Dynamically Scaled Interaction for Precise, Direct 3D Manipulation

Scott E. Frees and G. Drew Kessler
Department of Computer Science and Engineering, Lehigh University
{sef3, gdk2}@lehigh.edu

Abstract

We present a new interaction technique to make 3D direct manipulation more precise. The technique scales the motion of controlled objects based inversely on the speed of the user's motion.

1. Motivation

We are interested in providing an immersive virtual environment (IVE) interface that allows users to quickly create and manipulate 3D objects. A significant benefit of an IVE 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. Unfortunately, it is difficult and frustrating for users to directly specify a precise location or move in a uniform direction. This precision problem can be summed up by the design challenge adeptly described in [2], stating that 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." Human factors research shows that, in the absence of force feedback or props, it is generally difficult 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 [5]. 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.

2. Scaled Manipulation

The premise of our interaction technique is that users manipulate objects at different speeds depending on their intentions. When moving an object from one 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. Unfortunately, unsteady hands and the lack of force feedback make being precise nearly impossible. Our solution to this problem is to adjust the "control/ display" ratio [1] that determines the relationship between physical movements and movements of the 3D cursor, by making the cursor less sensitive to movement. This is done independently for each axis, which allows us to recognize when a user is trying to move in one of the principle directions and eliminate slow drift in the other directions.

In order to determine if the user is in scaled mode, we examine the current velocity of the user's hand (or stylus), if it is below a threshold, the cursor movement is scaled. Otherwise, the hand motions control the object movements directly. In our implementation of direct manipulation, we initiate an interaction 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. During scaled manipulation, we apply a simple function to the position of the stylus (equation 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 a scaling constant (SC). We then move the object by the resulting distance.

$$D_{object} = \min(V_{hand} / SC, 1) * D_{hand} \qquad (1)$$

Implicit in this method is the accumulation of an offset value representing the distance between the hand and the object being manipulated (the white line in Figure 1). 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. The first mechanism works when the user is in scaling mode, has accumulated an offset in a particular direction and then changes direction,

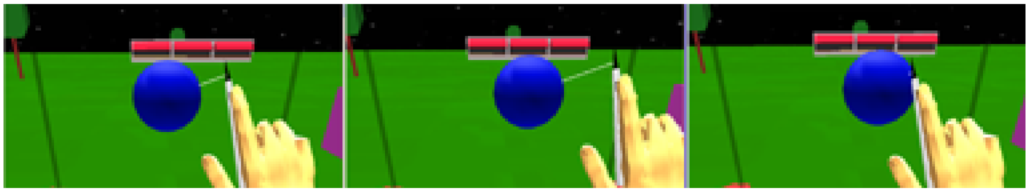

Figure 1. Interaction sequence of user moving object in scaled mode and then recovering offset

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. The second mechanism occurs when the user moves from the scaling state back to direct manipulation by moving their hand beyond the maximum velocity threshold. Instead of immediately eliminating the offset, it is reduced gradually so that the object appears to be catching up; allowing the user to realize they are switching to direct manipulation and to slow down if that isn't what they intended.

We have implemented this interaction technique as an interactor component using the SVIFT toolkit [3] for SVE [4]. Our preliminary results have been quite promising. Scaling hand movements has made interaction much less sensitive to the small sudden movements and slow drift indicative of unsteady hands. In addition to the increases in accuracy, the interactor also allows for quick and direct movements without any modal input from the user. In fact, one of the most effective aspects of this technique is that shifting between direct and scaled manipulation modes requires no explicit input from the user.

3. Future Work: Customizing Parameters

Although scaled manipulation 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 sensitivity and precision, which is shown in Figure 2.

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, users sometimes find the interaction technique frustrating and unhelpful; but with some manual adjustment, the technique is easily improved to the user's satisfaction. Examining interaction characteristics will allow us to customize the scaling constant automatically. 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 increasing and decreasing the scaling constant in small increments over time we can achieve an optimum value for each user.

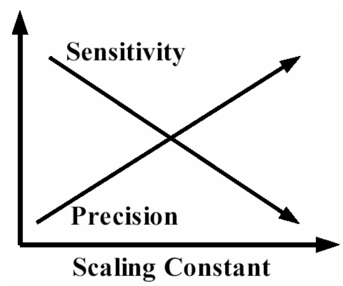

Figure 2. Effects of scaling constant

4. References

[1] 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.

[2] Hinckley, K., Pausch, R., Goble, J., Kassell, N.F., "A Survey of Design Issues in Spatial Input", *Proceedings of ACM UIST*, Marina del Rey, CA, Nov. 1994, pp213-222

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

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

[5] Kroemer, K.H.E., Kroemer, H.J., Kroemer-Elbert, K.E., *Engineering Physiology: Basis of Human Factors/ Ergonomics*, Van Nostrand Reinhold, 1997