation. The presentation and visualization of city information include lists, graphs, 2-D maps, and computer graphics. Computer graphics is easy to understand and can eye point selections.

We will change the building form to a 3-D model and view city information using city information databases, 3-D models, and computer graphics. The system is structured so that the analysis results can be viewed by combining data attributes with the 3-D building model. In this way it is possible to create and restructure the simplified model serving our purpose of analysis. Moreover, computer graphics has the advantage of setting up and switching over the eye point. The system can grasp analysis results and shift to and from the whole city model and its parts quite easily. Because these portions can be removed from the 3-D model easily, it is possible to visualize, analyze and eliminate city information based on a hypothesis. In other words, city information is simplified by specializing and emphasizing the attributes and functions using computer graphics, and can explain, interpret, and present analysis results.

In addition, with interactive viewing of analysis results, it becomes possible to express results in various ways based on the user's ideas. Namely, this interactive 3-D City Information Visualizer can be an ‘idea processor’ for the city planner.

2. The requirements for City Information Visualizer

The following are the conditions necessary for constructing the City Information Visualizer.

2.1. CONSISTENT GRAPHICAL USER INTERFACE

To use this city planner system as an ‘idea processor’, the computer environment must be user friendly. The use of a consistent Graphical User Interface (GUI) across various applications becomes necessary. GUI uses visible and intuitive icons, standardized designs, and presents a user friendly interface. Computer use is simplified and easy to use.

2.2. INDEPENDENT OPEN SYSTEM ARCHITECTURE

To reduce costs, an independent open system which processes text, images, movies, 3-D models and networks adapted to a wide area network will be utilized eliminating the need for high performance machines and professional teams. The system is composed of free software and an open data format, low in price, easy to use, and can change the attribute data. It works as an informal, intense, open, and collaborative planning process.

2.3. ECONOMICAL SYSTEM ARCHITECTURE

Low initial and running costs are essential for system standardization. This system uses and combines the free software and techniques.

3. Key Technologies

Low initial and running costs are required for system standardization. Free software and techniques are combined and utilized.

3.1. VIRTUAL REALITY MODELING LANGUAGE (VRML) 2.0

VRML is one of the most important standard technologies in 3-D graphics and internetworking. VRML browsers allow the user to walk-through the 3D virtual environment in real time. One of the most important characteristics of VRML2.0 is its extensibility and ability to support other languages. It is this ability to add new elements and support an extensibility model that allows new objects to be defined. In other words, VRML data activity can be manipulated from other external development languages.

3.2. JAVA AND JAVASCRIPT

Java is a network object oriented programming language that is platform independent and can be very useful in the creation of distributed applets (small Java applications) that forms the body of a larger system. JavaScript is a programmable API that allows cross-platform scripting of events, objects, and actions. VRML2.0 supports Java and JavaScript as well as other external development languages.

3.3. LIVECONNECT

LiveConnect enables communication between JavaScript and Java applets within a page and between JavaScript and plug-ins loaded on a page. JavaScript is used to access Java variables, methods, classes, and packages directly and controls Java applets or plug-ins with JavaScript. It is equipped with Java code to access JavaScript methods and properties.

3.4. EXTERNAL AUTHORING INTERFACE (EAI)

The EAI allows control of the contents of a VRML browser window embedded in a web page from a Java (tm) applet on the same page. This is accomplished with a browser plug-in interface that allows embedded objects on web pages to communicate with each other.

3.5. OPEN DATABASE CONNECTIVITY (ODBC) AND JAVA DATABASE CONNECTIVITY (JDBC)

Open Database Connectivity (ODBC) technology provides a common interface for accessing heterogeneous SQL databases. This enables a developer to build and distribute a client/server application without targeting a specific Database Management System (DBMS). ODBC Applications are not tied to a proprietary vendor API as a benefit and a standard in the PC world. JavaSoft has developed a standard SQL database access interface, the JavaDatabase Connectivity (JDBC) API. This API provides Java programmers with a uniform interface to a wide range of relational databases, and provides a common base on which higher level tools and interfaces can be built.

4. The City Information Visualizer at present

4.1. SYSTEM STRUCTURE

The City Information Visualizer has been developed using PC/AT hardware and the WindowsNT operating system. The Windows environment satisfies the system conditions and can take advantage of the new technologies as mentioned above.

Figure 1. System Output.

4.2. CONTENT OF THE DATABASE

In this system the database consists of various fixed quality data compiled by Tokyu Research Institute regarding land, buildings, tenant, and pedestrian traffic flow in the Shibuya-zone, Tokyo in addition to the 3-D model data of the buildings and infrastructure.

First, the survey of the land and buildings served the purpose of comparing the change in situation in 1988 with that of 1996. The survey area consists of a radius of about 700 meters with Shibuya station as its center, containing approximately 2400 buildings. The survey was carried out as follows and the results have been tallied.

Change in situation of buildings after 1988 (name, scale, use)

Change in situation of tenants after 1988 (name, scale, use, category of business)

Second, the pedestrian traffic survey served the purpose of comparing the change in the situation in 1988 with that of 1996, similar to the survey of the land and buildings. The survey was conducted at 25 points of pedestrian traffic flow in and around Shibuya station. The survey was carried out as follows and the results have been tallied.

Direction (flow of pedestrian traffic from station to town and from town to station)

Time (every thirty minutes)

Weekdays and holidays

Group classification (men and women, employee or students, six groupings on weekdays, two groupings on weekends and holidays)

Stationary pedestrian traffic flow

A database on the above mentioned survey results is available in spreadsheet form. 3-D model data on the buildings, area, railway, and road in the area of survey is also available. In particular, the buildings pinpointed in the survey are transformed from Geographical Information System (GIS) data.

4.3. RELATED PROGRAM

We have developed programs for an automatic 3-D building modeler and a data transformer.

4.3.1. 3-D building modeler using GIS data

The system requires work saving to make 3-D building data. We have developed a program to create 3-D building models which automatically extracts height data of buildings using Shibuya data from GIS software (MapInfo).

4.3.2. Data transformer to VRML format

This system requires one to one correspondence of 3-D building model data in VRML format and the attribute data of the buildings. In order to accomplish this a unique number must be added to the 3-D building model data. Therefore, we have developed a data transformer program that adds the unique number to VRML data automatically.

4.4. GRAPHICAL USER INTERFACE (GUI)

We have developed the GUI based on Hyper Text Markup Language (HTML). The GUI selects the attribute data using the 'form language' of HTML enabling the user to select the attribute data easily. The 'form language' has a 'radio button' for selecting only one object from several choices, a 'check box' for selecting plural choices, and a 'text area' for inputting text. It is adapted to select the search area of databases similar to this system.

Analysis results are expressed by computer graphics using the browser, 'Cosmo Player'. The selection of attribute data and the analysis results by computer graphics is expressed using the 'frame' of HTML.

4.5. SYSTEM FLOW

4.5.1. Whole system flow

The system flow is as follows, composing key technologies as previously stated and the Client/Server System.

4.5.2. Selecting the Attributes

We have developed the GUI based on Hyper Text Markup Language (HTML) using the 'form language' of HTML as previously stated. However, 'form language' has a drawback. It can call CGI and JavaScript directly, but it cannot call Java applets.

Therefore, we have used LiveConnect. It is a communication tool between Java applets and JavaScript. This is accomplished by calling JavaScript by 'form language', JavaScript then calls Java method which in turn allows Java applets to communicate with ‘form language’.

4.5.3. Communication database

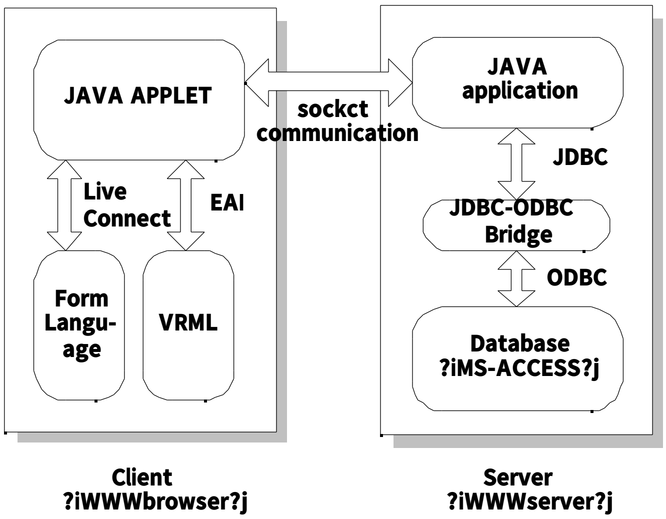

Figure 2. Figure caption.

Java applets called by JavaScript gets the value selected by 'form language' and sends the result to the server. Java is equipped with ‘socket class’ specifications; therefore, it is simple to develop a system with socket communication.

For security, Java applets communicate with only the server. This system has a WWW server and the Java applet is downloaded from this WWW server.

4.5.4. Search Database

The server searches the database according to the conditions sent by the client system and sends the results back to the client system. We have developed the server system by Java based on the same language of the client system. This enables communication of complicated data such as the class method.

Java supports JDBC interface for database access. However, the number of databases which supports JDBC is small. Therefore, we will use the ODBC-JDBC bridge. If the database supports ODBC, we can use the database. For this system we will use 'MS-Access' supported ODBC.

4.5.5. Presentation of Analysis

The results are presented in computer graphics using the browser, 'Cosmo Player'. As previously stated, VRML 2.0 enables access from an external program and can change the attributes of VRML format dynamically.

Therefore, it is not necessary to read VRML data whenever the attribute data is changed as the system realizes high speed expression. In this way the system can change the color of specific 3-D model buildings and the proportional 3-D graph of pedestrian traffic flow.

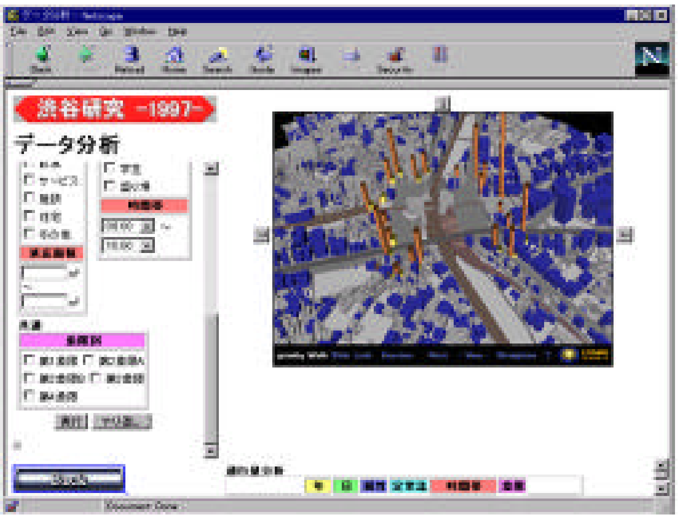

Figure 3. Office buildings and pedestrian traffic flow in the morning.

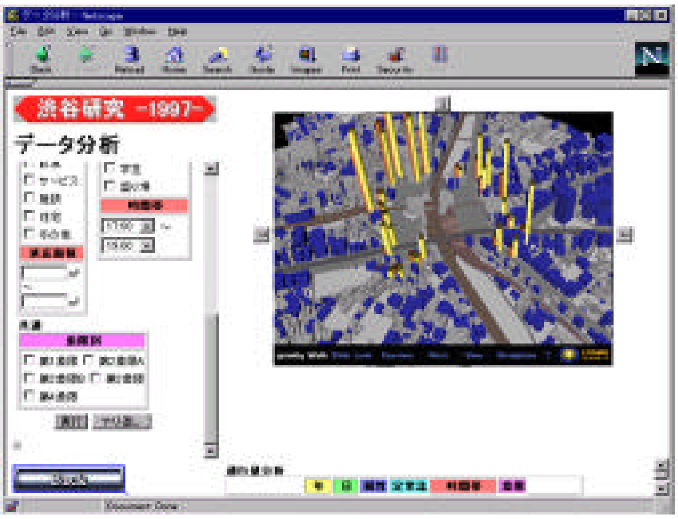

Figure 4. Office buildings and pedestrian traffic flow in the evening.

5. Evaluation

We used this system in the analysis of 2400 buildings. By using a PC with 133 MHz, a Pentium processor and 48M bytes of RAM, results were processed in approximately six minutes. Using a PC 133 MHz, a Pentium processor and 128M bytes of RAM, results were processed in approximately two minutes.

We found that the processing speed is dependent on the machine capacity of the WWW client.

6. Conclusion

In this research we have developed an interactive and easy to use system using new key technologies for the City Information Visualizer. This system is network adaptive, developed for cross-platform support, and low in cost. However, the calculation speed is not fast enough for an 'idea processor'.

The following points must still be addressed: First, an improvement of the VRML browser interface is necessary. On a PC it is impossible to browse in real-time, as the VRML accommodates 1.5M bytes of data. Second, the calculation speed is a problem as it is dependent on the amount of RAM available. We will reconsider the algorithm portion for calling the VRML data, etc....

References

Tokyu Research Institute: 1997, Shibuya Kenkyu 1997.
Japan Sun Microsystems (ed.): 1997, Java Database Connectivity.
Laura. L. and Michel M.:1997, JavaScript 1.1