Bridging Virtual Reality to Internet for Architecture

Field: Virtual Reality and Collaborative Team Work

Keywords: Virtual Reality, Architecture Models, Relational Database, Client-Server

Author:

J. Bradford, Associate Professor

W. S. Wong, M. Phil. Candidate

H. F. Tang, M. Phil. Candidate

Department of Architecture, University of Hong Kong

C. S. K. Yeung, Assistant Professor

Department of Electrical and Electronic Engineering , University of Hong Kong

Index

Abstract

This paper presents a virtual reality interface tool which allows a user to perform the following action :

1. Import design from other CAD tools.
2. Assemble an architecture structure from a library of pre-built blocks and geometry primitives

dynamically created by user.

3. Export the design interactively in VRML format back to the library for Internet browsing.

The geometry primitives include polygon, sphere, cone, cylinder and cube. The pre-built blocks consist of fundamental architecture models which have been categorized with architectural related style, physical properties and environmental attributes. Upon a user's request, the tool or the composer, has the ability to communicate with the library which indeed is a back-end distributed client-server database engine. The user may specify any combination of properties and attributes in the composer which will instantly bring up all matching 3-dimensional objects through the database engine. The database is designed in relational model and comes from the work of another research group.

The composer can modify the design in form of constraints, materials, lighting, etc. The design created or modified by the composer can be exported at any time to a collaborative design tool which is the research project of another research group. The export format is in VRML. Through this collaborative design tool, design groups around the world can review the design through the Internet simultaneously.

The composer is developed on a high-performance parallel processor-based VR platform. Advanced imaging features such as surface texturing and multiple light sources can be rendered in real-time.

1. Introduction

Traditionally, design teams are confined to groups who are working close to each other. It is very difficult to discuss the design work with other people without looking at the design, no matter in what form the design is. With the Internet, people can share ideas with others miles away. Discussing and debating topics with people across the world is common in the Internet. A design group scattered around the world is now able to discuss their project, with complete layouts, sketches and computer generated 3D view, with each other.

Static images can be transmitted through the Internet easily. However, sometimes it may be inadequate to discuss a design project over a static image. Real time change of the 3D model is a better way to express ideas in some cases. Our project is to develop a composer, with functions mainly to enhance the 3D model of the design, to assist a design group in displaying their ideas.

The composer allows the user to dynamically modify the design. It has functions that aim at enhancing the design. It can export the design in VRML format to a specific web page where a chat area for discussion, a VRML display area to display the design, and the hierarchy of the database where the design is stored in, are shown. The composer is developed on the Reality Engine supercomputer and is based on a commercial VR software, dVISE and dVS from Division Ltd.

2. Motivation

Begin in 1993, Department of Architecture of the University of Hong Kong has organized a joint design project with other universities over the world every year. This joint design project is called the Virtual Design Studio (VDS). I have participated in the VDS′ 96. Several problems are encountered. To be able to discuss the design ideas with the overseas partners, we connect a network video program with other universities on Hong Kong Time 0800. The performance of the network video is unsatisfactory in both resolution and smoothness. We have to resort to use e-mail to discuss ideas and ftp to transfer graphics files back and forth. Not much improvement compared to previous VDS.

A place where participants can gather and discuss with each other, where the design can be shown, where the setup and access cost is reasonable, is the most ideal. With a database where participants can store their design, as well as retrieving building blocks to enhance their design is also desirable. Our part is to develop a program for a user to dynamically modify the design and share it with others.

3. Functions

The composer is based on a commercial VR software which has a graphical user interface (GUI) option when the computer monitor and mouse is used instead of the head-mounted display and 3D mouse. Basic functions such as move, rotate, and scale are presented in the GUI, as well as translation and rotation constraints, attributes modification, lighting, etc. The functions added to the composer are used to enhance the design in virtual reality. They are :

. Stub Setting
b. Scale Constraints
c. Gluing of Objects
d. Relative Orientation Adjustment

3.1 Stub Setting

The VR software provides translation, rotation and scaling grid setting for individual object. Only the object that has been set will be affected by the grid value. The stub setting we provided is a global setting. All objects will be snapped to nearest multiple of the stub value after they have been moved.

3.2 Scale Constraints

This set of constraints is to limit the scaling of an object. With scale constraint in any direction set on an object, that object cannot be scaled in that direction. The scale factor of that direction is fixed. This set of constraints does not have any effect normally, as we cannot scale an object directly. It is useful when scale factor in a certain direction of a part of a glued object is supposed to be invariant. For example, a column with special decorations around the top and bottom ends. Both ends shall be invariant in scale factor in vertical direction. When the column is scaled in vertical direction, the decorations will remain the same.

3.3 Gluing of Object

This function reformats the object hierarchy such that two or more selected objects will be linked under a dummy object, which is linked under the parent of the first selected object. After the gluing is done, all selected objects are of the same level in hierarchy. The GUI has already provided LINK and UNLINK functions to manipulate the hierarchy, and these two functions create (or break) a parent-child relationship between two objects. With a dummy parent, if any one of the children was moved, rotated, or scaled, all the siblings would also be moved, rotated and scaled, with the selected object as origin of operation. In some cases, we want to maintain the objects to be separated but to be translated, rotated or scaled together. For example, we want the walls, the floor and the roof of a house to be as separate objects (for applying texture, constraints, etc.) but when we modify the coordinates of any of these, the others' coordinates will also be modified.

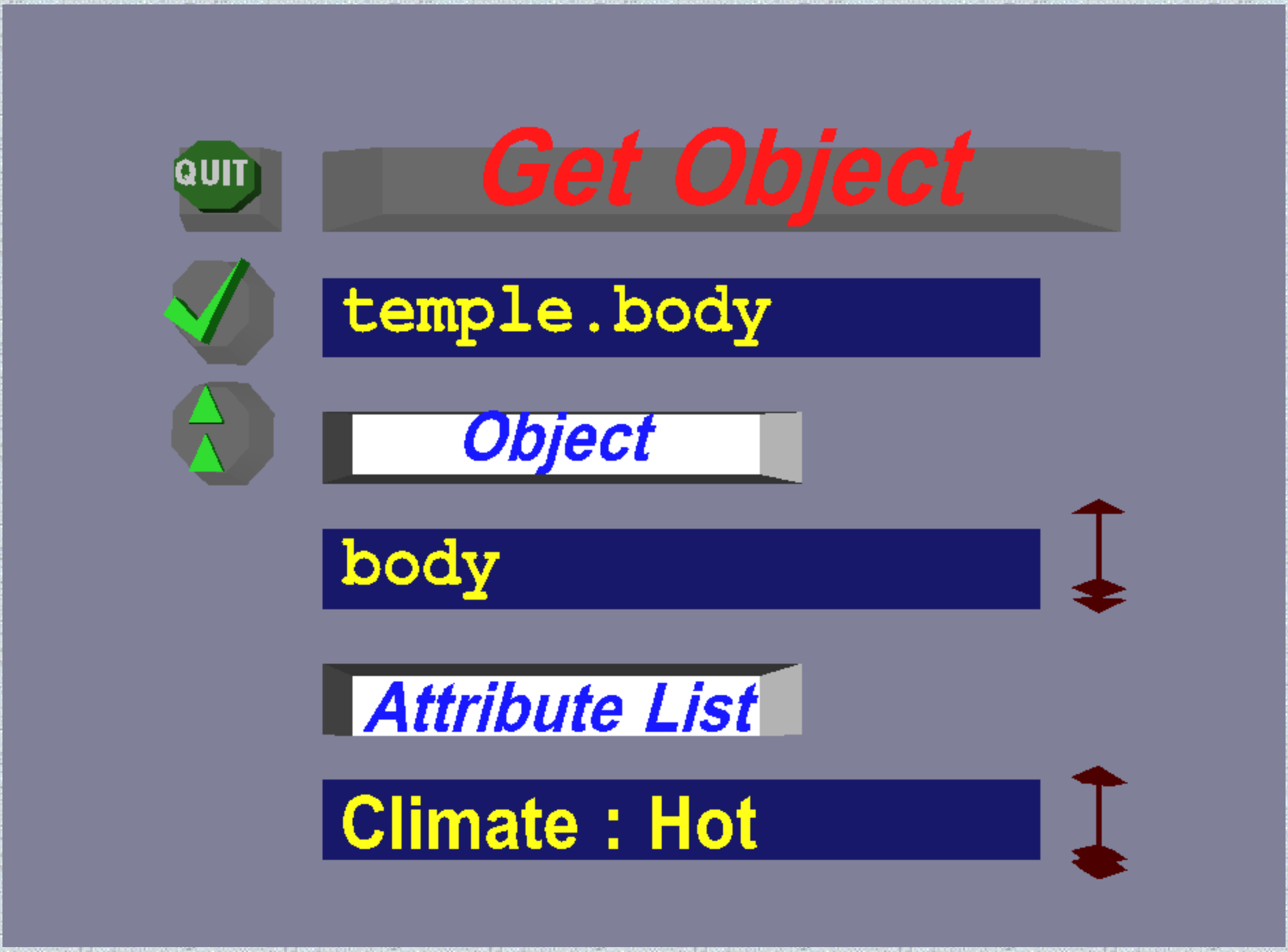
3.4 Relative Orientation Adjustment

This function rotates and translates (if necessary) an object such that one of its patch surface is in contact with another object, the base. The object appears to be resting on the base. Although the GUI can rotate any object precisely, sometimes we do not know how much we should rotate an object. Assumptions are made to make the task easier. Both the object and the base are convex. The base is assumed to be large, comparing with the object. The user has already moved the object to the location near the base.

---

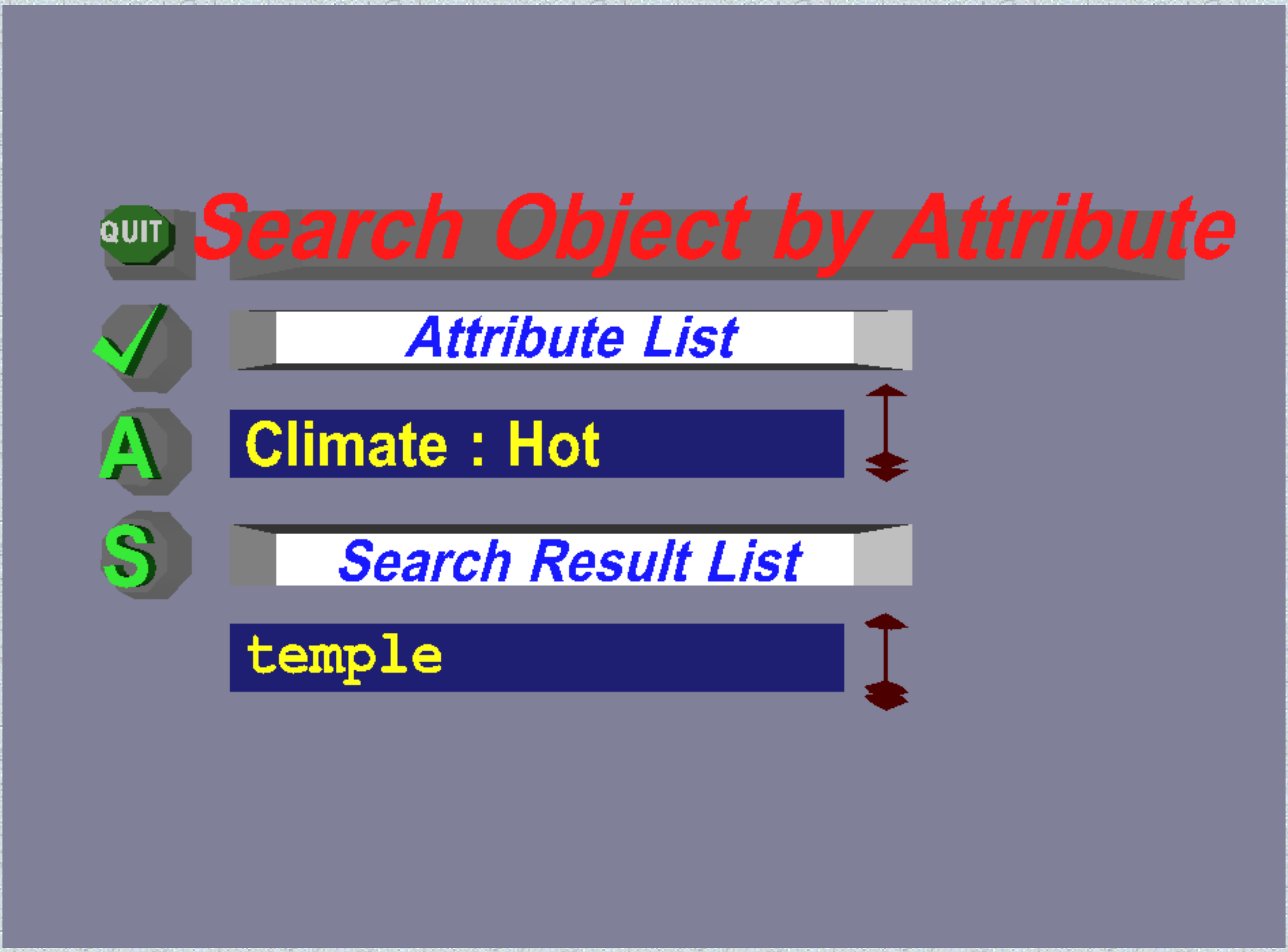
4. Object Inquiry

User can extract pre-defined object models from a object database through the toolbox interface. The user can choose an object and then browse into the hierarchy of that object. There are components of the object, or different type of that object in the hierarchy. When he or she has made decision, that object model will be sent to the workspace and the user can make further adjustment like scaling or translating.



With the toolbox named "Get Object", the user can select objects inside the distributed database. The user can choose an available object under the item named "Object" by using the slider beside, and the item named "History" stores the object hierarchy. That object hierarchy shows how the objects are organized inside the database. For example, when the user choose <temple> at first, he or she can then choose <roof>, <body>, <platform>, <decoration>, <finish>, and <fixture>. He or she then selects roof. The history list will display <temple.roof>. Inside roof, the user can get classification of different roofs. A user can get one of them and the database will send that corresponding file to the workspace.

In the "Get Object" toolbox, the last entry displays the attributes of the object. Here, a user can only examine what attributes an object inherits. If a user wants to retrieve a particular building with certain attributes, the user has to use another toolbox, the "Search Object by Attribute" toolbox.



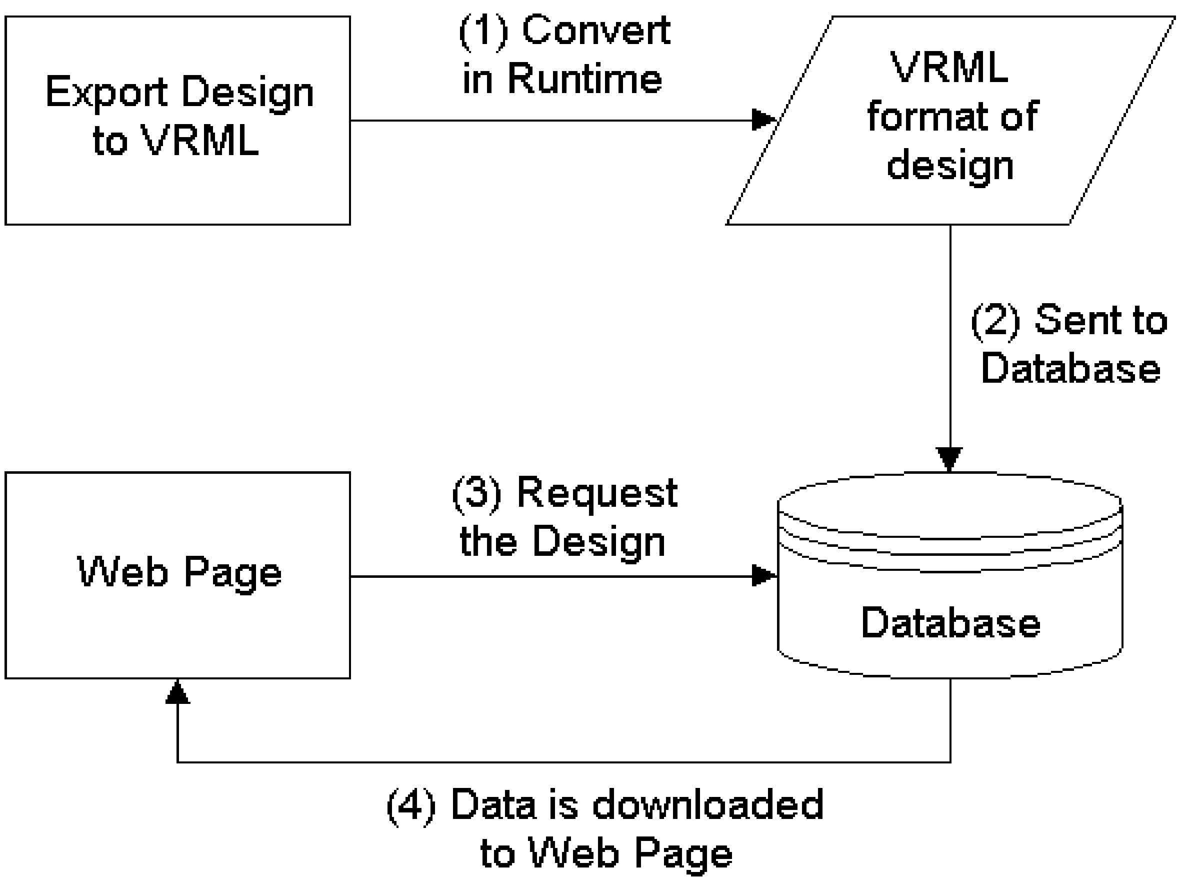
The attributes are classified into five categories, namely time period, climate, geographical region, function and civilization. Time period shows the time of the appearance of that object, for examples, Ancient, 1800-1900, Ming Dynasty, Modern. Climate describes that the object is suitable in hot, cold, rainy, sunny, snowy, windy, dry, or moist weather. Geographical region states the geographical location of the appearance of that object, for examples, mountain range, plain, different area (culturally or globally) etc. Function indicates the usage of that object like administration, learning, religion, etc. Civilization states the civilization background of that object, for example, Greek, Roman, Chinese, Indian, etc. A user does not need to input all attributes for selection of objects. The user can select any one of matching objects. Only the objects that have matched all the input attributes can be selected. At present, there are some pre-defined values stored in the selection list of each attribute in order to enable user to select.

The button "A" is used to call up the attribute selection toolbox. The "S" button is used to start searching the matching object. The "Attribute List" displays the selected attributes. The "Search Result List" displays the currently selected matching object. The object will only be downloaded from the database if the "tick" button is pressed. Using this method can only download a whole building from the database, not an individual part of a building.

5. Design Export

The export of design to VRML will be invoked only in a specific toolbox. We can only export the design when the composer is running. There is actually no converter from the saved data files of the composer to VRML.

A data entry will be created in the database. The computer where the VRML file is located in, path name for the VRML file, filename of the VRML file and the login name of the user who exports the design will be recorded in the data entry. It can be accessed through the database. All embedded VRML files are copied to the same directory for easy access.



The composer is based on the commercial software dVS and dVISE from Division Ltd. It has its own data structures and formats. To convert to VRML format, only those attributes present in both formats will be converted. The following table shows conversion relation in the design export.

Data Attributes	Composer Data Format	VRML Corresponding Data Format
Boundary	BBOX	Collision : bboxCenter Collision : bboxSize
Level of Detail	DISTANCE : in , out	range[ ]
	reference	center
Geogroup	Vertex	Coordinate
	Vertex : NORMAL	Normal
	Vertex : TEXTURE	TextureCoordinate
Geometry	SPHERELIST	Sphere
	POINTLIST	Coordinate
	LINE	IndexedLineSet
	PMESH : CONNECTIONLIST	IndexedFaceSet

Material	AMBIENT	ambientIntensity
	DIFFUSE	diffuseColor
	SPECULAR	specularColor + shininess
	EMISSITY	emissiveColor
	OPACITY	transparency
Texture	MAP	ImageTexture : url
	WRAP_U	repeatS = TRUE
	WRAP_V	repeatT = TRUE
	WRAP : REPEAT	Both repeatS and repeatT are TRUE
Collision (priority higher than Boundary)	Bbox	Collision : bboxCenter Collision : bboxSize
Light	State : On , Off	on : TRUE , FALSE
	Color	color, ambientIntensity, intensity
	Translation of the light object	location (if applicable)
	Orientation of the light object	direction
	Cone : theta, exponent	cutoffAngle, beamWidth (if applicable)
	Type : Directional, Spot, Point	DirectLight, SpotLight, PointLight
Fog	fogColor	color
	mode	fogType
	farFog	visibilityRange
Navigation information	farClip	NavigationInfo : visibilityLimit
Background	background <color>	Background : skyColor
Visual	Geometry	Inline : url
	Billboard	Billboard : with children = Inline
	FrontMaterial	Check for <Material>

	LOD	Check for <Level of Detail>
Transformation	Position	Transform : translation
	Orientation	Transform : rotation
	Scale	Transform : scale

The original data format supports many zones (different regions) in a single world. The contents of all zones are read and converted into a single world in VRML. The unit of the converted design is meters. Since the entry SCALE appears in several locations, we have to be careful about applying the SCALE data to which objects. Special EVENT that is associated with specified object will not be converted.

Some important attributes that exist in the data format of our composer will be lost after conversion. One of them is the constraint attribute. There are no constraint attributes in VRML format. All the lost information will be recorded in the VRML files as comment for references. Some of the attributes are very difficult to convert. For example, *keyframe* (motion or animation) of object is such an attribute. VRML supports motion or animation of objects in a series of Interpolators nodes. However, the difference between two kinds of data format is very large. We have left out the conversion of *keyframe* in this version.

We can create primitive in the composer. The primitives are regular polygon, cube, sphere, cone and cylinder. Except sphere, all other primitives are actually built from mesh, a geometry data format used by the commercial software (*pmesh : connectionlist* in the above table). However, in VRML, there are geometry nodes for cone, cylinder and cube (or rectangular box). To reduce conversion effort, if a cube, cone or cylinder is detected during conversion, the corresponding geometry nodes in VRML will be used.

6. Conclusion

In the paper, a virtual reality interface tool, the VR composer, is presented. The composer can be used to modify 3D designs in real time. The interface makes use of a computer monitor and a normal mouse. We assume that after a design is imported into the VR software, the general shape and design are nearly finished. The composer can improve a design to make it more realistic, by applying material, texture, attributes, behavior, etc., to objects in the design. VRML format of the design can inherit most of the properties added, and it is a standard in the Internet now. The result is, over the Internet, people can inspect a realistic model of a design, express their opinions, and discuss with the designers and other people.

7. Reference

Bradford, J. , Wong W.S., Tang A.H.F. : 1997, CAADRIA'97 : *A Virtual-Reality Building Block Composer for Architecture*

Jerzy Wojtowicz (ed) : 1995, Virtual Design Studio, Hong Kong University Press

Green, M. and Liang, J. : 1994, JDCAD: A Highly Interactive 3D Modeling System, *Computers & Graphics*, **18** (4), 499-506

Carlson, W. E. : 1982, *Techniques for the Generation of Three Dimensional Data for Use in Complex Image Synthesis*, The Ohio State University

ISO/IEC CD 14772 : 1996, *The Virtual Reality Modeling Language Specification* Version 2.0

Tang, H. C., Ho, K. C., Li, W. L., Ho, C. M. : 1983, *Chinese Temples on Hong Kong Island*, Center of Asian Studies, University of Hong Kong

Maia Engeli, David Kurmann : 1996, Spatial Objects and Intelligent Agents ina Virtual Environment, *Automation in Construction*, **5** (3), 141-150

Division Ltd. : 1996, *dVISE© and dVS© for Unix Workstation*.

Hughes Technologies Pty Ltd. : 1996, *Mini SQL© , a Lightweight Database Engine*.