The DesignersWorkbench: Towards Real-Time Immersive Modeling

Falko Kuester[a], Mark A. Duchaineau[b], Bernd Hamann[a], Kenneth I. Joy[a] and Kwan-Liu Ma[a]

[a]Center for Image Processing and Integrated Computing
Department of Computer Science
University of California, Davis, CA 95616-8562

[b]Lawrence Livermore National Laboratory
7000 East Avenue, Livermore, CA 94550

ABSTRACT

This paper introduces the DesignersWorkbench, a semi-immersive virtual environment for two-handed modeling, sculpting and analysis tasks. The paper outlines the fundamental tools, design metaphors and hardware components required for an intuitive real-time modeling system. As companies focus on streamlining productivity to cope with global competition, the migration to computer-aided design (CAD), computer-aided manufacturing (CAM), and computer-aided engineering (CAE) systems has established a new backbone of modern industrial product development. However, traditionally a product design frequently originates from a clay model that, after digitization, forms the basis for the numerical description of CAD primitives. The DesignersWorkbench aims at closing this technology or "digital gap" experienced by design and CAD engineers by transforming the classical design paradigm into its fully integrated digital and virtual analog allowing collaborative development in a semi-immersive virtual environment. This project emphasizes two key components from the classical product design cycle: freeform modeling and analysis. In the freeform modeling stage, content creation in the form of two-handed sculpting of arbitrary objects using polygonal, volumetric or mathematically defined primitives is emphasized, whereas the analysis component provides the tools required for pre- and post-processing steps for finite element analysis tasks applied to the created models.

Keywords: Virtual Reality, Immersive Environments, 3D Modeling, Computer Aided Geometric Design (CAGD)

1. INTRODUCTION

As companies focus on streamlining productivity as a response to global competition, the migration to computer-aided design CAD, CAM and CAE systems has established a new backbone of modern industrial product development. Primarily driven by product quality, cost, and time-to-market considerations, most automotive and aerospace companies have heavily invested in the development and implementation of new virtual reality (VR) technology during the last five years. These newly created synthetic environments (SEs) have matured into valuable tools in the areas of human factors and usability studies, manufacturing, and simulation-based design. Until now, while immersive or augmented VR technology has been used by most of these applications to visualize, modify or interact with pre-defined data, its potential for actual content creation and integrated design tasks has been largely neglected. While most of these technological advances are of high benefit to the engineering and design community [2], they still lack some of the important visual and haptic features crucial to product development. A car design, for example, traditionally originates from a clay model, which, after digitization, forms the basis for a numerical CAD description in Bezier, B-spline, or NURBS format [7,23]. Furthermore, physical models, so called mock-ups, still play a key role in the areas of design evaluation and verification, introducing a discontinuity in the otherwise CAD-centered development cycle. On the other hand, the engineer eventually has to perform analysis tasks on the newly created model, which also can benefit from a 3D-enabled user interface for certain pre- and post-processing tasks. The DesignersWorkbench system is a collection of visualization, modeling and analysis tools [18,19,20] that aim at closing the technology gap experienced by design and CAD engineers by transforming the classical design paradigm into its fully integrated digital and virtual analog (Figure 1). The goal is to facilitate the cooperation between designer and engineer and to aid in streamlining design, analysis and testing phases. This is achieved by providing a real-time 3D semi-immersive design environment that supports modeling metaphors such as intuitive hand gestures and virtual tools for the creation and

In *Stereoscopic Displays and Virtual Reality Systems VII*, John O. Merritt, Stephen A. Benton, Andrew J. Woods, Mark T. Bolas, Editors, Proceedings of SPIE Vol. 3957 (2000) • 0277-786X/00/$15.00

modification of geometric and CAD primitives. This approach allows the preservation of the hands-on modeling experience from the physical world while overcoming the classical 2D constraints introduced through the keyboard. The appeal of VR-based design technology lies in its unprecedented amount of real estate in the form of a "3D desktop". Users experienced in working with multiple open and overlapping windows or virtual 2D desktops on a regular display will appreciate that objects, tools, and other components can now be placed and arranged in an almost unlimited 3D domain. In its final version, this integrated environment should allow designers to turn their insight into designs, designs into models, and finally models into production data that in turn can be manipulated and analyzed by the engineer without having to break the digital chain. Therefore, the objective is to design and develop a next-generation suite of tools that will dramatically reduce cost, time, and complexity in industrial design and development.

2. SYSTEM LAYOUT

The system was specifically designed to work with a new generation of stereo projection systems currently marketed under names like *Immersive Workbench*, *Responsive Workbench* and *ImmersaDesk* [17,22]. We used the *Immersive Workbench* from Fakespace, which allows stereo projection of 3D computer-generated images onto an approximately 2×1.5m projection area. However, the system is not limited to this technology and will work in any setting, provided the appropriate device drivers are available. In our current setup a 4-processor SGI Onyx2 InfiniteReality system (225MHz, R10000 processor) is used as the rendering engine. The basic hardware setup is illustrated in Figure 2. The user is wearing shutter glasses with integrated head tracking for stereoscopic viewing and uses a set of pinch gloves combined with a stylus device for interaction with the VE. The spatial data describing the user's head position and hand movements is fully incorporated into the VE. We briefly describe the input devices:

- **Stylus**: Using a fixed transmitter as reference, this pencil-like system accurately computes position (x, y and z coordinates) and orientation (yaw, pitch and roll) of a tiny receiver contained in the stylus. In addition, it provides an integrated button that can be used for picking actions.
- **Gloves**: The pinch system uses cloth gloves with electrical sensors in each fingertip. Contact between any two or more digits completes a conductive path, providing a variety of possible "pinch" gestures that can be associated with distinct actions. Additionally, an attached electromagnetic tracker captures the position of each glove.

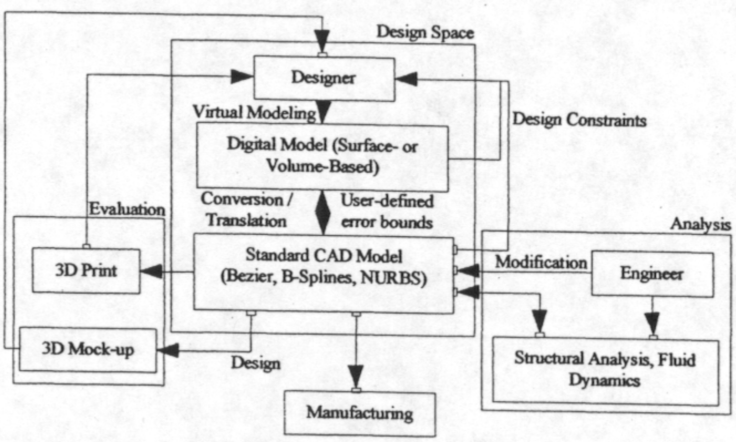

Figure 1: Integrated modeling environment

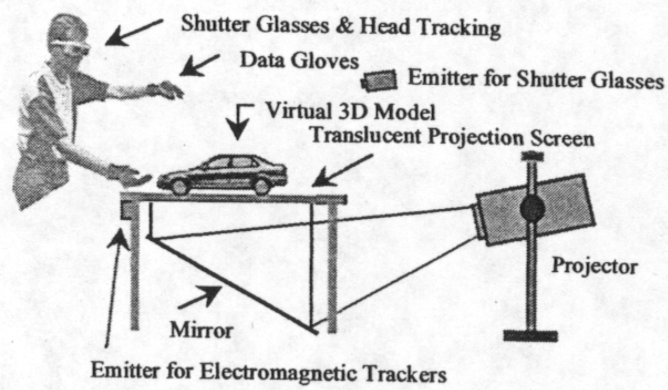

Figure 2: Hardware setup

3. IMPLEMENTATION

For the creation of a model in a VE a different type of human-computer interface is required. The observation that humans develop certain patterns on how to distribute tasks between their hands [14,15] has lead to the development of natural two-handed interfaces based on spatially tracked input devices such as data gloves and pointers in combination with user head tracking [3,6,9,27]. In our implementation, the user generally wears two gloves and has access to a stylus device, which can be used as an additional tool with situation-specific meaning.

An object-oriented design approach was chosen, which treats every visual component within the VE as an object that can be freely positioned, manipulated, verified, analyzed and visualized. Special behavioral actions can be attached to any object and turn into a tool for the manipulation of other objects. Any regular non-static object within the scene graph can be directly accessed, manipulated and grouped. Non-static objects come without any attached constraints, whereas static objects come with a set of distinct attributes defining which manipulations and interactions are allowed within certain boundaries. These objects are of particular interest for industrial design since they preserve predefined design constraints. 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. In order to allow intuitive object based modeling, the environment has to provide a basic set of controls, including:

- Object creation
- Object selection
- Object manipulation
- Object verification
- Object analysis

In addition, accessibility controls for these operations have to be provided in the form of:

- Scene/object navigation
- Virtual menus

Another important prerequisite was that an object design history must be accessible. This approach allows us to enable important features including undo and redo operations, but it also supports tracing of efficient artistic or engineering design patterns.

1. Scene Navigation

Using head tracking, the designer can study an entire model in a VE by simply moving his/her head or physically walking around the model. The navigation mode is enhanced in two different ways. By performing a pinch action the user can select and re-position an object by using either one of the data gloves. This mode, called object-navigation mode, allows the user to freely re-position and analyze the active object. If both gloves are pinching at the same time, the system switches into scene-navigation mode. The imaginary segment between the pinching points is used as a five-degrees-of-freedom manipulator. The length of this segment, i.e., the distance of the two picking points, controls the scale of the model. In the case that no object has been selected, the navigation action is applied to the entire scene, which, by definition, is just another object composed of a group of objects. This scheme also supports a user-defined level of accuracy in which finer or coarser levels of precision can be defined by scaling the workspace. This navigation scheme is intuitive and versatile, and new users are able to easily examine even complex scenes after only a few minutes of practice.

2. Virtual Menus

Menus are a vital component of all modeling systems since they in general provide intuitive 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 have to 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 reaching sub-menu options in highly cascading menus. We distinguish between gesture-based trigger and invocation events that allow the user to activate and select from various menus. For the novice user, the "watch-menu" or "communicator-badge" option is supported, which displays the base menu as soon as the user pretends to check the time on his/her wristwatch. The menu is composed of 3D buttons assembled on a rectangular palette and is attached to the "watch" glove. Besides its clarity and accessibility, we choose this 3D "palette" menu since its structure is well accepted in the design and engineering community. All the sub-menus pop up from the original palette in a different plane and can be accessed or closed with the other hand. Each plane defines a sub-menu level while the higher level is slightly faded to help the user focus on the selection. When working in a semi-immersive environment the menu has to be translated slightly to be visible and not obscured by the user's hand. Following the original design philosophy, all menus are implemented as static-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 in either text, a graphical presentation of the associated function or a combination thereof (Figure 3).

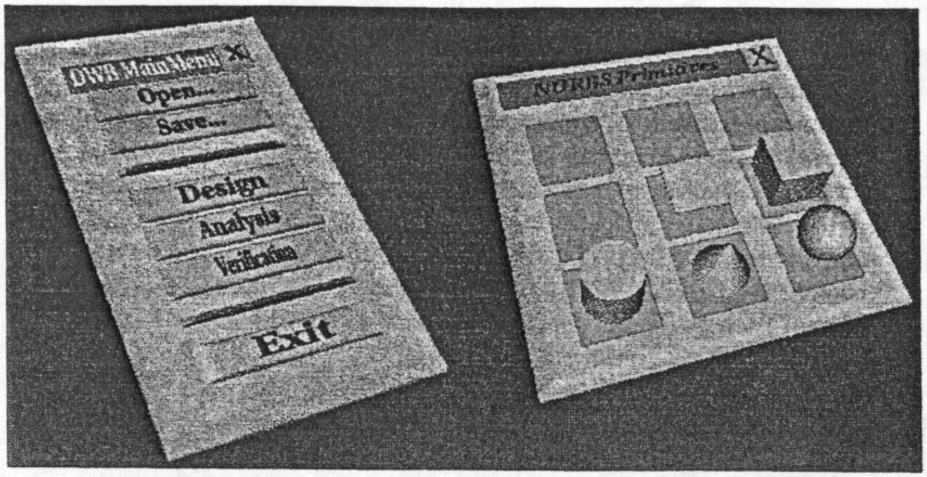

Figure 3: Virtual menus

3. Object Creation

The DesignersWorkbench provides a variety of mechanisms for the creation and manipulation of objects. Objects can be either imported from a file or interactively created by using drawing primitives, freeform shapes, an octree-based clay model or a set of CAD primitives. The included drawing primitives include lines, basic polygonal shapes and triangulated point sets and support the conventional line- and polygon-oriented sketching concept. The CAD primitives are represented in Bezier, B-spline or NURBS form, and for smaller models tessellated on the fly. Supported CAD primitives and operations are: point, line, curve, circular arc, elliptic arc; surface, ruled surface, surface of revolution (cylinder, sphere, etc.); symmetric geometry creation by using reflection, curve fitting, surface fitting, Coons patches and trivariate solids [7]. Most CAD primitives have a predefined set of default properties assigned to them. In the case of the NURBS surface primitive, the system creates a predefined set of control points, knot vectors, weights and orders.

4. Object Selection

This operation is the starting point for any interaction task. The basic idea is to use a 3D input device and to select the closest object to the spatial position of the device when a device-specific action is invoked in the form of a button press, pinch or gesture. While this operation can be easily performed in 2D by using the intersection of a ray with the screen, this approach is not satisfying for 3D environments. Specifying a target based on the absolute (x, y, z) position of the tracker can be a tiresome task and better solutions using ray casting and cone casting metaphors exist. Instead of directly specifying the 3D point, the spatial input device is used to shoot a ray or a spotlight into the scene, allowing the user to hold the input device in a comfortable position and rotate it to change the ray direction. Nevertheless, since accuracy and ease-of-use are key design factors for our application, we decided to use the absolute position of the tracker mapped to world coordinates. In the case of the stylus, a virtual proxy at the end of its tip visualizes the target pointer. When the proxy intersects the bounding box of a selected object, the object is highlighted and ready for selection. By allowing translation or scaling an entire scene or particular set of objects, interactivity is improved substantially. Once an object is selected it can be rotated, translated, scaled, re-shaped, grouped, copied, deleted, or otherwise manipulated, provided the object constraints allow the particular operation.

5. Object Manipulation

It is important to distinguish between engineering and design metaphors in the context of object creation and manipulation. For example, the engineering metaphor for a NURBS surface provides the user with direct access to control points, weights, degrees and knot vectors. This type of representation is most likely not suitable for designers in the sculpting and modeling arena and demands for a generic abstraction layer hiding the mathematical description while introducing hands-on modeling concepts. Instead of picking a control point and moving it around in space, all a designer should have to worry about is a given surface and a set of tools that operate on it. Therefore, the design metaphor masks this information using hand-surface interpolation with local feature control.

6. Object Verification

Design verification tasks such as visibility, reachability and accessibility, frequently encountered in human factors and ergonomic evaluations, are by the nature of our modeling environment a built-in feature that is automatically used throughout the design cycle. If required, relevant viewpoints can be stored and visited as desired.

7. Object Analysis

Finite element analysis (FEA) is an integral component of the product design cycle. The DesignersWorkbench emphasizes on offering an integrated pre- and post-processing visualization environment that provides the required input to an off-the-shelf finite element solver and in a post-processing step, supports the visualization of the results [20]. In the pre-processing step, the previously created virtual model is passed to a mesh generator that creates a shell system that can interactively be refined. This is used as the basis for the definition of the appropriate boundary constraints as required by the solver. In other words, the visual wrapper can be used to interactively determine desired or optimal design parameters and visualize simulation results.

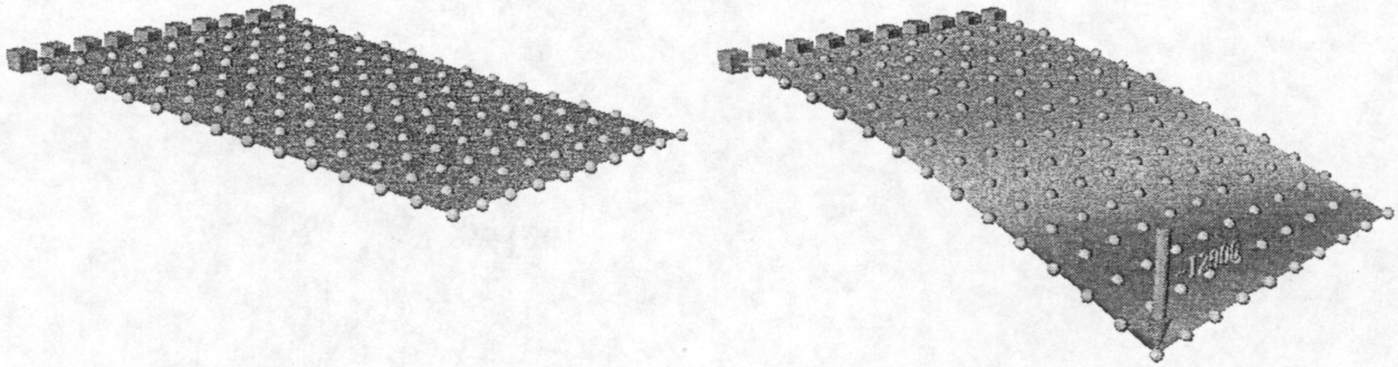

Figure 4: Finite element analysis example. (FEA-solver courtesy of A. Liverani, see [20])

8. Virtual Toolbox

The virtual toolbox merges the advantages of conventional physical tools with the high-precision components of today's CAD systems. Instead of actually defining tools, we define actions and functionality, which can be applied to arbitrary objects in the VE. By doing this, we provide the designer with 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 includes this functionality:

- **Boolean operations** are supported for volumetric objects represented as octrees.
- **Blueprint** generators enable arbitrary-angle projection of both internal and external object boundaries.
- **Brushes** can apply color, material or physical properties to objects they come in touch with.
- **Cutters** turn objects into cutting tools, which can slice through arbitrary material.
- **Editors** allow a user to adjust object material, color, physical or behavioral properties.
- **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. This constraint could be to move only in 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. Either a string or spring model applied to a user-specifiable influence volume, such as sphere or cone, controls the magnet behavior. The numerical spring model relies on control points, that when manipulated, propagate changes based on material properties and otherwise defined constraints through the entire object. On the other hand, the string system supports linear changes to the entire model such as distortions.
- **Masks** allow the specification of object constraints by marking an entire object or parts of it as sensitive or insensitive to particular operations.
- **Morphs** can be used to convert an object to another one or to approximate its geometric properties.
- **Peelers** allow the uniform removal of material layers from an object for modeling or visualization purposes.
- **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. The imprint can be a combination of geometric, material and behavioral properties.
- **Smoothers** turn an object into a 3D putty knife or piece of sandpaper for surface smoothing.
- **Snappers** aid in connecting objects within the scene.

Magnets and stamps turned out to be very efficient tools for common modeling tasks particularly when artistic creativity is emphasized over engineering precision (Figure 7).

9. Application Framework

The application framework for the DesignersWorkbench is depicted in Figure 5. The conceptual idea is to develop a fully object-oriented framework that can handle a variety of different drawing primitives and CAD objects on a plug-in basis. Each object provides the system with information about its type, properties and visual representation. CAD objects additionally furnish the description of their control grid or control mesh, which is used to compute the visual representation for the particular object. The visual representation of any object is stored in a hierarchical scene graph that maintains distinct callbacks to the creating objects. The object type is used to associate the object with a particular virtual menu providing access to a set of common options, and its property field provides the means for adding object-specific controls to the VE. The FEA preprocessor registers its functionality with the main menu and supports a customizable wrapper for the numerical solver itself.

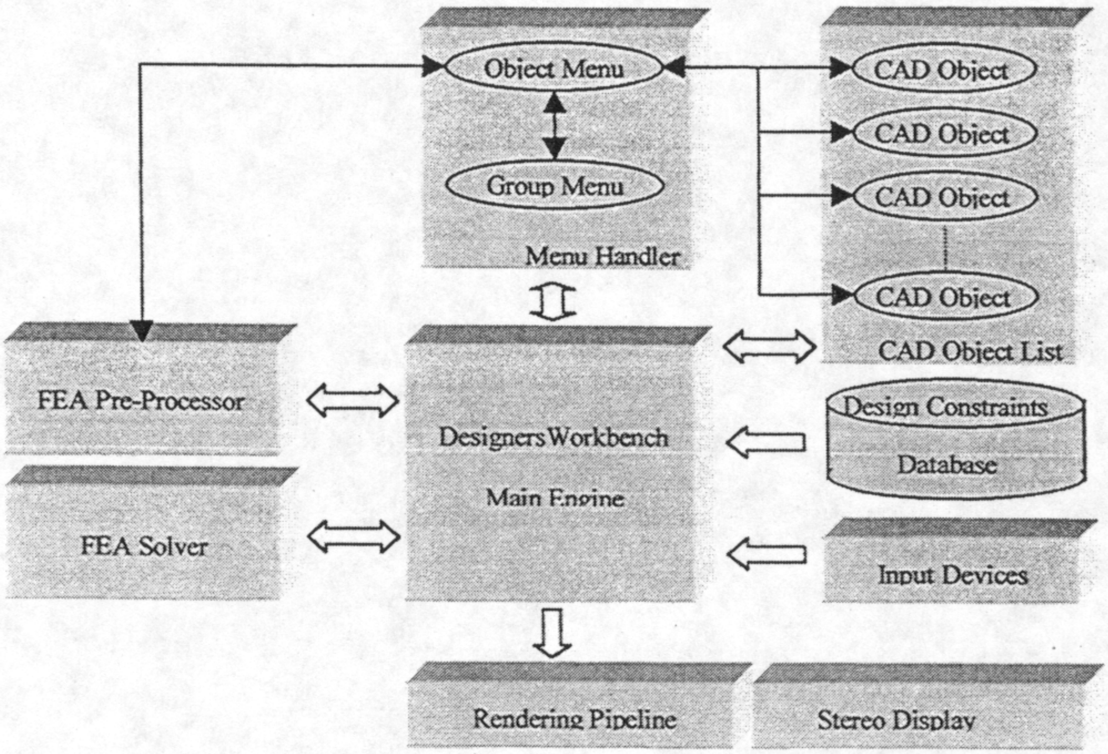

Figure 5: Object-oriented application framework

4. CONCLUSIONS

We have described some of the components required for a fully integrated semi-immersive VE. Although only at a preliminary stage, the available components of the DesignersWorkbench have demonstrated to be very efficient, both for quick sketching, sculpting, CAD modeling and the pre- and post-processing of finite element data. The evaluation of the finite element component was limited to providing an intuitive input mechanism for certain design constraints and input parameters and to visualizing results obtained from an integrated solver. In its current implementation, our system demonstrated that for more complex models real-time performance between the input definition and output verification stage is difficult to achieve. Nevertheless, working on simpler models our test subjects reported being able to see and conceptualize information that was either invisible in traditional modeling/analysis environments or could not be interpreted correctly.

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 core tasks for the near future is to provide the necessary modeling precision required for standard engineering design tasks. The object-oriented framework is easily extendable and provides a user-friendly prototyping environment. An enlarged CAD feature set is currently under development and eventually will be accompanied by a collection of CAD file converters. Appropriate schemes for volume rendering, texture mapping, shading and surface quality interrogation are currently being investigated. 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 gives reason to hope that this technology will be applicable to high-end PC systems within a few years. With increasing graphics system performance and the use of computer clusters for numerical simulation tasks we hope to be able to address the pressing performance issues on the analysis side.

5. ACKNOWLEDGEMENTS

This work was supported by the National Science Foundation under contract ACI 9624034 (CAREER Award), through the National Partnership for Advanced Computational Infrastructure (NPACI), and through a Large Scientific and Software Data Set Visualization (LSSDSV) award; 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; and the North Atlantic Treaty Organization (NATO) under contract CRG.971628 awarded to the University of California, Davis. 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, 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., Guided navigation of virtual environments, in 1995 Symposium on Interactive 3D Graphics,
11. Galyean, T. A. and Hughes, J. F., Sculpting: An interactive volumentric modeling technique, Computer Graphics 25 (Proc. ACM SIGGRAPH '91), pp. 267-274, July 1991.
12. Green, M. and Halliday, S. , A geometric modeling and animation system for virtual reality, Communications of the ACM 39, pp. 46-53, May 1996.
13. Green, M., Shared virtual environments: The implications for tool builders, Computers and Graphics 20, pp. 185-189, Mar.-Apr. 1996.
14. 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.
15. 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.

16. 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. Wolterand and N.M. Patrikalakis, eds., pp. 393-395, IEEE Computer Society, Los Alamitos, California, June 22-26 1998.
17. Krueger, W., Froehlich , B. Visualization blackboard: The responsive workbench, IEEE Computer Graphics and Applications 14(3):12-15, May 1994.
18. 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, pp. 132 –139, Kansas City, Missouri, November 1999.
19. Kuester, F., Uva, A., Hamann, B. and Monno, G., 3DIVS: 3-dimensional immersive visual sketching, in: Proceedings of 12th International Conference on Engineering Design (ICED '99), 1999.
20. 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.
21. 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. Wolterand and N. M. Patrikalakis, eds., pp. 187-197, IEEE Computer Society, Los Alamitos, California, June 22-26 1998.
22. 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., pp. 345-349, IEEE, Los Alamitos, Oct. 27-Nov. 1 1996.
23. Piegl, L. and Tiller, W., The NURBS Book, 2nd edition, Springer Verlag, New York, New York, 1996
24. Rabaetje, R., Integration of basic CAD functions into a VR environment, in: Proceedings of Computer Graphics International 1998 (CGI-98), F.-E. Wolter and N. M. Patrikalakis, eds., pp. 238-241, IEEE Computer Society Press, Los Alamitos, California, June 22-26 1998.
25. Rosenblum, L., Durbin, J., Doyle and Tate, D., Projects in VR: Situational awareness using the responsive workbench. IEEE Computer Graphics and Applications 17(4):12-13, July/August 1997.
26. 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.
27. Shaw, C. D, Green, M., THRED: A two-handed design system, Multimedia System Journal 3(6), November 1995.
28. 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, pp. 205-212, 1994.
29. 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., pp. 178-186, IEEE Computer Society, Los Alamitos, California, June 22-26 1998.
30. Yeh, T.P., Vance and Judy, M., Combining MSC/Nastran, sensivity methods, and virtual reality to facilitate interactive design", MSC/Nastran Web site: http://www.macsch.com/tech/paper1/paper1.html

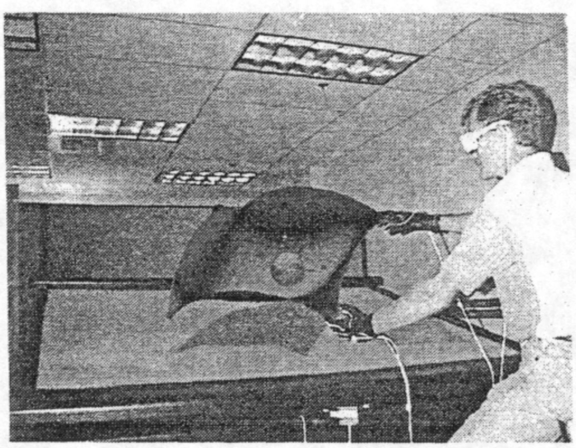

Figure 6: Example modeling task using Bezier surfaces

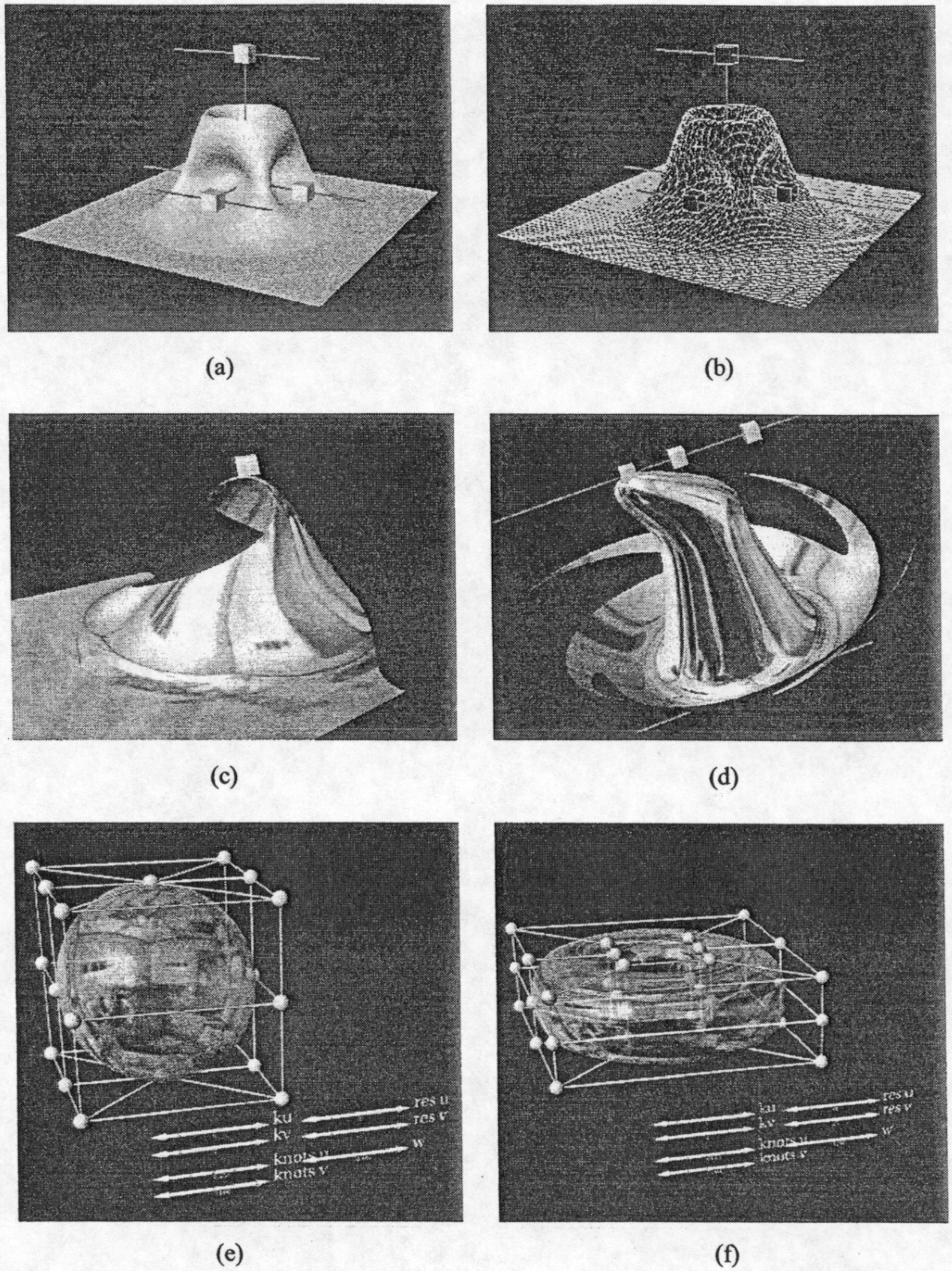

Figure 7: Modeling examples. (a), (b) Modeling with Magnets. (c), (d) Stamps and magnets including twist. (e),(f) Example of NURBS primitives in engineering mode.

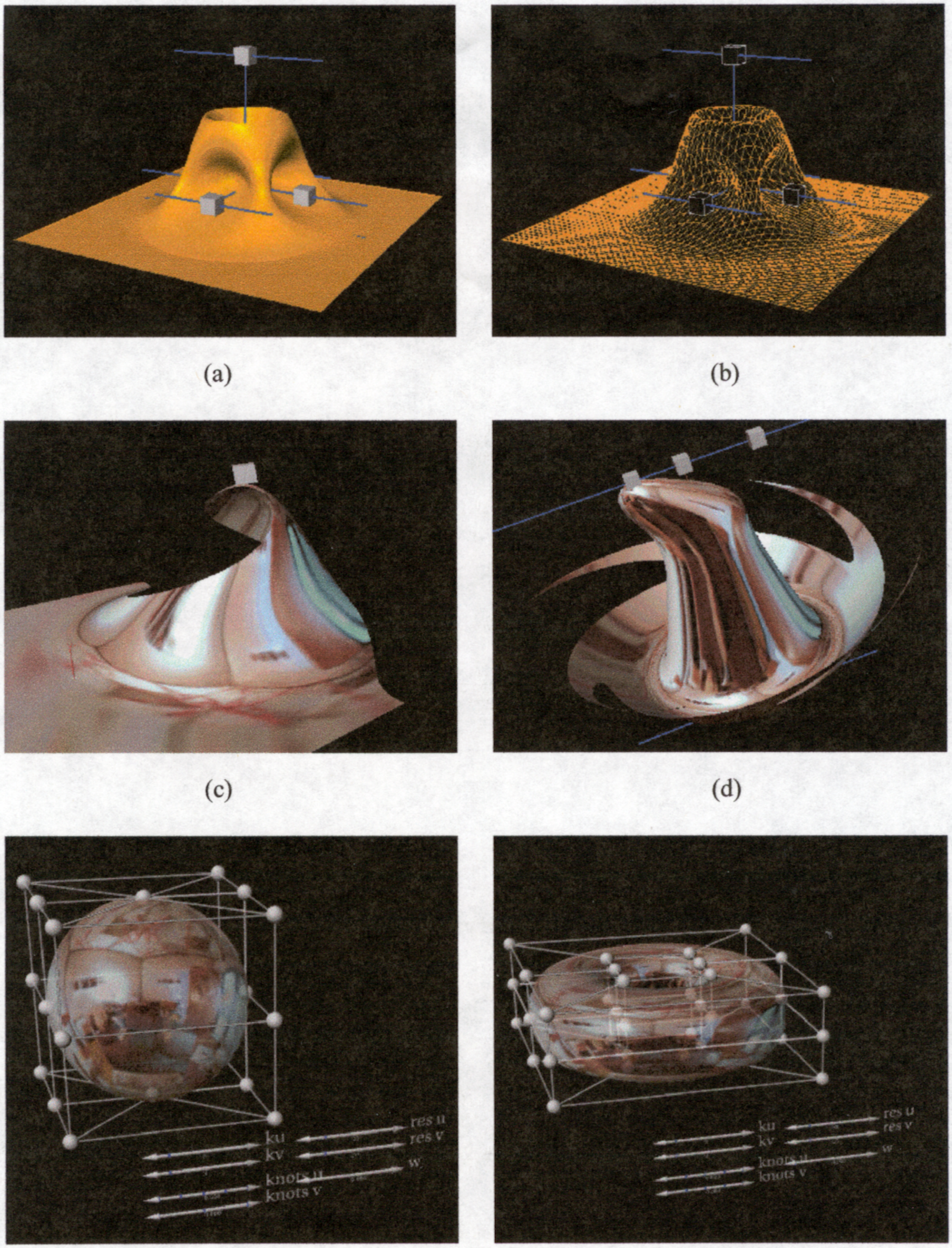

Figure 7: Modeling examples. (a), (b) Modeling with Magnets. (c), (d) Stamps and magnets including twist. (e),(f) Example of NURBS primitives in engineering mode.