

Using Transparent Props For Interaction With The Virtual Table

Dieter Schmalstieg¹, L. Miguel Encarnação², and Zsolt Szalavári³

¹Vienna University of Technology, Austria

²Fraunhofer CRCG, Inc., Providence, RI, USA

³Imagination GmbH, Vienna, Austria

Abstract

The Virtual Table presents stereoscopic graphics to a user in a workbench-like setting. This paper reports on a user interface and new interaction techniques for the Virtual Table based on transparent props—a tracked hand-held pen and a pad. These props, but in particular the pad, are augmented with 3D graphics from the Virtual Table's display. This configuration creates a very powerful and flexible interface for two-handed interaction that can be applied to other back-projected stereographic displays as well: the pad can serve as a palette for tools and controls as well as a window-like see-through interface, a plane-shaped and through-the-plane tool, supporting a variety of new interaction techniques.

1. INTRODUCTION

While the desktop metaphor is well-understood and represents an effective approach to human-computer interaction for document-oriented 2D tasks, transplanting it to 3D reveals inherent limitations (e.g. [8]). In contrast, interfaces that incorporate true 3D input and output technologies, e.g., six degree of freedom (6DOF) sensors and stereoscopic displays seem more promising, even though the use of advanced interface devices does not guarantee a superior user interface.

We present a system that uses transparent props for two-handed interaction on the Barco BARON [4] Virtual Table (VT), a tabletop VR display based on a workbench metaphor [14]. The hand-held transparent props are a pen and a pad and related to earlier research on the Personal Interaction Panel (PIP) [24], an

¹Vienna University of Technology, Institute of Computer Graphics, Schönbrunner Strasse 7/A/1, A-1040 Vienna, Austria (email: dieter@cg.tuwien.ac.at)

²Fraunhofer Center for Research in Computer Graphics, Inc., 321 S. Main St., Providence, RI 02903, USA (email: mencarna@crcg.edu)

³Imagination GmbH, Schönbrunner Strasse 7/A/1, A-1040 Vienna, Austria (email: zsolt@cg.tuwien.ac.at)

augmented reality interface. While augmented reality systems use semi-transparent or video-based head-mounted displays to overlay computer graphics onto real-world objects (e.g., [10] or [3]), our system overlays transparent physical props onto the back-projected display of the VT. to achieve a kind of inverse augmented reality, which we call *augmented VR*. The VT thereby provides an enhanced workspace with capable multipurpose tools. As Wloka & Greenfield [27] point out, the tactile feedback that the physical props provide makes the tools feel real.



Figure 1: The transparent pen and pad props.

Our system unifies several previously isolated approaches to 3D user interface design, such as two-handed interaction and the use of multiple coordinate systems, but more importantly it allows for the experimentation with the affordances [17] of transparent props that—with the exception of [25]—are generally unexplored. Our interface supports the following important features:

- two-handed interaction
- multi-purpose physical props
- embedding 2D in 3D
- use of multiple coordinate systems (i.e., of the table and the pad)
- transparent tools, especially window-tools and through-the-plane tools.

Each of the listed properties allows the design of distinct forms of interaction. This paper describes our efforts to explore these possibilities of transparent props for 3D interaction. After an overview on related work in Section 2, we describe the system setup used for our experiments in Section 3. We then report on the interaction techniques supported by our transparent props in Section 4. Our ideas are illustrated by examples from a *Virtual Landscaping* application developed to depict the capabilities of

our platform. In Section 5 we give some implementation details and finally present results and observations of the system in practice.

2. RELATED WORK

Our approach was originally inspired by the work of Szalavári & Gervautz [24] on the Personal Interaction Panel. This work explored the use of (opaque) pen and pad props in a head-mounted, see-through augmented reality system, called *Studierstube* [21]. Other researchers use pen and pad props, though either in fully immersive or desktop setups: Sachs et al. [19] describe a system for the design of 3D curves and shapes. Angus & Sowizral [2] report on their use of pen and pad props for embedding traditional 2D GUIs in a 3D immersive system. Billinghamurst et al. [7] describe 3D Palette, a virtual content creation tool using pen and pad props in a fishtank VR setup.

Several researchers reported on the use of two-handed interaction. For tabletop VR devices, Cutler et al. [9] have developed a two-handed object manipulation framework using two gloves or a glove and a stylus. Other uses of two-handed interaction for object design and manipulation can be found in [16] and [11]. These designs are based upon Guiard's observation of how humans distribute work between the two hands [12].

The window-based tools we have developed are related to the toolglass and magic lenses proposed by Bier et al. [6] and extended to 3D by Viega et al. [26], but their approach has some drawback in terms of generality and is not fully embedded into a VR system. Our window-based tools—having real extension into all 3 dimensions—share the goals of 3D magic lenses, but are based on the more flexible implementation of SEAMS originally developed for navigation of virtual environments [20].

Our work also shares aspect with both the active and passive lens of the metaDESK [25]. The metaDESK passive lens is a transparent prop, but does not use stereoscopic graphics and is not used for general-purpose interaction like the props in our system.

Wloka & Greenfield [27] point out that using tools are equally expressive as using one's hands. They propose the use of a one-handed multi-function tool, the virtual tricorder, which inspired our work as well.

Finally, Pierce et al. [18] report on image-plane interaction techniques for immersive virtual environments and let users interact with 2D projections of 3D objects, an approach related to our through-the-plane metaphor.

3. SYSTEM SETUP

The system we have developed uses the Barco *Baron* Virtual Table as its display device. This device offers a 53"x40" display screen built into a table surface and connects to an SGI Indigo2 Maximum Impact workstation. Together with CrystalEyes shutter glasses from StereoGraphics, a large stereo display of very high brightness and contrast is available.

The transparent props we use are an 8"x10" Plexiglas sheet and a large, pen-shaped, plastic tube (Figures 1,2) which is additionally fitted out with a button. Both props as well as the shutter glasses are equipped with 6DOF trackers (Ascension Flock of Birds) for position and orientation tracking. For details on tracker calibration, refer to Section 5.

Using the information from the trackers, the workstation computes stereoscopic off-axis projection images that are perspective correct for the user's head position. This property is

essential for the use of augmented VR, as the physical props and their virtual counterparts have to appear aligned in 3D.

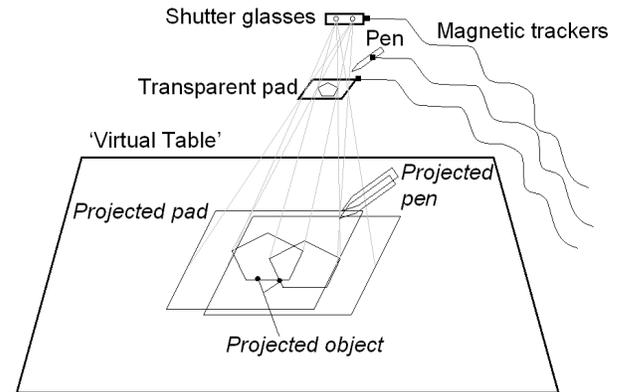


Figure 2: The Virtual Table's display creates the illusion of graphics aligned with the pen and pad.

The material for the pen and pad was also selected for minimal reflectivity, so that with dimmed lights—the usual setup for working with the VT—the props become almost invisible. While they retain their tactile property, in the user's perception they are replaced by the graphics from the VT. Our observations and informal user studies indicate that virtual objects can even appear floating above the Plexiglas surface, and that conflicting depth cues resulting from such scenarios are not perceived as disturbing. Conflicts occur only if virtual objects protrude from the outline of the prop as seen by the user because of the depth discontinuity. The most severe problem is occlusion from the user's hands. Graphical elements on the pad are placed in a way so that such occlusions are minimized, but they can never be completely avoided.

The pen was chosen to be relatively large to provide room for graphics displayed inside the pen. In that way, the pen also provides visual feedback such as showing the tool is currently associated with. So far, however, we have made only basic use of this capability and have instead focused on the pad as a carrier for the user interface.

4. THE TRANSPARENT PROPS' DESIGN SPACE

The focus of our work is to explore the user-interface and interaction possibilities of the transparent pad as a distinct object. While the two-handed pen-and-pad metaphor is asymmetric [12] and the pad is assigned the more "passive" role (e. g., it is held in the non-dominant hand), it has much more interesting affordances than the pen. Pen and pad have a relationship similar to mouse pointer and window in a conventional desktop system. However, the difference to the desktop is not only that pen and pad operate in 3D, but also that the pad is directly controlled by the user's non-dominant hand and can therefore additionally be used as an active tool.

The pad therefore represents an embedding of 2D in 3D, as already pointed out by Angus & Sowizral [2]. Yet its possibilities extend far beyond that by *combining* several individual metaphors:

- *Tool and object palette:* The pad can carry tools and controls, much like a dialog box works in the desktop world (as e.g. in Smartscape [13]). It can also offer collections of 3D objects to choose from.
- *Window tools:* As the user can see through the pad into the scene, the pad becomes a see-through tool (as e.g. the Virtual Tricorder [27]).
- *Through-the-plane tool:* The user can orient the “window” defined by the pad and then manipulate objects as seen through the pad, i. e. manipulate the 2D projections of objects on the pad.
- *Volumetric manipulation tool:* The pad itself can be used for active object manipulation (as e.g. the WIM [22]) exploiting the fact that the pad has a spatial extent (unlike the point represented by the pen tip).

These options co-exist in the design space of our user interface and together form a very powerful and general framework for 3D interaction. Due to the fact that the physical and geometric properties of the pad are of very basic nature, it is possible to use all the metaphors mentioned above for application tasks without confusing the user. Our transparent props form a two-handed multi-purpose tool in the spirit of Wloka & Greenfield [27].

4.1 Tool and Object Palette

In its basic use, the pad serves as a palette offering various tools and controls. The pad resembles a dialog box in a desktop system by grouping various application controls such as buttons, sliders, dials etc. Since the pad is hand-held, it is always in convenient reach for the user, which is an advantage if working on different areas of the table. It is easy to remember where the controls are, and the pad can even be put aside temporarily without causing confusion.

Controls are manipulated with the pen, which is also used to select tools. The active part of a chosen tool is generally associated with the pen, while the pad acts as a passive counterpart for many tools.



Figure 3: In its basic function, the pad serves as a palette for tools and controls. Shown is an RGB color selection tool.

The basic mode of our sample landscaping application is object placement. The pad serves as an object browser presenting a collection of objects to select from. Objects are then dragged and dropped into the scene via direct 3D manipulation. Additional controls—some implemented as space-saving pop-up button bars—allow to scale, colorize, and delete objects. 2D controls and 3D direct manipulation naturally blend as the pad represents a 2D surface similar to many real-world control containers of other

application areas (e. g., a remote control, a radio, a dishwasher’s front panel).

Another interesting property is the multiple surfaces of reference with which the user simultaneously interacts, a fact also observed as being beneficial by Ullmer & Ishii [25]. A sample use is the drag and drop operation from the pad to the table space. We make further use of this property with the window and through-the-plane tools.

4.2 Window Tools

Because it is transparent, our pad prop invites users to look through it. Consequently, we chose to experiment with a set of functions which we call window tools. Conceptually, they are very similar to 3D magic lenses introduced by Viegas et al. in [26]. However, we have their work in two significant ways: First, the underlying implementation does not have the limitations of the original work (see Section 5) and have real extension into all 3 dimensions. Second, our two-handed interaction allows us to manipulate objects seen *through* the lens instead of just the magic lens itself. Our window tools are therefore more related to the toolglass of Bier et al. [6]. This approach is not unlike that of a watchmaker using a magnifying glass together with other tools—a task that naturally fits into a workbench-like environment.

Instead of manipulating controls and objects on the pad, the user manipulates objects on the table surface *under* the pad, which divides the table space into two ‘design’ spaces.

In our landscaping application, we have implemented a cable TV tool that provides the user with X-ray vision (Figure 4). The user can look under the surface of the landscape representing an island (using wireframe rendering) and use the pen to lay wire and connect houses to a cable TV network. The X-ray tool is bound to the *backside* of the pad, making use of the pad’s two-sided property, thus having the X-ray tool always available. (Which side of the pad the user looks at is easily determined by examining the pad’s normal vector).

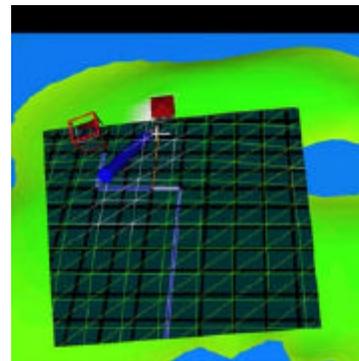


Figure 4: The cable TV routing tool is a special “X-ray” view attached to the back of the pad and allows the placement of wires underneath the island.

While the X-ray tool is an example of a modified view of the environment, a window can also show different content. Windows in the desktop world usually show different content: multiple windows can either be entirely unrelated, or they can show data from different points in space (different viewpoints) or time (different versions). CAD systems normally use four windows

with different viewpoints, and text tools like *xdiff* show a side-by-side comparison of different versions of data.

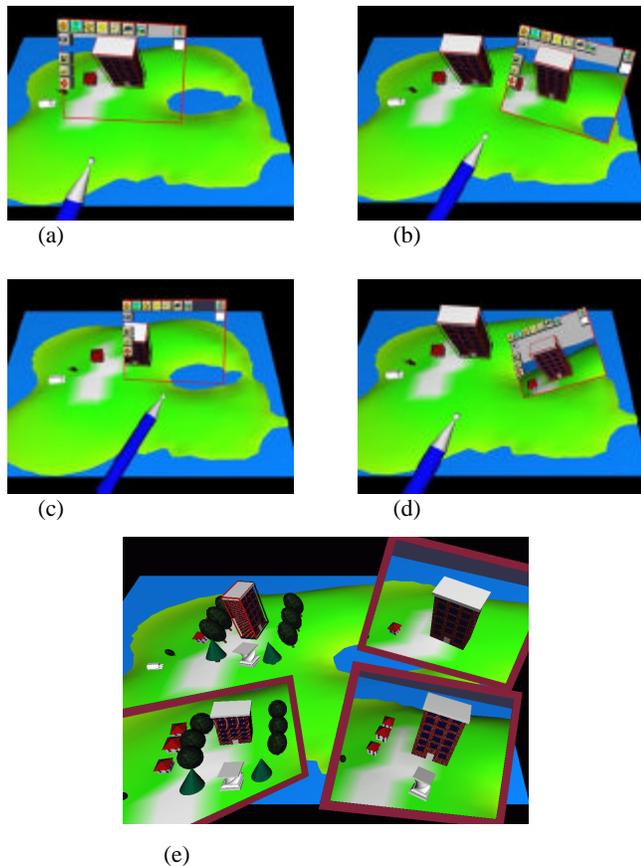


Figure 5: The snapshot tool allows the user to manage a collection of scenes that are viewed from different perspectives and in different stages of development. Note how the scene in the snapshots is not flat, but a real 3D view (compare a to b and d), how a scene variant is visible as a snapshot for comparison (c), and how multiple snapshots can be kept, floating in windows above the virtual scene (e).

We built this idea into our landscaping application using a *snapshot* facility. In normal mode, the view through the window (pad) is identical to the normal scene. However, a particular view (or more precisely, viewpoint) of the scene can be locked on the pad (Figure 5a). This snapshot not a flat photograph, but a real 3D scene that can be viewed from an arbitrary angle and distance by moving the pad or one's head (compare Figure 5b and d to Figure 5a).

Such a snapshot may be decoupled from the pad and left floating in the scene at any position, and possibly be picked up again later. By strategically placing multiple of such snapshots in the scene, a user can inspect multiple views at once from inside a virtual environment, a strategy equivalent to the aforementioned multiple views of CAD systems.

Changes to the objects are reflected in all views simultaneously. However, if the user indicates so, the scene observed through the window can be isolated from the scene shown on the table and from other windows' scenes; thus multiple individual scenes are seen simultaneously. This feature resembles a multiple document

interface from the desktop world, an aspect that to our knowledge has not been explored for VR system so far.

When scenes are isolated in multiple windows, changes to one scene are not reflected in another scene. It is therefore possible to modify the main scene on the VT while the scene in the window remains the same: it becomes a historical reference.

For the landscaping applications, multiple versions of development (or possible design alternatives) can be presented side by side with little effort by the user. This feature is potentially very useful for any kind of 3D-design application. By picking up a floating window that is carrying a particular variant of a scene and unlocking the frozen viewpoint of the window (i.e. the scene through the window is again rendered with the same viewpoint as the scene on the VT), a special kind of portable magic lens for in-place comparison of two variants is created. An example is shown in Figure 5c, where the large building has been deleted in the main scene but is still visible through the window tool.

The possibilities of the snapshot tool are summarized in Figure 6 in the form of a state diagram.

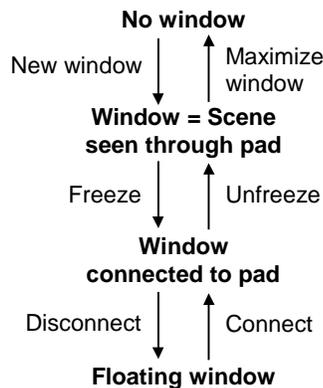


Figure 6: State diagram for managing scenes using the window tools of the landscaping application.

4.3 Through-The-Plane Tool

The look-through affordance of the transparent pad allows the development of yet another class of user interface tools that we call *through-the-plane tools*. They are related to the image plane techniques reported by Pierce et al. [18]. Image plane techniques manipulate 3D objects based on their 2D projection on a plane perpendicular to the line of sight. The pad as a through-the-plane tool differs from this approach in two important aspects:

1. The 2D plane onto which objects are projected is easily manipulated by moving or rotating the pad without the need to move one's point of view.
2. The physical surface of the pad provides a clear definition of the 2D manipulation plane and a tactile surface for making gestures with the pen. (Image plane techniques require a user to make hand gestures in the air without a clearly defined depth of the plane.)

As a consequence of these properties, we have not experienced problems with ambiguities resulting from the stereo projection as reported in [18], although the problem itself remains.

In the landscaping application, we have implemented two tools using the pad as an through-the-plane tool. The first tool is a context sensitive information and manipulation dialog. The user may point the pad into the scene, and the object closest to the tool's center (in the 2D space of the tool) is selected. The object's description is displayed, and context-sensitive controls are displayed on the pad.

In Figure 7, different collections of colorize buttons appear, depending on the type of an object.



Figure 7: The context sensitive tool uses 2D manipulation through the pad. Depending on the position of the pad, objects in the scene get selected, and context sensitive color controls are offered.

Many desktop applications, such as illustration programs, use context-sensitive menus and toolbars that appear and disappear as different objects are selected. The context-sensitive tool brings these possibilities into a VR system. Note that context-sensitive manipulation requires two steps in a one-handed desktop system: The user selects an object, looks for context-sensitive controls to appear somewhere on the screen, and then manipulates the controls. Although marking menus as proposed by Kurtenbach & Buxton [15] are an already much more effective one-handed interaction, they still require the employment of the user's hand for menu marking and then selection. In contrast, only one two-handed step is required in our system, yet controls still always appear near the selected object. Manipulation of pad and pen can be almost instantaneous and is cognitively more similar to context-sensitive pop-up menus, but without the corresponding disadvantages (e.g., display often obscured by menu, mouse button must be held and cannot be used for interaction).

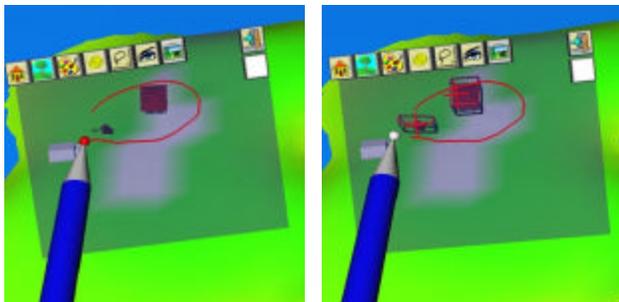


Figure 8: The lasso tools uses the pad as a plane through which objects in the scene are targeted. Instead of selecting objects in the scene, they can then be selected through a 2D circular gesture on the pad. An outline drawn on the pad being held into the scene defines a conical sweep volume that has its tip in the eye point

and its contour are selected (Figures 8,9).

Again, the lasso tool is just one example for a wide design space of tools based on 2D gestures for 3D objects (e.g., objects may be deleted by "crossing them out"). The through-the-plane tool allows us to reuse all the ideas for 2D manipulation of 3D that are known in the desktop world (cf. e. g., draggers and manipulators of Open Inventor [23]). It remains to be verified, however, in which cases this 2D manipulation is more capable than direct 3D manipulation. From our observations we conclude that the power of such 2D gesture tools lies in manipulation at-a-distance, for example when attempting to manipulate objects on one side of the table when standing at the other side.

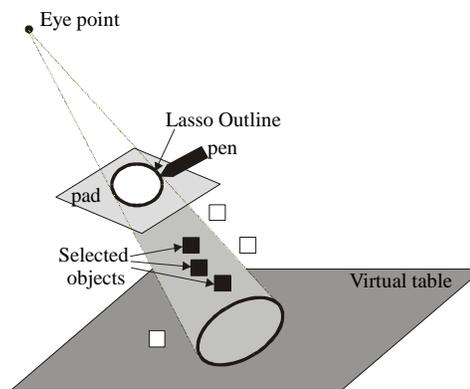


Figure 9: The lasso defines a conical sweep volume to select objects.

4.4 Volumetric Manipulation Tool

Most of the tools we have described so far use the pad to provide the context or frame of reference with the pen (more specifically, the pen tip) being the active part, quite in the spirit of Guiard's observations [12]. However, the pad can be an active (one-handed) tool in its own right.

What sets the pad apart from conventional 3D manipulation tools like a bat, wands, stylus, or buttonball, is its dimension: all these devices have a zero-D (point-shaped) "hot spot" for triggering actions. A laser pointer like tool (which is a popular metaphor for selecting objects at a distance) uses a ray and therefore has a dimension of one. Errors introduced by human inaccuracy make it difficult to perform precise manipulation with tools of essentially no spatial extent, which lack correspondence to real world tools. This is why techniques such as 3D snap-dragging [5] were developed to overcome the mentioned difficulties.

Instead of artificially enhancing the input, we propose to use a tool with a spatial extent, which more naturally resembles real world tools. The 2D surface of the pad can serve such a purpose.

As an example, we have implemented a *fish net* selection tool for the landscaping application. By sweeping the tool through the scene, the user may select objects (Figure 10, top), which are all objects that are intersected with the pad. Since it is undesirable for the user's landscaping efforts to be destroyed as a result of actual objects becoming caught in the fish net, small 3D replicas of the objects are "caught" in the net (or rather, shown on the pad's surface). The replicas are placed in the position on the pad where the object was penetrated, and an "arrange" button aligns the replicas in a regular grid for better overview (Figure 10, bottom).

We have found that sweeping a path with the pad is surprisingly effective for the selection of objects that lie in front of the projection plane of the table, especially when a large number of objects must be selected quickly but selectively. We attribute this ease of usability to the users' real world experience with similar tools.

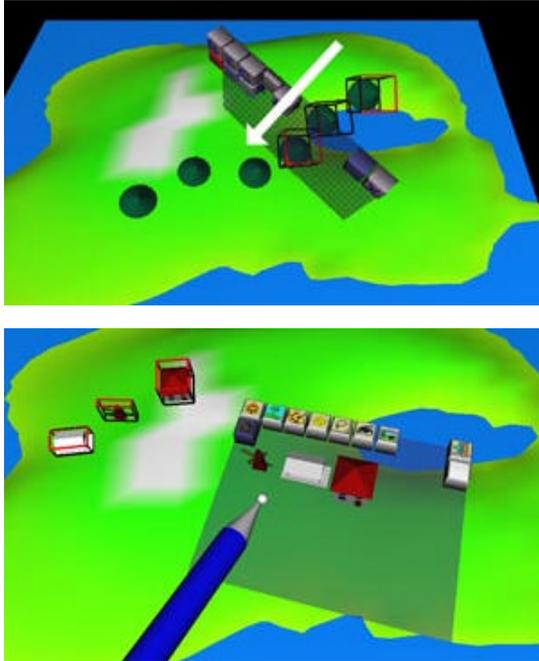


Figure 10: The fish net tool makes use of the pad as a tool with spatial extent. By sweeping it through the scene, objects are selected (top image) and replicas of the objects appear on the pad for further manipulation.

Sometimes it may happen that an object is involuntarily selected together with others. If this occurs, the small replicas on the pad can be discarded by wiping them off the pad, and the corresponding object becomes deselected.

Although we have not yet implemented them, we have imagined several other volumetric manipulation tools that could be used, such as a shovel, a ruler, and a rake. Another possible area of applications are deformation tools for objects made of clay (similar to features found in MultiGen's SmartScene [13]).

5. IMPLEMENTATION

Software architecture. Our system is based on the *Studierstube* [21] software framework. It is realized as a collection of C++ classes extending the Open Inventor toolkit [23]. Open Inventor's rich graphical environment approach allows rapid prototyping of new interaction styles, typically in the form of Open Inventor node kits. Tracker data is delivered to the application via an engine class, which forks a lightweight thread to decouple graphics and I/O. Off-axis stereo rendering on the VT is performed by a special custom viewer class. Open Inventor's event system has been extended to process 3D (i. e., true 6DOF) events, which is necessary for choreographing complex 3D interactions like the ones described in this paper. The .iv file format, which includes our custom classes, allows convenient scripting of most of an application's properties, in particular the scene's geometry. Consequently very little application-specific

C++ code—mostly in the form of event callbacks—was necessary.

Calibration. Any system using augmented props requires careful calibration of the trackers to achieve sufficiently precise alignment of real and virtual world, so the user's illusion of augmentation is not destroyed. With the VT this is especially problematic, as it contains metallic parts that interfere with the magnetic field measured by the trackers. To address this problem, we have adopted an approach similar to the one described by Agrawala et al. [1] and Krüger et al. [14]: The space above the table is digitized using the tracker as a probe, with a wooden frame as a reference for correct real-world coordinates. The function represented by the set of samples is then numerically inverted and used at runtime as a look-up table to correct for systematic errors in the measurements.

Window tools: The rendering of window tools differs from the method proposed by Viega et al. [26] in its use of hardware stencil planes. After a preparation step, rendering of the world "behind the window" is performed inside the stencil mask created in the previous step, with a clipping plane coincident with the window polygon. Before rendering of the remaining scene proceeds, the window polygon is rendered again, but only the Z-buffer is modified. This step prevents geometric primitives of the remaining scene from protruding into the window. For a more detailed explanation, see [20].

6. CONCLUSIONS AND FUTURE WORK

We have presented a system that uses transparent props—the pen and pad—for two-handed interaction with the Virtual Table, a desktop VR system. The system exploits the fact that the VT can display 3D graphics aligned with the props, turning them into multi-purpose tools. In this sense, transparent props seem even to be a tool for the guiding person in a Surround-Screen Projection-Based Virtual Reality System (SSVR), who's viewpoint is tracked, and therefore in correct stereoscopic relation to the interface on the panel's surface. We consider such a configuration an interesting next step for our research.

We have explored and prototyped various interaction metaphors, most of which are inspired by the physical properties of the props and analogies to the desktop metaphor. Our experiments have led us to believe that the rich set of user-interface designs developed for the desktop world in the last decade can be transposed to VR systems if proper attention is paid to the requirements of 3D.

Our system was informally tested with several users, most of which had computer (desktop) experience but little experience with VR systems. They generally found our design very appealing and were able to perform simple landscaping tasks after a few minutes of initial instruction. We did not observe any difficulties in understanding the tools. Complaints mainly addressed technical inadequacies like tracker error, lag or frame rate. Fatigue resulting from prolonged use of the props did not seem to be an issue. However, since most test sessions did not last longer than 20 minutes, this usability aspect will require further investigation. One significant disadvantage we found lies in the restriction of the VT to a single head-tracked user, as oftentimes multiple users wanted to use the system concurrently. As a side note, a possible solution to this problem is presented in [1] for two users yet the described approach probably does not scale beyond a few users.

A promising area of future work encompasses the window tools we have discussed in Section 4.2. The snapshot tool built into the landscaping application makes only very basic use of the

possibilities of window tools. We observe that there is a trend in computer systems towards "browser" tools that invoke adequate representations for different flavors of multimedia data, and we speculate that windows in the style we have shown may prove to be an adequate metaphor to organize data in a browser for 3D scenes. Furthermore, the windows can also serve as containers for distinct 3D applications, with possibilities such as object drag and drop between them. We also intend to explore the possibilities of creating a *workspace*, the 3D equivalent to a multi-windows desktop.

ACKNOWLEDGMENTS

This work has been sponsored by the Fraunhofer CRCG Student and Scholar Exchange Program (SSEP) and the Austrian Science Foundation (*FWF*) under contract number P-12074-MAT. Special thanks to Michael Gervautz for supporting the way to this research with the PIP, and to Anton Fuhrmann, Markus Krutz, Hermann Wurnig and Andreas Zajic for their contributions to the implementation.

REFERENCES

- [1] M. Agrawala, A. Beers, B. Fröhlich, P. Hanrahan, I. McDowall, M. Bolas: The Two-User Responsive Workbench: Support for Collaboration Through Individual Views of a Shared Space. Proceedings of SIGGRAPH, 1997.
- [2] I. Angus and H. Sowizral: Embedding the 2D Interaction Metaphor in a Real 3D Virtual Environment. Proceedings SPIE, vol. 2409, pages 282-293, 1995.
- [3] M. Bajura, H. Fuchs, and R. Ohbuchi. Merging Virtual Objects with the Real World: Seeing Ultrasound Imaginery within the Patient. Proceedings of SIGGRAPH'92, (2):203-210, 1992.
- [4] Barco BARON, URL: <http://www.barco.com/projecti/-products/bsp/baron.htm> 1997.
- [5] E. Bier: Snap-dragging in three dimensions. Proceedings of the 1990 Symposium on Interactive 3D Graphics, pp. 193-203. ACM SIGGRAPH, March 1990.
- [6] E. Bier, M. Stone, K. Pier, W. Buxton, and T. DeRose. Toolglass and Magic Lenses: The See-through Interface. Proceedings of SIGGRAPH'93, pages 73-80, 1993.
- [7] M. Billingham, S. Baldis, L. Matheson, and M. Philips. 3D Palette: A Virtual Reality Content Creation Tool. Proceedings of ACM VRST'97, pages 155-156, 1997.
- [8] S. Bryson and C. Levitt: The virtual windtunnel: An environment for the exploration of three-dimensional unsteady flows". Proceedings Visualization'91, pages 17-24, 1991.
- [9] L.D. Cutler, B. Fröhlich, and P. Hanrahan: Two-Handed Direct Manipulation on the Responsive Workbench. Proceedings of SIGGRAPH Symposium on Interactive 3D Graphics '97, RI, USA, pages 39-43, 1997.
- [10] S. Feiner, B. MacIntyre, and D. Seligmann. Knowledge-Based Augmented Reality. Communications of the ACM, 36(7):53-62, 1993.
- [11] J. Goble, K. Hinckley, R. Pausch, J. Snell, and N. Kassel: Two-Handed Spatial Interface Tools for Neurosurgical Planning. IEEE Computer, 28(7):20-26, 1995.
- [12] Y. Guiard. Assymetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as Model. Journal of Motor Behaviour, 19(4):486-517, 1987.
- [13] Homan. SmartScene: Digital Training - Learn the System by Being Part of the System. Technical report, available from <http://www.multigen.com/>
- [14] W. Krüger, C. Bohn, B. Fröhlich, H. Schüth, W. Strauss, and G. Wesche: The Responsive Workbench: A Virtual Work Environment. IEEE Computer, 28(7):42-48, 1995.
- [15] G. Kurtenbach, G. and W. Buxton: User learning and performance with marking menus. Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems (1994), pp. 258-264.
- [16] D. Mapes and J. Moshell: A Two-Handed Interface for Object Manipulation in Virtual Environments. Presence, 4(4):403-416, 1995.
- [17] D. Norman: The Psychology of Everyday Things. New York, Basic Books, 1988.
- [18] J.S. Pierce, A. Forsberg, M. J. Conway, S. Hong, R. Zeleznik, and M.R. Mine: Image Plane Interaction Techniques in 3D Immersive Environments. Proceedings of SIGGRAPH Symposium on Interactive 3D Graphics '97, RI, USA, pages 39-43, 1997.
- [19] E. Sachs, A. Roberts, and D. Stoops: 3-Draw: A Tool for Designing 3D Shapes. IEEE Computer Graphics & Applications, pages 18-26, 1991.
- [20] D. Schmalstieg, G. Schaufler: Sewing Virtual Worlds Together With SEAMS: A Mechanism to Construct Large Scale Virtual Environments. Technical Report TR-186-2-87-11, Vienna University of Technology, 1998.
- [21] D. Schmalstieg, A. Fuhrmann, Z. Szalavari, M. Gervautz: "Studierstube" - An Environment for Collaboration in Augmented Reality. Extended abstract appeared Proc. of Collaborative Virtual Environments '96, Nottingham, UK, Sep. 19-20, 1996. Full paper in: Virtual Reality - Systems, Development and Applications, Vol. 3, No. 1, pp. 37-49, 1998.
- [22] R. Stoakley, M. J. Conway, and R. Pausch: Virtual Reality on a WIM: Interactive Worlds in Miniature. Proceedings 1995 Conference on Human Factors in Computing Systems (CHI'95), pages 265-272, 1995.
- [23] P. Strauss and R. Carey: An Object Oriented 3D Graphics Toolkit. Proceedings of SIGGRAPH'92, (2):341-347, 1992.
- [24] Zs. Szalavári and M. Gervautz: The Personal Interaction Panel - A Two Handed Interface for Augmented Reality. Computer Graphics Forum (Proceedings of EUROGRAPHICS'97), 16(3):335-346, 1997.
- [25] B. Ullmer and H. Ishii: The metaDESK: Models and Prototypes for Tangible User Interfaces. In Proceedings of ACM UIST'97, Banff, Alberta, Canada, pages 223-232, 1997.
- [26] J. Viegas, M. Conway, G. Williams, and R. Pausch: 3D Magic Lenses. In Proceedings of ACM UIST'96, pages 51-58. ACM, 1996.
- [27] M. Wloka and E. Greenfield: The Virtual Tricoder: A Uniform Interface for Virtual Reality. Proceedings of ACM UIST'95, pages 39-40, 1995.