

Interactive Menus in Augmented Reality Environments

Frederik Brudy

Abstract— Through augmented reality (AR) applications a user can experience and interact with his enhanced surrounding: The user's real environment is combined with virtual objects. To control such a system or interact with his environment one needs certain system controls, often being found in menus. Interfaces for system control tasks have been studied well for two-dimensional menus in conventional desktop environments. For augmented reality applications one more dimension has to be considered. To issue a system control command a user might not want to leave the augmented reality system, instead these menus have to be available to him inside the augmentation and adapted to it.

In this paper menus suitable for AR systems are described. They are categorized in three sections: menus from WIMP environments, enhanced WIMP menus adapted to AR and menus purely developed for AR. Some design considerations are given which have shown to be crucial in developing a menu for augmented reality applications, such as the placement of a menu on screen, the maximum number of menu items and the need for visual, auditory or tactile feedback.

Index Terms—Augmented Reality, Interactive Menus, System Control

1 INTRODUCTION

In an virtual environment system the real world is replaced by a virtual one, in which a user cannot see the world around him. By contrast, in an augmented reality (AR) system a user can see the world surrounding him superimposed with virtual objects. The presentation of the augmented reality to a user can, for example, happen through a head mounted display, a mobile phone or tablet, a projector or by other means. AR is not limited to the sense of sight, as it potentially affects other senses as well [22].

Azuma surveyed the field of AR thoroughly. He defined augmented reality environments "as systems that have the following three characteristics:

- Combines real and virtual
- Interactive in real time
- Registered in 3-D" [1].

Therefore augmented reality can be summarized as follows: It has three dimensions, runs in real time and is interactive to the user. It provides local virtuality, which means that virtual objects are added at or around the user's position to the real world [22].

Interaction in many virtual environments is characterized by Bowman and Hodges [3]:

- **Navigation:** this describes the task of moving through a virtual environment. In augmented reality systems the user moves through the real world, therefore a virtually augmented map might be an example for this task.
- **Selection:** is the task of choosing a virtual object from the environment or from a list of objects.
- **Manipulation:** selected objects can be manipulated, for example rotated or scaled.
- **System control:** this task refers to changing a systems state or the mode of interaction [23].

System control is an integral part of conventional 2D desktop interfaces and takes many forms, such as in a pop-up, pull-down, palette-based, pie and various other menus [10]. The result of a command in

a computer systems is always a selection of an element out of a set of command items. Selection can be made by voice input, gestural interaction, tools and graphical menus, or through a mixture of these [23].

Simply transferring a two-dimensional task to 3D space might be insufficient since the constraints of the physical desk are missing and for example touching a menu item floating in mid-air is much more complicated than selecting it with a mouse in 2D. Not only the environment needs to be augmented, but also menus need to evolve from their conventional 2D desktop techniques to suitable menus in 3D space. They have to be *inside* the augmentation and preferably be augmented as well. For the feeling of good immersion it is important that the user does not need to leave augmented reality in order to issue a control command [36].

Lots of research is being made on the topic of the augmentation of interaction in virtual and augmented environments. Some of these interaction techniques are an integral part of menu interaction and they will be covered as far as they are relevant for the menus described.

At the beginning of the 21st century very little research has been done on ways to change the system's state and the mode of interaction, which is called the *system control task* [4]. Menus are one kind of interface for the task of system control. Other examples are direct manipulation of objects or command line interfaces. In the past years more research has shown improvements for system control and especially menus in AR.

This paper aims to list the research made on menus in AR so far and is organized as follows: In the next section several menus, created for or related to AR, will be described in detail. After that several design principles are given which should be considered when creating menus for AR. A short summary concludes this paper in section 4.

2 MENUS

In this section menus, that have been proposed for augmented reality environments, will be described in detail. The menu systems will be grouped in three categories. Some of the menus might fit into more than one or all of these categories, since they are not clearly confined. For clarity they are only listed in one of them, the one that fits most.

Categorization of the described menus will be done as follows:

- **2D menu in 3D environments:** menus from two-dimensional environments in augmented reality. These are the menus from the conventional WIMP desktop environment. Some of them work in AR without much adaption.
- **Enhanced 2D menus:** enhanced two-dimensional menus, adapted to AR.
- **Augmented reality specific menus:** menus which have been explicitly developed for AR, or other virtual environments and are suitable for AR.

• Frederik Brudy is studying Media Informatics at the University of Munich, Germany, E-mail: brudy@cip.ifi.lmu.de
• This research paper was written for the Media Informatics Advanced Seminar 'Beyond the Desktop', 2012/2013

Menus from 2D environments will be listed since several of the menus proposed for AR, originate in two-dimensional systems. Hence they bridge to a two-dimensional graphics system, known by most users nowadays. Some of the menus from 2D systems have been adjusted to fit better in virtual or augmented reality environments. They still might work in their earlier environments but are adapted to a third dimension.

2.1 2D menus in 3D environments

These menus are directly transferred from 2D desktop graphical user interfaces. Though some limitations apply on those WIMP elements, as the user might not have a mouse or keyboard. Some well known examples are the pull-down menu in which the trigger mechanism might be of a proprioceptional nature (pull-down from above ones head). Point-at menus are also part of this category.

2.1.1 Floating and Pull-Down Menu

The conventional pull-down menu from two-dimensional desktop environments is easily transferred to AR [19]. These menus appear upon a gesture by the user. The interaction might be of a proprioceptional nature. This means that a user can trigger such menus by moving his hand to specially assigned body parts, for example by pulling them down from above his head. More about proprioceptional menus can be found in section 2.2.1.

2.1.2 Circular Menus

Circular menus have been introduced by Callahan et al. In circular (or pie) menus, the items are placed on an invisible circle around the mouse cursor's position. Items are selected with the mouse cursor. With increasing item count pie menus become polynomially larger. Despite this drawback, with a small number of items these menus reduce target seek time and lower error rates in selection, compared to linear menus [7].

HoloSketch: Deering adapted the two-dimensional circular menu to virtual reality in his *HoloSketch* [12]: By pushing down a button on a pointing device, the *wand*, a circular menu fades up, centered around the wand tip. From then on, its position is fixed in the initial position. The user selects an item by pointing to it with the wand and releasing the pressed button. This menu supports hierarchies: when a sub-menu is to be shown, the main menu moves back into the screen and a sub-menu fades up around the wand tip, meaning around the currently selected menu item.

3D Ring Menu, objects representing menu items: Liang introduced a three-dimensional ring menu in the *JDCAD* 3D modeling application [27]. It was designed for a hand-held 3D input device. 3D objects are distributed on a ring as shown in figure 1. Each object represents one available menu item. A gap in the ring, facing the user, is used as a selection spot. The items move along the circular ring with the rotation of the input device.

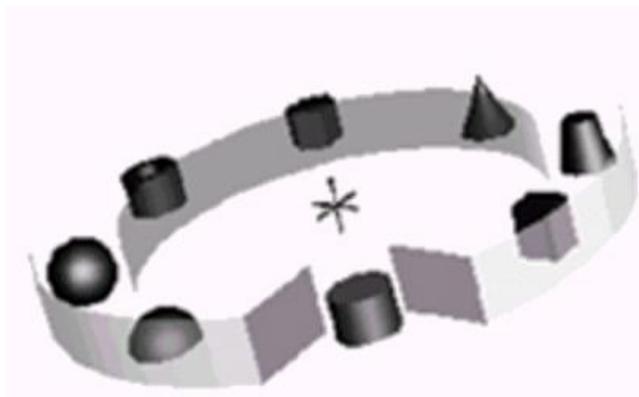


Fig. 1. The ring menu used in *JDCAD* [27].

3D Ring Menu, rotated by user's wrist rotation: Gerber et al. [13] found the above mentioned ring menu not to scale very well to a large number of menu items, or which may not be easily represented by a 3D graphics icon. Still they found the concept of a ring menu in 3D space interesting and built upon it. They used 160 degrees of an unclosed ring with a fixed radius on which equally shaped boxes are distributed. Because of these constraints the number of items on one ring is limited to 9 to 11 items, depending on the size of the boxes. The ring shows up with the push of a button. When the user turns his wrist, the movement is mapped to the items on the ring. The selected item lies in the back of the ring, surrounded by an orange selection box with a text label, describing the item.

3D Ring Menu, rotated by user's wrist rotation, different sub-menu designs Gerber and Bechmann [14] expanded the previously mentioned ring menu concept with means for sub-menus. They evaluated three different layouts in a user-study: stacked, concentric and crossed.

- **Stacked layout:** The different levels of the menu are stacked vertically, with the level, currently being manipulated, on top of the stack. The selection path can be read from bottom to top. This layout was found to be the fastest and most accurate in terms of selection quality. Figure 2 shows a user manipulating the ring menu, with a sub-level opened.
- **Concentric layout:** The first level starts on an inner circle. Upon the selection of each sub-level another circle wraps around the currently selected item. The selection path can be read by looking at the items which follow in one line behind the active one.
- **Crossed layout:** Upon selection of a sub-menu, the level is displayed rectangular to the first row of menu items, starting at the active item.

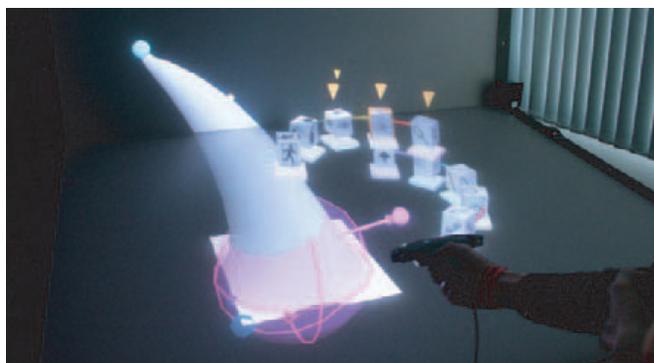


Fig. 2. The ring menu with a sub-menu in stacked layout [13].

The Rotary Tool Chooser Mine combines a one-dimensional ring menu with a two-dimensional menu known from conventional WIMP interfaces [29]. One dimension means, that the user only has one degree of freedom to move between selections. This limitation makes selection in 3D space easier. This ring-menu, the *Rotary Tool Chooser*, was introduced in the *ISAAC* project to quickly select frequently used tools and commands. By pushing a button menu items are shown around the user's hand; rotating the hand around a chosen axis causes the tools to slide across the arc. Selection is made when the item is in a selection box and the button is released.

Sundial Menus Another technique are sundial menus, described by Shaw et al. [35]: a circle is divided into equally sized pie-shaped sections. The shadow stick starts at the center of the menu, or in a hierarchical menu in the center of the parent node. The user rotates the stick around its center to select a pie section and thus the menu item.

Marking Menus Marking menus have been introduced by Kurtenbach et al. [25]. They are one kind of a ring menu in which the user makes selections of items by drawing a line between the center of the circle and his desired item. A novice user waits for the menu to pop up and makes his selection. An expert user knows his selection path and draws it immediately, without waiting for the graphical output. The support for sub-menus is given. Kurtenbach et al. evaluated their marking menus in a later work [24]: The number of maximum menu items on one ring depends on the depth a user might need to go. They claim that there should not be more than eight items on one menu ring, with a maximum depth of two levels. With increasing menu depth the items on each level need to decrease. For example error rates for menus with four items and depth of four are the same as for menus with eight items and two levels.

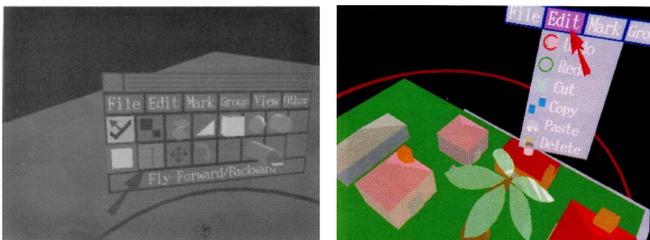
2.2 Enhanced 2D menus

Enhanced two-dimensional menus mainly draw their graphical representation from conventional 2D menus. The interaction with the menu is enhanced in order to make it more natural in an augmented reality environment and thus these menus cannot exist in pure 2D applications.

2.2.1 Proprioceptional Menu

Proprioception means the sense of the relative position and orientation of ones body parts [30]. In menus with body-relative interaction, a user attaches menus to his body. He always carries those objects with him and knows their whereabouts relative to his body, similar to the tools on a tool-belt. The objects can be of virtual or physical nature. Although being developed for 3D desktop applications, *Toolspaces* [33] could also be suitable for augmented reality: storage spaces attached to a user's virtual body, in which, for example, menu items or system control tasks can be stored.

Butterworth et al. describe *3dm* [6] which is a surface modeling program for model manipulation and understanding. In their user interface they provide a flying menu which is called the *toolbox*. The *toolbox* initially appears near the user's waist, from which he can move it to a more convenient location. It stays attached to the user when he moves around in the physical space. He can also detach it and leave it in the world surrounding. Very much similar to a tool-belt. The *toolbox* is divided into a rectangular grid, with each cell being a menu item, either representing a tool, a command or a toggle (shown in figure 3(a)). Selections are made by pointing with a cursor. To reduce the initial number of items, they can be grouped and hidden in a pull-down menu on top of the *toolbox*, as shown in figure 3(b).



(a) Each cell represents a menu item.

(b) Item can be grouped to reduce their number.

Fig. 3. The toolbox. Selection is made by pointing with a cursor [6].

2.2.2 The Virtual Tricorder

Wloka and Greenfield argue that the *toolbox* metaphor, described in section 2.2.1 in this paper, is allusive. They state that the tool metaphor is important, since humans are used to work with tools. Still *toolbox* hides those tools in a graphical 2D representation without direct manipulation. They propose the *Virtual Tricorder*, a multi-purpose

tool which immediately follows the user's hand movements via a six-degrees-of-freedom input device [40].

2.2.3 The Interaction Ball

One extension to the circular menus, described in section 2.1.2, is the *Interaction Ball* proposed by Häfner et al. [17]. As the name suggests the menu items are placed onto a virtual ball. When a button is pressed, the context sensitive menu shows up as a ball. The surface of the ball is evenly divided into four parts, each showing one menu item. With the rotation of his hand, the user can rotate the ball in defined angles. It snaps to each menu items' position. When the button is released, selection is made. Blind operation is possible for advanced users.

2.2.4 3D Widgets

Not entirely being a menu in augmented reality, three-dimensional widgets are of interest for menu considerations. Conner et al. define a widget as "an encapsulation of geometry and behavior used to control or display information about application objects" [8]. 2D widgets have been well studied. Often they are used in 3D space, and therefore the full potential of the six-degrees-of-freedom is not used. Conner et al. present a system and some basic considerations for widget creation in 3D space [8].

2.3 Augmented reality specific menus

Most of the menus described here use some sort of a virtual or real object, the user interacts with. It has an implicit function or mode it controls. By replacing a generic device with a more realistic real life object this leads to a more natural interaction for the user. The physical device can be positioned in the environment and therefore the user always knows its whereabouts even when the device is currently not being tracked and thus not being visualized. Furthermore users get some sort of haptical feedback from the device itself. For example touching a button on a tablet naturally gives tactile feedback from the tablet itself. This helps the system overcome a user's "feeling of interacting in the air" [9].

2.3.1 The ToolFinger: Supporting Complex Direct Manipulation in Virtual Environments

Wesche describes *ToolFinger* [37] which is a finger-shaped interaction widget. It is controlled by a pointing device the user holds in his hand. The *ToolFinger* is made up of thin and thick sections, each of the thick section corresponding to a specific command, such as copying, deleting, etc. To apply a command to a virtual object, the user intersects the *ToolFinger* with that object using his pointing device. When he presses a button the command is applied to that object. After releasing the button, the *ToolFinger* is ready for the next action. Support by visual feedback is given. A text label appears on the segment crossing an object. An example interaction with the *ToolFinger* is depicted in figure 4.

The advantage of the *ToolFinger* is, that it combines the task of tool selection and tool application to one single step. In many other menu designs, selection and application are two separate steps.

2.3.2 Personal Interaction Panel

The everyday work with pen and paper is transferred to augmented reality with the pen-and-tablet paradigm. With the *Personal Interaction Panel* [36] Szalavri et al. proposed a two-handed interface: The user holds a tablet in his non-dominant hand and interacts with it with a stylus in his other hand. He can manipulate objects in augmented reality with the tools in his hand. Where a six-degrees-of-freedom mouse is often used to transform or move objects, only a pen is used here. From a tool-palette on the tablet, which groups functions and system controls tasks, the user can select his desired action and for example transfer it to a virtual object. Also direct manipulation of objects is supported. A sample interaction with the *Personal Interaction Panel* can be found in figure 5.

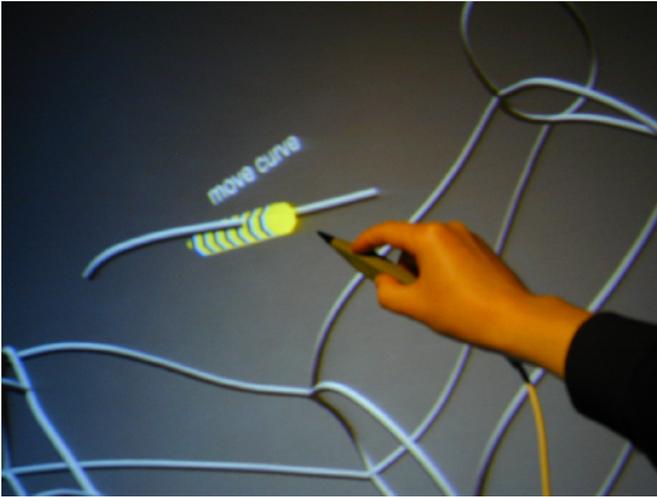


Fig. 4. A curve is being moved with the *ToolFinger*. Thick sections mark manipulation tools, which are selected by crossing them with an object. The *ToolFinger* is being moved by a tracked pointing device. [37]

The pen-and-tablet metaphor was first introduced by Billingham et al. [2], although they used implicit system control, instead of explicit menus: direct manipulation and drag-and-drop of objects, creating texture maps by simply drawing and changing viewpoints by drawing lines for example. Most of the tasks usually requiring a graphical menu were enabled through speech commands.

Coquillart and Wesche proposed the *Virtual Control Panel* [9]. They also used the pen-and-paper metaphor for their two-handed system control device and argue that the tablet, or palette, itself enables tactile feedback without any additional devices. Localizing the menu in space is easy since the user has to find a physical device (the tablet) which he can grab. Also interaction with the menu is easy since the buttons can be found by touching the palette, looked at from a different angle and by changing the orientation of the physical device more details can be shown.

Pen-and-tablet menus are sometimes referred to as *hand-oriented menus* [28].

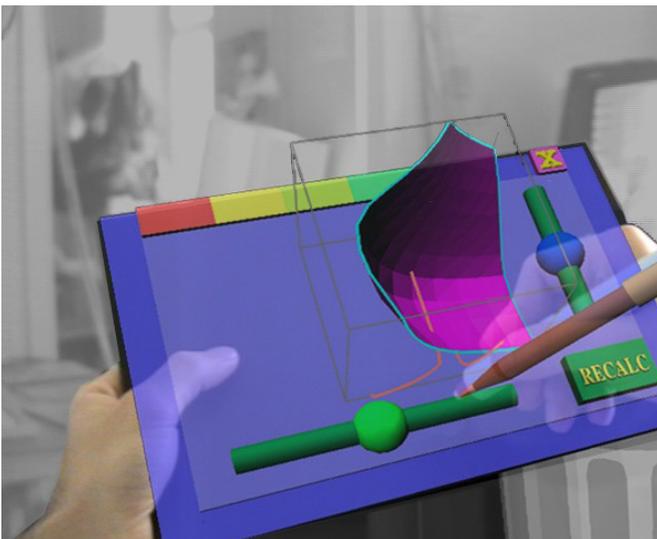


Fig. 5. The *Personal Interaction Panel* [36].

2.3.3 Command and Control Cube / C^3

The *Command and Control Cube* (C^3) [16] tries to fill in where hotkey and keyboard shortcut mechanisms in 2D environments help the user save time, and reduce mouse movements: Grosjean et al. propose a menu system which is based on the idea of marking menus, described in 2.1.2 in this paper. The C^3 is a cube, called the "bounding cube", made up of $3 \times 3 \times 3$ smaller cubes, called "slots". When the user pinches his thumb and index finger the C^3 shows up and a yellow pointer, called "sphere", appears in the bounding cube. The sphere starts in the center slot and follows the movements of the user's non-dominant hand, which is tracked in space. The sphere's movement is limited by the boundaries of the bigger cube and thus can be placed in all 27 slots. Each slot represents a menu item. When the sphere is placed in a slot and the user releases the pinch, the corresponding menu item is selected. With this system 26 menu items are possible, since the center slot is reserved as the sphere's starting point and as cancel option when no action is desired.

For visibility reasons only one horizontal plane of slots is visible at a time, which is the plane the sphere is currently in. A different horizontal plane can be selected by vertical movement of the user's hand and thus the sphere. Each menu item is represented as a graphical icon on top of its respective slot.

A blind mode is possible, which provides no visual feedback and can be used by expert users for quick selection.

Grosjean et al. evaluated their earlier proposed C^3 system [15]: They investigated the effect of the localization of the items in the bounding cube on performance, different levels of interaction (visual and no visual feedback) and, for the blind mode, audible and tactile feedback. Interestingly with the blind-mode, sound and tactile feedback has been found to decrease performance and users said it even disturbs them. The positioning of items in the cube shows a significant effect on performance. Items on the central plane were found fastest, followed by the upper plane. The lower plane's accuracy was the worst. The authors state that interactions too close or too far from the body perform worse than the ones where only little bending of a user's arm is required. For each slot of the C^3 they list an overall accuracy, in order to give designers recommendations on where to place often needed or time sensitive items.

2.3.4 TULIP Menu

Bowman et al. [4] describe the design of a menu, using the Pinch GlovesTM. Pinch GlovesTM are commercial input devices for virtual reality. They are cloth gloves with conductive finger tips. Once two fingers are pressed together a circuit is closed and a pinch between these two fingers is registered. Simply assigning each menu item to a pinch gesture would be easy, but also a high cognitive load would be generated, since either the user has to remember each gesture, or the screen would be cluttered with all available options. The authors decided to limit the number of pinch gestures to those in which a finger is pressed with the thumb of the same hand. Also they make the menu options visually available to the user. The hands of the user are camera tracked and the menu items are displayed as labels at the four relevant fingers of each hand. They decided to present the top level of a menu on the non-dominant hand, while a sub-menu is displayed on the dominant hand. Thus hierarchic menus can be created. A rendering of this menu is shown in figure 6. Selection is made by pinching the desired finger with the thumb. Blind operation, with the hands and thus the menu out of sight, can be achieved for advanced users.

In a first user-study they found that their menu design caused fatigue, since the user had to hold his hand up high for the menu items to be visible. In their final prototype they decided to let the menu float 0.25 meters above the physical location of the hands.

The number of menu items is limited to four on each hand. To enable larger menus they replaced the fourth option with a "more" item. Three items are available at once, while more options on the same menu level are arranged in columns of three along the palm, wrist and forearm of the user. This is where *TULIP* derived its name from: "*Three-Up, Labels In Palm*".

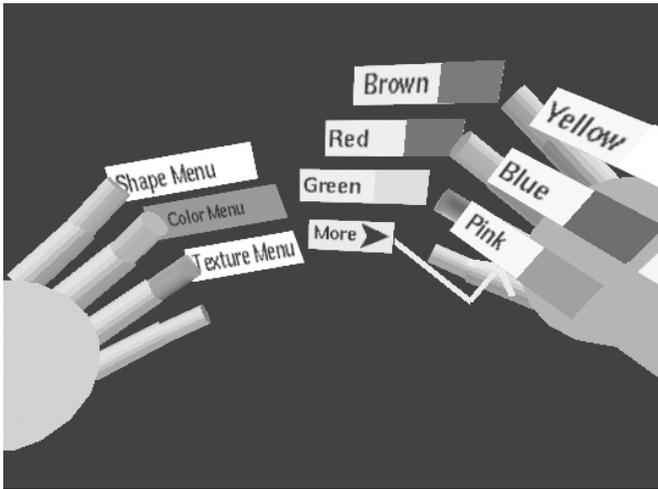


Fig. 6. The *Three-Up, Labels In Palm (TULIP)* menu, showing the top level on the non-dominant hand and a sub-menu on the dominant hand [4].

Another approach for menus in AR using Pinch Gloves™ can be found with the *Tinmith-Hand* in section 2.3.8.

2.3.5 FingARtips Gesture Based Direct Manipulation in Augmented Reality

The *FingARtips* control system [5] was designed with effective gesture interaction in mind. To achieve this, the authors decided for the support of tactile or haptic feedback, provision of occlusion cues and multi-fingered input. They used marker tracked gloves to select objects. Virtual menu items can be grabbed with two fingers from a list of objects or a shelf and placed into the desired position. Grabbing an object from the menu duplicates it, leaving the user with such an object in his hand. He then can move and tilt the object with his hand movement. Further transformation is provided in the same way: changing size works just like changing the size of a physical object by dragging the top 3cm up or down. Haptic feedback is provided by a buzzer mounted on the fingertip.

Informal user questionnaire showed that the system is easy and intuitive to use. Many were fascinated by the ability to manipulate virtual objects just in the same way as physical objects. With longer usage they fatigued after a while since they had to perform long movements.

2.3.6 Shake Menus

For *Shake Menus* White et al. [38] have been inspired by the human behavior of shaking a wrapped gift to discover what is hidden inside. The metaphor of shaking an object to reveal more information about its content is used to open up the menu and show the menu items. The interface is hidden as long as it is not needed and thus does not occlude the environment. Shaking an marker-tracked object displays the menus items as a radial menu around that object. The proposed system allows to display the menu object-, display- and world-referenced. They also provide ideas to switch between the alignments. For example by simply tilting an object-referenced menu, the menu is "released" from its parent and stays in the world-referenced position where it has been dropped. The user makes his selection by simply moving the tracked object below his desired option. Figure 7 shows the radial menu, displayed around the tangible.

They conducted a user-study to evaluate their system. Their results show that the average selection time for display- and object-referenced alignment was significantly faster than the world-referenced setup. Also the display-referenced setup was less error-prone than the object- or world-referenced setup. They also admit that in their study design they did not ask the user to look or move around. Therefore some participants did not perceive a difference between display-, object- and world-referenced alignment. They still argue that object-referenced

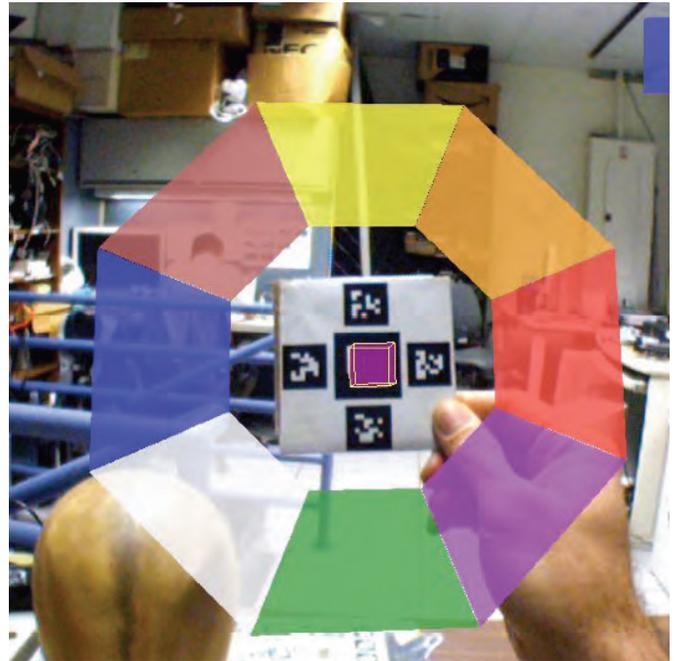


Fig. 7. *Shake Menus* are radial menus, intended for use in tangible augmented reality, and activated by shaking [38].

menus can be preferable since the menu options can be tilted and thus be viewed from a different angle. In a future research they plan a hybrid of the position condition and the ability to change orientation.

The authors are aware that shaking an object takes more time than simply pressing a button. They argue that a shaking gesture has the advantage of being applied to any visually tracked object and therefore no additional hardware is needed. Furthermore a shaking gesture is very easy to learn and achieve.

2.3.7 Tiles: A mixed reality authoring interface

Popyrev et al. propose the *Tiles system* [34]. They describe an authoring interface for collaborative environments. It allows several users to interact with the same workspace, for example a desk with a magnetic discussion board. In the *Tiles system* conventional tools used in discussion and collaboration such as board markers and paper cards are combined with marker tracked cards, called tiles, displaying virtual items. The aim was to decouple physical properties from the virtual data as much as possible, allowing for great flexibility in usage. Tracked cards can be attached to a wall and carry virtual information. To find the desired information, to be attached, a book is used. The user flips through the pages of the *menu tiles* and sees the virtual objects on each page. When he moves a blank data tile next to a menu tile the desired item is copied to the blank tile. For some operations, such as clearing a data tile or moving an item from one tile to another, special operator tiles exist.

Although they did not evaluate their system yet, informal questionnaire showed that user's found their system "easy to use, intuitive and quite joyable" [34].

2.3.8 Tinmith-Hand: unified user interface technology for mobile outdoor augmented reality and indoor virtual reality

As part of their *Tinmith-Metro* system Piekarski et al. [32] describe a menu system for augmented reality, which they explicitly propose for a wide variety of wearable AR applications. They describe how their system can be used in collaborative environments [31].

Tinmith-Metro allows the user to build and model 3D graphical objects. Since the modeling systems allows for a wide range of commands, the display would be cluttered with all the menu option visible at all times. For object manipulation the user wears Pinch Gloves™

which recognizes the pinch of a finger with any other finger. For their menu system they only evaluate pinches of the four fingers on each hand, thumb exempt, with the thumb on that same hand. The menu items are presented display-referenced left-to-right on the bottom of the screen. Therefore they support a maximum of eight menu items on one level. Selection is made by pressing a finger, mapped to a menu item, with the thumb of the same hand. To go back one level the user can touch his palm with any finger.

2.3.9 AR In-Situ 3D Model Menu for Outdoors

Hoang et al. describe how Pinch Gloves™ can be used with a 3D model menu [19]: The user has two options of selecting a 3D model for placement in an augmented environment. First he can walk to his desired position and with the use of a finger pinch he can enter a placement mode in which a linear array of models is shown. With the pinch of his left or right hand he can cycle through the models. Then he can walk around and view the placement from different positions. With another pinch the model is placed in the AR. With the second option the 3D models are displayed head relative. Therefore the authors recommend this menu option for tasks which include consideration of the final position of a model. Selection is also made, using Pinch Gloves™.

Another approach for menus in AR using Pinch Gloves™ can be found in section 2.3.4 of this paper.

3 DESIGN CONSIDERATIONS

Menu and user interface design in conventional 2D desktop environment has been well studied and lot of work has been published on this matter. With the emerging of virtual environments and augmented realities new attempts are needed. Menus in augmented reality exist in 3D space and are not necessarily limited to being two-dimensional. They can have a depth, rotation and position in space. Additionally the user's viewing angle and distance to the menu may vary. Objects might stand between the user and the menu itself, covering parts or all of the menu.

Interaction with a menu in AR is usually not done with a mouse or keyboard as it is in conventional desktop environments, since the user has other control devices or none at all. [20]

All of the above mentioned reasons result in a need for menus adapted to augmented reality. In the remainder of this section different considerations are given which have shown to be crucial for a menu in augmented reality applications. Only a few of the surveyed menus have been evaluated so far. Therefore only few general assumption on best practices can be given. Some more design considerations which apply only to certain menus were given in section 2 in the description of the particular menus.

3.1 Placement

Kim et al. [21] classified the position of the menu on the screen in three categories. A fourth category has later been introduced by Dachsel et al. [10].

- **Display-referenced:** the menu is at a fixed position on the display, meaning its position moves with the display. For example with a head mounted display the menu is always viewed from a fixed offset by the users. When a user moves around, the menu stays in the same position relative to the display.

Menus always facing the user are easy to read and interact with, but they can occlude important parts of the environment: these menus might be suitable to be displayed for a short amount of time for immediate focus, at the cost of making it harder for the user to orient himself in 3D space because of the occlusion [10].

Sometimes this is also referred to as view-fixed [21]. An example is shown in figure 8(a).

- **World-referenced:** the menu is associated with a fixed location in the augmented world. Either a user can drop the menu at a certain position or it is shown close to its context. For example a menu can be positioned by the user in the upper right corner of

a room. From now on it will be fixed to that position even when the user leaves the room and returns later, or until the menu is re-positioned by the user or the system.

In contrast to display-referenced menus world-referenced ones can be better displayed for a longer time, since the user can turn around and does not see the menu anymore. Still these menus might occlude some parts of the environment or a user might lose them in the environment.

Sometimes it is referred to as target-referenced [26] or world-fixed [21]. An example is shown in figure 8(b).

- **Object-referenced:** the menu is attached to an object. As the object is being moved or tilted, the menu is also being moved or tilted. Thus if the object is taken out of sight, the menu cannot be seen anymore. A study showed such menus can cause fatigue because the user has to hold the object in height of his eyes [4].

Object-referenced menus can always be displayed, since the user can push them quickly out of his sight, or menus can only be activated during the time of interaction with the object.

Object-referenced menus are also called manipulator-referenced [26] or object-fixed [21]. An example is shown in figure 8(c).

- **Body-referenced:** this describes a proprioceptional approach. The menu or certain control items are placed relative to the user's body. Examples are the *TULIP menu* [4] and *Toolspaces* [33]. More on the topic of proprioception is discussed in section 2.2.1.

An example menu, which allows the user to switch between these alignments is the the *TULIP menu* [4]. In a study they also found out, that, with longer usage, it can become uncomfortable for a user to hold the hands in eye's height. Thus they decided to position the menu 0.25 meters above the actual object it is attached to (a user's hands), in order to allow for a comfortable hand position. No subject in their study realized this offset.

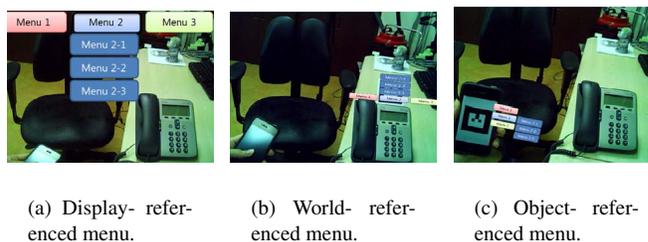


Fig. 8. Referencing methods of menus on the screen [26].

3.2 Orientation

In addition to the placement of a menu on the display, the orientation of it influences the space needed. When a menu always faces the user, such as with display-referenced menus, it is easy to read, but comes at cost of occluding the environment [10]. With the *Tinmith-Hand*, described in section 2.3.8, the authors tried to solve this problem by placing the menu in a single row at the bottom of the screen.

In order to achieve a better readability of the menu items at focus, Bowman et al. rotated them by thirty degrees compared to the user's handy in their *TULIP menu* [4]. The items following on the same or next level were not rotated, to allow for better differentiation.

3.3 Trigger mechanisms

A menu might be visible all the time or hidden. When it is hidden it has to be invoked by the user. Dachsel et al. categorized three actions for invocation [10]:

- **Selecting an icon:** the menu is explicitly activated by the user, for example through the selection of an icon on the screen.

- **Context dependent activation:** the menu is either explicitly or implicitly activated. Implicit activation might occur when the user looks at a certain position in the room or when he picks up a physical object.
- **Free activation:** the menu is activated for example through a gesture, the pinch of two fingers or the push of a button. One major advantage of free activation is, that the menu always travels with the user, stays connected to his body and is always within reach [30].

Besides that, a menu might always be visible on the display. For these menus no trigger mechanism is needed.

3.4 Interaction with the menu

With the introduction of menus in augmented reality it soon became clear, that not only the presentation and visualization of the menus itself need to be adapted for augmented reality systems, but also the control mechanisms. It is not sufficient to just translate well known interaction mechanisms from 2D desktop user interfaces to 3D space but new means have to be found [36]. With the increased number of dimensions of the input device, the possibility to make errors increases [18]. Therefore the mapping of a input device to a 3D task has to be thought of carefully [11]: one solution might be to constrain the degrees of freedom of the input device to only one dimension. Another option is to add additional hardware such as a scrolling wheel for selection in a list, or a button for item selection. A third solution might be to split a three-dimensional action into two separate actions, such as a wrist rotation for turning a circular menu plus a button push for selection.

These interaction techniques are an integral part of the menu itself and are described in detail with each menu in section 2.

3.5 Graphical animation

The animation of a menu is very closely coupled to interaction. Since in 3D space there is one more dimension than in 2D space, more animations than in 2D graphical user interfaces are possible. For example one object can be positioned further away than another. Dachsel et al. list some more animation possibilities in 3D space, compared to 2D [10]: zooming and blending, rotating and turning, opening, expanding, fanning and collapsing, either the entire menu or only parts of it. Graphical animation can help to visualize and clarify bigger menu structures, e.g. through collapsibility: menu items can be temporarily shown and hidden. Another example is a fish-eye zooming effect which can provide better readability of the items currently in focus, with all other items still being shown.

Some menus might not be usable without proper animation. For example some of the circular menus described in section 2.1.2 cannot function without proper animation since their rotation is closely coupled to the user's hand movement. [10]

Also graphical animation of the menu or animated effects on the menu can help a user navigate through a menu. For example in the *Command and Control Cube*, described in section 2.3.3, a user navigates the sphere through a three-dimensional cube, whereas only one of three levels is shown at one time. Without proper animation on level-change, a user might have a hard time navigating through the menu structure.

3.6 Number of menu items

The number of items on one menu level and the number of levels a user has to go to make his selection significantly influences the performance of a menu system as for example shown by Gerber et al. [13]: a selection task becomes significantly less efficient when a ring menu shows more than nine items. Kurtenbach et al. [24] showed that eight item per level and a depth of two levels or four items per level with a maximum depth of four should not be exceeded. Dachsel et al. [10] suggest no more than seven items to be shown at the same time in a menu.

Some menus allow only for a certain number of items by design, for example the *TULIP menu* [4] allows for a maximum of four items

on one level and with the *Command and Control Cube* 26 items can be used. [16] Both of these menus are described in section 2 in this paper.

3.7 Feedback

Coquillart et al. [9] describe that a menu in augmented reality has to overcome the "feeling of interacting in the air" [9], which users get when interacting with a menu without any tactile feedback, by giving them haptical feedback. Also visual or acoustic feedback upon selection of a menu item might increase the usability [28] [21] [15] [38].

In augmented reality physical objects might gain a new set of possible actions. For example they can trigger a menu action by turning them over or shaking it, which a user normally would not do with that physical object. Therefore these objects need to communicate their abilities to a user beforehand. White et al. describe how *visual hints* [39] can enhance the experience, especially to novice users. They propose a graphical representation of possible actions and consequences in augmented reality. These hints can be shown through ghosting, a written description or a diagram, and are depicted in figure 9.

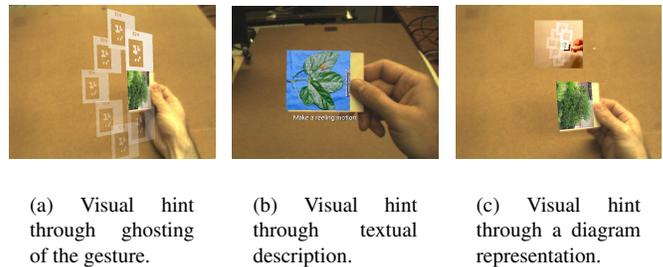


Fig. 9. Three examples for visual hints for a reeling gesture [39].

4 CONCLUSION

In this paper the need for special menus in augmented reality environments was presented. Augmented reality systems, being in three-dimensional space, offer more freedom compared to conventional 2D environments. Therefore more thought has to be given especially to graphical representation and interaction with such a menu.

Several menus, developed or adapted to augmented reality, have been explained in detail and their drawbacks or advantages have been pointed out where applicable. Especially menus specific to AR offer a new system control experience to the user, not known from two-dimensional desktop environments. The implementation of such menus can be realized at different costs: the *Shake Menu* for example does not need additional hardware to function, whereas the *TULIP menu* only works with the commercially available Pinch GlovesTM. Other problems arise when the user has to interact with physical items or do gestures with his hands in mid-air: he might experience fatigue from unfamiliar movements. One solution was shown with the *TULIP menu*: The menu is positioned slightly above the actual object in order to reduce the risk of fatigue. In the *C³* often needed menu items are placed in conveniently located slots, whereas less needed items demand greater movements by the user. Some publications [11] state it to be a good idea to control a menu with use of more than one object, since the dimensionality decreases. In contrast the *ToolFinger* menu approach even suggests only one object for task selection and task application. Another thought has to be given on a mechanism to relocate world-referenced menus: A user might lose such a menu, or it might be occluded by other objects.

Various paradigms known from 2D desktop environments, have been transferred to AR. Examples are the hotkey mechanism in the *Command and Control Cube* or the everyday work experience with pen and paper being transferred to AR with the *Personal Interaction Panel*.

In section 3 design considerations have been explained which are important especially in AR applications, such as the possibilities to position a graphical menu in 3D space. Also the considerations that have to be made on the number of menu items and its hierarchy depth were stated. Research has shown that there is a need for feedback so that users do not get the feeling to act in mid-air.

REFERENCES

- [1] R. Azuma. A survey of augmented reality. *Presence-Teleoperators and Virtual Environments*, 6(4):355–385, 1997.
- [2] M. Billinghurst, S. Baldis, L. Matheson, and M. Philips. 3D palette: a virtual reality content creation tool. *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 155–156, 1997.
- [3] D. a. Bowman and L. F. Hodges. Formalizing the Design, Evaluation, and Application of Interaction Techniques for Immersive Virtual Environments. *Journal of Visual Languages & Computing*, 10(1):37–53, Feb. 1999.
- [4] D. A. Bowman and C. A. Wingrave. Design and evaluation of menu systems for immersive virtual environments. In *Proceedings of IEEE Virtual Reality*, pages 149–156, 2001.
- [5] V. Buchmann, S. Violich, M. Billinghurst, and A. Cockburn. FingARtips - gesture based direct manipulation in Augmented Reality. In *Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 212–221. ACM, 2004.
- [6] J. Butterworth, A. Davidson, S. Hench, and M. T. Olano. 3DM: A three dimensional modeler using a head-mounted display. *Proceedings of the 1992 symposium on Interactive 3D graphics*, pages 135–138, 1992.
- [7] J. Callahan, D. Hopkins, M. Weiser, and B. Shneidermann. An empirical comparison of pie vs. linear menus. *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 95–100, 1988.
- [8] B. D. Conner, S. S. Snibbe, K. P. Herndon, D. C. Robbins, R. C. Zeleznik, and A. van Dam. Three-dimensional widgets. *Proceedings of the 1992 symposium on Interactive 3D graphics - SI3D '92*, pages 183–188, 1992.
- [9] S. Coquillart and G. Wesche. The virtual palette and the virtual remote control panel: a device and an interaction paradigm for the Responsive Workbench TM. *Proceedings of the IEEE Virtual Reality 1999*, pages 213–216, 1999.
- [10] R. Dachselt and A. Hübner. Three-dimensional menus: A survey and taxonomy. *Computers & Graphics*, 31(1):53–65, Jan. 2007.
- [11] R. P. Darken and R. Durost. Mixed-dimension interaction in virtual environments. *Proceedings of the ACM symposium on Virtual reality software and technology - VRST '05*, page 38, 2005.
- [12] M. F. Deering. HoloSketch: a virtual reality sketching/animation tool. *ACM Transactions on Computer-Human Interaction*, 2(3):220–238, Sept. 1995.
- [13] D. Gerber and D. Bechmann. Design and evaluation of the ring menu in virtual environments. *Immersive projection technologies*, 2004.
- [14] D. Gerber and D. Bechmann. The spin menu: a menu system for virtual environments. *IEEE Proceedings. VR 2005. Virtual Reality, 2005.*, 2005:271–272, 2005.
- [15] J. Grosjean, J.-M. Burkhardt, S. Coquillart, and P. Richard. Evaluation of the Command and Control Cube. *Proceedings. Fourth IEEE International Conference on Multimodal Interfaces*, pages 473–478, 2002.
- [16] J. Grosjean and S. Coquillart. Command & control cube: a shortcut paradigm for virtual environments. *7th EG Workshop on Virtual Environments & 5th Immersive Projection Technology Workshop*, pages 1–12, 2001.
- [17] U. Häfner, A. Simon, and M. Doulis. Evaluation of Complex CAD Data in a Virtual Environment. *International Immersive Projection Technology Workshop*, 10:157–165, 1999.
- [18] C. Hand. A survey of 3D interaction techniques. *Computer graphics forum*, 016(005):269–281, 1997.
- [19] T. N. Hoang and B. H. Thomas. Augmented reality in-situ 3D model menu for outdoors. *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 185–186, Sept. 2008.
- [20] R. Jacoby and S. Ellis. Using virtual menus in a virtual environment. In *Proceedings of SPIE, Visual Data Interpretation*, pages 39–48, 1992.
- [21] N. Kim, G. J. Kim, C.-M. Park, I. Lee, and S. H. Lim. Multimodal menu presentation and selection in immersive virtual environments. In *Proceedings-Virtual Reality Annual International Symposium: IEEE Virtual Reality*, page 281. IEEE Comput. Soc, 2000.
- [22] D. V. Krevelen and R. Poelman. A survey of augmented reality technologies, applications and limitations. *International Journal of Virtual Reality*, 9(2):1, 2010.
- [23] E. Kruijff. System control. *3D User Interface Design: Fundamental Techniques, Theory, and Practice. SIGGRAPH*, pages 147–165, 2000.
- [24] G. Kurtenbach and W. Buxton. The limits of expert performance using hierarchic marking menus. *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '93*, pages 482–487, 1993.
- [25] G. P. Kurtenbach, A. J. Sellen, and W. A. S. Buxton. Some articulatory and cognitive aspects of marking menus: an empirical study. *Journal of Human-Computer Interaction*, pages 1–31, 1991.
- [26] H. Lee, D. Kim, and W. Woo. Graphical Menus Using a Mobile Phone for Wearable AR Systems. *2011 International Symposium on Ubiquitous Virtual Reality*, pages 55–58, July 2011.
- [27] J. LIANG and M. GREEN. JDCAD: a highly interactive 3D modeling system. *Computers & Graphics*, 18(4):499–506, 1994.
- [28] R. W. Lindeman, J. L. Sibert, and J. K. Hahn. Hand-Held Windows : Towards Effective 2D Interaction in Immersive Virtual Environments. In *Virtual Reality, 1999. Proceedings., IEEE*, pages 205–212, 1999.
- [29] M. R. Mine. ISAAC: A virtual environment tool for the interactive construction of virtual worlds. *UNC Chapel Hill Computer Science Technical Report TR95-020*, 1995.
- [30] M. R. Mine, F. P. J. Brooks, and C. H. Sequin. Moving objects in space: exploiting proprioception in virtual-environment interaction. *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 19–26, 1997.
- [31] W. Piekarski and B. Thomas. Tinmith-Hand: unified user interface technology for mobile outdoor augmented reality and indoor virtual reality. *Proceedings IEEE Virtual Reality 2002*, 2002:287–288, 2002.
- [32] W. Piekarski and B. H. Thomas. Tinmith-metro: New outdoor techniques for creating city models with an augmented reality wearable computer. In *Proceedings of Fifth International Symposium on Wearable Computers*, pages 31–38. IEEE Comput. Soc, 2001.
- [33] J. Pierce and M. Conway. Toolspaces and glances: storing, accessing, and retrieving objects in 3D desktop applications. *Proceedings of the 1999 symposium on Interactive 3D graphics*, pages 163–168, 1999.
- [34] I. Poupjrev, D. Tan, and M. Billinghurst. Tiles: A mixed reality authoring interface. *INTERACT 2001 Conference on Human Computer Interaction*, pages 334–341, 2001.
- [35] C. Shaw and M. Green. THRED: a two-handed design system. *Multimedia Systems*, 5(2):126–139, 1997.
- [36] Z. Szalavári and M. Gervautz. The Personal Interaction Panel - a Two-Handed Interface for Augmented Reality. *Computer Graphics Forum*, 16(3):C335–C346, June 2008.
- [37] G. Wesche. The ToolFinger: supporting complex direct manipulation in virtual environments. *Proceedings of the workshop on Virtual environments 2003*, pages 39–45, 2003.
- [38] S. White, D. Feng, and S. Feiner. Interaction and presentation techniques for shake menus in tangible augmented reality. *2009 8th IEEE International Symposium on Mixed and Augmented Reality*, pages 39–48, Oct. 2009.
- [39] S. White, L. Lister, and S. Feiner. Visual hints for tangible gestures in augmented reality. In *6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 47–50. Ieee, Nov. 2007.
- [40] M. M. Wloka and E. Greenfield. The Virtual Tricorder : A Uniform Interface for Virtual Reality. *Proceedings of the 8th annual ACM symposium on User interface and software technology*, pages 39–40, 1995.