

Interactive Two-Handed Terrain and Set Design in Immersive Environments

Falko Kuester, Bernd Hamann and Kenneth I. Joy

Center for Image Processing and Integrated Computing

Department of Computer Science

University of California, Davis, CA 95616-8562

{kuester, hamann, joy}@cs.ucdavis.edu

Abstract

This paper introduces *Vscape*, a virtual environment for intuitive, hand-based terrain design. We present a design environment that provides an intuitive interface for the creation and manipulation of 3D scenes as required for terrain, game-level and set design. *Vscape* was developed to provide the user with maximum design flexibility while providing a small, yet powerful set of easy-to-use tools and functions.

Keywords: Digital Design, Virtual Reality, Immersive Environments

1. Introduction

Virtual environments (VEs) are being used for industrial product design, analysis and verification tasks, medical imaging, architectural walkthroughs, geo-scientific exploration and sculpting. *Vscape* combines data analysis, designs and verification capabilities of these environments and applies them to scene design suitable for urban planning, game-level design and set design. Modeling environments traditionally rely on the use of polygonal, volumetric or mathematically defined primitives. Since primarily developed for the interactive design of terrains, *Vscape* is currently surface-based and supports hand-based sculpting, painting and texturing on a polygonal level. For this type of application the goal was to “think visually” in terms of shapes, colors and textures instead of vertices, edges or curves and surfaces. As a consequence, the modeling concept is different from traditional keyboard and mouse centered computer based design and closer to traditional hands-on modeling. The user is equipped with a set of spatially tracked gloves and

can employ a head-mounted display, immersive workbench or standard monitor-based stereo to create a 3D scene (Figure 1, 2). The core design criteria was to provide technical and non-technical users with an easy-to-use environment, for the creation of realistic environments. At the same time it was important to offer an unconstrained interface to the user that reduces or removes the pre-meditative design phase. This was accomplished by providing an environment that fosters the use of built-in verification tasks and the development of “game strategies” as part of the design cycle, resulting in a thoroughly developed and tested final product. Visibility, reachability and accessibility controls are built-in features that are automatically used throughout the design cycle. Relevant viewpoints or paths can be created as required and revisited throughout the design cycle for verification tasks.



Figure 1: Terrain modeling and verification

2. Implementation

The observation that humans develop certain patterns on how to distribute tasks between their hands [14,15] has lead to the development of two-handed interfaces supporting this natural dexterity. Most of these interfaces are based on spatially tracked input devices, such as data gloves and pointers. *VScape* uses a set of spatially tracked pinch gloves, which can be used to navigate and manipulate the environment. This hand-based modeling approach provides access to efficient sculpting and painting metaphors that enable efficient and effortless expression of design ideas.

Furthermore, *VScape* is based on an object-oriented design approach, which treats every visual component within the VE as an object that can be freely positioned, manipulated, verified, analyzed and visualized. Once an object is created, its visual representation is added to a hierarchical scene graph. All visible objects contained in the scene graph can be selected and their properties visualized using a simple hand gesture. Special behavioral actions can be attached to any object and turn it into a tool for the manipulation of other objects. Any regular object within the scene graph can be directly accessed, scaled, translated, cloned and grouped. In order to allow intuitive object based modeling, the environment provides a basic set of controls, including:

- Object creation
- Object selection
- Object/scene manipulation
- Object/scene verification

In addition, accessibility controls for these operations are provided in the form of:

- Object/scene navigation
- Virtual menus

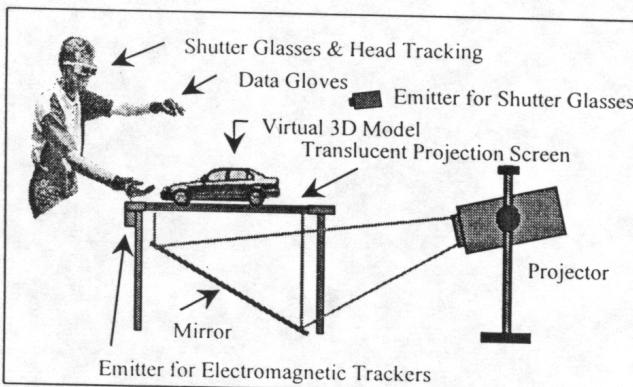


Figure 2: Hardware setup

Scene Navigation

Using head tracking, the user can study an entire model by simply moving his/her head or physically walking

around the model. However, since the variety of possible application settings requires a less restrictive navigation paradigm, a two-handed interface is provided to freely translate, rotate and scale individual objects or the entire scene. In the object navigation mode the user can select an object through a particular one-handed gesture and then freely re-position and analyze it. If no particular object is selected while a navigation action occurs the system switches into scene-navigation mode and the action is applied to the entire scene, which by definition, is just another object composed of a group of objects. In this mode the user can use an imaginary rope to pull himself/herself through the scene using consecutive pinch-pull-release sequences. If both gloves are pinching at the same time, the imaginary segment between the pinching points is used as a five-degrees-of-freedom manipulator that allows to scale or rotate the scene or object. The relative position of the two points in regards to the original center point between the hands when the initial pinch event occurred determines the orientation of the scene and the distance between the two points defines its scale. This scheme supports a user-defined level of accuracy in which finer or coarser levels of precision can be defined by scaling the workspace. In other words, viewpoint movement supports an intuitive translation between working scales and provide direct access to different levels of modeling accuracy. This navigation scheme is intuitive and versatile, and new users are able to easily examine even complex scenes with minimal effort.

Virtual Menus

Menus are a vital component of all modeling systems since they provide access to the available system functions. With the transition from a 2D to a 3D environment, a new set of VR input devices and consequently new concepts must be implemented. Different solutions to this problem were proposed during recent years opting for either a direct port from the classical 2D menu to its 3D counterpart or new implementations designed specifically for 3D space [4]. Commonly observed problems are interference between the 3D menus and the scene and sub-menu access in highly cascading menus. We distinguish between gesture-based trigger and invocation events that allow the user to activate and select from various menus. A simple pinch gesture gives the user access to a base menu, which can be freely positioned in the VE. The menu is composed of 3D buttons assembled on a rectangular palette. All the sub-menus are opened within this original palette and can be traversed using simple hand gestures. Following our original design philosophy, all menus are implemented as objects that can be translated, rotated and scaled as desired. The menu items can apply associated functionality to other objects when

activated and be represented as text, a graphical presentation of the associated function or a combination thereof.

Terrain Creation

Vscape provides a variety of mechanisms for the interactive creation and manipulation of terrain data. Terrain information can be either imported in polygonal form from a file or interactively created by using drawing primitives or freeform shapes and manipulated with a suite of virtual tools. Based on the desired terrain, game level or set, the design cycle can start at different levels from either a planar surface, artist sketch, a blueprint or even a satellite image mapped onto either a surface, 3D model or other types of geometry.

Object Creation

Arbitrary polygonal objects can be used to provide additional scene contents. Application-specific modeling libraries are supported and provide access to a wide range of primitives. These objects can be accurately positioned above the terrain using a rod level and then attached by simply dropping them onto the scene. User-definable objects and libraries including components such as houses and bridges are easily added through customizable menus, allowing the creation of additional scene contents. These objects can be accessed through the virtual toolbox and configured or extended to meet application specific demands. After invoking the virtual menu and selecting the appropriate object library, the chosen object can be simply grabbed and positioned on the terrain where desired. The application supports a “snapping mechanism”, which enables accurate object placement onto the defined terrain. Objects automatically snap to the surface and can easily be cloned, moved, scaled, rotated or planted with a simple gesture. These libraries let the user design a scene, while immersion enables real-time verification. *Vscape* reads and writes most of the standard file formats, including flt, wrl, 3ds and dxf among others.

Object Selection

This operation is the starting point for a variety of interaction tasks. The basic idea is to use a 3D input device to select the closest object to a spatial position. When a device-specific action is invoked in the form of a particular state event the absolute position of the tracker is mapped to world coordinates. The data gloves are visualized with virtual proxies. When the proxy intersects the bounding box of a particular object, the object is highlighted and ready for selection.

Object Manipulation

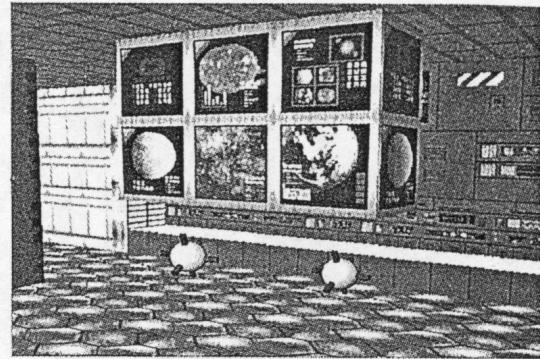
Once an object is selected it can be rotated, translated, scaled, cloned, re-shaped, grouped, deleted, or otherwise manipulated.

Terrain Manipulation

While constructing scenes it is important to observe specific boundary conditions such as construction time and cost. The available virtual tools discussed in the next section support user-definable design constraints, such as the amount of terrain movement per second.

Scene Verification

Design verification tasks such as visibility, reachability and accessibility, are frequently encountered during evaluation tasks. In our environment, they are built-in and are automatically



used throughout the design cycle. If required, relevant viewpoints can be stored and visited as desired.

3. Toolbox

The virtual toolbox merges the advantages of conventional physical tools and unconstrained virtual tools with the natural dexterity of a two-handed design environment. Instead of merely defining tools, we define actions and functionality, which can be associated with a set of geometrically defined modeling primitives provided as part of the toolbox or any object in the scene. Thus giving the user unlimited space for creativity and the means for the creation of new tools and design concepts. In our object-oriented framework, tools can be used to shape models, which subsequently can be turned into tools on their own. The virtual toolbox of *Vscape* includes these types of tools:

- **Brushes** can apply color, material or texture to physical objects they come in touch with.
- **Filters** can be applied to an object or scene, and aid in smoothing, stitching or decimation tasks.
- **Guides** constrain the movement of an object and can be used in combination with any of the listed behaviors. Constraints could be movement in only a certain plane, around a certain axis, in a certain volume, etc.
- **Manipulators** allow high-precision positioning, rotation, and scaling of objects.

- **Magnets** can apply attractive or repulsive forces to objects. The “influence volume” of a magnet is determined by the scale of the environment. One can scale down the environment, with a simple hand movement, and the influence area gets smaller, or scale it up to increase it.
- **Paintbrush** paints color and applies texture to the surface. This tool can be used in brush-mode in combination with any other object in the scene.
- **Rulers** can be used to verify object dimensions or aid in the construction of objects.
- **Stamps** turn an object into a “3D printing stock” imprinting its information on another object.
- **Smoothers** turn an object into a “3D putty knife” or piece of sandpaper for surface smoothing and are primarily used for the removal of hard edges.
- **Snappers** aid in connecting objects within the scene.
- **Tesselators/Simplifiers** add detail at user-specifiable locations.

Magnets and stamps are very efficient tools for common modeling tasks, particularly when artistic creativity is emphasized.

4. Conclusions

Vscape provides an intuitive environment for rapid prototyping of terrain, sets or game levels. It preserves the natural dexterity of physical modeling environments while providing the benefits of a digital design space. The current challenge lies in the development of more complex interaction and modeling schemes in the form of new virtual tools and input devices, using voice, gesture and pattern recognition. One of the most challenging tasks for the near future is to provide the necessary modeling precision required for engineering design tasks.

The object-oriented framework is easily extendable and provides a user-friendly prototyping environment. An enlarged feature set is currently under development. The generation of viewpoint-dependent adaptive meshes in real-time, subject to user-specified frame rates and/or error bounds, is targeted for performance reasons. Additionally, the rising number of programmable force-feedback devices and decreasing cost promises even more intuitive interaction potential. As for most new technologies, the initial investment of resources is substantial, but the rapid development of graphics hardware already shows good performance on high-end PC systems.

5. Acknowledgements

This work was supported by the National Science Foundation under contracts ACI 9624034 and ACI 9983641 (CAREER Awards), through the Large

Scientific and Software Data Set Visualization (LSSDSV) program under contract ACI 9982251, and through the National Partnership for Advanced Computational Infrastructure (NPACI); the Office of Naval Research under contract N00014-97-1-0222; the Army Research Office under contract ARO 36598-MA-RIP; the NASA Ames Research Center through an NRA award under contract NAG2-1216; the Lawrence Livermore National Laboratory under ASCI ASAP Level-2 Memorandum Agreement B347878 and under Memorandum Agreement B503159; the Lawrence Berkeley National Laboratory; the Los Alamos National Laboratory; and the North Atlantic Treaty Organization (NATO) under contract CRG.971628. We also acknowledge the support of ALSTOM Schilling Robotics, Chevron, General Atomics, Silicon Graphics, Inc. and ST Microelectronics, Inc. We thank the members of the Visualization Thrust at the Center for Image Processing and Integrated Computing (CIPIC) at the University of California, Davis.

6. References

1. Agrawala, M., Beers, A. C., Froehlich, B., Hanrahan, P., McDowall, I. And Bolas, M., The two-user responsive workbench: Support for collaboration through independent views of a shared space, in SIGGRAPH 97 Conference Proceedings, T. Whitted, ed., Annual Conference Series, pp. 327-332, ACM SIGGRAPH, Addison Wesley, Aug. 1997.
2. Beier, K. P., Virtual reality in automotive design and manufacturing. In: Convergence '94, International Congress on Transportation Electronics, Dearborn, Michigan, October 1994. SAE (Society of Automotive Engineers).
3. Cutler, L. D., Froehlich, B., Hanrahan, P., Two-handed direct manipulation on the responsive workbench, in Proceedings of the Symposium on Interactive 3D Graphics, pp. 107-114, ACM Press, (New York), Apr. 27-30 1997.
4. Deering, M. F., The HoloSketch VR sketching system, Communications of the ACM, 39(5):54-56, 1996.
5. Durlach I., Mavor, A.S., Committee on Virtual Reality Research, Commission on Behavioral Development, Social Science, Mathematics Education, Commission on Physical Sciences, and Applications, National Research Council. “Virtual Reality: Scientific and Technological Challenges,” National Academy Press, 1994.
6. Ebert, D. S., Shaw, C. D., Zwa, A. and Starr, C., Two-handed interactive stereoscopic visualization, in: Proceedings of IEEE, R. Yagel and G. M. Nielson, eds., pp. 205-210, IEEE Press, Los Alamitos, Oct. 27-Nov. 1 1996.

7. Farin, Gerald E., Curves and Surfaces for Computer-Aided Geometric Design, 4th edition, Academic Press, San Diego, California, 1997
8. Froehlich, B., Fischer, M., Agrawala, M., Beers, A. and Hanrahan, P., Projects in VR: Collaborative production modeling and planning, IEEE Computer Graphics and Applications 17, pp. 13-15, July-Aug. 1997.
9. Froehlich, B., Barrass, S., Zehner, B., Plate, J. and Goebel, M., Exploring geo-scientific data in virtual environments, in: D. S. Ebert, M. Gross and B. Hamann, eds., Proceedings of IEEE Visualization '99, IEEE Computer Society Press, Los Alamitos, California, pp. 169-173, October 1999.
10. Galyean, T. A. and Hughes, J. F., Sculpting: An interactive volumetric modeling technique, Computer Graphics 25 (Proc. ACM SIGGRAPH '91), pp. 267-274, July 1991.
11. Green, M. and Halliday, S., A geometric modeling and animation system for virtual reality, Communications of the ACM 39, pp. 46-53, May 1996.
12. Green, M., Shared virtual environments: The implications for tool builders, Computers and Graphics 20, pp. 185-189, Mar.-Apr. 1996.
13. Guiard, Y., Symmetric division of labor in human skilled bimanual action: The kinematic chain as a model, The Journal of Motor Behavior, 19(4):486-517, 1987.
14. Guiard, Y. and Ferrand, T., Asymmetry in bimanual skills, in: Manual asymmetries in motor performance, D. Elliott and E. A. Roy, eds., CRC Press, Boca Raton, FL., 1995.
15. Hui, K. C. and Ma, M. C., Deforming virtual objects with an instrumented glove, in: Proceedings of the Conference on Computer Graphics International 1998 (CGI-98), F.-E. Wolter and N.M. Patrikalakis, eds., pp. 393-395, IEEE Computer Society Press, Los Alamitos, California, June 22-26 1998.
16. Krueger, W., Froehlich, B. Visualization blackboard: The responsive workbench, IEEE Computer Graphics and Applications 14(3): May 1994, 12-15.
17. Kuester, F., Duchaineau, M. A., Hamann, B., Joy, K. I., and Uva, A. E. (1999), 3DIVS: 3-dimensional immersive virtual sculpting, in: Ebert, D. S. and Shaw, C. D., eds., Workshop on New Paradigms in Information Visualization and Manipulation (NPIV '99), ACM Press, New York, New York, Kansas City, Missouri, November 1999, pp. 132-139.
18. Liverani, A., Kuester, F. and Hamann, B. (1999). Towards interactive finite element analysis of shell segments in virtual reality, in: Banissi, E., Khosrowshahi, F., Sarfraz, M., Tatham, E. and Ursyn, A., eds., in: Proceedings of 1999 IEEE International Conference on Information Visualization (IV'99) - Augmented and Virtual Reality Symposium, IEEE Computer Society Press, Los Alamitos, California, pp. 340-346.
19. Noble, R. A. and Clapworthy, G. J., Sculpting and animating in a desktop VR environment, in: Proceedings of the Conference on Computer Graphics International 1998 (CGI-98), F.-E. Wolter and N. M. Patrikalakis, eds., pp. 187-197, IEEE Computer Society, Los Alamitos, California, June 22-26 1998.
20. Obeysekare, U., Williams, C., Durbin, J., Rosenblum, L., Rosenberg, R., Grinstein, F., Ramamurthi, F., Landsberg, A., and Sandberg, W., Virtual workbench: A non-immersive virtual environment for visualizing and interacting with 3D objects for scientific visualization, in: Proceedings of IEEE Visualization '96, R. Yagel and G. M. Nielson, eds., IEEE, Los Alamitos, Oct. 27-Nov. 1 1996, pp. 345-349.
21. Piegl, L. and Tiller, W., The NURBS Book, 2nd edition, Springer Verlag, New York, New York, 1996
22. Rosenblum, L., Durbin, J., Doyle and Tate, D., Projects in VR: Situational awareness using the responsive workbench. IEEE Computer Graphics and Applications 17(4):July/August 1997, pp. 12-13
23. Sastry, L., Ashby, J. V., Boyd, D. R. S., Fowler, R. F., Greenough, C., Jones, J., Turner-Smith, E. A. and Weatherill, N. P., Virtual reality techniques for interactive grid repair, Numerical Grid Generation in Computational Field Simulations, Ed.
24. Shaw, C. D., Green, M., THRED: A two-handed design system, Multimedia System Journal 3(6), November 1995.
25. Shaw, C. D. and Green, M., Two-handed polygonal surface design, in: Proceedings of ACM Symposium on User Interface Software and Technology, Two Hands and Three Dimensions, 1994, pp. 205-212.
26. Wong, J. P. Y., Lau, R. W. H. and Ma, L., Virtual 3D sculpturing with a parametric hand surface, in: Proceedings of Computer Graphics International 1998 (CGI-98), F.-E. Wolter and N. M. Patrikalakis, eds., IEEE Computer Society, Los Alamitos, California, June 22-26 1998, pp. 178-186.