Case Studies in Building Custom Input Devices for Virtual Environment Interaction

Joseph J. LaViola Jr. Daniel F. Keefe Robert C. Zeleznik Daniel Acevedo Feliz
Brown University
Dept. of Computer Science, Box 1910
Providence, RI, 02912, USA
{jjl,dfk,bcz,daf}@cs.brown.edu

Abstract

We present examples of four different custom built input devices and discuss their benefits to virtual environment interfaces. In each case, we describe the underlying motivation behind the device's creation and present interaction techniques that were made possible by each device's unique characteristics.

1 Introduction

In virtual environment (VE) applications, users must express their intentions through various input devices. At Brown University, an important research focus has been the development of custom built input devices for improving a user's expressive power when interacting in VEs. These devices often inspire new interaction techniques, improve the interface in specific VE applications, provide more interaction styles, and improve upon existing techniques. In this paper, we present four different custom built input devices, the FingerSleeve, the CavePainting Table, the Interaction Slippers, and Flex and Pinch input as examples of custom built input devices used in the development of our VE applications and interface research. For each device, we also describe some of the interaction techniques developed using them.

2 FingerSleeve

Building custom input devices often provides inspiration and creativity in the development of novel interaction techniques. The FingerSleeve is an example of such a device that explored how the concept of pop-through buttons could be used to improve upon traditional button based devices[9]. Pop-through buttons have two clearly distinguished activation states corresponding to light and firm finger pressure.

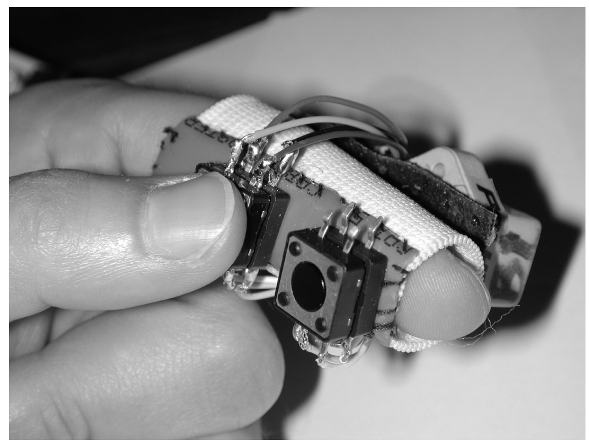

Figure 1. The FingerSleeve device mounts two small pop through buttons on an elastic frame, with the tracker placed on the back of the sleeve.

This design adds functionality without increasing device obtrusiveness.

2.1 The Input Device

The FingerSleeve, shown in Figure 1, is a device that can be worn on the index finger of either the left or right hand. The frame is made out of an elastic fabric and a small piece of flexible plastic that can be found at any arts and crafts store. The fabric is sewn into a sleeve with a varying diameter that fits snuggly for most users. The plastic is sewn onto the front of the sleeve to provide a solid mount for the pop through buttons. The buttons are glued into place a few millimeters apart on top of the plastic. Finally, a 6 DOF tracker is secured to the back of the sleeve using velcro.

A primary design consideration in creating the Finger-

Sleeve was selecting appropriately sized buttons. If the buttons protrude too far from the sleeve housing, the pressing gestures needed to activate them can be uncomfortable. The buttons we chose are small enough that users can operate the device comfortably. Both pop through buttons are constructed using two tactile switches with the same geometrical layout in width and length, but slightly different heights. The base button's switch is raised slightly above its mount enabling the exposed (top) button to be placed on the raised switch. This configuration has a smaller force differential than our previous pop through designs but is still easily controlled, perhaps because of the extra sensitivity of thumb-index finger interaction.

2.2 Interaction Techniques

A convenient way to facilitate the exploration of an environment is to allow users to quickly inspect a distant location in order to be automatically transported there, and then after arriving, make the decision about whether to stay or return to the starting location. Mine[7] previously explored this inspection navigation style using Head-Butt Zoom; however, pop through buttons, leveraging the naturally sequential nature of this inspection task, enable an alternative that requires significantly less user activity.

The *ZoomBack* technique allows a user to select a target point on the surface of an object in a virtual environment using a virtual laser pointer that continuously emanates from the FingerSleeve (see Figure 2). Then, by pressing a button lightly, the user is translated directly toward that target point such that he ends up two feet in front of the targeted point in approximately two seconds. If the user then releases the button, he is returned to his original location, again in two seconds. Alternatively, if the user presses firmly on the button to pop through, then his location is "locked" so that he can remain where he is after the button is fully released.

We believe the ZoomBack technique exemplifies a generally effective principle for mapping application behavior to buttons: that light pressure performs a temporary action that must be confirmed by firm pressure. This notion was supported by informal testing in a mock-museum environment where users found the device mapping to be natural, and the technique effective for moving about.

The *Snapshot* technique for taking pictures from within a virtual environment is a representative sequential operation technique. With the FingerSleeve, users invoke a simple cropping widget by pressing lightly. By pressing harder, the user takes a snapshot of the area seen through the frame of the widget. Since the size of the widget frame is constant, users move the frame closer to or farther from their heads to modify the region of the virtual world that will appear in the snapshot image. These images are stored in a wall-menu. By pointing to a snapshot on the wall-menu, and pressing the same button lightly, users are temporarily transported back to the place where the snapshot was taken. Similar to the ZoomBack technique, releasing this button returns the user to the original position; whereas applying additional pressure to the same button to pop through leaves the user in the location indicated by the snapshot. In this case, the wall-menu includes an option for returning to the previous location.

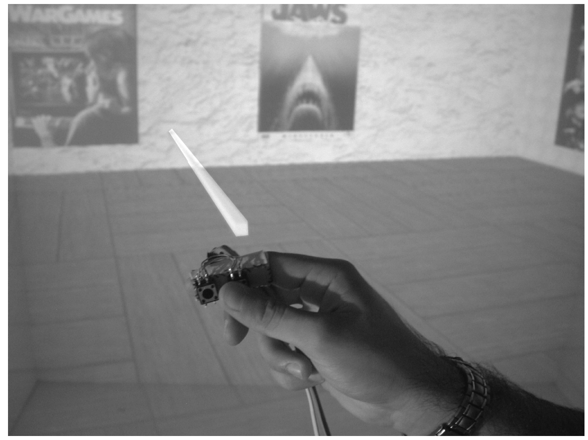

Figure 2. The primary axis for the FingerSleeve tracker is perpendicular to the user's finger orientation. The image shows a virtual laser pointing in that direction.

Taking snapshots with the cropping widget is similar to taking pictures in the real-world with conventional cameras that have a two-level shutter release mechanism. In informal evaluations, users claimed to have no difficulty controlling the FingerSleeve device for either taking snapshots or controlling the temporary and permanent transitions using the wall-menu of snapshots.

3 CavePainting Table

In many cases, focusing on specific VE application tasks provide a conduit for creating custom built input devices that increase the user's expressive power in the application. CavePainting, a tool for creating 3D artistic scenes, is an example of such an application. The CavePainting [1] table was developed in an effort to explore natural prop-based interfaces inside a Cave and provide the user with an interface specifically designed for 3D painting.

A painter often works effortlessly with 5 or 6 paintbrushes at a time. When modeling this type of interface in a Cave, where do the brushes go when not in use? Keep in mind, "in my other hand", might not be the best answer to this question if the interface requires a glove, FingerSleeve (see Section 2), or other input device to be used in the user's non-dominant hand. A slippery glove or encumbering device can turn the task of switching brushes into one that completely distracts from the relevant task, creating 3D works of art.

3.1 The Input Device

The CavePainting table uses a prop-based design that relies upon multiple cups of paint and a single tracked paintbrush, as seen in Figure 3, rather than multiple brushes. These paint cup props stay on a physical table that slides into the Cave and also houses knobs and buttons that are used for various interaction tasks. This table does disrupt the projection onto the Cave walls, but it is placed in the back corner of the Cave, so it does not affect the projection in key working areas of the Cave. CavePainting is our most successful Cave application in terms of enticing a user to move around the physical space of the Cave. Users are quite active and often pace back and forth as a real painter often walks toward and away from her easel. Thus, the fact that the placement of the table often requires the user to walk over to the back corner of the Cave is not a significant distraction for this application.

In conjunction with the table, a real paintbrush was augmented with a single button which turns the "paint" on and off. The bristles of the brush are covered with conductive cloth[6]. This allows us to sense a circuit between the brush bristles and similar cloth that lines the inside of the paint cups on the table.

Figure 3. The painting table interface.

3.2 Using the CavePainting Table

The main CavePainting interaction technique is to hold down the button on the brush and move one's hand through the air. The deposits a trail of virtual paint along the path that the brush takes. The stroke that comes out of the brush can take on many forms. The form is changed by dipping the brush into one of the paint cups on the table. These "contain" the various stroke types. When the brush touches the inside of the cup a closed circuit is formed and the program responds with an audible cue. Our users found this technique immediately understandable and fun. We even observed one group of middle school students repeatedly dipping the brush into the cups and moving it back and forth, as if to make sure it was fully coated with paint! This speaks to the power of application specific devices in immersive environments and their potential for increasing one's sense of presence.

A typical approach to Cave interfaces is to map vast amounts of the functionality of an application onto the trackers and buttons of generic wands held in the user's hands. While this technique has advantages in its ease of implementation and its ability to be easily reproduced given many varied hardware configurations, it can often distract a user from important tasks by requiring her to work through a menu or similar widget in order to activate commonly accessed commands. In contrast, the CavePainting table provides a mechanism for unloading much of the application's functionality from the hands to physical controls in the back of the room. This leaves the hands free to immediately perform the most common and most important tasks. In CavePainting, these are painting, grabbing and moving the painting, and resizing the virtual brush.

4 Interaction Slippers

Another example of how custom built input devices affect virtual environment interaction is in providing more powerful methods of expression. In many cases, offloading interaction from the user's hands to the feet can improve an interface, especially when the user's feet are used in navigation tasks. However, foot-based input devices are not common. Therefore, we developed the Interaction Slippers (see Figure 4) for exploring how the user's feet could be used in virtual environment interaction[2].

4.1 Input Device

Two important design considerations when creating the Interaction Slippers were that they be both comfortable and untethered. We addressed these considerations by embedding a Logitech Trackman Live!^{TM} wireless trackball device that uses digital radio technology[4] into a pair of commercially available slippers. We chose wireless radio technology over other approaches, such as infrared, because it provides a range of up to 30 feet, and does not require unoccluded line-of-sight to a sensor. We inserted the Trackman into a hand-made pouch on the right slipper and rewired two[1] of the Trackman's three buttons by connecting each one to a pair of conductive cloth[6] patches on the instep of the right slipper. On the instep of the left slipper, we placed two more conductive cloth patches. Touching a cloth patch on the left slipper to a cloth patch pair on the right slipper completes the button press circuit. This design enables us to distinguish two gestures corresponding to heel and toe contacts respectively.

Figure 4. The Interaction Slippers allow users to tap either their toes or heels to invoke operations.

4.2 Using the Slippers to Navigate

The Interaction Slippers were used to interact with the Step WIM (shown in Figure 5), a miniature version of the world that is placed on the ground, under the user's feet in the virtual environment[2]. To invoke the display of the Step WIM with these Slippers, the user taps his or her toes together, establishing a conductive cloth contact which is easily sensed and treated as a "button" press. Once displayed, the user can move to a new location by simply walking to a desired place in the Step WIM and clicking the toes together again, while looking at the Step WIM. To dismiss the Step WIM, the user makes the same clicking gesture while looking away from the floor. The inspiration for this technique is from the scene in The Wizard of Oz where Dorothy taps her heels to return to Kansas.

5 Flex and Pinch Input

In many cases, a given input device has certain properties making it easy to perform certain operations but not others. For example, bend sensing gloves can be used to mimic interface widgets such as sliders and dials [8], but do not have useful methods for signaling the activation or deactivation of the widgets. In contrast, Pinch gloves provide a series of button widgets that are placed on each finger tip allowing for discrete pinching postures, but they have no way of determining the flexing of the fingers. Interaction techniques are often incomplete because of these limiting factors. The Flex and Pinch input system (see Figure 6) was developed in order to combine bend sensing gloves and pinch style input to improve upon interaction techniques[3].

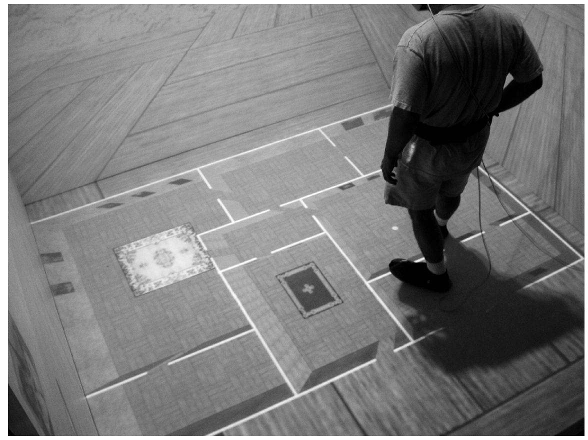

Figure 5. The Step WIM widget allows users to quickly navigate anywhere in the virtual world. The small sphere by the user's foot indicates his position in the miniature.

Figure 6. The Flex and Pinch input system. Although a CyberGlove™ is shown, any bend-sensing glove can be used.

[1] Our current implementation of interaction slippers utilizes only two of three Trackman buttons. In future work we plan to use of the third button as well as the trackball.

5.1 The Input Device

We constructed a device based on the Fakespace Pinch™Glove. As a hardware input device, it provides more functionality than the Pinch Glove since it uses eight cloth buttons instead of five which allows for more button combinations. In general, five of these cloth buttons can be placed around each of the finger tips, while the other three can be placed arbitrarily about the hand[2]. These cloth buttons can be placed on a bend-sensing glove providing both bend angle measurements (Flex) and pinch style input (Pinch).

5.2 Improving Existing Techniques

With Flex and Pinch input, we can improve on a number of existing techniques for selecting objects in virtual environments. For example, one of the major problems with the image plane interaction techniques such as the *head crusher*, *sticky finger*, *lifting palm*, and *framing hands* object selection techniques[5] is that the user cannot activate the selection with the primary hand. As a result, the user requires an additional, separate input device for triggering the selection operation.

Flex and Pinch input provides a simple yet effective and seamless method for starting and stopping object selection by placing the cloth buttons in appropriate places on the users primary hand. For example, with the head crusher technique, we can place the cloth buttons on the thumb and middle finger so when the user positions the thumb and forefinger around the object (using the bend angle information to detect the posture) a middle finger to thumb contact signals the object should be selected. Another button press would signal the release of the object. The cloth contacts can be placed in other positions such as on the middle finger and on the palm by the base of the thumb or on the right side of the index finger and the left side of the middle finger. In a similar manner, cloth contacts are placed on the hand for the sticky finger and lifting palm techniques to start and stop object selection while cloth contacts are placed on both hands for the framing hands selection technique.

6 Conclusion

We have presented four different examples of custom built input devices used in virtual environment interaction. The FingerSleeve inspired the creation of novel interaction techniques. The CavePainting table was used to improve a specific VE application. The Interaction Slippers provided new methods of expressive power through their creation. Finally, Flex and Pinch input was used to improve existing interaction techniques limited by the devices for which they were originally designed. These examples illustrate that building custom input devices can be a valuable component of interaction research and in the development of VE applications.

[2]This presents one of many possible combinations for placement of the cloth buttons. The device could have be worn with anywhere from two to 16 cloth buttons of any shape or size. This presents a clear advantage over other inflexible input devices.

References

[1] Keefe, Daniel, Daniel Acevedo, Tomer Moscovich, David Laidlaw, and Joseph LaViola. CavePainting: A Fully Immersive 3D Artistic Medium and Interactive Experience. In the *Proceedings of the 2001 Symposium on Interactive 3D Graphics*, ACM Press, 85-93, 2001.

[2] LaViola, Joseph, Daniel Acevedo, Daniel Keefe, and Robert Zeleznik. Hands-Free Multi-Scale Navigation in Virtual Environments. In the *Proceedings of the 2001 Symposium on Interactive 3D Graphics*, ACM Press, 9-15, 2001.

[3] LaViola, Joseph, and Robert Zeleznik. Flex and Pinch: A Case Study of Whole Hand Input Design for Virtual Environment Interaction. In the *Proceedings of the Second IASTED International Conference on Computer Graphics and Imaging*, 221-225, October 1999.

[4] Logitech, http://www.logitech.com, 2004.

[5] Pierce, Jeff, Andrew Forsberg, Matt Conway, Seung Hong, Robert Zeleznik, and Mark Mine. Image Plane Interaction Techniques in 3D Immersive Environments. *Proceedings of the 1997 Symposium on Interactive 3D Graphics*, ACM Press, 39-44, 1997.

[6] Mann, Steve. Smart Clothing: The Wearable Computer and WearCam. *Personal Technologies*, Volume 1, Issue 1, March, 1997.

[7] Mine, Mark, Fred Brooks, and Carlo Sequin, Moving Objects In Space: Exploiting Proprioception In Virtual Environment Interaction. In *Proceedings of SIGGRAPH 97*, ACM Press, 19-26 , 1997.

[8] Sturman, D. and D. Zeltzer. A Survey of Glove-based Input. *IEEE Computer Graphics and Applications*, 14(1):30-39, 1994.

[9] Zeleznik, Robert, Joseph LaViola, Daniel Acevedo, and Daniel Keefe. Pop Through Buttons for Virtual Environment Navigation and Interaction, In the *Proceedings of Virtual Reality 2002*, IEEE Press, 127-134, 2002.