PRISM Interaction for Enhancing Control in Immersive Virtual Environments

SCOTT FREES
Ramapo College of New Jersey
G. DREW KESSLER
Sarnoff Corporation
and
EDWIN KAY
Lehigh University

When directly manipulating 3D objects in an immersive environment we cannot normally achieve the accuracy and control that we have in the real world. This reduced accuracy stems from hand instability. We present PRISM, which dynamically adjusts the C/D ratio between the hand and the controlled object to provide increased control when moving slowly and direct, unconstrained interaction when moving rapidly. We describe PRISM object translation and rotation and present user studies demonstrating their effectiveness. In addition, we describe a PRISM-enhanced version of ray casting which is shown to increase the speed and accuracy of object selection.

Categories and Subject Descriptors: I.3.6 [**Computer Graphics**]: Methodology and Techniques—*Interaction techniques*; I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism—*Virtual reality*

General Terms: Human Factors

Additional Key Words and Phrases: 3D object manipulation, direct object manipulation, precision manipulation, virtual reality

ACM Reference Format:

Frees, S., Kessler, G. D., and Kay, E. 2007. PRISM interaction for enhancing control in immersive virtual environments. ACM Trans. Comput.-Hum. Interact. 14, 1, Article 2 (May 2007), 31 pages. DOI = 10.1145/1229855.1229857 http://doi.acm.org/ 10.1145/1229855.1229857

Authors' addresses: S. Frees, School of Theoretical and Applied Science, Ramapo College of New Jersey, 505 Ramapo Valley Road, Mahwah, NJ 07430; email: sfrees@ramapo.edu; G. D. Kessler, Sarnoff Corporation, CN 5300, Princeton, NJ 08543-5300; email: gkessler@sarnoff.com; E. Kay, Department of Computer Science, Lehigh University, 19 Memorial Drive W, Bethlehem, PA 18018; email: ejk0@lehigh.edu.

1. INTRODUCTION

Designing models within immersive virtual environments is unique in that we directly manipulate objects in a manner similar to how we work in the physical world. Direct 3D interaction offers a highly intuitive and transparent user interface, in contrast to using a mouse and keyboard to indirectly control a 3D model through two-dimensional abstractions. A significant drawback of direct interaction however is the limited accuracy and fidelity of interactions. In the physical world, we use an object's weight and inertia, along with friction and supporting objects, to control our hand movements and counteract hand instability. In the virtual world, these aids are normally not at our disposal and hand instability dramatically reduces the accuracy with which we interact with the world.

Hand instability affects many of the design choices made when developing an immersive application. Objects must be large enough to be easily selected, limiting the complexity of the world in which a user can work. Once the object is selected, the user must be able to precisely specify its final position and orientation. One might provide anchor [Mapes and Moshell 1995] and grid points [Beir 1990], which snap the object to predefined (and thus limited) positions and orientations in the world. Another technique is to explicitly zoom in or scale up the workspace in order to work in small areas and make fine adjustments to objects [Mine et al. 1997; Bederson et al. 2000]. This works well in some instances but in an immersive environment occlusion and the loss of context can be problematic. The virtual inertia technique [Ruddle et al. 2002] addresses hand instability directly by making objects resistant to movement, limiting the rate in which they can be moved. Virtual inertia does not support unconstrained motion when precision is not desired, however. Yet another approach is to use indirect object control, perhaps a 3D menu, slider, or button. Once again, hand instability limits the design of these widgets, requiring them to be larger than their 2D desktop counterparts. The increased size of widgets limits the number of widgets that can simultaneously be placed within a comfortable working space.

We present a 3D interaction technique called PRISM: Precise and Rapid Interaction through Scaled Manipulation. The goal of PRISM is to increase the accuracy and control of the user in a virtual world to a level closer to what they are accustomed to in the physical world. PRISM increases the control/display (CD) ratio, which causes the cursor or object to move and rotate more slowly than the user's hand, reducing the effect of hand instability. One challenge of improving precision is deciding how to constrain interaction without sacrificing speed when precision is not desired. Scaling the movement and rotation of the hand slows the speed of interaction and is only suitable when the user is interested in accuracy. This suggests the need for two distinct modes: one that scales hand movement when accuracy and precision is needed and one that provides direct, unconstrained interaction when moving the object from one general location or orientation to another.

One method to provide these two modes requires the user to explicitly indicate what mode is desired, perhaps through a menu, button, or other control metaphor. Recalling Fitts' Law [Fitts 1954], however, the user is continuously providing the interface a clue as to whether they have a precise goal in mind. Fitts' Law states that, as the target gets smaller, we must slow our hand movement. Conversely, if the target is large we can, and will, move more quickly. This implies that when the user is moving their hand slowly they are likely to have a precise position or orientation in mind; PRISM uses this premise to dynamically adjust the CD ratio according to the user's current hand speed. As the speed of the hand decreases, PRISM increases the CD ratio to provide scaled manipulation and filter out hand instability. As the speed of the hand increases, PRISM reduces the CD ratio back towards 1, providing direct, unconstrained interaction. By taking advantage of the principals of Fitts' Law, PRISM provides a very natural control metaphor to facilitate modal switching; the user's natural behavior indicates how much accuracy and precision to provide. This type of dynamic adjustment of the CD ratio has its roots in 2D mouse/cursor control [Foley et al. 1984]; the mouse is commonly configured to cover more pixel space when moved at high speeds and less when moved across the user's desk slowly.

The next section provides an overview of other techniques that relate to precision interaction in both 2D and 3D. Section 3 describes the implementation of PRISM translation and rotation in detail. Section 4 discusses how PRISM can be applied to several common tasks in an immersive environment including object manipulation, ray casting, and widget control. Section 5 presents a series of user studies showing the effects of PRISM followed by future directions and conclusions.

2. RELATED WORK

Techniques adjust the control display ratio for various purposes such as overview vs. detailed modes while navigating documents [Igarshi and Hinckley 2000] and desktop-based 3D worlds [Tan et al. 2001] as well as increasing precision on desktop [Blanch et al. 2004] and touch screen displays [Albinsson and Zhai 2003]. Although similar to PRISM in concept, these systems (along with mouse cursor control) involve indirect manipulation, where reconciling offset/divergence between the physical device and the hand is a more straightforward task. For 3D direct manipulation, Dominjon et al. [2005] altered the control/display ratio (in the opposite direction from PRISM) in mixed reality environments to alter the user's perception of object weight. For their purposes, users could not see the divergence between the hand and object.

The Voodoo Dolls technique [Pierce et al. 1999] enables users to scale their workspace (and thus scale their movements) by selecting a voodoo doll of an appropriate size. PRISM also scales down the user's movement; however, it does so without requiring an appropriately sized reference object, which might not always be available. Other techniques use relationships between objects to constrain interaction [Bukowski and Séquin 1995; Stuerzlinger and Smith 2002]. These techniques still require some form of direct interaction on an object, which is where PRISM can be useful, possibly working in conjunction with these techniques. Although PRISM is aimed at virtual environments without force feedback, many applications benefit from the use of physical props to aid and constrain interaction. The use of props in neurosurgery applications [Hinckley et al. 1994b], menu interaction, and object manipulation [Lindeman et al. 1999; Ishii and Ullmer 1997] have been shown to be effective. Although there is little doubt that manipulating physical props helps in cognition and performance, props are specific to the application and are not general solutions to the precision problem.

Much of the earlier work in 3D rotation tasks focused on 2D mouse-based techniques for the desktop, including the Virtual Sphere [Chen et al. 1988] and the Arcball [Shoemake 1992]. Unlike object translation, task completion time is generally quite high in 3D rotation tasks, as reported by Ware [1990] and Zhai et al. [1996]. The difficulties of these indirect rotation techniques can be mitigated by direct manipulation, where the controlled object and the physical hand are co-located [Hinckley et al. 1997; Ware and Rose, 1999]. Poupyrev et al. [2000] have amplified 3D rotation (control/display ratio less than 1) and concluded that amplifying rotation decreases task completion time without significantly effecting accuracy.

Another area of research concerning object manipulation focuses on selecting and interacting with objects at a distance. Some of the more common techniques include ray-casting, Scaled-world grab [Mine et al. 1997], Image Plane interaction [Pierce et al. 1997], Go-Go [Poupyrev et al. 1996], HOMER [Bowman and Hodges 1997], and World in Miniature (WIM) [Stoakley et al. 1995]. As discussed below, PRISM can be used in conjunction with some of the above techniques to increase precision.

3. PRISM INTERACTION

Direct manipulation of virtual objects normally consists of "grabbing" an object, moving it to a new location and/or orientation, and "releasing" it. In our implementation, an object is grabbed when the user places the tip of a hand-held stylus within a virtual object and holds down the stylus button. While the user holds the button, the position and orientation of the virtual object directly follows the movement of the stylus. The virtual object is released when the stylus button is released.

A specific design challenge for a direct manipulation interface in 3D is to provide the user with the ability to perform deliberate, precise, fine-grained adjustments to the position and orientation of objects without removing the ability to move objects quickly from one general position to another [Hinckley et al. 1994a]. This design goal implies two distinct modes of interaction; one where constraints are supplied to aid in precision manipulation and another where the user directly controls the movement of the object, free of any artificial constraints. In response to this, PRISM uses the hand speed of the user to gradually switch between modes by altering the control/display ratio.

As shown in Figure 1, PRISM uses speed thresholds—defined by thee constants—to determine the CD ratio that controls the held object. The first constant is a minimum speed (*MinS*), below which a user is unlikely to be moving purposefully. Any motion below this speed is most likely tracking error

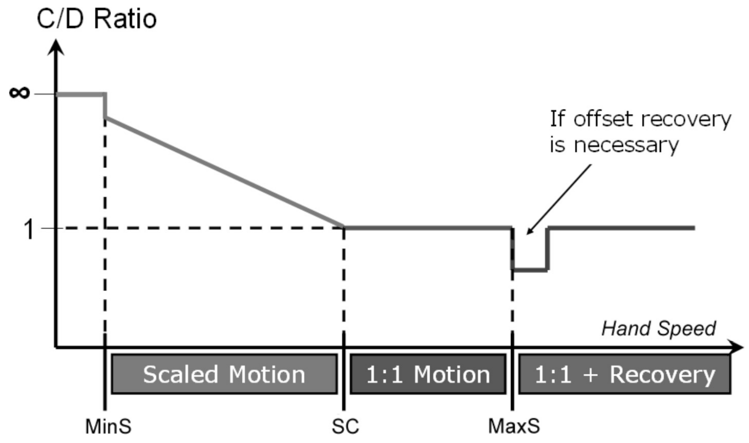


Fig. 1. Simplified interface diagram showing how PRISM uses Hand Speed to adjust CD.

or inadvertent drift and the controlled object is not moved. The second, and most significant constant, is the Scaling Constant (*SC*). If the user is moving their hand slower than this relatively low speed, they are likely to have a precise goal in mind. While in scaled manipulation, PRISM sets the CD ratio inversely proportional to the hand speed. For example, if the user is moving or rotating their hand at a slow speed, close to the *MinS*, a very high CD ratio is used and the object would move very little. If the hand speed is closer to *SC*, the CD ratio approaches 1 and the controlled object might move or rotate 90% of the distance the hand has moved during the most recent sampling interval. Any time the user is moving their hand at a speed above *SC* they are transitioned into imprecise, direct manipulation mode and the object mimics each movement and rotation of the hand.

Implicit in this method is the accumulation of an offset value representing either the positional or angular displacement between the hand and the object being manipulated. Each time movement is scaled, the virtual object moves or rotates a fraction of the distance the hand does. Although there are implementation differences between the translation and rotation techniques, in both cases a third constant, *MaxS*, triggers automatic offset recovery. Once the speed of the hand reaches this threshold, the offset is automatically reduced, causing the object to catch up to the hand. As shown in Figure 1, *MaxS* is typically larger than *SC* in order to provide the user with a “buffer” between speeds that lead to direct interaction and speeds that will trigger automatic offset recovery.

There exists an important relationship between the scaling constant (*SC*) and the sensitivity and precision (via scaling) the user experiences. The lower the *SC* value the less scaling will take place, making the controlled object more sensitive to hand movement. In contrast, a high *SC* value provides more scaling and the object will be resistant to hand motion, giving the user the ability to be more precise. Some users naturally move their hands more slowly and steadily than others and thus feel more comfortable using a lower scaling constant. On the other hand, some users have a lower degree of dexterity and have particular difficulty keeping their hands steady. Users in this category require the object to be less sensitive to their movements, thus they are likely to benefit from a higher scaling constant. The ability of the interface designer to easily change the behavior and "feel" of PRISM by adjusting *SC* not only allows it to be helpful for different people with different skills, it also allows it to be applied in many different situations or interaction tasks where users might require more or less sensitivity or accuracy.

3.1 PRISM Translation Implementation Details

PRISM uses hand speed to govern the CD ratio for object translation. The hand speed is determined by taking a sample of the hand position before each frame and determining the speed using the current location and the location of the hand 500 ms in the past. Note that this is a somewhat large interval and that small movements back and forth within the last 500 ms will not necessarily count towards the speed value. By using a "smoothed out" speed, a small movement in the positive direction quickly followed by a movement in the negative direction will result in a total hand speed below MinS and the CD ratio will be set to ∞, nullifying hand movement. This is crucial since small frequent movements back and forth are often indicative of tracking jitter or hand instability that ideally should not cause the object to move. Intentional hand movement in a particular direction will almost always last more than 500 ms—resulting in a non-zero speed. This interval was chosen through observation and proved to be a reasonable value for most users. Slightly different intervals may be more desirable for specific individuals. This is a simplistic form of input noise reduction compared to other techniques designed to combat tracking jitter [Lian et al. 1991; Welch and Bishop, 1997], however, this technique has proved suitable for our purposes. No subjects in our experiments have stated that the latency introduced by the use of this interval was perceptible or undesirable in any way. To move the object itself, PRISM determines the distance and direction moved between the last two frames and moves the object some proportion of that distance according to the hand speed. By using the "smoothed out" speed and applying it to the instantaneous movement of the hand, PRISM accurately reflects purposeful and sustained movement when determining the CD ratio while still being responsive to hand movement. As implied in (1), once the hand speed reaches the Scaling Constant (*SC*), the CD ratio is set to one, providing direct translation.

D_{object} *is the distance the controlled object will move*
D_{hand} *is the distance the hand itself moved since the last frame*
S_{hand} *is the speed of the hand over the last 500 milliseconds*
SC is Scaling Constant (meters per second)

$$\text{(1)} \qquad \mathrm{K} = 1/CD = \begin{cases} 1 & \text{for } \mathrm{S_{hand}} \geq \mathrm{SC} \\ S_{hand}/SC & \text{for } \mathrm{MinS} < \mathrm{S_{hand}} < \mathrm{SC} \\ 0 & \text{for } \mathrm{S_{hand}} \leq \mathrm{MinS} \end{cases}$$

$$\text{(2)} \qquad D_{object} = K \cdot D_{hand}$$

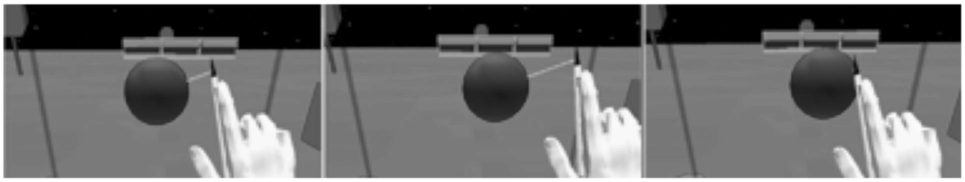


Fig. 2. Example of offset recovery with PRISM.

Table I. Offset Recovery Procedure when Hand Speed Exceeds *MaxS*

Offset is reduced over time, starting as soon as the speed threshold is met and continuing untill the offset is eliminated or the hand speed falls below MaxS. Offset is reduced before each frame according to the psuedocode below.*

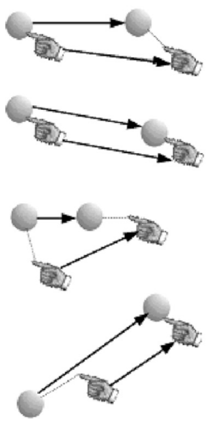
```
let T = time MaxV threshold was exceeded
let curTime = current time (on each frame)
for T < curTime < (T + 0.5 sec)
      offset = offset * 0.80
for (T + 0.5 sec) < curTime < (T + 1.0 sec)
      offset = offset * 0.50
for curTime > (T + 1.0 sec)
      offset = 0
```

*—In our trials the framerate was quite constant, making a frame based offset recovery feasible. A more general approach would be to reduce offset strictly based on time.

3.1.1 *Offset Recovery.* Whenever the user is in scaled mode an offset accumulates between the hand and the controlled object. This offset is graphically represented by a white line between the virtual hand and the object and is shown in Figure 2. The PRISM translation technique uses two mechanisms to recover this offset without interrupting the user's interaction. The first mechanism works when the user is in any mode, has accumulated an offset in a particular direction, and then changes direction moving their hand back towards the object. Under this circumstance, the object is not moved until the hand crosses back through the object, reducing the offset to zero. This mechanism was effectively used as a clutching/ratcheting technique by users and also guarded against unintended hand movements.

The second automatic offset recovery technique activates once the user's hand speed exceeds the *MaxS* threshold. At this speed, the user is likely trying to move the object to a general location and is not concerned with accuracy. Starting at the moment the speed exceeds *MaxS,* PRISM speeds up the movement of the object so it catches up to the hand. Once the object and hand are at the same position, the CD ratio returns to one. If at any time during the recovery period (around one second) the hand slows below *MaxS*, the recovery is stopped and the object's position will follow the hand according to (1). Table I summarizes this offset recovery technique.

3.1.2 *Axis Independent Scaling.* PRISM translation operates on each axis (x, y, and z in world coordinates) independently. For instance, the hand speed in the X direction only affects the scaling mode (and movement of the controlled object) in the X direction. This allows the user to move their hand rapidly



a. User moves hand slowly to the right and down. Some movement is scaled down in the horizontal direction, nearly all is scaled down in the vertical

b. User moves in the same direction as in (a), this time more quickly. Since interaction is in direct mode, object follows hand to its exact location.

c. User slowly moves to the right and up back towards the object. The vertical offset is recovered. Scaled motion is performed in the horizontal direction.

d. After accumulating offset, the user moves hand quickly up and to the right. Offset is eliminated and translation has switched to direct translation.

Fig. 3. Examples of typical interactions when using the PRISM translation technique.

in the X direction and retain direct control while simultaneously being more precise (in scaled mode) in the Y and Z direction, eliminating inadvertent drift. PRISM could have been implemented using the Euclidean speed of the hand to calculate the mode and degree of scaling as well, however this method would not eliminate drift to the same extent. The biggest advantage of independent scaling is that it helps users move the object in a straight line along a principal axis. The disadvantage is that PRISM scales diagonal movement more than movement along a principal axis. Scaling against the principal world axis was a fairly arbitrary design choice; other options could be the principal axis of the user, view, or perhaps another object's coordinate system. Further investigation is required to determine which scaling method is most suitable for different situations.

3.1.3 *Interaction Examples.* Figure 3 presents several typical interaction examples. For simplicity, the examples are illustrated in 2D. In Figure 3(a), the user moves a virtual sphere to the right at a speed just under *SC*. The hand motion in the horizontal direction is scaled by a small amount. At the same time, the user also moved their hand down in vertical direction at a very slow speed, indicative of an inadvertent drift in their hand position. Since the scaling values are calculated independently in each axis, this vertical movement is completely filtered out, without affecting the movement in the horizontal direction. In Figure 3(b), the user moves their hand in the same manner as in Figure 3(a), however, this time the hand is moved quickly in both the horizontal and vertical direction. This quick motion (faster than *SC*) results in direct manipulation in both directions and the sphere maintains its relationship with the hand.

In Figure 3(c), the user begins with an offset in the vertical direction (from some previous interaction) and then moves slowly up towards the object and to the right. In this situation, the vertical offset is completely recovered and none of the upward hand movement translates into vertical movement of the sphere. Since the hand also moved slowly to the right, the user is left with an offset in the horizontal direction. In Figure 3(d), the user again begins with an offset in both the horizontal and vertical directions. Since the hand motion is faster than *MaxS*, the offset is recovered in both directions as the controlled object "catches up" to the user's hand.

3.2 PRISM Rotation Implementation Details

The PRISM rotation technique uses the angular speed of the hand and three constants to determine the CD ratio of the interface. When the hand is rotating slowly, below a *MinS* value, the CD ratio will be ∞ and the object will not rotate. As the rotational speed of the hand increases, the CD decreases and the object is rotated by an angle proportional to the distance the hand has rotated in the most recent sampling interval. As the angular speed approaches an *SC* value, the object rotates directly with the hand. The third constant, *MaxS*, triggers automatic offset recovery and is discussed in Section 3.2.1.

Working with 3D rotation is more complex than the Euclidean geometry involved in translation. With translation, PRISM separates the movement and speed of the hand into three values corresponding to the x, y, and z directions. The distance covered in each of these directions can be easily scaled up or down while preserving the general direction of movement in three dimensions. Although a similar implementation may be possible using Euler angles (pitch, yaw, and roll), there are multiple combinations of these angles that describe a single change in orientation, and it is difficult to choose one that is consistent with the user's model of the change. Along with Gimbal lock, these problems make describing 3D rotation using Euler angles a poor approach. Instead, PRISM uses a quaternion representation for the orientation of the hand and the controlled object. The mathematics of quaternions is more difficult to understand, however concatenation, interpolation and scaling of 3D rotations is more straightforward. A simplified view of a quaternion is a four-dimensional vector consisting of a 3D vector $v\{x, y, z\}$ and a real number w. The real component w relates to the angle rotated around the axis defined by the vector v (w is $\cos(\alpha/2)$ where α is the angle rotated around v).

Before each frame a quaternion representation of the hand orientation is recorded. PRISM rotation calculates the rotational speed of the hand before each frame by comparing the current orientation and the orientation of the hand 200 ms in the past, in the same manner as in PRISM translation. This sampling interval was determined through observation and is less than its counterpart in the translation technique because a continuous rotation normally has a shorter duration than a translation. As with translation, users did not report negative consequences stemming from the latency introduced by using this interval.

Equation (3) calculates the rotational difference between the current ($\mathrm{Q_t}$) and last orientation (Q_{t-1}) of the hand, in the form of $\mathrm{Q_{diff}}$. Note, to find the quaternion needed to rotate from $\mathrm{q_1}$ to $\mathrm{q_2}$, $\mathrm{q_2}$ is divided by $\mathrm{q_1}$. Equation (4) converts the angle represented by $\mathrm{Q_{diff}}$ from radians to degrees and Eq. (5) simply divides the angle by 200 ms (the time between $\mathrm{Q_t}$ and $\mathrm{Q_{t-1}}$) to obtain the rotational speed of the hand. Equation (6) is used to determine the control display ratio to be used. The inverse of the control display ratio, k, is used to scale rotation. In Eq. (7), the quaternion representation of the angle the hand has rotated ($\mathrm{Q_{diff}}$)

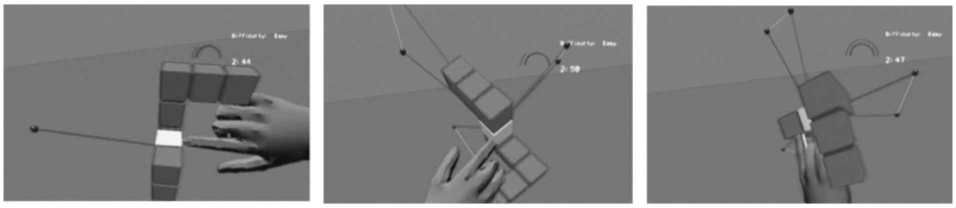


Fig. 4. Screen shots showing angular offset accumulation with PRISM rotation.

is scaled by raising it to the power k, where k is a real number between 0 and 1. The reader is referred to Poupyrev et al. [2000] and Shoemake [1985] for details on how the quaternion power function amplifies and scales rotations and how it is calculated). The scaled rotation is then added to the current orientation of the object (Q_{object}), which gives the new orientation of the object

*Q_{diff} is the quaternion representing the angle the hand has rotated in the last 200 ms.**

Q_{t} is the quaternion representing the current hand orientation

Q_{t-1} is the quaternion representing the hand orientation 200 ms before the current time

Q_{new} is the quaternion representing the new orientation of the object

Q_{object} is the quaternion representing the current orientation of the controlled object

*A is the angle (in degrees) the hand has rotated in the last 200 ms.**

RS is rotational speed of the hand in degrees / second.

SC is Scaling Constant (degrees per second)

**The axis the hand has rotated around is not needed for this calculation*

$$Q_{diff} = (Q_t / Q_{t-1}) = \left(Q_t Q_{t-1}^{-1}\right) \tag{3}$$

$$A = (2 \cdot acos\ (Q_{diff} \rightarrow w)) \cdot 180/\pi \tag{4}$$

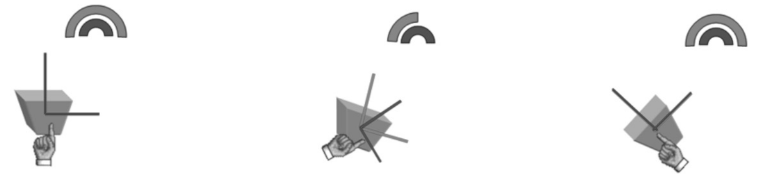
$$RS = A/0.20\ sec \tag{5}$$

$$k = 1/CD = \begin{cases} 1 & \text{for RS} \geq \text{SC} \\ RS/SC & \text{for MinS} < \text{RS} < \text{SC} \\ 0 & \text{for RS} \leq \text{MinS} \end{cases} \tag{6}$$

$$Q_{new} = Q_{diff}^{k}\ Q_{Object} \tag{7}$$

$\cdot$ $Q_{\text{diff}} \rightarrow$w denotes the real component of the quaternion Q_{diff}

3.2.1 *Offset Recovery.* When scaling the rotations of the hand during slower movement, an angular offset will be accumulated between the hand and the controlled object. PRISM helps the user visualize this offset by drawing two sets of 3D axes, as shown in Figure 4. A lower axis follows the rotation of the hand while a upper axis follows the orientation of the controlled object. When the axes are aligned, the hand and the object are at the same base orientation. As they diverge, the user can see that scaling is taking place through the separation of these axes. As an additional aid for the user, two arcs are drawn



a. When the user first grabs hold the CD ratio is 1 and there is no angular offset

b. When rotating the hand clockwise slowly, the rotation of the hand is scaled and an offset accumulates

c. Quick hand rotation counter clockwise eliminates the offset and the object and hand align.

Fig. 5. PRISM Rotation 2D interaction examples.

on the top right of the viewing plane. The lower arc is always the same length (half circle) but the upper arc is scaled in real time to show the user the current CD ratio. When the upper arc is very small, the object will resist movement (high CD) and, when it is a half circle, it will follow the hand more directly.

As with translation, PRISM rotation automatically reduces the accumulated angular offset once the user rotates their hand beyond the *MaxS* speed threshold. While PRISM translation reduces this offset gradually over one second, PRISM rotation eliminates the offset immediately; which causes the object to align with the hand. We use immediate offset reduction because the amount of rotation that can be performed quickly is quite limited. Unlike translation, rotation is bounded in that one can only rotate an object 360 degrees before returning to the same position. Further limiting rotation is the physical limitations of the wrist—normally a user can rotate their wrist no more than about 180 degrees in one motion without putting an undesirable amount of strain on their elbow. Thus, using a gradual offset reduction (accelerating the object) would force the user to rotate quickly for a longer period of time than necessary, and would be undesirable.

For translation, PRISM also provides offset recovery whenever an offset is accumulated in a particular direction and the hand is then moved back towards the object. Translation can be easily broken down into three orthogonal directions, simplifying the implementation of this form of recovery. For rotation, the quaternion representation of the hand and object rotation cannot be broken down as easily; it is difficult to determine how much motion is actually aimed at reducing the offset. Although a similar automatic clutching, or ratcheting in this case, may be possible for 3D rotation, we have not implemented this form of offset recovery.

3.2.2 *Interaction Examples.* As shown in Figure 5(a), whenever a user gains control of an object the CD ratio is initialized to one, which is direct manipulation. The upper arc is at its full size, indicating to the user that they are in direct control of the object and the object and hand axes are aligned indicating that there is no angular offset present. In Figure 5(b), the user has slowly rotated their hand clockwise at a speed of around *SC*/2 (*SC* was 30 degrees/second in our implementation). Here the upper arc is at half size indicating that the object will only rotate about half the distance the hand rotates.

Since the object rotation has been scaled, there is an angular offset between the hand and the object, indicated by the gap between the upper and lower axes. Figure 5(c) shows that the user has now quickly rotated their hand counter clockwise. All offset is reduced since the speed of rotation was above *SC*. The upper arc is once again at full size, indicating that if the user continues to rotate at this speed the object will directly follow.

4. PRISM APPLICATIONS

The primary goal of PRISM is to enhance the precision of translational and rotational input coming directly from the user's hand. Although PRISM was designed to aid in direct manipulation of virtual objects, there are many other situations where the position and orientation of the hand is used to interact with the virtual world or the interface. In this section, we describe several situations aside from direct object manipulation where PRISM interaction can be used.

4.1 Selection

By increasing the precision in which the user positions and orients the cursor, the density of the selectable objects within a confined place can be increased without introducing selection errors. One of the most fundamental ways of selecting objects is to simply move the cursor (controlled by the user's hand) such that it intersects the object. Using the PRISM translation technique, the position of the cursor can be specified more accurately, allowing for easier selection of small, densely packed objects. This increase in density and precision could improve interaction in scientific visualizations and surgical training applications. In addition, PRISM can allow designers to increase the density or decrease the size of input widgets such as the Command & Control Cube [Grosjean and Coquillart 2001] without decreasing usability. The Command & Control Cube is a 3D cube divided into 27 equally sized cubes ($3 \times 3 \times 3$) in which each cube represents a specific command. In order to invoke a command the user must select one of the cubes by placing the cursor inside it. Using PRISM, each individual cube could be smaller allowing for either more commands ($4 \times 4 \times 4$, $5 \times 5 \times 5$ etc.) within the same dimensions or the entire interface could be shrunk uniformly, saving screen space. Other menu interfaces could benefit from the same increase in density as well.

There are many situations where users desire to select objects beyond arms reach. A common technique for this task is ray casting [Mine 1995], were the user points his or her hand at the object of interest and a ray is drawn which extends outwards from the user's hand/cursor. One problem with this approach is that small rotations of the wrist sweep out relatively large arcs at the end of the selection ray, which is shown in Figure 6. The increased sensitivity to slight hand rotations makes it difficult to select distant objects that do not occupy a large area in the viewing plane, a problem noted by Poupyrev et al. [1997]. One solution to this is to "snap" the ray to the nearest intersecting object. This method breaks down when working in a cluttered environment where slight movements of the wrist cause the ray to "snap" between several proximate

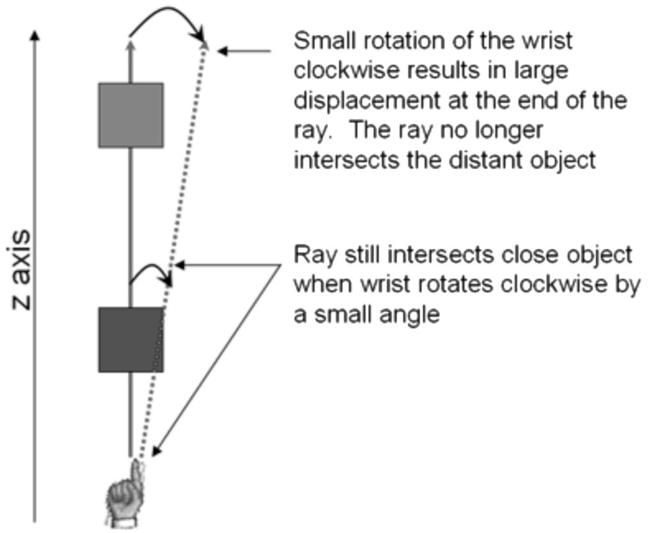

Small rotation of the wrist clockwise results in large displacement at the end of the ray. The ray no longer intersects the distant object
Ray still intersects close object when wrist rotates clockwise by a small angle
z axis

Fig. 6. Example of how small rotations of wrist make selection of distant objects more difficult.

objects. Our solution to improve the accuracy of ray casting employs PRISM rotation to control the angles swept out by the ray. When the user slows the rotation of their wrist the orientation changes are scaled down which offsets the amplification of these rotations at the other end of the ray. Section 5 presents a user study that provides more detail on the implementation of PRISM enhanced ray casting, along quantitative performance results.

4.2 Widget Control

Most 3D widgets incorporate some form of direct 3D translational or rotational hand input. The World in Miniature (WIM) widget [Stoakley et al. 1995] provides users with a hand held, miniaturized model of the virtual world, complete with all (or most) of the objects within it. The user directly manipulates the miniaturized version of the objects with the cursor and their actions are mimicked in the full-sized world. One problem with the WIM is that in order to interact with the miniaturized objects one must scale down the movements of the cursor as well. Moving an object within the WIM one centimeter might result in a translation of one meter in the full-sized world. If the user wishes to move an object by a centimeter in the full-sized world it would be nearly impossible to do so using the WIM, the user would need to navigate to the location in the full-sized world and manipulate the object directly. A WIM containing a cluttered world with many objects poses another problem in that many small, miniaturized objects might occupy a very small space, making selection increasingly difficult. With PRISM, users would be able to move the cursor quite accurately at very small scales, allowing them to select small closely packed objects and translate them at scales, which would be difficult with direct manipulation due to hand instability.

Table II. Cubes Were Larger than the Sphere; Error Tolerance Is the Length of the Cube Minus the Diameter of the Sphere

Target #	Difficulty Level	Error Tolerance
1	Easy	1.00 cm
2	Moderate	0.35 cm
3	Difficult	0.20 cm
4+	Very Difficult	0.12 cm

5. EVALUATION

We have conducted four separate user studies examining PRISM. Before each experiment users were provided with a training session requiring them to complete practice trials using both direct manipulation and PRISM. Participants were not permitted to participate in more than one of the four experiments. The equipment used included a four-port Polhemus 3SPACE FASTRAK electromagnetic tracking system with a hand-held stylus (Polhemus ST8, approx. 18 cm long, 1.3 cm diameter, 1 button) and a Virtual Research Systems V8 Head Mounted Display. Most users gripped the stylus as they would a pencil or pen (using the index finger to press the stylus button), however several used their thumb to depress the button. We used the Simple Virtual Environment toolkit [Kessler et al. 2000] to implement the environment. The system rendered the test environments in stereo and the frame rate was held between 20 and 30 frames/sec throughout the experiments. Tracking jitter and latency were within expectable limits; users did not report any difficulties. In each experiment, participants were standing and free to walk within tracking range (approximately 1m radius), although completion of the experiments did not require a large degree of mobility.

5.1 PRISM Translation

The object translation task required the user to pick up a virtual sphere and place it completely inside a translucent virtual cube such that no part of the sphere protruded any side of the cube. The cube changed color (and turned opaque) each time the user positioned the sphere completely inside the target (and released the stylus button to "drop" the sphere) to indicate a completion. A short time later the sphere re-appeared in its original starting position outside the cube. The participants were asked to repeatedly place the sphere inside the cube as many times as possible during the 3-minute trial. In each trial the target cube became smaller after each completion, beginning with an "Easy" difficulty level and progressing to the fourth target in which all subsequent cubes were "Very Difficult". This variation in difficulty level was included to increase the chance that all participants could complete at least one target in a trial without becoming overly frustrated. The difficulty levels the user experienced are summarized in Table II. Since all trials and users were given the same sequence of difficulty levels, this variation does not affect the overall performance measure (number of completions per trial).

The starting position of the target cube and mobile sphere were the same for each participant and trial. The target cube was placed at shoulder height

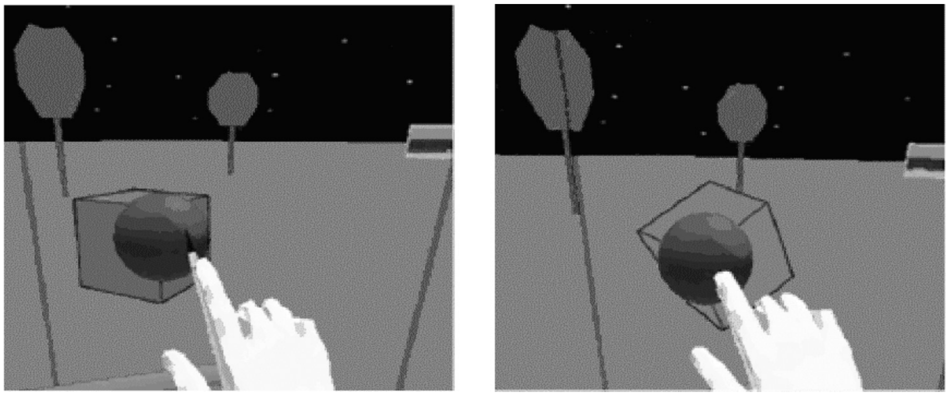


Fig. 7. Axis aligned (left) and rotated (right) target cubes. Edges of cube have been enhanced for visibility.

(25 cm below the top of the HMD) near the center of the virtual world. Each sphere was placed one meter away from the target in order to eliminate the possibility of the distance from the target affecting the difficulty level [Accot and Zhai, 1997; Fitts 1954]. The spheres were also placed at shoulder height. The distance between the target and initial position of the sphere ensured that movements consisted initially of arm/shoulder movements to bring the sphere close to the target and then fine-grained hand motion to precisely align the objects. The target and sphere positions relative to the *user* varied, since the user was free to move around as they wished. Most users stood at a position where the target was directly in front of them and the spheres appeared on the same side as their dominant hand. The training session prior to the experiment allowed them to become familiar with the layout of the world.

We conducted a within-subject experiment consisting of 18 (14 male, 4 female) undergraduate and graduate students. The experiment consisted of six trials and included two factors; the first was the interaction technique, at three levels: Direct, PRISM with Generic Scaling, and PRISM with Custom Scaling. The Custom Scaling trials used a Scaling Constant (*SC*) chosen by the participant during their training session. The difference between Generic and Custom PRISM was found to be insignificant and will not be discussed in detail. For more detail on the Custom Scaling aspect of this experiment see Frees and Kessler [2005]. A scaling constant of 0.15 m/sec was used in the Generic trials; *MinS* and *MaxS* were 0.01 m/sec 0.25 m/sec respectively for all trials. Target orientation was the second factor; target cubes were either axis-aligned or rotated 45 degrees in each principal axis (shown in Figure 7). Although target orientation has no theoretical effect on the difficulty of the task (the sphere still needs to be placed at the centerpoint of the target), users have a tendency to move the sphere diagonally towards the rotated targets instead of along a principal axis. We compared the use of rotated and axis aligned targets to draw conclusions about the axis-independent scaling strategy. Trial orders were balanced independently for interaction type and target orientation.

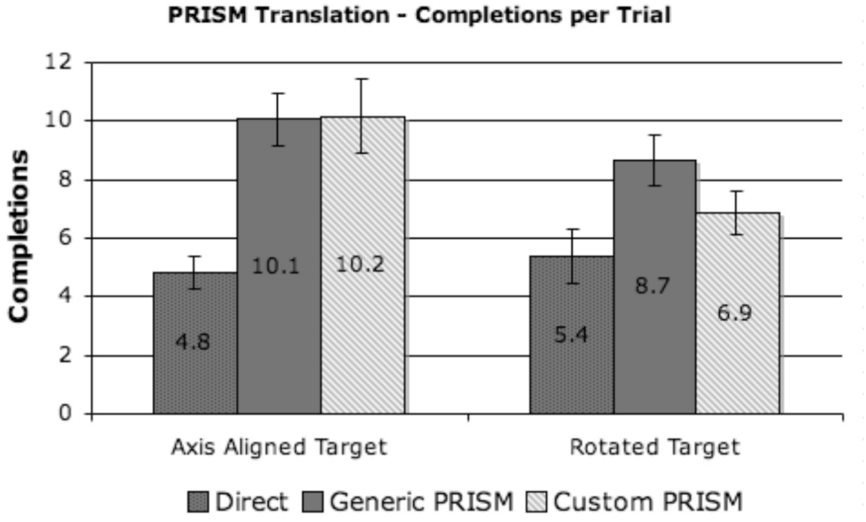


Fig. 8. Mean completion rates for translation task. Error bars represent standard error.

Table III. ANOVA Table for Translation Experiment

Translation Experiment Completions per Trial					
Source	SS	DoF	MS	F Ratio	P Value
Interaction Type (1)	365.17	2	182.58	33.17	0.00
Target Orientation (T)	50.70	1	50.70	3.57	0.08
Participant (P)	745.67	17	43.86		
I*T	66.13	2	33.07	3.50	0.04
I*P	187.17	34	5.51		
T*P	241.63	17	14.21		
Error	*	0			
Total	1656.47	34			

5.1.1 *Results.* The number of completions (number of spheres placed inside the cube) in each trial was used as the dependent variable in the experiment. The fixed-effect independent factors were Interaction Type (Direct, Generic PRISM, and Custom PRISM) and Target Orientation (Axis-Aligned and Rotated). Participants were treated as a random factor. Figure 8 shows the mean performance under each of the conditions. Table III reports the results of the ANOVA on these data. There was a significant effect due to Interaction Type ($F(2,34) = 33.17$, $p < 0.001$), the effect of target orientation was not significant ($F(1,17) = 3.57$, $p > 0.05$), and there was a significant Interaction-Type × Target-Orientation interaction ($F(2,34) = 3.50$, $p < 0.05$). A visual inspection of Figure 8 indicates that there were more completions for the Generic PRISM and Custom PRISM conditions, that this effect was greater for the Axis-Aligned conditions, and that there was no difference between the two PRISM conditions. This last observation was confirmed by an insignificant post-hoc Tukey test of pair-wise difference between the two PRISM conditions.

5.1.2 *User Feedback.* We administered an exit survey in which users responded to our questions by giving a score ranging from 1 (Strongly Disagree) to 5 (Strongly Agree). When asked whether they were able to adequately learn PRISM during the training period, 14 participants agreed, 2 were neutral, and 2 felt they did not. Nearly all participants agreed that either Generic or Custom PRISM was better than direct manipulation (13 preferred Generic over direct, 16 preferred Custom over direct). There was no consensus as to which set of targets (axis aligned or rotated) were more challenging (7 preferred axis aligned while 8 preferred rotated targets).

5.1.3 *Discussion.* PRISM allowed users to place the sphere more accurately inside targets then when using direct manipulation. As noted above, difficulty level increased within each trial until the participant reached the most difficult level; afterwards each target remained at that level until the completion of the 3-minute trial. This meant that the number of completed targets at the higher difficulty levels varied based on how quickly the participant completed the easier ones. This aspect of the design prevents us from making any statistical claims as to how PRISM or direct manipulation was affected by the difficulty level; however, we observed a clear trend. On the first two targets in each trial (the largest cubes), participants did not have trouble using PRISM or direct manipulation. When using direct manipulation, performance dropped off dramatically around the third or fourth cube. When using PRISM, performance did not noticeably drop off until the targets reached their smallest size and even then performance remained acceptable. The common strategy when using *direct* manipulation on the smaller targets was to rapidly and repeatedly pick up and drop the sphere—with the hope that it would eventually be dropped in the correct position. This strategy suggests that users did not have confidence in their ability to *purposefully* move the sphere to its proper position. This lack of confidence or control was not observed during the PRISM trials.

Although performance was better with PRISM for both types of targets, improvement over direct was reduced when using rotated targets. We were unable to determine the true cause of this and none of the participants expressed concerns about how PRISM was responding to their movements; most participants cited difficulties with getting a good view of the target when asked why they favored one set of targets over another. As discussed in the future work section below, the most obvious reason performance suffered is PRISM's axis-independent scaling, which slows diagonal motion more than movement along a principal axis.

5.2 PRISM Rotation

The rotation task required the user to rotate a virtual object (seen in Figure 9) such that it was completely inside a translucent target object that was the same shape but slightly larger. The centers of the object were always placed at the same location in the world (at shoulder height) and translation of the objects was disabled, leading to a purely rotational task. Participants were free to walk about the world as they wished. Once the user rotated the object (and released

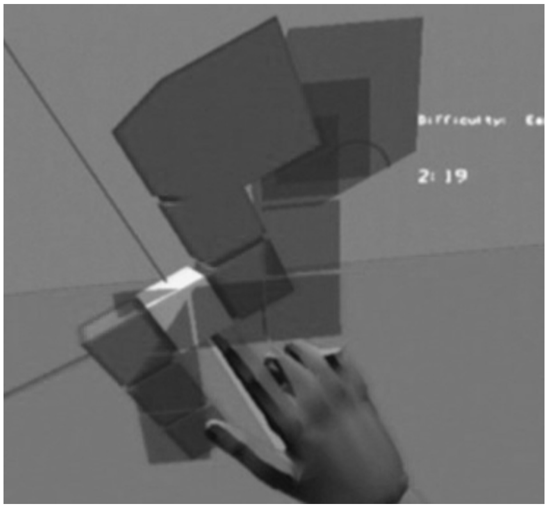


Fig. 9. User rotating the object such that it will match the target's (translucent object) orientation.

the stylus button) such that it was completely inside the target, the target changed color (and became opaque) to indicate a completion. A few seconds later the object re-appeared with an orientation offset from the target (the target orientation was held constant). The starting orientation of the mobile object was randomly selected at runtime from a set of four orientations to prevent the participant from being able to perform the same exact rotation repeatedly (which could lead to significant learning effects). Each of the four orientations required between 50 and 80 degrees of rotation about an arbitrary axis in order to align with the target.

Both objects were made up of nine individual cubes of equal size, numbered 1 to 9 in Figure 10. In order to record an alignment, the user needed to rotate the mobile object such that the displacement between the *centers* of each set of corresponding cubes (cube 1 in both the mobile and target object) were within a maximum allowed tolerance. Note, because of the 3D nature of the objects, it is impossible to align the objects without aligning each corresponding cube. Alignments could have been measured using the pure orientation of the objects; however, determining alignments based on the displacement between each set of corresponding cubes is more straightforward for situations when translation and rotation are involved, as is the case in Section 5.3.

This experiment included two factors: the first factor was the interaction technique (Direct and PRISM), the second factor was target size (or error tolerance), which was at three levels described in Table IV. A Scaling Constant of 30 degrees/sec was used, which was determined from pilot studies. For each PRISM trial, *MinS* was set to 1 degree/sec and *MaxS* was at 35 degrees/sec. There were six trials, each lasting 2 minutes. Users were asked to get as many completions as possible within the trial. Each user performed the task with each interaction type/error tolerance combination. We conducted this experiment with 15 (14 male, 1 female) undergraduate and graduate students. For

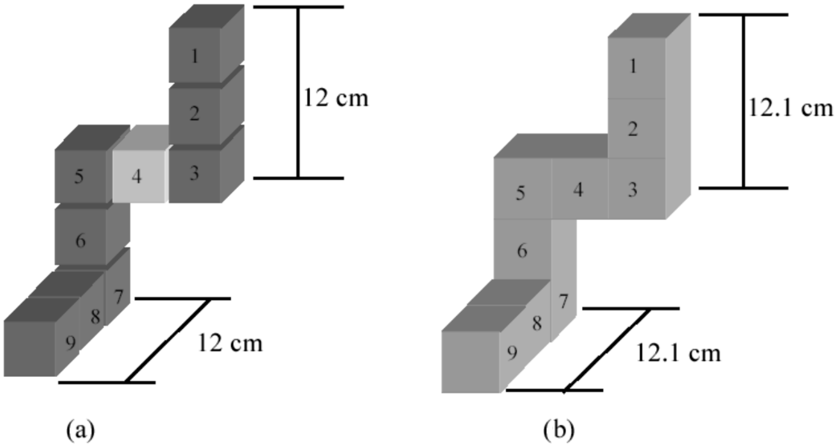


Fig. 10. (a) Mobile object, comprised of 9 cubes of equal size with a 0.05 cm space between each adjacent cube. (b) Target object, each of the 9 cubes are centered at the same position but are 0.05 cm larger than the corresponding cubes in the mobile object. In total, target object is a total of 0.1 cm larger than mobile object. In order for the mobile object to appear completely inside the target, displacement between corresponding cubes must be less than or equal to 0.05 cm.

Table IV. Error Tolerances for Rotation Task.
Tolerance (degrees) = atan (Tolerance (cm)/0.5 × Object_Length) * 180/PI

Error Tolerance Levels—Rotation Only		
Level	Tolerance (cm)	Tolerance (degrees)
Easy	0.18	1.72
Medium	0.12	1.15
Hard	0.10	0.95

Tolerance (cm) is the maximum displacement allowed between any part of the mobile and target objects in order to be accepted as a completion. Tolerance (degrees) can be derived from the Tolerance (cm) based on the equation in the table. The length of the object manipulated in the experiment was 12 cm.

each subject, the trial ordering was randomly determined. Note that the error tolerances used in this experiment are less than what was used in the translation task (more accuracy is needed). Based on pilot studies we felt users would be able to achieve a higher level of precision in a “rotation only” task than they were able to achieve with translation. In addition, rotation error propagates when larger objects are being used—which is not the case with translation. For instance, when rotating an object 2 m long, an error of just 1 degree will result in a 1.75 cm displacement of the object at its end points.

The size of the target was not adjusted to represent the error tolerance. During pilot studies, several users complained that occasionally the system did not register an alignment even when the mobile object appeared to be completely within the target. This was extremely confusing to the user since it gave no indication as to the nature of the misalignment. This was most likely a visual artifact caused by the graphics system. The target object remained slightly larger

Table V. ANOVA Table for Rotation-Only Experiment

Rotation Experiment Completions per Trial					
Source	ss	DoF	MS	F Ratio	P Value
Interaction Type (1)	22.50	1	22.50	5.28	0.04
Difficulty (D)	110.29	2	55.14	20.67	0.00
Participant (P)	333.49	14	23.82		
I * D	6.47	2	3.23	2.27	0.12
D * P	74.71	28	2.67		
I * P	59.67	14	4.26		
I * D * P	39.87	28	1.42		
Error	*	0			
Total	647.00	89			

than the mobile object only so the geometries of the objects could never completely overlap (which caused an awkward blending of the objects). Participants were told to align the two objects as closely as possible and that depending on the error tolerance, the alignment would be accepted when they released the object. Each error tolerance allowed a larger displacement than the size difference between the target and mobile object. This ensured any alignment that appeared to be completely within the target would always be accepted—thus ensuring that the user always had a visual indication of the misalignment. A wire frame visualization of the error tolerance could have been used. It is questionable whether users would have been able to effectively use that type of feedback however, as it would be challenging to identify where the solid object was protruding out of a wire frame object by a small amount.

5.2.1 *Results.* In this analysis, Interaction Type and Difficulty were the fixed independent variables. We also included participant number as a random independent variable. The dependent variable was the number of completions per trial. The ANOVA table for this experiment is shown in Table V, and the mean number of completions per trial are displayed in Figure 11. There were more completions per trial in the PRISM condition than in the Direct condition ($F(1,14) = 5.28$, $p < 0.05$). As the difficulty of the target increased, the number of completions per trial decreased ($F(2,28) = 20.67$, $p < 0.0001$). The Interaction-Type × Difficulty interaction was not significant ($F(2,28) = 2.27$, $p > 0.05$).

5.2.2 *User Feedback.* Feedback from a post-experiment survey was similar to the results we received for PRISM translation. Eleven out of 15 participants preferred PRISM over direct rotation, three preferred direct rotation and one participant had no preference. As for PRISM being easy to learn, PRISM received a score of 3.8 on a 5-point scale where 5 was Very Easy and 1 was Very Difficult.

5.2.3 *Discussion.* The results show a clear increase in performance when using PRISM; however, throughout the experiment, we noticed that many participants had difficulty determining which way to rotate the object in order to fit it inside the target. Often the user would see the object protruding out of a section of the target but could not determine which axis to rotate the object

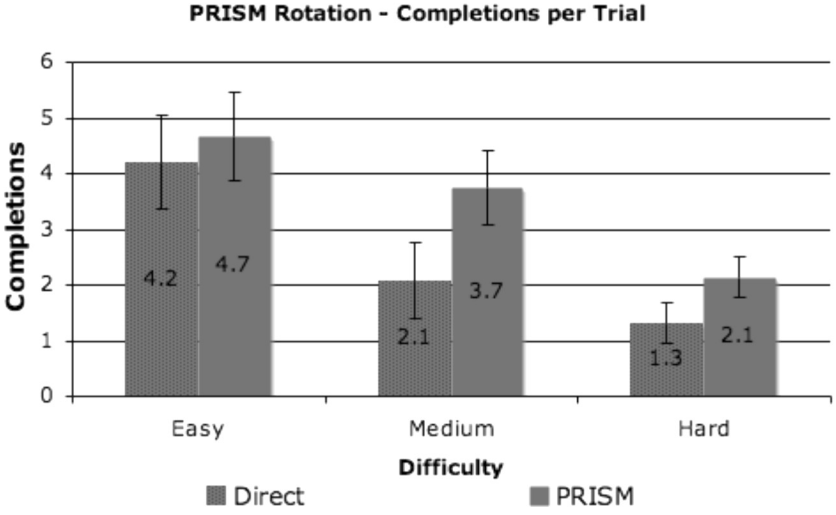


Fig. 11. Results of Rotation-Only experiment. Error bars represent standard error.

around in order to get the desired effect. These types of problems with understanding 3D rotation along with poor performance in 3D rotation tasks have been recognized in the literature [Parsons 1995]. When using direct rotation, this was not as problematic since accidental rotations of the hand resulted in object rotation. This accidental rotation occasionally caused the object to go inside the target on the easier difficulty levels, or at least cued the user as to which direction they needed to rotate the object on the more difficult targets. This confusion was a more serious problem when using PRISM. Since PRISM filters out much of the accidental and unintentional movement, the user needs to thoroughly understand which direction and around which axis they need to rotate the object. From observation, it was clear that, once the user knew which rotation would result in success, they were able to execute that rotation far more effectively with PRISM. In short, PRISM enhanced their ability to rotate objects *purposefully*, but it was a detriment if the user did not know which rotation was needed in the first place. This seems to be a general problem with 3D rotation rather than a problem with PRISM; one solution might be to administer spatial ability tests and lower the Scaling Constant (which would allow for more unintentional movement) for those who score poorly. We also suspect that providing a wire-frame representation showing the user how the object would rotate using direct manipulation would alleviate some of this confusion when using PRISM. This technique has been successfully employed by Ruddle et al. [2002] in cluttered environments.

5.3 Six DOF with PRISM

This experiment was quite similar to the rotation-only experiment; however, the object being manipulated was not constrained to a fixed position. The mobile object started approximately 0.5 meters away from the target and needed to be translated and rotated to fit inside the target. The layout of the world mimicked

Table VI. Error Tolerances for Rotation + Translation Task

Error Tolerance Levels—Rotation and Translation		
Level	Tolerance (cm)	Tolerance (degrees)
Easy	0.30	2.86
Medium	0.20	1.91

The length of the object manipulated in the experiment was 12 cm.

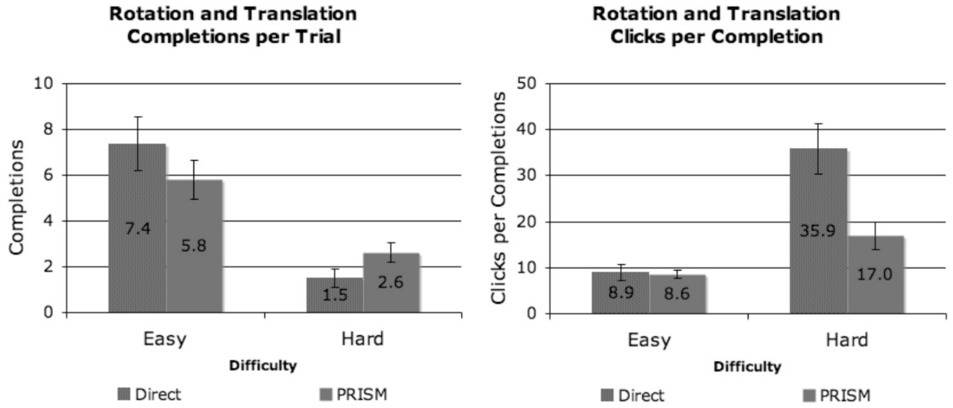


Fig. 12. Results of Rotation + Translation Experiment. Error bars represent standard error.

that of the translation experiment—the target object and the starting position of the mobile object were always at the same location in the world. Just as in the translation experiment, the distance between the target and initial position of the mobile object ensured that the task required both arm/shoulder movement and fine-grained hand motion. This task required more effort and skill than the first two tasks, the user needed to accurately control the position *and* orientation of the object. This is more difficult because hand rotation often changes the hand position and vice-versa [Bowman et al. 2001]. Due to the added difficulty, we increased the time for each trial to 3 minutes and only examined two difficulty levels, which are shown in Table VI.

The experiment consisted of four trials; each participant performed the task with direct manipulation and PRISM with both tolerance levels. In each trial, once the user rotated and translated the object (and released the stylus button) such that it was completely inside the target, the target changed color and became opaque. A few seconds later the object reappeared at the original starting position and with a random orientation chosen from the same set used in the rotation-only experiment. Sixteen undergraduate and graduate students (12 males, 4 females) participated in this experiment. A scaling constant of 0.15 m/sec was used for translation and 30 degrees/sec was used for rotation. For each subject, the trial ordering was randomly determined. We recorded the number of completions made in each trial.

5.3.1 *Results.* The results of this experiment are displayed in Figure 12 and Table VII below. As expected, there were fewer completions for the hard target than for the easy target (F(1,15) = 54.00, p < 0.0001). The Interaction-Type

Table VII. ANOVA Tables for Translation + Rotation Experiment

Translation + Rotation Experiment Completions per Trial					
Source	ss	DoF	MS	F Ratio	P Value
Interaction Type (I)	0.56	1	0.56	0.06	0.81
Difficulty (D)	324.00	1	324.00	54.00	0.00
Participant (P)	283.44	15	18.80		
I * D	30.25	1	30.25	6.24	0.03
D * P	90.00	15	6.00		
I * P	137.44	15	9.16		
I * D * P	92.75	15	4.85		
Error	*	0			
Total	958.44	53			

Translation + Rotation Experiment Button Clicks per Completion					
Source	ss	DoF	MS	F Ratio	P Value
Interaction Type (I)	2134.79	1	2134.79	14.25	0.00
Difficulty	5666.89	1	5666.89	31.20	0.00
Participant (P)	4949.49	15	329.97		
I * D	1915.05	1	1915.05	14.09	0.00
I * P	2246.62	15	149.77		
D * P	2724.42	15	181.63		
I * D * P	2038.03	15	135.87		
Error	*	0			
Total	21675.29	63			

× Difficulty interaction was significant (F(1,15) = 6.24, $p < 0.05$), there were fewer completions for the PRISM condition with easy targets and more with the hard targets. We observed that when using direct manipulation with the smaller error tolerances the participants often resorted to "guessing", where they quickly picked up and dropped the object repeatedly in hope that it would fall into place. To look at this more closely, we also used the number of button clicks per completion as a dependent variable. Participants did not seem to have much trouble on the easier targets however for the more difficult targets participants clearly had more trouble using direct, as evidenced by a nearly 100% increase in the number of times the user picked up and dropped the object. This is further evidenced by a significant Interaction-Type × Difficulty interaction (F(1,15) = 14.09, $p < 0.01$).

5.3.2 *User Feedback.* Overall, participants found the trials with the higher error tolerance to be quite easy and some told us that PRISM slowed them down on these trials. On the other hand, most participants felt PRISM was a large improvement over direct manipulation for the more difficult trials. Overall, 75% of the participants thought PRISM was more preferable and more effective than direct manipulation. Most users felt PRISM was relatively easy to learn, with an average score of 3.44 out of 5 (5 being "Very Easy to Learn").

5.3.3 *Discussion.* We saw a number of outside factors contributing to performance when using PRISM. First, it was quite obvious that the participants were having difficulty determining which direction they needed to turn the

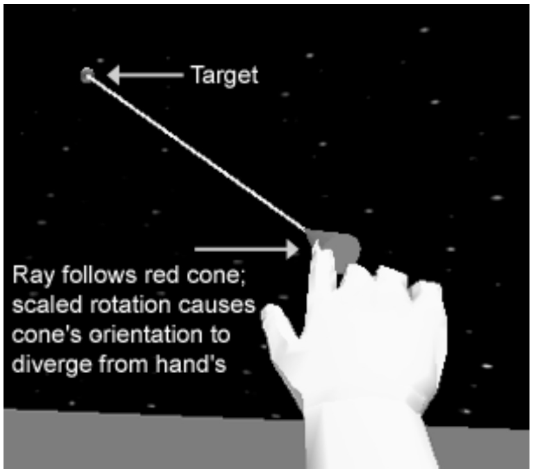


Fig. 13. Ray Casting with PRISM.

object to move it inside the target, just as we saw in the rotation-only experiment. For some participants there was even more confusion due to the translation component of the task. In situations where a small rotation would have moved the object inside the target, many participants instead tried to translate it repeatedly. Once again, this type of confusion favored direct manipulation; accidental movement of the object at least gave the user the chance of moving it in the correct direction; when using PRISM this accidental movement was filtered out. Despite this issue, PRISM outperformed direct manipulation for the more difficult (smaller) targets, and we believe that the gap between PRISM and direct manipulation's effectiveness is larger than the results suggest. The increase in button clicks (guessing) indicates to us that the user did not feel confident in their control over the object when using direct manipulation. This is precisely the type of behavior PRISM is trying to remedy.

5.4 Ray Casting with PRISM Rotation

We implemented ray casting by drawing a white line extending out from a small red cone attached to the user's stylus. To aid in targeting, a red sphere was placed at the far end of the ray (far beyond the targets). When the ray intersected the target, the far end of the ray was attached to the object and the sphere in the distance turned green to indicate that the target could be selected with a button press. When using direct manipulation, the red cone attached to the stylus always pointed in the same direction as the stylus; when using PRISM, the rotation of the red cone was scaled proportionally to the rotational speed of the stylus, which can be seen in Figure 13. In pilot studies, it was determined that the scaling constant did not need to be as high as it was for direct object rotation—the PRISM implementation of ray casting used a scaling constant of 20 degrees/second.

This experiment required users to select small, far away objects as quickly and as accurately as possible. Each trial presented the user with a series of 15 targets that "exploded" when selected. Users were given a starting value of

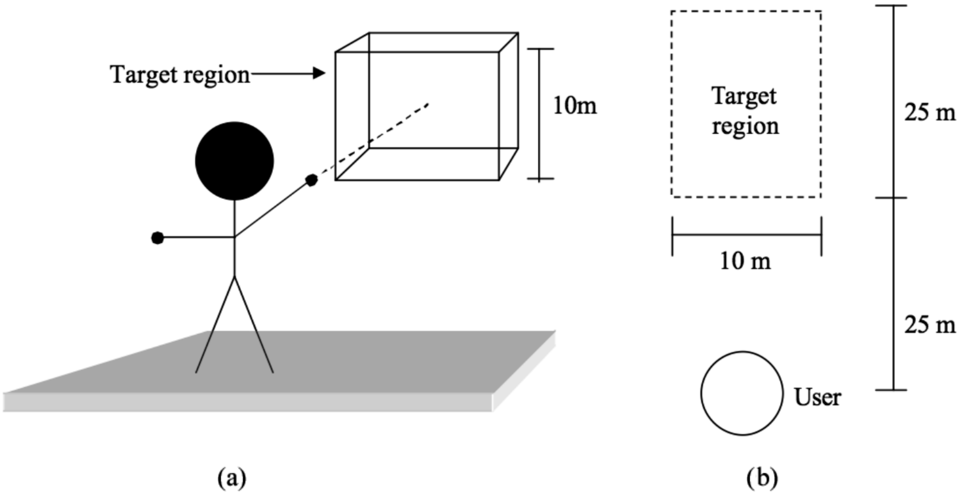


Fig. 14. (a) Perspective view of ray casting test environment. Both the floor and the base of the target region are at y = 0 height in world coordinates. (b) Top view of ray casting test environment.

3000 points; their points decreased at a rate of 10 points per second. In addition, each time the user pressed the stylus button but *missed* the target they lost 200 points. The point system was explained to the user prior to the experiment and they were instructed that in order to maximize their points for each trial (a trial was complete once all 15 targets were selected) they needed to value accuracy over speed. Each participant completed four trials during the experiment, two trials using PRISM ray casting and two with directly controlled ray casting. The participants alternated between direct and PRISM. There were a total of 12 participants (1 female). Half of the participants began with direct, half with PRISM. Each target was a 30 cm cube and was positioned within a $10 \times 10 \times 25$ meter (x, y, z) region centered at (0, 0, −37.5) in world coordinates, as shown in Figure 14. The participants began the experiment at world coordinate (0, 0, 0) looking down the negative z-axis. They were told in advance where the target region was (relative to their position) and the training session allowed them to become accustomed to the layout of the environment. The target sizes and the set of positions within the target region remained constant for each trial and for each participant. To avoid search time becoming a factor, the user was presented with one target at a time. Each time a target was shown, it first became visible directly in front of the user and then was slowly moved to its final position. Once the target reached its position a sound played indicating the user was permitted to select it. Upon selection, the next target would become visible and the procedure was repeated. The backdrop of the virtual environment was a star field (the white spots in Figure 13 are the stars), which helped increase the users' situational awareness and depth perception.

5.4.1 *Results.* We performed an ANOVA (Table VIII) on the data with completion time and error rate (misses/total button clicks) as the dependent variable. The independent variables included Interaction Type (direct or PRISM). In addition, we also looked at the trial number corresponding to the

Table VIII. ANOVA Tables for Ray Casting Experiment

Ray Casting Experiment Error Rate					
Source	ss	DoF	MS	F Ratio	P Value
Interaction Type (I)	1.6990	1	1.6980	187.23	0.00
Order (O)	0.0020	1	0.0020	0.27	0.61
Participant (P)	0.0920	11	0.0080		
I * O	0.0000	1	0.0000	0.00	0.98
O * P	0.0800	11	0.0070		
P * I	0.1000	11	0.0090		
I * O * P	0.0490	11	0.0040		
Error	*	0			
Total	2.0220	47			

Ray Casting Experiment Completion Time					
Source	ss	DoF	MS	F Ratio	P Value
Interaction Type (I)	13937.90	1	13937.90	28.89	0.00
Order (O)	0.19	1	0.19	0.00	0.97
Participant (P)	12681.49	11	1152.56		
I * O	144.40	1	144.40	0.67	0.43
O * P	1439.87	11	130.90		
P * I	5305.91	11	482.45		
I * O * P	2382.56	11	215.60		
Error	*	0			
Total	35893.32	47			

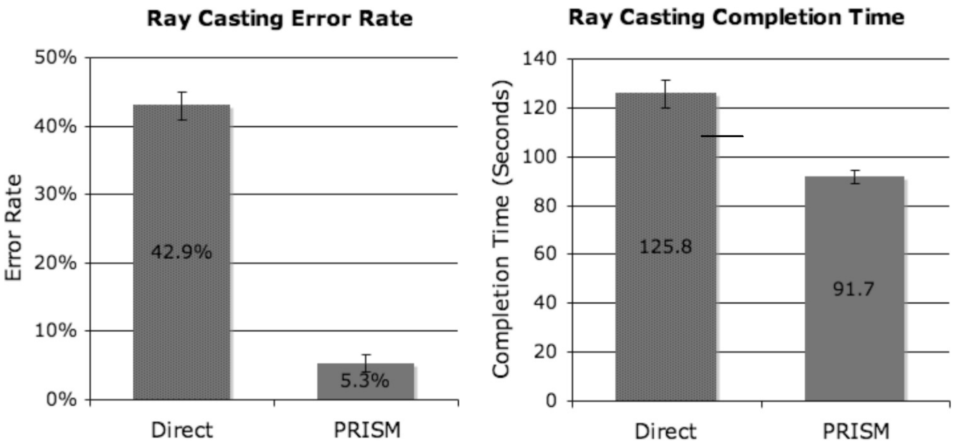


Fig. 15. Performance results for Ray Casting experiment. Error bars represent standard error.

technique used (1 if it was the first trial using direct or PRISM, 2 if it was the second trial using the respective technique). Overall, our results show that PRISM was extremely effective in increasing the accuracy and decreasing completion time when using ray casting with distant targets. Interaction type was highly significant for both Completion Time (F(1,11) = 28.89, p < 0.001) and Error Rate (F(1,11) = 187.23, p < 0.001) and there was no evidence of learning effects between trials. These results are summarized in Figure 15.

5.4.2 *User Feedback.* Most participants thought PRISM was more effective than the direct form of ray casting, with 11 of 12 participants favoring PRISM. PRISM ray casting also received 3.75 out of 5 when asked how easy it was to learn. A common problem with ray casting and precision is the slight movement of the wrist when the user attempts to press the stylus button—an example of the “Heisenberg Effect” described in Bowman et al. [2001]. This problem has also been recognized by Gerber and Bechmann [2005] as a factor limiting the number of menu items that can be placed in their spin menu and was echoed by our participants. Many participants told us that whenever they pressed the button their hand would move slightly and cause them to miss when using direct ray casting. These participants told us that this problem was almost completely eliminated during the PRISM trials and that they felt like the ray “locked into place”.

5.4.3 *Discussion.* The results of this experiment show that using PRISM with ray casting is extremely effective. PRISM solves a fundamental problem with ray casting, the inability to select small, distant objects. Carrying this further, it is likely to solve the equally important problem of not being able to accurately *move* objects from a distance. The problems with ray casting are caused by the fact that small rotations of the wrist result in large translations at the end of the ray, where the cursor or an object is being moved. This worsens as the length of the ray increases. PRISM works to counteract this by scaling down the rotation of the hand or source object (the red cone in our implementation).

We also noticed that using PRISM for ray casting encouraged more comfortable hand positions and movements. We observed that when users had difficulty targeting objects with the directly controlled ray casting they tended to point outwards with their arms, sometimes using two hands much like they would when aiming a pistol. A more experienced user would normally “shoot from the hip” instead, which is much less fatiguing. PRISM encouraged the user to hold their hand in a comfortable position and move the ray only by rotating their wrist. This was partly due to the increased ability to target objects, however PRISM was also easier to use when the wrist was the primary joint involved. Sweeping out arcs with the shoulder or elbow is in general a much slower movement than a wrist rotation and this slower rotation forces PRISM into scaling mode. Once users realized they could take advantage of PRISM’s modal switching more effectively when rotating their wrist quickly to cover larger angular distance they became much more effective. Of course, when designing a technique that requires the use of the shoulder or elbow to rotate something the Scaling Constant could be decreased to provide direct control when moving more slowly; however, for traditional ray casting, the Scaling Constant used in this experiment was a good choice.

6. FUTURE DIRECTIONS

A short-term issue we would like to further investigate is PRISM translation, specifically the axis-independent scaling technique. Axis-independent scaling offers a distinct advantage over uniform scaling; it helps the user move an object in a straight line or along a plane by eliminating drift along the axis of movement in which movement is slowest. Unfortunately, performance is slowed for diagonal movement, seen in the tasks with rotated target cubes in our user study. In the future, we plan to implement a version of PRISM that scales translation uniformly in all directions by using the absolute speed of the hand to determine one true CD ratio.

In the long term, we would like to further investigate PRISM's applicability to other tasks and with a number of user interface widgets. As described in our user study, PRISM rotation can enhance selection with ray casting significantly. To follow this up, we would like to work with user interface controls such as menus and buttons. We believe that by using a PRISM controlled cursor we can increase "touch selection" accuracy and thus decrease the size of the typical widgets used in VR. In addition, the usability of a World in Miniature (WIM) may be significantly enhanced through PRISM manipulation and we plan on running experiments focused on this.

7. CONCLUSIONS

In the physical world, we move and rotate objects quite accurately by using friction and the object's inertia to help steady our movements. In the virtual world, none of these physical properties normally exist, and object manipulation is negatively affected by hand instability. We have shown in a series of user studies that PRISM increases precision and allow users to make purposeful and fine-grained adjustments to the position and orientation of an object in a timely manner. When examining translation and rotation separately, there was a statistically significant increase in performance when using PRISM over direct manipulation. With PRISM rotation and translation combined (6DOF task) results were less clear, however PRISM continued to out perform direct manipulation for trials requiring very high precision. Accuracy was increased in these tasks because PRISM successfully filters out a significant amount of hand instability; however, in situations where there is absolutely no tolerance for error, other techniques might be more appropriate. Unlike many other techniques aimed at increasing precision, PRISM does not place any limits on the granularity in which the user can work with objects and does not over-constrain interaction when precision is not required. By utilizing the principle of Fitts' Law, PRISM dynamically adjusts the control/display ratio without explicit user intervention to provide extra precision when moving slowly and direct, unconstrained manipulation when the hand is moving quickly. We believe that the natural control metaphor used by PRISM is also what makes it so easy to learn. We have shown that PRISM can also apply to the control of a 3D cursor. In particular, we have shown that a PRISM-enhanced version of ray casting can significantly increase the speed in which objects (or buttons and controls) can be selected and dramatically decreases miss rates. PRISM is applicable across a wide variety of tasks in immersive virtual environments wherever the hand directly controls the position and orientation of the cursor. It is our hope that this increase in precision will allow designers to increase the complexity of their virtual worlds without suffering from a manipulation and interaction "bottleneck".

REFERENCES

ACCOT, J. AND ZHAI, S. 1997. Beyond Fitts' law: Models for trajectory-based HCI tasks. In *Proceedings of SIGCHI Conference on Human factors in Computing Systems*. ACM, New York, pp. 295–302.

ALBINSSON, P.-A. AND ZHAI, S. 2003. High precision touch screen interaction. In *Proceedings of SIGCHI Conference on Human factors in Computing Systems*. ACM, New York, pp. 105–112.

BEDERSON, B. B., MEYER, J., AND GOOD, L. 2000. Jazz: An extensible zoomable user interface graphics toolkit in Java. In *Proceedings of ACM Symposium on User Interface Software and Technology*. ACM, New York, pp. 171–180.

BEIR, E. A. 1990. Snap-dragging in three dimensions. In *Proceedings of ACM Symposium on Interactive 3D Graphics*. ACM, New York, pp. 193–204.

BLANCH, R., GUIARD, Y., AND BEAUDOUIN-LAFON, M. 2004. Semantic pointing: Improving target acquisition with control-display ratio adaptation. In *Proceedings of SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, pp. 519–526.

BOWMAN, D. AND HODGES, L. 1997. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics*. pp. 35–38.

BOWMAN, D. AND WINGRAVE, C. 2001. Design and evaluation of menu systems for immersive virtual environments. In *Proceedings of IEEE Virtual Reality*. IEEE Computer Society Press, Los Alamitos, CA, pp. 149–156.

BOWMAN, D., WINGRAVE, C., CAMPBELL, J., AND LY, V. 2001. Using pinch gloves for both natural and abstract interaction techniques in virtual environments. In *Proceedings of HCI International*. pp. 629–633.

BUKOWSKI, R. AND SEQUIN, C. 1995. Object associations: A simple and practical approach to virtual 3D manipulation. In *Proceedings of SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, pp. 131–138.

CHEN, M., MOUNTFORD, S. J., AND SELLEN, A. 1988. A Study in interactive 3D rotation using 2D control devices. In *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques*. pp. 121–129.

DOMINJON, L., LECUYER, A., BURKHARD, J., RICHARD, P., AND RICHIR, S. 2005. Influence of control/display ratio on the perception of mass of manipulated objects in virtual environments. In *Proceedings of IEEE Virtual Reality*. IEEE Computer Society Press, Los Alamitos, CA, pp. 19–25.

FITTS, P. M. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *J. Exper. Psych. 47*, 381–391.

FOLEY, J. D., WALLACE, V., AND CHAN, P. 1984. The human factors of computer graphics interaction techniques. *IEEE Comput. Graph. Appl. 11*, 4, 13–48.

FREES, S. E. AND KESSLER, G. D. 2005. Precise and rapid interaction through scaled manipulation in immersive virtual environments. In *Proceedings of IEEE Virtual Reality*. IEEE Computer Society Press, Los Alamitos, CA, pp. 99–106.

GERBER, D. AND BECHMANN, D. 2005. The spin menu: A menu system for virtual environments. In *Proceedings of IEEE Virtual Reality*. IEEE Computer Society Press, Los Alamitos, CA, pp. 271–272.

GROSJEAN, J. AND COQUILLART, S. 2001. Command & control cube: A shortcut paradigm for virtual environments. In *Proceedings of Immersive Projection Technology and Virtual Environments*. pp. 1–12.

HINCKLEY, K., PAUSCH, R., GOBLE, J., AND KASSELL, N. F. 1994a. A survey of design issues in spatial input. In *Proceedings of ACM Symposium on User Interface Software and Technology*. ACM, New York, pp. 213–222.

HINCKLEY, K., PAUSCH, R., GOBLE, J., AND KASSELL, N.F. 1994b. Passive real-world interface props for neurosurgical visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, pp. 452–458.

HINCKLEY, K., TULLIO, J., PAUSCH, R., PROFFITT, D., AND KASSEL, N. 1997. Usability analysis of 3D rotation techniques. *Proceedings of ACM Symposium on User Interface Software and Technology*. ACM, New York, pp. 1–10.

IGARASHI, I. AND HINCKLEY, K 2000. Speed-dependent automatic zooming for browsing large documents. In *Proceedings of ACM Symposium on User Interface Software and Technology*. ACM, New York, pp. 129–148.

ISHII, H. AND ULLMER, B. 1997. Tangible bits: Towards seamless interfaces between people, bits and atoms. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, pp. 234–241.

KESSLER, G. D., BOWMAN, D., AND HODGES, L. 2000. The simple virtual environment library, an extensible framework for building VE applications. In *Presence: Teleoperators and Virtual Environments 9*, 2, 187–208.

LIANG, J., SHAW, C., AND GREEN, M. 1991. On temporal-spatial realism in the virtual reality environment. In *Proceedings of ACM Symposium on User Interface Software and Technology*. ACM, New York, pp. 19–25.

LINDEMAN, R., SIBERT, J., AND HAHN, J. 1999. Towards usable VR: An empirical study of user interfaces for immersive virtual environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, pp. 64–71.

MAPES, D. P. AND MOSHELL, J. M. 1995. A two-handed interface for object manipulation in virtual environments, *Presence: Teleoperators and Virtual Environments 4*, 4, 403–416.

MINE, M. 1995. Virtual environment interaction techniques. University of North Carolina Computer Science Tech. Rep. TR95-018.

MINE, M., BROOKS, F., AND SÉQUIN, C. 1997. Moving objects in space: Exploiting proprioception in virtual-environment interaction. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. pp. 19–26.

PARSONS, L. 1995. Inability to reason about an object's orientation using an axis and angle of rotation. *J. Exper. Psych.: Human Percep. Perf. 21*, 6, 1259–1277.

PIERCE, J., FORSBERG, A., CONWAY, M., HONG, S., ZELEZNIK, R., AND MINE, M. 1997. Image plane interaction techniques in 3D Immersive Environments. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics*. pp. 39–44.

PIERCE, J. AND PAUSCH, R. 2002. Comparing voodoo dolls and HOMER: Exploring the importance of feedback in virtual environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, pp. 105–112.

PIERCE, J., STEARNS, B., AND PAUSCH, R. 1999. Two-handed manipulation of voodoo dolls in virtual environments. In *Proceedings of the 1999 Symposium on Interactive 3D Graphics*. pp. 141–145.

POUPYREV, I., BILLINGHURST, M., WEGHORST, S., AND ICHIKAWA, T. 1996. The go-go interaction techniques: Non-linear mapping for direct manipulation in VR. In *Proceedings of ACM Symposium on User Interface Software and Technology*. ACM, New York, pp. 79–80.

POUPYREV, I., WEGHORST, S., BILLINGHURST, M., AND ICHIKAWA, T. 1997. A framework and testbed for studying manipulation techniques for immersive VR. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*. ACM, New York, pp. 21–28.

POUPYREV, I., WEGHORST, S., AND FELS, S. 2000. Non-isomorphic 3D rotational techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, pp. 540–547.

RUDDLE, R., SAVAGE, J., JONES, D. 2002. Evaluating rules of interaction for object manipulation in cluttered virtual environments, Presence*: Teleoperators and Virtual Environments 11*, 6, 591–609.

SHOEMAKE, K. 1985. Animating rotation with quaternion curves. In *Proceedings of the 12th Conference on Computer Graphics and Interactive Techniques*. pp. 245–254.

SHOEMAKE, K. 1992. ARCBALL: A user interface for specifying three-dimensional orientation using a mouse. In *Proceedings of the Conference on Graphics Interface*. pp. 151–156.

STOAKLEY, R., CONWAY, M., AND PAUSCH, R. 1995. Virtual reality on a WIM: Interactive worlds in miniature. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, pp. 265–272.

STUERZLINGER, W., AND SMITH, G. 2002. Efficient manipulation of object groups in virtual environments. In *Proceedings of IEEE Virtual Reality*. IEEE Computer Society Press, Los Alamitos, CA, pp. 251–258.

TAN, D. S., ROBERTSON, G. G., AND CZERWINSKI, M. 2001. Exploring 3D navigation: Combining speed-coupled flying with orbiting. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, pp. 418–424.

WARE, C., 1990. Using hand position for virtual object placement. *The Visual Computer: Int. J. Comput. Graph. 6*, 5, pp. 245–253.

WARE, C. AND ROSE, J. 1999. Rotating virtual objects with real handles, *ACM Trans. Comput. Human Interact. 6*, 2, pp. 162–180.

WELCH, G. AND BISHOP, G. 1997. SCAAT: Incremental tracking with incomplete information. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. pp. 333–344.

ZHAI, S., MILGRAM, P., AND BUXTON, W. 1996. The influence of muscle groups on performance of multiple degree of freedom input, In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* ACM, New York, pp. 308–315.

Received June 2005; revised January 2006; accepted May 2006 by Alistair Sutcliffe