

Command & Control Cube : a Shortcut Paradigm for Virtual Environments

Jérôme Grosjean and Sabine Coquillart

INRIA
domaine de Voluceau,
78153, Le Chesnay Cedex
France

Jerome.Grosjean, Sabine.Coquillart@inria.fr

Abstract. New immersive virtual environments (VE) such as the Responsive WorkbenchTM provide users with a very attractive way of interacting with 3D computer-generated worlds. The feeling of immersion is one of the many advantages of such configurations. Being able to interact naturally with the virtual world is a very important part of this feeling. Programs developed for the virtual environments need powerful, intuitive and rapid application control interfaces. Previous work on this topic have focused on the development of menu systems for the VEs: converted 2D menus, hand-oriented menus and 3D widgets. To our knowledge, no 3D equivalent of the quick keyboard hotkey mechanism has ever been proposed for VEs. In this paper, we propose a 3D paradigm: the Command and Control Cube (CCC or C^3), inspired by marking menus. The C^3 aims to be a rapid and intuitive mechanism for issuing a set of commands to an application.

Keywords

System control, workbench, 3D interaction, interaction styles, graphical user interfaces, user-centered design, virtual reality, virtual environment

1 Introduction

Virtual environment configurations are changing our ways of interacting with 3D objects, navigating inside 3D worlds, and controlling applications. Some of these configurations: the head-mounted display, the CAVE[5], or the Responsive Workbench[10][9] (RWB) to name but a few, have a great potential to increase the benefits of computer generated 3D worlds. Among these configurations, the *workbench*¹ is one of the most attractive for direct manipulation. The configuration provides a virtual workspace where 3D objects can be manipulated directly with the hands. Though it can be considered to be an immersive (or

¹ “workbench” is used to denote the responsive workbench as well as other similar configurations with one or two screens.

semi-immersive) configuration, users don't lose contact with their physical environment, their own body or their fellow coworkers, which is often regarded as disorienting. Many applications already running on workstations, e.g., modeler and interactive visualization of scientific data, can greatly benefit from this configuration, thanks to the head-tracked stereo display and the direct interaction with the virtual scene, for example, moving cutting planes, grabbing and manipulating objects with the hand, direct modeling, deformation of 3D objects and moving light sources.

However, porting these applications to the workbench raises two problems: interaction techniques for manipulation and system control, i.e. changing modes, variable states, issuing commands. Both have to be entirely rethought for a 3D environment. This configuration differs widely from the classic workstation. The output device of the workbench is a head-tracked stereo display allowing users to see, turn around or manipulate 3D objects due to the virtual world being superimposed onto the physical one. In contrast, for a workstation with a monitor, visualization and manipulation space are separated. The input devices are very different too. On workstations, users communicate with applications through multi-purpose interaction devices: a keyboard and a mouse. The number of available input signals is very high (102 keys on a PC keyboard). The common input devices available on workbenches are in comparison often designed for spatial manipulation only (light and wearable) and provide very few buttons, e.g. only one or two for a tracked stylus.

A now-standard application control interface has become popular and widely used for workstations: the WIMP interface (Windows, Icons, Menus and Pointing), based on the keyboard and mouse input devices. This choice of human computer-interaction is commonly accepted as the de-facto standard for these configurations. To provide lighter and faster access to the most-often used commands, shortcuts have been introduced as well. The keyboard is a good tool to issue these commands with a simple hotkey mechanism (like CTRL-S for save). Indeed the quickest way to enter a command for one's application in 2D is the hotkey mechanism associated with the keyboard.

To our knowledge no solution has ever been proposed to provide a user in 3D VE with a similar shortcut system to very rapidly issue a small set of commands to the application. In this paper we focus our work on this aspect and propose a new application control interface: the Command & Control Cube (later called in this paper: CCC or C^3). This interface has been developed for a holobench[18], a two-screen responsive workbench (see Figure 1 and Appendix). The C^3 is controlled by the user with a 6DOF tracked button. With the C^3 a user can send up to 26 different commands to the application. A two-speed mechanism similar to the marking menus[13][14][12] in 2D allows the C^3 to work in a novice mode with a graphical feedback or in an advanced mode for quicker "eyes-off" selections. In this paper we first present related work on menus and application control solutions for 3D VEs. Next, we describe the C^3 and its functions. The last section includes the conclusion and a discussion on future work.



Fig. 1. The C^3 on the Responsive Workbench

2 Related work

Application control inside virtual environments is a new field of research. The first applications developed for these environments faced the immediate need to provide an interface to control functions or states of variables inside the 3D world. At the SIGGRAPH'2000 course on 3D user interface design[11], Ernst Kruijff proposed a categorisation for the current system control techniques influenced by the description of non-conventional control techniques by MacMillan et al.[19]. He divided the approaches into graphical menus, voice commands, gestural interaction and tools.

Graphical menus are the 3D equivalent of 2D menus. The following criteria should be considered when designing graphical menus: placement, selection, representation and structure.

Feiner et al.[8] is a primary source for placement issues. However Kruijff proposed a finer division of the different placement solutions. Menus can be freely placed inside the virtual world (world-referenced), connected to a virtual object (object-referenced), linked to a part of the user's body such as the head (head-referenced) or the rest of the body (body-referenced) or placed in reference to a physical object such as the borders of the workbench (device-centered).

The selection of menu items in a 3D environment is quite different from that in 2D. Selecting an item inside a set of available functions is conceptually a 1D choice system. The addition of another unnecessary dimension is in this case an added difficulty. Moving inside a plane to make a 1D choice is quite easy with a mouse despite the second dimension. It becomes more complicated when the user has in addition to control the depth of his movements. The representation of the menu (form, size, space and affordance) is another whole topic. Finally, as the number of items grows, it becomes necessary to set a mechanism to diminish the cognitive load placed on the user: hierarchical subdivision or context-sensitive menus.

In this group of graphical menus, three different categories have been distinguished by Kruijff: converted 2D-menus, hand-oriented menus and 3D widgets.

2.1 Converted 2D Menus

An initial natural approach for designers of 3D interfaces has consisted of porting 2D interfaces to the 3D world. In the 2D world (screen, keyboard and mouse) the WIMP interface is now a commonly accepted choice to control applications. Since this interface is popular and quite efficient for 2D, porting the interface has the advantage of easily providing users with a non-disturbing familiar interface. However as stated earlier, the third dimension hampers the ease of selection [1][6][20]. Using a virtual ray controlled by the hand as a pointer tool is one method to alleviate this difficulty [21]. However it is still not adapted to making quick selections (like the hotkeys of a keyboard) due to the indirect manipulation and the precision required. Another approach, based on the virtual tricorder device [23], suggests using “2D anchored menus”. The 2D menu is displayed at the position of the device held in the hand. Selection inside the menu is done by pressing the mouse buttons.

2.2 Hand-oriented Menus

The two categories of hand-oriented menus are 1DOF menus, where the items are put on a circular object [15][22] and menus stored at a body-relative position, where selection of items is dependent on the hand relative position. This approach exploits proprioceptive feedback [16]². A hybrid approach would materialize the plane of selection by holding a flat physical transparent palette in the non-dominant hand ([4][20]), and selecting items with the dominant one. The graphical menu is displayed directly onto the transparent physical surface (using a tracker on the palette), taking advantage of both the hand-held prop and body-centered aspects.

2.3 3D widgets

Conner et al. [3] describes widgets as the combination of geometry and behaviour. This section describes 3D widgets, including all menus whose functions can be mapped onto an object.

The C^3 , in its novice working mode, is a graphical menu, hand-oriented, hand-held widget. The advanced user mode with “eyes-off” manipulation makes extensive use of proprioceptive feedback. The C^3 has been developed with the idea of exploiting the third dimension as an advantage and not a new undesirable dimension. It is based on 2D marking menus. Pie menus are in fact significantly better than linear menus when it comes to fast selection [2].

² Proprioception is “a person’s sense of the position and orientation of his body and limbs”. It provides strong advantages for direct manipulation (excellent control of one’s own hand), physical mnemonics (finding body-centered objects) and gestural actions (recall of actions).

3 Description of the C^3

The C^3 proposes an equivalent of the keyboard hotkey mechanism for VE. It can also be used as a quick 3D graphical menu. Menu items are arranged into a cubic configuration, rather than the traditional 2D pull-down menus. The C^3 has roughly the appearance of a cubic volume (or *cubic structure*), divided into $3 \times 3 \times 3$ smaller cubes (see Figure 2). Smaller cubes have the same size and are separated from each other by a fixed distance. They are associated with menu items and called “slots”. A ball can be moved inside the bounding cube of the structure. The center of this ball acts as a pointer for the selection.

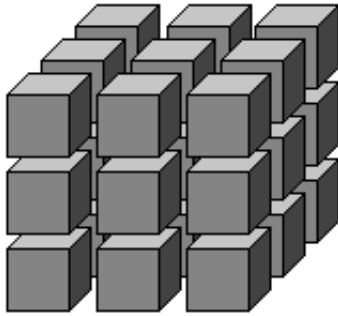


Fig. 2. Division of the space into 27 cubes

The C^3 is controlled by a 6DOF tracked button. The Pinch glove system developed by Fakespace[7] or the ring mouse[17] are valid devices. In essence, any device with a button attached to a tracker fits the requirements. Actually, in our experiments, we did not use a commercial system but instead built one from a simple mouse and a tracker (see Figure 3 and Appendix). The mouse buttons serve as “*pinch buttons*” placed on three fingers of the hand (although only one button was used). They can be pressed with the thumb. The tracker is attached to the user’s wrist.

Whenever the user pinches, the C^3 is displayed at small horizontal distance from his hand position. It stays visible while the button is pressed and disappears when released. While the pinch button is pressed, all movements of the hand are mapped onto the ball. The ball’s starting position is the center of the bounding cube, which is the center of one of the 27 slots, the middle one. By moving the hand, the user moves the ball inside the bounding cube. The ball cannot leave the cubic structure. If a user’s hand movement implies that the ball should go outside the volume, then the corresponding movement of the ball is truncated so that it stops at the surface of the bounding box.

The ball can thus be moved freely inside the cubic structure and placed inside any of the 27 slots. Releasing the button when the ball lies inside a slot results



Fig. 3. The tracked buttons

in a user's selection of a menu item, the one associated with the slot. Note that the "middle slot" shouldn't have any function associated with it, to allow the user to pinch accidentally without any consequences. Furthermore, it gives the user a way to cancel the C^3 without selecting an item. There is no restriction on the user's hand movements inside the cubic volume while the pinch button is pushed down, nor is there any time limit to select an item.

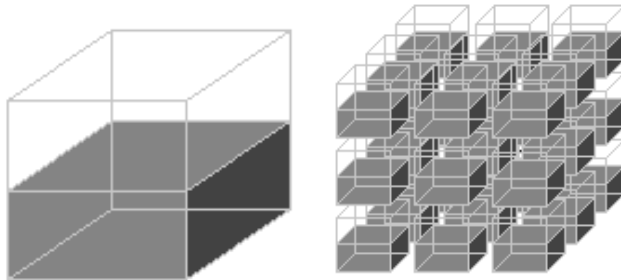


Fig. 4. Visual feedback inside a slot

Each slot is visually represented by a half cube (see Figure 4). The half cubes are transparent blue, while the ball is yellow with no transparency. Since the half cube occupies the lower part of the slot, it's possible to put a transparent texture on top of each that shows an icon, as in classic 2D menus, explaining the function of the slot. The sensitive zone for the selection is still the whole cubic volume of the slot, but the visual part of the slot lies only in the lower half. The opaque ball can be moved anywhere inside the slot to trigger the menu item and is always

visible, thanks to the transparency of the slot. The center of the ball is the true pointer of the C^3 . To further help the selection, the currently selected slot is highlighted.

As was previously mentioned, pinching two fingers triggers the display of the C^3 . Only one horizontal plane of slots, i.e. 9 slots, are represented simultaneously. The visible plane of slots is always the one containing the pointer: the yellow ball controlled by the user's hand. Moving the hand in space moves the ball from slot to slot and triggers, if necessary, the display of another plane (see Figure 5 and Appendix). A short delay to the display of the C^3 allows "eyes-off" manipulation.

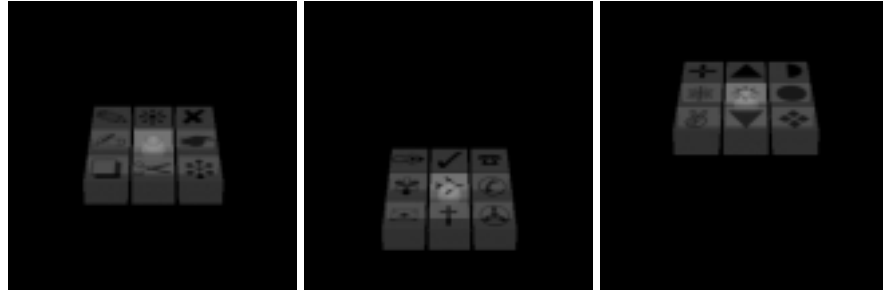


Fig. 5. The C^3 appearance when the user pinches and stands still, moves his hand down, or up.

4 Design issues

This section discusses the reasons behind the most important design choices of the C^3 .

4.1 Shape

We decided to use a cubic structure to divide the space around the hand. At first the C^3 was influenced by 2D marking menus, which divided the screen around the mouse pointer into equal pie-shaped sections. We could have divided the 3D space equally around the 3D pointer (the yellow opaque ball). With a simple space decomposition into 6 parts (up, down, left, right, front, back) it would have worked well. However six is a small number of commands for a shortcut mechanism, although it could be adapted to a menu system with a hierarchical structure and was judged too limited. An equal division of the space in more than 6 directions becomes very complicated for a user to imagine. Without any mental representation of the spatial division, it's rather difficult for a user to remember and execute the correct movement with his hand for each menu item.

Thus we decided to use a simple metaphor: the cube. By dividing the space around the ball into 3x3x3 cubes we give the user an intuitive way to remember the path to each command. For example: the upper front right command, or the lower left back one. Each of the 26 available commands can be accessed by mentally figuring the cubic form of the menu and/or by thinking in terms of a combination of simple directions (up, down, left, right, front, back, middle). The task of selecting an item is facilitated by three mechanisms:

- The cubic structure (mental representation of the space)
- The proprioceptive sense (all movements start from the middle position so selecting an item equals moving the hand in one direction).
- The directions are aligned with the screen borders of the workbench (device-centered graphical menus).

4.2 Number of commands

The cube design dictates that the 27 slots are associated with the commands of an application. Since the “middle slot” is reserved for a cancel action there remain 26 options to be configured with the functions of one’s application. This number is extended if all the fingers of the human hand are used to trigger different cubes (i.e. set of functions). The Pinch Glove allows 4x26 different functions to be selected. With regard to this large number, we believe the C^3 could be used as much as a shortcut mechanism than a rough 3D menu.

4.3 Visibility

Many visualization concerns have been taken into account in the design of the C^3 . First, the C^3 acts as a pop-up menu, and is displayed only when a user performs a specific action: pinching two fingers. Therefore there is no obstruction of the view by 3D widgets in idle mode. The pointer is opaque and remains visible as long as the C^3 is activated. Thus all the slots are transparent and in different colors so that the ball cursor can move inside each and stay visible. The small size of the pointer allows the icon of the selected slot to be seen when the opaque ball is inside the slot (see Figure 5 and Appendix). The transparency of the C^3 does not obscure the 3D scene the user is working on. Indeed, it is still visible through the transparent slots of the C^3 .

All the slots are partially “filled” in blue, although the sensitive selection zone is the whole volume. The slots are only filled in their lower half part so that icons can be placed on top of the blue transparent blocks to help the user know which command is associated to each slot. Since the user’s hand, at the height the cube is displayed, is lower than the head, there is a good line of sight on the icons. The ball is also more visible with half-filled slots.

The C^3 is displayed in front of the user’s hand, lower than the user’s eyes. However since the multiple planes of 3x3 slots are lying one above the other, it is rather difficult to see the icons of the lower planes. To avoid this problem, only one horizontal plane (9 slots) is displayed when the C^3 is active: the plane

where the ball is currently lying. Moving the ball with the hand from slot to slot triggers the display of another plane, if necessary, but there is always only one visible plane at a given time. Horizontal planes have been chosen in preference to other plane orientations, because the user's eyes are generally above the cube and because of the table-like shape of the RWB.

The size of the little cubes is roughly 10x10x10cm. Some experiments with different sizes guided this choice. The size should be adapted for each workbench with regard to the screen size and resolution.

4.4 Learning and quickness

With the help of the icons, a novice user is able to invoke the C^3 by pinching two fingers and using the cube metaphor to have quick access to 26 commands. For each command the icons provide the necessary feedback to let the user make his choice. Selecting a menu item is simple: pinching, moving the hand in one direction and releasing the pinch. As stated earlier, choosing between distinct 3D directions is easier than selecting an item inside a linear list. Movements can also be very quick and precise since they are relative to the starting position of the hand. No tracker calibration interferes in the selection process.

The C^3 approach, similar to the 2D marking menus can be improved by adding a short delay (less than a second) before displaying the menu. With this feature, the C^3 enhances the application control experience for well-trained users. Indeed, when the positions of the functions inside the cube are perfectly known it is very intuitive to blindly make a move with the hand in a specific direction to trigger the appropriate function. One advantage of this method is that the interface doesn't change between the novice level and the advanced user level. Furthermore, there is no "switch" to be activated to change from one mode to another. A novice user who begins to remember the position of his favorite functions will naturally begin to move his hand blindly. And at all times, an advanced user suddenly hesitating about the right direction to find an option would automatically trigger the display of the menu after the short delay.

In comparison to the hotkey mechanism of the keyboard, it can be argued that chain-triggering the same command with the C^3 seems slightly slower at first sight. However when the same command is not always repeated several times, results should be similar. More testing needs to be done on this issue.

4.5 Focus of attention

The non-dominant hand is used to control the shortcut system, thus leaving the dominant hand free. It is important to leave the user's dominant hand out of the shortcut process, since the main task in a VR application is usually performed by the dominant hand. We don't want to force the user to switch tools to do a simple and quick menu item selection. In addition, the *visual* attention of an advanced user is freed because he can use the C^3 blindly, thus focusing more on his main task. Actually, most of the classic menus require a user to look at a 3D widget, forcing him to break his attention.

5 Conclusion & future work

The C^3 system was initially developed with the idea of finding an equivalent solution to keyboard hotkeys for a RWB. Although in the past some solutions have been proposed to provide users with menu interfaces, to our knowledge researchers had never before focused on a shortcut approach for VE. The third dimension is used by the C^3 as an advantage to propose more menu options to the user. This addition is never an inconvenience for the user as in previous attempts to convert 2D menus into 3D menus. The C^3 has been thought of as and is indeed a real 3D menu. Since the C^3 is controlled with the non-dominant hand, the user is not distracted from the main task, usually performed or led by the dominant one. Controlling the C^3 directly with the hand is a way to use proprioceptive feedback for a better manipulation and recall of actions. The movements of the hand can be quick and precise, because they are relative to the starting position of the hand. No tracking calibration or precision problems could interfere in the process of selection. In addition, the pop-up approach of the C^3 preserves the menu to block the view when idle.

Based on the marking menus, the C^3 provides the advantage of offering a single interface for both novice and advanced users. As the user memorizes the commands associated to each direction, he moves smoothly from the novice level where the menu is displayed, to the advanced level where he can make an “eyes-off” manipulation to select a command. Furthermore, remembering a direction is easier than remembering a position in a linear menu. An open-ended gesture is also much faster than a Fitt’s law task, as in a standard linear menu. The ability to select a command blindly allows the user to rapidly control the application commands, without disturbing his attention since he doesn’t have to look at an interface graphical component.

The C^3 can be extended to $nx26$ configurable menu items, n being the number of available buttons. This is a large number of options for a shortcut system. We found that the C^3 can then be used as a menu in itself, if the number of menu items needed by an application falls below the $nx26$ limit.

We are planning to add a hierarchical structure to the cube metaphor to extend the number of menu items available and to allow a better placement and recall of the application commands inside the C^3 . Some discussion of the placement of the C^3 with regard to the user’s hand and body and its potential visual occlusion by 3D application data is also still necessary.

The C^3 has been used in two different applications on the RWB: the visualization of a car cockpit with some basic exploration tools and a more complex visualization application for fractal models. Initial impressions of users seem positive. However we need to test the performance of the C^3 in terms of quickness of selection and accuracy for each slot, usability and to compare its efficiency on the workbench with that of the hotkeys for the keyboard. These data could help in the placement of the more frequently-accessed options inside the C^3 . The C^3 has been developed on a holobench but could be extended to other VE configurations. The orientation of the cube relative to the frames of the screens should be reexamined. Some other similar considerations such as the choice of

displaying horizontal planes inside the C^3 could also change, vertical planes are better for a wall display or a HMD.

References

1. J. Butterworth, A. Davidson, S. Hench, and T. M. Olano. 3dm: A three dimensional modeler using a head-mounted display. In *Proc. 1992 Symposium on Interactive 3D Graphics*, pages 135–138, 1992.
2. Jack Callahan, Don Hopkins, Mark Weiser, and Ben Shneiderman. An empirical comparison of pie vs. linear menus. In *Proceedings of ACM CHI'88 Conference on Human Factors in Computing Systems*, Menus, pages 95–100, 1988.
3. D. Conner, S. Snibbe, K. Herndon, D. Robbins, R. Zeleznik, and A. van Dam. Three-dimensional widgets. In *Computer Graphics (Proc. 1992 Symposium on Interactive 3D Graphics)*, pages 183–188, 230–231, 1992.
4. S. Coquillart and G. Wesche. The virtual palette and the virtual remote control panel: A device and an interaction paradigm for projection-based virtual environments. In *IEEE VR'99*, 1999.
5. Carolina Cruz-Neira, Daniel J. Sandin, and Thomas A. DeFanti. Surround-screen projection-based virtual reality: The design and implementation of the CAVE. In James T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 135–142, August 1993.
6. Michael F. Deering. HoloSketch: a virtual reality sketching/animation tool. *ACM Transactions on Computer-Human Interaction*, 2(3):220–238, September 1995.
7. Fakespace. <http://www.fakespacelabs.com/>.
8. S. Feiner, B. Macintyre, and D. Seligmann. Knowledge-based augmented reality. *Communications of the ACM*, 36(7):53–61, 1993.
9. B. Frölich, B. Kirsch, W. Krüger, and G. Wesche. Further development of responsive workbench. In M. Göbel, editor, *Virtual Environments '95*, Eurographics, pages 237–246. Springer-Verlag Wien New York, 1995.
10. Wolfgang Kruger, Christian-A. Bohn, Bernd Frohlich, Heinrich Schuth, Wolfgang Strauss, and Gerold Wesche. The responsive workbench: A virtual work environment. *Computer*, 28(7):42–48, July 1995.
11. Ernst Kruijff. 3D user interface design: Fundamental techniques, theory, and practice. SIGGRAPH 2000 Course.
12. Gordon Kurtenbach, Thomas P. Moran, and William Buxton. Contextual animation of gestural commands. In *Proceedings of Graphics Interface '94*, pages 83–90, Banff, Alberta, Canada, May 1994. Canadian Information Processing Society.
13. Gordon Kurtenbach and William Buxton. User learning and performance with marking menus. In Beth Adelson, Susan Dumais, and Judith Olson, editors, *Proceedings of the Conference on Human Factors in Computing Systems*, pages 258–264, New York, NY, USA, April 1994. ACM Press.
14. Gordon Kurtenbach, George W. Fitzmaurice, Russell N. Owen, and Thomas Baudel. The hotbox: Efficient access to a large number of menu-items. In Marian G. Williams, Mark W. Altom, Kate Ehrlich, and William Newman, editors, *Proceedings of the Conference on Human Factors in Computing Systems (CHI-99)*, pages 231–237, New York, May 15–20 1999. ACM Press.
15. Jiandong Liang and Mark Green. JDCAD: A highly interactive 3D modeling system. *Computers and Graphics*, 18(4):499–506, July–August 1994.

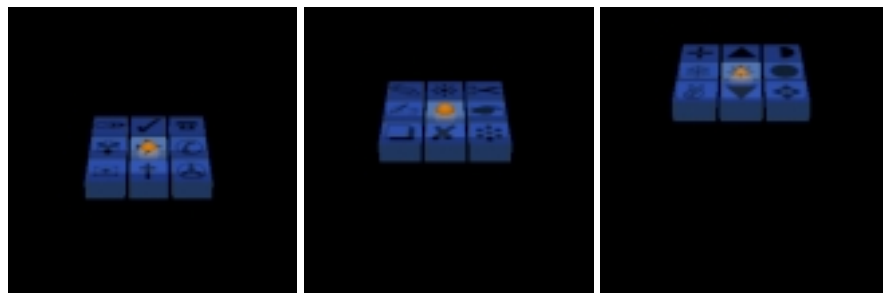
16. Mark R. Mine, Frederick P. Brooks, Jr., and Carlo H. Séquin. Moving objects in space: Exploiting proprioception in virtual-environment interaction. In Turner Whitted, editor, *SIGGRAPH 97 Conference Proceedings*, Annual Conference Series, pages 19–26. ACM SIGGRAPH, Addison Wesley, August 1997. ISBN 0-89791-896-7.
17. Ring Mouse. <http://www.worklink.net/ringmouse.html>.
18. TAN Projektionstechnologie. <http://www.tan.de/>.
19. McMillan G. Eggelston R. and Anderson T. Nonconventional controls. In *Handbook of human factors and ergonomics*, pages 729–771, 1997.
20. J. Sibert R. Lindeman and J. Hahn. Hand-held windows: Towards effective 2d interaction in immersive virtual environments. In *IEEE VR'99*, 1999.
21. Ron van Teylingen, William Ribarsky, and Charles van der Mast. Virtual data visualizer. *IEEE Transactions on Visualization and Computer Graphics*, 3(1):65–74, January – March 1997. ISSN 1077-2626.
22. G. Wesche and M. Droske. Conceptual free-form styling on the responsive workbench. In *VRST 2000*, 2000.
23. Matthias M. Wloka and Eliot Greenfield. The virtual tricorder: A uniform interface for virtual reality. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, Virtual and Augmented Realities, pages 39–40, 1995. TechNote.



The C^3 on the Responsive Workbench (Jérôme Grosjean et al., Fig. 1)



The tracked buttons (Jérôme Grosjean et al., Fig. 3)



The C^3 appearance when the user pinches and stands still, moves his hand down, or up. (Jérôme Grosjean et al., Fig. 5)