# Designing Explicit Numeric Input Interfaces for Immersive Virtual Environments

*Jian Chen* *Doug A. Bowman* *Chadwick A. Wingrave* *John F. Lucas*
Department of Computer Science and Center for Human-Computer Interaction
Virginia Tech
Blacksburg, VA 24061, USA
Tel: 1-540-231-2641
E-mail: {jichen8, bowman, cwingrav, jolucas}@vt.edu

**ABSTRACT**
User interfaces involving explicit control of numeric values in immersive virtual environments have not been well studied. In the context of designing three-dimensional interaction techniques for the creation of multiple objects, called *cloning*, we have developed and tested a *dynamic slider interface (D-Slider)* and a *virtual numeric keypad (V-Key)*. Our cloning interface requires precise number input because it allows users to place objects at any location in the environment with a precision of 1/10 unit. The design of the interface focuses on feedback, constraints, and expressiveness. Comparative usability studies have shown that the newly designed user interfaces were easy to use, effective, and had a good quality of interaction. We describe a working prototype of our cloning interface, the iterative design process for D-Slider and V-Key, and lessons learned. Our interfaces can be re-used for any virtual environment interaction tasks requiring explicit numeric input.

**KEYWORDS:** 3D user interface design, numeric input, slider, keypad, virtual environment.

**INTRODUCTION**
Three-dimensional (3D) objects in many application domains, such as architecture and construction, can be extremely complex and can consist of a large number of repeating components. *Cloning* techniques, which generate multiple spatially distributed copies of an object, can be used to generate and position copies of objects more efficiently. Such techniques are important and useful in desktop three-dimensional modeling systems, and we have shown their usefulness in 3D user interfaces for immersive virtual environments (VEs) [8]. For example, using one of our interfaces, a user, starting with only two beams and one column, can build a two-story structure (Figure 1) in only 20 seconds – much faster than placing a single item at a time.

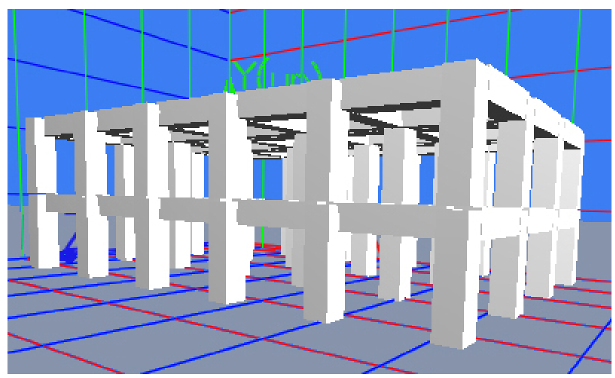



Figure 1: Example structure built in our environment from two beams and one column

Cloning is an example of a *domain-specific* 3D interaction task, since interaction techniques for this task need to take the characteristics of the domain (architecture and construction in this case) into account. Architects generally think of design with regard to space. Therefore, we can design the cloning techniques with regard to how the newly generated copies are distributed in space including: the *number of clones*, the *distance* between copies, and the *direction* (either positive or negative) along the x, y, and z axes. Since the number of clones is discrete, the distance is continuous, and the direction is binary, multi-dimensional numeric input strategies are required.

One general approach for designing interfaces for the cloning task is the “pen-and-tablet” metaphor, where widgets are displayed on a 2D physical surface held by the user [1, 4, 18, 26, 29, 30]. Users can interact with the widgets using a pen to perform spatial actions (such as pick, drop, or drag), which is a task familiar to most users.

The interfaces we designed were classified by their degrees-of-freedom (DOF). They were called *numerical spinners interface* (Spin), *orthogonal 1-DOF sliders interface* (1-DOF-O), *cavalier 1-DOF sliders interface*, *2-DOF widgets interface* (2-DOF), and *3-DOF widgets interface* (3-DOF) [8]. Dragging a widget causes the modification of one, two or three parameters immediately in the environment according to the DOF of the widget.

During the development of these interfaces, we used various three-dimensional design principles with regard to speed of input, perceived usefulness, ease of use, affordances, feedback, and attention. The results from the previous usability study confirmed that (1) slider widgets were better suited for discrete rather than for continuous numeric input; and that (2) we should constrain widget movement as much as possible. The results motivated our current research, which is intended to measure the size of the region that a use can place a widget in one DOF; and to design new interfaces with greater usability.

We begin this paper by describing related work and our original design, and then present our experiment to measure the minimum width of the region within which users can precisely place a slider widget (Resolution study). Next, we describe two new interfaces, *dynamic slider* (*D-Slider*) and *Virtual numeric keypad* (*V-Key*), which allow explicit and more efficient numeric input. We then describe a comparative study of the new design and the best of the previous interfaces. Finally, we discuss the lessons learned and future work for this ongoing project.

## RELATED WORK

Recent years have seen a growing number of immersive design tools for computer-aided design. Sachs' 3-Draw [26] and Deering's HoloSketch [12] used a pen to draw shapes directly on a tablet for interactive 3D shape design. Butterworth's 3DM [6] developed a toolbox, an icon-based user interface, to change the mode of operation for modeling. Liang's JDCAD [20] presented many novel ideas, such as the spotlight technique for conic selection and the ring menu for primitive creation, alignment, and reshaping using a 3D input device. Bowman's Virtual Habitat environment [5] allowed user editing of many types of objects and constrained the interaction using domain-specific information. Mine's ISAAC [23] used various menu display techniques and world-in-miniature to modify the space and create new elements. Most recently, Bowman's Virtual-SAP[4] presented menus for creating, transforming, and deleting architectural elements to provide input to earthquake simulations.

With these existing systems, however, it is still difficult to produce complex, but repetitive structures. A common characteristic of many of these systems was that they used direct manipulation to create or move vertices or objects. This had certain advantages, but also limited users' ability to model complex objects containing hundreds of elements or to provide precise positioning information. Our cloning interfaces, on the other hand, allow the user to generate complex structures quickly and with high levels of precision.

We used widgets in our design. Widgets are small objects with geometry and associated behavior. They can be displayed in the 3D scene or on a 2D tablet to be manipulated to change the scene or mode of the program by the user. Herndon and Meyer [16] designed widgets, including Probe, Rake, and Hedgehog, to control different parameters for scientific visualization. They also discussed the design issues regarding the widget's appearance (the geometry) and how the widget should behave. Slider widgets have been used in 3D user interfaces to define variables. Chen [10] grouped the sliders together on the interface or attached them to objects within the virtual world. Users manipulated the variables by directly controlling the widgets in 3D space.

To simplify the development of widgets and to allow the interface designer to concentrate on the application, a few toolkits have been developed. Conner [11] used ATNs (Augmented Transition Networks) in the design of 3D widgets. The models allowed for the rapid prototyping, modification, and easy integration of the 3D widgets into a system. Zeleznik and his co-authors [31] made early attempts to build a toolkit to construct new widgets from existing ones by building constraint relations among widgets.

However, there are still challenges in the design of widgets for 3D user interfaces. Existing examples of 2D interfaces embedded in 3D environments (such as pen and tablet interfaces) have not always been designed carefully for the needs and limitations of VE input and display devices. Design principles exist [28], but need to be studied within the context of applications [17]. In particular, it is not clear how to specify numbers precisely [24].

Furthermore, the precise positioning of objects is a difficult task although it is among the most fundamental interactions between humans and the VE [3, 14]. Graphical techniques, such as snap-dragging [2], have the problem that they create holes in the user's input space because it is not possible to specify points not on the snapping grid.

Our cloning interfaces were carefully designed explicit object positioning and numeric input techniques that could address these problems. We iteratively designed our interface to make it more applicable to real world applications.

## INITIAL DESIGN OF THE 1-DOF SLIDERS INTERFACE

### Overview

We used an interface based on the pen-and-tablet metaphor (Figure 2). The physical size (working area) of the tablet is 480mmx350mm. The bimanual interaction could increase performance compared to accomplishing the task unimanually [7].

Figure 2: Physical devices used in pen-and-tablet interaction metaphor

The task of cloning is to specify variables (number, distance, and direction). The different properties of the three types of variables (the number of clones as discrete, distance as continous, direction as binary) allowed us to investiage the slider design space throughly.

**Orthogonal 1-DOF slider interface (1-DOF-O)**

Our original design [8] of 1-DOF-O used six sliders, two on each axis, to control the number of clones and distance variables (Figure 3). The interface was called 1-DOF-O because all slider widgets were constrained to move along one dimension. Slider widgets were displayed in different colors: red-colored widgets controlled the number of clones and green-colored widgets controlled distance. The direction variables were automatically accounted for by the distance widgets since we allowed these widgets to indicate both positive and negative values. We constrained the movement of the "number of clones" widgets to be along the positive axis because these variables are inherently positive.

To avoid clutter on the tablet, only four widgets were displayed at a time. The two widgets on the x axis were always visible. Either the two widgets on the y axis or on the z axis were visible depending on the angle between the tablet and the horizontal axis. If the angle was within a range of $0^{o}$ to $45^{o}$ (Figure 3(a)), the z axis and the two attached widgets were displayed; and if the angle was within a range of $45^{o}$ to $90^{o}$, the y axis and its widgets were displayed. We chose to use angle to determine the mode because it is fast and easy for users to rotate the tablet while performing the task. They only need to make a small adjustment of their hand's or arm's position to switch axes. Such visibility constraints were made inactive when the user was interacting with a widget. This was because the user might get confused if the widget s/he was interacting with suddenly became invisible.

The interface also included fine-adjustment widgets. Clicking the arrow buttons shown at the end of each axis

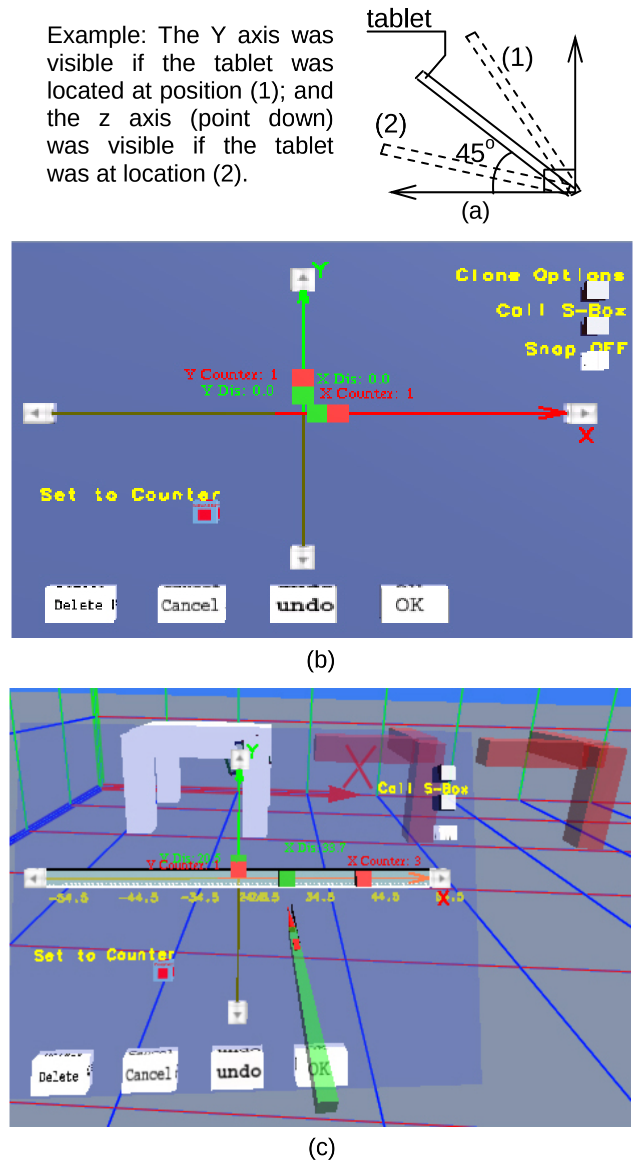



Figure 3: Orthogonal 1-DOF sliders interface (1-DOF-O)

would move either the number of clones or distance widgets along that dimension by predefined increments. The annotation and color themes used on the tablet were carefully designed for easy reading. The annotations were drawn on the screen (as a heads-up display). The current values of the variables were displayed next to the widgets in the same color as the widgets. The same color scheme was used to display the axes in the world. Also, the negative axis directions were drawn in a very different brown color for ease of interpretation.

We carefully designed the interface to reduce the requirements on the users' attention and to provide good feedback. A preview mode provides a "what you see is what you get" (WYSIWYG) interface. Users can immediately visualize the structure and decide on the next action. *Slider bars* were drawn to serve as self-explained constraints, to convey the behavior of the widgets and to inform the user how to act on them for spatial input.

### Design decision

In our previous usability study [8], the 1-DOF-O interface and the cavalier 1-DOF slider interface (1-DOF-C) were the two interfaces that were rated most highly for preference, perceived usefulness, and ease of use. The only difference between these two interfaces was that 1-DOF-C used a cavalier projection to draw the axes so that all axes and widgets were visible. No change of the arm angle was required to change the visibility of axes.

In our current work, we chose to refine the interface of 1-DOF-O rather than 1-DOF-C, because 1-DOF-O could take less space on the tablet than 1-DOF-C, if all widgets were moved only in the positive direction of an axis. The dotted line in Figure 4 illustrates the maximum tablet space taken by these two interfaces. 1-DOF-C presented a good separation of y and z axes, as reported by participants, but was spread out on the tablet because of the cavalier projected z axis.

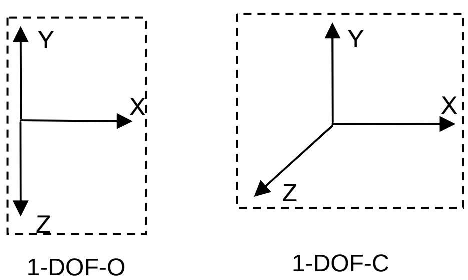



Figure 4: Tablet space required by different layout strategies

### Usability problems

The 1-DOF-O interface showed promises. However, it had two important and related usability problems:

(1) The sliders presented a fixed range of choices for the values of the variables, and therefore limited the tasks a user can do. For example, the number of clones was limited to a value between 1 and 5. Suppose an architect wants to build a 20-story building based on a one-story building. With this interface, s/he needs to do it in at least three steps: clone a five-story building from the one-story one, then select the newly created five-story building, and finally create the 20-story building.

(2) For the number of clones variable, the spacing between the "ticks" (locations at which the slider widget could be dropped) was too large. The physical distance from one tick to the next was 40mm. This large distance made slider positioning easier, but not so much useful in the working context. On the other hand, spacing between ticks was too small for the distance variables, so that precise positioning of the sliders was difficult. The physical distance from one tick to the next was 0.67mm. The greater required accuracy increased the level of frustration a user had while using the interface.

## FITTS' LAW EXPERIMENT

Before these problems can be addressed, we need to know the proper resolution of the tick, i.e., the width of the region that a user can accurately specify on a tablet-based slider. We, therefore, ran an experiment based on the Fitts' law concept. Fitts' law is an information-theoretical view of human motor behavior [13, 21, 22]. It expresses movement time (MT), the time to complete a movement task, in terms of the distance or amplitude of the move (A) and the width of the region within which the move must terminate (W). The purpose of our study was to quantitatively measure the width of the region that was suitable for the dragging sliders using the pen-and-tablet metaphor.

### Design

The two independent variables were the region width (W) and the direction of the sliders, either horizontal (for X) or vertical direction (for Y). The dependent variable was the task completion time. We fixed the amplitude of the movement (A) at 150mm. The testing conditions for W were W = 30, 15, 7.5, 3.75, 2.5, and 1.875mm. A balanced within-subjects design was used. We used six participants in our experiment. They were randomly assigned to different conditions using a Latin square pattern to a unique order.

### Equipment, environment, and software

The display device was a Virtual Research V8 head-mounted display (HMD) with binocular display (640 x 480 resolution, $60^o$ diagonal field of view). The user's head, pen, and tablet were all tracked by an InterSense IS-900 VET tracker. Participants used the pen to click (indicated by pressing a button), pick (indicated by pressing the same button), drag (indicated by holding the same button), and drop (release the button) widgets. Only the slider that should be set was displayed on the tablet. The scene did not include any object besides a grid to avoid any distraction. We implemented the system using SVE[19] and OpenGL.

### Tasks

Participants performed numeric input tasks by dragging the slider from the origin to the desired position. They performed six tasks given the six different Ws. The task description was displayed on the screen during the experiment for the user's reference.

### Participants

All participants were graduate students from the computer science department who volunteered for the experiment – two females and four males.

### Procedure

Participants filled out an informed consent and demographics form. Next, participants were told how to drag the sliders and were given practice tasks until they felt comfortable with the task. Errors were not counted and they were asked to continue their tasks until they positioned the slider at the right place. Finally, they performed the required tasks. All participants completed the tasks successfully.

### Results

We preformed a two-factor Analysis of Variance (ANOVA) on the width of the region (W) and the direction of the motion (X is horizontal and Y is vertical). The results were: (1) the widths of the regions were significant ($F=10.16$, $p<0.0001$); (2) the directions were not significant ($F=0.71$, $p=0.4$); (3) the two-way interaction was not significant. By performing the least significant difference (LSD) test on W, we found three groups, 30 and 15 (group 1); 7.5 and 3.75 (group 2); and 1.875 (group 3). We did not have enough power to decide which group W=2.5 belonged to, either group 2 or group 3.

### Design decision

We chose 15mm as the most appropriate region width for our future interface designs. This width provides 10 distinct positions on the sliders for all three directions because of the physical size limitation of the tablet (480mmx350mm) and the space taken by other widgets. We then made modifications to our initial design based on this result.

### DYNAMIC SLIDERS INTERFACE (D-Slider)

The Fitts' Law experiment told us how many distinct locations we should represent on our sliders, but this does not solve the problem of the fixed range of the slider. What if our hypothetical architect wants to create all 20 copies in a single action? We designed a "dynamic sliders" interface to address this problem. A dynamic slider (D-slider) allows the user to explicitly set the range of the slider based on the task requirement. In other words, the slider's "slot" can move in either direction to change the variable's range (Figure 5, 6).

We used the same number of sliders, color and visibility schemes as in the 1-DOF-O interface, but redesigned the location of the axes on the screen in order to accommodate 10 units in the slot. The X axis was moved to the border in order to accommodate more units on the Y or Z axis. It is displayed either near the bottom of the tablet when the Y axis is visible or near the top of the tablet when the Z axis is visible. Animation between the two locations is shown when switching axes. We do not display the negative axes, because users in our prior usability [8] preferred that distance and direction be controlled separately.

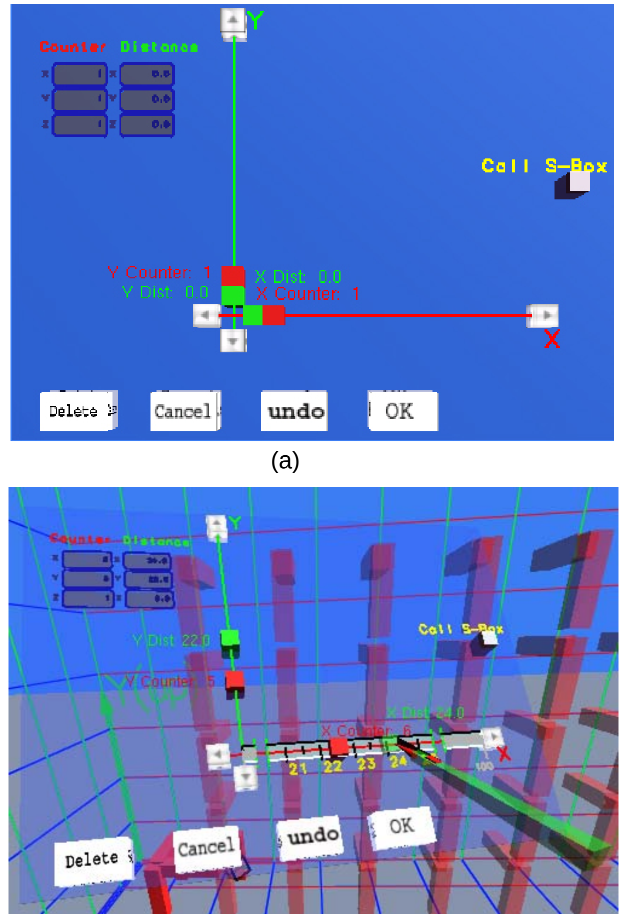



Figure 5: Dynamic sliders interface

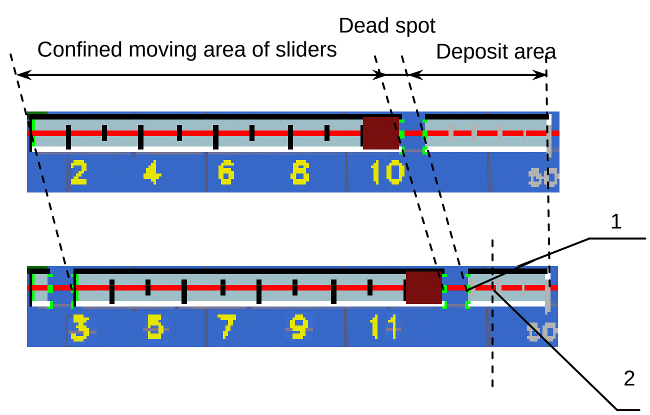



Figure 6: Behavior of the Dynamic Slider

Ticks are marked on the virtual tablet to indicate the current numbers for the slot (Figure 5(b), 6). The slider's numeric value does not change until the widget is moved at least one tick. Moving from one tick to the next changes the number of clones by one unit, and the distance by 0.5 units. The user can still achieve precise distances because of the integration of the fine control buttons, which change the distance by 0.1 units. We displayed labels between every two ticks to avoid cluttering the display. With 10 ticks, we can represent 10 units of the number of clones, and five units of the distance variable.

Figure 6 illustrates the behavior of the D-Slider if the projected position of the stylus falls on the right side of point 1. The slider slot (the confined moving area of sliders) moves to the right side and the ticks are updated. Moving the stylus back to the range of the confined moving area stops the movement of the slider slot. The movement would also stop if it reached the upper or lower limit, and there would be no deposit area, which gives the feedback that no more sliding can be done.

*Dead spot(s)* are drawn next to the slider slot. Moving the stylus within this area does not cause any change of the state. This design helps to avoid the accidental movement of the stylus beyond the allowed sliding area. A *deposit area* is drawn next to the dead spot (the point beyond the line at point 1 in Figure 6). Moving the stylus within the deposit area causes the movement of the slider slot in that direction and the increase of either the number of clones or the distance based on the widget selected. To allow both slow movements for fine adjustments and fast movements for quickly reaching a large number, we use two speeds. If the position of the stylus projected onto the axis falls between points 1 and 2, a slow speed is used, and if the position of the stylus is beyond 2, then a fast scrolling speed is used. The width between points 1 and 2 and the width of the dead spot are each 15mm. The same dead spot and the speed control fields exist on both sides of the slot.

**VIRTUAL NUMERIC KEYPAD (V-Key)**

We designed the virtual numeric keypad, or V-Key, as a complementary tool to the D-Slider. Just like desktop modeling tools (e.g., 3D Studio Max and AutoCAD), explicit numeric input is included as an alternative method to set up parameters. The keypad can be activated by clicking any of the six boxes (three for number of clones and three for distance variables) displayed on the upper left corner on the tablet.

A magnified view focus (Figure 7) at the cursor position of the stylus is drawn, similar to the fish-eye view focus in the Apple MacOS X Dock. Unlike the Dock, the magnified view focused on the cursor position *does* change the size of targets in motor space to make the selection easier, because previous experiments showed a target size problem with Apple's design [32]. Also, when one variable's box is magnified, the others may also be "virtually magnified." So if the stylus directly moves down from the current position displayed in Figure 7, the Z distance parameter box will be activated. We found this was the most intuitive way to enable easy selection of displayed boxes.

Clicking on any of the six box fields activates the V-Key interface (Figure 8). The action also causes the animated movement of the D-Slider interface to the right side of the tablet. The axes are still drawn to show the context of the


Figure 7: A magnified-view of the keypad

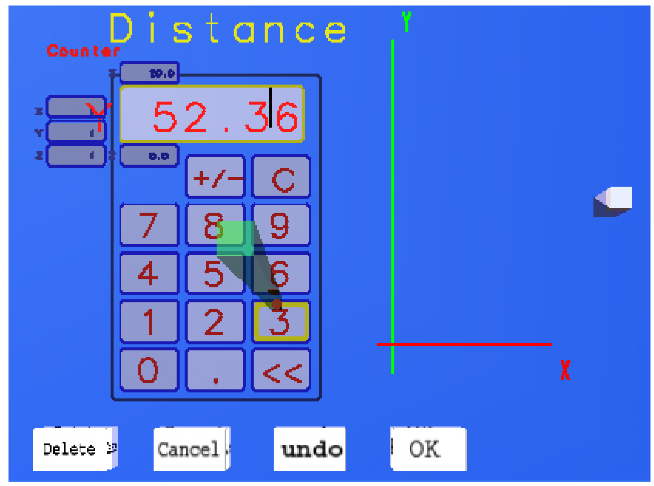



Figure 8: The Virtual numeric keypad interface with '3' highlighted

space. Clicking on the space formed by the X and Y/Z axes reactivates the D-Slider interface.

For entering numbers, familiarity with the number arrangement on the computer keyboard was used as a memory shortcut. *Stylus over*, similar to the mouse-over technique used in the desktop environment, was implemented: a yellow box is shown to highlight the button if the stylus is within the button area. Clicking on the key causes the insertion of the number at the insertion point. Figures 7 and 8 shows an example of changing the distance along the Y direction from 52.6 to 52.36 by clicking the button '3' when the insertion point is located between '.' and '6'. User can also move the insertion point by clicking on the number display field. This action and result are similar to desktop text editing tools like Microsoft Word™, but are novel in the context of an immersive VE.

**EXPERIMENT FOR THE COMPARISON OF 1-DOF-O, D-SLIDER, AND V-KEY**

We were interested in how much improvement we had made in the redesign process. Therefore, we performed a study to compare users' performance with the original interface and the two new techniques. The purpose of the evaluation was to (1) measure task performance and (2) measure the learnability of each user interface. We chose the 1-DOF-O interface as a baseline, because this interface was among the best of the original designs, before any improvements were made. We thus compared three interfaces: 1-DOF-O, D-Slider, and V-Key.

In addition to studying the users' performance, understanding the ease of initial adoption and the ease of learning presents an equally critical research challenge. In our experiment, we also tested whether the interfaces could help users in the very initial stage of learning the operation.

### Design

The independent variable was the interface used. To measure learnability, participants were not trained and performed the same task four times on each interface. The dependent variable was the task completion time. A balanced within-subjects design was used. The 12 participants were randomly assigned using a Latin square pattern to a unique order.

### Tasks

Participants performed numeric input tasks to assign specific values to the number of clones and distance variables. The task was "Generate a new structure that has three copies along the x axis, four copies along the y axis, and three copies along the z axis. The distance between the adjacent copies should be 50 units along the x axis, 40 units along the y axis, and 60 units along the z axis." The task was displayed on the screen during the experiment for the user's reference. Participants performed the same task four times with each user interface, which gave them 12 tasks in total.

Notice that the subtask of setting up the number of clones variables did not require the participants to change the slider's range while using the D-Slider interface, because the numbers required are part of the initial range. However, the subtask of setting up the distance variables did require the participants to change the slider's range since the numbers were not initially displayed on the slider bar.

### Equipment, environment, and software

The same equipment was used during this study as in the Fitts' law experiment, including the Virtual Research V8 HMD and the InterSense tracking system. The scene was rendered with SVE [19], and the interface on the tablet was implemented in OpenGL. The initial environment consisted of a structure with four beams and four columns structure

### Participants

All participants volunteered for the experiments. They were six undergraduate and three graduate students from the computer science department, and three undergraduate students from the architecture department. There were 11 males and 1 female between the ages of 20 and 31. None of them had used cloning user interfaces before.

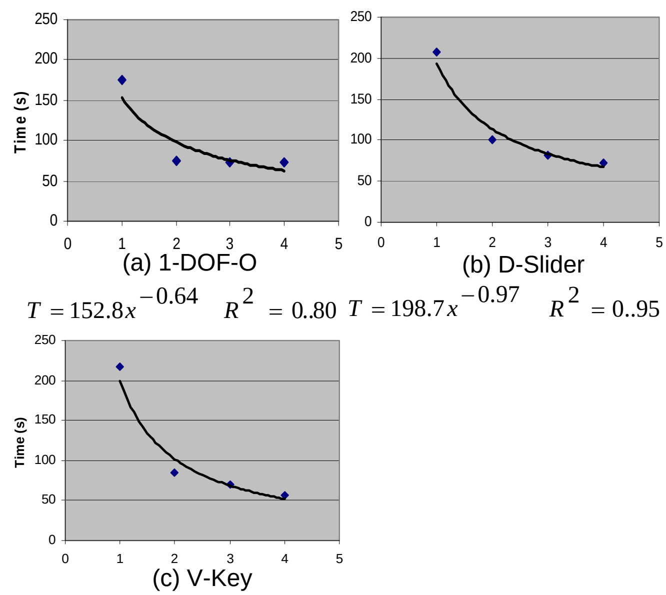
### Procedure

Participants were introduced to the idea of cloning, which was to set up six parameters (three for the number of clone and three for distances). They were given a task and asked to perform it with no prior training using the interface presented. After finishing all tasks, participants filled out a questionnaire, which elicited subjective responses on ease of use, preference, visual attention, satisfaction, and feedback.

## RESULTS AND ANALYSIS

### Learnability

The task completion time fit well with the learning curve [25] modeled by *the power law of practice* shown in Figure 9, where the vertical axis shows the task completion time (T), and the horizontal axis shows the practice time (x). The corresponding equations are shown below each curve. The results show that: (1) Participants had the longest start time using the V-Key and the shortest using the 1-DOF-O interface; (2) V-Key had the sharpest learning curve of the three (participants improved their performance very quickly).




$T = 152.8x^{-0.64}$ $R^2 = 0..80$ $T = 198.7x^{-0.97}$ $R^2 = 0..95$

$T = 193.4x^{-0.76}$ $R^2 = 0..96$

Figure 9: Learning curve

### Participants' performance

We performed a single-factor Analysis of Variance (ANOVA) on task completion time using the data from the last three trials. We did not use the first trial because the task completion time was significantly longer than the other three trials ($p<0.0001$). This was mainly attributed to the thinking time. There was no significant difference among the three user interfaces ($F = 1.64$, $p = 0.2$). The means of the task completion time were 74s, 83.9s, and 70.3s for the 1-DOF-O, D-Slider, and V-Key interfaces respectively. The D-Slider interface had the worst performance (although not significant), because changing the range of the sliders required some manual dexterity. We can see from the learning curve that greater skill was obtained with practice. In fact, the D-slider had a shorter task completion time than

1-DOF-O for the fourth trial (72.7s for D-Slider compared to 73.2s for 1-DOF-O).

We also compared the slider dragging times (pure working time, not including times when no interaction) of the 1-DOF-O and D-Slider interfaces. The results showed that there was no significant difference between the two ($mean_{1\text{-}DOF\text{-}O}$ = 41s, $mean_{D\text{-}Slider}$ = 44.2s, $F = 0.63$, $p = 0.4$). This result suggests that we did not lose performance while increasing the power of the interface.

The comparison of dragging time between the number of clones and distance widgets for the D-Slider interface showed that participants spent significantly more time in setting up distance parameters ($mean_{Number}$ = 7.4s, $mean_{Distance}$ = 36.8s, $F = 106.4$, $p<0.0001$).

Three out of 12 Participants dragged the number of clones widget and misunderstood it as the distance widget during the first trial. They immediately realized the mistake while "previewing" the results and corrected it. None of the participants had such misunderstanding after the second trial.

### Participants' ratings and comments

Participants rated the interfaces for user satisfaction, preference, visual attention, and ease of use on a scale of 1 to 7, where 7 was the best, 1 was the worst and 4 was neutral. Figure 10 shows the results. There was no significant difference on any of these ratings among the three interfaces. The results matched participants' comments that the V-Key was the best designed. All the questions we asked were with regard to the usability of the interface itself, but not about the architecture/construction domain. All three architecture domain participants mentioned that 1-DOF-O was good for parameter setup, but not that useful in the context of their work because of the limited range. Two participants commented that they disliked V-Key although it was easy to use, because it is more difficult to make a small change to a parameter with this interface, and because the preview of the cloning does not change smoothly when numbers are entered in this way.

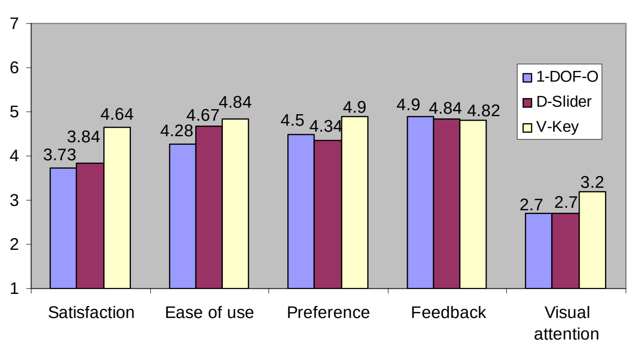



Figure 10: Subjective ratings

### Observations

The main observations we had were: (1) every participant used the fine control button while using 1-DOF-O to setup the three distance variables because it was frustrating to position the sliders; (2) participants were confused about the layout of the six small buttons after the keypad was activated. However, this was not the case when the keypad was not visible and the magnified view was shown. Four out of 12 participants deactivated the keypad (by clicking any position on the right side of the tablet) and then activated it again to setup other parameters; (3) Five participants wanted to confirm their input (by clicking ok) after they input a number, and 10 participants cleared the input field in V-Key and then started from blank although they knew they could use the backspace. This result may suggest exchanging the position of the two buttons "+/-" (positive/ negative input) and "<<" (backspace).

## DESIGN PRINCIPLES FOR WIDGET-BASED 3D INTERFACES

During our iterative design and evaluation process, we gathered participants' feedback and performance data to gain insights into user requirements for the task of cloning. This process has led to several proposed design principles.

### Prevent users' errors

The likelihood of execution errors increases with poor interface design. Errors users make increases the level of frustration and prevents their further use of the interface. We reduced users' error using two techniques: the preview mode and the dead spots in the D-slider.

### User fewer DOF for widgets if possible

The users' frustration increases with an increasing number of DOF. This is unlike the 2D interface where a mouse can easily specify 2-DOF. In a VE, the tracking delay and the error caused a high DOF input strategy to be impractical for precise input. We adopted the 1-DOF design in our study.

### Reduce divided attention in the interface

We want to design the interface to have the users' attention less focused on the interface and more focused on the tasks they are doing. Too much divided attention will reduce task performance.

The pen-and-tablet interface produces a dichotomy between the user interface and the 3D space in which the user is working. A user's focus of attention must constantly change from some point in the 3D space to the 2D widget interface, and then refocus on the 3D space again.

To address this, one main difficulty was keeping the slider selected with the pen. We designed the *take-off* mode: once a widget is grabbed and dragged, the position of the widget is decided by the projected position of the stylus on the axis. It can be moved with the stylus even when the stylus is not touching it, until the button is released. Therefore, users do not need to look at the interface while dragging.

Also, the preview mode helps to reduce the attention on the interface.

### Design spatial interface for spatial tasks

This is an interface layout issue. The layout should match the task models for the tasks that a user is going to perform. This avoids cognitive load. For example, we found that some participants did not like the V-Key interface because they did not feel that they were doing tasks in 3D. They had to map the parameter back to the 3D space, which could potentially increase the thinking time. Presenting the widgets interface in a 3D continuous space using sliders was a better fit.

### Provide appropriate annotation and visual feedback

Another source of feedback was semantic in nature [15]. Text was drawn on the tablet to indicate the current parameters in the system. During our experiments, we found that an interface with appropriate annotation would reduce errors. Participants preferred the interfaces with good visual feedback [8].

The annotations in our interfaces' design play two roles: (1) they help users choose the correct widget; and (2) they provide feedback for the values of variables. Furthermore, they are color-coded to augment the correspondence between the widgets and the text displayed.

### Consider users' difference

Users have different spatial abilities. They have different preferences and performance. We need to have interfaces designed for different user groups.

Our user study found that some people preferred the spatial input with sliders because of the feeling of 3D; while some participants preferred V-Key because of its ease of use. We have provided alternative choices by integrating different input methods into one application.

### Increase the quality of display

When readability is a dominant requirement for the system, heads-up displays (HUDs) can improve the readability of the interface significantly based on our previous experience on designing information-rich virtual environments [9]. We have displayed the information that needs to be frequently accessed on a HUD (e.g., the current parameters of the number of clones and distances), and others on the tablet directly for reducing the clutter on the screen.

### Provide the context of change plus an overview

There are at least three benefits when the interface provides a good overview of the current state of the system (in our case, the six variables): (1) the overview aids users in keeping track of the current state in the information space which has similar benefits in two-dimensional information visualization; (2) the overview gives users task-relevant information, for example, the number of clones and the distance parameters the user entered; and (3) the overview gives users a feeling of control [27]. We drew the slider interface on the right side of the screen while the V-Key interface is activated, this could remind the user that they can switch back to the D-Slider user interface.

## CONCLUSION AND FUTURE WORK

We iteratively designed user interfaces for cloning. We designed the D-Slider and V-Key interfaces which would increase the functionality available to the user to set up any range of numbers without losing performance. Our research on the low-level interface issues will lead to detailed guidelines for widget and overall interface design in 3D space.

It is easy to extend our interface to other applications that require explicit numeric control tasks, e.g., modeling and moving objects in 3D. Our results from the Fitts' law experiment can be used by researchers to estimate parameters without further empirical validation to make predictions of target size.

One piece of future work will be to integrate the slider interface and keypad into the SVE library to make them available to more developers, since the development of the slider-widget and keypad design was very time-consuming. We also plan to investigate direct manipulation techniques to clone objects in the 3D space.

## ACKNOWLEDGMENTS

The work presented in this paper was sponsored by the National Science Foundation (NSF-IIS-0237412). The authors are grateful to the participants for volunteering for this study.

## REFERENCES

1. Augus, I. and Sowizra, H. Embedding the 2D Interaction Metaphor in a Real 3D Virtual Environment, *Proc. SPIE (Stereoscopic Displays and Virtual Reality Systems)*, (1995), 282-293.
2. Bier, E.A. Snap-Dragging in Three Dimensions, *Proc. Symposium on Interactive 3D Graphics*, (1990), 131-138.
3. Bowman, D.A. and Hodges, L.F. An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments, *Proc. Symposium on Interactive 3D Graphics*, (1997), 35-38.
4. Bowman, D.A., Setareh, M., Pinho, M.S., Ali, N., Kalita, A., Lee, Y., Lucas, J.F., Gracey, M., Kothapalli, M., Zhu, Q., Datey, A., and Tumati, P. Virtual-SAP: an Immersive Tool for Visualizing the Response of Building Structure to Environmental Conditions, *Proc. IEEE Virtual Reality*, (2003), 243-250.
5. Bowman, D.A., Wineman, J., Hodges, L.F., and Allison, D. Designing Animal Habitats within an Immersive VE, *IEEE Computer Graphics and Applications*, September/October (1998), 9-13.
6. Butterworth, J., Davidson, A., Hench, S., and Olano,

M.T. 3DM: a Three Dimensional Modeler Using a Head-mounted Display, *Proc. Symposium on Interactive 3D Graphics*, (1992), 135 - 138.
7. Buxton, W. and Myers, B.A. A Study on Two-handed Input, *Proc. ACM SIGGRAPH*, (1986), 365-372.
8. Chen, J., Bowman, D.A., Lucas, J.F., and Wingrave, C.A. Interfaces for Cloning in Immersive Virtual Environments, *To appear in the Proc. Eurographics Symposium on Virtual Environments*, (June 2004). Also available as Virginia Tech, CS, Tech Report TR04-08.
9. Chen, J., Pyla, P.S., and Bowman, D.A. Testbed Evaluation of Navigation and Text Display Techniques in an Information-Rich Virtual Environment, *Proc. IEEE Virtual Reality*, (2004), 181-188.
10. Chen, M. A Study in Interactive 3D Rotation Using 2-D Control Devices, *Computer Graphics (ACM SIGGRAPH)*, 22,4 (1988).
11. Conner, D.B., Snibbe, S.S., Herndon, K., Robbins, D.C., Zeleznik, R., and van Dam, A. Three-Dimensional Widgets, *Computer Graphics*, 25,2 (1992), 183-188.
12. Deering, M.F. HoloSketch: a Virtual Reality Sketching/Animation Tool, *ACM Transactions on Computer-Human Interaction (TOCHI)*, 2,3 (1995), 220-238.
13. Fitts, P.M. The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement, *Journal of Experimental Psychology*, 47 (1954), 381-391.
14. Foley, J.D., Wallace, V.L., and Chan, P. The Human Factors of Computer Graphics Interaction Techniques, *IEEE Computer Graphics and Applications*, 4 (1984), 13-48.
15. Green, M. and Jacob, R. Software Architectures and Metaphors for Non-WIMP User Interface, *Computer Graphics (ACM SIGGRAPH)*, 25,3 (1990), 229-235.
16. Hernandon, K.P. and Meyer, T. 3D Widgets for Exploratory Scientific Visualization, *Proc. ACM UIST*, (1994), 69-70.
17. Johnson, J. Selectors: Going Beyond User-Interface Widgets, *Proc. ACM CHI*, (1992), 273-279.
18. Keckeisen, M., Stoev, S.L., Feurer, M., and StraBer, W. Interactive Cloth Simulation in Virtual Environments, *Proc. IEEE Virtual Reality*, (2003), 71-78.
19. Kessler, G., Bowman, D.A., and Hodges, L.F. The Simple Virtual Environment Library: an Extensible Framework for Building VE Applications, *Presence: Teleoperators and Virtual Environments*, 9, 2 (2000), 187-208.
20. Liang, J. and Green, M. Geometric Modeling Using Six Degrees of Freedom Input Devices, *Proc. 3rd International Conference on CAD and Computer Graphics*, (1993), 217-222.
21. MacKenzie, I.S., Sellen, A., and Buxton, W. A Comparison of Input Devices in Elemental Pointing and Dragging Tasks, *Proc. ACM CHI*, (1991), 161-166.
22. MacKenzie, I.S. and Ware, C. Lag as a Determinant of Human Performance in Interactive Systems, *Proc. Interchi*, (1993), 488-493.
23. Mine, M.R. A Meta-CAD System for Virtual Environment, *Computer-Aided Design*, 29,8 (1997), 547-553.
24. Muller, A., Conrad, S., and Kruijff, E. Multifaceted Interaction with a Virtual Engineering Environment using a Scenegraph-oriented Approach, *Proc. International Conferences in Central Europe on Computer Graphics, Visualization and Computer Vision*, (2003).
25. Ritter, F.E. and Schooler, L.J. The Learning Curve, in *International Encyclopedia of the Social and Behavioral Science*: Amsterdam: Pergamon, (2003), 8602-8605.
26. Sachs, E., Roberts, A., and Stoops, D. 3-Draw: a Tool for Designing 3D Shapes, *IEEE Computer Graphics and Applications*, 11,6 (1991), 18-26.
27. Shneiderman, B. *Designing the User Interface*, 3rd edition, (1997).
28. Snibbe, S.S., Herndon, K., Robbins, D.C., Conner, D.B., and van Dam, A. Using Deformations to Explore 3D Widget Design, *Computer Graphics (SIGGRAPH'92)*, 26,2 (1992), 351-352.
29. Stoakley, R., Conway, M.J., and Pausch, R. Virtual Reality on a WIM: interactive worlds in miniature, *Proc. ACM SIGCHI*, (1995), 265-272.
30. Szalavari, Z. and Gervautz, M. The Personal Interaction Panel - a Two Handed Interface for Augmented Reality, *Computer Graphics Forum (Eurographics)*, 16,3 (1997), 335-346.
31. Zeleznik, R.C., Herndon, K.P., Robbins, D.C., Huang, N., Meyer, T., Parker, N., and Hughes, J.F. An Interactive 3D Toolkit for Constructing 3D Widgets, *Computer Graphics (SIGGRAPH'93)*, 27,4 (1993), 81-84.
32. Zhai, S., Conversy, S., Beaudouin-Lafon, M., and Guiard, Y. Human On-line Response to Target Expansion, *Proc. ACM CHI*, (2003), 177-184.