# Interaction Techniques for 3D Modeling on Large Displays

Tovi Grossman[1,2], Ravin Balakrishnan[1,2], Gordon Kurtenbach[1,2], George Fitzmaurice[1], Azam Khan[1], Bill Buxton[1,2]

| [1]Alias|wavefront | [2]Dept. of Computer Science |
|---|---|
| 210 King Street East | University of Toronto |
| Toronto, Ontario | Toronto, Ontario |
| Canada M5A 1J7 | Canada M5S 3G4 |

tgrossman | ravin | gordo | gf | akhan | buxton @aw.sgi.com

## ABSTRACT

We present an alternate interface for 3D modeling for use on large scale displays. The interface integrates several concepts specifically selected and enhanced for large scale interaction. These include 2D construction planes spatially integrated in a 3D volume, enhanced orthographic views, smooth transitions between 2D and 3D views, tape drawing as the primary curve and line creation technique, visual viewpoint markers, and continuous two-handed interaction.

### Keywords

Large scale displays, tape drawing, 3D modeling, two-handed input, interaction techniques

## 1 INTRODUCTION

Designers in the automobile industry typically create concept sketches of cars on large scale upright surfaces (walls) that preserve a 1-1 or "full-size" scale factor between the sketch and the final physical car. The main reason for these large scale upright sketches is that designers and managers want to determine and evaluate the principal shapes of a design as early in the design process as possible. Working at 1-1 scale is critical to this, if one wants to avoid any unpleasant surprises that might occur if, for example, work were done at a reduced scale or on a conventional desktop sized display. While these measures may seem extreme, it is important to recognize that the product being designed could cost up to $1billion to bring to market. As such, using the right tools and minimizing iterations is of utmost importance.

Traditionally, these sketches have been created not by using pencil and paint, but with a technique called "tape drawing" where black photographic tape is laid onto the drawing surface. Given the large scale size of the sketches, tape drawing has several fundamental advantages over freeform sketching, including the ability to create smooth continuous curves without other physical aids like french curves. Unfortunately, although the resolution and fidelity of these large tape drawings is extremely high, this is difficult to retain when transferring the drawings from the physical media to electronic formats for use within the rest of the automotive design process which utilize various computer applications including 3D

modeling and CAD software. To alleviate the disadvantages of using physical media, it would make sense if these tape drawings were created in electronic media from the start.

In earlier work [1], we designed and implemented a prototype 2D digital tape drawing system that retained much of the fluidity and affordances of the physical technique, including working on a large scale display, while providing the advantages inherent in using electronic media. In discussions with designers at various auto design studios during the development and subsequent demonstration of that prototype system, we obtained a wealth of user feedback and made two main observations.

First, large scale (greater than 8x6 feet) electronic projection displays are being widely deployed in most design studios [3]. However, most of these displays are currently used as passive output devices for evaluating final renderings of designs at large scale and are not used in an interactive manner in the design creation process. One reason for this is that user interfaces for current sketching and modeling software are designed for desktop scale interaction and are often awkward when used interactively on a large display. Our previous tape drawing system, designed from scratch to work on a large display, was seen by the designers as a start towards being able to work interactively at this scale in the early stages of the design process. Clearly there is a need for sketching and modeling applications, with capabilities beyond that of our initial tape drawing prototype, whose user interface is explicitly designed to work with large displays.

Second, traditional tape drawings, and our previous system, only allow for the creation of planar 2D drawings. Since these 2D drawings eventually form the basis of the final 3D model, the designers felt that it would be valuable if they could create the 2D profiles while the underlying 3D model was being simultaneously updated, and easily transition between the 2D and 3D views. Interestingly, the very designers who rejected the use of conventional software in favour of drawing with physical tape, when given an electronic analogue of tape drawing were asking for capabilities that the original physical technique could not provide!

Based on these observations, we decided to extend our initial prototype. The result, whose design and implementation we discuss in this paper, is an interface for 3D modeling that integrates several interesting concepts, including: large scale interaction, 2D construction planes spatially integrated in a 3D volume, enhanced orthographic views, smooth transitions between 2D and 3D views, tape drawing as the primary curve and line creation technique, visual viewpoint markers, and continuous two-handed interaction (Figure 1). Our primary goal for developing the current system was not so much to create a system for actual deployment, but rather a vehicle that would allow us to explore and integrate various user interface techniques suitable for 3D modeling on large scale displays.
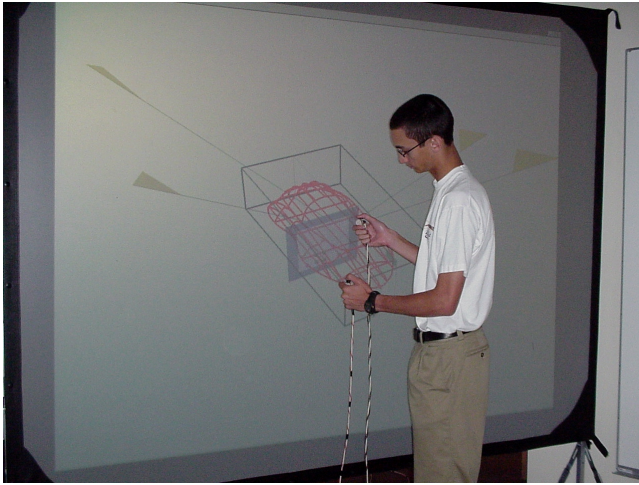
**Figure 1. 3D modeling on a large display.**

## 2 PROTOTYPE SYSTEM

### 2.1 Display

Our implementation uses a Hughes/JVC G1000 digital projector with a true 1280x1024 image back projected onto a collapsible 8x6ft screen. The size of the screen and projector ensures that the system is portable, which is important given that we intend to demonstrate the system at different auto design studios. The 8x6ft screen represents the minimum size for large displays used in the auto industry, and is sufficient for us to implement interaction techniques whose scale of interaction is vastly different from desktop scale interaction. Similar display sizes have also been used in research systems such as Krueger's Videoplace [8].

### 2.2 Input Devices

Since our system heavily utilizes two-handed interaction techniques, we need to be able to sense the position of both hands on the display surface. There are potentially several solutions to this sensing problem. These include optical tracking techniques [4], the use of a transparent digitizing tablet on the display surface, and electromagnetic/ultrasonic trackers. Our prototype uses an Ascension Flock-of-Birds six degree-of-freedom tracker held in each hand. Each tracker is augmented with a single momentary switch. We only use two translational degrees-of-freedom (up/down and lef/right) of the tracker in our prototype. Two cursors on the screen indicate each tracker's position.

### 2.3 Interaction Techniques

#### 2.3.1 2D Construction Planes Spatially Integrated in a 3D Volume

The basic interaction model of our system is to allow for the creation of 3D models by drawing appropriate 2D profile curves. Status-quo modeling applications provide this functionality by having separate 2D orthographic views of the 3D model on which the profile curves are drawn. However, because these orthographic views are typically shown in their own windows separate from the underlying 3D model, it is not easy for a user to see the correspondence between these different views. While users ultimately appear to be able to integrate these views, this is achieved only after much experience with such views. In our system, in order to maintain correspondence with the underlying 3D model being created and to ease the learning process, the 2D profile curves are created on construction planes displayed as sides of a cuboid which acts as a bounding volume for the 3D model within it (Figure 2). Earlier work by Sachs [11] similarly use construction planes in a 3D working volume to create 3D models. Our prototype allows for three primary construction planes, representing the top, side, and front views of the 3D model being created (Figure 2).
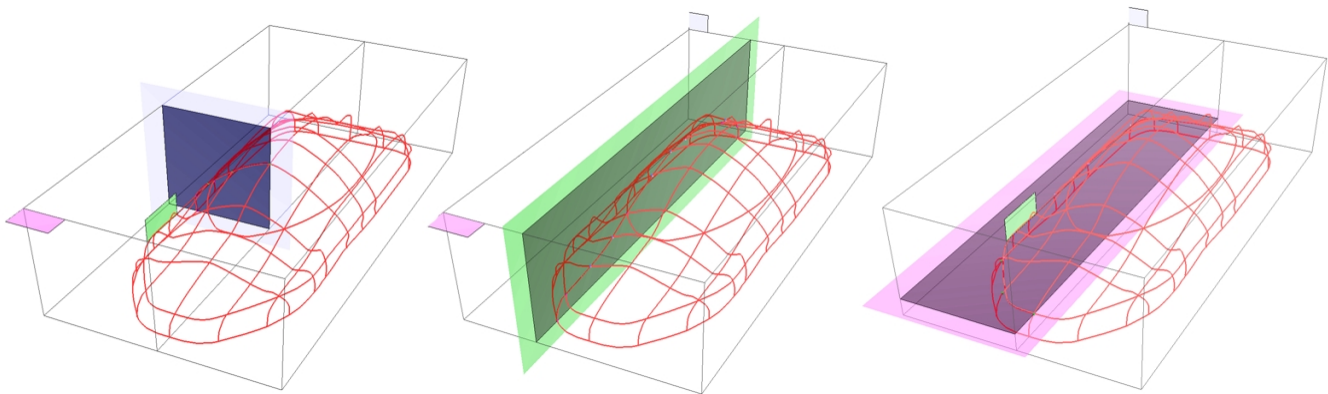


**Figure 2. 2D Construction planes in 3D perspective view. The left image shows the front/back plane, the middle image shows the side plane, and the right image shows the top/bottom (horizontal) plane. Although not visible in this picture, the intersection points between the plane and the 3D model are highlighted as each plane moves through the cuboid space.**

Switching between construction planes is achieved by clicking, using the dominant hand tracker's cursor, on the coloured tab on the corner of the plane. When switching between planes, the previous plane moves away to the edge of the cuboid, while the newly active construction plane moves to the position it was in when it was last active. This reduces clutter in the cuboid since only the active plane is visible while the others are pushed off to the periphery. This "construction plane memory" feature allows the user to switch between different construction planes while retaining the precise location of each plane for later recall.

### 2.3.2  Symmetric Reflections

Depending on the type of 3D model being created, one or more of the construction planes may allow for automatic symmetric reflections about the medial axis. For example, when modeling a car, the top/bottom horizontal construction plane allows for symmetry about the medial axis so that whatever is drawn on the left side is mirrored onto the right.

### 2.3.3  Intersection Points on Construction Plane

The construction planes in our system are semi-transparent so that the 3D model is not obscured as the plane is moved back and forth through the cuboid. To highlight the intersection of the active construction plane with the 3D model and to reinforce the construction plane's position, we display the relevant intersection points on the plane. These points are dynamically updated as the plane is moved.

### 2.3.4  Enhanced Orthographic Views

While we can create 2D profile curves on the relevant construction planes while in a 3D perspective view, it is often more accurate to draw these curves in an orthographic view. In status-quo modeling applications, orthographic views are truly 2D views of the 3D scene. All curves in the 2D view have the same "weight" and are visually not disambiguated regardless of their distance away from the camera. To create and position a new curve in 3D space the user typically has to work in two or more orthographic views. This is the reason for the common configuration of having top, side, and front orthographic views simultaneously visible in most applications. While this configuration is adequate, it results in dividing the user's attention between several views. This problem may be exacerbated when working on a large screen since the user cannot easily view the entire display when up close.

To reduce this divided attention problem and to allow for a single orthographic view to be usable when displayed full size across the entire display, we developed some enhancements to the traditional orthographic view. First, the position of the appropriate construction plane within the cuboid determines the depth position at which new curves are drawn. When the model is viewed orthographically, we display the curves in that view at different grey levels depending on their distance from the current construction plane's position (Figure 3). This essentially provides another dimension of information when in orthographic view, albeit at a coarse granularity, that is not typically available in status-quo applications. Furthermore, we can move the position of the construction plane while still in orthographic view by clicking on any of the curves (at which point the selected curve becomes black and the grey levels on the other curves are adjusted accordingly). Alternatively, a slider widget at the bottom of the screen (Figure 3) allows for positioning of the construction plane. This slider, which is operated by the dominant hand, can work in either continuous mode through the entire space of the cuboid, or in discrete mode where it snaps between the positions of the existing curves. On the slider, these curve positions are represented by tick marks. This discrete sliding mode provides an alternative way to easily move between curves for editing.

Secondly, as in the perspective view, we show the intersection points between the current construction plane with the 3D model. These intersection points are dynamically updated if we move the position of the construction plane. Further, the points are colour coded to indicate the orientation of the curves that are being intersected. This is useful when creating new curves that are to intersect two or more existing curves which may be located in different planes. We have also found these intersection points to be useful in orthographic view as they act as portholes into the perspective 3D model, providing a coupling between the 2D and 3D views even if only one view is visible at a time.

We note that the advantages of our enhanced orthographic views are not restricted to large scale displays, but would likely be useful on standard desktop scale displays as well.
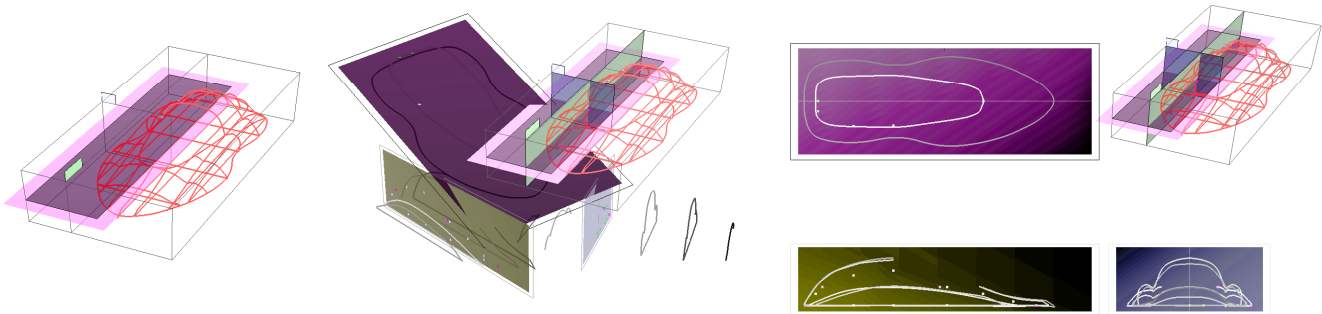
### 2.3.5  Animated Transitions Between Views

The two enhancements described in the previous subsection are examples of our efforts to provide a strong connection between 2D and 3D views in our system. In combination, these two enhancements allow for the user to work in a single orthographic view when needed, without requiring the other views to be present in order to maintain context with the 3D model. However, users will often still need to go back to the 3D view to evaluate the model in its entirety. To visually indicate the relationship between the 2D and 3D views as we transition between them, we smoothly animate the transition between orthographic and perspective views. (Figure 4). We not only animate the window outline as is commonly done in modern 2D window managers, but we also animate the transition of the underlying data. This prevents the often jarring immediate switch between views that is present in status-quo modeling software. The smooth transition provides yet another cue for the user to maintain the appropriate spatial correspondence between the two views.
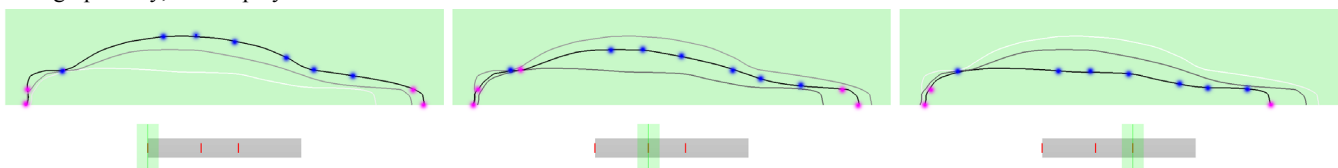


**Figure 3. Enhanced orthographic views. The curves are drawn in different grey levels depending on how far they are in depth from the position of the construction plane. Users can navigate between curves either by selecting the curves themselves or using the slider below. Points of intersection between the curves and other construction planes are indicated as well (Note that these points have been enlarged in this image for clarity).**
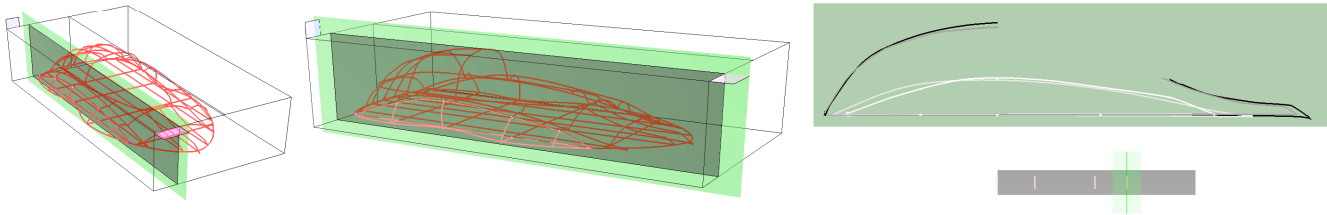
**Figure 4. Animated transition between 3D perspective and orthographic views. The image on the left is the system in perspective view, the image in the middle was captured midway during the animated transition, and the image on the right is the final enhanced orthographic view. The data is animated along with the bounding cuboid volume during the transition. This helps the user maintain context when switching back and forth between views**



**Figure 5. Animated transition between 3D perspective and multiple views (three orthographic and one perspective). The image on the left is the system in perspective view, the image in the middle was captured midway during the animated transtition, and the image on the right is the final multiple view. In the multiple view, moving the construction planes in the perpective view dynamically updates the intersection points and grey levels on the three enhanced orthographic views**

### 2.3.6   Multiple Views

Apart from separate perspective and enhanced orthographic views, our system also supports a multiple view layout (Figure 5, on the right), similar to the four view layout of status-quo systems, where the perspective view is shown along with three orthographic views. Our system is unique in that the orthographic views are enhanced as described above. Sliding the construction planes in the perspective view, either continuously through the cuboid space or discretely between the profile curves, updates the grey levels and intersection points in the appropriate enhanced orthographic view. This facilitates inspection of the model. Again, to maintain 2D/3D spatial integration, transitions between multiple views and single views are smoothly animated (Figure 5).

### 2.3.7   Marking Menus

Activating the transition between 2D and 3D views, switching between discrete and continuous sliding, and other command based actions are accomplished by making a selection on a marking menu [9]. On large displays, conventional menu bars at the edges of the screen are cumbersome to operate given the large movements required of the user. Marking menus, however, have the advantage of appearing at the location of the dominant hand's cursor, requiring very little movement from the user and are thus particularly suitable for use on large displays. Also, once users have learnt the location of the various menu items in the menu, they simply have to make a mark in the direction of that item in order to activate it and do not have to wait for the menu to be displayed. In our system, we activate the systemwide marking menu by making a mark using the dominant hand tracker with its button pressed.

### 2.3.8   Tape Drawing

As discussed in the introduction, the motivation for the present work was an earlier digital tape drawing system we designed which mimicked the functionality of traditional tape drawing techniques. We retain this digital tape drawing technique as the primary method for creating and editing curves in the present system. Figure 6 illustrates the techniques described in this section. When drawing in a construction plane the dominant hand cursor represents the roll of tape. The nondominant hand cursor represents the end of the tape. Both cursors are controlled in a 1-1 manner by the trackers which operate in absolute, linear position control mode. A segment of digital tape, represented as a polyline,

extends between the two cursors. We refer to this as the "unfastened tape segment". Moving the two hands around effectively moves the unfastened tape segment on the screen. The distance between the two hands determine the length of this unfastened tape segment.

In order to fasten portions of the digital tape, the nondominant hand presses the button on its tracker (putting it in "fasten mode"). This corresponds to the act of pressing down on the tape in the physical version. Releasing the nondominant hand button cuts the tape currently being laid at the position of the nondominant hand cursor.

In a manner similar to drawing with physical tape, straight lines are created by holding the dominant hand steady while the nondominant hand, with tracker button pressed, slides along the unfastened tape segment, fastening the tape as it moves. While in the physical version the tape itself serves as a constraint for creating straight lines, our digital version enforces this constraint in software by restricting the nondominant hand cursor to move only along the unfastened digital tape segment, towards the dominant hand cursor. This ensures that once digitally fastened, the digital tape cannot be unfastened inadvertently (without invoking an edit operation to be described later). If the absolute nondominant hand tracker position strays from the unfastened digital tape segment, its cursor position is determined by a simple projection of tracker position to the nearest point on the unfastened tape segment.

Creating curves in our digital system also mimics the physical equivalent. Unlike the creation of straight lines, both hands must move at the same time. Thus, the unfastened tape segment moves while the nondominant hand fastens the tape. The length of the unfastened tape segment effectively serves to regulate the smoothness of the resulting curve. Since the nondominant hand cursor is constrained to move along the unfastened tape segment, a longer segment effectively reduces the range of movement of the fastening point controlled by the nondominant hand, resulting in smoother curves (i.e., a curve whose tangent changes gradually). A short (approaching zero) unfastened tape segment length reduces the technique to the equivalent of free-hand sketching with the non-dominant hand. An interesting aspect of this technique is that it effectively uses constrained two-handed gestures to control the smoothness of curves. In contrast, most computer tools for generating curves rely on mathematical approaches for specifying smoothness.

From the perspective of two-handed interaction, our digital tape laying techniques are interesting in that they allow for the generation of straight lines and curves without a conventional mode switch. The simple act of moving or not moving the dominant hand while the digital tape is being fastened determines whether a straight line or curve is generated. Thus, what we have is a simple curve *and* line drawing technique integrated in a single tool that does not require the user to understand the underlying graphical representation of the curve and its components (control vertices, edit points, etc).

Our tape drawing technique also supports editing of curves and lines, identical to that described in [1]. First, while laying down tape, the user can back-up and undo at will. Second, a cut operation is provided for removing sections of tape after they have been laid down. The reader is referred to [1] for the details.

We note that there are some important differences between our electronic tape drawing tool and the traditional technique using real tape. First, in our current system the collapsible screen we use does not have a rigid surface. As such, unlike in traditional tape drawing, users cannot press against the drawing surface (i.e., the screen). While this is somewhat of a drawback, we have found that users are able to adapt by holding the input devices just above the surface of the screen. The use of a rigid screen would solve this problem. However, this would be at the expense of system portability that is crucial at this stage of our exploration in order for us to be able to demonstrate our system at different design studios worldwide. Another difference between the electronic and traditional media is that in the traditional technique the user gets kinesthetic feedback in the form of tension in the tape, whereas the electronic version provides only visual feedback. While we do not believe that the lack of physical tension in the electronic version is a serious handicap, we have experimented with providing physical tension using spring loaded cords (much like a spring loaded tape measure). Our initial efforts indicate that providing physical tension in this manner is rather different from the tension in real tape and therefore detracts from the drawing task. In some sense, it appears that it is better to not provide any kinesthetic feedback than to provide feedback that is inferior or vastly different to that of real tape.
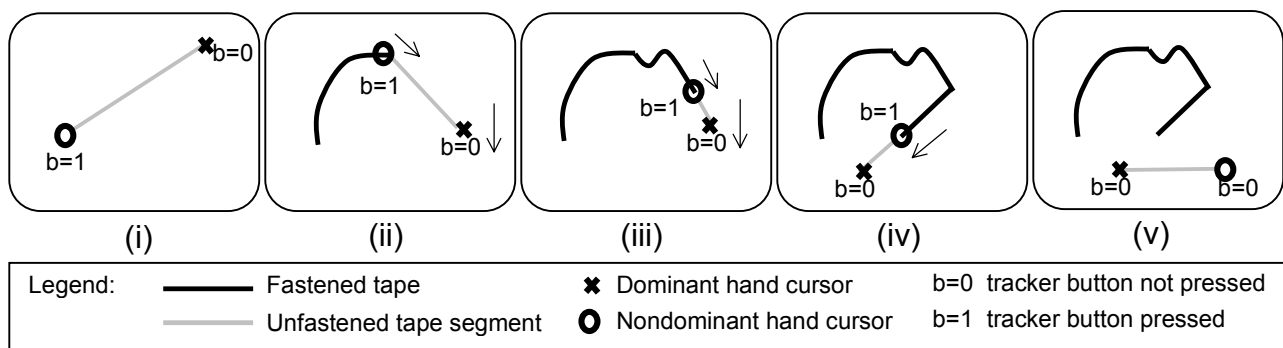


**Figure 6. Tape Drawing. (i) To start taping, the nondominant hand tracker button is pressed. (ii) While moving the dominant hand, the nondominant hand lays down tape as it slides along the unfastened tape segment between the two cursors. Movement of the nondominant hand cursor is constrained to the unfastened tape segment in the direction towards the dominant hand cursor. A long unfastened tape segment results in smooth curves with a gradually changing tangent. (iii). Reducing the length of the unfastened tape segment permits the generation of higher variation curves with a more rapidly changing tangent. The length of the unfastened tape segment can be changed on-the-fly by simply moving the two cursors closer or farther apart. (iv) Switching from taping curves to taping straight lines is achieved by keeping the dominant hand cursor in a fixed position while taping with the nondominant hand. An explicit mode switch is not required. (v) releasing the nondominant hand tracker button cuts the tape.**

### 2.3.9 Camera Control

When in 3D perspective view, we support the usual camera control operations of tumble, pan, and zoom. Based on the results of earlier research which showed benefits in operating the camera with the nondominant hand [2], and in line with theoretical models of bimanual interaction [5], we assign the nondominant hand to control tumbling of the camera, freeing the dominant hand for other operations. Pressing the nondominant hand tracker button invokes camera tumbling which is controlled by movement of the nondominant hand. Panning and zooming are two handed operations. With both tracker buttons pressed, moving the nondominant hand pans the camera, while moving the dominant hand zooms the camera. Other two handed camera controls have been previously explored in 2D by Kurtenbach et. al. [10] and in 3D by Zeleznik et. al. [12, 13]

### 2.3.10 Viewpoint Markers

When creating 3D models, designers typically go back and forth between a few particular views of the scene. In status-quo systems, users are given the ability to define viewpoints and switch between them via a menu or hotkeys. Each preset view is usually abstractly named and no visual indication is provided as to what each view is until the user actually selects that view. In our system, we increase the directness in which these special viewpoints can be set and selected using a new widget we call "viewpoint markers". Essentially, when the user wants to save a particular view for later recall, they make a selection on the system's marking menu. A visual marker flag is then drawn in the scene, pointing in the direction of that camera view (Figure 7). The directionality of each viewpoint marker provides an immediate visual indication to the user as to the approximate view each marker corresponds to, without having to actually select that view. The user can return to any of these views simply by clicking on the appropriate viewpoint marker. Transitions between views are always smoothly animated to maintain visual continuity. Other techniques for saving and restoring camera viewpoints were previously explored by Zeleznik and Forsberg [12].
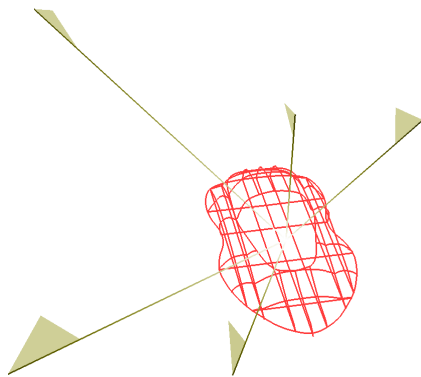


**Figure 7. Viewpoint markers. Viewpoints can be saved using a marking menu, at which point a viewpoint marker flag is added to the scene to visually indicate the direction of that camera view. Clicking on any of the viewpoint marker flags transitions to that view with a smooth animation.**

## 3 DISCUSSION, CONCLUSIONS, and FUTURE WORK

Our design rationale throughout this work was to maximize the potential of the large display and promote integration between 2D and 3D views of the data. At the same time we sought to minimize the level of intrusiveness of user interface components. As such, only user interface widgets that are absolutely necessary are displayed at any one time, resulting in a system that emphasizes the artwork over user interface components. In designing the interaction techniques, rather than inventing everything from scratch, we chose to borrow from previous work and modify or enhance the techniques as required. Some techniques like marking menus and camera controls we use as is, others like orthographic views and animated transitions are enhanced, while some techniques like viewpoint markers are new. Thus, our contribution is not so much in the individual techniques themselves, but in the combination of these techniques into a fluid system for 3D modeling.

While we have not yet formally user tested our system, we have shown it to several designers who have visited our lab. From their reactions and our own experiences in using the system while under development, we feel that some of the techniques are "clear winners" while other areas could use improvement.

Firstly, we believe that the use of tape drawing as the curve and line creation tool significantly reduces the complexity of creating curves that is present in status quo applications. As shown in the accompanying videotape, very smooth curves, with the desired curvature, are easily created with this technique. Also, users only have to use a single tool to create both lines and curves.

Another aspect of our system that works well is the use of animated transitions between views, and our enhanced orthographic views. Both of these increase the coupling between 2D and 3D views of the underlying 3D model and allows users to easily work in 2D without overly sacrificing their ability to comprehend the 3D scene. Other approaches that could assist in 2D/3D integration which we intend to explore include interactive shadows [6].

The use of two handed input throughout the system builds upon previous research [2, 5, 7] which has shown the benefits of bimanual interaction when designed properly. Our nondominant hand camera controls in particular are based on earlier empirical work [2].

Working on a large display poses the problem of not being able to easily operate a keyboard. As such, we are restricted to a few buttons on whatever hand held trackers we use. In our case we designed the entire system to work with just two buttons, one on each tracker, and used marking menus to access commands. In order to increase the number of states that can be triggered by just two buttons, we have considered the idea of using the sequence in which the two buttons are pressed as an additional state. This is not currently needed in our system but is reserved for use when we add features in the future.

Our system currently supports the creation of wireframe models. While this was sufficient for us to explore interaction techniques, adding various surfacing tools is clearly a next step. Tools that allow for sculpting and painting directly on the surface, such as that found in Alias|wavefront's Maya, will need to be explored for use in this context.

We also intend to extend our system to allow for construction planes of arbitrary orientation. Further, the ability to create 3D, non-planar curves will need to be supported.

Finally, our prototype works on a display that is approximately 1/3 the size of a real car. We feel that working with truly 1-1 scale displays and multiple views will result in yet another set of interaction issues that will have to be addressed. For example, imagine having the multiple view layout in our system where each view is full 1-1 size. This would require the user to possibly *walk* around the display in order to interact with different parts of it. The challenges posed by this setup are yet to be investigated.

# 4 ACKNOWLEDGEMENTS

# REFERENCES

1. Balakrishnan, R., Fitzmaurice, G., Kurtenbach, G., and Buxton, W. (1999). Digital tape drawing. *ACM UIST'99 Symposium on User Interface Software and Technology*, pp. 161-169.

1. 2. Balakrishnan, R., and Kurtenbach, G. (1999). Exploring bimanual camera control and object manipulation in 3D graphics interfaces. *ACM CHI'99 Conference on Human Factors in Computing Systems*, pp. 56-63.

3. Buxton, W., Fitzmaurice, G., Balakrishnan, R., and Kurtenbach, G. (2000). Large displays in automotive design. *IEEE Computer Graphics and Applications*, July/Aug 2000, pp. 68-75.

4. Elrod, S., et. al. (1991). Liveboard: a large interactive display supporting group meetings, presentations, and remote collaboration. *CHI'92 Conference on Human Factors in Computing Systems*, pp. 599-607.

5. Guiard, Y. (1987). Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of Motor Behavior*, 19(4), pp. 486-517.

6. Herndon, K., Zeleznik, R., Robbins, D., Conner, B., Snibbe, S., and Dam, A.v. (1992). Interactive shadows. *ACM UIST'92 Symposium on User Interface Software and Technology*, pp. 1-6.

7. Hinckley, K., Pausch, R., Proffitt, D., and Kassell, N. (1998). Two-handed virtual manipulation. *ACM Transactions on Computer-Human Interaction*, 5(3), pp. 260-302.

8. Krueger, M. (1991). VIDEOPLACE and the interface of the future, in *The art of human computer interface design*. B. Laurel, Editor. Menlo Park: CA: Addison Wesley. pp. 417-422.

9. Kurtenbach, G., and Buxton, W. (1993). The limits of expert performance using hierarchical marking menus. *CHI'93 Conference on Human Factors in Computing Systems*, pp. 35-42.

10. Kurtenbach, G., Fitzmaurice, G., Baudel, T., and Buxton, W. (1997). The design of a GUI paradigm based on tablets, two-hands, and transparency. *ACM CHI'97 Conference on Human Factors in Computing Systems*, pp. 35-42.

11. Sachs, E., Roberts, A., and Stoops, D. (1991). 3-draw: A tool for designing 3D shapes. *IEEE Computer Graphics and Applications*, 11(6), pp. 18-26.

12. Zeleznik, R., and Forsberg, A. (1999). UniCam - 2D Gestural Camera Controls for 3D Environments. *ACM Symposium on Interactive 3D Graphics*, pp. 169-173.

13. Zeleznik, R., Forsberg, A., and Strauss, P. (1997). Two pointer input for 3D interaction. *ACM I3DG'97 Symposium on Interactive 3D Graphics*, pp. 115-120.