The Virtual Remote Control - An Extensible, Virtual Reality, User Interface Device

Michael Rorke, Shaun Bangay

Computer Science Department
Rhodes University
Grahamstown, 6139
South Africa
{mrorke; cssb}@cs.ru.ac.za
http://cs.ru.ac.za/homes/g97rc001/

Abstract

Immersive virtual reality (VR) places the user inside of the computing environment, blurring the distinction between the environment itself and the user interface to access that environment non-obvious. This lack of distinction between environment and interface makes it difficult to place menus and other interface elements where the user is both able to access them easily and where they do not obscure large parts of the users field of view.

We propose a system called the 'Virtual Remote Control' (VRC). The VRC consists of a physical device (a small touch-pad tracked using a Polhemus InsideTrak magnetic tracker) which the user is able to hold and for which there is a representation in the virtual environment. The VRC is represented in the environment by a virtual menu. The user is able to make selections from the virtual menu by moving their finger around the touch pad part of the VRC and 'tapping' on the required action. Additionally, the user is able to select an object for the action to be applied on, by 'pointing' the representation of the VRC at the object - as one would point a remote control at a Hi-fi of TV set.

This device offers an interface to a set commands, through a uniform, immersive system. Furthermore, it offers methods for constructing dynamic, intelligent (i.e. able to reflect some aspect of the current system/user) and re-usable immersive interfaces for VR.

1. Introduction

> *"In an immersive environment, one can directly perceive and manipulate three dimensional objects instead of interacting with abstract interface elements"* [2]

The placement of the user inside of the computing environment via head-mounted displays, magnetic trackers, etc. has many advantages, but is not without disadvantages. In the realm of interactive applications, where the user attempts to alter the system into which they have been placed, the lack of much of the input information that we take for granted in everyday life, (haptic feedback for example) often leaves the user disorientated and with an inability to complete even simple tasks [3]. Furthermore, the desktop interface metaphor of menus and simple control widgets is no longer applicable when the user is inside the computing environment, and therefore inside the interface itself.

> *"The differences between working in a conventional environment and working immersed are analogous to the differences between a craftsman at a workbench and one moving around a worksite wearing a toolbelt. His toolbelt had better be large and filled with powerful tools."* [2]

This paper presents a method whereby the user is able to easily and directly access the control elements of a system, via a real world control device - the VRC.

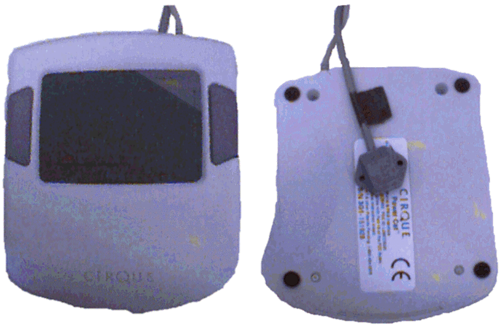

Figure 1 – The VRC, Top and bottom

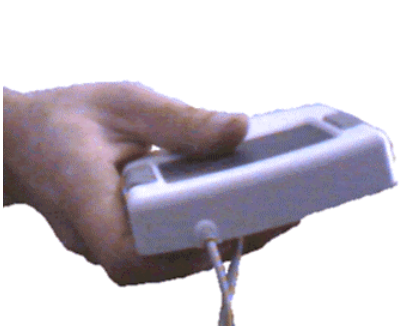

Figure 2 – The VRC 'in hand'

2. The Virtual Remote Control

Essentially, the VRC is a *single handed* i.e. meant to be held in one hand only, input device for making a choice from among a given set of options. The VRC system consists of a Cirque Power Cat touch-pad device, attached to a Polhemus InsideTrak 6DOF magnetic tracker (Figure 1). The touch-pad has an active surface measuring 77 x 55 mm (3 x $2^1/_4$ in). The device is held in the hand (Figure 2) with the thumb in contact with the touch-pad surface. The thumb is moved over the surface of the touch-pad, interactively positioning a cursor object over a representation of the VRC in the computing environment (Figure 3). Once the cursor is over the desired action, the thumb is *tapped* on the pad and the action under the cursor is selected. The VRC has been implemented as part of the Rhodes University CoRgi distributed virtual reality project. The system was developed on SGI Octane and Intel Pentium II systems and executes under the Irix and Linux operating systems.

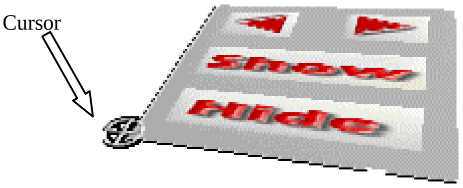

Figure 3 - the VRC object in the virtual environment

The format and layout of the buttons on the surface of the VRC object in the virtual environment is specified via a configuration file and can be dynamically changed during execution. The buttons themselves are texture-mapped objects, with the texture conveying the details of the particular action, to the user. The VRC allows for actions to be applied to particular objects, by allowing the user to select an object by *pointing* the VRC at the object before choosing the action required. This action is analogous to pointing a real world remote control at a TV or Hi-fi set, before pressing a button.

The example in Figure 3 shows a set-up for an immersive presentations application where the VRC is used to move between slides and to *show* and *hide* the current slide from the user.

3. Advantages

Unlike other methods of issuing commands inside an immersive VR application (e.g. using gesture recognition) where there are a finite number of easily recognised commands, the VRC offers a large number of commands that are issued through a single common interface. The interface is uniform in that any action, not matter how simple complex it is to implement in the system, is accessible to the user through a simple common interface. The introduction of a common interface for issuing commands allows the user to lower the learning curve required to learn an application that uses the VRC [3]. The user is also able to apply that knowledge to other systems that use the same interface. This uniformity of the user interface is one of the accepted bases for the success of the WIMP (Windows, Icon, Mouse and Pointer) interface currently seen on all desktop machines.

The user is able to relate the haptic feedback they receive from holding a real-world object to the image of the object that they see in the virtual world. Additionally, any movement of the physical VRC device in the real world creates a corresponding movement in the virtual representation of the device. This coupling of a real object with a virtual representation of that object reinforces the user's sense of presence, making the virtual world seem more real. This feeling of presence allows the user to exploit the natural spatial reference provided by the sense of their own body, called *proprioception* [1, 2, 4]. Utilising proprioception has been found to increase a users effectiveness in a virtual environment, allowing them to interact more intuitively and precisely with the virtual environment [1, 2]. The VRC system offers an advantage over similar systems which use a fixed device for interaction in the world [2] in that the user achieves an increased sense of presence by being able to move the VRC around the world and not being forced to stay in some fixed location. The ability of the VRC to be moved around also provides a far more flexible interface for interaction.

Many VR systems use input devices, like datagloves, to mimic the direct human-object interaction of the real world. Yet, humans typically interact with their world through *tools* [4]. The VRC mimics a tool common to most users - a remote control. The user is therefore able to apply their intuitive knowledge of the real world to the virtual world. Glove interfaces are also problematic to implement effectively [4]. For example, the choice of what gestures to assign to what actions is often based on the personal bias of the developer and there is usually no indication in the system as to what gestures correspond to what actions. The VRC on the other hand provides a simple interaction paradigm, with the ability to annotate the representation of the device in the virtual environment, to give some indication of what action a particular *button* produces. For example, in the texture selector application (see the *Applications* section) each button is texture mapped with the texture that it will produce on the selected object, when activated. The user is able to make a choice as to what texture would best suit their needs before performing any actual actions in the system. The Virtual Tricorder Device [4] is another example of a real world based, tool interaction device similar to the VRC in the advantages it provides. The VRC improves on this design by offering a flexible interface for obtaining actions from the user. The VRC is not limited to a fixed number of action points (i.e. like the buttons on a mouse), rather the introduction of a touch pad turns the whole surface of the device into a possible action point, offering greater flexibility in the design and implementation of the interface offered to the user.

The fact that the VRC is made to be used with only one hand is another advantage of the system. Other touch pad systems use larger touch pads that require both of the user's hands to operate, while the VRC frees the user's other hand for use with a second interaction device. For example, the texture selector application (see the *Applications* section) allows the user to use their other hand to manipulate the walls in the scene, using a standard *virtual hand* paradigm whereby the user manipulates a virtual version of their hand with which they are able to complete actions like moving or selecting objects in the scene. The VRC acts as part of a multi-modal interface where, as in the real world, the user is able to use both their hands when interacting. This allows for better use of the user's hands and a more realistic way of interacting with the system.

4. Dynamic Menus

The format and layout of the buttons, as well as the actions each performs, can be changed dynamically while the application is running. This allows for a potentially limitless amount of possible actions. Furthermore, the dynamic nature of the layout provides the basis for the implementation of various different methods of choosing what commands are presented to the user.

The layout of the VRC may be set depending on some internal context of the system. For example, if an object in the world has been selected (either by the VRC itself, or by some other interface element e.g. a virtual hand) the VRC presents the user with commands which allow the user to manipulate this object in some way (e.g. scale or translate). On the other hand, if no object is selected, the VRC presents the user with commands which allow them to move around the environment (e.g. walk or fly).

The application is able analyse which of the commands are requested most frequently and present these on the default menu. Instead of the user having to switch between multiple menus to find some commonly used command, they are immediately presented with it. The particular actions that appear on this default menu are not set, rather they vary dynamically during application execution, depending on what the user is doing. Since the menu layout is controlled via the application itself (and not built in to the VRC) a decision about when to change them can be based on any parameters the developer cares to implement.

The method used for switching between the various menu layouts is also dependant on the application and not on the VRC itself. For example, the virtual cube application (see the *Applications* section) maps various different menu layouts to the faces of a cube. The user is able to select actions from the layout that they are currently viewing and able to move to another layout simply by rotating the VRC by 90^0.

5. Applications

We have implemented several applications that make use of the VRC. Following are descriptions of a few and how the VRC is specifically suited to that particular application.

5.1 Virtual Keypad

One of the most difficult forms of input to obtain in a virtual environment, is direct numeric input. For example, objects often need to be precisely placed in the environment and the only way to accomplish this is to enter the exact required position of the object in a numerical form. Mark Mine [2] explains various methods for entering numeric data through a virtual representation of, for example, a numeric keypad, which the user points at to choose the number they require.

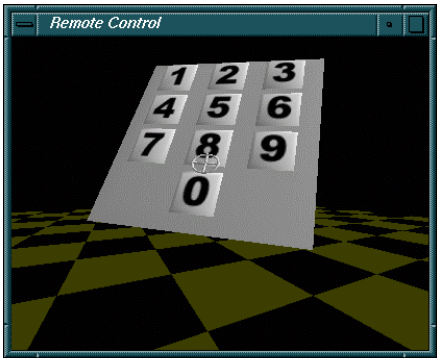

Figure 4 - The Virtual Keypad

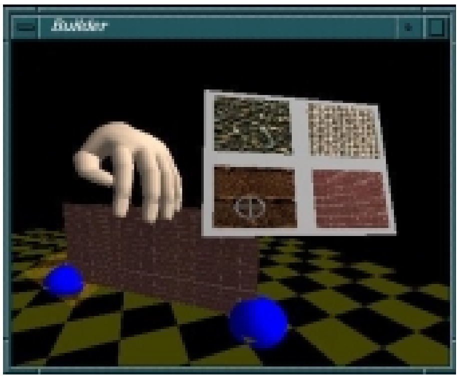

Figure 5 - The Texture Selector

The VRC (Figure 4) provides another method for obtaining direct numeric input. The VRC can be used to emulate a simple keypad device, thus allowing simple, direct input of numeric (or any *exact* for of data) into the system. The VRC keypad system allows the keypad to be placed where it does not interfere with or obscure the object being placed, for example, while still allowing the user to effectively enter the numeric data. This method also applies to entering of text data. Here, the VRC is used to simulate a complete keyboard.

5.2 Texture Selector

The texture selector system allows the user to interactively choose an object in the virtual world and then use the VRC to choose the texture that is to be mapped onto that object. The example here (Figure 5) shows a simple architectural walkthrough modeller, where the user is able to place walls (using their other hand in a *traditional* virtual hand interaction metaphor) and, by selecting a wall and choosing the appropriate action from the VRC, texture them for the desired effect. The VRC in this case gives the user a good idea of what texture they are applying, by showing a sample of it on the button that gets pressed to apply the texture.

5.3 Virtual Cube

The virtual cube application is an example of a method used to switch between different layouts for the VRC. The virtual representation of the VRC in this case is a cube, with different button layouts mapped to each face. The user is able to make selections (using the VRC) from among the actions on the face that they are currently viewing. By rotating the VRC device by 90^0, they are able to signal the application to rotate the cube, and present them with a different set of possible actions.

6. Conclusion

The VRC offers the user a simple way of selecting a specific action, from a group of actions, inside an immersive VR application. The use of the touch-pad to replace the *physical buttons* found on similar VR input devices [4] provides for a much greater flexibility when defining which actions the device is able to initiate at a particular time, and how these actions are accessed by the user. The ability of the device to be move in the real world and to be tracked for accurate display in the virtual world, provide a good sense of presence to the user promoting a more intuitive interaction with the system. The design of the device, to be used with only one hand, offers an advantage over other larger, two-handed touch pad devices in that it frees the users other hand to operate another form of input device, allowing for full multi-modal input. The ability for the user or designer to dynamically select from a range of possible VRC layouts enables large numbers of commands to be issued through a single, common, interface. Finally, the VRC relays information to the user about the specific actions that are possible, using texture maps to overlay text or diagrams of some sort onto the button that initiates the action.

The VRC has been shown to have various uses in immersive VR applications. It is fairly cheap and easy to construct, and offers a uniform interface to actions which would otherwise be difficult for the to conduct in the virtual world.

7. Future Work

Future plans for the system involve extending the idea of a simple virtual menu, to a more complicated device with various types of active widgets (e.g. scrollbars, switches, etc.) as opposed to the current system that allows for simple button widgets only. An example of this could be a virtual book, where the VRC represents the pages and you are able to turn over pages, set bookmarks, make memos, etc. Another possible extension would be to include the ability to change the shape of the VRC to provide some further indication of its current use. Additionally, we plan to conduct several usability studies to test the effectiveness of the tool in an *end user* situation.

8. References

[1] Mine, Mark, Frederick P. Brooks Jr., and Carlo Sequin (1997). ***Moving objects in Space: Exploiting Proprioception in Virtual-Environment Interaction***. Proceedings of SIGGRAPH 97, Los Angeles, CA

[2] Mine, Mark. ***Exploiting Proprioception in Virtual-Environment Interaction***. Doctoral dissertation, Department of Computer Science, University of North Carolina, Chapel Hill, 1997.

[3] Rorke, M., Bangay, S. and Wentworth E., ***Virtual Reality Interaction Techniques***. Proceedings of the 1st Annual South African Telecommunication, Networks and Application Conference (SATNAC), Cape Town, South Africa, September, 1998, pp. 526-532.

[4] Wloka, M. and Greenfield, E., ***The Virtual Tricorder: A Uniform Interface for Virtual Reality***. In Proceedings of UIST'95, ACM Press, November, 1995, pp. 39-40.