CaveCAD: Architectural Design in the CAVE

Cathleen E. Hughes* Lelin Zhang Jürgen P. Schulze† Eve Edelstein Eduardo Macagno

University of California San Diego, Calit2, La Jolla, CA, USA

Abstract

CaveCAD is our in-house developed 3D modeling tool, which runs in immersive virtual reality environments, such as CAVEs. We built it from the ground up, in collaboration with architects, to explore how immersive 3D interaction systems can support 3D modeling tasks. CaveCAD offers typical 3D modeling functions, such as geometry creation, modification of existing geometry, assignment of surface materials and textures, the use of libraries of 3D components, geographical placement functions, and shadows. CaveCAD goes beyond traditional 3D modeling tools by utilizing direct 3D interaction methods. We evaluated our modeling system by running a small pilot study with four participants: two novice users and two expert users were tasked to build Disney World's magic castle.

Index Terms: H.5.2 [Information Interfaces and Presentation]: User Interfaces—Graphical user interfaces (GUI); J.6 [Computer Applications]: Computer-Aided Engineering—Computer-aided design (CAD)

1 Introduction

CaveCAD was designed as a software package for computer-aided design that provides architectural designers with the ability to design within 3D immersive virtual reality environments. Compared with existing software applications like Autodesk 3dsMax, Maya, AutoCAD and Google SketchUp that normally run on flat 2D screens with a limited field of view, CaveCAD enables designers to gain a pure 3D experience in human-scale modeling, utilizing the stereo rendering facilities of our StarCAVE [3]. Competing approaches, which allow running an existing 3D modeling tool, such as the ones listed above, on a virtual reality system by intercepting OpenGL calls, such as Conduit [2] or TechViz [6], do not allow 3D interaction with the geometry, but instead merely provide a way to view the 3D model immersively. Another distinctive feature CaveCAD provides is a more intuitive approach to the design process by replacing panels of buttons and flat list-style menus with interactive multi-functional displays of design instructions and options, which allow using the software without intensive training and to build blueprints from scratch in a quick and convenient way. CaveCAD also incorporates a set of tools to increase rendering realism, for instance by creating real-time shadows based on user-defined settings for location, date and time.

2 Related Work

A number of 3D modeling systems for virtual environments have been created in the past. Some of them are geared towards freeform drawing, such as Brown university's CAVE Painting project [5]. Others use 3D sculpting approaches, in which a solid object gets modified with freeform 3D operations, such as Galyean et al. [4]. While CaveCAD is geared towards cluster-based VR systems, others have been implemented with head mounted displays in mind, such as Butterworth et al.'s [1]. Existing virtual reality toolkits, such as WorldViz [7], are designed to be used to implement VR applications, similar to CalVR, but they do not readily come with 3D modeling applications.

*e-mail:cehughes@eng.ucsd.edu

†e-mail:jschulze@ucsd.edu

3 Software Design

CaveCAD was written as a module for CalVR, our in-house middleware for cluster-based virtual reality applications. CalVR was written in C++ and uses the OpenSceneGraph (OSG) library to manage its visual output and interactions. In a PC cluster, each node runs an identical copy of CalVR, synchronization happens by synchronizing the input events.

4 Modeling Functions

At the core of CaveCAD is a set of functions to create 3D geometry.

CaveCAD defines geometric objects by grouping together a number of geometry faces. For example, a box is defined as a group of six rectangular faces. Each face is stored in the scene graph as its own geometry node object, then these faces are combined into a group. Each of the faces can be manipulated independently to modify the shape of the 3D object, while CaveCAD ensures that the object's surface remains closed. Each group shares a material definition entry which defines its color and texture.

For example, a user can select a box and perform a manipulation action (move, scale, rotate, or copy) on the object as a whole, or select a single face or subset of faces and manipulate just those faces. Users can also select multiple objects to operate on them as a group, but this type of grouping is temporary.

5 Support Functions

In order to make 3D modeling easier and provide greater visual realism, CaveCAD offers the following support functions.

5.1 Ground Texture

Architects often already have a floor plan or aerial image of the site on which a building is to be designed. In CaveCAD, such images can be loaded from disk and placed on the ground plane in front of the user. In our contest experiment, the users were placed on a default stone tile texture to fit the theme of castles.

5.2 Pre-Defined Geometry

In order to allow an architect to quickly add standard nature elements (trees, bushes, flowers), people or movable objects (furniture, cars), CaveCAD gives the user access to a library of 3D models. These models have to reside on local disk, but could have been fetched from 3D model servers. From the menu the user can select an object, which will show a copy of it in the 3D scene. In the Magic Castle trials, we offered the users a few different types of trees to populate the landscape around the castle.

5.3 Shadows

We use OSG's shadow toolkit to render real-time shadows, which the built geometry casts onto itself and the surrounding environment. The notion of a sun provides the position of the light source. CaveCAD has a menu item, which allows setting time of day, date, and geo-location, from which it computes the location of the sun, and thus the appearance of the shadows.

6 User Interface

The main user interface widgets in CaveCAD are a number of differently colored and textured spheres, some of which contain icons inside of them. Interaction with these spheres happens with a hand-held 3D wand, which sends a virtual laser beam into the 3D scene. Interaction with menu widgets or existing geometry happens by intersecting the laser beam with them and clicking a button on the wand.

The sphere menu is loaded at startup at a pre-configured position, with the position manually configured depending on the setup of the screens. In the future, the menu will be moveable and hideable.

There are six different sphere widgets.

New 3D shape: allows the creation of boxes, cylinders and cones.

Set material: colors and textures existing geometry by pulling one of the textured sample spheres onto the 3D object.

Set location, time and date: 3D widgets allow setting these parameters visually.

Set ground plane: uses the visual effect of a spiral-bound notebook to select a ground texture, which can then be applied by dragging it to the ground.

Background: allows setting a panoramic background texture for the scene.

Load pre-defined 3D shape: allows loading ready-made 3D objects, such as trees, from disk into the scene.

During the creation and manipulation of geometry, the user can adjust the unit size of their movement. Using the joystick on the wand or scroll wheel on the mouse, snapping unit sizes can be moved up and down, from 2 inches to 20 feet (1 degree to 30 degrees for rotation). This allows for creation and editing of objects on different scales with the same amount of hand and arm movement. CaveCAD also offers a snapping option, which allows snapping corners of newly created geometry to existing geometry.

7 Hardware Environment

We ran our user trials in the StarCAVE, a 15-screen rear-projected surround virtual environment with 34 HD passive stereo projectors, 18 graphics PCs and an ART optical tracking system with four cameras and a wireless wand (Flystick2).

8 Pilot Study

For the pilot study, we gathered four users who volunteered to attempt to build the Magic Castle in CaveCAD in fifteen minutes. The users consisted of the current programmer of CaveCAD and three other visualization programmers. Of the four, only two (including the CaveCAD programmer) had previous experience in 3D modeling. They had all worked in the StarCAVE before and were familiar with 3D interaction methods. They were each given 20 minute training sessions in using CaveCAD, then given 15 minutes to build the Magic Castle from a picture provided. Afterwards, they were provided with a short questionnaire.

We found that the two users who had no experience in 3D modeling were unable to create a castle without a much longer training period. While they understood the CaveCAD interface, they were not familiar with the 3D modeling concepts necessary to build a complex structure from scratch within the timeframe of fifteen minutes. They were able to build and texture a simple house within the time limit.

The two who had previous 3D modeling experience were able to build basic castles with textures within 15 minutes (see Figure 1), although they reported having to work very swiftly and omit details such as windows. The users requested an undo/redo feature most frequently. Additionally, when asked about the difference in modeling at a desktop compared to modeling in 3D, we found users arms tired from manipulating the wand for extended periods of time.

We found that our interface relies on previous knowledge of 3D modeling to be successful. While the interface itself has a minimal number of buttons, using them successfully depends on experience in manipulating geometry in 3D. In order to be more successful in building the Magic Castle in 15 minutes, we hypothesize that more directed, less freeform tools are needed. For a user looking to create environments quickly, CaveCAD may provide too much direct control over the geometry.

Figure 1: Magic castle created by expert user during 15 min trial.

9 Conclusion and Future Work

We presented a novel implementation of an architectural design software system named CaveCAD. Our pilot study showed that CaveCAD is intuitive enough for a user with prior, desktop-based 3D modeling experience to quickly learn it and successfully create 3D models with it.

In the future, we would like to add a variety of features to make CaveCAD more useful. At the top of the list is an undo/redo function, which was requested by some of the trial users. Furthermore, we would like to explore supporting two-handed interaction, as well as gesture-based control of the environment, possibly with an RGB+D device such as the Microsoft Kinect.

Acknowledgements

This work was supported in part by HMC Architects and the Academy of Neuroscience for Architecture (ANFA).

References

[1] J. Butterworth, A. Davidson, S. Hench, and M. T. Olano. 3dm: a three dimensional modeler using a head-mounted display. In *Proceedings of the 1992 symposium on Interactive 3D graphics*, I3D '92, pages 135–138, New York, NY, USA, 1992. ACM.

[2] Conduit. Conduit home page on mechdyne web site, *http* : *//www.mechdyne.com/conduit.aspx/*. 2013.

[3] T. DeFanti, G. Dawe, D. Sandin, J. Schulze, P. Otto, J. Girado, F. Kuester, L. Smarr, and R. Rao. The starcave, a third-generation cave and virtual reality optiportal. *Future Generation Computer Systems*, 25(2):169–178, Feb. 2009.

[4] T. A. Galyean and J. F. Hughes. Sculpting: an interactive volumetric modeling technique. *SIGGRAPH Comput. Graph.*, 25(4):267–274, July 1991.

[5] D. F. Keefe, D. A. Feliz, T. Moscovich, D. H. Laidlaw, and J. J. LaViola, Jr. Cavepainting: a fully immersive 3d artistic medium and interactive experience. In *Proceedings of the 2001 symposium on Interactive 3D graphics*, I3D '01, pages 85–93, New York, NY, USA, 2001. ACM.

[6] TechViz. Techviz home page, *http* : *//www.techviz.net/*. 2013.

[7] WorldViz. Worldviz home page, *http* : *//www.worldviz.com/*. 2012.