Guidelines for 3D Positioning Techniques

Robert J. Teather
rteather@cse.yorku.ca

Wolfgang Stuerzlinger
wolfgang@cse.yorku.ca

Department of Computer Science and Engineering
York University
Toronto, Ontario

ABSTRACT
In this paper, we present a set of guidelines for designing 3D positioning techniques. These guidelines are intended for developers of object interaction schemes in 3D games, modeling packages, computer aided design systems, and virtual environments. The guidelines promote intuitive object movement techniques in these types of environments.

We then present a study comparing 3D positioning techniques based on these guidelines with 2D and 3D/6D devices across VR display technologies. Display technologies such as stereoscopic graphics and head-coupled perspective provide additional depth cues and could affect how a user perceives and thus interacts with a 3D scene – regardless of the input device/technique used. Thus they are examined as well. The results suggest that 2D devices using "smart" movement algorithms can outperform 3D devices.

Categories and Subject Descriptors
H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems – *virtual reality.* H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – *input devices, interaction styles, standardization, theory and methods.*

General Terms
Experimentation, Human Factors.

Keywords
3D object positioning, guidelines.

1. INTRODUCTION
To this day very few, if any, games support the kind of full-featured 3D object manipulation that many naïve users believe to be possible in VR (virtual reality) environments. This is at least in part due to the lack of suitable input devices as well as the lack of intuitive interaction techniques for these devices. Manipulating 3D objects requires the handling of 6 degrees of freedom (DOFs), i.e. there are 3 axes of movement and 3 axes of rotation for every object. A large body of VR research focuses on using 3D input devices such as 6DOF trackers and wands for 3D manipulation tasks. The motivation for this is that they allow the user to simultaneously position and orient a virtual object, and thus provide the most efficient manipulation interface compared to input devices that control less DOFs.

However, most users are extensively familiar with 2D input devices, in particular the mouse. Furthermore, practically all commercially successful 3D graphics systems (including 3D modeling packages and computer games) are based on a mouse-based user interface. Using a mouse for 3D interaction introduces the problem of mapping 2D mouse motions into 3D operations. While several solutions have been proposed, all of them require that users mentally translate 2D mouse movements into low-level 3D operations, which is unsuitable for naïve users. However, there is evidence that 2D input devices can outperform 3D devices for certain 3D positioning tasks, through the use of software techniques that map mouse movement to intuitive 3D object movement. Based on this observation, we present a list of guidelines for developing intuitive 3D manipulation techniques that can be used both in games and VR environments when using a mouse (or other 2D devices) for input.

Based on these guidelines, we developed and compared three object manipulation techniques with different input devices in a fish tank VR environment. Additionally, because fish tank VR typically uses stereoscopic graphics and head-coupled perspective, each of these interaction techniques was tested in several display modes to assess possible interactions between display mode and input technique.

1.1 Related Work
Most VR systems include the ability to modify or move objects with various techniques using different input devices. We first discuss 3D manipulation work with 3/6DOF devices and then with 2DOF devices. For the latter category we also look at stereoscopic graphics and/or head tracking, as there are few, if any, studies that investigate the effect of these visualization techniques on 3D object manipulation tasks with 2D devices. We also summarize related studies that examine the general benefits of stereoscopic graphics, and particularly those that aim to quantify the benefits of stereo on 3D interaction tasks.

Two previous works present taxonomies of 3D selection/manipulation techniques [4, 18]. Poupyrev et al. compared selection and manipulation with ray-casting and a virtual hand metaphor [18]. They found that there was no clear winner – each technique tested had advantages and disadvantages, depending on factors such as distance to the target, object size and visual feedback. Bowman et al. presented a study that compared several techniques created from basic 3D interaction components, and evaluates them in a selection and manipulation test-bed [4]. They found that selection based on ray-casting and occlusion was significantly faster than selection techniques requiring 3D hand/cursor movement. For manipulation, they found that the degrees of freedom of the manipulation task had a significant effect on task completion time. In fact, they note that it dominated the results, with 2DOF techniques significantly outperforming 6DOF techniques, on average.

Zhai et al. [27] conducted a study of their "silk cursor", a technique utilizing transparency and volumetric selection for 6DOF selection tasks. They compared their semi-transparent volumetric cursor to a wire-frame volumetric cursor, as well as stereo to mono graphics. They found that in addition to significant differences by cursor type, the stereoscopic display significantly improved user speed and accuracy. Their results suggest that both (partial) occlusion and stereopsis are beneficial in depth perception, but using both simultaneously provides an even stronger depth cue.

Boritz and Booth [2, 3] conducted a series of studies on 6DOF input devices for 3D interaction tasks. They first studied the use of 6DOF input devices for selection tasks [2]. In their study, they compared stereoscopic to monoscopic display with and without head tracking, as well as different target positions. Their second study also considered orientation of the target [3], requiring users to dock a cursor with a target, matching both position and orientation. Both studies showed that stereo viewing was significantly better than mono, allowing quicker task completion, but no significant effect was found for head tracking. The authors reason that their tasks required only minimal head movement after the initial discovery of target locations. They note that although positional error was reduced in the stereo viewing mode, display mode showed no significant difference between stereoscopic and monoscopic for *rotational* error. It is interesting to note that, with the exception of the docking task in Boritz et al.'s second study [3], *all* studies mentioned above used only 3DOF of the six afforded by the 6DOF input devices used, during manipulation. In all but the docking study, the 6DOF input device was only used for positioning, not orientation. Only Boritz and Booth's second study involved a real 6DOF task.

Other work points out that 2D interface devices work well for 3D interaction when ray casting is used for selection and manipulation [17, 18, 20, 24]. Ware and Lowther conjecture that users rarely wish to interact with totally occluded objects, and as ray-casting allows the user to pick any (even only partially) visible object [24] this is sufficient. It is interesting to point out in this context that a 2D image of a 3D scene is already fully representative of all visible objects in that scene. Ware and Lowther's study found that a ray-casting based 2D selection technique using a cursor rendered to a single eye in a stereo display was more accurate than a 3D selection cursor.

Manipulation is less straightforward than selection, since it is a 6DOF task, and the mouse only affords the simultaneous manipulation of two degrees of freedom. Thus, 2D input must be mapped to 3D operations. One solution used in most modeling and commercial CAD systems is 3D widgets or handles [7, 21], which separate the DOFs by explicitly breaking the manipulation down into its individual components. Small handles are provided for movement along each of the three axes, and for each axis of rotation. This is usually complemented by different simultaneous orthogonal views of the same scene. Bier's skitters and jacks technique [1] provides a similar solution by interactively sliding the 3D cursor over objects in the scene via ray-casting, and attaching a transformation coordinate system to the object where it was positioned. The downside of such manipulation techniques is that users need to mentally decompose every movement into individual operations along the axes of the coordinate system – which don't necessarily align with the axes of the scene. The "simple" solution of allowing users to change the axes for the widgets increases the user interface complexity greatly and carries the potential for the well-known problem of mode errors.

Another approach is to constrain the movement of objects according to physical laws such as gravity and the inability of solid objects to inter-penetrate each other. Such constraints can also be used to limit object movement according to human expectations [20], e.g. chairs sit on the floor, and desk lamps sit on top of desks. However, this approach lacks generality, as it requires object-specific constraints to be designed a priori for each available type of object. For games, constraints may be suitable as they typically support only a limited set of objects in a restricted environment.

A more general approach is based on the observation that in the real world (almost) all objects are attached to other objects and hence remain in contact with other objects at all times [16, 17]. To achieve this, the movement algorithm presented in that work uses the surfaces occluded by the moving object to determine the current movement surface, while still avoiding collisions. An extension allows users to also move objects partially behind other objects. If an object is moved over the background, it moves in free space on a plane orthogonal to the viewer. The result is that the object being moved always slides "over" the remainder of the scene in a very natural, predictable way, consistent with recent results from visual perception research. The algorithm does not use the notion of gravity, i.e. one can move objects from the floor to walls or onto the ceiling and back. For efficiency, most of the computations are performed in graphics hardware.

Some researchers have also considered combinations of 2D and 3D UI components [6, 9, 14, 23]. These approaches either use 2D interface widgets in a virtual environment and allow interaction via 6DOF devices [9], use physical props to constrain 2D interface components to a surface [14, 23] or provide a secondary 2D interface that controls the 3D environment, such as a tablet PC [6]. However, these approaches still involve a very strict separation of 2D and 3D interface components, thus increasing the cognitive overhead for the user.

It is also interesting to note that while a large number of games use a mouse for 3D *navigation* (e.g. Doom3, Half-Life, etc.), few games allow 3D *manipulation* of any degree. On recent exception, Black & White 2 from Lionhead Studios (see http://www.lionhead.com/bw2) allows movement of 3D objects in the game world using the mouse as a metaphorical hand. Clicking objects picks them up and holds them in-hand. The game's physics engine constrains objects to move according to user expectations when objects are released or thrown. However, orientation of objects is seldom, if ever, relevant to the game, and other than rotating the view around an object before grasping it, no facility is provided for rotating objects.

2. GUIDELINES FOR 3D MOVEMENT TECHNIQUES

This paper presents a set of guidelines based on observations from previous work [2, 3, 4, 5, 9, 16, 17, 18, 20] as well as recent research in perception. The intent of these guidelines is to provide suggestions to designers of games and virtual environments for developing intuitive 3D manipulation techniques.

We work under the assumption that a typical 3D manipulation task can be decomposed into the following 3 distinct phases:

1. The selection phase, during which the user indicates which object they intend to manipulate.
2. A positioning phase, where the selected object is brought into the vicinity of the target area.
3. A "fine-tuning" phase, where the object is rotated and positioned relative to the target.

The distinction between the first and second phase is the same as in Bowman et al.'s taxonomy [4]. The third phase is based on the observation that few people, if any, rotate and move the object simultaneously. While experts may rotate and translate an object simultaneously, this is something that novices do not appear to do. We do not believe that further decomposition of these manipulation phases is warranted, at least for novice users. We propose that the *entire* act of positioning an object be handled at once, without requiring the user to think in terms of movement along each of the three separate axes. As 3D rotations introduce a whole new layer of complexity to the problem, we limit ourselves to the 3DOF task of positioning objects in 3D, in the scope of this paper.

In this context, we introduce the following set of guidelines for designing 3D object movement techniques, which also encapsulate the most important design decisions for 3D object movement techniques. Most of these guidelines are based on results of experiments with novice users, i.e. users who have no or limited 3D computer graphics experience. While it may be possible that expert users can achieve higher performance with techniques that ignore these guidelines, we believe that for many kinds of routine scene modifications even expert users will greatly benefit from them.

2.1 Avoid floating objects.

In the real world, (almost) all objects are attached or connected to other objects. Floating objects are exceptional and our experimental observations suggest that most novice users are surprised when an object starts to float when moved. That indicates that the correct default for any 3D object movement technique is that objects should stay in contact with the rest of the world! However, most 3D modeling/CAD systems by default allow objects to "float" in space, which we see as an area ripe for improvement. Solutions to this problem include gravity, contact detection to always keep objects in contact with others, or other similar techniques.

2.2 Objects should not interpenetrate each other.

Many novice users get confused if objects interpenetrate each other, particularly for complex objects, because it is difficult to tell which components belong to what object. Furthermore, they can't easily figure out how to resolve such problems. Incorporating collision detection/avoidance into movement techniques solves this problem. Today, the necessary computations are easily performed in real-time, even for complex scenes (see e.g. [8, 11]). Note that there may also be certain situations where relaxing this guideline may be beneficial. As an example, attempting to insert a peg into a tight hole may actually be easier if the objects *can* pass through one another. However, in general, major collisions should not occur.

2.3 Support relative positioning of objects by bringing them in contact with one another.

The paradigm of sliding an object on the surface of another until it reaches the desired position is a very natural way to position objects. This is easily demonstrated by watching a child position toy blocks. To implement this in a computer system, one must choose a movement surface from the set of surfaces of the static scene and then displace the moving object relative to that surface. One good way to realize this is by using constraints on object movement, see section 1.1. Another option is to ensure that objects always remain in contact with the rest of the scene.

2.4 Only visible objects can be manipulated.

Users typically do not even try to manipulate objects that are not visible. Instead, they tend to rotate or move the viewpoint so that the desired object becomes visible. One indication for this is that previous work found that the most efficient techniques are based on the notion of ray casting [12, 18, 24] or occlusion [4]. Ray casting identifies the first object that is visible along an infinite ray from the manipulation device into the scene. Occlusion is similar, except involves the user blocking the object to be selected with their hand, or another object. Hence, we suggest that it is sufficient to allow the user to select all objects from a 2D image [24], rather than using full 3D cursor selection techniques. And indeed, researchers argue that all ray casting techniques can be approximated as 2D techniques [18]. This is also true of occlusion techniques using an occluding 2D shape as a cursor.

2.5 The most important cues for judging 3D position in real scenes are perspective and occlusion.

As documented by research into visual perception, people judge 3D position based on several depth cues. Besides perspective, the most important cue for 3D position is occlusion [26]. In our previous work, we found that for scenes without floating objects (see 2.1), perspective and occlusion combined with the ability to quickly move the viewpoint are usually sufficient to allow humans to understand the 3D position of an object in relation to other objects. Finally, it is interesting to note that recent research confirmed that from an end-user's point of view, most stereo technologies are not very mature and are tiresome and/or problematic if used on a daily basis [10, 25]. In other words, the addition of stereo viewing to a system does not appear to increase the usability of the system.

2.6 Avoid technical computer graphics techniques such as "handles" and "3 orthogonal views".

Using handles or widgets to move an object in 3D is an instance of an indirect manipulation technique. In the domain of (2D) desktop environments this idea was very rapidly eclipsed by the idea of direct manipulation [19], as this paradigm proved to be much simpler to understand. Furthermore, it has been shown that novice users can manipulate 3D objects *more* effectively in a *single* perspective view and *without* handles when intelligent manipulation techniques are used [17].

2.7 In general, 3DOF or 6DOF input devices provide less precision than 2DOF input devices.

A human hand held in free space will jitter more than a hand that is supported by a physical surface. That means that any input device that is physically limited to 2DOF tends to be more precise and hence usually affords also more efficient manipulation. In VR/AR research, this has been already realized through the adoption of techniques such as the Personal Interaction Panel [23], or physical props [6, 14], which effectively transforms a 6DOF input device into a physical 2DOF input device.

2.8 Use the entire area of visual overlap of the moving object with the static background scene when deciding the position of the object.

Practically all techniques for 3D object motion use only the current position of the cursor to compute the 3D position of a moving object. This effectively reduces the computation to a point mapping problem. However, research into vision in primates discovered that the perceptive field for an object that is being held in the hand covers the *whole* object [15]. In other words, there is strong evidence that the whole visual area of an object is used to judge 3D position. And indeed, previous studies have shown that point-based techniques do not work as well as area-based techniques [17].

3. ISSUES IN COMPARING 3D POSITIONING TECHNIQUES

To assess the value of these guidelines, we designed an experiment to compare how various 3D positioning techniques perform relative to each other. We made the following choices to ensure the validity of our results.

3.1 3D Positioning Techniques

For our initial study, we implemented three positioning techniques, based to varying degrees on the guidelines above. All of them use 2D ray casting for selection of 3D objects. The *first* technique used the mouse, with the assistance of the 3D sliding movement algorithm presented in [17]. This technique is fully based on the above guidelines. In this mode, an object being moved slides on other objects in the scene. Effectively, this algorithm reduces the dimensionality of the movement task from 3D to 2D and permits the use of the mouse to perform common 3D object positioning tasks.

The *second* input technique used a 3D wand/tracker input device, but used only two axes of motion. The Y (up-down) motion of the wand was mapped to cursor movement in Y on the screen, and the X (side-to-side) motion of the wand was mapped to cursor movement in X. The depth of the moved object was controlled automatically via the same sliding algorithm described in [17]. In other words, the user had no direct control of the Z (depth) axis in this mode; they merely move the input device in X and Y, and the software handled the depth. In effect, this creates a "mouse emulation" mode, although a mouse is pushed away to move the cursor up, while our technique requires the user to move the wand up for this. We used this mode to investigate the differences between 2DOF and 3DOF input devices. This technique is referred to as the "WandSlide" technique for the remainder of the paper. It supports all of the guidelines listed above, except 2.7 – the observation that 2DOF devices tend to provide grater accuracy.

The *third* input technique also used the 3D wand/tracker, but object movement was directly mapped to 3D position of the device. This mode did not use the sliding algorithm described above. Selection, however, was still based on ray casting. Upon selection of an object, the object moves in 3D according to the 3DOF motion of the wand. No collision detection/avoidance was used in this mode, which makes this a "raw" 3D direct manipulation mode. This is representative of traditional VR object movement techniques and ignores most of the guidelines above. We refer to this technique as the "Wand3D" technique.

3.2 3D Positioning Tasks

We chose two different positioning tasks for the study presented here. The first task, depicted in Fig. 1, involved the selection and movement of the red target cube to the top of the pedestal. This task was based on a similar tasks used in previous work [4]. It was chosen because the motion required to position the cube is relatively simple to perform with any input device and thus it can serve as a representative "abstract" movement task. While somewhat overly simplistic, it may give indications of how a series of movements comprised of such simple, short movements performs with different positioning techniques. Also, because the cubes were positioned in the foreground, and the target pillar was placed in the background, we hypothesized that this task would help us analyze any potential benefits of the extra depth cues provided by stereoscopic graphics and head tracking.

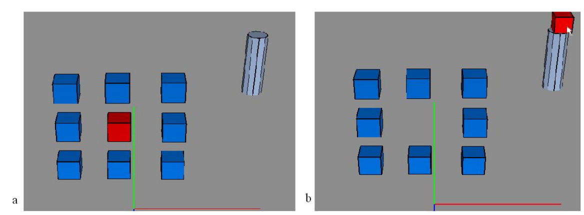

Figure 1. The cube positioning task, (a) starting scene, (b) target scene.

The second task was the assembly of a chair from several pieces (see Fig. 2). This task was chosen as a representative real-world assembly task. It is slightly harder than the cube placement task, as it requires the accurate placement of multiple objects. This task was also previously used to compare the mouse sliding movement technique to 3D widgets [17].

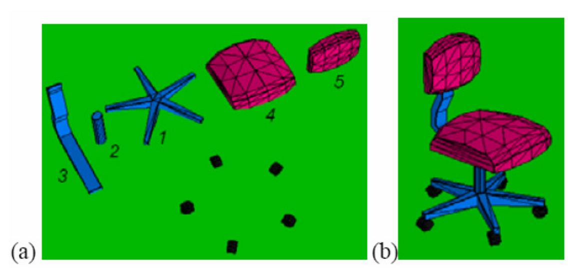

Figure 2. The chair assembly task, (a) starting scene, (b) target scene.

Note that this task cannot be adequately handled by techniques that use only gravity and collision avoidance. The chair task involves the backrest (part #5 in fig. 2), that must be attached *horizontally* to the support behind it (part #3). Using gravity alone it is impossible to perform this attachment. The sliding paradigm easily handles cases like this, as the backrest can be slid up the support up to the desired position. The object then remains affixed to the position where it was released. Hence, we believe the sliding movement technique to be more appropriate for assembly type tasks compared to traditional approaches.

4. USER STUDY

We conducted a user study to compare the input techniques described above and to determine if stereo graphics and head tracking provide any benefits to 3D positioning using these techniques. Furthermore, this study also indirectly validates the proposed guidelines.

4.1 Hypotheses

4.1.1 3D positioning technique

We hypothesize that the mouse mode will outperform the two conditions with the 3D input device. In addition to the effect of extensive user familiarity with the mouse, the reduced hand jitter in this condition will favor 2D input over 3D input. Another factor that should play a role here is the lack of collision detection in the Wand 3D mode.

4.1.2 Stereoscopic graphics

We hypothesize that the addition of stereoscopic graphics will improve the participants' ability to position objects in 3D, thus reducing task performance time and improving accuracy due to the extra depth cue provided. In other words, stereo should make it easier to perform 3D object positioning, even with 2D input devices.

4.1.3 Head-coupled perspective

We hypothesize that the addition of head coupled perspective will also improve accuracy, despite previous findings that suggested little benefit from it [2]. The extra motion depth cue provided by head coupling should assist users in gauging depth better thus obviating the need to rotate the entire scene.

4.2 Participants

Twelve paid volunteers participated in the study, with age ranging from 23 to 34 years, mean 25.7. Seven participants were male. Nine of the twelve reported using a mouse for 10 or more years, the remainder reported 5 – 10 years of experience. Since approximately 8% of the population is incapable of fusing stereo pairs [13], participants were also screened for stereoscopic viewing ability.

Participants' game playing habits were also recorded, as it is possible that they are a confounding factor. We found in a pilot study to this work that gamers tend to skew the results of studies of 3D interaction with 2D input devices, performing significantly better that those with limited game experience. Only one participant reported playing games more often than once per week. Two others reported playing games roughly once every week, and the rest played approximately once per month, or less frequently. Based on similar reasoning, we also asked participants about prior experience with 3D modeling tools. The majority of the participants had little to no experience with 3D modeling, with seven having never used such software, and the remaining five only using it approximately once per month, or less frequently.

4.3 Equipment

Tasks were performed in a fish tank VR system. The system was an AMD Athlon 64 1.81GHz with 1GB of RAM, and an NVIDIA Quadro FX3400 graphics card. A standard desktop optical mouse was used as input device in one condition, and an Intersense 6DOF wand was used in the other two. Stereoscopic graphics was provided using a Stereographics emitter and CrystalEyes shutter glasses. An Intersense IS900 was used for 3D head tracking and 3D wand. The head tracking sensor was mounted on the shutter glasses. The display was a Silicon Graphics monitor at 1024x768 @ 120HZ. The software used was written in C++ with OpenGL.

In stereo mode, using the system cursor with stereo graphics produces a "dual cursor" effect when a user focuses on the cursor. To avoid this, the software was modified to only draw the mouse cursor synchronized with the dominant eye, as discussed in [24]. This one-eyed cursor was aligned to the position of the operating system cursor, to allow accurate selection of faces and objects as required by the experimental tasks.

4.4 Procedure

Participants completed a series of object movements for each trial, using the tasks described in section 3.2. In each trial, they used one of the three movement modes from section 3.1. For the chair task, participants were informed of the order in which parts should be assembled, and were asked not to move the chair's wheels. Hence, they started with part #1 in Figure 2 (the base of the chair). This ensured that the experiment was not testing 3D construction skills, but only the input techniques and display modes. Prior to both tasks, participants were given a brief practice period of up to 5 minutes to familiarize them with the 3D sliding movement algorithm used in the system, as well as the various input devices. During the experiment, participants repeated each task twice.

In all trials, participants were asked to complete the assembly or placement task as quickly and accurately as possible. Prior to each trial, participants were informed of the status of each of the experimental factors, namely, whether head-tracking and stereo graphics were on or off, and which input device and technique was for this trial.

4.5 Design

The experiment was a 3×2×2×2 design. The independent variables were movement technique (Mouse, WandSlide and Wand3D mode), display mode (monoscopic or stereoscopic), head tracking (enabled or disabled), and task (chair assembly or cube placement) respectively. All factors were within-subjects and there were two repetitions of each condition. The orderings of display type, head tracking mode, input device and task were counter-balanced according with a balanced Latin square to compensate for possible asymmetric learning effects across conditions. Participants wore shutter glasses during *all* trials, to mitigate any confounding effect of the glasses themselves. they reduce the amount of light seen by the viewer, which can adversely affect the user's stereoacuity [13].

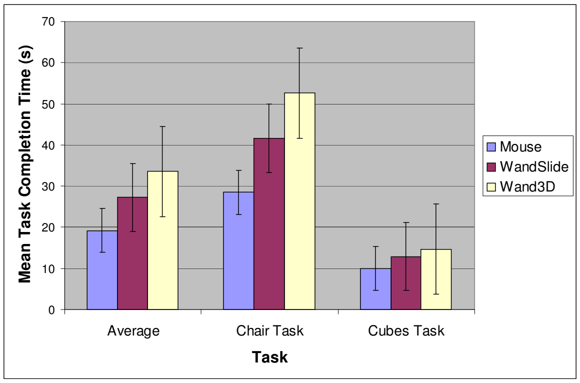

Figure 3. Mean task completion times by input technique and task. Error bars show standard error.

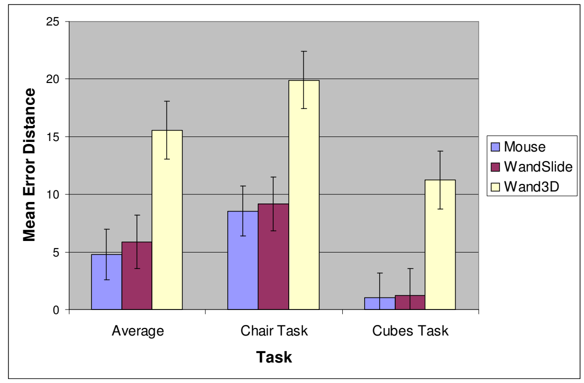

Figure 4. Mean error distance by input technique and task. Error bars show standard error.

Every participant completed every combination of movement technique and display mode twice, for a total of 48 trials. Participants took approximately 1 hour to complete this series of trials.

We had considered splitting the conditions into two separate experiments, one to compare the movement techniques, and the other to compare just the display modes. However, this would have made determining potential interactions between conditions nearly impossible. Since we were interested in comparing combinations of the conditions, we opted instead to include all in this single experiment. This way, we could determine if the addition of stereo and/or head-tracking aided any specific positioning technique more than others.

4.6 Results

We performed a repeated-measures ANOVA on the task completion times for all trials. A significant difference for positioning technique ($F_{2,22}$=34.348, p<<.01) was found. Tukey Kramer post-hoc analysis revealed that all three techniques were different, with Mouse (mean 19.3s) outperforming the WandSlide technique (27.3s), which in turn outperformed the Wand3D technique (33.6s). There was no significant difference for stereo or head tracking. Participants performed significantly better upon the second repetition of each trial ($F_{1,11}$=0.491, p<.05), as is to be expected without training. The mean completion time was 40.9s for the chair task, and 12.5s for the cube task; these were also significantly different, ($F_{1,11}$=64.053, p<<.01). Beyond that, there were no significant differences, with the exception of a significant interaction between task and positioning technique ($F_{2,22}$=17.574, p<<.01).

Accuracy was measured by summing the total error distance for each object in the scene compared to the target scene. There was a significant difference in accuracy for positioning technique ($F_{2,22}$=17.122, p<<.01) and for task ($F_{1,11}$=17.172, p<<.01). There was also significant difference for stereo mode ($F_{1,11}$=7.982, p<.05). The mean errors by positioning technique were 4.8 cm for the Mouse mode, 5.89 cm for WandSlide, and 15.6 cm for Wand3D. Post-hoc comparisons indicated no significant difference in accuracy between the Mouse and WandSlide modes – both of these modes were significantly more accurate than the 3DOF movement technique.

4.7 Discussion

The significant difference in speed between tasks was unsurprising. Intuitively, the cube placement task was far simpler than the chair assembly, requiring only a single precise object placement, rather than multiple actions.

The fact that full 3DOF movement with the wand took longer than the other two modes confirmed our first hypothesis. There are several likely causes for these results.

The first, as mentioned, is the participants' familiarity with the mouse compared to the wand. Essentially, the participants were already experts with the mouse but had no experience with the wand. This gives a major advantage to the mouse. Second, because the Wand3D condition used neither collision detection nor front-face sliding like the Mouse and WandSlide movement modes, participants required additional time to accurately position the manipulated object in 3D. Some participants commented on this, that the lack of collision detection and/or collision feedback made it difficult to judge when the object was positioned correctly. Another aspect is that hand jitter and fatigue combined with the relative sensitivity of the wand reduced the accuracy of the Wand3D technique significantly, compared to the other two techniques. We believe that the participants took extra time trying to correct for this reduced accuracy, eventually giving up when the scene looked "good enough". This is substantiated by the significantly worse accuracy with this technique.

Third, observations made during the experiment suggest that participants came to rely on the front-face sliding movement after they had been exposed to it in the WandSlide and Mouse conditions, often leaving objects floating well in front of their intended target in the Wand3D condition – an oversight that the 2D sliding algorithm automatically accounts for. This even occurred during stereo and head-tracked trials, where we believed that the additional depth cues provided would aid the users' accuracy. This suggests that the input technique has a much stronger effect on accuracy and speed than either stereo or head tracking.

Finally, the 2D sliding algorithm used in both the Mouse and WandSlide modes effectively reduces the dimensionality of the movement task from 3D to 2D. This is a clear benefit over "full 3D" movement techniques, as the user is only required to position the object accurately in two dimensions rather than three. Phrased differently the user is only required to line up the image of the object being moved with the image of the target. This strongly suggests that smart 3D movement algorithms can overcome the limitations of an input device (e.g. degrees of freedom) and can allow such input devices to outperform devices that seem to be better suited to the problem. Although this is technically no longer a 3D positioning task, but rather a 2D positioning task, the end result is the same – the object has been moved to a new 3D location in the scene.

Despite the relative quantitative performance of the input techniques, several of the participants commented that they found the Wand3D mode to be the most fun to use. Given the recent success of the Nintendo Wii game console, which uses a similar input device (www.nintendo.com), this is not very surprising. However, several users also commented that it was frustrating to use, and that they preferred using the mouse. Interestingly, no participants chose the WandSlide technique as their favorite. "Fun factor" is an important consideration in interface design as well, especially for games, and suggests that if 3DOF interaction techniques could be made as effective as 2DOF techniques (e.g. by following the guidelines suggested above) they may be a clear winner.

Our hypothesis regarding stereoscopic graphics was confirmed by the significant effect observed on accuracy. This conforms to previous studies, and as indicated, the extra depth cue allowed the users to more easily perceive the distances between objects. To our surprise, head tracking had no effect on accuracy or completion time. This is likely because the participants seldom *intentionally* used head tracking. One possible reason for this is that they simply forgot about it during the trials when it was active, despite being informed about the status of each factor at the beginning of each trial. It is also possible that they did not understand the full value of head tracking or felt the effect was too subtle to be useful. One participant even commented that the scene rotation by head movement would be more useful if rotation was exaggerated beyond realism. A third possibility is again related to the apparent reliance of the users on the front-face sliding movement algorithm – the users may have been assuming that the objects were sliding and that this feature was ensuring their accuracy, hence they felt they had no need to use the head tracking. Objects were often left floating far in front of the target, but appeared properly positioned in 2D. A subtle shift of the head in head-tracked mode would have revealed the distance between the cube and the target. Finally, it has been previously suggested that more complex scenes require more reliance on stereo and head-tracking [2]. Because the scenes used in our experiment were fairly small, consisting of only a few objects, only minimal view movements were required by the participants to determine the relative 3D location of the objects, which is yet another way to explain the lack of effect.

The interaction effect noted between task and positioning technique is interesting, as it suggests that some input devices are particularly well suited to specific tasks. Figure 3 shows that times for the Wand3D positioning mode were much closer to the WandSlide positioning mode for the cubes task than for the chair task. This is likely due to both the lack of collision detection (in this case, beneficial because the user could just move the cube through the others), and the fact that the 3D wand allowed for effectively a straight-line movement towards the target pedestal upon selecting the cube. Comparatively, the chair assembly task was much more difficult, requiring numerous accurate placements. Consequently, the wand fares worse under this higher accuracy requirement – especially in the 3DOF positioning mode.

Finally, it is interesting that there was no significant difference in accuracy between the Mouse and WandSlide modes, while there was a significant difference in speed. We believe that the reason is that the table on which the mouse slides provides a firm foundation upon which the participants' can rest their hand – and thus gain accuracy. Another factor is that the friction between mouse and table enables users to fairly rapidly stop their movement, compared to stopping a wand movement in the air. The wand, however, does not provide these benefits. Previous findings support this as well [6, 14]. Furthermore, the 2D sliding algorithm makes it quite easy to correct minor misplacements very quickly, hence the participants seemed more inclined to trade a bit of time for improved accuracy in this condition. Correcting such mistakes in 3DOF mode requires a significantly greater amount of work due to jitter and the additional axis that needs to be controlled simultaneously.

5. CONCLUSION

We presented several guidelines for the design of 3D positioning techniques based on observations from prior research. We then performed an evaluation of several 3D positioning techniques, two of which were based on the guidelines. The evaluated techniques included mouse-based 3D positioning with an intelligent sliding movement algorithm, and two techniques using a 3D wand. One of them used only 2DOF and the same intelligent sliding algorithm and the other allowed full 3D movement. The mouse was significantly faster than the 2DOF wand mode, which was significantly faster times than the 3DOF wand technique. However, no significant difference was found in accuracy between the mouse and the 2DOF wand modes. Additionally, we evaluated the effects of stereoscopic graphics and head coupled perspective on 3D positioning tasks. Stereoscopic graphics had a significant effect on accuracy, but head tracking did not.

5.1 Future Work

We are interested in determining how the guidelines presented above apply to more general virtual reality environments such as CAVEs. Due to the inherent reliance on 3D tracking equipment, it seems plausible that our results for fish tank VR systems can also generalize to other VR environments as well.

6. ACKNOWLEDGMENTS

Thanks to Rob Allison for the usage of his lab equipment for the user study.

7. REFERENCES

[1] E. Bier. Skitters and Jacks: Interactive 3D Positioning Tools. In Proc. of Interactive 3D Graphics 1986, pp.183-196, January 1987.

[2] J. Boritz, and K. S. Booth. A Study of Interactive 3D Point Location in a Computer Simulated Virtual Environment. In ACM VRST 1997, pp. 181-187, September 1997.

[3] J. Boritz, and K. S. Booth, A Study of Interactive 6 DOF Docking in a Computerised Virtual Environment. In Proc. of

the Virtual Reality Annual International Symposium, pp. 139-146, March 1998.

[4] D. Bowman, D. Johnson, and L. Hodges. Testbed Evaluation of Virtual Environment Interaction Techniques. In Proc. of VRST99, pp. 26-33, December 1999.

[5] H. Baradaran, W. Stuerzlinger, A Comparison of Real and Virtual 3D Construction Tools with Novice Users. In CGVR'06, pp. 10-15, June 2006.

[6] J. Chen, M.A. Narayan, and M.A. Perez-Quinones. “The Use of Hand-held Devices for Search Tasks in Virtual Environments,” In Proc. of the IEEE Virtual Reality 2005 workshop on New Directions in 3DUI, pp. 15-18, March 2005.

[7] D. Brookshire Conner, S. Snibbe, K. Hemdon, D. Robbins, R. Zeleznik, and A. van Dam. Three Dimensional Widgets. In Proc. of Symposium on Interactive 3D Graphics, June 1992.

[8] K. Dave, K. Dinesh. CInDeR: Collision and Interference Detection in Real-time using graphics hardware. In Proc. of Graphics Interface, pp. 73-80, May 2003.

[9] M. Deering. The Holosketch VR Skething System. Communications of the ACM, 39, 5, pp. 55 – 61, 1996.

[10] D. Diner, and D. Fender. Human Engineering in Stereoscopic Viewing Devices. Plenum Press, New York, 1993.

[11] N. Govindaraju, S. Redon, M. Lin, and D. Manocha. CULLIDE: interactive collision detection between complex models in large environments using graphics hardware. SIGGRAPH Workshop on Graphics Hardware, pp. 25-32, 2003.

[12] Tovi Grossman, Ravin Balakrishnan. The Design and Evaluation of Selection Techniques for 3D Volumetric Displays. In ACM Symposium on User Interface Software and Technology 2006, pp. 3-12, 2006.

[13] J. Hsu, Z. Pizlo, D. Chelberg, C. Babbs, and E. Delp. Issues in the Design of Studies to Test the Effectiveness of Stereo Imaging. In IEEE Transactions on Systems, Man and Cybernetics, Part A, 26, 6, pp. 810-819, 1996.

[14] R. W. Lindeman, J.L. Sibert, J. K. Hahn. Towards Usable VR: an empirical study of user interfaces for immersive virtual environments. In Proc. of CHI ’99, pp. 64-71, 1999.

[15] S. Obayashi, T. Suhara, K. Kawabe, T. Okauchi, J. Maeda, Y. Akine, H. Onoe, and A. Iriki. Functional brain mapping of monkey tool use. NeuroImage 14, pp. 853-861, 2001.

[16] J.-Y. Oh, W. Stuerzlinger. A system for desktop conceptual 3D design. In Virtual Reality 7, pp. 198-211, 2004.

[17] J.-Y. Oh, W. Stuerzlinger. Moving Objects with 2D Input Devices in CAD Systems and Desktop Virtual Environments. In Proc. of Graphics Interface 2005, pp. 195-202, 2005.

[18] I. Poupyrev, S. Weghorst, M. Billinghurst and T. Ichikawa, Egocentric object manipulation in virtual environments: empirical evaluation of interaction techniques. In Eurographics ’98, pp 41-52, 1998.

[19] B. Shneiderman. Direct manipulation: A step beyond programming languages, IEEE Computer 16, 8, pp. 57-69, 1983.

[20] G. Smith, T. Salzman and W. Stuerzlinger. 3D Scene Manipulation with 2D Devices and Constraints. In Graphics Interface ’01, pp. 135-142, 2001.

[21] P. Strauss and R. Carey. An Object-Oriented 3D Graphics Toolkit. In Proc. of SIGGRAPH 2002, pp. 341-349, July 2002.

[22] W. Stuerzlinger, D. Dadgari, J.-Y. Oh. Reality-Based Object Movement Techniques for 3D, CHI 2006 Workshop: "What is the Next Generation of Human-Computer Interaction?”, April 2006.

[23] Z. Szalavári, M. Gervautz, The Personal Interaction Panel, A Two-Handed Interface for Augmented Reality, Eurographics ’97, pp. 335-346, September 1997.

[24] 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, pp. 309-322, 1997.

[25] Z. Wartell, L. F. Hodges, W. Ribarsky, A geometric comparison of algorithms for fusion control in stereoscopic HTDs. IEEE TVCG, 8, 129-143, 2002.

[26] C. Wickins, J. Hollands, Spatial displays, in Engineering psychology and human performance, Prentice-Hall 3rd Edition, 1999.

[27] S. Zhai, W. Buxton, P. Milgram. The “Silk Cursor”: Investigating Transparency for 3D Target Acquisition. In Proc. of CHI ‘94, pp.459-464, 1994.