A Framework for 3D Visualisation and Manipulation in an Immersive Space using an Untethered Bimanual Gestural Interface

Yves Boussemart, François Rioux, Frank Rudzicz, Michael Wozniewski, Jeremy R. Cooperstock
Centre For Intelligent Machines
3480 University Street
Montréal, Québec, Canada
{yves, frioux, frudzi, mikewoz, jer}@cim.mcgill.ca

ABSTRACT

Immersive Environments (IE) offer users the experience of being submerged in a virtual space, effectively transcending the boundary between the real and virtual world. We present a framework for visualization and manipulation of 3D virtual environments in which users need not resort to the awkward command vocabulary of traditional keyboard-and-mouse interaction. We have adapted the transparent toolglass paradigm as a gestural interface widget for a spatially immersive environment. To serve that purpose, we have implemented a bimanual gestural interpreter and parser to recognize and translate a user's actions into commands for the toolglasses. In order to satisfy a primary design goal of keeping the user completely untethered, we use purely video-based tracking techniques.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Representation (HCI)**]: User Interfaces—*Graphical user interfaces, Interaction styles, Theory and methods*

Keywords

Immersive Environment, Toolglass, Bimanual Interaction, Scene Modelling, Telepresence, Gesture Recognition

1. INTRODUCTION

We present a framework for a fully immersive 3D computer-augmented environment aimed to improve visualization for 3D applications. Immersive environments are well-suited for such applications as they allow users to work and interact within the virtual space while being physically and perceptually surrounded by displays on which the synthetic world is rendered. Ultimately, a user should receive no percepts that violate the congruence between physical and virtual reality.

The architecture for such an environment should be as natural as possible, allowing users to see the virtual world and directly interact with it without the use of intermediate hardware such as gloves, 3D-mouse, or VR headgear. This leads to the requirement that the system be *walk in and use*, in the sense that the user should not have to don any special apparatus before being able to interact with the environment. Further, because much of human interaction with the physical world occurs through gesture such as gripping, manipulating and releasing, we wish to employ the same paradigm for our interactions in the virtual world.

A user of the proposed system employs two hands to control the environment by manipulating a virtual interface widget, which we refer to as a *pieglass*. This widget is based on the transparent toolglass, and serves as a mapping layer between gestures and actions specific to an application. The user grasps a widget with one hand while applying actions with the other, hence the system is designed with the properties of two-handed interaction in mind.

In the following sections, previous work in related fields is described including immersive environments, bimanual interaction, and toolglass interfaces. An overview of the research framework and its system architecture is provided in the context of the McGill Shared Reality Environment (SRE). Finally, future research directions are explored.

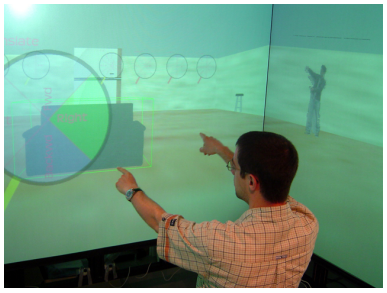


Figure 1: Scene modelling application with embedded remote participant.

2. RELATED WORK

Immersive environments often follow the architecture proposed by the Cave Automated Virtual Environment (CAVE) [5]. Systems based on CAVElib typically consist of a cube with rear-projection displays, where stereoscopic video is provided by the use of shutter glasses. Ascension *Flock of Birds* or other 3D positioning sensors track the position of the user for perspective correction. With increasing graphics performance of current computer hardware, many other systems have emerged including as X-Rooms [11], BNAVE [13], Chromium [10], and Blue-C [6].

VRST'04, November 10-12, 2004, Hong Kong.

Providing natural and efficient control techniques in an immersive or large-screen environment is a difficult problem. Gesture-based interaction is a natural choice that relates to real-world human interaction, and several projects have considered gestures for typical tasks. For example, metaphors for navigation in a virtual world has been explored by Ware & Osborne [15], such as holding a camera in one's hand, holding the world in one's hand, or using the hand to control a virtual flying vehicle. Apart from navigational tasks, researchers such as Pierce et al [14] describe numerous gestural primitives for object selection and manipulation. Examples include resting objects on the palm of the hand, using two hands to frame and object, or pinching an object between two fingers. Work by Balakrishnan & Kurtenbach [2] has gone into combining these tasks into a bimanual system, where navigation and object manipulation is divided between two hands.

These methods however tend to be applied to simple tasks or application-specific systems, thus allowing for gestures to be mapped directly to actions on the virtual world. We propose a more generic style of interaction where gestures are used to control menu-like widgets. The widgets proposed are based on a combination of the transparent toolglass proposed by Bier et al [3] and pie menus proposed by Hopkins [9].

3. ARCHITECTURAL FRAMEWORK

In order to deploy an immersive environment that uses untethered gestures as input, we combine several architectural components. The user is surrounded by an immersive display, and cameras are used to track gestures that the user performs.

3.1 Immersive Environment

The Shared Reality Environment (SRE) at McGill University is a spatially immersive physical space that is similar to the original CAVE, in that it uses display surfaces forming three sides of a 2.5m cube. An optimized graphics renderer based on OpenGL has been implemented, called *Qave*. Qave offers higher frame-rate and is more streamlined for our application than the similar CAVELib. It uses monoscopic rear-projection rather than a stereoscopic approach since our design goal of a *walk in and use* system precludes the use of extraneous gear such as stereoscopic goggles. The user is not constrained to stand in a specific location, since *off-axis projection* [5] is used and frustums for each projection are constantly updated to provide the correct perspective.

The framework has also been extended to be used by multiple remote participants or viewers, by means of a high-efficiency network transport mechanism [4] that distributes NTSC video and multichannel audio. This permits sound spatialization to be used in conjunction with video avatars of remote participants, as seen in Figure 1.

3.2 Video Tracking

To allow for *untethered* sensing of human motion, only video cameras are employed, thus freeing the user from body-worn sensors or other special clothing. The tracking algorithms make use of several strategically placed cameras that view the user from the top, front, and side of the space. A gesture recognition system uses these cameras to track the position of the user's head and hands over time.

To accomplish this, the physical space that the user stands in is modelled in three dimensions, with the origin at the center. Cameras are aligned so that coordinates returned from feature extraction algorithms can be combined with other camera views to estimate the true 3D position of an object.

For the feature extraction process, we use a combination of background removal via image differencing, and skin colour segmentation. Bounding boxes are formed for blobs of connected pixels that represent each hand, the user's head, and the user's entire body. Center of mass calculations are made for these blobs, providing estimates of current body posture. Uncertainties do however arise due to problems of occlusion, varying illumination, false positives due to skin-like colours (eg. wood surfaces), and poor pixel connectivity. For reliable tracking and minimization of tracking noise, the CONDENSATION algorithm [12] is being explored to model these uncertainties.

3.3 Gestural Interface

As a first step of the gesture recognition process, we must determine whether the detected user motion was intentional and meaningful in the current context. In terms of mappings, some are relatively obvious, such as grabbing or pointing to an object of interest, while others, such as modifying a texture or adjusting the properties of a light source, have no direct physical equivalent. While one may be tempted to assign the latter group to non-obvious gestures we should be aware that as we increase the number of gestures, we increase the need for sensing technology more precise than simple video-based tracking.

We could assist the user in this regard by taking advantage of the multimodal and immersive nature of the environment, for example, using voice commands to select attributes that are not easily specified by gestures. However, until we have an opportunity to incorporate speech recognition or other non-gestural input modalities into the environment, we wish to provide a mechanism for carrying out actions for which no obvious mappings to a gestural vocabulary exist. This motivates the use of a menu hierarchy based on the pieglass widget and the development of a simple gesture set for interacting with the widgets in an immersive context.

4. PIEGLASS & BIMANUAL INTERACTION

Virtual interface widgets have been designed, called pieglasses, and are used as a mapping layer between gestures and their corresponding effects. This permits rich and diverse interaction while keeping the core two-handed gesture set simpler.

4.1 Pieglass Metaphor

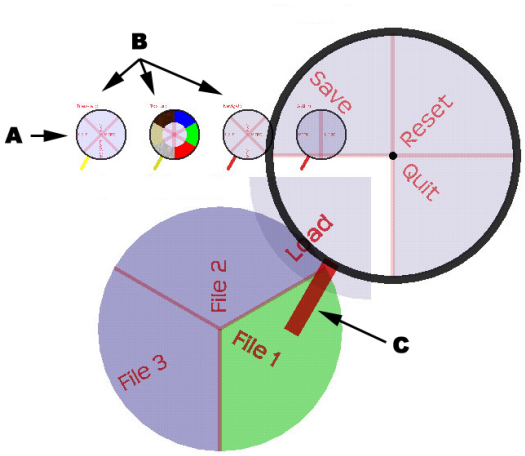
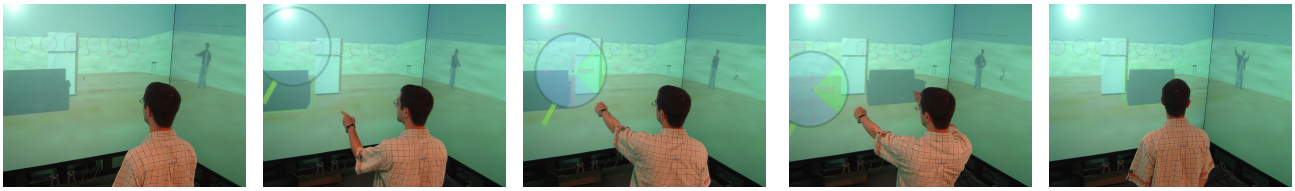


Figure 2: typical pieglass showing file management operations.

Our pieglass metaphor, illustrated in figure 2, is an extension of the transparent toolglass proposed by Bier et al [3], which is an im-



(a) Standby (b) Pieglass Pointing (c) Pieglass Selection (d) Wedge Selection (e) Deselection

Figure 3: Gestural interaction with the pieglass metaphor.

proved palette or tool that allows the user to select an option and target simultaneously with one click. The pieglass uses the same technique, yet it is used for all menu functions and is structured differently. Each pieglass is composed of a semi-translucent circular menu, angularly partitioned into pie-like wedges where each wedge represents a unique action within the context of the pieglass. An action is executed by *clicking through* a wedge, and the action is applied (in general) to the object directly behind the pieglass. We adopt this paradigm to facilitate the construction of complex, movable, and *see-through* menus whose actions can be applied to the underlying objects.

Early feedback with a small focus group led to the development of a number of novel features. In order to improve visibility of system status, a *Pieglass rack* (fig. 2(A)) consisting of multiple pieglasses was arranged across the screen. Each pieglass is therefore more semantically coherent, and has fewer available actions visible. *Captions* (fig. 2(B)) above each pieglass in the rack indicates their high-level functionality, and more feedback is now available to the user. It was also observed that the basic visual affordance of a *handle attachment* (fig. 2(C)) to the widget was evocative of an object that could be grabbed and moved about in space - an especially important affordance for new users in an immersive environment. The collection of pieglasses could also be further partitioned by colouring these handles according to logical categories. For example, separating those that operate on specific objects from those whose actions affect the global environment.

Since the software framework is intended for many different applications, we provide for flexible configuration of the available operations through an XML file, which further specifies the layout and properties of pieglasses. Custom wedges and actions can also be programmed by users with specific needs by extending software classes, building dynamically linked libraries and editing the XML configuration file.

4.2 Gesture Set

There is an obvious disparity in the utility, and hence preference for the use of our left and right hands for certain tasks, often described as *handedness*. Guiard [7] defines a number of rules related to this phenomenon. Most significantly, a typical two-handed interaction usually begins with the non-preferred hand choosing a frame of reference while the preferred hand then applies more precise motion, both spatially and temporally, to that reference.

Following Guiard's principles, we use the direction of the pieglass handle as a bias to motivate the user to grab the widget with the non-preferred hand,[1] and select the wedge corresponding to the desired action with the preferred hand. The target object or position for the action is found by a projection of the user's view through the *crosshair-like* center of the pieglass.

The use of pieglasses allows a greatly simplified gesture set since all actions on the system can be performed through the use of these menus. A complex gesture syntax need not be defined, and instead a deictic gesture metaphor is used where the widgets are controlled by simple pointing gestures. By holding an arm partially extended, a virtual cursor is moved in an arc about the user according to the hand motion. There are two such cursors, one for each hand. To select or invoke actions, the user simply extends the appropriate hand in the direction of the target.

Figure 3 shows an example of using this method to move a sofa to the right. To begin, it is assumed that the user stands in the center of the environment, with both hands at rest (fig. 3(a)). The non-preferred hand will point to a pieglass in the rack (fig. 3(b)) then fully extend to grasp it (fig. 3(c)). Once grasped, the pieglass attaches itself to the cursor and moves with the non-preferred hand. The user positions it over a target, then the preferred hand is used to point to a desired pieglass wedge. Once the correct wedge is highlighted, the user fully extends the hand to invoke the action for that wedge (fig. 3(d)). In the example shown, this results in a *translation* action that moves the sofa to the right.

5. DISCUSSION AND FUTURE WORK

Since the proposed framework remains an ongoing research effort, several components still need to be implemented and tested with experimental studies on real subjects. Additional implementation effort is required to permit complete configurability through XML files and further aesthetic modifications are needed such as hiding the pieglass rack when it is not needed.

5.1 Interface Paradigm

Several unexpected consequences of the pieglass metaphor were discovered as a result of the design process. Early paper prototyping of the modelling system used pieglasses which were merely circular menus without the handle shown in Figure 2. Given this design, early evaluations showed that 8 out of 12 users did *not* initially assume that the pieglasses were movable, and did not attempt to grab them without instruction. This resulted in perhaps the most significant change to the original design, which was the addition of a graspable handle protruding from the pieglass menu. With this attachment, the widget takes on the familiar shape of a magnifying glass, and with it *all* subsequent test users immediately recognized the interaction paradigm. This could indicate a general preference of users for graphical widgets to resemble familiar, real-world objects.

From a technical standpoint, hand detection by video for bimanual interaction is much easier if the extremities are separated in space. The handle attachment naturally separates the hands and hence occlusion problems can be avoided while maintaining the advantages of the pieglass. While improvements to the video tracking sytstem are continuously being explored, separating the graspable area from functional areas results in a more comfortable partitioning of the visual space.

[1]For example, in the case of right-handed users, the pieglass handle descends to the left.

5.2 Architecture

Preliminary benchmarks with our new Qave graphical engine indicate that consumer-grade graphic cards can provide levels of graphics performance easily exceeding that provided by the far more expensive SGI Onyx II only a few years earlier. To solve the problem of large surface high resolution display, we plan on exploring PC-based clustered rendering methods, where a number of graphics cards on separate PCs are used to drive multiple front- and back-projected surfaces. This requires frame synchronization, provided either by hardware or software [1]. A related issue is the coordination of multiple overlapping projectors to render a coherent scene and eliminate front-projection shadows through closed-loop analysis by calibrated cameras [8].

5.3 Empirical Testing

Given more advanced gesture recognition technology, one avenue of exploration would be to add the ability to grab, move and manipulate arbitrary virtual objects in the scene. This would allow the user to bypass the pieglass layer for simple tasks such as translation of an object. Empirical comparisons of such interaction techniques with that of the pieglass would then be possible. Similarly, formal studies on the differences between unimanual pieglass interaction and complex bimanual gestural interaction are being undertaken and the result will be reported elsewhere. This work will examine the codependence of the pieglass metaphor with bimanual interaction in terms of efficiency, naturalness, and cognitive principles. As another empirical study, it would be worth investigating the benefits of pieglass menus over both classical linear menus and a complex set of gestures.

6. CONCLUSIONS

The framework presented here introduces the infrastructure needed to interact with an immersive environment where two-handed gestures are employed to control virtual interface widgets, which in turn, act on the environment. The pieglass layer separates gesture recognition technology from the core of the application, thus creating a framework that is sufficiently general to evolve for future needs. Our prototype application involves 3D scene creation by placing and manipulating models in the environment.

The video-based gesture recognition system is currently employed to track simple two-handed gestures. As mentioned earlier, much of the limitations of this framework involve the limited fidelity of tracking, and is thus an important area of improvement for our ongoing research. Another challenge is increasing the rendering performance of the system. As synthetic worlds become increasingly rich, the necessary processing power for real-time graphics increases accordingly. We believe that this can best be addressed by a PC-based clustered rendering approach.

To conclude, we argue that this framework will provide a valuable infrastructure for research of new two-handed interaction methods, in particular for immersive environments, in which the traditional unimanual point-and-click paradigm is either inappropriate or insufficient to cope with the demands made on users.

7. ACKNOWLEDGMENTS

The authors would like to thank all the members of the SRE lab for their invaluable feedback and discussions. This work was supported by the Natural Sciences and Engineering Research Council (NSERC), Fonds Québécois de la recherche sur la nature et les technologies (FQRNT), Valorisation Recherche Québec (VRQ), and the Canada Foundation for Innovation (CFI). This support is gratefully acknowledged.

8. REFERENCES

[1] J. Allard, V. Gouranton, G. Lamarque, E. Melin, and B. Raffin. Softgenlock: Active stereo and genlock for pc cluster. In *Proceedings of the Joint IPT/EGVE'03 Workshop*, Zurich, Switzerland, May 2003.

[2] R. Balakrishnan and G. Kurtenbach. Exploring bimanual camera control and object manipulation in 3d graphics interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 56–62. ACM Press, 1999.

[3] E. Bier, M. Stone, K. Pier, W. Buxton, and T. DeRose. Toolglass and magic lenses: The see-through interface. In *Proceedings of SIGGRAPH '93*, pages 73–80, 1993.

[4] J. R. Cooperstock and S. P. Spackman. McGill Ultra Video-Conferencing System. http://ultravideo.mcgill.edu.

[5] C. Cruz-Neira, D.J.Sandin, and T. DeFanti. Surround-screen projection-based virtual reality: The design and implementation of the cave. In *Proceedings of SIGGRAPH '93*, pages 135–142. ACM Press, 1993.

[6] M. Gross, S. Wurmlin, M. Naef, E. Lamboray, C. Spagno, A. Kunz, E. Koller-Meier, T. Svoboda, L. Van Gool, S. Lang, K. Strehlke, A. V. Moere, and O. Staadt. blue-c: a spatially immersive display and 3d video portal for telepresence. *ACM Trans. Graph.*, 22(3):819–827, 2003.

[7] Y. Guiard. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a mode. *Journal of Motor Behavior*, 19(4):486–517, 1987.

[8] M. N. Hilario and J. Cooperstock. Occlusion detection for front-projected interactive displays. In *Proceedings of Pervasive 2004*, Vienna, Austria, April 2004.

[9] D. Hopkins. The design and implementation of pie menus. *Dr. Dobb's J.*, 16(12):16–26, 1991.

[10] G. Humphreys, M. Houston, R. Ng, R. Frank, S. Ahern, P. D. Kirchner, and J. T. Klosowski. Chromium: a stream-processing framework for interactive rendering on clusters. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 693–702. ACM Press, 2002.

[11] K. Isakovic, T. Dudziak, and K. Kchy. X-rooms. In *Proceeding of the seventh international conference on 3D Web technology*, pages 173–177. ACM Press, 2002.

[12] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking, 1998.

[13] J. Jacobson, M. S. Redfern, J. M. Furman, S. L. Whitney, P. J. Sparto, J. B. Wilson, and L. F. Hodges. Balance nave: a virtual reality facility for research and rehabilitation of balance disorders. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 103–109. ACM Press, 2001.

[14] J. S. Pierce, A. S. Forsberg, M. J. Conway, S. Hong, R. C. Zeleznik, and M. R. Mine. Image plane interaction techniques in 3d immersive environments. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 39–ff. ACM Press, 1997.

[15] C. Ware and S. Osborne. Exploration and virtual camera control in virtual three dimensional environments. In *Proceedings of the 1990 symposium on Interactive 3D graphics*, pages 175–183. ACM Press, 1990.