Boom Chameleon: Simultaneous capture of 3D viewpoint, voice and gesture annotations on a spatially-aware display

Michael Tsang[2,1], George W. Fitzmaurice[1,2], Gordon Kurtenbach[1,2], Azam Khan[1,2], Bill Buxton[1,2]

[1]Alias|wavefront
210 King Street East
Toronto, Ontario
Canada M5A 1J7
{gf | gordo | akhan | buxton}@aw.sgi.com

[2]Dept. of Computer Science
University of Toronto
Toronto, Ontario
Canada M5S 3G4
{mtsang | gf | gordo | buxton}@dgp.toronto.edu

ABSTRACT

We introduce the *Boom Chameleon*, a novel input/output device consisting of a flat-panel display mounted on a tracked mechanical boom. The display acts as a physical window into 3D virtual environments, through which a one-to-one mapping between real and virtual space is preserved. The Boom Chameleon is further augmented with a touch-screen and a microphone/speaker combination. We present a 3D annotation application that exploits this unique configuration in order to simultaneously capture viewpoint, voice and gesture information. Design issues are discussed and results of an informal user study on the device and annotation software are presented. The results show that the Boom Chameleon annotation facilities have the potential to be an effective, easy to learn and operate 3D design review system.

KEYWORDS: 3D navigation, annotation, gesture, voice, spatially-aware display.

INTRODUCTION

In the realm of digital media, three-dimensional data has come of age. The use of 3D models is prolific, ranging from industrial design to scientific visualization. Tools for the creation and modification of complex models have become incredibly sophisticated and powerful. Yet the technology for annotating these models has been left far behind, resulting in a gaping discontinuity in the high-technology workflow. Due to the widespread use of 3D data, an effective annotation system that addresses this gap is likely to prove extremely useful in industry and offers exciting design opportunities for creating a compelling and interactive user experience.

From our work with automotive designers, they have stated that reviews and critiques of digital 3D models can be awkward and unwieldy. An example is a representative automotive design review where managers and designers are grouped together in front of a large display. The manager critiques the design from various viewpoints but must rely on a technician to operate the interface and move the 3D model. Meanwhile, the designers furiously scribble the manager's comments into a notebook, perhaps marking-up images of the 3D model or making quick sketches to capture the manager's intentions. This scenario highlights three major problems:

Viewpoint control – Managers and designers feel frustrated because they cannot directly control the viewpoint. This contrasts with the evaluation of a physical model where anyone is free to move around the model and direct the group's attention.
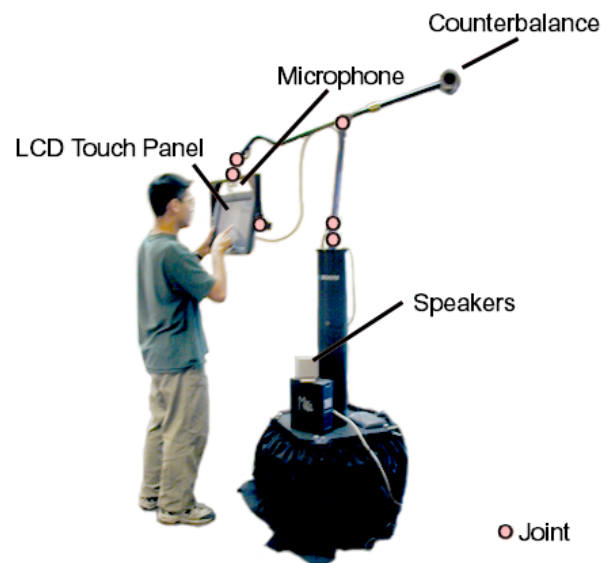

Figure 1. The Boom Chameleon.

Information capture - Designers feel that critique information is lost because attention is divided between recording the annotation and participating in the discussion.

Shared view - Both managers and designers feel misunderstandings could be reduced if notes and annotations could be created and viewed by all parties simultaneously.

Outside the realm of 3D visualization, a substantial body of literature exists on supporting reviews and critiques through digital annotations on text documents, images and other 2D assets. The Wang Freestyle system [19] illustrates an emergent approach of capturing existing work-practice annotation styles, such as marking on documents or simply speaking. A key to the success of this approach is the minimally intrusive capture of information, and one method of minimizing intrusion is to simultaneously capture the different modes of input. In effect, the systems that do this constantly record all data streams, removing the burden of the user having to explicitly engage different annotation tools.

Over the past few years we have been experimenting with a specialized device for 3D visualization that we call the *Boom Chameleon* (see Figure 1). The Boom Chameleon is a spatially-aware device [11, 12], where the display acts as a window onto a virtual world and moving the display in space changes the viewpoint into this virtual world. This idea can also be thought of as a "video camera" metaphor.

Figure 2. Navigating with the Boom Chameleon.

The Boom Chameleon has been designed to be a strong specific device [6], meaning that it performs extremely well in a narrow application domain, in contrast to a weak general device. Viewpoint controls typically found in commercial applications have discrete functions for zooming, panning, tumbling, etc. In contrast, the Boom Chameleon replaces all of these operations with simple physical manipulation of the display. The virtual model remains fixed in physical space while the user moves the display around in that same physical space. This allows an untrained user to move around a 3D model and inspect it from different viewpoints. As Figure 2 shows, when a user moves to the front of the space the front of the virtual car is viewable, moving to the side shows the corresponding side of the car, and moving closer to the center zooms to a close-up of the car. To return to a particular view, the user simply returns to the same physical point in space. Because it uses simple and familiar mechanisms for navigation, the Boom Chameleon is very well suited for allowing non-technical users to control the viewpoint when reviewing and critiquing 3D models.

Given the research on simultaneous capture of 2D annotations and the properties of the Boom Chameleon, this paper explores combining them to create a tool for the effective review and critique of 3D digital models. This integration is intended to address the three major problems brought forward by managers and designers. First, the ease of use and operation of the Boom Chameleon addresses the *viewpoint problem*. Second, information capture techniques from 2D annotation schemes can be adapted to address the *information capture problem*. Lastly, the display will address the *shared view problem* by serving as a common viewing and annotation space.

In this paper we review the related research and then present our system design and implementation details. We next describe the results of several informal user studies. Lastly, we discuss lessons learned, design insights and the general feasibility of our approach.

RELATED RESEARCH

Chameleon Navigation
The Boom Chameleon is based upon the "window in hand" metaphor introduced by Fitzmaurice [11, 12]. A number of other systems have also used this navigation metaphor. One system closely related to the Boom Chameleon is the room-scale Art+Com virtual car installation [1], and a smaller handheld display is used by Small and Ishii to navigate virtual newspaper media and other 2D information [31]. This metaphor has also been used with video cameras on handheld devices to view the physical world with computer graphics superimposed on the display [22, 24, 29].

Other work [7] discusses the relationship between the eye, hand, and display for different devices and the impact on the interactions that they afford. This work highlights the unique properties of Chameleon navigation in relation to other virtual reality approaches such as head-mounted-displays, CAVEs, and the standard workstation configuration.

The WindowsVR system [38] is a cousin of the Boom Chameleon consisting of a touch sensitive LCD panel, which is hung from an overhead arm using a retractable cable. The device comes in two configurations. In the first, only the orientation of the display is sensed and to specify translation, the user must operate two joysticks mounted on the sides of the LCD. In contrast to the Boom Chameleon, this method of interaction breaks the one-to-one spatial mapping between the physical and virtual space, as the user must navigate using a flying metaphor. In the second configuration both the orientation and translation are sensed. However, the display can be cumbersome to operate since it is only counterbalanced vertically.

2D Annotation
There is a wide range of research concerning annotation of 2D media such as images, text, and video. The most relevant work from this area involves free-form hand drawn mark-up and voice annotations in a modeless fashion. The earliest work that makes use of voice input and gesture is Bolt's "Put-That-There" system [4]. However, the Wang FreeStyle system [19] may be the first system to use voice and gesture input to capture voice and ink mark-up for documents in an annotation context. Other systems have followed similar approaches for capturing meetings [24, 27]; collaborative writing [14]; video annotation [20, 34]; note taking [36, 37] and voicemail [13, 35]. Studies of annotation in traditional media such as paper documents suggest that a free-form strategy is attractive since people show great diversity in how they annotate [23, 25]. Furthermore, there has been evidence that spoken annotations can be richer than textual annotations [8], and that users almost universally prefer to interact multimodally, especially when working in the spatial domain [26].

3D Annotation
The notion of annotating 3D scenes has been explored in many different forms. In terms of commercial systems, these are generally designed for traditional monitor, mouse and keyboard configurations and typically use a fairly structured and modal approach to the interface. For example, the Silicon Graphics IRIS Annotator™ [16] allows annotation of 3D models with several media types from fixed viewpoints, employing explicit modalities for input including keyboard text, audio and video. Other research systems have explored similar methods for embedding structured data into the 3D scene. Loughlin and Hughes [21] use virtual "Post-it" text notes and some systems [15, 18] use structured graphics such as pointer arrows or icons to indicate annotations in the 3D scene. Researchers have observed that these structured inputs are not always sufficient [17] and the ability to free-form

draw would be beneficial. Early explorations have been conducted on the ability to draw free-form annotations directly on 3D models [9, 17] and onto 2D drawing planes in a 3D scene [5, 18, 32]. Our work continues to develop along this vein of research.

The Virtual Notepad [28] system uses a head-mounted display and a spatially tracked tablet to control the location of a virtual writing surface in an immersive world. Annotations may be placed in the virtual world by positioning the tablet and writing on its surface. The Boom Chameleon differs from this work in that our display is not head mounted and, consequently, our combination of display/touchscreen allows for a different set of user affordances [7].

The main contribution of this paper over prior work is the combination of a spatially-aware display with the ability to capture voice, gesture and viewpoint information in a continuous, free-form fashion.

DESIGN PHILOSOPHY
Our design philosophy with the Boom Chameleon annotation system is to have annotations that can be created with minimal overhead and user pre-planning. In order to accomplish this we constantly record all streams of annotation data. We believe that the capture of many types of annotation data without requiring explicit action for each data type allows a richer and more fluid set of annotations when compared with a highly structured and modal approach to annotation entry.

BOOM CHAMELEON CONFIGURATION
The Boom Chameleon is a specialized input/output device that is a modified version of the Fakespace BOOM3C™. The BOOM3C consists of a pair of stereo goggles mounted on a position and orientation-sensing armature [10]. In our version, the goggles were replaced with a 15" (16:9 aspect ratio) Silicon Graphics LCD panel (see Figure 1). The armature was then re-balanced in order to have the display float weightlessly in space. With this configuration, the display has a 10-foot spherical range of motion around the center of the device.

To facilitate touch input a customized Elo™ touch screen was overlaid onto the LCD, allowing the hands to operate two modes of input simultaneously. This means that one hand may control the position of the display while the other utilizes the touch screen. To record voice, we attached a podium microphone to the top of the display frame. Standard computer speakers were used for sound output.

ANNOTATION APPLICATION

To facilitate design reviews and critiques, we have developed an annotation application to take advantage of the simple navigation metaphor of the Chameleon. Our graphical interface has been designed to allow for a wide range of grip and hand positions (see Figure 3). The wide button strips running along the left and right sides of the display are accessible to the thumbs from a wide variety of positions. There are also large push buttons running along the top of the screen.

Figure 3. Hand grip and button layout. (a) Flashlight (b) Pen (c) Snapshot (d) Application exit (e) Play (f) Record.

Viewpoint and Voice Capture

From the user's perspective, the view reacts immediately in response to changes in the position and orientation of the display. The microphone above the display is well positioned to capture speech from the primary annotator and from discussions with colleagues who are also viewing the display. While the Record button is active, both viewpoint and voice information are saved as separate data streams, ready for immediate playback.

Gesture Capture

Gestures supported by our system are divided into three categories: pen, flashlight, and snapshot. We support pen mode and snapshots to allow the making of gestures that consist of precise, concrete annotations. We allow flashlight mode for more ephemeral, non-marking purposes. A description of each mode follows:

Pen. We provide the ability to draw lines directly on 3D objects in the scene, which allow for the resulting annotations to be viewable from multiple viewpoints (see Figure 4a,b). Although the Pen mode is the default, either Flashlight or Pen mode can be rapidly toggled by selecting the appropriate button at the top of the screen (see Figure 3a,b).
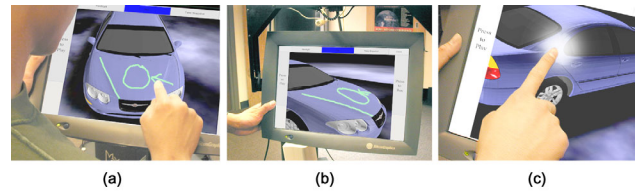
Figure 4. (a) Drawing with pen (b) Inspecting marks from a new view (c) Flashlight feature.

Flashlight. In order to support non-permanent gesture, we implemented a mode that simulates a flashlight placed at the tip of the finger (see Figure 4c). The 'flashlight' is itself invisible, but a spotlight appears in the location at which the user is pointing. We found through preliminary tests that the flashlight was good for indicating large areas of the model, but it was often not precise enough for detailed pointing tasks. Consequently, we decided to indicate the precise location by a small ball floating on the model. Since the ball is directly on the point of interest, a reviewer of the annotation can view the area from different angles and still have the gesture appropriately aligned with the model. It is also possible to infer the original location of the annotator from the shape of the spotlight. When using the 'flashlight', the virtual spotlight is placed above the model one-tenth of the distance between the viewpoint and the model. This allows the user to highlight a larger area of the model when farther away, smaller regions when closer, and the sphere can specify the exact location of features when the user is extremely close.

Snapshots. Snapshots in our system are flat 2D images floating in the 3D scene (see Figure 5). This differs from the approach in [30] where snapshots are windows into 3D volumes. To create a snapshot in our system, the user clicks on the "Take snapshot" button (see Figure 3c), which causes an image of the current viewpoint to be captured and positioned in front of the camera. This also temporarily locks the viewpoint enabling the LCD panel to be used as a writing pad. The pen can be used to write directly on the snapshot while the system continues to capture voice annotations. When finished, the user presses the "Done Snapshot" button, releasing the snapshot image.
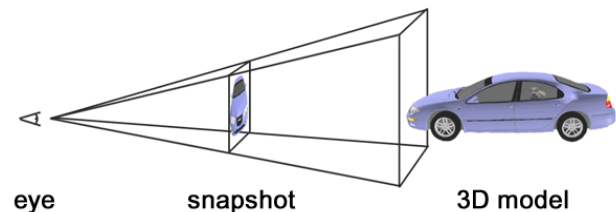
**Figure 5**. Snapshot of the active view.

The snapshot capability is useful for four main reasons. First, although the pen can be used to draw directly on 3D objects, without snapshots there would be no mechanism for drawing where objects do not exist. This can be an especially acute problem in a sparse scene. For example, it is difficult to circle a group of atoms in a molecule (see Figure 6). The snapshot feature allows drawing "in empty space" since the markings appear on the surface of the snapshot.

Second, without using a snapshot, the viewpoint can change while the user is drawing with the pen. This displaces the pen tip relative to the scene and can lead to deviated strokes, a phenomenon that we term *ink drift*. Normally, the viewpoint can be held steady enough to make rough marks. However, ink drift becomes a problem when exact precision is required. The frozen viewpoint of the snapshot helps to alleviate this ink drift problem.

Third, some viewpoints are awkward to reach and to hold in position. Taking a snapshot allows users to "grab" a difficult viewpoint and reposition the display to a more comfortable viewing and writing posture.

Finally, drawing directly on an object can be effective in many situations, yet users may mistakenly assume the marks are being drawn on a plane in 'screen-space' rather than 'object-space'. This results in marks looking correct from the original perspective but extremely confusing and distorted from other viewpoints. For example, a perfect circle may be drawn on the ground plane from one viewpoint, yet from another angle it may appear as a long thin oval. Snapshots allow a user to explicitly choose to draw in 'screen-space' and avoid this problem.
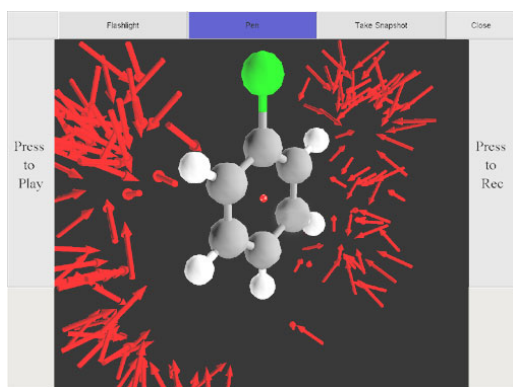

Figure 6. Difficult to draw on sparse 3D scene.

Animated Snapshot Transitions. Releasing a snapshot has two effects. First, it places the snapshot in the 3D scene. Secondly, to preserve visual continuity, a smooth animated transition occurs between the original viewpoint in which the snapshot was taken and the current position of the display (see Figure 7). Depending on the movement of the display while the viewpoint is frozen, it is possible

to have the view suddenly positioned in an unexpected location. Since the abrupt change of context can be very confusing for users, we employ animated transitions to keep users oriented and prevent a visually jarring experience.

Even after the viewpoint is unfrozen, marks can still be made on the snapshots. This "always drawable" design minimizes the mode switching the user has to consider.
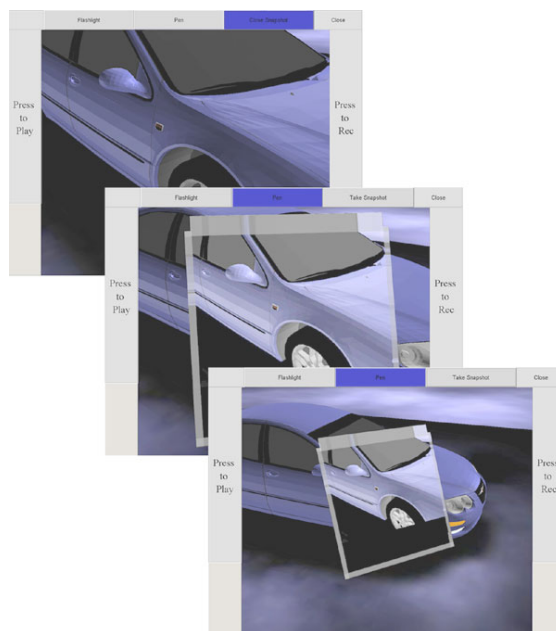
Figure 7. Animated transition after snapshot is released.

In keeping with the concept of omni-viewable widgets for a volumetric display [2], the snapshot frames are double-sided, in order to show the snapshot image and annotation ink on both the front and the back of the image plane (see Figure 8a). Thus, these annotations may be marked and viewed from many vantage points.

(a)   (b)
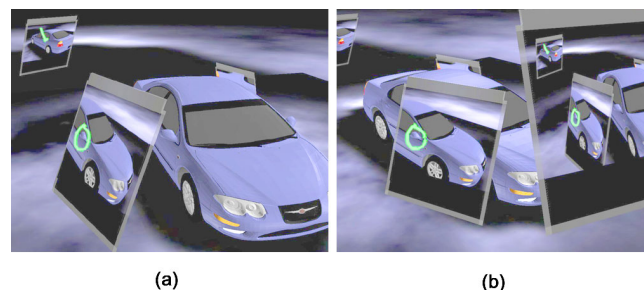
Figure 8. (a) Double-sided, viewable and writable snapshots. (b) Layering snapshots.

A convenient byproduct of spatially embedding snapshots is the ability to layer multiple snapshots (see Figure 8b). Users can take a snapshot of the 3D object with previously created snapshots in view. This allows users to

capture snapshots-within-snapshots to comment on existing annotations within the shared 3D scene. We have noticed that there is a natural layering in which the earliest snapshots are positioned closest to the center of the scene while new snapshots are positioned further away.

Snapshot Titlebars. Populating the 3D scene with many snapshots can clutter and obscure the 3D object. To counteract this effect, we have added the ability to hide the snapshot images. Clicking on the titlebar of a snapshot toggles the visibility of the image. The titlebar is clickable from both the front and back, and is drawn semi-transparently to reduce the problem of obscuring the model. Lastly, while a snapshot is hidden, the position, size and orientation of the residual titlebar provide the user with an awareness of the hidden annotation's scope (see Figure 9).

Figure 9. Toggling snapshot visibility using the titlebar.

Session Management and Workflow
Recording. To begin an annotation session, the user presses the Record button. This activates the capture of all three data streams: view, voice and gesture. The Record button turns red to indicate that data is being captured, and clicking on the button again (see Figure 3f) will end the annotation session.

Playback. To replay the last annotation session, the user may click on the Play button located on the left side of the display (see Figure 3e). The annotation session is then played back in a separate window (see Figure 10).

Figure 10. Playback window.

Note that during playback, the 3D object and annotation data is also presented "live" in the main window. This real-time view of the annotation session allows the user to have alternative viewpoints during playback. This is useful, for example, if the recording is pointing out a flaw with a part of the model and the user wishes to get a closer look, or view the flaw from a different angle. Alternative 3D views have previously been offered for navigation [33] but not for live annotations.

Saving. From the playback window, a user can choose to save the annotation session. A representative thumbnail image of the annotation session is then generated and used as part of the workflow, which is described below.

Workflow. To begin a design review, the user can select a 3D object from a 2D image browser (see Figure 11) running on the Boom Chameleon. Clicking on an image will launch our annotation software with the appropriate 3D model. Saved annotation sessions will also appear in the browser and, if selected, will playback the session. Users can transmit an annotation session to their colleagues by sending the data as an e-mail attachment.
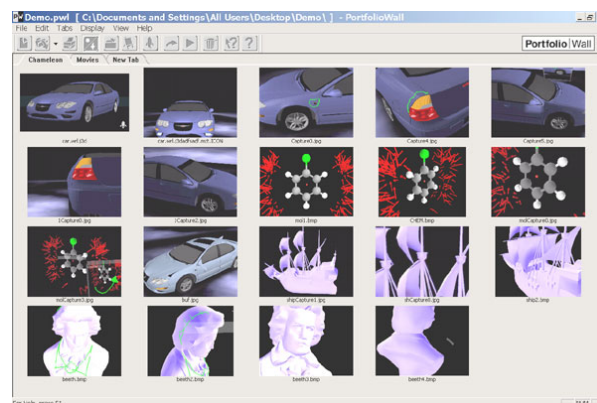

Figure 11. 2D PortfolioWall browser for accessing 3D models and annotations.

SYSTEM IMPLEMENTATION
The system uses an Intel Pentium III 500 MHz dual processor running Microsoft Windows2000 with a NVIDIA GeForce2™ graphics card. The annotation application is implemented in Java™ and the Java3D™ API with an event-based architecture to collect and record information. VRML files are read into the scene-graph courtesy of a loader provided by Sun Microsystems. A device driver produces navigation information and the system synthesizes the various angles from the joints of the armature into a position and orientation value event.

Touch input is received as a standard mouse event. Depending on the state of the interface and location of the input, the event may be processed as a button push, pen drawing, or a highlighting gesture. Drawing on objects in

the scene is accomplished by attaching Java3D™ behaviors to the scene graph to detect the picking intersection with scene geometry. When appropriate mouse drags occur, the system performs continuous picking to determine the intersection. This picking information is then used to place annotation marks that follow the scene geometry, but are slightly offset above the geometry to make them visible to the user.

Capturing the image in the main window as a JPEG image is the first step in creating snapshots. The image is used as a texture on both sides of a new flat polygonal surface to be placed into the scene. The polygons and texture are placed in the scene graph under one node for each snapshot. This convention allows snapshots to be easily hidden or displayed by simply detaching or reattaching the appropriate node.

Audio is recorded and played by the standard Windows and Java™ sound application programming interfaces respectively.

During recording, all events are time stamped and saved for later use. However, these events are also processed immediately, regardless of recording state, in order to update the display with the appropriate changes in the viewpoint or annotation of the scene.

During playback, a playback window appears as a view on the 3D scene, while the main window remains active in the background. Recorded events are sent to both the playback and main windows but interpreted in different ways. In the playback window, all events are interpreted and thus all session viewpoint movements and annotations reappear in the window. On the other hand, the main viewing window ignores the viewpoint information being replayed and the main viewpoint remains controlled by the position of the display. Other recorded events such as flashlight pointing or drawing do occur in both the windows. The alternate, user-controlled view of the main window allows the user to view the real-time creation of annotations from viewpoints that differ from the original viewpoint being shown the playback window.

EVALUATION

View-only Boom Chameleon
The first version of the Boom Chameleon was built three years ago. This initial version was a "view only" device with no touch screen or microphone. This design limited users to move around a 3D scene and permitted no annotation. We have gained extensive experience with hands-on demonstrations of this initial version, as it has been operated by hundreds of users at SIGGRAPH'99 as part of the trade floor exhibits. We have also conducted hundreds of hands-on demos in our laboratory with users

ranging from experienced 3D computer graphics artists to technology-shy executives of major automotive companies.

We have observed that every one of the users immediately understand the navigation metaphor and are able to control their navigation within seconds. The single instance of a user "becoming lost" occurred when a user moved below the virtual floor in a model and encountered an entirely black scene. We have also observed a few users initially confused when navigation started inside a model of a car without a modeled interior. Such an interior presents an unfamiliar and confusing display, but is quickly remedied as the user moves by chance to the outside of the car. We would venture to say that, in this instance, the main cause of confusion is the model and not the navigation metaphor.

Another interesting and often repeated social observation is what we term the *herding effect*. This effect occurs when small groups of people attempt to observe the screen of the Boom Chameleon while someone is attempting to navigate through a model. This results in a "follow the leader" situation where the group flocks back and forth to look over the shoulder of the user. This effect is especially pronounced when the user is hierarchically superior to others in the group. Since the herding effect results in a clumsy and ineffective method of observation, we have improved the situation by providing a fixed duplicate of the display close to the Boom Chameleon. This setup works well and suggests the Boom Chameleon's role as a 3D podium whereby a presenter may drive the interaction and an audience may watch on a separate fixed display.

We have also observed that some users realize the limitations of the viewpoint control technique. For example, one user asked how he could "fly above the car to look straight down from the top".

Boom Chameleon Annotation
Using the Boom Chameleon augmented with a touch screen, microphone and our specially designed annotation software, we performed a range of user evaluations. Approximately twenty-five people tried and commented on the system, resulting in modifications to our application. After these modifications were implemented the users had little trouble learning and operating the refined interface.

We then conducted an informal study focused on how users simultaneously use speech, viewpoint control, drawing, and pointing in annotation. To study this behavior we recruited six subjects and asked them to perform a task with the Boom Chameleon. None of the subjects had previously had experience with the annotation system.

We first presented the subjects with an operational overview of the system. The annotation functionality was demonstrated and the user was then given a short time to familiarize themselves with the system by navigating and annotating on top of a standard car model.

After this short acquaintance period, we revealed that their task involved searching for, and reporting on, major defects that were introduced into an altered model. The car they were to inspect had been vandalized resulting in "some severe and dramatically strange defects – probably the work of an angry wizard". Subjects were asked to play the role of an insurance claim adjuster and annotate both the defects and possible repairs.

To keep task times short, we added six very obvious defects to the car model (see Figure 12) ranging from a sagging driver's mirror, a crooked sunroof, an extra tail fin, an oddly rotated tire and some large dents in the body of the car.
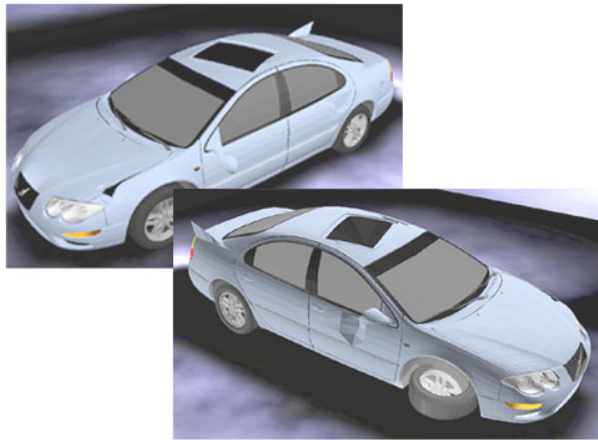
Figure 12. User study: a car model with defects.

We reviewed all of the annotation sessions and, for two users, we catalogued the annotation activity in detail. Figure 13 shows a visualization of the user's activities over time for their annotation session. We note five major observations:

First, many different, but small, annotation events are highly interspersed within a short period of time. We believe that if annotation between data types required explicit actions, e.g. switching between a viewpoint control mode and gesture input mode, this intermingling would not have occurred.

Second, although a variety of annotation session styles were observed, we noticed that most users segmented the annotation session into natural repeating patterns, clustering on each defect. For example, one user moved the viewpoint to locate a defect, positioned the flashlight on the area of interest, and then spoke and drew markings to comment on the defect. This series of actions was repeated for each defect the user found. Another user adopted a style that involved inspection by repeatedly orbiting around the model. When a defect was spotted, the user left orbit and zoomed in on the region of interest.

Third, users spoke the majority of the time during the annotation session, although this may have been due to slight nervousness about their performance in the task. Nevertheless, we found that their speech often occurs in parallel with viewpoint movement or gesture. Indeed, we observed that voice annotations were the predominant style of commenting in conjunction with "framing" the proper 3D view. The flashlight mode was used less frequently than the pen mode.

Fourth, one surprising result was how users employed the snapshot feature. While the test task could be performed easily without using snapshots, we observed that some users applied an annotation and then pulled back to make a snapshot of it. One user in particular annotated all of the defects and then made a second pass taking a snapshot of each defect with its annotation. In both of these cases, we speculate that users took snapshots to emphasize the defects and to produce a visual summary.

Finally, we observed a few occasions where the users accidentally confused drawing on the object and drawing on a screen snapshot. Users would make an annotation, move the viewpoint and then realize that they had not been drawing on the screen snapshot, but were instead drawing on objects in the 3D scene. Users seemed to recover easily from this mistake and move quickly back to the appropriate viewpoint to either vocalize a comment, or take a snapshot and re-annotate.
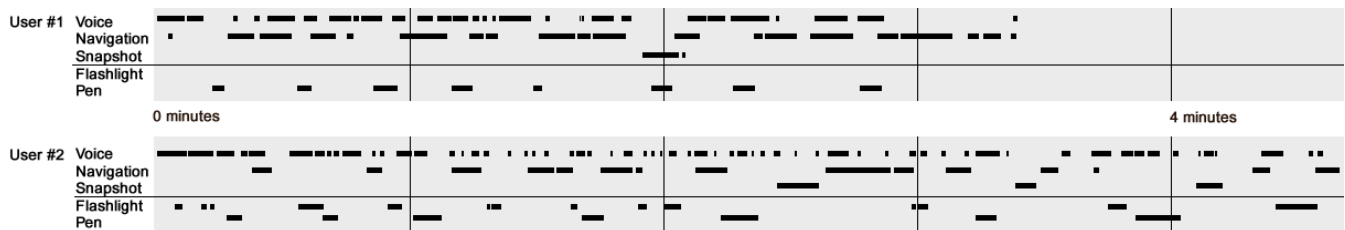
Figure 13. Annotation session analysis for two users.

DISCUSSION & FUTURE WORK

There were many insights gained from our experience with the Boom Chameleon system. Users were clearly adept at operating the Boom Chameleon and its annotation functions; however, conventions relating to the "new and best practices" for this method of annotation have yet to be developed. For example, what common set of mark-up symbols should be used, if any? How effectively can snapshots be used to organize annotations? Should the user speak as if the recipient of the annotations is standing beside him or her? Since the microphone captures ambient sounds in the room, as with any audio recording situation, privacy issues need to be considered. While we indicate when a recording session is in progress through a large red button on the display, a secondary light could be added to the top of the Boom Chameleon's central post to signal to nearby colleagues that a recording session is in progress.

Our experience with the annotation software has also revealed tradeoffs in determining the depth of placement for a snapshot in a scene. If a snapshot is placed closer to an object (see Figure 5), it is larger and close to the area of interest. However, this comes at the cost of obscuring and possibly even intersecting parts of the model. Yet, if the snapshot is positioned farther away the correspondence between model and snapshot is not as obvious. Toggling the visibility of a snapshot in a scene addresses these problems somewhat, but future research may develop better schemes for users to control the depth and placement of snapshots.

The physical layout of the Boom Chameleon affords egocentric viewpoint manipulation but limits other navigation styles. As in our examples, looking inwardly at an object centered on the base of the armature prevents the user from interfering and colliding with the arm and, with an object positioned at the center, the base acts as a physical barrier which does not allow the viewpoint to be moved through the center of the object. In contrast, using the unit to view outwards from the center is more difficult because there is only a small space to walk between the base and the display. Solutions to this problem may entail use of a larger armature, inversion of the device by mounting it to the ceiling, or offsetting the origin of the virtual scene.

The Boom Chameleon is a weighted and counter-balanced device. Because of this, inertia contributes to the trajectory of the display and results in a Steadicam™-like quality to the movements. We believe that it may be worthwhile to emulate this physical property in software for non-counterbalanced spatially-aware displays.

We have insights that are also applicable to other areas, such as with augmented reality applications. Since our system allows the user to keep in direct contact with others while annotating virtual objects, the resulting hand, eye, and group dynamics may well afford rich social cues while working in a shared collaborative space [3].

In future work, we would like to address the many limitations of our system. For example, methods for deleting and editing portions of an annotation session may be introduced. Another avenue for future exploration is the use of Chameleon navigation in combination with other types of camera controls, mappings and control/display ratios. For example, a graphical widget could be used to control the view magnification.

Other directions to explore are the contexts of use and application specific domains. For example, the Boom Chameleon could be used to produce camera shots in virtual scenes for early-production visualization in the entertainment industry. Household applications could include the use of specially instrumented spatially-aware TabletPCs to, for example, describe how well a new couch would fit in the corner of a room.

CONCLUSIONS

Our work with the Boom Chameleon has demonstrated that the "window on a virtual world" metaphor is extremely quick to learn and an easy to operate 3D navigation technique. A weakness of this approach is that it allows only a limited type of navigation and requires special hardware. Nonetheless, we feel that it could be used effectively in many applications where non-technical users require the ability to inspect 3D virtual objects. Studies using our annotation system have shown that users utilized the simultaneous capture of 3D viewpoint, voice and gesture and easily transitioned between the data types. Because of this, we believe that the Boom Chameleon has the potential to be an effective tool in 3D design reviews.

ACKNOWLEDGMENTS

REFERENCES

1. Art+Com. http://www.artcom.de.
2. Balakrishnan, R., Fitzmaurice, G.W. and Kurtenbach, G. (2001). User Interfaces for Volumetric Displays. *IEEE Computer*, 34(2), 37-45.
3. Billinghurst, M., and Kato, H. (2002). How the Virtual inspires the real: Collaborative Augmented Reality. *Communications of the ACM 45,* 7 (July 2002), 64-70.
4. Bolt, R.A. (1980). Put-That-There: voice and gesture at the graphics interface. *SIGGRAPH Computer Graphics*, 14(3), 262-270.
5. Bourguignon, D., Cani, M-P. and Drettakis, G. (2001). Drawing for Illustration and Annotation in 3D, In *Proc. of EUROGRAPHICS'01*, 114-122.

6. Buxton, W (2001). Less is More (More or Less), in *P. Denning (Ed.), The Invisible Future: The seamless integration of technology in everyday life.* New York: McGraw Hill, 145 – 179. Also at http://www.BillBuxton.com/LessIsMore.html.

7. Buxton, W. & Fitzmaurice, G.W. (1998). HMD's, Caves & Chameleon: A Human-Centric Analysis of Interaction in Virtual Space. *Computer Graphics, The SIGGRAPH Quarterly*, 32(4), 69-74.

8. Chalfonte, B.L, Fish, R.S. and Kraut, R.E. (1991). Expressive Richness: A comparison of speech and text as media for revision, In *Proc. of CHI'91*, 21-26.

9. Cohen, P., McGee, D., Oviatt, S., Wu, L., Clow, J., King, R., Julier, S. and Rosenblum, L. (1999). Multimodal Interaction for 2D and 3D Environments, *IEEE Computer Graphics and Applications*, 19(4), 10-13.

10. Fakespace Labs. http://www.fakespacelabs.com.

11. Fitzmaurice, G.W. (1993). Situated Information Spaces and Spatially Aware Palmtop Computers. *Communications of the ACM*, 36(7), 38-49.

12. Fitzmaurice, G., Zhai, S. and Chignell, M. (1993). Virtual reality for palmtop computers, *ACM Transactions on Information Systems*, 11 (3), 197-218.

13. Frohlich, D. and Daly-Jones, O. (1995). Voicefax: A shared workspace for voicemail partners, In *Proc. of CHI'95* Short papers, 308-309

14. Hardock, G., Kurtenbach, G., and Buxton, W.A. (1993). Mate: A marking based interface for collaborative writing. In *Proc. of UIST'93*, 259-266.

15. Harmon, R., Patterson, W., Ribarsky, W. and Bolter, J. (1996). The Virtual Annotation System, In *Proc. of Virtual Reality Annual International Symposium (VRAIS'96)*, 239-245.

16. Iris Annotator. http:// www.sgi.com/software/annotator.

17. Jung T., Gross M. D. and Do E. Y. (2002). Annotation and Sketching on 3D Web Models, In *Proc. of ACM Conference on Intelligent User Interfaces'02*, 95-102.

18. LaViola, J., Holden, A., Forsberg, D. and Bhuphaibool, D. (1998). Collaborative Conceptual Modeling Using the SKETCH Framework, In *Proc. of the first IASTED International Conference on Computer Graphics and Imaging*, 154-158.

19. Levine, S. and Ehrlich, S. (1991). The Freestyle System: A Design Perspective. In Klinger, A. (Ed.) Human-Machine Interactive Systems, Plenum 3-21. Also in Baecker, R.M., Grudin, J., Buxton, W., and Greenberg, S. (Eds.) Readings in Human-Computer Interaction: Toward the Year 2000, 2nd edition, 871-880.

20. Lienhart, Rainer. (2000). A System for Effortless Content Annotation to Unfold the Semantics in Videos. In *Proc. of the IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL'00)*, 45-50.

21. Loughlin, M. and Hughes, J. (1994). An Annotation System for 3D Fluid Flow Visualization. In *Proc. IEEE Visualization '94*, 273-279.

22. Maeda, J., Greenwold, S. Installation. Aesthetics and Computation Group, MIT. http://acg.media.mit.edu/people/ simong/installationNew/cover.html.

23. Marshall, C.C. (1997). Annotation: from paper books to the digital library. *Proc. of the Second ACM International Conference on Digital Libraries*, 131 – 140.

24. Matsuoka, H., Onozawa, A., Hosoya, E., Sato, H., and Nojima, H. (2002). Regeneration of Real Objects in the Real World. *Presented at SIGGRAPH Symposium on Interactive 3D Graphics 2002 - Emerging technologies. http://kankyo.lelab.ecl.ntt.co.jp/real_in_real/eng/index.htm.*

25. O'Hara, K., Sellen, A., (1997). A comparison of reading paper and on-line documents. In *Proc. of CHI'97*, 335-342.

26. Oviatt, S. L. (1999) Ten myths of multimodal interaction, Communications of the ACM, Vol. 42, No. 11, November, 1999, pp. 74-81.

27. Pedersen, E.R., McCall, K., Moran, T.P., Halasz, F.G., (1993). Tivoli: an electronic whiteboard for informal workgroup meetings. In *Proc. of CHI'93*. 391-398.

28. Poupyrev, I., Tomokazu, N., and Weghorst, S. (1998). Virtual Notepad: Handwriting in Immersive VR, In *Proc. of IEEE Virtual Reality Annual International Symposium'98 (VRAIS'98)*, 126-132.

29. Rekimoto, J. and Nagao, K. (1995). The World through the Computer: Computer Augmented Interaction with Real World Environments, In *Proc. of UIST'95*, 29-36.

30. Schmalstieg, D., Encarnacao, L.M. and Szalavari, Z. (1999). Using transparent props for interaction with the virtual table," In *Proc. SIGGRAPH Symposium on Interactive 3D Graphics 1999*, 147-154.

31. Small, D. and Ishii H. (1997). Design of Spatially Aware Graspable Display*, ACM CHI'97 Electronic Publications*.

32. Stellingwerff, Martijn. (1999). SketchBoX. In *Proc. of Architectural Computing from Turing to 2000. eCAADe*, ISBN 0-9523687-5-7, Liverpool (UK) 15-17 September 1999, 491-497.

33. Stoakley, R., Conway, M. and Pausch, R. (1995). Virtual reality on a WIM: Interactive worlds in miniature. In *Proc. of CHI'95*, 265-272.

34. Weber, K., and Poon, A. (1994). Marquee: A Tool for Real-Time Video Logging, *Proc. of CHI '94*, 58-64.

35. Whittaker, S. Davis, R. Hirschber, J., and Muller, U. (2000). Jotmail: a voicemail interface that enables you to see what was said, *Proc. of CHI'2000*, CHI Letters 2(1), 89-96.

36. Whittaker, S., Hyland, P., and Wiley, M. (1994) Filochat: Handwritten notes provide access to recorded conversations. *Proc. of CHI'94*, 271-277.

37. Wilcox, L.D., Schilit, B.N., Sawhney, N. (1997) Dynomite: A Dynamically organized Ink and Audio Notebook.*Proc. of CHI'97*, 186-193.

38. WindowsVR. http://www.abs-tech.com/Produtos/3D_VR/ VR-Hardware/hmds1/Virtual_Research/win_vr.html.