Available online at www.sciencedirect.com
ScienceDirect
Computers & Graphics 31 (2007) 53–65
COMPUTERS & GRAPHICS
www.elsevier.com/locate/cag

Virtual Environments

Three-dimensional menus: A survey and taxonomy

Raimund Dachselt*, Anett Hübner

TU Dresden, Multimedia Technology Group, D-01062 Dresden, Germany

Abstract

Various interaction techniques have been developed in the field of virtual and augmented reality. Whereas techniques for object selection, manipulation, travel, and wayfinding have already been covered in existing taxonomies in some detail, application control techniques have not yet been sufficiently considered. However, they are needed by almost every mixed reality application, e.g. for choosing from alternative objects or options. For this purpose a great variety of distinct three-dimensional (3D) menu selection techniques is available. This paper surveys existing 3D menus from the corpus of literature and classifies them according to various criteria. The taxonomy introduced here assists developers of interactive 3D applications to better evaluate their options when choosing, optimizing, and implementing a 3D menu technique. Since the taxonomy spans the design space for 3D menu solutions, it also aids researchers in identifying opportunities to improve or create novel virtual menu techniques.

Keywords: Virtual reality; Desktop VR; Augmented reality; 3D user interfaces; 3D widgets; Interaction techniques

1. Introduction

In the past decade much of the research in the field of virtual reality (VR) has been devoted to developing interaction techniques for object selection, manipulation, travel, and wayfinding. In addition to that, techniques for application or system control were introduced which allow for changing states, adjusting scalar values, and especially for choosing from alternative objects or options. However, interfaces for system control tasks in virtual environments (VEs) have not been extensively studied [1].

Since these tasks are also an integral part of conventional desktop interfaces, well-known 2D desktop interaction techniques were adapted to VEs. This works quite well for a number of 3D state control widgets (e.g. buttons), but causes problems for menu selection techniques. Thus, all menu solutions integrating 2D approaches into space face problems such as the greater skills required in reaching a menu item in space as well as the lack of tactile feedback [2]. Since the early work introducing virtual menus by Jacoby and Ellis [3], a great variety of distinct 3D solutions has been proposed for VEs. As Kim et al. postulate in [4], menus show enough idiosyncrasies that warrant a more in-depth look as a generic task of its own (compared to object manipulation and selection). Hence it is worth studying existing solutions and developing a perspective on their design and use.

Still, researchers and application developers cannot rely on an established set of 3D menu techniques being at hand for implementing interactive 3D applications. There is no repertory of menu solutions available comparable to 2D user interface development. Applications are often developed from scratch, especially in the even younger research areas of augmented reality (AR) and desktop VR. For these promising fields even less menu solutions are available. Moreover, to our knowledge there exists neither a survey nor a unifying taxonomy of application control and especially menu techniques in these fields. Along with missing usability studies, this makes it difficult for developers to choose and optimize a menu technique for their VE applications.

This work attempts to close this gap and to provide a comprehensive survey of graphical 3D menu solutions for all areas of the mixed reality continuum including the field of desktop VEs. In addition, by proposing detailed criteria for building a taxonomy, the surveyed solutions are

*Corresponding author.
E-mail address: dachselt@acm.org (R. Dachselt).
URL: http://www-mmt.inf.tu-dresden.de/dachselt/.

0097-8493/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cag.2006.09.006

classified in a way which supports VE application developers in the comparison and choice of appropriate menu solutions. Since the taxonomy spans the design space for 3D menu solutions, it also enables the identification of opportunities for improving or creating novel virtual menu techniques. Such novel techniques are likely to influence not only future VE applications, but also alternatives to present WIMP interfaces (windows, icons, menus, pointing device) on desktop computers.

The paper is structured as follows. The next section discusses previous work related to classification and taxonomy approaches in the field of 3D menus. In Section 3, existing menu techniques from the field of VR, AR, and desktop VR are surveyed. The main Section 4 presents our set of classification criteria and the menu taxonomy. This is followed in Section 5 by a discussion of resulting issues, such as alternative taxonomies and future research directions. Finally, Section 6 concludes this paper.

2. Related work

The work presented here builds on a large body of previous work from different areas, such as VR, AR, and desktop VR research. In addition, the taxonomy introduced in this paper relates to previously published classification approaches. Thereby we are mainly interested in techniques and 3D widgets [5] for application or system control.

In their fundamental work [3] Jacoby and Ellis provide a frame of reference for the design of virtual menus. They consider various design characteristics of menus, among them *invocation, location, reference frame, highlighting, selection*. They were taken into account in the development of our classification criteria (see Section 4.1).

A comprehensive overview of interaction techniques for immersive VEs is given in the book on 3D user interfaces by Bowman et al. [6], where techniques are classified in terms of task decomposition. The main categories identified are *selection*, *manipulation*, *travel*, *wayfinding*, *system control*, and *symbolic input*. They divide system control methods in *graphical menus*, *voice* & *gesture commands*, and *tools*. A further subdivision of graphical menus was conducted by means of *adapted 2D menu*, *1-DOF menu*, *3D widget*, and *TULIP*. This classification appears to be slightly arbitrary, mixing DOF, origin, and interaction devices as classification criteria. In particular, it offers no assistance to the application developer, since it is difficult to make an appropriate selection from it. We would like to continue this work by especially providing detailed criteria for building a taxonomy of menu solutions also including AR and desktop VR techniques. From the mentioned characteristics *placement*, *selection*, *representation*, and *structure* [6] we incorporated some in our classification categories presented in Section 4.1.

Kim et al. investigate in their study [4] the usability of various menu presentation and multimodal selection schemes in immersive VEs. They reclassify several 2D and 3D menu presentation styles in VEs and identify five major menu display methods: *pull-down*, *pop-up*, *stack menu*, *object-specific*, *oblique/layered*. By viewing the menu selection task as a composite task of *positioning* (manipulation) and *making a command*, and furthermore assigning different interaction modalities for each subtask, they identify 13 possible menu selection methods. The suitability of various combinations is compared and evaluated. Though it is one of the few papers entirely devoted to menu techniques, it is still limited to list menus in immersive environments and does not provide a comprehensive survey.

A comparison of specific VR menu systems was presented by Bowman and Wingrave in [1]. In this work the design of the Tulip menu is introduced and compared to the two common alternatives floating menus and pen & tablet menus in an empirical evaluation. Again, this work belongs to the few devoted solely to menu techniques.

All of the papers presented so far only address menus for immersive VEs. They do not consider AR or desktop VR solutions. Though various interaction techniques have been developed in the field of AR in the past few years, to our knowledge there exists no classification work or taxonomy, especially not for menu techniques.

3D interaction in the field of desktop VR is usually facilitated by 3D widgets [5]. Dachselt and Hinz present in [7] a first classification which is also devoted to desktop VR. Our widget classification scheme according to the criteria *interaction purpose/intention of use* constitutes the basis for this work. The following main categories were identified: *direct 3D object interaction*, *3D scene manipulation*, *exploration & visualization*, and *system/application*

Table 1
3D Widget classification according to interaction purpose [7]

Direct 3D Object Interaction	
	Object Selection
	Geometric Manipulation
3D-Scene Manipulation	
	Orientation and Navigation
	Scene Presentation Control
Exploration and Visualization	
	Geometric Exploration
	Hierarchy Visualization
	3D Graph Visualization
	2D-Data and Document Visualization
	Scientific Visualization
System / Application Control	
	State Control / Discrete Valuators
	Continuous Valuators
	Special Value Input
	Menu Selection
	Containers

control (see Table 1). As part of the latter, the approach presented here fits into the menu selection subcategory and significantly extends our previous simple taxonomy of menu widgets in a more systematic and comprehensive way including AR solutions as well.

3. Surveying existing 3D menu techniques

In order to classify existing menu techniques it is first of all necessary to find and analyze existing solutions to get an overview about state-of-the-art 3D menus. Our survey examines previous literature including the field of VR, AR, and desktop VR. According to the classification of Bowman et al. [6], we concentrate on graphical menus and do not further consider menus with only voice or gestural commands. Since menus often consist of or employ 3D widgets, the focus of this work lies on classifying interactive 3D widgets, which meet the criteria of a geometric representation as postulated by Conner et al. [5]. In addition, techniques which do not require a very specific I/O setting (usually an input device) are naturally of more interest pertaining to menu re-use. This is due to their generalization potential for various AR, VR, and desktop VR settings. However, the close dependency of menu widgets and their operation with specialized input devices has also contributed to some excellent solutions depending on specific hardware settings.

It can be observed that most of the menu techniques found in the literature were developed within the context of some broader system or application, thus being a by-product rather than a specific 3D menu contribution. Exceptions are recent developments such as the *command & control cube* [8], *TULIP menu* [1], *collapsible cylindrical trees* [9], *ToolFinger* [10], *Spin Menu* [11], or *generalized 3D carousel view* [12]. However, most of the literature stems from the middle of the nineties and is rooted in VR research.

The following three subsections roughly group the 3D menu solutions we have found with regard to their origin. Menus are surveyed from immersive and semi-immersive VEs, from AR applications, and from the field of desktop VR. Though we aimed at providing a comprehensive survey, some of the menus developed in the past might not appear here. Given similar solutions, not all existing variants and implementations are listed in this paper. Due to space limitations not all surveyed menus are mentioned and described in the following three subsections, but can be found online [13] in a list including their properties.

3.1. Menus from immersive and semi-immersive VEs

As already pointed out earlier, the largest number of 3D menu techniques comes from this area.

2D solutions in 3D environments: The embodiment of 2D menus in 3D interfaces started with the introduction of WIMP elements into VE. Examples are the *pop-up and pull-down 3D virtual menus* by Jacoby and Ellis [3] or the work done by Angus and Sowizral [19] by means of generally integrating a 2D interaction metaphor into 3D VEs. By combining the 3D interaction metaphor of a hand-held virtual tool with the software support available for a 2D user interface tool, the user is provided with familiar interaction concepts. However, associated 2D interaction techniques, such as click-and-drag remained an element of desktop environments only [20]. Several attempts were made by means of making 2D *X Windows widgets* available within 3D contexts, thus also incorporating traditional 2D menus. Early work by Feiner et al. [21] describes a heads-up window system, where images of 2D windows generated by an X server are overlaid on the user's view of the physical worlds. In [22] a hybrid 2D/3D user interface is described for immersive modeling, where *X Windows menus* are cloned to provide each eye its own copy. A Pinch Glove is used for simulating 2D mouse events to control the 2D menus. Recent work by Andujar et al. [23] even suggests an approach for developing portable application control GUIs by means of extending current 2D toolkits to display menus and other widgets either as 2D shapes on the desktop or as textured 3D objects within the virtual world. Several interaction techniques for object selection are thus conceivable, such as ray-casting and arm extension. Usually, all these solutions align 2D windows, menus, and other widgets to the user's view plane to facilitate interaction.

Menu selection, i.e. choosing from a list, is basically a 1D task. In 2D desktop environments menus can be operated by pressing a cursor key or using the mouse pointer. Even though the task is thereby transformed to a 2D one, such environments still provide the huge advantage of inherent constraints. The 'classical' *floating menu* (see Fig. 1a for an example) has been implemented in various VR scenarios, where the user needs to make the 3D cursor intersect the appropriate menu choice. It changes the 1D task into a 3D one and increases the possibility of making errors [2]. Selection from 2D pop-up menus in space is mostly done by casting a ray from the 3D mouse position. Typically, the user's finger or some sort of laser-pointer is used for selection in combination with a button-click on a physical device for activation. Directly touching menu items is often difficult, since menus might be out of reach [3]. Amongst various others, implementations can be found in [14,15, 24–27].

To pick an example, 2D menus are included in the Isaac interaction techniques [24]. They are virtual equivalents of conventional workstation pull-down menus floating in 3D space. Extensions of floating menus include pull-down menus with scrollable lists of items (e.g. a solution by Serra et al. in [28]) or fully fledged menu hierarchies such as the menus of the virtual windtunnel application by Bryson [15] (see Fig. 1c).

A different menu solution in terms of menu access and geometric layout was presented with the HoloSketch project [29] for a fish-tank VR setting. The *3D fade-up*

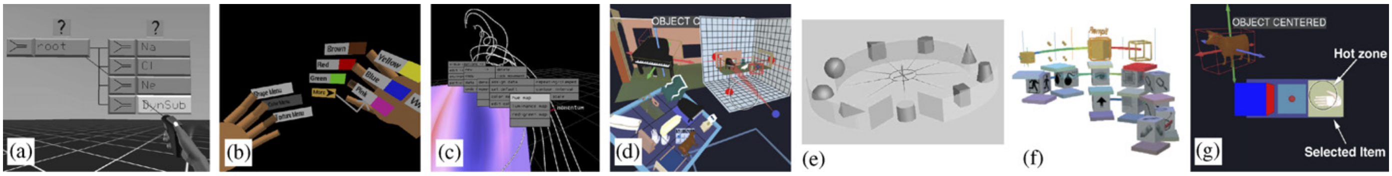

Fig. 1. VR-Menus: (a) floating menu ([14], ©1997 IEEE); (b) TULIP menu ([1], ©2001 IEEE); (c) virtual windtunnel menus [15] (image courtesy of NASA); (d) tear-off palette ([16], image courtesy of Mark Mine); (e) ring menu (reprinted from [17]. (Copyright 1994, with permission from Elsevier)); (f) spin menu ([11], ©2005 IEEE); (g) look-at-menu ([18], image courtesy of Mark Mine). All images reprinted by permission.

menu is a 3D pie menu which pops up by depressing the right wand button. Thereby the scene is faded out and the menu is faded up from the background at the same time. To select a menu item, the user pokes the wand tip at one of the circularly arranged buttons. This menu solution also combines textual display with 3D icons.

Glove-based menu selection allows a more natural style of selection using the fingers and hands. Typically, finger pinches are used to control a menu system, for example in the *Tinmith* [30] or *FingARtips* [31] techniques. The *TULIP menu* [1] (see Fig. 1b) uses Pinch Gloves, where up to 16 menu items are assigned to different fingers and a pinch between a finger and the thumb is interpreted as a menu selection.

Speech recognition enhanced menus: One of the problems resulting from the usage of instrumented gloves for gestural and spatial input is that hands can be too encumbered to use other tools [2]. That was the motivation for the development of the *hands-off interaction technique* [32] for menu display which involves the presentation of menu items as a 2D overlay onto the 3D world. Textual menu items are displayed on a view plane which moves relative to the user. The menu items are selected via speech recognition. Other menu solutions employ speech recognition as an alternative input channel in addition to graphical selection, among them the 3*D Palette* by Billinghurst et al. [33].

Hand-held menus improve upon the previously mentioned solutions by allowing a virtual menu (usually an object palette) to be controlled with one hand, whilst the other is selecting items from it. Prominent examples are the interaction techniques developed in the CHIMP project by Mine [18]. The *tear-off palette* (see Fig. 1d) contains miniature representations of available objects which the user can grab and add to the scene by moving his hand away from the palette. Another hand-operated menu is the *ring menu* developed for the JDCAD system by Liang and Green [17] (see Fig. 1e). With this 1-DOF menu geometric items are arranged on a ring. It can be rotated by the movements of hand and wrist, which results in the selection of the object currently displayed in the focus. An extension and evaluation of the ring menu is presented in [34].

Prop-based 'physical menus' also belong to the group of hand-held menus and were investigated in several research projects. Usually, 3D widgets and especially menus are attached to physical surfaces, which inherently provide means of constraining the interaction and providing natural feedback to the user. Usually, the position of the physical surfaces is tracked to allow for appropriate visual representations in space. The menu items placed on the surface can be selected with a tracked physical pen/stylus so that these menus are also called *pen-and-tablet menus* [6]. An example of this type of menus is the 3*D Palette* by Billinghurst et al. [33]. It is an interface for creating virtual scenes using a tracked tablet and digitizing pen. Another similar example for two-handed direct manipulation interfaces and menu selection based on tracked props is the *tool and object palette* attached to the so-called Personal Interaction Panel [35], which can be used in various VE settings. A related but slightly different approach was introduced with the *Virtual Tricorder* by Wloka and Greenfield [25]. It confines 2D interaction and information to a virtual hand-held object, which differs from plane surfaces. A 2D tricorder-anchored menu lets the user select among different tools. Since the menu is attached to the interface, the target acquisition is a relative task, thus allowing greater performance.

Workbench Menus: The responsive workbench and similar configurations are very attractive for direct manipulation [8]. Typically, menus are used by means of a toolbox containing various 3D-icons. Interaction is done with the stylus or by pinching with the gloves as for example on the responsive workbench by Cutler et al. in [27]. Another workbench system introduced the *virtual tool rack* [26]. It holds buttons with icons to enable tools and activate different modes of operation. Interaction is done with a two-handed approach, where 3D-intensive operations are performed with the 3D stylus while the 1D row of buttons can be controlled with a constrained mouse. The *command & control cube* (C^3) [8] was developed for a holobench setting and presents a 3D equivalent of the quick keyboard hotkey mechanism known from WIMP interfaces. This menu solution was inspired by marking menus, arranges menu items into a cubic configuration, and facilitates the quick selection of commands with a 6-DOF tracked button. Another quick menu selection technique based on marking menus was introduced with the *Spin Menu* [11] (see Fig. 1f). Items are arranged on a portion of a circle and controlled by rotating the wrist in the horizontal plane. Since at most 9–11 items can be displayed on a ring, hierarchical spin menus are suggested with crossed, concentric, or stacked layouts.

Menus with body-relative interaction can for example be attached to the user's body and thus take advantage of proprioception during operation. An example being developed in the CHIMP project is the *look-at-menu* [18] (see Fig. 1g), which can be attached to any object in the environment including the user. It is activated by the intersection of a user's current direction of view with a hot point representing the menu. To choose a new item the user moves his head to simply look at the desired item to select it. Thus, head orientation is employed instead of the traditional hand position (or movement of hand as with the ring menu [17]) to control the cursor and select an object. Another interesting approach with body-relative storage was developed by Pierce et al. with the *toolspaces and glances* technique [40]. Instead of switching contexts between 2D and 3D interaction, this approach relies entirely on 3D widgets, which are stored on the so-called toolspaces attached to the user's virtual body. Examples for the toolspaces are object or color palettes. Though widgets or objects are out of view, they can be accessed and retrieved when needed with the help of a navigation technique called glances.

3.2. Menus from AR applications

X Windows overlays including menu widgets as presented by Feiner et al. [21] belong to the very early menu solutions in the field of AR. They were already mentioned in the VR section, which indicates that AR applications basically use a number of similar menus to those in VR. Thereby solutions with hand-based interaction offer the advantage of an intuitive use of gestures such as pointing, grabbing, or stretching [36] and allow the user the freedom to also interact with physical objects.

Glove or hand-based menu selection: Some of the developed solutions even explicitly aim at combining the VR and AR domains, for example the *Tinmith-Hand menu system* by Piekarski and Thomas [30] (see Fig. 2a). It is a system employing gloves, where each finger maps to a displayed menu option at the border of a display. The user selects one item by pressing the appropriate finger against the thumb. The eight-item menu is located at the bottom of the user's display to navigate the options and select the actions required. This system resembles a menu style, which was quite common before the introduction of the mouse, i.e. pressing a function key to either activate a menu item or display a submenu. With the *FingARtips* 3D object menu by Buchmann et al. [31] gloves are also used to select objects such as buildings (see Fig. 2b). This time the gloves have three markers, which are used for hand tracking, i.e. for selecting objects or pressing buttons. By means of gesture recognition using the two fingers, objects can be selected and grabbed from the menu.

A tablet-and-pen-based approach was developed in the Studierstube project, with the previously mentioned *Personal Interaction Panel* as a two-handed interface for AR applications [35]. Controls associated with a magnetic tracked panel can be manipulated in a desktop manner using a pen. For object browsing and menu selection a tool-palette (3D clipboard) is employed. However, tracked props have the disadvantage of being rather inappropriate in truly mobile setups since they permanently occupy the user's hands and prevent him from performing everyday tasks [36]. Therefore, a solution featuring hand-based interaction with a wrist-attached augmented display was also developed within the Studierstube project [36] (see Fig. 2c).

Typically, AR applications address the domain of collaborative planning by seamlessly combining real and virtual objects. In the *mixed reality stage* planning application by Broll et al. [37] virtual models can be loaded from a *virtual menu* (see Fig. 2d). Menus are opened by issuing a single modal voice command or by pressing buttons of a wearable input device. Navigation in the menu hierarchy and selection of entries is accomplished by using the view pointer (crosshair shown in a head mounted display) and issuing a voice command.

Using physical props for menu selection: Whereas some of the previously mentioned approaches are similar to VR solutions, AR especially features computer vision techniques (e.g. for tracking the user's hand on a virtually augmented table) and integration with the real world by using physical objects for interaction as in tangible and graspable user interfaces. Take for example the TILES interface developed by Poupyrev et al. [38] where a book serves as a catalog or menu in presenting different virtual instrument models on each page. The so-called *menu tiles* (see Fig. 2e) make up a book with tiles attached to pages. As users flip through them, they can see virtual objects attached to each page, which can be chosen and copied from the book.

Besides these marker-based interaction techniques using props-like pads or paddles, other approaches use tools with real-world correspondence. An example is the *Tuister* introduced by Butz et al. in [39], which is a tangible interface for presenting and navigating hierarchical structures.

Fig. 2. AR-Menus: (a) Tinmith-Hand ([30], image courtesy of Wayne Piekarski); (b) FingARtips ([31], video frame courtesy of Oakley Buchmann); (c) Studierstube ([36], photograph courtesy of Dieter Schmalstieg); (d) mixed reality stage ([37], image courtesy of Fraunhofer FIT); (e) menu tiles ([38], photographs courtesy of Ivan Poupyrev); (f) TUISTER ([39], photograph courtesy of Andreas Butz). All images reprinted by permission.

Menu items are shown on a real cylindrical display, e.g. consisting of small discrete panels arranged to form a cylinder (see Fig. 2f). This display part can be rotated by hand against the second part of the Tuister interface, the handle held fixed by the other hand. By changing the rotation between both hands, menu hierarchies of arbitrary depth can be examined.

Another *explicit AR menu solution* avoiding complex hand-based or tool-based interfaces is a *3D spherical menu* as presented by Faisstnauer and Chihara in [45]. The menu is based on spherical menu layers and can be operated by simple 2D input devices, thus being suitable for rapid prototyping and testing of mobile AR applications. It is basically a 3D counterpart to the classical 2D desktop menu, which can alternatively operated by direct manipulation, i.e. using a hand-based interface.

3.3. Menus from the field of desktop VR

Widget-based solutions mainly dominate this area, since 3D widgets allow the subdivision of higher-dimensional interaction tasks into subtasks suitable for lower-dimensional input devices. Many of the solutions from the field of VR and AR can also be used in 3D desktop applications. A huge advantage of desktop solutions is the familiarity and high precision possible with well-known interface devices. The intrinsic constraints prove to provide benefits for certain 3D manipulation tasks [46].

Interaction is usually done by the mouse or keyboard and therefore often requires additional 3D widgets. Take for example the *ring menu* in a desktop version implemented by the authors [41], where buttons are added to allow for rotating the ring to the left or right side (see Fig. 3a). Another recent ring menu approach addresses the problem of a potentially high number of menu entries, such as for document browsing. It is the *generalized 3D carousel view* presented by Wang et al. in [12]. Document icons are arranged on a ring. Through the use of a clipping area and a termination marker even large amounts of menu items can be displayed. The menu is operated by click-selection, stepwise and free rotation using mouse or arrow keys.

The *revolving stage menus* (e.g. in [47], also called *rondel* in [42], see Fig. 3b) improve on simple ring menus in displaying a number of conventional flat menus arranged in a circular manner. The stage can be rotated until the desired single menu faces the user. Afterwards a selection can be made from this menu. In [46] Smith et al. introduce a menu, where objects such as pieces of furniture are not only arranged in a circular fashion but also in a vertical way on an invisible cylinder. This can be rotated, and an object can be selected by simply clicking on it.

3D hierarchy visualizations: With the help of detail-and-context techniques, the required screen space to display hierarchies can be significantly reduced, which is desirable even with modern displays. Whereas the well-known *cone trees* [48] and derivate solutions focus on the visualization of hierarchical information, other solutions emphasize the fast interaction and thus can rather be seen as menu solutions. Take for example the *collapsible cylindrical trees* technique introduced by Dachselt and Ebert [9], which uses rotating cylinders to display items of a menu. Submenus are made possible by smaller cylinders appearing from the supermenu cylinder, which is very much like a telescope (see Fig. 3c). Other higher level menus are squeezed but still visible, thus providing focus and context at the same time. Another work related to menu hierarchies is the *polyarchy visualization technique* described by Robertson et al. [49]. Separate hierarchies including the same item (e.g. a person) can be linked in 3D space and navigated using animated techniques.

3D desktop solutions: It is also worth looking at some commercial and experimental 3D desktops in order to find interesting spatial menu solutions. We have investigated Win3D [50], 3DNA [43], and Sun's project Looking Glass [51]. Several 3D menu widgets can be found, among them a *hinged menu* [50] containing 3D objects on different fold-away layers representing system controls for peripheral devices. Most menus are various geometric arrangements or layouts of items, e.g. in *horizontal or vertical stacks, drawers, panoramic walls* [43] (see Fig. 3d), *shelves*, or even *wardrobes*. The 'it3D' interactive toolkit library [52] for developing 3D applications provides several 3D widgets including *combo boxes* and *list menus* being very close to traditional 2D interface elements. A rather unusual approach with a loose layout was chosen with the *start palette* of the Task Gallery by Robertson et al. [44] (see Fig. 3e), presenting program and document icons on a 3D painter's palette.

4. A taxonomy of 3D menus

As we have seen in the previous survey section, there exists a huge number of 3D menu techniques. The way the

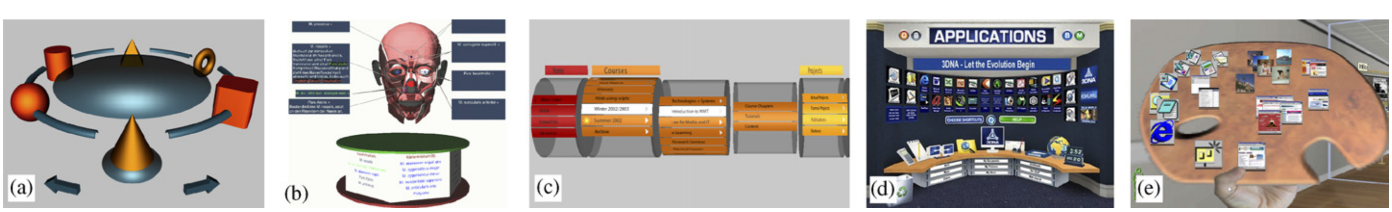

Fig. 3. Desktop VR menus: (a) Contigra RingMenu [41]; (b) rondel ([42], ©1997 Bernhard Preim); (c) collapsible cylindrical trees [9]; (d) 3DNA shelves ([43], image courtesy of 3DNA Corp.); (e) Task Gallery start palette ([44], image courtesy of George Robertson et al., ©2005 Microsoft Corporation). All images reprinted by permission.

solutions were ordered, i.e. by their *origin* and *basic interaction type*, already constitutes a first categorization. However, in order to better understand, describe, compare, and classify them we need to identify distinct properties. While analyzing specific menu solutions and related taxonomies in the field we devised the following classification criteria and associated properties.

4.1. Classification criteria and menu properties

The following paragraphs list and describe the main criteria (i.e. axes of the taxonomy) and associated properties of 3D menu solutions. Examples were added from our survey to better illustrate the characteristics.

Intention of use: This category describes menus by answering the question: What does the application developer want the user to choose from and for which purpose? The *number of displayed items* is an important characteristic of a virtual menu, for example influencing the item selection time as investigated in [34]. Certain menus allow only for a very limited or specific number of options (e.g. 8 items in the Tinmith-Hand menu system [30] or 26 menu items for the $3 \times 3 \times 3$ cubic grid (minus the center) of the C^3 technique [8]). Others can contain virtually any number of entries (e.g. 2D scrollable lists in space). A limited number of items (e.g. on the top level) can be a serious constraint for designing and balancing an efficiently structured menu [53]. Moreover, the number of menu items might decrease the usability of a solution, as was reported for the extended ring menu in [34], where the selection task is becoming harder with more than 9 items.

This leads to the property of the menu's *hierarchical nature*. It is an important property reflecting the intention of use. We distinguish between the following four types:

- *Temporary option menus*: These allow the user to quickly select from a limited number (usually $\leqslant 7$) of temporarily displayed items (mainly options). The menu is only invoked for a short time and vanishes after the selection. Typical representatives are the ToolFinger [10] or the rotary tool chooser [24].
- *Single menus*: Basically the same as the first type, but displayed for a longer time or even visible all the time. The number of selectable items can be greater than with the first menu type, and arbitrary items can also be displayed. This type includes toolbars and tool palettes such as [26,27,33].
- *Menu systems*: This is the same as the second type but extended to contain a submenu for each entry (if appropriate). That is, menu systems are menu hierarchies with a depth of 2. This is exemplified with the revolving stage/rondel [47,42].
- *Menu hierarchies*: These menus allow an arbitrary number of items, which are arranged in an arbitrary number of submenus (depth of hierarchy $\geqslant 3$). This type resembles well-known menu solutions from traditional desktop environments and is also called cascading menu or tree-structured menu [53]. Examples for it are the TUISTER [39] or the virtual windtunnel menus [15] (see Fig. 1c).

It should be noted that the term menu system is used in traditional desktop environments for menu hierarchies with an arbitrary depth, whereas in this survey we distinguish between menu systems and menu hierarchies. We consider this to be reasonable, since there are many examples within the literature, where menu solutions allow for displaying exactly two hierarchy levels.

Appearance and Structure: The *geometric structure* describes the appearance of a menu in terms of the supporting geometry. This might be a flat list (as in floating 3D menus), a disc (e.g. ring menu [17] or carousel view [12]), a sphere (e.g. Boule menu ball [54]), a cylinder (e.g. TUISTER [39]), cube (e.g. C^3 [8]), other Platonic bodies, or none at all.

Moreover, the *structural layout* describes how the items are arranged either on the supporting geometry or within space. This includes the acyclic list, cyclic list (usually ring), matrix, free arrangements (e.g. in the start palette [44]), and layouts following the geometric structure. Geometric structure and layout have a significant influence on memorability and interaction speed (compare results presented in [46]).

The *type of displayed data* is an important property, too. We distinguish between menu options appearing as:

- *3D-objects*, i.e. previews (e.g. on a 3D palette [33]).
- *Text entries* (e.g. with hands-off interaction [32] or TUISTER [39]).
- *Images*, i.e. icons (e.g. C^3 [8]).
- *Images and text combined* (e.g. generalized 3D carousel view [12]).
- *3D-objects and text combined* (e.g. 3D fade-up menu [29]).

Note that it essentially influences the selection of an appropriate menu, whether a geometric object (e.g. a product) or another abstract option (e.g. screen resolution) is to be chosen. In addition, the well-known and persistent problem of text readability in VEs also prohibits certain menu structures.

Moreover, the *size and spacing of menu items* also play an important role in selection, overall space consumption, and usability of a menu. Usually, flat menus or simple text lines consume the least amount of space, whereas Platonic bodies or revolving stages require far more space.

Placement: This category was introduced by [6] and comprises similar categories as already presented in [4,21]. In their early work on integrating 2D windows in a 3D environment, Feiner et al. make a distinction between *surround-fixed*, *display-fixed*, and *world-fixed windows* [21]. Surround-fixed windows are displayed at a fixed position within the world, display-fixed windows are positioned at a fixed location relative to the head-mounted display itself, whereas world-fixed windows are fixed to locations or objects in the 3D world. According to the extended placement options by Bowman et al. [6] menus can be placed in the following ways:

- *world-referenced* (most desktop VR menus).
- *object-referenced* (e.g. combo box in [52]).
- *head-referenced* (e.g. look-at-menu [18]).
- *body-referenced* (e.g. TULIP [1]).
- *device-referenced* (e.g. tool menu of the responsive workbench [27], PIP tool-palette [35], or fade-up menu [29]).

In addition to the general placement, *orientation* also plays an important role and influences the space needed. An example of a menu always facing the user is the hands-off interaction menu technique [32], and an example for the fixed location at the bottom of a display is the Tinmith-Hand menu system [30]. Within this context the question also arises, whether menus can be *repositioned* by the user, either to avoid occlusion or for personal preferences.

Invocation and availability: This category comprises menu properties which describe how users actually invoke a menu (make it appear or activate it). The first characteristic is *visibility*. Menus can be visible all the time (such as in typical 2D applications), can be temporarily displayed for the duration of the selection, or can be shown as long as the user wants them to be in sight. *Invocation* of non-visible menus can result from:

- *Selecting an icon* or other miniature.
- *Context dependent activation* related to either an object, other menu (for submenus) or some specific background.
- *Free activation* at an arbitrary point (menu hidden).
- *No action*, i.e. the menu is persistently visible.

Generally, menus can be activated by pressing some virtual or physical button, pinching two fingers or making a gesture. To provide an example, in the look-at-menu [18] the user's current direction of view activates a pop-up menu from a so-called hot point (e.g. a small red sphere). Virtual menus can also be hidden in locations fixed relative to the user's body, e.g. above the current field of view. An advantage is that menus attached to the user's body can be moved with the user as he moves through the environment, thus being always within reach [18].

Animation is also an important property of virtual menus. Although it can be associated with the category appearance, it is listed here, since it very much affects the user's interaction with the menu. Take for example some of the polyarchy visualization techniques described by Robertson et al. [49]. Without animation techniques it would hardly be possible to explore the multiple intersecting hierarchies. It is interesting to note that there are far more animation possibilities in 3D space than in 2D. Some of them are: blending, zooming-in, opening, expanding, collapsing, turning, rotating, fanning, or drawing out menus or parts of it.

The property *collapsibility* relates to the properties mentioned above, because it allows for compressing or temporarily hiding a menu without completely removing it. Usually, animation techniques are employed to expand or collapse parts of the menu, for example with the collapsible cylindrical trees menu [9]. Thus, the user is still provided with a coherent interaction with a menu without losing context or position.

Interaction and I/O setting: First of all, the proposed *interaction device* is of great importance. The menu solutions described in the literature include almost all devices and input channels, such as mouse, space ball, gesture and speech, 6-DOF tracking devices, computer vision, pen-and-tablet settings, and Pinch Gloves. Whereas some solutions are quite device-independent, many of the described solutions were developed in specific VR or AR settings. This is exemplified with the tool finger [10] or the TUISTER [39] requiring special input hardware. Moreover, the precision of an input device, for example sensor accuracy, together with the layout, size, and spacing of items strongly influences the operation of a menu.

Output devices also have an impact on interacting with menus. Take for example 2D menus in a VE displayed on a stereoscopic display, where problems such as appropriate display depth, disparity, or occlusion arise [2]. Naturally, the size of the display and the view distance are also of some importance. Other displays, e.g. head mounted displays, exclude the user from using devices such as mice or keyboards.

Summarizing these dependencies and system constraints, many 3D menu techniques support a certain *application type and I/O setting*. This could be a specific VR or AR setting as well as a desktop VR application type described in the original paper. Examples for that are input/output settings such as the responsive workbench [27] or the mixed reality stage [37].

According to Darken and Durost [20] *dimensionality* plays a crucial role in interaction design, especially the proper match of the dimensions of interaction techniques and tasks. With floating menus used in VR scenarios the user needs to make a 3D cursor intersect the appropriate menu choice. The 1D task is transformed into a 3D one and increases the possibility of making errors [2]. It was demonstrated in [20] that dimensional congruence results in superior performance. A menu solution might be directly congruent to the degrees of freedom of an input device. Take for example a device, which facilitates only rotation (plus an activation button), which can be directly mapped to a circular menu, such as a ring or cylinder. A second choice is to constrain the multiple degrees of freedom of an input device to support e.g. selection within a plane instead of positioning a 3D cursor. This can be done either with additional widgets or by not interpreting additional input channels from the device. A third way is to enhance an input device by adding better-suited facilities, such as a scrolling wheel for accomplishing the 1D menu selection task.

Another characteristic associated with interaction is appropriate *feedback/highlighting* provided by a menu [53]. There are many different solutions to this problem, including movement of items or their animation, highlights, item changes in color, brightness, geometry, size, as well as additional selection geometries. A related issue is the appropriate *visualization of the selection path*. Whereas in 2D menu systems this is a matter of course and state of the art, 3D menu systems and hierarchies rarely support it. An exception is the spin menu [11], which displays the path as stacked objects (compare Fig. 1f).

Usability: This category is closely related to the previous one. It contains information about available user studies and *evaluation criteria* concerning the usability of a menu solution. Most of the surveyed menus have not yet been evaluated formally, which makes it difficult to judge their deployment in a mixed reality application. Exceptions are to be found for the ring menu [34], spin menu [11], FingARtips [31], Tulip [1], and for various menu presentation and multimodal selection schemes in [4]. Typical criteria observed are the selection speed or average selection time for different numbers of items, the difficulty or ease of use, the number of wrong selections or error rate, the overall efficiency, the user satisfaction and comfort, and the ease of learning. Available *comparisons* also fall into this category. This can be the comparison of different layouts for a menu solution as in [11], the comparison of different selection methods (e.g. gestures and pinch actions vs. tracking-buttons in [4]), or the comparison with other menus, such as the Tulip against floating and pen-and-tablet menus in [1] or the ring menu against floating palettes and the command & control cube in [34].

Combinability: This is a feature of a 3D menu describing whether it can be aggregated with other menu solutions to form menu systems or hierarchies. Some single menus are well-suited to be combined to build a menu system; others prevent combination due to their geometric structure. Menus sometimes use a different technique especially for the top level. Take for example the revolving stage/rondel [42,47], where a ring menu is used with simple floating menus at each position. Another example is the spin menu [11] combining different approaches for displaying submenus (e.g. ring and stack menu). Again, animation techniques are employed in combined solutions to establish the link between them.

With the criteria and properties described in this subsection (see Table 2 for a summary) we define the design scope for 3D menus. Although certainly neither all orthogonal nor equally applicable to every menu solution, they form a reasonable basis for characterizing and evaluating existing approaches. Most of the surveyed menu solutions were described according to these criteria. Due to space limitations of this paper this description can be found as a comprehensive table online [13]. See Fig. 4 for an example snapshot from the online repository containing all mentioned as well as additional menu techniques.

Table 2
Summary of the classification criteria for 3D menus

Intention of use	
Number of displayed items	limited or not, range or definite value
Hierarchical nature	temporary option menu, single menu, menu system, menu hierarchy
Appearance and Structure	
Geometric structure	none, list, disc, sphere, cylinder, cube…
Structural layout	acyclic list, cyclic list (ring), matrix, free arrangement, geometric structure
Type of displayed data	3D-objects, text entries, images, images & text, 3D-objects & text
Size & spacing of items	
Placement	
Reference	world, object, head, body, device
Orientation	
Repositioning	
Invocation and availability	
Visibility	whole time, temporarily, user-dependent
Invocation	icon/miniature, context dependent, free, none
Animation	various ways
Collapsibility	
Interaction and I/O setting	
Interaction device dependence	mouse, gloves, pen & tablet, 6-DOF devices, computer vision, multiple etc.
Application type & setting	VR, AR, Desktop VR, 3D-Mobile
Dimensionality	mapping of interaction device and task
Feedback/highlighting	various ways
Visualization of selection path	
Usability	
Evaluation Criteria	selection speed, error rate, efficiency, user comfort, ease of use and learning
Comparison	of different layouts, selection methods, menu solutions
Combinability	

4.2. A taxonomy according to the hierarchical nature

The survey of 3D menu solutions presented in Section 3 already constitutes a first rough classification according to the criteria *origin* and *basic interaction type*. In addition, the classification criteria presented in detail in the previous subsection may serve as axes of taxonomies such as the one presented here. In our opinion purely academic classifications would less helpful for developers of 3D user interfaces. We therefore chose the main characteristic *hierarchical nature* as well suited for building a taxonomy. Having a mixed reality application developer in mind who searches for an appropriate 3D menu solution, this criterion enables the fundamental decision to use a *temporary option menu, single menu, menu system*, or *hierarchy*, which is necessary for most applications.

It seems sensible to further divide the taxonomy. The category *appearance and structure* lends itself as a secondary axis for further subgrouping existing solutions. Typically, VEs also possess a basic spatial structure and appearance, which should be matched by an appropriate choice of menus. Application developers often decide on

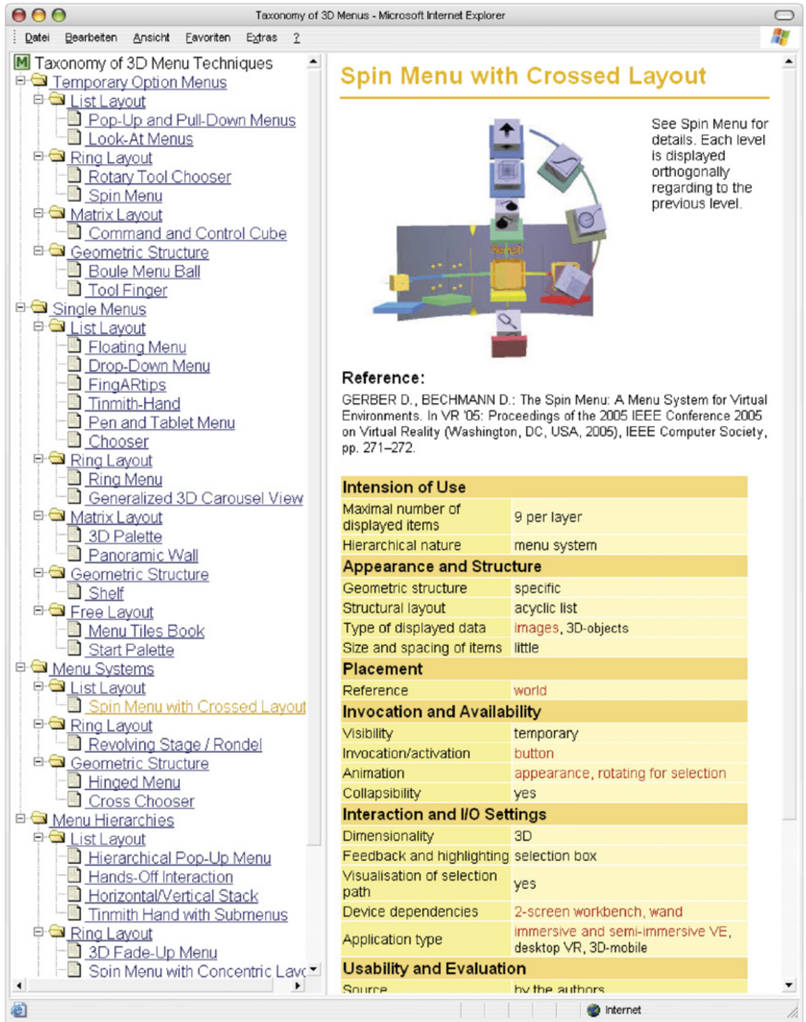

Fig. 4. Snapshot from the surveyed menu techniques available online at [13].

the structure (and also position) of a virtual menu after having made the basic decision e.g. to use a full menu hierarchy. By combining the criteria *hierarchical nature* and *structural layout* we conceived a taxonomy as depicted in Table 3. We have applied this taxonomy to the menu solutions examined, thereby summarizing similar solutions. In addition, an exclusive assignment is neither always possible nor necessary and some overlaps do exist.

5. Discussion

Looking at Table 3 one can observe that the majority of solutions were developed for single menus including tool palettes. This is not surprising, since not all VE applications need to have full-fledged menu hierarchies. Since the border between temporary option menus and single menus cannot be drawn sharply, solutions can be allocated to either of them. We do not consider this to be a disadvantage. The quick operation and limited number of items are practical reasons justifying a separation. Similarly, menu systems can be seen as subclasses of menu hierarchies. As stated in Section 4.1, the distinction is useful, if one thinks of common cases, where an optimized menu system with a hierarchy depth of 2 is sufficient. We can also notice that acyclic lists dominate the field, which is very similar to 2D desktop solutions. Among the geometric structures, circular arrangements are used more frequently, which can probably be attributed to the simplicity of rotation as an input dimension.

5.1. Alternative taxonomies

Since there are always numerous ways of classifying interaction techniques, the introduced taxonomy naturally exhibits limitations, depending on one's perspective. However, due to the extensive list of categories presented in Section 4.1, a number of other classification approaches can be imagined. In fact, we have also investigated other classifications, e.g. by using and combining important criteria such as *hierarchical nature*, *application type* (origin), *type of displayed data*, *placement*, *reference*, or *dimensionality* of the interaction task.

Table 3
Taxonomy of 3D menus according to the criteria *hierarchical nature* and *structural layout*

Temporary Option Menus		
List		
	Pop-up & pull-down menus [4, 18, 25]	
	Look-at menus [18, 40]	(Fig. 1g)
Ring		
	Rotary tool chooser [24]	
	Spin menu [11]	(Fig. 1f)
Matrix		
	Command & Control Cube [8]	
Geometric structure		
	Boule menu ball [54], Tool finger [10]	
Single Menus		
List		
	Drop-down menus [20, 21]	(Fig. 1a)
	FingARtips [31], Tinmith-Hand [30]	(Fig. 2a,b)
	Pen-and-tablet menus [33, 35]	
	Chooser [41], virtual tool rack [26]	
Ring		
	Ring menus [17, 41]	(Fig. 1e, 3a)
	Generalized 3D carousel view [12]	
Matrix		
	3D palettes [28, 33, 35, 36, 40]	(Fig. 1d)
	Panoramic wall [43]	(Fig. 3d)
Geometric structure		
	Shelves, horizontal/vertical stacks [4, 43, 50]	
Free layout		
	Menu tiles book [38]	(Fig. 2e)
	Start palette [44]	(Fig. 3e)
Menu Systems		
List		
	Spin menu with crossed layout [11]	
Ring		
	Revolving stage/rondel [47, 42]	(Fig. 3b)
	Spherical menu [45]	
Geometric structure		
	Hinged menu, Cross chooser [50]	
Menu Hierarchies		
List		
	Hands-off interaction [32]	
	Tinmith-Hand with submenus [30]	(Fig. 2a)
	Virtual Windtunnel menus [15]	(Fig. 1c)
Ring		
	3D fade-up (pie) menu [29]	
	Spin menu with concentric layout [11]	
	Collapsible cylindrical trees [9]	(Fig. 3c)
Geometric structure		
	Cone trees [48], TUISTER [39]	(Fig. 2f)
Free layout		
	TULIP [1]	(Fig. 1b)

However, not all categories and properties mentioned are suitable for a taxonomy of 3D menus; some of them are just of a descriptive nature. They are better suited to filter existing menu solutions in order to find an appropriate technique. Simple queries of system developers could be answered such as 'I am looking for a menu hierarchy being operated with gloves in the field of AR' or 'Which solutions are available with a circular layout containing 4–20 items displayed as either 3D objects or icons'. More complex queries are also conceivable to support the process of maximizing menu design goals. Certain parameters of a menu solution, such as hierarchical nature, placement, relative size, interaction devices, task performance, etc. could be related to each other in order to optimize the design and find or adapt the solution best suited with regard to given boundary conditions.

To allow such queries, we developed an initial website [13], where all surveyed solutions can be browsed, queried and ordered according to their properties. Fig. 4 depicts on the left the hierarchical taxonomy, which can be interactively navigated. On the right a data sheet is displayed for each menu technique with the actual values for all identified categories. The query and sorting functionality is currently achieved by an Excel sheet available at the same site. This needs to be improved and extended, perhaps with the help of a database or content management solution. From that follow some interesting directions for future research.

5.2. Research directions

Community effort would be of great value to improve the mentioned website, since it is neither possible to cover every single development in this field nor to correctly describe every detail of a menu solution. This work needs to be continued. A Wiki-powered website could be of much help to the community and extend far from 3D menus to 3D interaction elements in general. Moreover, besides classifying techniques it is desirable to consistently specify them in order to achieve portability across various applications within the mixed reality continuum. Community involvement might also pave the way to future standardization in the field of 3D user interfaces.

Several future developments are conceivable for 3D menus as a subclass of interaction elements. Beside acyclic lists dominating the menu solutions, other geometric structures have not been used frequently until now (compare Table 3). We consider them as possessing some potential for further developments, encouraged for example by the development of the command & control cube [8], collapsible cylindrical trees [9] or TUISTER solutions [39]. With the last, an important research direction becomes apparent. If special input devices are used at all, it is very sensible to tightly couple them to the geometric structure of a 3D menu. Imagine a steering wheel in a theme park, which directly controls object selection from a ring menu.

On the other hand, some of the menu solutions can and should be generalized to other application types and system settings, even to mobile devices. It can also be expected that 3D menus with a geometric representation will play a role as an integral part of multi-dimensional user interfaces realizing alternatives to present WIMP desktop interfaces. The work on polyarchies by Robertson et al. [49] points in that direction. Since it also incorporates animations, we suggest that animation techniques for 3D menu solutions should be studied further. They could be applied more intensively, since they allow a smooth combination of various techniques and reduce the cognitive burden. Another area largely unexplored until now is the combination of several menu techniques. Recent developments such as the layout variants presented with the spin menu [11] show the potential especially for combining certain types of geometries.

Since the taxonomy spans the design space for 3D menu solutions, it also aids researchers in identifying opportunities to improve or create novel virtual menu techniques. New techniques can be developed through (a) combining existing menus, (b) improving them, (c) developing new ones using empty or promising gaps within the taxonomy. To provide an example, we noticed that non-linear detail-and-context techniques have so far only rarely been applied to 3D user interfaces, though they have a huge potential for accommodating a larger number of items or bigger hierarchies. In any case it is desirable and necessary to thoroughly compare and evaluate existing and newly developed 3D menu techniques.

6. Conclusion

In this paper we have surveyed a multitude of 3D menu solutions from the area of VR, AR, and desktop VR. In order to describe, compare, and classify 3D menus, several characterizing categories and properties were presented along with a taxonomy considering their hierarchical nature. The identified criteria not only serve as axes of the presented taxonomy, but prepare the ground for further classifications. The taxonomy can be applied to evaluate the suitability of an existing menu solution for a particular mixed reality application. Thus, it facilitates the choice of appropriate widgets and techniques from the provided repertory according to selected criteria. Moreover, the design space described in this work allows researchers to create new menu solutions or to improve existing ones. We hope to have made a contribution to the field of 3D user interfaces in fundamentally examining this rather unexplored area of application controls in VEs. Hence the foundation is laid for an agreement on well-established 3D menu techniques eventually leading to standardization.

Acknowledgements

We would like to thank the reviewers for their useful comments, which helped us to improve this article. Special thanks to all the authors who permitted the reproduction of figures from their menu solutions.

References

[1] Bowman DA, Wingrave CA. Design and evaluation of menu systems for immersive virtual environments. In: IEEE virtual reality. Yokohama, Japan: IEEE computer society; 2001. p. 149–56.

[2] Hand C. A survey of 3D interaction techniques. Computer Graphics Forum 1997;16(5):269–81.

[3] Jacoby RH, Ellis SR. Using virtual menus in a virtual environment. Visual Data Interpretation 1992;1668(1):39–48.

[4] Kim N, Kim GJ, Park C-M, Lee I, Lim SH. Multimodal menu presentation and selection in immersive virtual environments. In: VR '00: Proceedings of the IEEE virtual reality 2000 conference. Washington, DC, USA: IEEE Computer Society; 2000. p. 281–8.

[5] Conner DB, Snibbe SS, Herndon KP, Robbins DC, Zeleznik RC, van Dam A. Three-dimensional widgets. In: ACM symposium on interactive 3D graphics. New York, Cambridge, MA, USA: ACM Press; 1992. p. 183–8.

[6] Bowman D, Kruijff E, Joseph J, LaViola J, Poupyrev I. 3D user interfaces: theory and practice. Reading, MA: Addison-Wesley; 2004.

[7] Dachselt R, Hinz M. Three-dimensional widgets revisited—towards future standardization. In: Bowman D, Froehlich B, Kitamura Y, Stuerzlinger W, editors. New directions in 3D user interfaces. Shaker Verlag; 2005. p. 89–92.

[8] Grosjean J, Coquillart S. Command & control cube: a shortcut paradigm for virtual environments. In: 7th EG workshop on virtual environments and fifth immersive projection technology workshop (IPT-EGVE 01); 2001.

[9] Dachselt R, Ebert J. Collapsible cylindrical trees: a fast hierarchical navigation technique. In: Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01). Washington, DC, USA: IEEE computer society; 2001. p. 79–86.

[10] 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: ACM Press; 2003. p. 39–45.

[11] Gerber D, Bechmann D. The spin menu: a menu system for virtual environments. In: VR '05: Proceedings of the 2005 IEEE conference 2005 on virtual reality. Washington, DC, USA: IEEE Computer Society; 2005. p. 271–2.

[12] Wang S, Poturalski M, Vronay D. Designing a generalized 3D carousel view. In: CHI '05: CHI '05 extended abstracts on human factors in computing systems. New York, NY, USA: ACM Press; 2005. p. 2017–20.

[13] Online Survey of 3D Menu Solutions ⟨http://www.3d-components.org/menus⟩, 2006.

[14] van Teylingen R, Ribarsky W, van der Mast C. Virtual data visualizer. IEEE Transactions on Visualization and Computer Graphics 1997;3(1):65–74.

[15] Bryson S. The virtual windtunnel: a high-performance virtual reality application. In: VR '93: Proceedings of the IEEE virtual reality annual international symposium; 1993. p. 20–6.

[16] Mine MR. Virtual environment interaction techniques. Technical Report TR95-018, University of North Carolina, 1995.

[17] Liang J, Green M. JDCAD: a highly interactive 3D modeling system. Computers and Graphics 1994;18(4):499–506.

[18] Mine MR, Frederick J, Brooks P, Séquin CH. 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. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co.; 1997. p. 19–26.

[19] Angus IG, Sowizral HA. Embedding the 2D interaction metaphor in a real 3D virtual environment. In: Proceedings of SPIE Vol. 2409, p. 282–293. In: Fisher SS, Merritt JO, Bolas MT, editors. Stereoscopic displays and virtual reality systems II, 1995. p. 282–93.

[20] Darken RP, Durost R. Mixed-dimension interaction in virtual environments. In: VRST '05: Proceedings of the ACM symposium on virtual reality software and technology. New York, NY, USA: ACM Press; 2005. p. 38–45.

[21] Feiner S, MacIntyre B, Haupt M, Solomon E. Windows on the World: 2D Windows for 3D augmented reality. In: UIST '93: Proceedings of the ACM symposium on user interface and technology; 1993. p. 145–55.

[22] Coninx K, Reeth FV, Flerackers E. A hybrid 2D/3D user interface for immersive object modeling. In: Computer graphics international; 1997. p. 47–55.

[23] Andujar C, Fairen M, Argelaguet F. A cost-effective approach for developing application-control GUIs for virtual environments. In: Proceedings of the IEEE virtual reality conference (VR 2006). Washington, DC, USA: IEEE Computer Society; 2006. p. 45–52.

[24] Mine MR. ISAAC: a virtual environment tool for the interactive construction of virtual worlds. Technical Report, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 1995.

[25] Wloka MM, Greenfield E. The virtual tricorder: a uniform interface for virtual reality. In: UIST '95: Proceedings of the eighth annual ACM symposium on user interface and software technology. New York, NY, USA: ACM Press; 1995. p. 39–40.

[26] Poston T, Serra L. Dextrous virtual work. Communications of the ACM 1996;39(5):37–45.

[27] Cutler LD, 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: ACM Press; 1997. p. 107–14.

[28] Serra L, Poston T, Hern N, Choon CB, Waterworth JA. Interaction techniques for a virtual workspace. In: Proceedings of VRST'95; 1995. p. 79–80.

[29] Deering MF. Holosketch: a virtual reality sketching/animation tool. ACM Transactions on Computation-Human Interactions 1995;2(3):220–38.

[30] Piekarski W, Thomas BH. Tinmith-metro: new outdoor techniques for creating city models with an augmented reality wearable computer. In: ISWC '01: Proceedings of the fifth IEEE international symposium on wearable computers. Washington, DC, USA: IEEE Computer Society; 2001. p. 31–8.

[31] Buchmann V, Violich S, Billinghurst M, Cockburn A. FingARtips: gesture based direct manipulation in augmented reality. In: GRAPHITE '04: Proceedings of the second international conference on computer graphics and interactive techniques in Australasia and South East Asia. New York, NY, USA: ACM Press; 2004. p. 212–21.

[32] Darken RP. Hands-off interaction with menus in virtual spaces. In: Proceedings of SPIE Vol. 2177, p. 365–371. In: Fisher SS, Merritt JO, Bolas MT, editors. Stereoscopic displays and virtual reality systems, 1994. p. 365–71.

[33] Billinghurst M, Baldis S, Matheson L, Philips M. 3D palette: a virtual reality content creation tool. In: VRST '97: Proceedings of the ACM symposium on virtual reality software and technology. New York, NY, USA: ACM Press; 1997. p. 155–6.

[34] Gerber D, Bechmann D. Design and evaluation of the ring menu in virtual environments. In: IPT 2004: Eighth immersive projection technology workshop, Ames, IA, USA, 2004.

[35] Szalavari Z, Gervautz M. The personal interaction panel—a two-handed interface for augmented reality. Computer Graphics Forum 1997;16(3):335–46.

[36] Veigl S, Kaltenbach A, Ledermann F, Reitmayr G, Schmalstieg D. Two-handed direct interaction with ARToolkit. In: Proceedings of ART'02 (poster paper), 2002.

[37] Broll W, Grünvogel S, Herbst I, Lindt I, Maercker M, Ohlenburg J, et al., Interactive props and choreography planning with the mixed reality stage. In: Rauterberg M, editor. ICEC, Lecture Notes in Computer Science, vol. 3166, Berlin: Springer; 2004. p. 185–92.

[38] Poupyrev I, Tan DS, Billinghurst M, Kato H, Regenbrecht H, Tetsutani N. Developing a generic augmented-reality interface. IEEE Computer 2002;35(3):44–50.

[39] Butz A, Gross M, Krüger A. Tuister: a tangible ui for hierarchical structures. In: IUI '04: Paroceedings of the ninth international conference on intelligent user interface. New York, NY, USA: ACM Press; 2004. p. 223–5.

[40] Pierce JS, Conway M, van Dantzich M, Robertson G. Toolspaces and glances: storing, accessing, and retrieving objects in 3d desktop applications. In: SI3D '99: Proceedings of the 1999 symposium on interactive 3D graphics. New York, NY, USA: ACM Press; 1999. p. 163–8.

[41] Online 3d widget classification, ⟨http://www.3d-components.org⟩, 2005.

[42] Preim B, Raab A, Strothotte T. Coherent zooming of illustrations with 3D-graphics and text. In: Proceedings of graphics interface 1997. Toronto, Ont., Canada, Canada: Canadian information processing society; 1997. p. 105–13.

[43] 3DNA Desktop by 3DNA Corp.: ⟨http://www.3dna.net/⟩, 2004.

[44] Robertson GG, van Dantzich M, Robbins DC, Czerwinski M, Hinckley K, Risden K, et al. The task gallery: a 3D window manager. In: CHI '00: Proceedings of the SIGCHI conference on human factors in computing systems. New York, NY, USA: ACM Press; 2000. p. 494–501.

[45] Faisstnauer C, Chihara K. Use of a spherical menu for mobile augmented reality applications. In: Computer graphics and imaging. IASTED/ACTA Press; 2003. p. 91–6.

[46] Smith G, Salzman T, Stuerzlinger W. 3d scene manipulation with 2d devices and constraints. In: Proceedings of Graphics Interface 2001. Canadian Information Processing Society; 2001. p. 135–42.

[47] Dachselt R. The challenge to build flexible user interface components for non-immersive 3D environments. In: Bullinger H-J, Ziegler J, editors. HCI (2). Lawrence Erlbaum; 1999. p. 1055–9.

[48] Robertson GG, Mackinlay JD, Card SK. Cone trees: animated 3d visualizations of hierarchical information. In: CHI '91: Proceedings of the SIGCHI conference on human factors in computing systems. New York, NY, USA: ACM Press; 1991. p. 189–94.

[49] Robertson GG, Cameron K, Czerwinski M, Robbins DC. Animated visualization of multiple intersecting hierarchies. Information Visualization 2002;1(1):50–65.

[50] Win3D platform by ClockWise Technologies Ltd.: ⟨http://www.clockwise3d.com⟩, 2005.

[51] Project Looking Glass by Sun Microsystems: ⟨http://www.sun.com/software/looking_glass⟩, 2005.

[52] Osawa N, Asai K, Saito F. An interactive toolkit library for 3D applications: it3d. In: Eigth eurographics workshop on virtual environments. Barcelona, Spain: Eurographics Association; 2002. p. 149–57.

[53] Shneiderman B. Designing the user interface: strategies for effective human–computer interaction. 3rd ed. Reading, MA: Addison-Wesley; 1998.

[54] VR Software Boule of the BMW AG, ⟨http://www.vr.iao.fhg.de/boule/⟩, 1999.