A Tangible User Interface System for CAVE Applications

Ji-Sun Kim[*] Denis Gračanin[†] H. Lally Singh[‡] Krešimir Matković[§] Josip Jurić[¶]

Virginia Polytechnic Institute & State University
660 McBryde Hall, Blacksburg, VA 24061, USA

VRVis Research Center Vienna
Donau-city-strasse 1, A-1220 Vienna, Austria

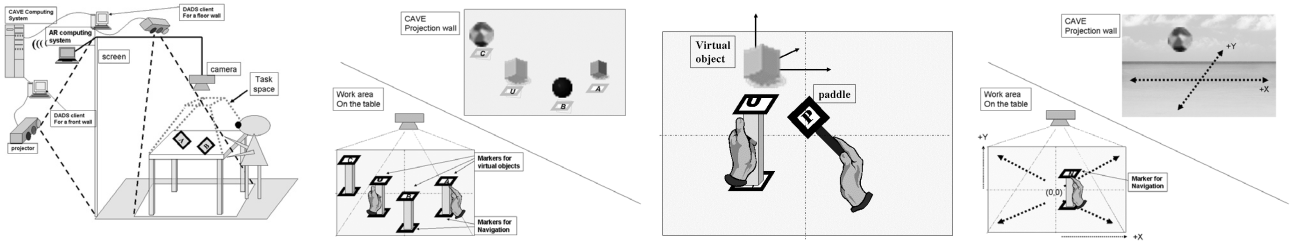

Figure 1: A tangible user interface system and related interaction techniques

ABSTRACT

This paper describes a 3D, tangible user interface system and related interaction techniques for a CAVE environment. The developed system is based on off-the shelf software and hardware component to provide 3D input data for applications. A CAVE installation with four sides (three walls and a floor) is used as a display and interaction space. Interaction tasks (navigation, selection, manipulation) are performed using interaction techniques based on manipulation of physical objects (props). All virtual objects are directly manipulated using the corresponding props to which AR physical markers are attached. Each physical marker corresponds to a specific virtual object. The floor projection (a white rectangle overlaid on top of the application generated video stream) or a directional light is used to create illumination necessary for computer vision based marker detection using ARToolKit. Initial evaluation results are positive and provide directions for future research.

CR Categories: H.5.1 [Information Interfaces and Presentation (e.g., HCI)]: Multimedia Information Systems—[Artificial, augmented, and virtual realities] I.3.4 [Computer Graphics]: Graphics Utilities—[Virtual device interfaces]

Keywords: 3D interaction, mixed reality, augmented reality

1 INTRODUCTION

Tangible User Interfaces (TUIs) use physical objects (props) to provide data input and output for a virtual world [6]. They enable users to manipulate virtual 3D objects by handling props. Similarly, Augmented Reality (AR) systems enable users to interact in real time with real and virtual objects in the real world [11]. The enhanced reality is provided by superimposing the computer generated virtual objects and information such as text or graphics onto real objects.

Many potential AR applications, in particular medical, entertainment, military training and industrial use, have been proposed to improve task performance. Main technologies used for an AR system include a display technology that enables a combination of real and virtual objects into a single view; a tracking technology for registration of virtual objects; and real-time interactions. Software tools such as ARToolKit enable easy and rapid development of AR applications [7].

The AR system should enable a user to seamlessly interact with virtual objects in a real world [1]. TUIs in AR, referred to as Tangible Augmented Reality (TAR) [7], allow users to interact with virtual objects in the real world while, at the same time, providing spatial registration and presentation in real-time. TAR interfaces require users to wear a see-through head-mounted display (HMD) in order to view the AR environment [2] in an immersive setup.

In typical AR applications, standard PCs and off-the-shelf USB cameras provide acceptable performances. Fast movements might cause a loss of objects's tracking information, but it does not cause any major problem because the global image processing can cover the whole frame and people can still see the real scene. Sometimes, an occlusion of the physical marker (marker, for shot) causes a loss of the tracking information. In fact, occlusion by physical objects including hands in vision-based AR technology can be an obstacle for interactions. Lee et al. demonstrated occlusion based interaction methods on how that can be used for system control tasks. This occlusion based manipulation is applied to authoring tools for 3D VR [9]. Carvalho et al. [3] presented two-hand manipulation using two thumbs that have attached markers.

A CAVE installation provides a large-scale immersive environment [14] where users can freely move within relatively large area while being able to see the real world. A stereoscopic projection of the virtual environment within the CAVE results in a constantly changing light level. As a consequence, AR software that processes real-world video stream cannot be used efficiently.

The proposed TUI system uses AR technology in another way. Virtual images are not directly superimposed onto props. Instead, the display space is "moved" to CAVE walls and props with AR markers are used as input devices. This provides separate spaces for tasks and interactions. Interaction tasks are performed using props and even though markers augment props, they do not augment the real world. Therefore, the proposed system is not a traditional TAR system because the use of AR technology is limited to implementation of interaction techniques.

---

[*] hideaway@vt.edu
[†] gracanin@vt.edu
[‡] lally@vt.edu
[§] Matkovic@vrvis.at
[¶] Juric@vrvis.at

The CAVE installation used in this research consists of four sides, three walls (front, left, right) and a floor. The floor projector, located above the CAVE, projects down onto the floor. That can be leveraged to provide the light level necessary for using AR software. A white rectangle overlaid on top of the floor projector video stream (or a directional light source) can provide the required lighting. This idea and experience from previous work [10] was used to implement a system that combines CAVE and ARToolkit to provide tangible user interfaces for CAVE applications.

2 SYSTEM DESIGN

The system design addresses the following issues:

- Separation of the task space from the interaction space using ARToolKit in CAVE: Codeless input devices based on props with markers are used for three interaction tasks, selection, manipulation, and navigation. This may cause context switching the task space and the interaction space.
- Limited work area: The area's size is determined by camera's characteristics like resolution, angle of view, and location.
- Use of TUI: intuitive and natural interaction provides direct manipulation and easy control of virtual objects.
- Directional light: Even though the floor projector is used to maintain a constant light level for the camera, additional light can be provided based on the camera's characteristics.

The system is implemented using two open source software toolkits, DIVERSE [8] and ARToolKit [7]. The ARToolKit is used to track markers' position and orientation. CAVE applications send data to a dedicated server and a client program puts that data into a shared memory. Data stored in the shared memory is then used as an input stream. Each wall and the floor have a dedicated display computer (DADS client).

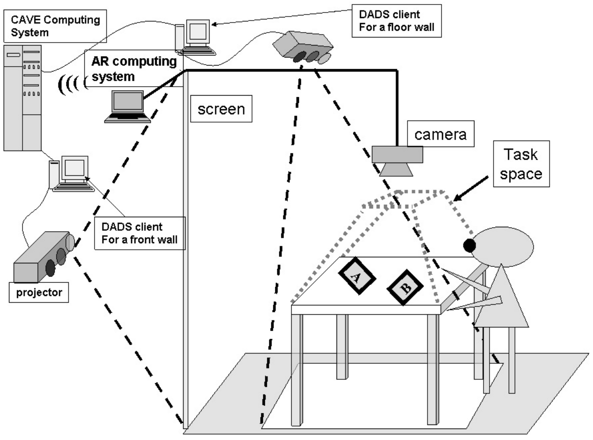

Figure 2: CAVE setup

The system consists of a work area (e.g. a non-transparent table), a camera, a directional light positioned above the work area, a dedicated ARToolKit server, and a client that receives marker updates from the ARToolKit server (Figure 2). The size of the work area depends on the camera's characteristics. The size is also a compromise between a sufficiently large work area and limited obstruction of the CAVE floor area. The physical markers are moved, picked by the camera, detected by the ARToolKit, transmitted by the ARToolKit server, received by the CAVE client, and then relayed into the DIVERSE based application.

The positional information is given to the application as if it was a regular wand device. The wand device has six degrees of freedom, four buttons, and an additional joystick built in. A similar approach was used by He et al. [4] to provide Device Unified Interface (DUI). In this case, virtual input devices are provided using ARToolKit and DUI is provided by DIVERSE. Each marker has a unique identification and this information is sent to the CAVE client together with positional data.

This implementation has some obvious consequences. Complexity is increased due to several network stacks (wireless and wired). Reliability may decrease due to (expected) packet loss, as well as intentional data loss in various race conditions. Image processing, network I/O, and data transformation add static and variable delays between user input and system response thus increasing latency

While the implications of adding network links in the system is well researched [13], the other concerns add some fairly unknown variability. However, most of the reliability and latency issues do not noticeably affect the system during normal use. However, the sensitivity and more subtle interactions (for example, fine-grained movement or accuracy at high speed) will undoubtedly suffer for it. Packet loss and delays are the most obvious cause for concern.

3 INTERACTION TECHNIQUES

There are two possible interaction spaces for object manipulation, a near space and a far space. The near space includes objects within arms' reach and can be directly manipulated. The far space includes remote objects (not within the arms' reach) and can be manipulated with the extended interaction technique like the Go-Go technique [2]. Only the near space interactions are described here. However, there are many possible interaction techniques that can be implemented using markers in the near space.

Figure 3 shows the setup of the TUI system. LEGO^{TM} blocks are used as props with two markers attached, one on the top and one at the bottom. One marker (either top or bottom) is used to manipulate the corresponding virtual object. The other marker is used to navigate in the virtual world. Thus, the user does not need to change a prop to change a task mode between object manipulation and navigation. The paddle prop is optionally used to control values such as direction, velocity, distance, and scaling.

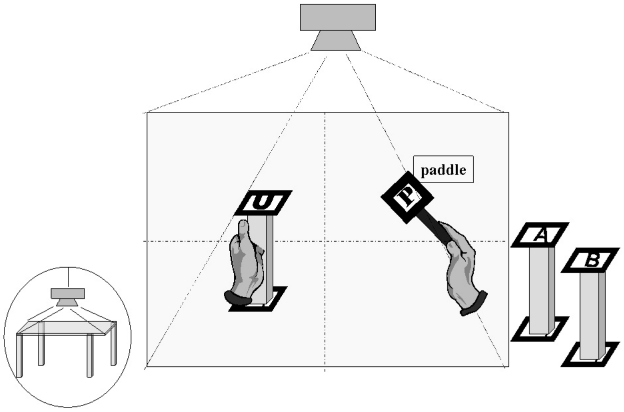

Figure 3: TUI system setup

3.1 Selection

For near objects, the user does not need to use any other interaction techniques for selecting virtual objects because each virtual object moves along with the associated marker's movement.

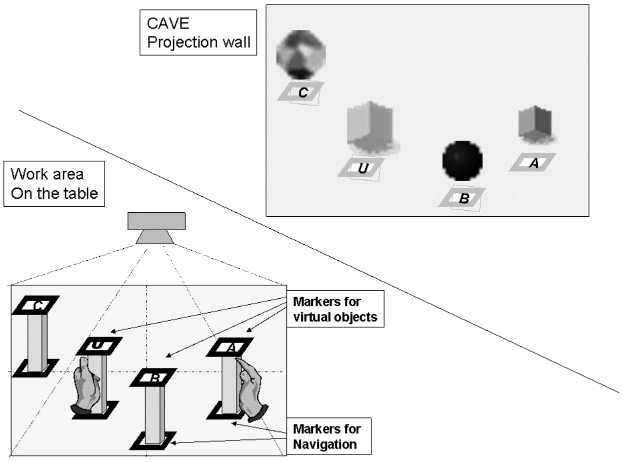

Figure 4: Selection

Figure 4 shows that virtual objects move based on the movement of the corresponding markers. Alternatively, the paddle can act like a mouse button for selection in case that the number of virtual objects is such that it is not possible to have one to one matching between props and virtual objects.

3.2 Manipulation

Unlike the selection task, the manipulation task requires another prop (paddle) for full 3D rotation and precise scaling of objects (Figure 5).

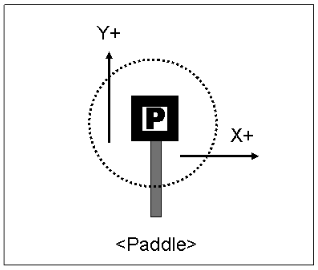

Figure 5: Paddle

3.2.1 Rotation

Since the marker coordinate system's Z-axis points up from the markers surface, marker's rotation determines virtual object rotations around Z-axis. The user uses the padddle to rotate the virtual object around X-axis or Y-axis (Figure 6). Rotation is performed when the distance between the paddle and the marker is sufficiently small. Rotational speed around +Y, -Y, +X, or -X direction (the marker's side nearest to the paddle) is specified by the application.

There are several other rotation techniques. For example, the paddle can help to fully rotate the virtual object. The +Y direction of the paddle indicates the direction of the rotation. Horizontal rotation (around Z) is done by the marker. Once the paddle is close to the marker, the virtual object starts rotating along the +Y direction of the paddle. The rotation stops when the paddle is moved away from the marker.

3.2.2 Scaling

The paddle is used (+Y direction) to indicate the direction for scaling. Once the paddle approaches the marker, then the user can scale the virtual object by changing the distance between the object marker and the paddle. This is a very simple scaling technique and there are several other possibilities to scale virtual objects.

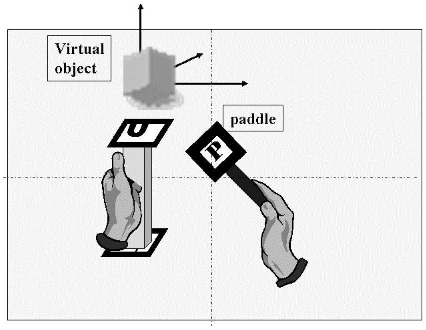

Figure 6: Manipulation

When the user wears stereoscopic glasses with a head tracker, the gaze direction can be used for scaling. When the paddle is close to a marker, the three axes are displayed on the virtual object to indicate scaling. The accessed side of the paddle to the object marker specifies the axis of resize. The users gaze direction is used to scale the virtual object around X- or Y-axis.

3.3 Navigation

Figure 7 shows the work area side, with origin in the middle of the area. Props have two markers (top and bottom). When the user put the prop in the origin, then navigation is stopped. As the user moves the prop away from the origin, navigation speed increases in the direction of props movement. Similarly, as the prop is closer to the origin, the navigation speed is reduced. This velocity range can be controlled at the application level.

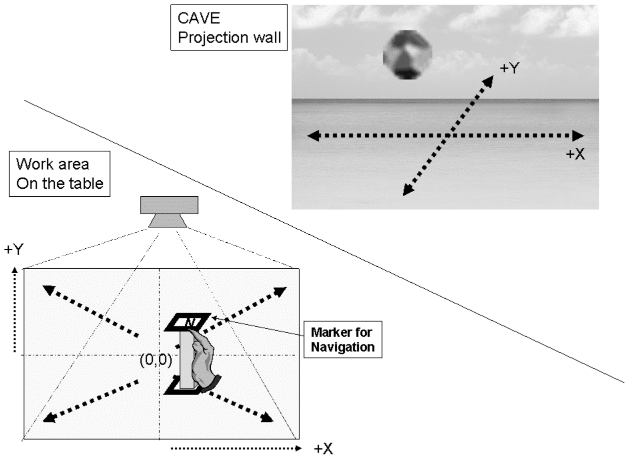

Figure 7: Navigation

4 DISCUSSION AND EVALUATION

The described system provides for direct manipulation of physical objects and alleviates some of the related concerns [12]. Compared to the common dedicated input devices such as wand or DataGlove, the relative simplicity of the system means that the required system resources are relatively smaller compared to similar systems. Since a user uses physical objects and interacts with the system as if moving an objects, the user feel immersive and direct manipulation in the virtual world. Saving and loading marker's position and orientation provides for history and tracing mechanisms. That can provide for quantitative measurements of task performance and for replaying the user's interactions. Combination and interactions between markers provide for some useful interaction techniques "shortcuts." Direct manipulation of physical objects on a table makes it easier for users with disabilities to navigate and interact with the system.

Unlike a wand, which is a common input device for CAVEapplications, the described system allows use of two hands and direct manipulation. For example, when selecting virtual objects, the user does not need to use any extra interaction technique because each virtual object is simply selected or deselected along as the corresponding prop is manipulated. Hinckley et al. [5] described advantages of two-handed user interface design, such as time-saving and structuring users' tasks.

Informal pilot study has been conducted to evaluate the system. The primary focus was to compare between a TUI interface and a wand based interface. A half of the subjects that participated in this study already had some experience using a wand in the CAVE. The others used various applications in the computer vision area.

They performed three basic interaction tasks, selection, manipulation and navigation, both with TUI and the wand. After completion of those tasks, they filled out a questionnaire designed to measure their personal preference. Regardless of their previous experience, they all preferred TUI to a wand for several reasons. TUI is very intuitive, especially for selection, and do not require much training. Props are lighter and easier to handling and there are no cables to obstruct movements.

However, when more than two markers are manipulated at the same time, the camera's performance affects the frame rate and may result in increased latency. Since the light is positioned above the middle of the work area, it generates shadows of props as well as the user's hands. The light position, as well as the quality and performance of cameras, significantly affect marker recognition.

The marker's information may be lost due to fast movements because it takes some time for the camera to detect a marker. Sometimes, the frame is "frozen" due to a short disconnection of wireless networking and latency is increased.

4.1 User Feedback

Subjects were interviewed in person after completing the questionnaire. Overall user feedback shows that they all find it more intuitive and when they can physically handle props to manipulate virtual objects and explore the virtual world. However, they indicate some problems with the system.

- Position of the light: The position of the light (if used) may obstruct the user's line of sight and reduce their immersion. Unexpected shadows may affect marker detection. Additional lights, located within props, can be used to remove shadows.

- Camera use: The work area can be to small for some tasks, especially navigation, resulting in tiredness. Occasionally they did not realize that their hands are out of the work area and out of the camera's view.

- Props use: Props work very well for selection and manipulation tasks. However, it requires an effort to remember which prop is assigned to which virtual object, especially when there is a large number of objects.

5 CONCLUSION AND FUTURE WORK

This paper describes an ARToolKit-based TUI system integrated with the CAVE platform. The unique and novel combination of the existing components (ARtoolkit and CAVE) and ease of use are the main contributions of the described system. It provides a low cost (time and resource reuse) development environment for interaction techniques. The interface can be easily integrated within existing CAVE applications and with existing common interaction techniques that use a wand or other tracked input devices. This approach can be extended by using a small and attachable wireless sensors or RFID tags. The experiments also identified several problems, including effects of camera characteristics, the size limits of the work area, and handling of the fast user movements. The future work will involve studying connections and relationships among different parts of the system and customizing user interaction techniques for specific interaction tasks.

REFERENCES

[1] O. Bimber and R. Raskar. *Spatial Augmented Reality: Merging Real and Virtual Worlds*. A K Peters, Wellesley, Massachusetts, 2005.

[2] D. A. Bowman, E. Kruijff, J. J. LaViola, Jr., and I. Poupyrev. *3D User Interfaces: Theory and Practice*. Addison-Wesley, Boston, 2004.

[3] J. E. Carvalho, J. Luzio, P. Santos, M. Dias, and J. Jorge. Usability evaluation of tangible user interfaces for augmented reality. In *IEEE International Augmented Reality Toolkit Workshop*, Tokyo, Japan, October 2003.

[4] T. He and A. E. Kaufman. Virtual input devices for 3d systems. In *VIS '93: Proceedings of the 4th conference on Visualization '93*, pages 142–148, 1993.

[5] K. Hinckley, R. Pausch, D. Proffitt, and N. F. Kassell. Two-handed virtual manipulation. *ACM Trans. Comput.-Hum. Interact.*, 5(3):260–302, 1998.

[6] H. Ichida, Y. Itoh, Y. Kitamura, and F. Kishino. Interactive retrieval of 3D shape models using physical objects. In *Proceedings of the ACM Multimedia 2004*, pages 692–699, Oct. 2004.

[7] H. Kato, M. Billinghurst, I. Poupyrev, K. Imamoto, and K. Tachibana. Virtual object manipulation on a table-top AR environment. In *Proceedings of the IEEE and ACM International Symposium on Augmented Reality (ISAR) 2000*, pages 111–118, 2000.

[8] J. Kelso, S. G. Satterfield, L. E. Arsenault, P. M. Ketchan, and R. D. Kriz. DIVERSE: A framework for building extensible and reconfigurable device-independent virtual environments and distributed asynchronous simulations. *Presence*, 12(1):19–36, Feb. 2003.

[9] G. A. Lee, G. J. Kim, and M. Billinghurst. Directing virtual worlds: Authoring and testing for/within virtual reality based contents. In *Proceedings of the 14th International Conference on Artificial Reality and Telexistence (ICAT)*, Seoul, Korea, 2004.

[10] K. Matković, T. Psik, I. Wagner, and D. Gračanin. Dynamic texturing of real objects in an augmented reality system. In B. Fröhlich, S. Julier, and H. Takemura, editors, *Proceedings of the IEEE Virtual Reality 2005 Conference*, pages 245–248,329, Mar. 12–16 2005. Application sketches paper.

[11] M. Odile Berger, B. Wrobel-Dautcourt, S. Petitjean, and G. Simon. Mixing synthetic and video images of an outdoor urban environment. *Machine Vision Applications*, 11(3):145–159, 1999.

[12] B. Schneirdeman and C. Plaisant. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison-Wesley, Boston, fourth edition, 2005.

[13] S. Singhal and M. Zyda. *Networked Virtual Environments: Design and Implementation*. ACM Press SIGGRAPH Series. Addison Wesley, Reading, Massachussetts, 1999.

[14] J. Vince. *Virtual Reality Systems*. ACM SIGGRAPH Books Series. Addison-Wesley Publishing Company, Wokingham, England, 1995.