Precise and Rapid Interaction through Scaled Manipulation in Immersive Virtual Environments

Scott Frees *
Lehigh University

G. Drew Kessler t
Lehigh University

ABSTRACT

A significant benefit of an immersive virtual environment is that it provides users the ability to interact with objects in a very natural, direct way; often realized by using a tracked, hand-held wand or stylus to "grab" and position objects. In the absence of force feedback or props, it is difficult and frustrating for users to move their arms, hands, or fingers to precise positions in 3D space, and more difficult to hold them at a constant position, or to move them in a uniform direction over time. The imprecision of user interaction in virtual environments is a fundamental problem that limits the complexity of the environment the user can interact with directly.

We present PRISM (Precise and Rapid Interaction through Scaled Manipulation), a novel interaction technique which acts on the user's behavior in the environment to determine whether they have precise or imprecise goals in mind. When precision is desired, PRISM dynamically adjusts the "control/ display" ratio which determines the relationship between physical hand movements and the motion of the controlled virtual object, making it less sensitive to the user's hand movement. In contrast to techniques like Go-Go, which scale up hand movement to allow "long distance" manipulation; PRISM scales the hand movement down to increase precision. We present the results of a user study which shows that PRISM significantly out-performs the more traditional direct manipulation approach.

CR Categories:
I.3.6 [Computer Graphics] Methodology and Techniques - Interaction Techniques; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism - Virtual reality.]

Keywords:
direct manipulation, precise manipulation

1 INTRODUCTION

Perhaps the most inherent and appealing aspect of immersive virtual environments (IVE) is their ability to allow users to interact with a simulated environment in the same way they do in the real world. Along with the increased sense of presence, direct interaction and natural viewing of the virtual world have the potential to increase productivity in tasks involving interaction in 3D, particularly 3D design and modeling tasks. Unfortunately this potential has not yet been realized. One of the obstacles to reaching this potential is the limited precision in which users interact with the virtual world, severely limiting the complexity of the virtual environment with which a user can effectively and intuitively interact.

* 19 Memorial Dr. West, Bethlehem PA, 18015, sef3@lehigh.edu
t 19 Memorial Dr. West, Bethlehem PA, 18015, gdk2@lehigh.edu

While tracking devices continue to have inherent error and/or jitter, a more general problem is our own inability to move our bodies (e.g. our hands and arms) precisely through 3D space without some form of additional support. While our bodies are quite flexible and capable of very complex movements, we are unable to hold our arms or hands at constant, precise positions. Due to the rotational nature of our joints, it is difficult for us to move virtual objects in straight lines or along planes without the presence of force feedback. Although current research in haptic interfaces is promising, a fully haptic interface for virtual reality remains a distant (and expensive) prospect.

In many 2D applications such as CAD programs, users often perform precise manipulation using indirect approaches (menus, buttons, etc.). Although indirect techniques can be used in a 3D design application, menus can be difficult for users to work with in immersive environments (3D menus continue to be a challenging research topic [6]). In addition to being easier to learn, a more direct approach avoids using input buttons and screen space which are often more scarce in immersive systems than in their desktop counterparts.

Implementing a 3D direct manipulation interface which allows for precise positioning and fine grain adjustment of virtual objects is a challenging design problem. Such an interface must "effectively integrate rapid, imprecise, multiple degree-of-freedom object placement with slower, but more precise object placement, while providing feedback that makes it all comprehensible" [10]. This design challenge remains regardless of the display type, tracking system, or any other hardware configuration used; it is relevant whenever 3D direct interaction is employed. To meet this challenge we have implemented PRISM (Precise and Rapid Interaction through Scaled Manipulation), which acts on the user's behavior in the environment to determine whether they have precise or imprecise goals in mind. Users do not need to explicitly switch to a new mode if they are attempting to be precise and the interface does not inhibit the user from making rapid, imprecise translations by over-constraining manipulation. Switching between precise and direct mode occurs seamlessly during natural interaction according to the current velocity of the user's hand. Precision interaction is provided through scaled manipulation, which dynamically adjusts the "control/ display" ratio that determines the relationship between physical hand movements and the motion of the controlled virtual object. This makes the control selectively less sensitive to hand movement.

As detailed below, there is an important tradeoff between precision and sensitivity (or responsiveness). Increasing precision by scaling down physical hand movements causes the controlled object to be less sensitive to hand movements, effectively reducing the size of the working space. On the other hand, increasing the controlled object's sensitivity to hand movement to provide for direct, rapid motion and a large working volume can only be done at the cost of precision. Effective mode switching, therefore, is critical for users trying to perform tasks that require precision and responsiveness.

The next section discusses some of the current interaction techniques in use today and how PRISM can work along side these techniques. Then we will more formally describe the interface and present an analysis of our user study. We conclude with our plans for future work.

2 Related Work

The dynamic adjustment of the control/display ratio (as defined in [9]) that PRISM uses has its roots in 2D mouse based interfaces, where a good deal of work has gone into making these interfaces easy to use. Mice are commonly configured to cover more pixel space when moved at high speeds and less when moved across the user's desk slowly. This divergence between physical movement and cursor translation is usually imperceptible to users since they rarely focus on the cursor and the physical mouse movements simultaneously. Accumulated divergence between physical and screen position is intuitively reconciled by simply lifting the mouse and moving it to a more comfortable position, essentially using a clutching mechanism. Accommodating for the adjustment of the control/display ratio becomes less straightforward when implemented in a 3D direct input interface, where the user's hand is directly represented in the virtual world. PRISM works to reconcile this divergence in an intuitive and seamless manner, as described in the next section. Other techniques that adjust the control/display ratio have been developed to provide overview vs. detailed modes while navigating through documents [12] and 3D worlds [27] on the desktop. Zoomable User Interfaces also adjust the control/display ratio in order to provide a better user experience [2]. Although similar to PRISM in concept, these systems involve indirect manipulation, where reconciling offset/divergence between the physical device or hand is a more straightforward task.

Another related technique which stems from 2D interfaces is the concept of zooming in or scaling up the workspace in order to work in small areas and make very fine adjustments to objects of interest. Although this technique works well for some 2D applications it suffers from a loss of context. When scaling up small parts of the workspace the user looses their sense of where they are working in the context of the overall space. This problem is exacerbated in a 3D environment where, as the workspace is scaled up, occlusion (objects obstructing the user's view of the object they are most interested in) becomes a more severe issue.

Interaction in 3D environments, whether through a 2D desktop display or in immersive virtual environments, has been a well studied problem. One technique used to position objects (or a cursor) precisely is the snap dragging technique [3], which was implemented as part of a 2D interface into a 3D world. This technique constrains the object being manipulated to grid lines or other objects in the world. When using the snapping technique with grid lines, objects can be placed at somewhat arbitrary locations in the world – however the precision of these placements is limited to the resolution of the grid. Other work [17] provides precise alignment by using anchors, constraints and "glue" to help the user snap and align objects together. Although these interfaces allow for precise positioning of objects relative to others, they do not provide for additional support for fine grain adjustments in object positions; snap points and alignments are pre-determined by the environment's designers.

Several techniques have been developed which allow the user to create or use existing nearby objects to help with placement tasks. Voodoo Dolls [20, 21], allows users to effectively scale their workspace (and thus scale their movements) by selecting voodoo dolls of appropriate sizes. For instance, holding two dolls representing very small objects in the world scales down the user's movements, while using a large reference object scales movements upwards. 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 [7]. In these systems objects such as walls constrain other objects, such as picture frames, to always lie along a particular surface. This type of technique can certainly help a user place a virtual picture frame directly on a wall; however it does not provide any additional support for placing the frame at a particular *place* on the wall. Although techniques that use object associations and reference objects have certainly been shown to aid in manipulation, they still require some form of direct interaction on an object (or reference object). This direct interaction component is where PRISM can be useful, possibly working in conjunction with these techniques.

Another area of research concerning object manipulation has focused on selecting and interacting with objects at a distance, beyond arms reach. A common technique has been ray-casting. However, other more powerful and productive techniques have been developed, such as Scaled-world grab[18], Image Plane interaction [19], Go-Go [22], HOMER [4], and World in Miniature (WIM) [26]. These techniques allow users to interact directly with distant objects; PRISM is focused on direct manipulation and can be viewed as a technique which can work in conjunction with the above techniques to increase precision. Other techniques have been proposed which address the lack of force feedback by making objects resistant to movement, such as Ruddle's concept of using virtual inertia [25] in cluttered virtual environments. Where virtual inertia is aimed at limiting the rate at which an object can be moved or rotated in order to reduce collisions, PRISM places no limit on the rate at which users can move objects; the speed of the object is governed by the interpreted user intentions. Many of the discussed techniques lend themselves well to various interaction tasks and have been reviewed and classified into taxonomies by a number of researchers [5] [23].

Although our research is focused primarily on virtual environments which do not incorporate force feedback there is a sizable set of IVE applications which lend themselves well to using physical props to aid and constrain interaction. The use of props in neurosurgery applications [11], menu interaction, and object manipulation [16] have been shown to be effective. Making even further use of props is Ishii's concept of tangible user interfaces [13], which aim to completely integrate the virtual (or cyber) world with real world objects. Although there is little doubt that manipulating physical props helps in cognition and performance, these are not general solutions to the precision problem and are not always readily available to the designer.

3 PRISM

In our implementation of *direct manipulation*, an interaction is initiated when a user intersects a virtual object with the hand-held stylus and holds down the stylus button. While the user holds the button, the position of the virtual object directly follows the stylus. As described in the introduction, a specific design challenge for a direct manipulation interface in 3D is to provide the user with the ability perform deliberate, precise, fine grain adjustments to the position of objects, along with the ability to move objects quickly from one general position to another. In order for such an interface to be efficient, it must allow the user to switch between these two modes of interaction without explicitly interacting with a menu, using key combinations, or performing any other action which disrupts their natural flow of interaction.

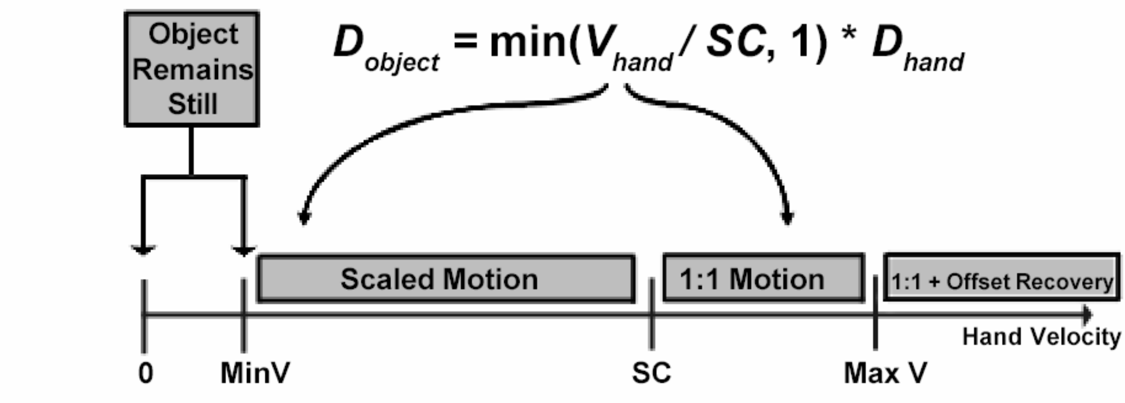

Figure 1: Interaction modes provided by PRISM, based on the current speed to the hand or cursor. Default Min V = 0.05 m/sec, Max V = 0.25 m/sec determined through pilot studies.

3.1 Interface Design

In PRISM, users indicate whether they are trying to be precise or rapid through a very natural control metaphor, their hand speed. PRISM uses the hand speed while interacting to implicitly switch between precision and non-precise modes. The underlying principle of PRISM is that users manipulate objects at different speeds depending on their intentions. When moving an object from one general place to another, the user is not necessarily interested in being precise and moves relatively rapidly. When users are focused on accurately moving an object to very specific locations, they normally slow their hand movements down and focus more on being precise. In order to determine which mode the user is currently in we use three constants which relate to the current velocity of the user's hand while interacting with an object. Figure 1 shows the different modes of the interface. The first constant is a minimum velocity (MinV), which is a very low velocity in which a user is unlikely to move purposefully. Any motion below this velocity is most likely tracking error or inadvertent drift and the controlled object is held still.

The second and most important constant is the Scaling Constant (SC), which is a relatively low velocity. If the user is moving their hand slower than this value they are likely to have a precise goal in mind. While in this mode we begin scaling the motion of the controlled object in order to give the user increased precision. During scaled manipulation, we apply a simple function to the position of the hand (See Figure 1) to determine how far to move the virtual object. This function is based on the current hand velocity, distance moved by the hand, and SC. The object is then moved by the resulting distance. For example, if the user is moving their hand at a slow speed, close to the MinV, the object would hardly move, however if the hand velocity was closer to SC, the controlled object might move 90% of the distance the hand has moved during the most recent sampling interval.

Implicit in this method is the accumulation of an offset value representing the distance between the hand and the object being manipulated (represented by the white line between the user's hand and the sphere in Figure 2). Each time movement is scaled, the virtual object will only move a fraction of the distance the hand does. We have implemented two mechanisms to recover this offset without interrupting the user's interaction, which are described in Table 1. 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 also guards against unintended hand movements.

Any time the user is moving their hand at a velocity above SC they are transitioned into imprecise, direct manipulation mode, where for each movement of their hand the object will move an equal distance. A third constant, MaxV triggers the second mechanism used to reclaim any offset accumulated during scaled interaction. Once the user's hand velocity exceeds MaxV the offset is reduced gradually so that the object appears to be catching up to the hand (see Table 1 for details on offset recovery). By doing this smoothly over a period of time, we allow the user some time to realize they have switched to direct manipulation and to slow down if this isn't what they intended. As shown in Figure 1, this MaxV is typically larger than the scaling constant (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.

Table 1: Offset Recovery Procedures

Offset Recovery Procedures	
Conditions	**Method**
Any mode (direct or scaled)	Moving the cursor (hand) back towards the object in the opposite direction of the offset results in no object movement, thus reducing accumulated offset.
Cursor is moving faster than MaxV	Offset is reduced over time, starting as soon as the speed threshold is met and continuing until the offset is eliminated or the hand speed falls below MaxV. Offset is reduced on each frame according to the psuedocode below.* let T = time MaxV threshold is 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 which made a frame based recovery feasible. A more general approach would be to reduce offset strictly based on time.	

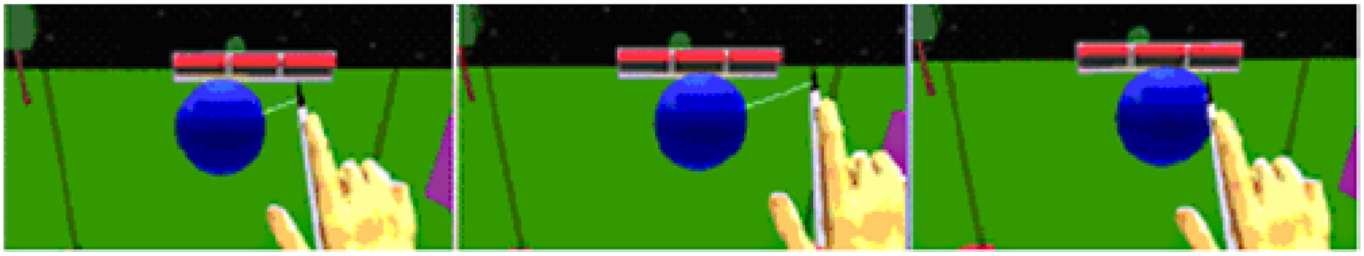

Figure 2: Example of offset recovery with PRISM

One implementation detail of note is that PRISM operates on each axis (x, y, and z) independently. For instance, the hand velocity 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 in the X direction and retain direct control while simultaneously being in precision (scaled mode) in the Y and Z direction, eliminating inadvertent drift. We could have easily implemented PRISM such that it used the Euclidean velocity 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 reader should note that regardless of whether scaling is calculated independently for each axis or not, the user can still move objects in any direction; scaling on each axis independently merely provides added benefit when moving an object along a principal axis or plane (this is demonstrated in our user study, described below).

3.2 Interface Walkthrough

Figure 3 presents a few typical interaction sequences using PRISM. For simplicity, the examples are illustrated in 2D. In Figure 3a, 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 effecting the movement in the horizontal direction. In Figure 3b, the user moves their hand in the same exact manner as in Figure 3a, 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 3c the user starts out with an offset in the vertical direction (from some previous interaction sequence) and then moves slowly up back 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. Finally, in Figure 3d, the user starts off with an offset in both the horizontal and vertical directions. In this sequence the user moves their hand quickly away from the ball. Since the hand motion is faster than MaxV, the offset is recovered in both directions as the controlled object "catches up" to the user's hand.

3.3 Sensitivity v.s Precision

Although PRISM allows for more precise positioning and manipulation, it isn't a "one size fits all" strategy. There exists an important relationship between the scaling constant and the sensitivity and precision the user experiences, which is shown in Figure 4. Through observation it is evident there are different "optimal" scaling constants for different individuals. Some users naturally move their hands more slowly and steadily than others. When using a generic scaling constant, these users sometimes find the interaction technique frustrating and unhelpful; but with some adjustment of SC, the technique is easily improved to the user's satisfaction. On the other hand, some users have a lower degree of dexterity and have particular difficulty keeping their hands steady and moving them in a slow, purposeful way – which is required to make fine grain adjustments to the position of the object of interest. 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.

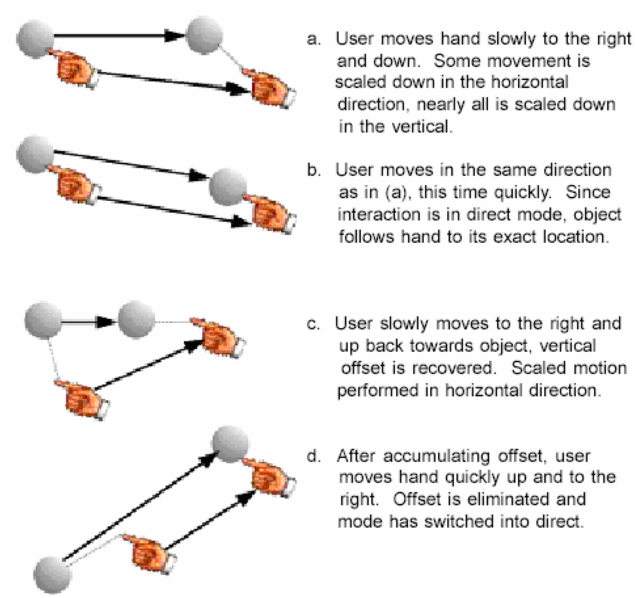

Figure 3: Typical interaction sequences when using PRISM.

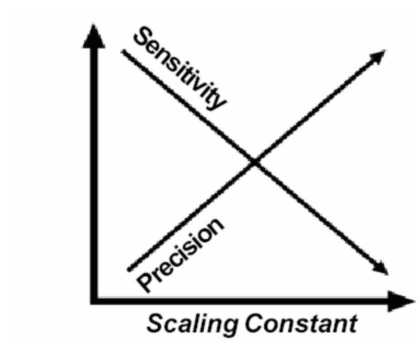

Figure 4: Tradeoff between sensitivity and precision. As the Scaling Constant gets larger, the controlled object becomes less sensitive to small hand movements, giving the user more precision.

4 User Study

In order to evaluate PRISM as an effective interaction technique we conducted a user study which had participants perform a series of simple tasks requiring varying degrees of precision. The task was to pick up a virtual sphere and place it completely inside a virtual cube such that no part of the sphere protruded any side of the cube. We chose this task for its generality; it requires the user to place the sphere as close to a specific location in the world as possible. Each time the user positioned the sphere completely inside the target the cube changed color to indicate a completion. A short time later the sphere re-appeared in its original starting position outside the cube. The experiment consisted of 6 trials, each lasting three minutes. The participants were asked to repeatedly place the virtual sphere inside the cube as many times as possible during the three minutes (the short time between completions was not counted towards their time).

In each trial the target cube became smaller with each completion, starting off with an "Easy" difficulty level and progressing to the fourth target in which all subsequent cubes were "Very Difficult". The gradual reduction in the size of the cube made the task more difficult (since the sphere remained the same size) as each trial progressed up to a certain level. 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, a problem identified in pilot studies. The difficulty levels the user experienced is summarized in Table 2 below. 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 within a trial). All target cubes were placed at shoulder height (25cm below top of HMD) in the same position in virtual world for all participants, as to eliminate the possibility of the distance from the target effecting the difficulty level [1, 8]. The starting point for the spheres was approximately 1 meter away from the target. Since the object was placed at this distance, most users chose to move the sphere rapidly towards the target and then slowed down only to make small adjustments (which triggers the modal switching of PRISM in trials using that technique).

The experiment consisted of 2 factors; the first factor was the interaction technique, which was at three levels: Direct, PRISM with Generic Scaling, and PRISM with Custom Scaling. The second factor was target orientation, where all target cubes were either axis-aligned or rotated 45° in each principle axis.

Table 2: Cubes were larger than the sphere; error tolerance is the length of cube – diameter of the sphere.

	Difficulty Level	Error Tolerence
1st Target	Easy	1.00cm
2nd Target	Moderate	0.35cm
3rd Target	Difficult	0.20cm
4th Target +	Very Difficult	0.12cm

We chose to vary the orientation of the targets to examine how the scaling functions were affected by movement which was less likely to be along the principle world axes. Although the orientation of the target has no theoretical effect on the difficulty (the sphere still needs to be placed at the center of the cube), users have a tendency to move the sphere diagonally towards the rotated target instead of along a principal axis. We hoped to compare the use of both rotated and axis aligned targets to draw some conclusions about the axis aligned scaling strategy (discussed in Section 3.1). The difference in appearance between the types of targets is shown in Figure 5 and 6.

In order to determine what scaling constant to use during the Custom Scaling trials, we held a pre-experiment training session. Users were given the opportunity to get familiar with direct interaction and PRISM and try out five different scaling values (Lowest, Low, Normal, High, and Highest, see Table 4); the higher the scaling value was, the less sensitive the object was to the user's hand movements. During this practice trial, the target cube was set at axis aligned. Participants were given as much time as they needed to determine the scaling value they preferred.

We conducted a within-subject experiment consisting of 18 (14 male, 4 female) undergraduate and graduate students. Four of the participants had previous experience in an immersive virtual environment. The order in which the participants conducted the trials was balanced as to eliminate any learning effects. The equipment used included a four-port Polhemus 3SPACE FASTRAK electromagnetic tracking system with a hand-held stylus, a Virtual Research Systems V8 Head Mounted Display, and a Pentium 4 PC. The software used to drive the virtual environment was the Simple Virtual Environment toolkit [15] and our interaction techniques were implemented as interactor components using the SVIFT toolkit [14]. Throughout the experiment the frame rate of the system was around 30fps.

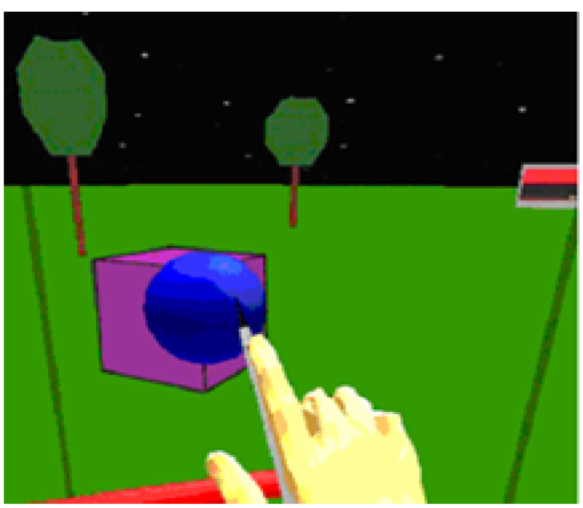

Figure 5: User placing sphere inside axis-aligned target cube. (Edges of cube have been enhanced for visibility)

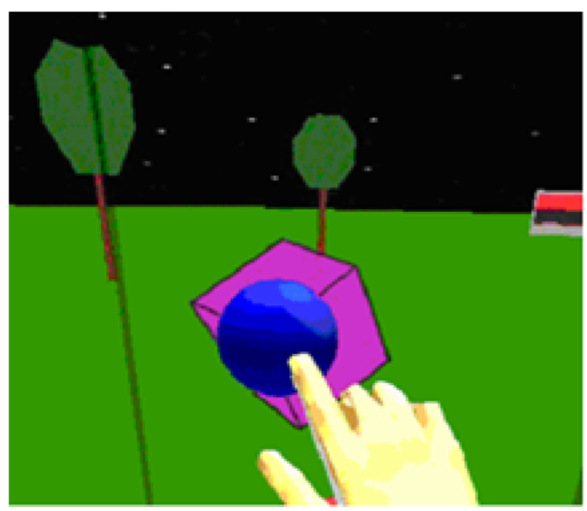

Figure 6: User placing sphere inside rotated target cube. (Edges of cube have been enhanced for visibility)

4.1 Effects of PRISM Interaction

The central question we wished to address was whether the PRISM techniques were more effective than Direct Manipulation for a precision placement task. To answer this question, we conducted a multi-factor ANOVA on the resulting experimental data with the number of completions (number of spheres placed inside the cube) in each trial as the dependent variable. The fixed factors were Interaction Type (Direct, Generic PRISM, and Custom PRISM), and Target Orientation (Axis-Aligned and Rotated). Participants were treated as a random factor. Table 3 summarizes the results of the ANOVA. The results of the experiment, (given in Figure 7) show that the mean performance of the PRISM techniques was higher than the mean performance of the Direct Manipulation technique. The ANOVA shows that this difference is highly significant (P value < 0.001). Tukey pair wise comparisons show no statistically significant difference between the mean performance of Generic and Custom PRISM.

Table 3: ANOVA table for user study

Source	SS	DoF	MS	F Ratio	P Value
A: Interaction Type	365.17	2	182.59	33.17	**0.00**
B: Orientation	50.70	1	50.70	3.57	0.08
C: Participant	745.67	17	43.86	4.64	**0.00**
AB:	66.13	2	33.07	3.50	**0.04**
AC:	187.17	34	5.51	0.58	0.94
BC:	241.83	17	14.21	1.50	0.15
Residual	321.54	34	9.46		

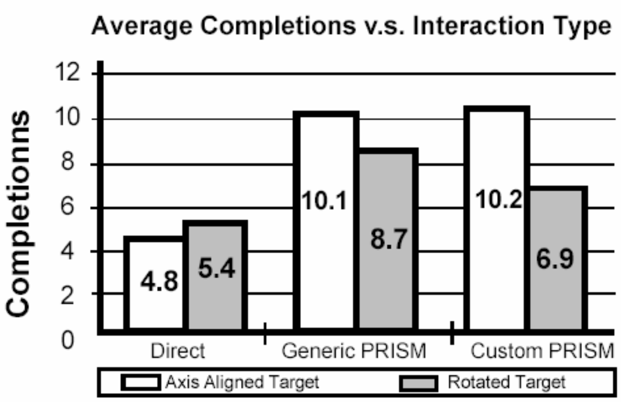

Figure 7: Mean completion rates categorized by interaction type and target orientation

4.2 Effects of Target Orientation

Another question we wished to address was whether the target orientation affected performance. The ANOVA shows that target orientation was not a statistically significant factor in performance (P-Value ~ 0.08). However, there is a significant interaction between interaction technique and orientation (P-Value ~ 0.04), with a performance loss when PRISM was used for rotated targets as opposed to axis aligned. These results suggest that although PRISM does help in precision for both target sets, performance is best when movement tends to be along a principal axis in the world. As described in Section 3.1, this is likely due to the fact that the scaling calculations are performed independently for each axis. Lower performance in rotated target trials might also have been caused by the added experience gained in with axis aligned targets in the practice trial, however the slight increase in performance for rotated targets in the direct (only) manipulation trials suggests otherwise. In order to realize better performance in arbitrarily oriented targets, one solution might be to alter the scaling algorithms to make use of the orientation of nearby objects instead of only using the principle axis of the world coordinate system. Despite the decrease in performance, both the Generic and the Custom PRISM techniques outperformed Direct Manipulation. Overall, our results indicate that providing scaled manipulation is more effective than direct manipulation for these types of precision interaction tasks.

4.3 Effects of Custom Scaling

During the pre-experiment training, participants were given five options for scaling values from which they were asked to choose. The "Normal" setting corresponded to the value used for the Generic PRISM trials. Table 4 summarizes which scaling values the participants chose to use. A question we wished to address was whether the Custom PRISM technique was more effective than the Generic PRISM technique. The ANOVA results presented in Table 2 and Tukey pair wise comparisons show that, although Interaction Technique was a significant factor, there is no statistically significant difference between the performance of the Custom and Generic PRISM techniques.

We were also interested in determining which participants were helped the most by the PRISM techniques. There is a statistically significant correlation between Direct Manipulation performance (which can be thought of as a measure of the user's raw skill) and the improvement realized in the PRISM trials. This correlation is not particularly strong however, with a correlation coefficient of -0.49 (P-Value = 0.04). This result suggests that users with lower performance with direct manipulation tended to improve more when using PRISM, however this could easily be attributed to the fact that the more skilled users simply performed well enough under direct that there was little room for improvement.

Table 4: Scaling Constants chosen during practice trials

Scaling Constant *		# of Participants
Highest	0.30 m/sec	1
High	0.20 m/sec	4
Normal	0.15 m/sec	3
Low	0.06 m/sec	9
Lowest	0.02 m/sec	1

* Scaling values determined through pilot studies.

Although, in this experiment, users chose their custom scaling settings in a pre-experiment training session, it would also be useful to be able to automatically determine the preferred settings. One way this could be done is to see if the user's skill level (shown by their performance using Direct Manipulation) predicts the scaling constant they chose. A statistical analysis shows that there is a significant correlation between performance for the Direct technique and the scaling constant chosen, but the correlation is not strong (correlation coefficient of -0.52, P-Value = 0.03). This does suggest that users who needed the most "help" tended to recognize this fact and chose accordingly. (Note that choosing higher scaling constants resulted in more scaling.) Similarly, the more skilled users chose lower scaling values. From this data, we suspect that a subject's "skill" *might* be a good parameter in determining which scaling values they might prefer.

Somewhat surprisingly, participants did not perform better when using custom scaling values as compared to the generic settings. There are a number of reasons this might be, with the most plausible being that users were simply able to adjust themselves to whichever scaling constant was used. Another possibility is that, although users in general seem to have chosen scaling values correctly, the values chosen were far from perfect. During the trials a number of participants indicated they could have chosen a better value. This is to be expected given that they were asked to pick values based on only around 10 minutes of experience. Perhaps with more training and a continuous set of scaling values the benefits of customization will come to fruition. Ideally, all users should reach the same performance levels when using customized settings, regardless of their dexterity; however we feel that our results are quite positive in that both generic and custom scaling improved user performance in all groups.

4.4 Exit Survey

As a further source of data, we administered an exit survey asking the participants how they liked the interaction techniques, the experimental process itself, and whether they thought rotated targets were more difficult than axis aligned. Users answered by giving a score ranging from 1 (Strongly Disagree) to 5 (Strongly Agree). Table 5 summarizes the result of the survey, where the Disagree column represents the sum of the number of Strongly Disagree and Disagree scores, and the Agree column represents the sum of the number of Strongly Agree and Agree scores. The remaining scores out of 18 responses were neutral.

When asked whether they were able to adequately learn the scaled interaction technique during the pre-experiment training session 14 participants agreed, 2 were neutral, and 2 disagreed. This is a very encouraging result since the training session was usually quite short (most were under 10 minutes), indicating that it is relatively easy to learn our technique. Nearly all participants agreed that generic PRISM was better than direct manipulation and 100% of the participants thought their custom settings were as good as or better than generic mode. This was a particularly surprising result given that performance did not increase when using custom settings over generic. At a minimum this indicates that users respond well to the ability to customize their interface (regardless of performance). On the other hand, it provides more evidence that although custom values can be important, users are able to adapt and use the generic setting just as well. Our final question asked the user if they felt rotated targets were more difficult than axis aligned, and the participants were almost evenly split. Regardless of which type of target the user preferred, their reason for preferring the configuration was most often that they could get a better view of three sides of the target cube when placing the sphere. Since target position in the virtual world was exactly the same for all participants (same x and z coordinates, height determined by user height) it is unclear why users disagreed on which target was easier to visualize. We plan to study this discrepancy, along with the significant interaction observed between scaling and rotated targets further in the future.

Table 5: Post experiment survey results

Survey Questions	Avg. Score	Disagree	Agree
Scaled interaction was easy to learn during training	3.78	2	14
Generic was better than Direct	3.67	3	13
Custom was better than Direct	4.33	0	16
Custom was better than Generic	3.78	0	11
Rotated was more difficult than axis-aligned	2.89	8	7

5 Future Work

Although our results suggest that PRISM is superior to more traditional direct manipulation, we were somewhat concerned with negative interaction between rotated target orientation and PRISM. Our hypothesis is that because the scaling function pushes interaction towards the principle world coordinate axes, the user is not aided as much when attempting to move in a more arbitrary direction (however scaled interaction still improved precision over direct mode). Currently, our scaling function makes no use of the position or orientation of nearby objects in the virtual world. This generality is something we are hesitant to change; however, in tasks where a very high precision is necessary our scaling function can be modified to scale against a nearby object's (a possible target) coordinate system.

Although users did not perform significantly better when using customized values as compared to the generic scaling value, we still believe that it is important that the scaling constant be customized based on the users needs. We are currently investigating ways to do this automatically by observing user behavior while interacting, which would be more accurate and flexible than relying on a training sessions and the user's opinion. As the user moves their hand, detecting a large offset indicates the scaling constant might be high and that the user would benefit from increased sensitivity. It is also possible to observe users continually making adjustments to an object's position, moving it back and forth. These adjustments likely indicate that the scaling constant is too low and that the user needs more precision. It is our hope that by changing the scaling constant in small increments over time, an optimum value for each user can be found.

Thus far, PRISM only focuses on allowing the user to perform precise and rapid translations to the controlled object's position. In order to achieve true precision in a 6DOF interface, we can apply the same filtering concept to the rotation and orientation of the controlled object. Accuracy when rotating an object can be enhanced through modifying the control/display ratio [24], however determining the user's intentions (precision or rapid) will involve a somewhat different approach. Just as scaling when translating produces an offset, scaling when rotating will produce an offset between the orientation of the hand and controlled object. We believe this rotational offset can be intuitively recovered in the same "ratcheting" manner as the translational offset, reconciling the offset when the user rotates their hand back in the opposite direction.

6 Conclusions

We have shown that PRISM allows users to achieve a higher degree of precision in a short amount of time when compared to the more traditional direct interaction approach. Given a limited amount of training and instruction, users were able to adequately learn this technique and realized a substantial and significant increase in performance in all conditions of our experiment.

Our new technique could be used as an addition to other existing, best-practice interaction techniques such as Go-Go, HOMER, and WIM and is not meant to replace such techniques. While these interaction techniques focus on novel methods of obtaining control and allowing for manipulation of distant objects, PRISM focuses on allowing users to precisely manipulate the object *after* they have gained control over it. Although we have primarily focused on manipulation of virtual objects in the world, our technique could also easily be modified to provide precision selection in the near field. By using the scaling function to control a 3D cursor a user could pick out very small objects in a potentially cluttered environment.

There are several other techniques which have been used to facilitate precise manipulation in virtual environments. These techniques provide the user with grid points, gravity, or constraints to help with precision. While these techniques produce high precision, they only work for specialized tasks, where the possible "target" positions are already known and pre-programmed into the environment. PRISM differs in that it is a general approach which can be used for any task to improve on direct manipulation. Using PRISM, the user is always in complete control of the position of the object being manipulating (in contrast to gravity and snapping techniques).

We believe that the natural control metaphor used by PRISM (hand speed) is what makes it so easy to learn. Users do not need to explicitly tell the interface if the are attempting to be precise and the interface does not inhibit the user from making rapid, imprecise translations by over-constraining manipulation. Switching between precise (scaling) and direct mode occurs seamlessly during natural interaction according to the current velocity of the user's hand.

PRISM is applicable across a wide variety of immersive virtual environments, relevant whenever 3D direct interaction is employed. This benefit could be realized regardless of the display type or tracking system used. 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

[1] Accot, J., Zhai, S., (1997). "Beyond Fitts' law: models for trajectory-based HCI tasks", *SIGCHI Conference on Human factors in computing systems*. pp. 295 - 302

[2] Bederson, B.B., Meyer, J., Good, L., (2000). "Jazz: An Extensible Zoomable User Interface Graphics Toolkit in Java", *In ACM UIST 2000*, pp. 171-180.

[3] Beir, Eric A., (1990). "Snap-Dragging in Three Dimensions", *ACM Symposium on Interactive 3D Graphics*, 24 (2), Mar. 1990, pp. 193-204.

[4] Bowman, D. and Hodges, L., (1997). "An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments". *Symposium on Interactive 3D Graphics*, 1997, pp. 35-38.

[5] Bowman, D., Johnson, D., and Hodges, L., (1999). "Testbed Evaluation of VE Interaction Techniques". *ACM Symposium on Virtual Reality Software and Technology*, 1999, pp. 26-33.

[6] Bowman, D. and Wingrave, C. (2001). "Design and Evaluation of Menu Systems for Immersive Virtual Environments". *IEEE Virtual Reality*, pp. 149-156.

[7] Bukowski, R. and Sequin, C. (1995). Object Associations: A Simple and Practical Approach to Virtual 3D Manipulation. *ACM Symposium on Interactive 3D Graphics*, 131-138.

[8] Fitts, P.M. (1954) "The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement". Journal of Experimental Psychology. 7(0) pp. 93-242.

[9] Foley, James D., Victor L. Wallace, and Peggy Chan (1984). "The Human Factors of Computer Graphics Interaction Techniques". *IEEE Computer Graphics & Applications*, Nov. 1984, pp. 13-48.

[10] Hinckley, K., Pausch, R., Goble, J., Kassell, N.F., (1994). "A Survey of Design Issues in Spatial Input", *Proceedings of ACM UIST*, Marina del Rey, CA, Nov. 1994, pp. 213-222.

[11] Hinckley, Ken, Randy Pausch, John C. Goble, Neal F. Kassell, (1993). "Passive Real-World Interface Props for Neurosurgical Visualization", *SIGCHI*, pages 452 - 458, October 1993.

[12] Igarashi, I., Hinckley, K. (2000). "Speed-dependent Automatic Zooming for Browsing Large Documents". *Proceedings of the ACM UIST*. pp. 129-148.

[13] Ishii, H. and Ullmer, B, (1997). "Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms", *Proc. of CHI'97*, ACM Press, pp. 234-241.

[14] Kessler, G. Drew, (1999). "A Framework for Interactors in Immersive Virtual Environments", *IEEE VR '99*, Houston, TX, Mar. 1999, pp 190-7.

[15] Kessler, G. Drew, Doug A. Bowman, and Larry F. Hodges, (2000). "The Simple Virtual Environment Library, an Extensible Framework for Building VE Applications", *Presence*, 9 (2), Apr. 2000, pp 187-208.

[16] Lindeman, R., Sibert, J, Hahn, J., (1999). "Towards Usable VR: An Empirical Study of User Interfaces for Immersive Virtual Environments," *Proc. Of the SIGCHI '99*, pp. 64-71

[17] Mapes, D. P. and Moshell, J. M. (1995). "A two-handed interface for object manipulation in virtual environments", *Presence*, 4(4), pp. 403 - 416.

[18] Mine, M., Brooks, F., Séquin, C., (1997). "Moving Objects in Space: Exploiting Proprioception in Virtual-Environment Interaction," *Proc. of SIGGRAPH '97*, pp. 19-26.

[19] J. Pierce, A. Forsberg, M. Conway, S. Hong, R. Zeleznik and M. Mine, (1997). "Image plane interaction techniques in 3D Immersive environments", *Symposium on Interactive 3D Graphics*, pp. 39-44.

[20] Pierce, J., Stearns, B., and Pausch, R.. (1999) Two Handed Manipulation of Voodoo Dolls in Virtual Environments. *Symposium on Interactive 3D Graphics*, 141-145.

[21] Pierce, J., and Pausch, R... Comparing Voodoo Dolls and HOMER: Exploring (2002) The Importance of Feedback in Virtual Environments. *CHI 2002*, 105- 112.

[22] Poupyrev, I., Billinghurst, M., Weghorst, S., Ichikawa, T., (1996). "The Go-Go Interaction Techniques: Non-Linear Mapping for Direct Manipulation in VR", *Proceedings of ACM UIST '96*, pp 79-80.

[23] Poupyrev, I., Weghorst, S., Billinghurst, M., and Ichikawa, T. (1998). "Egocentric Object Manipulation in Virtual Environments: Empirical Evaluation of Interaction Techniques", *Computer Graphics Forum*, vol. 17 no. 3, 1998, pp. 41-52.

[24] Poupyrev, I., Weghorst, S., Fels, S. (2000). "Non-Isomorphic 3D Rotational Techniques". *ACM Conference on Human Factors in Computing Systems*. pp. 540-547.

[25] Ruddle, Roy A., Justin C. D. Savage, and Dylan M. Jones (2003). "Levels of control during a collaborative carrying task," *Presence*, 12(2), Apr., pp. 140-155.

[26] Stoakley, R., Conway, M., Pausch, R. (1995). "Virtual reality on a WIM: interactive worlds in miniature", *CHI' 95*, 1995, pp. 265-272.

[27] Tan, D.S., Robertson, G.G., Czerwinski, M. (2001). "Exploring 3D Navigation: Combining Speed-coupled Flying with Orbiting". *ACM Conference on Human Factors in Computing Systems*. pp. 418-424.