

CaveCAD: A Tool for Architectural Design in Immersive Virtual Environments

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

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

ABSTRACT

Existing 3D modeling tools were designed to run on desktop computers with monitor, keyboard and mouse. To make 3D modeling possible with mouse and keyboard, many 3D interactions, such as point placement or translations of geometry, had to be mapped to the 2D parameter space of the mouse, possibly supported by mouse buttons or keyboard keys. We hypothesize that had the designers of these existing systems had been able to assume immersive virtual reality systems as their target platforms, they would have been able to design 3D interactions much more intuitively. In collaboration with professional architects, we created a simple, but complete 3D modeling tool for virtual environments from the ground up and use direct 3D interaction wherever possible and adequate. In this publication, we present our approaches for interactions for typical 3D modeling functions, such as geometry creation, modification of existing geometry, and assignment of surface materials. We also discuss preliminary user experiences with this system.

Keywords: 3D Modeling, Immersive Virtual Reality, Interaction.

1. INTRODUCTION

CaveCAD is a software package for computer-aided design that provides architectural designers with the ability to design within immersive 3D virtual reality environments. Compared with existing software packets, like Autodesk 3dsMax, Maya, AutoCAD and Trimble SketchUp, which normally run on regular monitors 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 a 360-degree surround immersive environment with precise optical tracking for head and input device to allow intuitive navigation and user interaction. Another distinctive feature CaveCAD provides is an intuitive approach to the design process by replacing panels of buttons and 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.

Our goal with CaveCAD was to implement some of the most basic 3D modeling functions, such as creating standard 3D shapes like boxes or cylinders, and modifying them by changing their shape and location. Instead of building on interaction paradigms used by existing desktop-based 3D modeling tools, we set out to fully utilize the 3D input device, the surround view, and the fact that VR can display objects at a human scale.

To interact with the system, we created a spherical menu without pull down menus or buttons. Each category of functions is visually represented within the interaction sphere, which we tried to make as self-explanatory as possible. Geometry in CaveCAD is organized in similar formats to those of existing commercial packages: models can be created, modified, colored or textured in either 'group mode' with large groups of shapes, or in 'geometry mode' with individual vertices, edges or surfaces. With respect to geometry that was created within CaveCAD and is fully modifiable, CaveCAD supports loading and limited manipulation of 3D model files in any data format our rendering library OpenSceneGraph supports (including, for example, OBJ and VRML). Other CaveCAD features include a camera viewpoint manager, location specification, and time of day-dependent shadows.

CaveCAD has been designed with an immersive visualization system in mind, and was implemented for the StarCAVE at UC San Diego. The StarCAVE is a 30 megapixel per eye, 360-degree surround VR system with 34 HD projectors, 18 rendering PCs, and 10 Gigabit networking. It was implemented for the CalVR middleware software,¹⁴ which provides support for graphics clusters, 3D displays and 3D input devices. Architects have used the StarCAVE in collaboration

Contact e-mail: jschulze@ucsd.edu (Jürgen Schulze)

with neuroscientists as an interactive environment to measure human responses to architecture by recording EOG or EEG signals for quantitative analysis,⁴ wayfinding,⁷ as well as ocular tracking.¹⁹ A surround sound system has been used to simulate the acoustic behavior of architectural spaces.²⁰ CaveCAD has been used at the intersection of these disciplines, as a research tool to assess the power of immersive virtual reality for architects.

2. RELATED WORK

Brown university's CAVE Painting project⁹ and Deering's² work were geared towards artists and focus on freeform drawing, but do not allow creating and modifying geometric shapes.

Other groups, such as Galyean et al.⁶ use 3D sculpting approaches, in which a solid objects gets modified with freeform 3D operations. This is more related to working with clay rather than designing CAD models. Sachs et al.'s¹³ 3-Draw system uses an approach in which the user draws 3D curves with a 3D input device. These curves are then used to define the skeleton of a surface.

Butterworth et al.¹ created a 3D modeling system with a similar goal as CaveCAD, but it was designed for head mounted displays instead of cluster-based VR systems. In 1994, Liang and Green¹⁰ presented their JDCAD modeler for VR systems. JDCAD uses a constructive solid geometry (CSG) approach for creating 3D models. Our approach is more intuitive, as it allows modifying existing geometry by directly manipulating its vertices, edges or faces.

Zeleznik et al.¹⁸ presented an innovative sketching-based 3D modeling approach, but it requires learning a set of 2D gestures. Drawing is done with a 2D mouse, not directly in 3D.

WorldViz,¹⁷ COVISE,¹² EON Studio⁵ are 3D virtual reality toolkits, which can be used to create VR applications and display ready-made 3D models, but they do not serve as 3D modeling tools.

Commercially available 3D modeling tools, such as Autodesk 3dsMax, Maya, AutoCAD, Revit or Google SketchUp, were designed for desktop computers with 2D displays and do not support virtual environments, except when run through TechViz¹⁶ or Mechdyne's Conduit,¹¹ which intercept their OpenGL calls and distribute them to a visualization cluster, while managing the camera independently from the desktop tool. But even in this combination, they do not allow 3D interaction with the geometry while in the VR environment, except when specific hooks are put in the desktop application to accept input from the VR system, which is rarely done.

To the best of our knowledge, there does not exist any comparable approach to CaveCAD, which would allow 3D modeling and the creation of architectural models directly in a virtual environment.

CaveCAD was an official entry in the 2013 IEEE 3DUI Symposium's user interface contest.⁸

3. CAVECAD SOFTWARE DESIGN

CaveCAD was built as a module for CalVR, our in-house virtual reality renderer, which can run on a PC cluster in Linux, or standalone on Windows or Apple computers. CalVR abstracts the rendering hardware and user interaction handling from the programmers so that they can focus on the logistics of the application itself. It is written in C++ and uses the OpenSceneGraph (OSG) library to manage its visual output and interactions. On a PC cluster, each node runs an identical copy of CalVR, synchronization happens by synchronizing the user input events.

3.1 Geometry Definition

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 edge, or subset of faces and edges, and manipulate just those. Users can also select multiple objects to operate on them as a group.

4. 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 video clip (Video 1) referenced at the end of the paper shows the CaveCAD application running in the StarCAVE. The sphere menu is loaded at startup at a pre-configured position, with the position manually configured depending on the setup of the screens. Figure 1 shows the main menu.

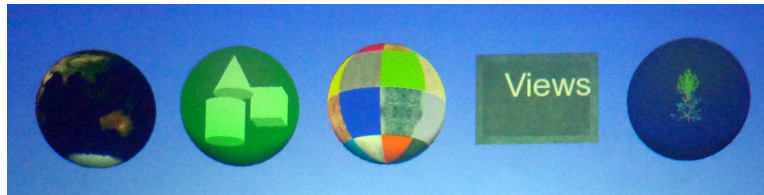


Figure 1. The main sphere menu, which is always visible.

The main menu contains the following menu items:

Environment: Contains 3D widgets to allow setting time, date, panorama image and floor plan.

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

Material: colors and textures existing geometry by pulling one of the textured sample spheres onto the object.

View points: allows saving and restoring view points.

3D Models: allows loading ready-made 3D objects, such as trees, from disk into the scene.

During creation and manipulation of geometry, the user can adjust a scale factor for the size of the created object: the distance by which the cursor is moved will be multiplied by this factor to allow for creation of objects on a very large scale with the same amount of arm movement. CaveCAD also offers a snapping option, which allows snapping corners of newly created geometry to existing geometry.

4.1 Manipulating Shapes

To create a new shape, the user clicks on the sphere icon of the shape they wish to make. They then either click with the ground or with an existing piece of geometry to select where they wish to place the shape. This will correspond to the corner of a box or the center of a cylinder or cone. The user now drags the wand to control the shape and size of the newly created geometry, see Figure 2. Movement in all directions is allowed, so the selected point of creation can correspond to any corner of a box or the top or bottom of a cylinder or cone. To complete creation, the user releases the button. To create another shape, they will need to click the shape icon again. If the user wishes to cancel the geometry creation after clicking on an icon, they click on the sky, though this solution currently does not work if a user is inside a piece of geometry with no line of sight to the sky.

To edit geometry, the user must have the geometry edit menu activated. The user clicks on a piece of geometry they wish to edit, and a virtual geometry manipulator (see Figure 3) appears in front of them which contains miniature versions of the selected geometry. They can continue clicking on existing geometry to edit all the selected pieces as a temporary group. To edit a single face or edge, or subset of faces or edges, the user clicks the miniature geometry contained within the virtual geometry manipulator on the desired face or edge.

Contained within the virtual geometry manipulator are tools for moving, scaling, rotating, and copying. To move, arrows are presented in six directions which the user clicks on to drag along the world axes. To scale, small boxes at the corners are clicked and dragged to scale along a particular axis, as well as a box at the center to scale uniformly along all axes. To rotate, protractors are presented in three directions which can be clicked and dragged. Finally, to copy, double arrows allow for interaction identical to movement, except the user is moving a copy of the selected geometry. The user switches between these tools using the joystick on the wand or scroll wheel on the mouse. To cancel editing, the user clicks on the ground or sky.

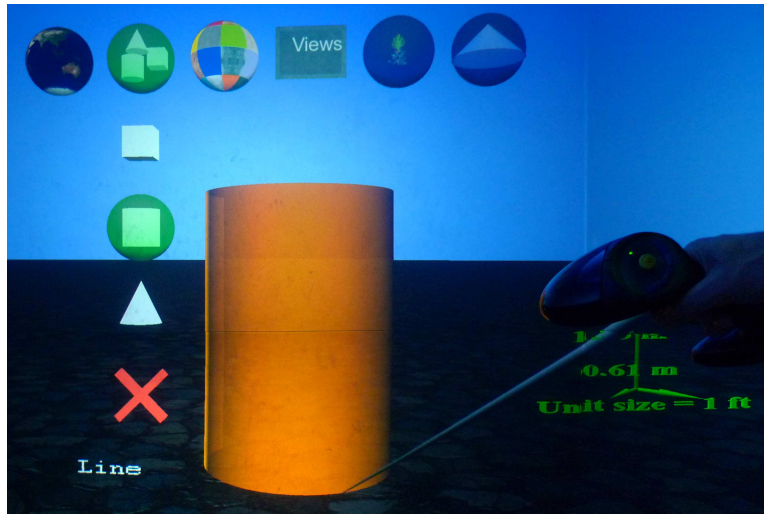


Figure 2. Creating a cylinder.

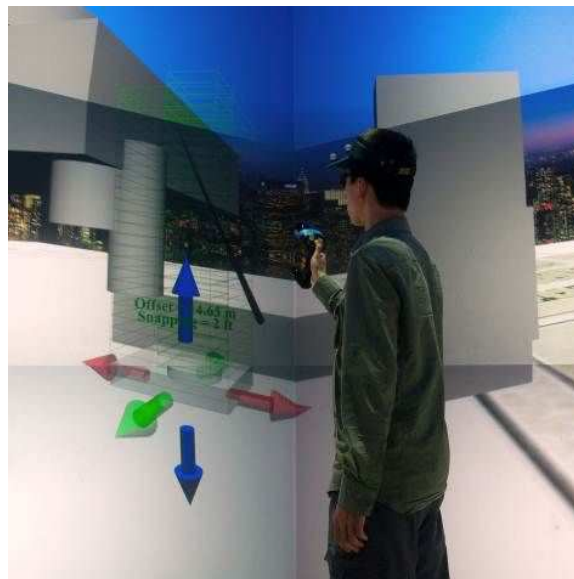


Figure 3. The shape manipulation interactor.

The virtual geometry manipulator is particularly useful for the creation and manipulation of large pieces of geometry, such as solid walls, columns and roofs, which are hard to work with in CaveCAD otherwise, because it displays the virtual world at 1:1 scale. The virtual geometry manipulator uses ideas from the World-in-Miniature¹⁵ approach, but applies them only to the selected shape, and does not use it to move the camera, but it does to interact with the object.

4.2 Materials: Color and Texture

Materials are selected by first clicking on the materials sphere. This creates a number of spheres representing textures or colors that can be selected to paint a shape with, see Figure 4. Mapping a color or texture to an object is done by clicking on the texture sphere, dragging the mouse and releasing the button on the shape that receives the material.

In addition to using preset colors, the user can open a tool for selecting and saving a custom color. A selection ball is dragged around a plane to select the hue, then up or down on two cones to select the brightness. The user clicks the save icon to add their custom color to the selection of colors in the menu.

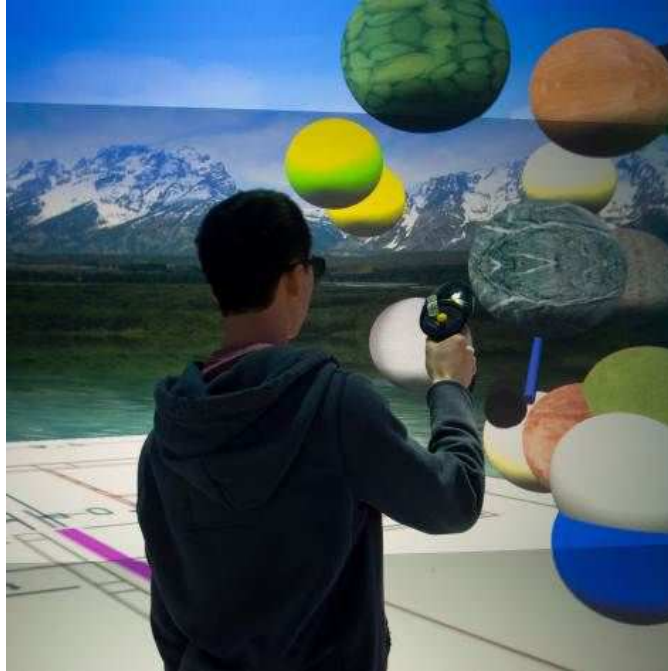


Figure 4. Selecting a color or texture with the sphere UI.

4.3 Background Panorama

Sometimes a 3D model, especially when it is a building, is best displayed in the context of what is located around it. For this reason, we allow importing panoramic photographs and draping onto the inside of a cylinder to be visible in the distance. To activate a panorama image, the user clicks the respective image in the panorama selection panel and drags it to the approximate place it should go (see Figure 5)

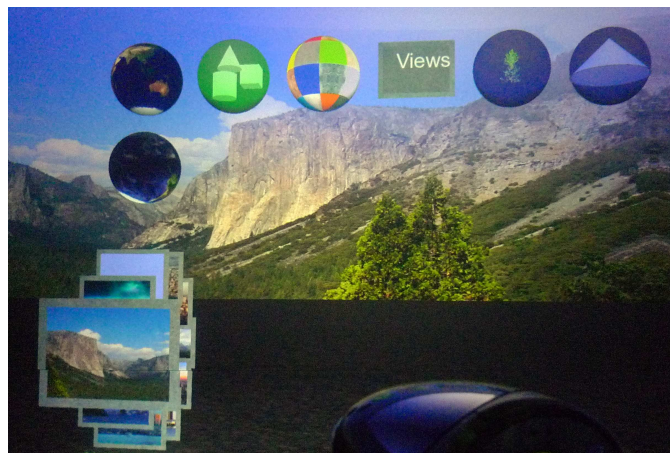


Figure 5. Selecting a panorama image.

4.4 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, see Figure 6. In our context experiment, the users were placed on a default stone tile texture to fit the theme of castles.



Figure 6. Selecting a floor plan image.

4.5 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, see Figure 7. In the Magic Castle trials, we offered the users a few different types of trees to populate the landscape around the castle.



Figure 7. Loading a 3D model from disk.

4.6 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.

4.7 Geo-Location and Time

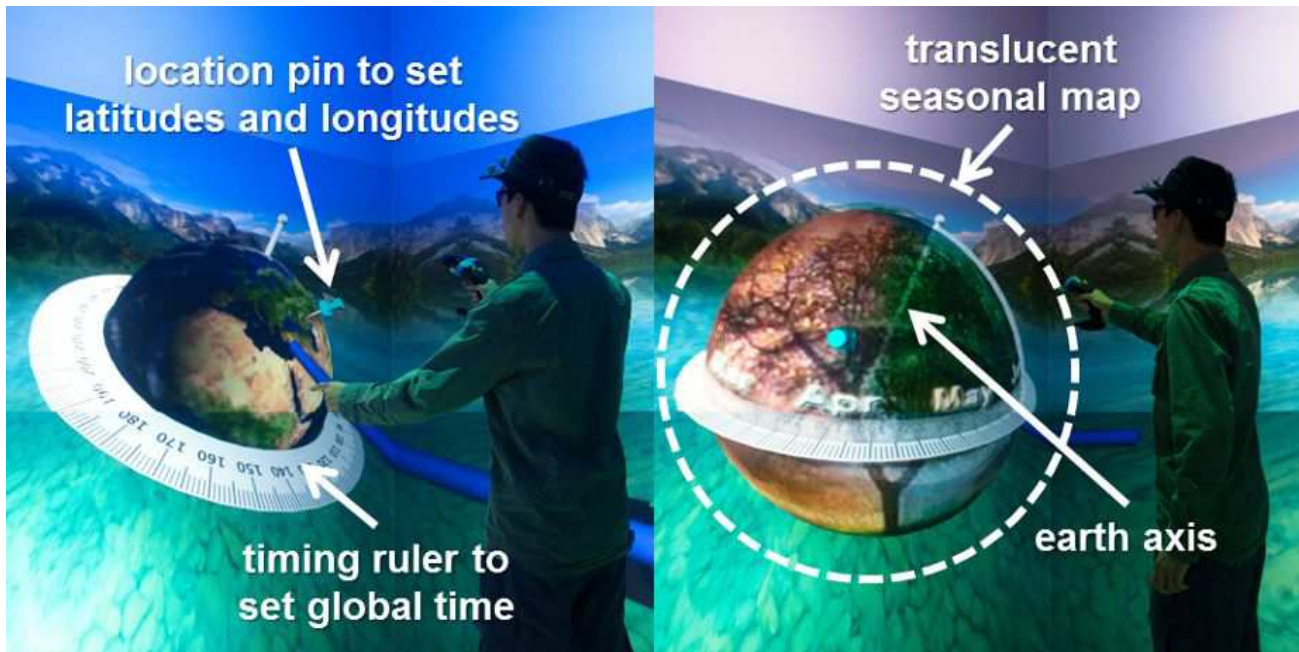


Figure 8. Location (left) and date/time setting (right) with the sphere UI.

The location and date/time selection widgets of CaveCAD are very effective, purpose-built 3D widgets, which are entirely built based on the model of our spherical state machine. The location for the model can be selected directly by moving a pin to a location on a spherical globe widget with the pointer, see Figure 8. As the user moves the pin, the position of the sun in the virtual world is updated in real-time, showing the effect of moving to higher or lower latitudes, or moving between areas that are in day or night. The time can be set by rotating the earth about its axis. The date can be set by using a special seasonal map widget, which allows setting the tilt of the earth's axis towards the sun.

5. RESULTS

We collected anecdotal information about our system from four of our laboratory colleagues by asking them to attempt to build Disney World's Magic Castle in CaveCAD within fifteen minutes. This was done for the 2013 IEEE 3DUI contest. The users consisted of the current programmer of CaveCAD and three other visualization programmers, who were familiar with CalVR but not CaveCAD. 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.

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 9), 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 9. Magic castle created by expert user during 15 min trial.

6. CONCLUSIONS AND FUTURE WORK

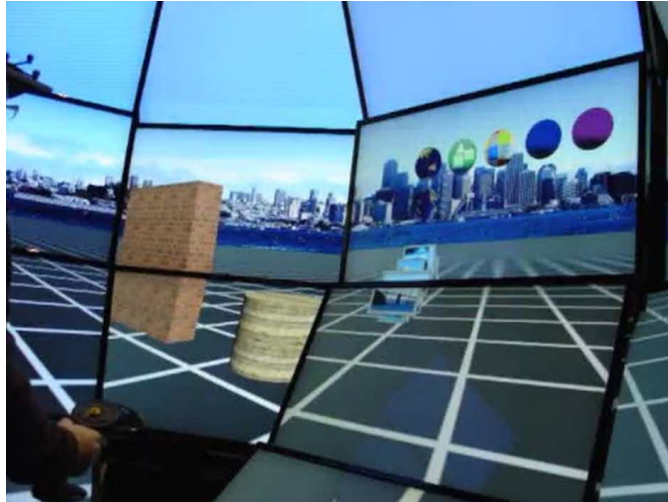
We presented a novel implementation of an architectural design software system named CaveCAD. Our pilot study showed that CaveCAD is intuitive enough for a user with prior, desktop-based 3D modeling experience to quickly learn it and successfully create 3D models with it.

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. Finger tracking would also be desirable, so that the user can use finger pointing and pinches for very high precision, but yet intuitive interaction with geometry.

Finally, an important aspect for professional users of the system is the integration with existing 3D modeling tools. We currently only support importing VRML models, and lack export functionality which would make CaveCAD models available in other 3D modeling tools. In the future, we want to add exporting to at least one commonly used 3D file format, without losing the meta-information about the geometry so that it can further be edited.

ACKNOWLEDGMENTS

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



. Video 1. The video clip at the following link shows CaveCAD in action: <http://dx.doi.org/doi.number.goes.here>

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] M. F. Deering. Holosketch: a virtual reality sketching/animation tool. *ACM Trans. Comput.-Hum. Interact.*, 2(3):220–238, Sept. 1995.
- [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] E. A. Edelstein, K. Gramann, J. Schulze, N. B. Shamlo, E. van Erp, A. Vankov, S. Makeig, L. Wolszon, and E. Macagno. Neural responses during navigation and wayfinding in the virtual aided design laboratory - brain dynamics of re-orientation in architecturally ambiguous space. *Report Series of the Transregional Collaborative Research Center SFB/TR8 Spatial Cognition, University of Bremen/University of Freiburg, Germany*, 2008.
- [5] EON. Eonstudio home page, http://www.eonreality.com/products_studio.html. EON, 2012.
- [6] T. A. Galyean and J. F. Hughes. Sculpting: an interactive volumetric modeling technique. *SIGGRAPH Comput. Graph.*, 25(4):267–274, July 1991.
- [7] K. Gramann, J. Schulze, A. Vankov, N. Bigdely, E. V. Erp, S. Makeig, and E. Macagno. Synchronization of brain dynamics with navigation behaviors in full scale, stereoscopic 3d virtual reality architecture. *Talk included in the Symposium "Environmental determinants of orientation and navigation", organized by C. Hlscher, A.-M. Nenci, R. Troffa for the 4th International Conference on Spatial Cognition (ICSC09), Rome, Italy*, 2009.
- [8] C. Hughes, L. Zhang, J. Schulze, E. Edelstein, and E. Macagno. Cavecad: Architectural design in the cave. *Refereed user interface contest entry, in proceedings of IEEE 3DUI Symposium 2013, Orlando, FL*, 2013.
- [9] 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.
- [10] J. Liang and M. Green. JDCAD: A highly interactive 3D modeling system. In *Computer and Graphics*, volume 18(4), pages 499–506, 1994.
- [11] Mechdyne. Conduit. <http://www.mechdyne.com/conduit.aspx/>, 2013.
- [12] D. Rantzau, K. Frank, U. Lang, D. Rainer, and U. Woessner. Covise in the cube: An environment for analyzing large and complex simulation data. *Proceedings of the IPTW*, 1998.
- [13] E. Sachs, A. Roberts, and D. Stoops. 3-draw: A tool for designing 3d shapes. *IEEE Comput. Graph. Appl.*, 11(6):18–26, Nov. 1991.

- [14] J. Schulze, A. Prudhomme, P. Weber, and T. DeFanti. Calvr: An advanced open source virtual reality software framework. In *Proceedings of IS&T/SPIE Electronic Imaging, The Engineering Reality of Virtual Reality*, San Francisco, CA, Feb. 2013.
- [15] R. Stoakley, M. J. Conway, and R. Pausch. Virtual reality on a wim: interactive worlds in miniature. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, pages 265–272, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.
- [16] TechViz. Techviz home page. <http://www.techviz.net/>, 2013.
- [17] WorldViz. Worldviz home page, <http://www.worldviz.com/>. WorldViz, 2012.
- [18] R. C. Zeleznik, K. P. Herndon, and J. F. Hughes. Sketch: an interface for sketching 3d scenes. In *ACM SIGGRAPH 2006 Courses*, SIGGRAPH '06, New York, NY, USA, 2006. ACM.
- [19] L. Zhang, Y. Chi, E. Edelstein, J. Schulze, K. Gramann, A. Velasquez, G. Cauwenberghs, and E. Macagno. Wireless physiological monitoring and ocular tracking: 3d calibration in a fully-immersive virtual health care environment. In *Proceedings of IEEE Engineering in Medicine and Biology Society Conference (EMBC'10)*, Buenos Aires, Argentina, 2010.
- [20] L. Zhang, J. Gossmann, C. Stevenson, M. Chi, G. Cauwenberghs, K. Gramann, J. Schulze, P. Otto, T.-P. Jung, R. Peterson, E. Edelstein, and E. Macagno. Spatial cognition and architectural design in 4d immersive virtual reality: Testing cognition with a novel audiovisual cave-cad tool. In *Proceedings of the Spatial Cognition for Architectural Design Conference*, 2011.