The Value of Constraints for 3D User Interfaces

Wolfgang Stuerzlinger
York University, Toronto, Canada
www.cse.yorku.ca/~wolfgang

Chadwick A. Wingrave
University of Central Florida,
USA

Abstract

User interfaces to three-dimensional environments are becoming more and more popular. Today this trend is fuelled through the introduction of social communication via virtual worlds, console and computer games, as well as 3D televisions.

We present a synopsis of the relevant abilities and restrictions introduced by both input and output technologies, as well as an overview of related human capabilities and limitations, including perceptual and cognitive issues. Partially based on this, we present a set of guidelines for 3D user interfaces. These guidelines are intended for developers of interactive 3D systems, such as computer and console games, 3D modeling packages, augmented reality systems, computer aided design systems, and virtual environments. The guidelines promote techniques, such as using appropriate constraints, that have been shown to work well in these types of environments.

1 Introduction

The interface is the bridge between the human and the effective use of their tools. In the beginning, the user interface for computers was text centric, constraining the human's expressiveness to command-line text. Later, two-dimensional graphical user interfaces (2D GUI's), using the WIMP (windows, icons, menus, pointer) metaphor became prevalent. GUI's offer precise tool control and have enabled many uses of computers in everyday life. Several post-WIMP interfaces operate outside these bounds, operating on human touch and voice modalities for multi-touch, tangible, sketching and voice interaction [63]. In addition, Reality-Based Interfaces [31] incorporate the human's body and natural understanding of the world into the interface as exemplified by three-dimensional user interfaces (3D UIs) in Virtual and Augmented Reality (VR/AR). Unfortunately, these new interface modalities, while liberating and potentially far more expressive, establish fewer bounds for the interaction between the human and their tools. Consequently interface understandability and task performance suffer. The extra dimensionality in 3D raises issues not seen in command-line and 2D applications. In these cases, appropriate interface constraints can preserve precision while retaining expressive interaction.

Today, examples of 3D user interfaces can be found in games, desktops, and computer-aided design (CAD) on a wide range of hardware configurations including traditional desktops, game consoles, and high-end virtual reality systems. These applications extend their interaction into 3D so as to gain some benefit of immersion, see e.g. [11]. Applications range from pure data visualization to highly interactive systems, with examples including static data visualization, architectural walkthroughs, and massively multiplayer online worlds for social and gaming purposes. Mobile devices also offer platforms more akin to 3D user interfaces as compared to their desktop predecessors.

However, creating effective user interfaces for 3D systems is a difficult problem [41,28,69]. The extra dimensionality of 3D gives the user much more freedom, raising issues not seen in 2D applications. Directly exposing all facets of the additional freedom of 3D to the user leads to extremely complex user interfaces, as evident in high-end CAD systems. Some recent systems have shown that it is possible to create user interfaces for 3D that are significantly less complex. For example, see Google Sketch-Up [24], SESAME [45], or the content editor in the Spore game [21]. Other examples use interaction techniques that are based on imagination to work around some of the issues [35].

In the following sections, we identify the challenges of creating interactive 3D applications. All of these challenges are based on the capabilities and limitations of humans and technical constraints. There are three main categories we will consider: input devices, display devices, and human issues. This is followed by a discussion of guidelines that can help constrain 3D interaction. This work amounts to a high-level overview of the state-of-the-art in 3D user interface technologies and human capabilities. For more details we refer the reader to technical survey articles, such as [14].

2 Capabilities and Limitations

Any user interface is part of a feedback loop, which involves a human reacting to system output and interacting with input devices to control the system. Hence, we discuss each of these three aspects here.

2.1 Input Devices

The variety of available 3D UI input devices is large. One factor is that applications have different requirements. Also the limitations and capabilities of each device class vary significantly. Hence, there is no single best input technology and no common hardware platform. Important to all input devices are their reported data in terms of degrees of freedom (DOF). Each degree is a dimension in which the device reports. A mouse is a 2DOF device, while a typical 3D UI input devices, often called a 3D tracker, provides 6DOF, for the 3 spatial dimensions and the rotations around each axis, i.e. heading, pitch, and roll. Additionally, devices can be described in terms of the independence of their dimensions, i.e. the number dimensions that can be controlled at once. For example, a mouse is a 2DOF device but the knobs on the child's toy Etch-a-Sketch are 1+1DOF. Moreover, and while several devices can track multiple points, the most common 6DOF devices track a single point of the physical device held by the user such as a pen, ball, or puck-shaped device.

The classic desktop input devices, the mouse and keyboard, are used in 3D UIs due to their ubiquitous availability and user familiarity. More commonly used, however, are position tracking systems that track multiple points in six degrees of freedom (6DOF). Between these extremes is a class of single point 6DOF devices, such as the Spaceball. Some of these 6DOF devices also provide haptic feedback. Input devices that operate on the surface are common as well, including sketch, touch and tangible input. Lastly, an emerging class of input devices, termed spatially convenient [70], is often used for gestures and basic input. A prominent example is the Nintendo Wii Remote (Wiimote).

Keyboards and mice vary in their utility in 3D UI's. Many CAD and modeling programs use them as the sole means of interaction with the third degree of freedom and rotations are made accessible via keyboard modifier keys, buttons, button combinations, or various on-screen manipulators such as 3D widgets

[58]. Most 3D desktop games also use the mouse by constraining it to viewpoint control and the keyboard for travel and lateral motions, an interface made popular by id Software. Many console game controllers operate in a similar manner but replace the mouse with thumb-sticks. As many users of game "applications" play very frequently, they quickly evolve expertise. This has led to rapid interface enhancements in this domain, such as the differentiation between viewpoint and orientation control across devices.

However, many 3D UIs enable the user to physically move freely through space. Then the user will often find themselves away from surfaces that can support a mouse or keyboard. Hand-held chorded keyboards are an option but then the user's hands are tasked with holding the device and cannot perform natural actions such as grasping and releasing. One practical use of a mouse is to pair it with a 3D tracker and to use only the buttons or click-wheel of the mouse for discrete input. This is often called a flying mouse, "wand", or "bat" [64]. More and more frequently the Wiimote is used in such a role, see also below.

Between 2D devices and full 6DOF multi-point tracking are several other classes of devices. Among them are ball or puck shaped devices, such as the Spaceball, that sits on the desk. The user can apply isotonic forces in three dimensions, as well as twist in three dimensions for full 6DOF input. Through constraining the user's movements to a very small region there is little fatigue and natural control of more dimensions simultaneously compared to a mouse. However, the inherent sensitivity to small movements of this device often leads to a negative first-use experience [10]. 3D mice are also used to provide 6DOF input. These are often wireless and easily passed between users. Some devices, such as the InterSense Inertia Cube, only report orientation information, often sufficient for head orientation tracking or simple ray-casting pointing. The CubicMouse [23] allows for separate control of each dimension by having one manipulatable stick along each axis that can be pulled and twisted, translating these motions into 6DOF input.

Another class of input devices, called haptics devices, uses small robot arms and enable the user to move a pen or any other device attached to the end of the robot actuator. The range of motion is usually limited to a soccer-ball sized volume, unless one considers expensive high-end devices. Practically all of these devices also use the motors in the joints to "push back, which can provide the user with a haptic experience, i.e. the sensation of hitting the pen onto the surface of an object.

A more recent trend in input devices has been the use of commodity devices and sensors, as exemplified by the Wiimote. This class is termed Spatially Convenient devices [70]. Such devices are defined by three characteristics: often incomplete or limited spatial input, yet many useful functionalities and convenience in terms of commodity price, easy setup and high durability. The Wiimote is a game controller designed for the Nintendo Wii console and wirelessly provides acceleration and orientation change at a single point. It also uses an infrared camera to sense emitted IR light for a limited form of 3D tracking that works only when the device is pointed at the screen. The Wiimote has multiple buttons, a speaker, LEDs and a rumble device. It is now common for game controllers to contain similar functionality. High-end mobile phones contain similar functionality and hence can also be used as spatially convenient input devices. The iPhone 3GS includes additionally a multi-touch screen, a GPS, a magnetometer, and a significant amount of processing power. The impact of such devices onto 3D UI's is only starting to become apparent.

Gloves afford a natural approach to interaction. This makes them a common choice for 3D input devices. They vary in the form of data collected and ease of use. The discrete input of Pinchgloves can be easier to work with than other gloves, making it possible to create menu systems [12] or virtual keyboards

[13]. The tactile feedback provided to the user makes it easy to understand when fingers touch and when they release. Some gloves return multiple dimensions of flex for each finger. However, this form of data is hard to handle and the sensors usually require periodic re-initialization to be accurate [34]. Moreover, long-term use can be problematic due to salt in the sweat affecting the sensors. Also, the lack of haptic feedback is an issue users report frequently. Mechanical tracking of finger bend angles can remove the re-initialization issue, but such gloves are cumbersome to put on and take off.

Free-space 6DOF tracking of a user's head, a hand, or a tool can be achieved with multiple approaches, each with associated tradeoffs [67,70]. Free-space tracking generally suffers from decreased precision, increased jitter and noise, and also increased lag relative to desktop devices such as a mouse. In the best case, precision can be in the millimeter range, i.e. in a freshly and fully calibrated system under ideal conditions. Compare this to the precision needs of a mouse, where miniscule hand and finger motions need to be tracked to enable pixel-accurate pointing. Hence, mice offer resolutions up to 2400 dpi and are generally one or two orders of magnitude more precise than trackers. Update rate is another important factor in these devices, with mice typically tracking at 125Hz [1]. This is comparable to current 3D input devices. However, increased jitter, noise and latency in 3D trackers relative to common desktop devices have a significantly negative impact on human performance, see below. The technical alternative, algorithmic smoothing and filtering, trades off precision for latency, which again impacts performance.

An important use of free-space tracking is head tracking. Head tracking aligns computer-generated images with the user's current eye positions. The reduction of technical lag and other encumbrances, and the increase of precision and accuracy have lead to reasonable success of head tracking in VR applications. In Augmented Reality systems, the precision requirements are much higher, as the computer generated environment is superimposed onto the real world and humans readily identify inconsistencies in alignment. This is one of the challenges that AR faces [4].

Human gesturing is another form of input, which can use various forms of free-space tracking. Simple gestures include shakes or orienting, while more complicated gestures can incorporate multiple movements over time. Accelerometer and gyroscopic sensors are an alternative to recognize such gestures, and these sensors require no external frame of reference. This approach has been used for games and mobile devices, for example on the Wii or iPhone. Existing work using paired accelerometers and gyroscopes have achieved user-independent 95% recognition accuracy for a set of 25 gestures with a small training set [30]. This corresponds to a 5% failure rate, which is still high enough to significantly impact user performance, see e.g. [2].

A final class of input devices is the tangible, surface and sketch-based hardware. Seen as research fields on their own, the potential benefit of these devices for 3D interaction, and the benefit to them, in understanding 3D interaction, is high. For instance, the tangible nature of near-field haptics has improved 3D UIs such as holding a clipboard to perform pen and tablet operations [12] or a baby head prop to improve a neurosurgery application [29]. The naturalness of writing notes and sketching in a virtual environment can have a lot of potential but is limited to hardware configurations that do not block the user's view of the work.

2.2 Display Devices

Display devices are the most readily identifiable aspect of 3D UIs, and are considered by some to be the first indication of the 3D experience to come.

More importantly, displays influence the interaction to such an extent that it would be appropriate to say they shape the type of possible interactions. A wide range of devices exist for 3D UIs, from head mounted displays (HMDs), large projected displays such as CAVEs or multi-walled systems, desktops, to volumetric displays [42]. The qualities of these displays will be discussed below, followed by display capabilities and limitations.

Two complimentary characteristics of displays include field of view (FOV) and field of regard (FOR). FOV is the viewing area from the user's viewpoint as measured in degrees. While this is typically measured diagonally so as to appear larger, vertical and horizontal FOV are actually important requirements for different applications, such as larger horizontal FOV for greater peripheral vision [18]. Field of regard is the amount of usable space around the user in degrees, which the display can provide. These are important for contrasting the capabilities between common displays such as CAVES which provide a large FOVs, equal to its FOR, and HMDs which provide relatively small FOVs (dependent upon the HMD) with complete FOR when a 6DOF head tracker is used.

Other qualities of displays are also important factors. They include display size, pixel density, brightness, costs, and 3D capabilities. Additionally, it matters how the displays are used and arranged. For instance, placing displays side-by-side in a tile-like fashion can create large high-resolution displays. Large displays, despite comparable viewing angles to smaller-up close displays, can be more engaging [52].

There are many output devices that can be used for 3D systems. The ubiquitous desktop monitors are the most commonly used form of display and continue to decrease in cost while increasing in size and resolution. Another cost-effective alternative is projected displays on a screen or wall, which easily creates very large displays. These too continue to drop in cost while increasing in resolution. One notable issue of projectors is that their resolution and update rates, important for stereo displays, are lower than desktop monitors. Additionally, they suffer from distortion due to off-axis key-stoning effects, colors, and marks on display surfaces and also the prominent need for large spaces for projection [51]. Additional issues, include heat, size and noise of these projectors. Inexpensive commercial 3D displays have appeared recently, brought to market for 3D gaming and entertainment purposes. They use either an internal projector or LCD display in conjunction with stereo glasses to achieve their effect.

Immersive projection systems such as CAVE's [17] or large immersive projection walls frequently use stereo projectors to immerse the user. However, all these systems are essentially single-user devices, as they afford only one perspectively correct image. While a few two-user systems have been demonstrated [39], they are exceedingly rare, as they require projection systems with more than 120Hz. Moreover, all these systems have large space requirements (6-sided CAVEs usually need a room that is three stories high), are very expensive to build and to maintain. The benefits of these systems are the wide field of view, completely surrounding for 6-walled CAVEs.

HMDs, whether stereo or not, can completely block-out the real world to allow the user to focus on the 3D world. Also, HMDs require minimum instrumentation of the surrounding space to operate. However, the devices are often bulky on the user's head and the benefit of blocking out the real world is also a drawback: 1) an immersed collaborator can't see surrounding collaborators and 2) it can restrict users from walking simply because they don't feel like they can. Moreover, most devices have a very small field-of-view, 30-40°, which is equal or less than what a typical computer monitor affords. A particularly insidious effect is that a smaller field-of-view also inhibits peripheral vision, e.g. [3], and/or spatial memory. e.g. [5], both of which greatly affect navigation. Head-mounted displays with a full field-of-view for each eye, approximately 110°, are now available, but are still expensive and heavy. On the positive side, head-mounted displays with head tracking can provide a field of regard only comparable with the most expensive six-walled surround-screen systems. Lastly, Augmented Realty see-through HMDs are a class of HMD where the user looks into the real world with the virtual world superimposed on top. These devices add virtual content to the real-world scene, but have issues regarding brightness, tracking latency, weight, and accuracy. Consequently, hand-held displays have been used more and more for Augmented Reality in recent work.

Lastly, there are true 3D display systems typically referred to as volumetric displays. A variety of technologies exist for this, and can generate “glowing points” inside a volume where these points are equally visible from every direction. The main issue with this concept is that users then see the front *and* back of objects simultaneously. However, this is something that the human visual system is not capable of interpreting for the general case. Hence, these displays are generally only usable for displaying wire frame or point-cloud data and/or require head tracking. For more issues with current 3D display technologies, such as low brightness, instable display, see [26]. One new class of system that has been demonstrated recently generates different images for different viewing directions by extending the concepts used in auto-stereoscopic displays, e.g. the Holografika 3D display systems and the USC Lightfield Display [32]. These technologies project many images into many (typically horizontal) directions simultaneously, which allows the viewer to move freely (typically side-to-side) without head tracking. However, they are not yet at a stage where they can be used in office or home settings.

2.2.1 Stereo

Stereo displays are often seen as a critical component of applications with 3D user interfaces. There are multiple technologies capable of generating a stereoscopic display; that is, the generation of a separate images for each eye. The most commonly used technology is stereo glasses, i.e. glasses where different images are displayed and the glasses separate a left and right image for the user. Active stereo glasses, glasses synced to the display to shutter between displaying two separate images, typically necessitate twice the frame rate of normal displays. CRTs, DLP TVs, and recently LCD displays offer sufficiently high update rates for stereo display (120Hz) and higher with affordable projectors lagging behind. This type of approach is often used in large surround-screen systems, created with tiled or projected displays. Passive stereo glasses typically used polarized light or filtered light approaches to achieve different views per eye. Polarized light approaches require display surfaces that retain polarization and often twice as many projectors, one for each eye.

A problem common to all glasses is that they negatively impact collaboration by making the eyes less visible. Yet, eye contact is very important to humans; well known to researchers of computer-mediated communication systems such as video conferencing. Another indication is that it is usually not socially acceptable to wear sunglasses indoors. Lastly, and with the exception of people already used to wearing glasses, most users prefer not to wear gear on their head.

Auto-stereoscopic displays generate different images that can be seen from different viewpoints by redirecting the light emitted by pixels on the screen in selected directions. In this way, different images can be created for the two eyes of a human. This is typically achieved via a lenticular screen in front of the actual display. Most technologies require that the user hold their head stable in a relatively small region to achieve a good stereo effect. Some create multiple "sweet spots" to allow for multiple users. As these sweet spots are usually relatively small, this leads to neck strain, prohibiting long durations of use. As such, some of the newest prototypes track the user's eyes and then "aim" images at the user in an active manner.

Image generation for 3D interactive systems normally involve the use of 3D graphics hardware. Great advances in performance and image quality have been achieved and image generation is usually not the bottleneck of 3D user interfaces, unless photorealism is a hard requirement. Hence, we do not discuss image generation for these displays.

2.2.2 Displaying 3D Text

Text never truly left the 2D desktop. It has been used in 3D interfaces, but mostly in labels or icons. One fundamental difference between 2D and 3D interfaces is the orientation of text and the variations in scale. Where 2D text is almost always parallel to the display surface, 3D text is often found at various orientations. This leads to sub-optimally rendered text as well as text smaller than the resolution of the display. Both effects make text significantly less readable.

Words become less readable when they are perspectively distorted or rotated in any direction on the screen. Because of technical limitations in anti-aliasing methods, 3D text is normally rendered sub-optimally. These methods blur the content and hence decrease readability. For angles less than about 60 degrees there is only a relatively small decrease in reading speed, which can be compensated by magnifying the text proportionally. For rotations larger than about 60 degrees there is a sharp decrease in readability, even with optimal anti-aliasing methods [36]. This effect has even been verified in 3D displays [27].

Much more important however, is that perspective distortion causes large parts of the characters in a window to become extremely small – often smaller than a pixel. Imagine a page of text on a screen, rotated around the vertical axis by 45 degrees so that the left side of the page is closer to the viewer. Then the beginning of each line is easily readable, but the text at the end of each line is practically guaranteed to be too small to be readable as the resolution of the screen is not sufficient. Hence it is not realistic to expect longer text to be readable in 3D unless it directly faces the user. Applications that display rotated text can hence really only use text for mnemonic or iconic reminders for the original content. In summary, and as information density is critical for many applications and unless significant increases in screen resolution occur, 3D text will continue to be problematic.

2.3 Human Issues

In this section we first mention "low-level" issues, i.e. motor skills and perceptual issues, and then discuss issues that are based on cognitive capabilities.

2.3.1 "Low-level" Issues

Sensitivity to latency or lag is a property of the human "system" that affects both input and output devices simultaneously. Any non-trivial delay in the handling of movements, regardless if it is in the tracking system, the VR simulation, or on the display side, has negative effects on human performance [22] and presence [40]. This applies both to head as well as hand movements. The negative effects of lag and variations in lag for head movements are well documented in Virtual Reality research and are believed to be one of the main causes of cybersickness, see e.g. [37]. Beyond this, we highlight in this document the effect of lag on human manipulation performance, a topic that is well known in 2D user interface research [38], but has received only recently attention in 3D user interfaces. For manipulation, measurements have shown that even delays as small as 16 milliseconds can affect performance adversely [22]. Systematic studies of the effects of latency and (spatial) jitter/noise show clearly that they have a negative effect on human performance [48,62]. While humans are able to sense constant latency, they are able to adapt to it to some degree, but still rate it negatively. However, any substantial *variation* in latency usually has disastrous effects for manipulation [22,66].

As for 3D manipulation, humans are good at manipulating an object in 6DOF if they can grab it up close, make use of small and large movements, and are able to pair the manipulation with proprioceptive cues, i.e feedback about the position of their limbs and the forces applied to them. Consider the task of plugging an ill-fitting electric plug into a wall-socket. The feedback provided by bumping into the socket is picked up by the fingers and responded to by fine-grained manipulations to guide the plug down the slopes of the socket. This also involves knowing how much force to apply, and that when too much resistance is encountered, the plug may be upside-down. Consider the difficulty orienting the plug properly without finger manipulations and/or without a second hand to assist. Consider how, without two hands, reorienting the plug would require awkward “clutching” movements (releasing, repositioning, and re-grabbing). Contrast this with typical 3D interfaces where users manipulate objects by a single tracked *point* or a long *ray* extending from their hands. Another related fact is that depth perception of humans is relatively less accurate compared to the accuracy across the visual field [68]. Moreover, if a contact surface is available, humans can leverage it to greatly simplify manipulation [53]. In summary, humans are not necessarily as proficient in full 6DOF manipulation tasks as many believe, see also below. They are just good at reacting to feedback and the use of the highly specialized sensing and actions of the body.

Fatigue and/or hand tremor is another problem that affects performance with 3D input devices. Devices held away from the user’s body, such as 3D wands, 3D gloves and similar devices, cause fatigue. People are not designed to holding their hands in the air for extended periods of time, without some form of support. As an exercise, try extending your arm straight out to the side for a minute or two and you will quickly find how fatiguing this is. As well, we ask the reader to reflect on how many real-world professions exist that require this. One of the few examples is a conductor, but even they drop their hands to their sides as often as possible. Regarding hand tremor, it is hard to hold a hand at a constant location in space if there is nothing to position the hand against. Many professions address this by using various forms of support. One good 3D example is a sculptor, who uses the surface of the object itself to stabilize their hands and tools before modifying the object.

Humans also prefer strongly to interact with objects that they can see directly. If something is invisible, people will either rotate the object or move themselves to see their focus of attention before working on it. The way a plumber works is a good example here. In other words, we argue that manipulation of invisible objects is the exception, not the rule.

2.3.2 3D Cognition

As far as cognition is concerned, we point out that humans are not “naturally” proficient at full 3D navigation. Most human environments are not fully 3D, nor do they require full 6DOF navigation as people constrain themselves to 4DOF, i.e. walking in the plane and looking around. Tilting the head is unusual and changing the height of the viewpoint is usually accommodated with a complete change of posture. People in “full” 3D professions, such as astronauts, divers, and fighter pilots, usually need extensive training (hundreds to thousands of hours) to do their job. Astronauts also need training because they work in an environment without gravity, and they have to “un-learn” their reliance on gravity. One profession that uses limited 6DOF navigation is a plumber, who contorts his body to see under a sink or in a tight space – but many people prefer not to do this. Lastly, consider that although systems such as Google Earth afford 3D navigation, most people use this only within a very small region. Larger travel is usually handled by “jumping” to a new location, either via search or bookmarks. In other words, people prefer to “teleport” for larger distances rather than navigate.

Moreover, 3D spatial memory is not *that* much better than 2D spatial memory. The main reason for this is that the world is only a restricted 3D environment. Consider that buildings have *numbered* floors, connected by elevators and stairs. Hence, most humans remember the floor number and the 2D location on that floor, but not the spatial location in 3D. Similarly, furniture has drawers or doors that are only accessible from the front, which forms again a 1D or 2D indexing system. And objects are organized inside the drawers to simplify access, too – very frequently in a 1D or 2D layout. Hence, most people are not trained to fully utilize 3D spatial memory, as the world around them doesn’t require it. Another indication for this is that experiments comparing information retrieval times across 2D, 2½D, and 3D interfaces showed that 3D interfaces were the slowest alternative, regardless if computers were involved or not [16]!

Last, but not least, we have to consider how “natural” user interface mechanisms need to be, see e.g. [55]. Consider for example that engineers need training to understand wireframe views or orthogonal projections. In other words, such displays are not appropriate for the average person. Or consider that 3D handles that move objects along the coordinate system axes or planes [15,58] require that the user has an understanding of the concept of local and global coordinate systems – something that again needs training for most people. Finally, many computer-aided design systems offer manipulation methods that are a one-to-one mapping of the underlying mathematics or a very thin layer above it. Then the user needs to understand the mathematics to be able to use such a system effectively, which is often not practical. Consider, for example, how difficult it is to put a particular kind of crease into a NURBS surface in current CAD systems.

After having established the above list of challenges, it becomes easier to see why certain 3D user interfaces are more successful than others. In the following section we put forth guidelines that encapsulate the most important lessons learned.

3 Guidelines for Constraining 3D Interaction

Three-dimensional user interfaces have not fully matured despite years of research [7]. Part of the problem is that 3D hardware technologies are still too immature to set up and keep running on a daily basis without incurring significant overhead [57]. Another problem is that many user interface techniques are implemented as a thin layer on top of the mathematical foundations. A good example is the use of handles to constrain movement along one of the three major coordinate axes or on a plane, see e.g. [15,58]. Consequently, only users who understand the underlying concepts, such as local coordinate frames in tilted surfaces, can effectively use such a system. Hence, naïve people cannot quickly interact with and change 3D content – all novices can do is "experience" a largely static world [14] in Virtual Reality systems. This is a primary barrier to broad acceptance.

In contrast, many 3D games and online virtual worlds offer easy access to 3D content. Most people adapt quickly to the way such systems afford interaction with 3D worlds. The content editor in the Spore game, also known as Spore Creator [21], is a good example as it enables even naïve users to perform a large range of 3D operations. To illustrate the difference, we encourage readers to compare this content creator with traditional CAD tools targeted at the same purpose. There is no fundamental reason why traditional CAD tools cannot adopt such an interface to simplify common operations. Moreover, most successful games and virtual worlds use essentially only 2D input for interaction, which involves the additional overhead of finding a good mapping of 2D interaction to the 3D scene depicted on the screen. Driven by market forces, a large number of games share the same fundamental user interface paradigms, which in turn encourages re-use of skills across games. A similar evolution is happening for online 3D worlds.

Here we present ten guidelines. They are based on the issues identified above, but are also based on knowledge present in the community of 3D games and online 3D worlds. Others are based on results of user studies with novice participants, i.e. persons without VR knowledge, or research in VR, perception, kinesiology, and 2D GUIs. These guidelines will help drive 3D UIs toward broader accessibility and will form a basis for the next generation of 3D UI techniques. The order of these guidelines corresponds largely to the sequence of issues identified above.

3.1 2D Input Devices Are Advantageous

Input devices such as the Personal Interaction Panel, which use a pen on a 2D tablet to provide interaction in a VR system, have been shown effective for 3D worlds [60]. Also, constraining the input to 2D reduces hand fatigue and provides more accuracy. While it may be possible to do symbolic input with a Wiimote or other accelerometer-based systems, and has been done with 3D trackers [13], such approaches are not optimal and should be reserved for special cases such as short text input.

Moreover, a comparison of input device specifications between mouse- or pen-based systems and 3D technologies reveals that 2D technologies are one to two orders of magnitude more precise and have much less latency [61,62]. This research also shows initial evidence that these technological differences are one of the main reasons why 2D input devices outperform 3D input devices for tasks that require only 2D motion and even for 3D tasks [6]. Combinations between 3D tracking and an interactive tablet with a tracked pen are a sensible approach.

3.2 Perspective and Occlusion Are the Most Appropriate Depth Cues

Motion parallax, either induced through self-motion or through moving objects, is the strongest depth cue [68]. However, most 3D user interfaces that afford interaction rely on a quasi-static viewpoint, as it is hard to manipulate objects with precision while in motion. Also, manipulating an object that is moving is similarly hard. Hence, most 3D user interfaces permit only interaction in a quasi-static view and scene or at least indirectly encourage a static view position.

In that situation, and for manipulation of objects beyond arm's length, perspective and occlusion are the strongest depth cues [68]. Assuming that there are no floating objects and sufficient texture is available, these two cues are usually sufficient to accurately and quickly judge an object's 3D position in an environment, unless optical illusions are involved. Although stereo display is valuable, it matters only for objects fairly close to the viewer [68]. Given that most 3D systems target large spaces, stereo display does not provide a clear value for 3D user interfaces. Last, but not least, stereo technologies are far from mature and are tiresome or problematic if used daily [20,65].

3.3 Interact Only with Visible Objects

Users interact with what they see. As such, they prefer to navigate so as to see or better see objects before interacting with them [49,64]. This is especially important when the 3D environment has no tactile feedback. There are several consequences of this guideline. First, it points to the importance of easy navigation. Second, because a 2D manifold can fully describe the set of all visible objects, 2D input is sufficient to select an object. This is also documented by the success of ray-casting and occlusion based techniques relative to point-based virtual hand techniques [8,50]. This also means that 2D input devices are sufficient to select objects in a 3D world – assuming that adequate 3D navigation techniques exist. Practically all current 3D games use this to simplify the interaction with the content.

3.4 People See the Object, Not the Cursor

Research into primate vision has demonstrated that monkeys attend visually to not only the tip of a tool in their hand but also the whole tool and the hand. This indicates that a cursor might not be the best choice for 3D UIs – a cursor is effectively a point, while an object covers an area in the visual field. The sliding technique introduced in the SESAME (Sketch, Extrude, Sculpt, and Manipulate Easily) system analyzes the visual-area overlap between the manipulated object and the static scene to determine a moving object's position. The associated user studies demonstrate that users can easily use and learn such techniques and that such methods provide clear performance benefits [44].

3.5 Floating Objects Are The Exception

In the real world, few floating objects exist, and almost all objects are attached to other objects. However, the default in most 3D systems is that every object floats. In the real world, gravity ensures that objects float only for short periods of time, unless they are attached to something else. Hence, and to leverage this experience from the real world, the better default for a 3D system is for objects to always attach to other objects in normal operation. There are clear performance benefits to this [54], as also documented through the interaction possibilities in most games. User interfaces can provide *secondary* user interface mechanisms to make objects stay in midair for the exceptional cases where this is warranted.

3.6 Objects Don't Interpenetrate

Solid objects – including the viewers themselves – can't interpenetrate each other. Humans are used to this and deal with it every day. However, many VR systems allow object interpenetration by default. Interpenetration leads to confusing visual display, and many novice users cannot easily recover from such situations. For example, consider the negative effect of users being "trapped" behind a wall in a game or a small object disappearing inside a larger – most novices need help to recover from such a situation. Real-time effective collision detection and avoidance for large environments is currently possible with the help of graphics hardware [19,25]. As an added benefit, collision detection and avoidance enables sliding contact, an efficient way to position objects in the real world [33]. These effects are frequently used in games to simplify the user interface.

3.7 2D and 2½D Tasks Are Simpler Than 3D

Most real-world tasks aren't fully 3D; they are 2D or 2½D, as the real world is often a subset of full 3D. For example, blueprints of buildings abstract the height dimension so as to better focus on 2D spatial relationships. Multistory buildings are layers of 2D floor plans. When needed, crosscuts show alternate dimensions or perspective drawings show 3D. Real 3D structures in buildings exist, but they are again the exception, not the rule. Consequently, most humans are used to dealing with 2D or 2½D and don't have the skills necessary to deal with problems that are fully 3D. There is experimental evidence that underlines this, e.g. [16].

Another example is the way stacks of objects, such as paper, clothes, cards, are handled. People quickly learn that one can't just pull an object out of a stack. Instead one has to lift the top of the stack away to reveal the desired object then work with that object and finally reassemble the stack. One example for a 3D UI that exploits this is the SESAME system, which analyzes the scene structure to afford quick and easy manipulation of such stacks [46]. Related to this are techniques for the easy manipulation of common object groups such as cabinets or chairs [43,59]. Hence, offering 2D methods to achieve most tasks is an excellent way to increase usability for 3D user interfaces.

3.8 Constrained Navigation And Rapid Transportation Is Good

In the real world, navigation rarely requires unconstrained manipulation of all 6 DOFs. And all professions that (can potentially) use true 6DOF navigation, such as fighter pilots, night and wreck divers, and astronauts, require large amounts of training. Furthermore, physics limits even a fighter plane to essentially 4DOFs of freedom in navigation. Helicopter pilots can access more degrees of freedom simultaneously, but require even *more* training. In general, most navigational tasks have 4 or less DOF's, a fact that can and should be used to simplify the user interface, as this makes navigation much more accessible.

As navigation for larger distances is cumbersome, many systems provide a means of instant transportation to different locations. This is usually associated with a search feature that allows users to specify a name for a location. One issue with teleportation is that users may become disoriented [9], and as such, cues to assist user's understanding of orientation should be provided. A reasonable mechanism is to provide an overview/radar view that highlights the users' current position in the larger environment or an animation transferring the user into a new position, as introduced in the World-In-Miniature technique [47].

3.9 Full 3D Rotations Aren't Always Necessary

Many common objects, such as chairs, desks, and shelves, have a clear "up" orientation. Other objects, such as hanging lamps and whiteboards, also have clear orientations. These objects are all attached to other objects. This attachment provides appropriate constraints for rotation – a chair is on its side only in exceptional cases. Consequently, providing a simple user interface to rotate an object around the axis afforded by that object's main attachment is a good design alternative for easy-to-use systems [56]. Although a 3D UI should support full 3D rotations, this option should not be the primary mode as full 3D motions are best delegated to secondary user interface mechanisms.

3.10 Reality Simulation Isn't Always Appropriate

One option for 3D user interfaces is to simulate reality more or less completely. However, besides being technically challenging, this is not appropriate for many applications. Consider e.g. an object being bumped off a table and rolling under a cupboard, or even breaking upon impact. Retrieving or repairing that object is cumbersome and not necessary in a 3D user interface – unless the application focus is on the retrieval task. Additionally, the more realistic the environment, the more users expect of it. Then, if the interaction fails to live up to expectations, they become frustrated. Hence, we suggest that reality be simulated as far as necessary to afford good skill transfer from a user's previous experience and easy manipulation, but not necessarily further.

4 Conclusions and Directions for Future Work

The next generation of 3D UIs can greatly benefit from user interface techniques that are adapted to how humans perceive and interact with the real world. Moreover, novel 3D UIs should leverage the strengths of humans and existing technologies – for both input and output – as far as possible and avoid known weaknesses. This will maximize the chances for skill transfer, thus increasing the usability of all developed techniques. This will lead to better 3D applications, a broader range of applications that use 3D productively, and increased adoption of 3D UIs.

In the following list, we target the main 3D user interface application domains with specific advice. Note that this is general advice that applies to the field as a whole, not necessarily to individual systems. As a disclaimer, we state that some of the advice listed is not necessarily always backed up fully by scientific inquiry, and we do not expect all of the items to stand the test of time.

4.1 Games

Most 3D games already include simple-to-use 3D user interfaces that follow directly or indirectly many of the above-mentioned guidelines. One challenge that we would like to pose to this community is to push the interactivity of games further, in the sense that in many games most of the environment is quite static and cannot be interacted with, or has only limited interaction possibilities. Pushing this limit will enable new kinds of game paradigms, as evidenced e.g. by the Spore content creator. Another boundary that is already being explored is new kinds of interaction devices, as evidenced by the Wiimote.

4.2 Virtual Reality

Most traditional VR systems employ a user interface based on the wand-in-hand paradigm and with stereo displays. Depending on the application area it may be worthwhile to revisit these decisions, as there are alternatives that necessitate less training and offer much better usability. E.g. is stereo really necessary or helpful for the application area? Would an interactive tablet tracked in 3D offer a simpler user interface for the domain and also afford more efficient interaction at the same time? Another direction to explore is haptic interfaces. However, only systems that can track both hands and fingers simultaneously with high accuracy can expect to benefit from effective skill transfer from human experience in 3D manipulation of objects.

4.3 Augmented Reality

With the transition to hand-held devices the user interface needs for this field have changed radically. This is visible by the fact that most AR systems are "view-only", i.e. not fully interactive. However, interactive manipulation of the content in a "live" setting is exactly one of the areas where AR systems can distinguish themselves from other approaches!

4.4 3D Desktops

On the one hand there are many 2D desktop windowing systems that have recently added 3D effects to increase visual attractiveness. This kind of pseudo-3D system has no real need for a 3D user interface, except to deal with the "stacking" of windows that occurs naturally in these systems. In this context it is interesting to point out that this "stacking" of 2D windows makes the normal desktop windowing system already a 2½D environment! On the other hand there are the "real" 3D desktop windowing systems that allow traditional 2D windows to be rotated and moved in 3D. Given that the readability of textual content suffers very significantly by this, we suggest that either live "icon" previews or similar thumbnails be considered – they may well offer all the benefits for a smaller price in terms of usability. Interaction with content in windows that are perspectively distorted and/or rotated is not a good idea in general. Finally, one of the drawbacks of 3D desktops is that the user needs to spend more effort on navigation and on the landmarks that aid that navigation, which may well cancel any benefits gained through the transition to a 3D world [16]. The concept of virtual desktop managers/spaces is a competitive concept that seems to have higher end-user acceptance, yet still leaves room for quasi-3D effects during transitions.

4.5 Computer Aided Design

In general, this class of systems can benefit greatly from a general refresh of the underlying assumptions and defaults. Google SketchUp is a great example of such a refresh. We believe that the additional introduction of a contact assumption will very likely improve manipulation performance for the most common interactions in practically all CAD systems. The work on SESAME [45] points to the potential gains. Clearly, CAD systems will have to enable the user to create floating objects. However, this should not be the default and should only be possible through secondary user interface mechanisms. It is far more efficient to provide primary user interface techniques that directly support and maintain the more common case of objects in contact.

5 Acknowledgements

Many thanks to the reviewers for the suggestions that helped greatly to refine the structure of this work.

6 References

[1] Agilent ADNS-3080 Optical Mouse Sensor datasheet, Agilent Technologies, (2005), http://www.alldatasheet.com/datasheet-pdf/pdf/193358/HP/ADNS-3080.html. Cited 13 April 2010.

[2] A. S. Arif, W. Stuerzlinger, *Predicting the Cost of Error Correction in Character-Based Text Entry Technologies*, In ACM CHI 2010, (April 2010).

[3] R. Arsenault, C. Ware, *Frustum View Angle*, Observer View Angle and VE Navigation, Proceedings of SVR, (2002).

[4] R. Azuma, *A Survey of Augmented Reality*. In Presence: Teleoperators and Virtual Environments 6, 4 (August 1997), pp. 355-385.

[5] J. Bakdash, J. Augustyn, D. Proffitt, *Large displays enhance spatial knowledge of a virtual environment*. In Symposium on Applied Perception in Graphics and Visualization, (2006), pp. 59-62.

[6] F. Bérard, J. Ip, M. Benovoy, D. El-Shimy, J. Blum, J. Cooperstock, Did "Minority Report" Get It Wrong? Superiority of the Mouse over 3D Input Devices in a 3D Placement Task. In Interact 2009, pp. 400-414.

[7] D. Bowman, J. Chen, C. Wingrave, et al., *New Directions in 3D User Interfaces*, In International Journal of Virtual Reality, 5(2), (2006), pp. 3-14.

[8] D. Bowman, D. Johnson, L. Hodges, *Testbed Evaluation of Virtual Environment Interaction Techniques*, In Virtual Reality Software and Technology, (1999), pp. 26-33.

[9] D. Bowman, D. Koller, L. Hodges, *Travel in Immersive Virtual Environments: An Evaluation of Viewpoint Motion Control Techniques*, In VRAIS, (1997), pp. 45-52.

[10] D. Bowman, E. Kruijff, J. LaViola, I. Poupyrev, 3D User Interfaces: Theory and Practice. (Addison-Wesley, 2005).

[11] D. Bowman, R. McMahan, *Virtual Reality: How Much Immersion is Enough?*, In IEEE Computer, 40(7), (2007), pp. 36-43.

[12] D. Bowman, C. Wingrave, *Design and Evaluation of Menu Systems for Immersive Virtual Environments*. In Proceedings of IEEE Virtual Reality, 2001, pp. 149-156.

[13] D. Bowman, C. Wingrave, J. Campbell, V. Ly, *Using Pinch Gloves™ for both Natural and Abstract Interaction Techniques in Virtual Environments*. In Proceedings of HCI International, 2001, pp. 629-633.

[14] F. Brooks, *What's Real About Virtual Reality?*, In IEEE Computer Graphics & Applications, 19(6), (Nov. 1999), pp. 16-27.

[15] J. Chen, D. Bowman, J. Lucas, C. Wingrave, *Interfaces for Cloning in Immersive Virtual Environments*, In Eurographics Workshop on Virtual Environments, (2004), pp. 91-98.

[16] A Cockburn, B McKenzie, *Evaluating Spatial Memory in Two and Three Dimensions*. In International Journal of Human-Computer Studies, 61(30), (2004), pp. 359-373.

[17] C. Cruz-Neira, J. Sandm, T. A. DeFanti, *Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE*, In Computer Graphics, ACM SIGGRAPH, (August 1993), pp. 135-142.

[18] M. Czerwinski, D. Tan, G. Robertson, *Women take a wider view*. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Changing Our World, Changing Ourselves, (2002), pp. 195-202.

[19] K. Dave, K. Dinesh, *CInDeR: Collision and Interference Detection in Real-time using graphics hardware*, In Graphics Interface, (2003), pp. 73-80.

[20] D. Diner, D. Fender, *Human Engineering in Stereoscopic Viewing Devices*. (Plenum Press, New York, 1993).

[21] Electronic Arts, Spore, http://www.spore.com/, Cited 4 April 2010.

[22] S. Ellis, M. Young, B. Adelstein, S. Ehrlich, *Discrimination of Changes of Latency During Voluntary Hand Movement of Virtual Objects*, In Human Factors and Ergonomics Society,

(1999), pp. 1129-1136

[23] B. Fröhlich, J. Plate, *The cubic mouse: a new device for three-dimensional input*, In ACM CHI, (2000), pp. 526-531.

[24] Google, Google SketchUp, http://sketchup.google.com/. Cited 4 April 2010.

[25] N. Govindaraju, S. Redon, M. Lin, D. Manocha, *Cullide: Interactive Collision Detection between Complex Models in Large Environments Using Graphics Hardware*, In SIGGRAPH Workshop on Graphics Hardware, (2003), pp. 25–32.

[26] T. Grossman, R. Balakrishnan, *An evaluation of depth perception on volumetric displays*, In AVI, (2006), pp. 193-200.

[27] T. Grossman, D. Wigdor, R. Balakrishnan, *Exploring and reducing the effects of orientation on text readability in volumetric displays*, In ACM CHI, (2007), pp. 483-492.

[28] K. Herndon, A. Van Dam, M. Gleicher, *Workshop on the Challenges of 3D Interaction*, In SIGCHI Bulletin, 26(4), (1994), pp. 1-9.

[29] K. Hinckley, R. Paush, J. Goble, N. Kassell, *Passive Real-World Interface Props for Neurosurgical Visualization*, In CHI, (1994), pp. 452-458.

[30] M. Hoffman, P. Varcholik, J. LaViola, *Breaking the Status Quo: Improving 3D Gesture Recognition with Spatially Convenient Input Devices*, In Proceedings of IEEE Virtual Reality, (March, 2010), pp. 59-66.

[31] R.J.K. Jacob, A. Girouard, L.M. Hirshfield, M.S. Horn, O. Shaer, E.T. Solovey, J. Zigelbaum, *Reality-Based Interaction: A Framework for Post-WIMP Interfaces*, In CHI, (2008), pp. 201-210.

[32] A. Jones, I. McDowall, H. Yamada, M. Bolas, P. Debevec, *Rendering for an interactive 360° light field display*, In ACM SIGGRAPH, (2007), article no. 40.

[33] Y. Kitamura, F. Kishino, *Consolidated Manipulation of Virtual and Real Objects*, In VRST, (1997), pp. 133–138.

[34] G. Kessler, L.Hodges, N. Walker, *Evaluation of a Whole-Hand Input Device*, In ACM Transactions on Computer- Human Interaction, 2(4) (Dec. 1995), pp. 263-283.

[35] A. Kulik, *Building on Realism and Magic for Designing 3D Interaction Techniques*, In IEEE Computer Graphics & Applications, 29(6), (November/December 2009), pp. 22-33.

[36] K. Larson, M. van Dantzich, M. Czerwinski, G. Robertson, *Text in 3D: some legibility results*, In ACM CHI Extended Abstracts, (2000), pp. 145-146.

[37] J. LaViola, *A Discussion of Cybersickness in Virtual Environments*, In SIGCHI Bulletin 32(1), (Jan. 2000), pp. 47-56.

[38] I. S. MacKenzie, C. Ware, *Lag as a determinant of human performance in interactive systems*, In ACM CHI, (1993), pp. 488-493.

[39] J. Marbach, *Image Blending and View Clustering for Multi-Viewer Immersive Projection Environments*, In IEEE Virtual Reality, (2009), pp. 51-54.

[40] M. S. Meehan, S. Razzaque, M. Whitton, F. Brooks, *Effects of Latency on Presence in Stressful Virtual Environments*, In IEEE Virtual Reality, (2003), pp. 141-148.

[41] B. Myers, *Why are Human-Computer Interfaces Difficult to Design and Implement?*, In Technical Report CMU-CS-93-183, (1993).

[42] T. Ni, G. Schmidt, O. Staadt, M. Livingston, R. Ball, R. May, *A survey of large high-resolution display technologies, techniques, and applications*, In IEEE Virtual Reality,

(2006), pp. 223-236.

[43] J.-Y. Oh, W. Sturzlinger, *Intelligent Manipulation Techniques for Conceptual 3D Design*, In Human Computer Interaction: Interact, (2003), pp. 319-326.

[44] J.-Y. Oh, W.Stuerzlinger, *Moving Objects with 2D Input Devices in CAD Systems and Desktop Virtual Environments*, In Graphics Interface, (2005), pp. 195–202.

[45] J.-Y. Oh, W. Stuerzlinger, J. Danahy, *SESAME: Towards Better 3D Conceptual Design Systems*, In ACM Designing Interactive Systems, (2006), pp. 80-89.

[46] J.-Y. Oh, W. Stuerzlinger, D. Dadgari, *Group Selection Techniques for Efficient 3D Modeling*, In IEEE Symposium on 3D User Interfaces, (2006), pp. 95-102.

[47] R. Pausch, T. Burnette, D. Brockway, M. Weiblen, *Navigation and locomotion in virtual worlds via flight into hand-held miniatures*, In SIGGRAPH, (1995), pp. 399-400.

[48] A. Pavlovych, W. Stuerzlinger, The Tradeoff between Spatial Jitter and Latency in Pointing Tasks, ACM Symposium on Engineering Interactive Computing Systems 2009, 187-196.

[49] C. Phillips, J. Granieri, N. Badler, *Automatic viewing control for 3D direct manipulation*, In Symposium on Interactive 3D Graphics, (1992), pp. 71-74.

[50] I. Poupyrev, S. Weghorst, M. Billinghurst, T. Ichikawa, *Egocentric Object Manipulation in Virtual Environments: Empirical Evaluation of Interaction Techniques*, In Siggraph/Eurographics Workshop Graphics Hardware, (1998), pp. 41–52.

[51] R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, H. Fuchs, *The office of the future: A unified approach to image-based modeling and spatially immersive displays*. In SIGGRAPH, (1998), pp. 179-188.

[52] B. Reeves, A. Lang, E. Y. Kim, D. Tatar, *The Effects of Screen Size and Viewer Contents on Attention and Arousal*. In Journal of Media Psychology, 1(1), (1999), pp. 49-67.

[53] G. Robles-De-La-Torre, *The Importance of the Sense of Touch in Virtual and Real Environments*, In Special issue on Haptic User Interfaces for Multimedia Systems, IEEE Multimedia, 13(3), (2006), pp. 24-30.

[54] T. Salzman, S. Stachniak, W. Sturzlinger, *Unconstrained vs. Constrained 3D Scene Manipulation*, In Engineering for Human-Computer Interaction, (2001), pp. 207-219.

[55] B. Shneiderman, *Why Not Make Interfaces Better than 3D Reality?*, In IEEE Computer Graphics and Applications, 23(6), (Nov./Dec. 2003), pp. 12-15.

[56] G. Smith, T. Salzman, W. Stuerzlinger, *3D Scene Manipulation with 2D Devices and Constraints*, In Graphics Interface, (2001), pp. 135–142.

[57] A. Steed, *Some Useful Abstractions for Re-Usable Virtual Environment Platforms*, In IEEE Virtual Reality Workshop for Software Engineering and Architectures for Realtime Interactive Systems, (2008).

[58] P. Strauss, R. Carey, *An object-oriented 3D graphics toolkit*, In SIGGRAPH, (1992), pp. 341-349.

[59] W. Stuerzlinger, G. Smith, *Efficient Manipulation of Object Groups in Virtual Environments*, In IEEE VR, (2002), pp. 251-258.

[60] Z. Szalavári, M. Gervautz, *The Personal Interaction Panel—a Two-Handed Interface for Augmented Reality*, In Computer Graphics Forum, 16(3), (1997), pp. 335-346.

[61] R. Teather, W. Stuerzlinger, *Assessing the Effects of Orientation and Device on (Constrained) 3D Movement Techniques*, In IEEE Symposium on 3D User Interfaces (2008), pp. 43–50.

[62] R. Teather, A. Pavlovych, W. Stuerzlinger, S. MacKenzie, *Effects of tracking technology, latency, and spatial jitter on object movement*, In IEEE Symposium on 3D User Interfaces (2009), pp. 43-50.

[63] A. van Dam, *Post-WIMP user interfaces*. Commun. ACM, 40(2), (Feb. 1997), pp. 63-67.

[64] C. Ware, K. Lowther, *Selection Using a One-Eyed Cursor in a Fish Tank VR Environment*, In ACM Transactions on Computer-Human Interaction, 4(4), (1997), pp. 309–322.

[65] Z. Wartell, L.F. Hodges, W. Ribarsky, *A Geometric Comparison of Algorithms for Fusion Control in Stereoscopic HTDs*, In IEEE Trans. Visualization & Computer Graphics, 8(2), (2002), pp.129–143.

[66] B. Watson, *The Effects of Variation of System Responsiveness on User Performance in Virtual Environments*, Human Factors, 40, (1998), pp. 403-414.

[67] G. Welch, E. Foxlin, *Motion Tracking: No Silver Bullet, but a Respectable Arsenal*, In IEEE Computer Graphics and Applications, 22(6), (2002), pp. 24-38.

[68] C. Wickens, J. Hollands, Engineering Psychology and Human Performance, 3rd ed., (Prentice-Hall, 1999).

[69] C. Wingrave, J. LaViola, *Reflection on the Design and Implementation of Virtual Environments*. In Special Issue of Presence: Teleoperators and Virtual Environments: Reflections on the Design and Implementation of Virtual Environment Systems, 19(2), (2010).

[70] C. Wingrave, B. Williamson, P. Varcholik, J. Rose, A. Miller, E. Charbonneau, J. Bott, J. LaViola, *Wii Remote and Beyond: Using Spatially Convenient Devices for 3D UIs*. In IEEE Computer Graphics and Applications, 30(2), 2010, pp. 71-85,.