Wheelie - Using a Scroll-Wheel Pen in Complex Virtual Environment Applications

M. Wögerbauer
VRVis – Research Center for Virtual Reality and Visualization, Ltd
Donau-City-Strasse 1
A-1220, Vienna, Austria
mwoegerb@vrvis.at

A. L. Fuhrmann
VRVis – Research Center for Virtual Reality and Visualization, Ltd
Donau-City-Strasse 1
A-1220, Vienna, Austria
fuhrmann@vrvis.at

ABSTRACT

Input devices and system control techniques for complex virtual environment (VE) applications are still an open field of research. We propose the use of a scroll-wheel as an extra, dedicated input stream on a tracked stylus for means of system control. We demonstrate how this enhanced stylus can be used together with an appropriate user interface to quickly select commands, change tools, and adjust parameters.

This user interface consists of two different styles: a toolbar and a graphical menu system, both accessible by the same hand that holds the stylus. The scroll-wheel extension does in no way impair the conventional use of the stylus.

Keywords

Input Devices, Scrolling, Virtual environments, Application control, System control, Tool selection, Menu system, Responsive Workbench, 3D Interaction

1. INTRODUCTION AND MOTIVATION

Virtual environments (VEs) promise a great amount of potential as a working environment for applications with a high degree of interactive complexity. They provide a virtual workspace for the user in which his or her hands can be used to grab and manipulate virtual objects in complex workflows. In these workflows, however, we need to make extensive use of overhead tasks like tool switching, adjusting parameters or issuing commands to the system, aside from the main task. Thus, there is a great demand of user interfaces that are capable of performing these so called *system control* tasks efficiently.

In VEs the user interaction can be categorized into four classes of universal interaction tasks [Bow99a].

Journal of WSCG, ISSN 1213-6972, Vol.14, 2006
Plzen, Czech Republic.

Navigation changes the viewpoint in the environment, and can further be divided into a cognitive part (wayfinding) and a motor part (travel). *Selection* refers to the task of choosing one or more objects from a set which is closely related to the third task of *manipulation*. Manipulation can be described as changing the properties of objects, such as their location in the scene. The above mentioned task of *system control* subsumes all actions that apply commands to change the mode of interaction or the system state. While three-dimensional (3D) navigation and object selection and manipulation were the focus of research in past years, the equally important task of exploring novel system control techniques was left behind. Often application designers settled for using 2D desktop methods implemented in VE applications. In 2D desktop environments the WIMP paradigm (Windows, Icons, Menus and Pointers) is usually adopted as the means of controlling applications. Unfortunately, these interaction techniques cannot be effectively transferred to the 3D world of VEs. 3D interaction methods that employ 2D concepts struggle with the presence of additional degrees-of-freedom (DOF), originating from the extra dimension which hampers the otherwise easy task of selection in menus. When circumventing this problem by bringing 2D into 3D VEs [Lin99a] [Sza97a], i.e. letting the user hold a physical tracked tablet to execute 2D tasks on, we

sacrifice the use of one hand to the passive task of holding and gain no additional functionality directly controllable by one hand at the spot of direct manipulation, which would be preferable in complex workflows.

The most widely used input devices on 2D desktop setups are keyboards, computer mice and graphic tablets using a stylus. Their basic concept has remained the same since their advent in the computer world, but even they are still subject to refinement. One important step of enhancing usability of desktop systems was the introduction of the scroll-wheel for computer mice. In document browsing and navigation they take on the otherwise distracting task of scrolling from the main task of reading and editing. The focus of attention does not have to be switched to moving the pointer to a scroll bar, dragging a slider, and then switch back the attention to the document to actually see where we are scrolling. As scroll-wheels have become a de facto standard for mice, it appears to be a logical step for us to equip a stylus, being one of the most frequently used input devices in VEs, with a scroll-wheel as well, and to take advantage of this additional input stream. It will provide application and user interface designers with supplementary input to permit new ways and methods of interaction, not limited to the interaction methods presented in this paper.

We propose a new input device, called *Wheelie*, combined with an appropriate user interface. Wheelie is a multi-stream input device, a tracked stylus with an embedded scroll-wheel that speeds up the execution of many system control tasks found in complex VE applications. It is designed with the purpose to provide a quick way of consecutive selection of different editing tools for direct manipulation or creation of objects, and the possibility of issuing commands extended in a new type of menu system that takes advantage of the constrained input of the scroll-wheel. Wheelie and its user interface are a general purpose approach that can be combined with several other input devices and interaction techniques.

2. RELATED WORK

In spite of the fact that system control tasks account for a large number of interactions, usability of application control in complex VE applications is still mediocre. Research on appropriate interaction techniques and input devices was long neglected and is an open problem. However, much has been done in recent years to mend this situation and some interesting work has emerged.

Kruijff [Kru00a] proposes a categorization of currently used system control methods influenced by the description of non-conventional control techniques by McMillan et al. [McM97a]. It distinguishes between graphical menus, voice commands, gestural interaction, and tools. The group of graphical menus is further divided into hand-oriented menus, converted 2D menus, and 3D widgets. Tools can be divided into physical and virtual tools.

In an attempt to overcome some of the drawbacks of conventional 2D menus transferred to 3D environments and to take advantage of proprioceptive "eyes-off" interaction with the menu, i.e. the person's sense of the position of the body and limbs, Bowman and Wingrave [Bow01a] presented a new menu system called TULIP. It is based on displaying menu items on top of each finger and selecting them by pinching of fingers, using a Pinch Glove™.

Grosjean and Coquillart [Gro01a] developed a novel quick-access menu system for workbench-like VE configurations, called C³ (command and control cube). It is a 3D extension of marking menus, taking advantage of the three dimensions of the space for the selection process. It consists of a 3D grid of small cubes with which commands are associated and executed by positioning a selection pointer in the corresponding cube, using a tracked input device. It is designed with the intent to allow the possibility of rapidly issuing a limited set of commands to the application, similar to hotkeys in 2D desktop environments equipped with a keyboard.

To integrate 2D interaction in 3D VEs, Coquillart and Wesche [Coq99a], and in a similar approach Schmalstieg et al. [Sch99a], proposed the use of transparent props for two-handed application control in projection-based environments. The virtual palette and the PIP (personal interaction panel) consist of a physical tracked transparent plate that is held in the non-dominant hand and can present 2D menus and widgets which are selected by using a tracked pen. The physical surface, acting as a constraint, eases the task of selection and also takes advantage of prop- and body-centered aspects, delivering a kind of passive-haptic feedback.

A way of using the proprioceptive sense of the user is the use of body-centered menus which were explored by Mine et al. [Min97a] and place menu items relative to the user's body. This technique can significantly enhance user performance and even allows for "eyes-off" interaction and selection of menu items and tools. Body-centered menus do not inherently support a hierarchy of menu items and this poses a problem since the selection becomes gradually more difficult with increasing numbers of menu items.

In hand-oriented menus, as used in several applications as a method of system control [Lia94a] [Min97b], the menu items are located on a circular

object. In order to select an item the user has to rotate the hand to align the desired menu item in a selection box. The rotation about a single axis, working as a constraint, makes these menus fast and accurate. As is the case with body-centered menus, hand-oriented menus also do not imply a menu hierarchy.

A thorough description of two-handed direct manipulation on the Responsive Workbench was given by Cutler et al. [Cut97a]. It shows how users perform certain tasks in a natural way using both hands. Pinch Gloves and/or a stylus were used as input devices and virtual tools could be selected from a toolbox located in front of the user by either clicking on their representations with the stylus or by pinching them with the gloves. To reduce the number of necessary explicit tool switches, power tools, controlled by the non-dominant hand, and implicit tool transitions were implemented.

The ToolFinger by Wesche [Wes03a] is an interaction technique for VEs that focuses on the problem of frequent tool selection during complex direct manipulation of objects and proposes the integration of the task of tool selection into the workflow of tool application. This is accomplished by subdividing a selection pointer of a tracked stylus into several sections and interpreting an intersection of an object with one of those sections of the pointer as a tool selection. With each section of the ToolFinger a tool is associated. As a result of this method, the ToolFinger is restrained to interactions which are always related to virtual objects. Other interaction modes, e.g. object creation or general application control tasks, are not possible, making the ToolFinger a special purpose technique for direct modification.

In [Ste03a] Stefani et al. discuss requirements for input devices in immersive environments and the design of two such input devices is presented. The Dragonfly and the Bug are two input devices for the dominant and the non-dominant hand, respectively, which work in unison. The Dragonfly is a lightweight, optical tracked, stylus-like pointing device with 6 DOF, not featuring any button. The Bug is, in essence, a 3 DOF (position only) tracked wireless mouse with a jog-dial (or scroll-wheel) and two buttons, held in hand like a TV remote control. The 3 DOF nature of the bug allows the permanent display of a context sensitive, graphical menu and the jog-dial facilitates selection of menu items therein. This way control of the menu system is decoupled from tracking and instead uses the sole input of the jog-dial.

Many interaction techniques described herein are an attempt to overcome the problem of too many degrees-of-freedom of system control interfaces and introduce some sort of constraint in order to ease the task of selection in menus. Wheelie takes the same approach as the Bug when tackling this problem by adding a dedicated one-dimensional discrete input stream to the VE, allowing the user to effectively perform application control tasks without interrupting the current workflow.

3. DESIGN CONCEPT

The basic idea of Wheelie is to extend the input possibilities of a stylus used in many VEs as an input device by adding a scroll-wheel as is commonly used in computer mice. Thus, a separate input stream is available that can, in combination with two buttons, be used to navigate a hierarchical system of application control functionality (Figure 1). Tool selection and navigation in the menu system are mainly controlled by changing between different entries in a set of tools or menu entries using the scroll-wheel.

Several other devices incorporating an additional input stream already exist (e.g. the Wanda available at http://www.wandavr.com/). But most of them are palm held devices that feature a thumb controlled joystick. A joystick is neither as appropriate to perform one-dimensional discrete selection tasks – by virtue of its 2D continuous input stream – as a scroll-wheel, nor do most devices support the use of the extra input stream in conjunction with the other buttons. In this regard Wheelie – as a stylus with a scroll-wheel operated by the index finger and buttons activated by the thumb – offers new ways of interaction previously unavailable.

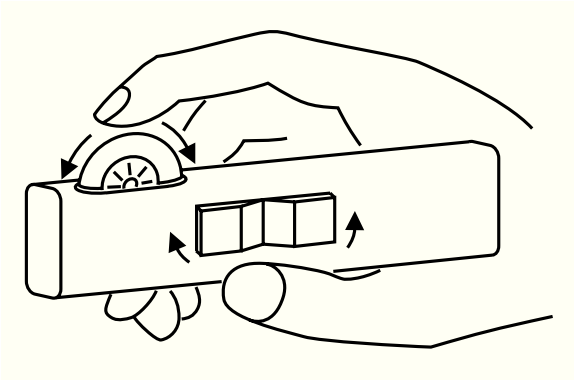

Figure 1: The conceptual design of Wheelie. The index finger is used to operate the scroll-wheel, while the thumb is used to activate two buttons on the side of the stylus.

Generally speaking, system control is the action in which a command is applied to change either the mode of interaction or the system state. In order to issue the command, the user has to select an item from a set and different interaction styles are employed in order to select the commands. As mentioned in the relating work above, these techniques

can be put into four categories: Graphical menus, voice commands, gestural interaction and tools. In the course of this work we want to use the following slightly customized terminology of system control tasks: A mode of interaction of direct manipulation or creation of objects is referred to as a *tool*, resulting in *tool switches* when changing between them. Functions of the application that change the state of the system or perform system tasks and should be accessible to the user for execution are called *commands*. If the value of a single variables is subject to explicit changes through the user (*parameterization*), the variable is called a *parameter* and the set of associated valid values is called the corresponding *parameter space*.

As the addition of a scroll-wheel to the computer mouse became a huge success and nearly all mice manufactured today feature one, it is worth looking into the question why it was so widely accepted and what it is used for. The scroll-wheel provides an additional input stream without hindering normal mouse interaction. Predominantly, document scrolling is the domain of the scroll-wheel but other uses have been adopted as well, like picture and document zooming or scrolling through and selecting available options in input fields. Another noteworthy usage of the scroll-wheel is in first-person 3D action games where it is used to quickly swap between different tools of interaction with the environment. It allows the player to change tools while constantly moving at the same time, which is crucial for success in these games.

Besides computer mice, the use of tablets is well established as an input method for graphical art and CAD applications. A currently available input device that probably comes closest in form to the new multi-stream input pen for VE applications proposed in this work is the Wacom Intuos Airbrush-stylus (http://www.wacom-europe.com/uk/products/intuos/input_airbrush.asp). It is an optionally available stylus for Wacom's Intuos line of pressure sensitive tablets for pen based input and features a finger wheel, similar to a scroll-wheel, with 1024 levels of activation and a side switch. The purpose of the finger wheel on this stylus is to provide artists with the possibility to control ink-flow in graphical applications, as is the case with real airbrushes.

Because of the fact that menu and tool selection is essentially a 1 DOF operation and many previous menu techniques suffered from difficult-to-learn selection methods due to their 3 DOF nature, a scroll-wheel should be well suited to this task, as it is naturally constrained to 1 DOF and provides an input stream utilizable for navigation in a one-dimensional space. Moreover, the input is in fact discrete and single "notches" in the wheel provide passive-haptic feedback when switching from one position to the other, making selection from a set particularly easy. By studying the use of scroll-wheels on mice, we learned that these sets can be the parameter space of an (pseudo-)continuous value (e.g. document position, ink-flow, zoom-level) or a discrete set of options and tools. We want to use the scroll-wheel on the stylus to switch between the following elements according to our terminology of system control tasks:

- *Tools* and *modes* can be changed which results in tool and mode switches when the scroll-wheel is operated.
- Single *commands* and *parameters* can be selected that can be executed or modified thereupon. If a set only consists of commands and parameters, navigation between these corresponds to navigation in a conventional one-dimensional menu.
- The space for navigation with the scroll-wheel can also be the *parameter space* of a scalar parameter itself.

A single set of tools, commands and parameters to choose from would only be acceptable for a limited number of elements, thus the implementation of the user interface should offer some sort of hierarchy and organization of tools and menu items using the input of two buttons on the pen to move between levels in the hierarchy.

4. IMPLEMENTATION

4.1. User Interface

We use two different styles of visual feedback: Toolbars and cylindrical menus.

In toolbars, 3D icons and labels are used to represent tools and commands. These icons can also reflect the parameterization of the tool, e.g. line size. Captions of commands or parameters are represented by billboarded labels. Apart from that, labels are also used to clarify the meaning of icons and appear above them in case they are needed by showing the name of the tool. It is common to both styles that the item currently selected – tool or menu entry – is always visible in front of the tip of the tracked pen (see Figure 2 and Figure 3 for representation styles).

In a toolbar, the 3D icons and labels are positioned on an imaginary linear list aligned parallel with and running through the entire length of the pen (Figure 2). Since the scroll-wheel on the pen is also aligned this way, turning of the wheel naturally maps to pushing away and pulling near elements in this linear list. Since the list of icons and labels would interfere with the scene, partly be obstructed by the pen, and thus distract the user if entirely visible, only the entry

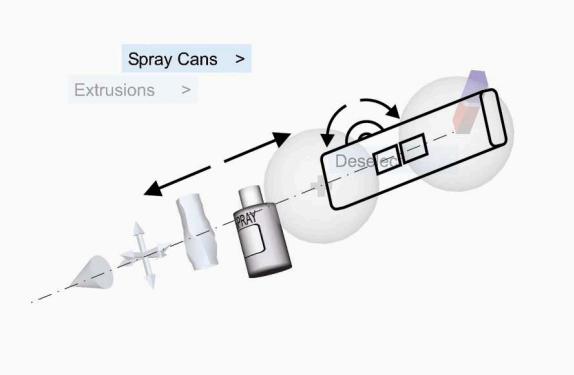

Figure 2: Toolbar representation of the user interface. Only the icon of the selected tool is actually displayed – semi-transparent icons in the illustration are added for the sake of clarity.

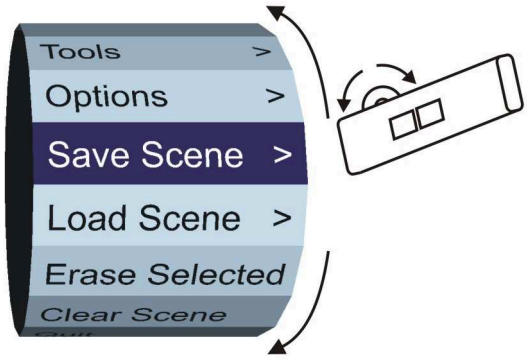

Figure 3: The graphical cylindrical menu representation of the user interface.

currently selected is shown in front of the tip of the pen. In order to maintain the sense of spatiality and to give the impression that the icons and labels are actually beaded on a line, icons and labels are sliding in and out the tip of the pen – fading in and out while they do so – in the direction of the rotation of the scroll-wheel. According to [Bed99a] this animated transition should also help users to better remember positions of tools in the toolbar. If the scroll-wheel is rotated by several notches in a short time and many entries in the list have to be skipped, the speed of the animation of the icons and labels sliding in or out at the tip of the pen is accelerated to allow for faster navigation [Hin02a].

A cylindrical menu is used for sets of application control tasks that are made up entirely of commands and parameters. In 2D desktop environments these are usually implemented as pull-down menus. As the turning of a scroll-wheel naturally maps to rolling of a horizontal cylinder, the use of a cylindrical rotational menu instead of a list on a plane is strongly suggested (Figure 3). The advantage of this representation over our toolbar style representation obviously is the visibility of several entries at once. This eases the process of identifying and targeting the desired item and significantly speeds up navigation. As the user turns the scroll-wheel the menu will rotate to bring the next or previous entry to the tip of the pen. As is the case with the animation of the toolbar fast scrolling of the wheel results in faster rotation of the menu. In addition to the rotation of the menu, the selected item will also get highlighted to be recognized at a glance. As the menu is shaped like a cylinder it always has the same size, regardless of the number of entries, which avoids cluttering of the workspace. As with labels, the cylindrical menu is view dependent and will always face the user. This ensures its readability, no matter how the pen is oriented.

As mentioned above, due to the large number of tools, application commands and parameters in even moderate complex applications, some sort of grouping and hierarchy is necessary. Two buttons on the pen – one being the main button every stylus features, the other being a supplementary button – are used to navigate between toolbars and menus and submenus of arbitrary depth, respectively. In order to conserve the reference of a submenu, its parent stays visible in its current state at one side of the submenu, much like in conventional 2D pull-down menus (Figure 4).

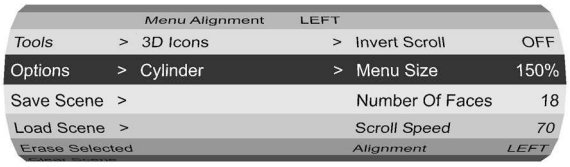

Figure 4: Example of a graphical cylindrical menu showing the root menu with two cascaded submenus. Menu items containing a submenu are indicated by the '>' character.

4.2. Control

The two buttons of the Wheelie are not only used for navigating the menu and tool hierarchy, but also for activation of items and direct manipulation:

- *Primary (activation) button*: If the item currently selected in the system control hierarchy is a single tool, the primary button is used to apply the tool while being pressed. If a command is selected, it is executed when the primary button is activated. If the selected item is a parameter, it can be adjusted by scrolling through the parameter-space with the scroll-wheel while the primary button is pressed. Should the entry currently selected contain a subset of tools or a submenu, activation of the primary button effectuates a

change to the subset and descends in the menu hierarchy.

- *Secondary (escape) button*: The supplementary button on the pen is used to ascend one level in the hierarchy structure.
- *Scroll-wheel*: The scroll-wheel is mainly used to browse through the current set of available tools, commands and parameters, or to select an item from the graphical menu. As mentioned above, in combination with the primary button it allows parameterization of the current tool or menu item. For instance, the diameter of the cross-section of an extrusion tool, transparency or the "ink-flow" of a spraying tool can be changed during direct manipulation activities.

4.3. Hardware Setup

In order to build a prototype Wheelie stylus we used a commercially available marker and the electronic parts and the scroll-wheel of an old mouse. As it turned out, the housing of a marker is very well suited for our purpose, since it provides enough space for the integration of a scroll-wheel in the narrow part of the grip as well as two buttons on the flat side of it. The scroll-wheel is installed in a position that, when the pen is held as if for writing, allows comfortable operation with the tip of the index finger. In the place where the thumb is resting against the flat part of the pen, two buttons are mounted in a row insofar as the thumb does not have to be moved to reach either one of them. This configuration resembles the handling of a mouse with two extra buttons on its side, used for document history navigation in internet browsers or the like. Since it is crucial for the user interface to deliver a smooth navigation experience, it is important that both buttons can be operated easily. Since push buttons are often awkward to activate if not pressed directly from above, we decided to use buttons that tilt forward (front button) and backwards (back button). This significantly enhances button control with the thumb. The button closer to the tip of the pen functions as the primary activation button whereas the rear button acts as the secondary escape button. Unlike the scroll-wheels on a mouse, the wheel on our prototype pen does not feature the additional function as a button. Although this would yield an extra degree of freedom, the force needed to activate it with the index finger is uncomfortable since the pen is held in hand and does not reside on a flat surface that provides resistance to this force in opposite direction. This is especially true since the force to activate a button click on the scroll-wheel must be higher than on ordinary buttons to avoid involuntary activation while operating the scroll-wheel.

The test environment consisted of a Barco Baron™ virtual table, an ART optical tracker system and a PC with a GeForce graphics card.

5. EVALUATION

To see how Wheelie performs in practice and to informally observe users while interacting in a complex application with it, we developed a test application allowing the user to select from a variety of tools, commands, and parameters. The test application, implemented in the Studierstube framework [Sch02a], resembles a desktop graphic application in which the user can spray objects, create extrusions from cross-sections and draw lines. Additionally, common editing tasks such as selecting, moving, scaling, painting and deleting of object parts are available as tools. The graphical menu offers commands to erase, save and load created scenes, as well as options and parameters to customize the user interface.

The concept of scrolling through a list of available tools with the pen was well received and the testers did not voice problems with the overall handling and

Figure 5: Spraying in the scene with a spray can shaped tool. The icon also reflects the aperture angle of the tool.

Figure 6: With the move tool objects not only can be moved but also scaled by using the scroll-wheel while the primary key is pressed.

navigation of the interface (Figure 5 and Figure 6).

The use of the scroll-wheel during the application of the tool, e.g. the change of the aperture angle of a spray can while spraying, was less obvious, but immediately adopted by the users when demonstrated. This, however, revealed another issue: Users often desired to adjust the parameter before actually applying the tool. While this does not pose a problem if the tool is only interacting with objects in the scene, it was a nuisance when the tool was actually adding new content. Besides, it is disputable which parameter of a tool should be subject to change in the course of a workflow. Another issue worth mentioning emerged from using our interface in combination with a virtual palette: Widgets on the palette must be operated with a simple pointing tool, having no function of its own. Since this would make recurrent switches between the pointer and actual tools necessary, it is solved by simply deactivating the function of a tool when interacting with a widget.

An interesting aspect was noted when we observed control of the graphical cylindrical menu system with the Wheelie: Although the cylindrical shape of the menu strongly suggests that its affordance is to be rotated using the scroll-wheel in the same direction, many users thought of it to work as moving the highlighted selected item with the scroll-wheel as opposed to turning the menu cylinder so that the desired entry gets aligned with the tip of the pen. We think this is mainly induced by the visual feedback that highlights the selected item and the notion of graphical menus to be static, gathered from experience during extensive use of 2D desktop menu systems. Tests with less experienced users should be made to investigate this theory. Nevertheless, we added the option of inverting the scroll-direction in the menu representation of the interface (Figure 7).

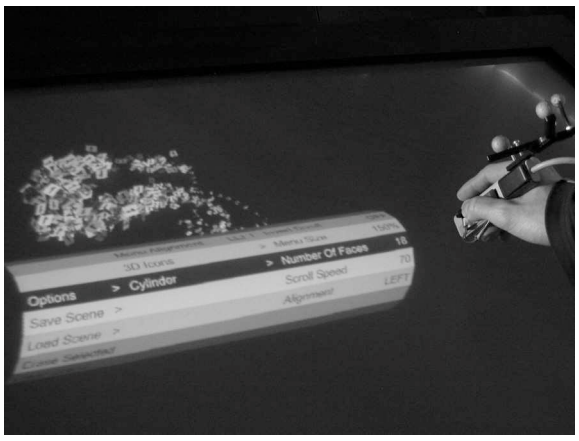

Figure 7: The graphical menu in action. Rotation direction of the menu when scrolling through entries can be set to user's preference.

Another point mentioned by our testers was that the smooth animation (i.e. rotation) of the cylindrical menu is dispensable. While the animation helps in the toolbar representation as only one item is visible at a time, the cylindrical menu provides an overview of surrounding entries and transitions need not be animated. On this account we implemented an "expert mode" (in both representations) that performs instantaneous jumps when operating the scroll-wheel. The default configuration was changed to smooth transition in the toolbar representation and instantaneous transition in the graphical cylindrical menu.

On the hardware side, a wireless device is preferable to our wired prototype of the Wheelie. Therefore future versions should only be made on a wireless (e.g. Bluetooth) basis to further ease the handling of the Wheelie.

6. CONCLUSION AND FUTURE WORK

Wheelie, a scroll-wheel enhanced tracked stylus for VEs presented in this paper, is an input-device that provides application and user interface designers with a separate one-dimensional discrete input stream applicable to various system control techniques for complex VE applications. The user interface and menu system proposed in this work allow the user to quickly change between different tools, execute commands and perform parameter manipulation with the tips of his fingers of just a single hand. It takes advantage of the fact that the input of the scroll-wheel is constrained to 1 DOF, which makes it perfectly tailored to the task of menu selection and tool switching, and certainly better than continuous input streams like a joystick for these purposes.

When compared to other 3D system control techniques, the Wheelie exhibits true general purpose properties:

Although more extensive experimental evaluation will have to be performed, switching between single different tools for direct manipulation is not as fast as ultimately possible with the ToolFinger approach which, on the other hand, is a special purpose technique limited to a single set of available tools. A perfect combination would be the use of entire ToolFingers as entries on the toolbar representation to choose from with the scroll-wheel.

The C³, designed as a quick-access menu, provides a faster way of activating commands but is also limited to a single set. The Wheelie menu system offers a familiar 2D-like representation of a hierarchical graphical menu, providing a virtually limitless space for commands while at the same time sustaining its size, not occluding much of the valuable visible

space. Other interaction techniques can still be used alongside Wheelie since Wheelie only extends the input capabilities of a stylus.

Quantitative analyses which compare Wheelie to other types of similar system control techniques will have to be made in order to get proper information on how Wheelie performs against them. Nevertheless, we can confidently assume that a scroll-wheel implements a useful addition to the conventional 3D stylus without cluttering the interface or complicating traditional stylus interaction.

7. REFERENCES

[Bed99a] Bederson B. B., Boltman A.: Does Animation Help Users Build Mental Maps of Spatial Information? In INFOVIS '99: Proceedings of the 1999 IEEE Symposium on Information Visualization (Washington, DC, USA, October 1999), IEEE Computer Society, pp. 28-35.

[Bow99a] Bowman D. A., Hodges L.: Formalizing the Design, Evaluation, and Application of Interaction Techniques for Immersive Virtual Environments. In The Journal of Visual Languages and Computing 10, 1 (1999), pp. 37-53.

[Bow01a] Bowman D. A., Wingrave C.A.: Design and Evaluation of Menu Systems for Immersive Virtual Environments. In VR ’01: Proceedings of the Virtual Reality 2001 Conference (VR’01) (Washington, DC, USA, 2001), IEEE Computer Society, p.149.

[Coq99a] Coquillart S., Wesche G.: The Virtual Palette and the Virtual Remote Control Panel: A Device and an Interaction Paradigm for the Responsive Workbench(TM). In VR ’99: Proceedings of the IEEE Virtual Reality (Washingron , DC, USA, 1999), IEEE Computer Society, p. 213.

[Cut97a] Cutler L. D., Fröhlich B., Hanrahan P.: Two-handed Direct Manipulation on the Responsive Workbench. In SI3D ’97: Proceedings of the 1997 symposium on Interactive 3D Graphics (New York, NY, USA, 1997), ACM Press, p.107 ff.

[Gro01a] Grosjean J., Coquillart S.: Command & Control Cube: A Shortcut Paradigm for Virtual Environments. In Immersive Projection Technology and Virtual Environments 2001 Proceedings (May 2001), pp. 1-12.

[Hin02a] Hinckley K., Cutrell E., Bathiche S., Muss T.: Quantitative Analysis of Scrolling Techniques. In CHI ’02: Proceedings of the SIGCHI conference on Human Factors in Computing Systems (New York, NY, USA, 2002), ACM Press, pp. 65-72.

[Kru00a] Kruifjj E.: System Control. In 3D User Interface Design: Fundamental Techniques, Therory, and Practice. SIGGRAPH 2000 Course Notes (July 2000), Bowman D. A., Kruijff E., LaViola Jr. J., Mine M., Poupyrev I. (eds.), pp. 147-165.

[Lia94a] Liang J., Green M.: JDCAD: A Highly Interactive 3D Modeling System. Computers and Graphics 18, 4 (July 1994), pp. 499-506.

[Lin99a] Lindeman R. W., Sibert J. L., Hahn J.K.: Hand-held Windows: Towards effective 2D Interaction in Immersive Environments. In VR (1999), pp. 205-212.

[McM97a] McMillan G. R., Eggleson R. G., Anderson T. R.: Nonconventional controls. In Handbook of Human Factors and Ergonomics 2nd ed., Salvendy G. (ed.) (New York, NY, USA, 1997), John Wiley & Sons, pp. 729-771.

[Min97a] Mine M. R., Brooks Jr. F. P., Sequin C. H.: Moving Objects in Space: Exploiting Proprioception in Virtual-Environment Interaction. In SIGGRAPH ’97: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (1997), ACM Press/Addison-Wesley Publishing Co., pp. 19-26.

[Min97b] Mine M. R.: Isaac: A Meta-CAD System for Virtual Environments. Computer-Aided Design 29, 8 (1997), pp. 547-554.

[Sch99a] Schmalstieg D., Encarnação L. M., Szalavári Z.: Using Transparent Props for Interaction with the Virtual Table. In SI3D ’99: Proceedings of the 1999 Symposium on Interactive 3D Graphics (New York, NY, USA, 1999), ACM Press, pp.147-153

[Sch02a] Schmalstieg D., Fuhrmann A.L., Hesina G., Szalavári Z., Encarnação L. M., Gervautz M., Purgathofer W.: The Studierstube Augmented Reality Project. PRESENCE: Teleoperators and Virtual Environments 11, 1 (February 2002).

[Ste03a] Stefani O., Hoffman H., Rauschenbach J.: Design of Interaction Devices for Optical Tracking in Immersive Environments. HCI International 2003 Conference Proceedings (2003), Lawrence Erlbaum Associates, Inc.

[Sza97a] Szalavári Z., Gervautz M.: The Personal Interaction Panel – A Two-handed Interface for Augmented Reality. In Proceedings of EUROGRAPHICS ’97 (September 1997), pp. 335-346.

[Wes03a] Wesche G.: The Toolfinger: Supporting Complex Direct Manipulation in Virtual Environments. In EGVE ’03: Proceedings of the Workshop on Virtual Environments 2003 (New York, NY, USA, 2003), ACM Press, pp. 39-45.