Evaluation of the Effectiveness of Cloning Techniques for Architectural Virtual Environments

Author

Organization or School

ABSTRACT

We made the first attempt towards building effective domain-specific interaction techniques for a cloning task. Five interaction techniques were designed and evaluated considering different aspects of domain requirements and human limitations. We demonstrated their effectiveness of designed techniques in two usability studies. The results suggested that no single technique is best for all task conditions. Techniques designed for cloning improved the domain task performance profoundly. The work suggests a further direction: passing domain knowledge to the design process to increase the usefulness of VEs.

CR Categories: H.5.m Information Interfaces and Presentation: Miscellaneous; I.3.6 Computer Graphics: Methodology and Techniques – Interaction Techniques; I.3.7 Computer Graphics: Three-Dimensional Graphics and Realism – Virtual Reality.

Keywords: Domain-specific interaction, virtual environments, structural engineering.

1 INTRODUCTION

One of the most significant obstacles in the use of virtual environments (VEs) in real-world applications is providing effective interaction techniques. Various techniques have been designed and valuable understanding has been gained (see [7]) but applications applying them remain sparse. Conversely, some useful techniques have been designed for specific applications, but are impractical for use by other applications [14]. A recent survey [5] on the 3D UI (three-dimensional user interaction) mailing list showed the lack of new categories of production applications of VEs, compared to a previous study in 1999 [10]. Why?

One possibility is that existing tasks and techniques are designed at either a very high or a very low level. For example, the majority of the techniques that have been designed were to be *general* enough so as to be applied in many different situations. This generality, however, hampered the design efforts of developers who have to customize these techniques to real-world applications. A lot of tweaking needs to be done to make interaction techniques which meet domain requirements [26].

We therefore chose a middle ground – design at the domain level. The idea is to focus on passing domain knowledge to the design process to discover *domain-specific tasks* and to design *domain-specific techniques*. Domain-specificity emerged from our long-term study of building VE applications for the AEC (architecture, engineering and construction) domain. These applications included walkthrough [9], conceptual design [4], constrained object layout techniques [22], and structural design [8]. One of the initial results of this work was the recognition of two important types of tasks for AEC applications: a cloning task [12] and a multiple object selection task[20].

Cloning refers to the creation of multiple distributed copies of a selected object or set of selected objects – a more powerful version

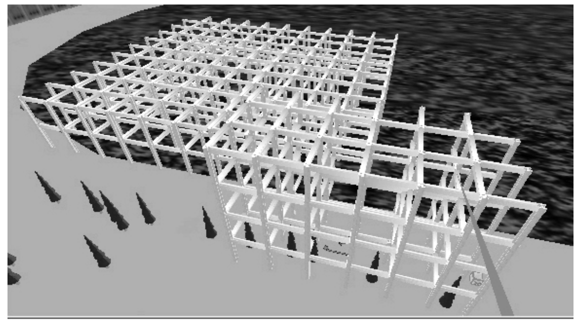


Figure 1. An example structure generated using our user interfaces

of "copy and paste." Previous work on cloning techniques relied on pen-and-tablet user interfaces with slider widgets [12]. Similar to their two-dimensional (2D) desktop counterpart, they were intuitive and easy to learn. They also offered haptic feedback. However, the pen-and-tablet interface acted as an intermediary between the user and the objects in the world. This reduced the "directness" [15] of the interface. Additionally, the slider interfaces, constrained by the physical size of the tablet, was less usable for defining large numbers.

It is our belief that the cloning interfaces could be better designed for the AEC domain. This led us to design alternative techniques: dynamic slider, keypad, copy-by-example, Pointer Orientation-based Resize Technique (PORT) [20] and space-metaphor based cloning techniques. These interfaces better utilized domain characteristics and overcame human limitations. Also, unlike previous work that focuses on the usability evaluation of the techniques itself, the goal of this work is to provide evidence that the techniques are useful and beneficial to the structural engineering applications.

The organization of this paper is as follows. Background and related work is presented in section 2. In section 3, we provide a list of domain characteristics we have considered so far for cloning techniques. Techniques and their design rationale are discussed in section 4. Two experiments evaluating the effectiveness of cloning techniques are in section 5 and their results and discussion are described in section 6. Finally, we concluded in section 7 with impact and future work.

2 BACKGROUND AND RELATED WORK

The study of cloning techniques was motivated by the Virtual-SAP application [8], a tool for structural engineers and students of structural engineering to observe the effects of earthquakes while immersed. In this environment, users can add and move architectural elements, such as beams, columns, slabs and walls. A pen-and-tablet based menu enables users to load a new structure, new elements or change an object's material. The application enabled demonstrations of earthquake effects to be shown in the classroom on quickly mocked-up structures. Even with this functionality, it was still tedious to build a complex structure with a large number of elements (e.g., a structure in Figure 1.) After observing structures in the architecture domain, we recognized that

Street Address and Electronic Mail Address

most have a repetitive pattern – cloning addressed this domain constraint.

VEs offer several advantages over traditional desktop modeling tools because VEs provides rich interaction and a direct view of the environment, and sometimes help users discover mistakes otherwise impossible to find. Modeling within VEs, however, is still a challenge. Most object manipulation is constraints-based and the tool may involve a complex constraints solver [17]. Others have used direct manipulation [15]. For example snap-dragging combined snap-to-grid and alignment to confine object movement to simplify object manipulation [3]. Various modeling techniques have been used in 3DM [11], JDCAD[19] and LEGO [23]. However, none of them are suitable for cloning operation. Desktop tools, like AutoCAD and SketchUp, have the function of cloning, but they are not practical to be used in VEs because of workflow issues. Users have to go back to the desktop to make a modification, and then import it to the VE for verification. Our goal is to conduct immersive design.

3 DOMAIN CONSIDERATIONS

To better design cloning techniques that are suitable to the AEC domain, we organized the domain characteristics into four categories: domain objects, environment, domain and users (Table 1). This domain knowledge is used as an instrument in which defining the characteristics that could have been considered for designing interaction techniques in the AEC domain.

Table 1. Domain characteristics classified for cloning

Characteristics of domain objects	
Geometrical	Standard size and shape
Physical	Objects have mass, material, texture
Mechanical	Objects are rigid
Characteristics of the environment	
Spatial	Repetitive patterns; Large in size; Specific mode definition
Dynamic	Movement followed objects' properties
Characteristics of the domain	
Mental model	Workflow / domain study
Characteristics of users	
Attitude	Cognitive and non-cognitive abilities that affect their choice of interaction techniques
Novice/expert	Behavior differences

For example, while describing cloning tasks, we have considered geometrical (beams, columns, slabs), mechanical, spatial (numbers of clones, spacing, position, direction, shape) and visual characteristics (such as color, material, texture) [12]. In this way, we won't miss out other factors that could potentially improve the quality of the user interface to make VEs more effective Also unexplored characteristics can be identified and added later to expand the design space.

4 CLONING USER INTERFACES

Cloning is a two-step operation: users select objects of interest then generate new copies. The numbers of copies, the distances between copies and the direction were considered in the design of cloning user interfaces. Cloning interfaces are designed for an immersive VE (Figure 2). The display device is a Virtual Research V8 head-mounted display (HMD) (binocular, 640x480 resolution and 60^{o} diagonal field of view). The user's head and hands are tracked by an InterSense IS-900 VET tracking system. A pen-and-tablet interface [2] includes a wand and a piece of tracked plexiglass. Several user interfaces designed do not use tablet unless mentioned. The travel technique is pointing based [7], i.e., users point in the desired direction with the wand and hold a button to fly.

Figure 2. Physical devices

4.1 Multiple object selection technique

Selection is the first step in the operation of cloning. In our interfaces, users select multiple objects with a selection box. It can be grabbed with ray-casting [21], manipulated with HOMER [6] and resized with PORT [20] (Figure 3). Any object(s) intersected with this box will be selected and highlighted. Selection and deselection activate sound cues to provide extra feedback.

PORT uses the relative direction of the hand-held pointer to the object to determine the axis of resizing. Pushing joystick forward/backward will resize the side that is perpendicular to the direction of the ray. An arrow and an X marker indicate the direction of resizing. The box is always aligned to the axis to reduce the degrees-of-freedom a user has to control since most structural elements are aligned to axes.

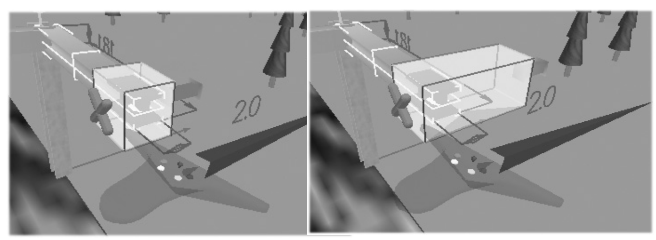


Figure 3. Pointer Orientation Based Resize Techniques

4.2 PORT cloning technique (PORT-C)

PORT also affords setting up the number of copies and distances, where users can simply point to the direction of interest and push forward/backward of the joystick to specify numbers. This reuse of the PORT metaphor for multiple stages of the cloning process produces consistency and therefore more intuitive operation of the interface.

The PORT cloning technique was designed to support this process. Four buttons on the wand integrated all operations. The interface displays currently available commands in a ring menu attached to the wand (Figure 4(a)). No label is presented on the upper left button to avoid extra visual search because that button was always used for travel.

The functions of each button changed according to users' action. The state transition of buttons is shown in Figure 4(b). The initial states of the four buttons are "Travel", "Select", "Delete" and "GetBox." The "GetBox" button is a homing command for the selection box, i.e., the box is moved to a position in front of the user and its size is reset. It is useful when the user does not know where the selection box is or just wants to recover the initial state.

Users select object(s) by grabbing the box then dropping it onto object(s). This action causes the lower right button to

(a)

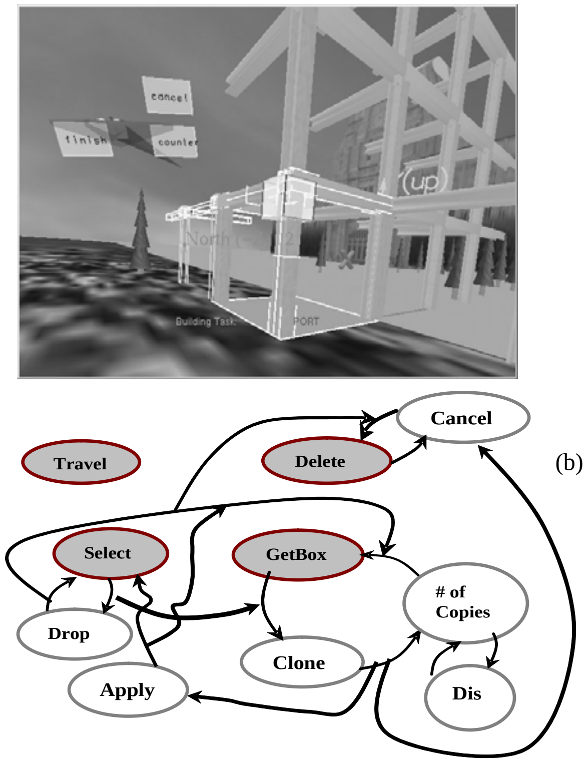


Figure 4. PORT cloning interface and state transitions

change to “Clone”. Upon selection, a bounding box appeared to indicate the bounding volume of the selected objects. Clicking the “Clone” button activates cloning. The lower right button then changes to “# of copies”. Simultaneously, the feedback objects attached to the selection box (the arrow and the X) moved to faces of the bounding box of the selected objects. Pointing to a direction and pushing forward/backward on the joystick will cause the increase or decrease of the number of copies. The user can click the lower right button to toggle between “# of copies” and “Dis” (to set up distances between copies) till s/he finishes or cancels the current cloning operation.

We have considered the number of copies along six directions as “separable” [16] parameters, in other words, they are controlled separately, while distances along negative and positive direction along the same axis as “integral” parameters, i.e., they are controlled together. Users found that it was intuitive [12].

During the editing process, ghost copies of created objects are drawn to show what the user expects to get. They are not added to the environment completely until the “Apply” command is issued. Parameter of the current state is displayed on the screen for task-related cue. For example, “East(X) = 5” means that the user is pointing to the east direction while setting the numbers of copies to 5.

4.3 Space metaphor interface (Space-metaphor)

The PORT cloning interface only enables four actions at a time. It may be too small a vocabulary compared to real-world applications that demand more commands. Therefore, additional user interface elements are need for more options. We designed the *space metaphor user interface* (Figure 5): a pen-and-tablet metaphor user interface. Four buttons, drawn on a small lightweight physical tablet (Figure 2, size of 25cmx25cm), defined four logical modes: operation, space, preview and parameter. Clicking the button will toggle the options as listed. The labels,

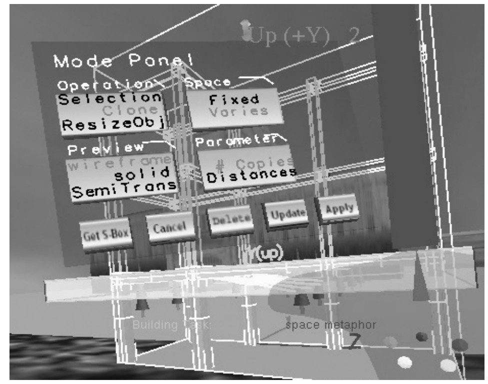


Figure 5. Space-metaphor interface

representing the current state of the toggle, are displayed in a distinct red color, while the other labels are in black.

This interface designed was chosen because we often observed that architects think about space and scales and how to arrange objects in that space. The interface therefore defines two “space” modes: fixed or varies. Under the “varies” mode, the cloning process is the same as PORT-C. Under the “fixed” space mode, however, the yellow box becomes a constrained volume that the cloned objects can occupy. Any objects outside the boundary are automatically deleted by the system. In this case, the number of clones and the distances between clones are dependent. Changing one affects the other.

4.4 Dynamic slider widget interface (D-Slider)

The *dynamic-slider user interface* (Figure 6(a)) has seven widgets displayed on a larger tablet (size of 48cmx35cm): three for the number of copies, three for distance and one for toggling the direction of copies. Four slider widgets are visible at a time and the user can toggle the direction by physically rotating the tablet. For example, moving the tablet to align it roughly with the horizontal plane will made the X- and Z-axis sliders visible. Users can pre-define the range of the slider based on the task requirement [12].

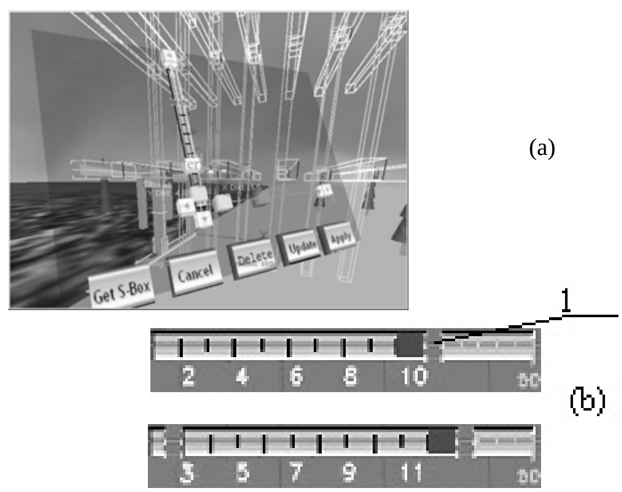


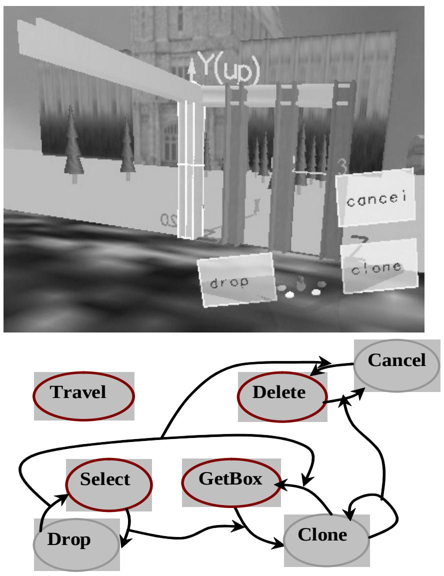
Figure 6: Dynamic slider user interface and sliders’ behavior

Figure 6(b) illustrates the behavior of the slider slot. The slider slot is divided into three parts by gap 1. The slider slot will slide to the right and scale up its scale when the projected position of the wand falls on the right side of gap 1. Moving the wand back to the left of the gap will stop the sliding so that users can start dragging the slider to set up numbers. No sliding if the wand falls into gap 1 in order to avoid the accidental movement of the wand beyond the working area of the slider.

4.5 Copy by example interface (copy-by-example)

The previous user interfaces rely on widgets or metaphoric interfaces to define the parameters. Without considering the parameters, the cloning task is simply an object creation and placement task like the one in [8]. Placement with a technique like HOMER [6] could be more direct; however, precise placement is tedious and time consuming if users have to travel a large distance for placing objects out of reach.

We designed the *copy by example interface* (Figure 7), that only requires users to place the first object, then the program automatically calculates the position of the next and creates a ghost copy there. This copy is added to the environment if another cloning command is issued. Users can adjust the location of the ghost copy and the user interface will automatically updates the relative location for placing the next.



(a)

(b)

Figure 7. Copy-by-example interface and state transitions

This interface utilizes the "repetitive" characteristic of the structure to avoid precise object placement. It also increases the flexibility of building repetitive shapes like stairs, spiral shape object etc, which may not be possible with the other cloning interfaces.

4.6 Virtual keypad (keypad)

Previous study [12] suggested that numeric input was an important complementary tool to direct or widget-based cloning. Numeric input is also widely used in desktop tools for computer-aided design. Unfortunately, not many numeric input techniques for VEs exist. We have designed the *virtual keypad interface* (Figure 8) on tablet to support fundamental text editing: insertion, deletion, clear etc. Users click a wand button to issue an operation. Figure 8 illustrates an example of setting up five copies along the Y direction. The user can also choose the parameter s/he wants to set up by selecting one of the six widgets on the tablet. This action will cause the keypad to be relocated to the position of the widget.

5 EXPERIMENTAL STUDIES

Even though the elementary parameters of creating and editing objects in the cloning user interfaces may be the same, the different context of use and domain constraints leads to different techniques.

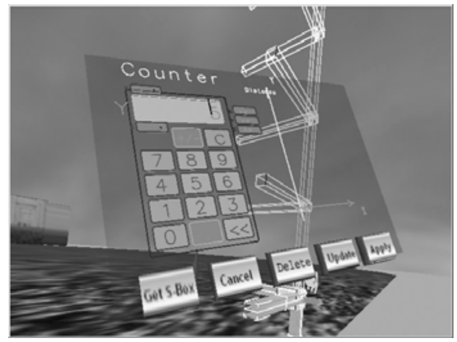


Figure 8. Keypad user interface

Some interfaces are more generic, i.e., applying keypad or slider to input numbers, while others are more specific, i.e., applying the space-metaphor interface to preset the space a building would occupy. Studies need to be done to learn their usability problems and applicability to the AEC domain.

We performed two comparative studies. Study I focuses on the effectiveness of cloning to provide an evidence of the power of considering the cloning task, a domain-specific task. Study II focuses on the comparison of the five cloning user interfaces.

5.1 Study I: Effectiveness of cloning

Our goal is to investigate the effectiveness of cloning in general. The research hypothesis is that performance improves when the cloning techniques are used.

5.1.1 Experimental Design

In order to single out the effect of the cloning task, we compared an interface with cloning to one without cloning. To meet this requirement, we picked the PORT cloning interface and a Naïve HOMER direct manipulation interface. The Naïve HOMER interface used was similar to standard copy-and-paste: users have to place objects one after another. The user interface was similar to copy-by-example with automatic object placement disabled. The "clone" button was labeled "paste".

The independent variable is the interaction technique and the dependent variable is task completion time. This study includes eight participants and uses a within-subjects design.

5.1.2 Tasks

Three tasks were chosen considering different buildings a structural engineer is likely to design in a VE.

Task 1 (building task) required participants to add two pieces one by one to an existing structure, then to raise the building up to four stories.

Task 2 (numeric input task) was designed for modeling a large repetitive structure. An example task was "Select that one column and two beams in the environment. Construct a two story building such that a five by five grid is created in the horizontal plane."

Finally, task 3 (picture matching task, Figure 9) required the participant to build a structure to match an existing design displayed as a two-dimensional (2D) image on the tablet. This task corresponds to the situation when the designer has finished a draft on a paper and brought it to a VE to construct the structure.

5.1.3 Environment

The virtual world was based on the campus of our university, where participants had a good mental map and would not get disoriented. In other words, navigation was not a confounding factor for this study.

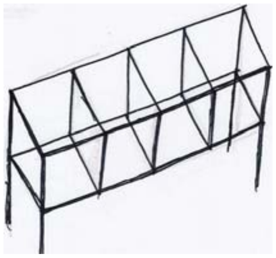


Figure 9. The structure for task condition 3

5.1.4 Procedure

Eight participants (20-27 years old, males) volunteered for this study. They were five senior undergraduate students and three graduate students whose major was structural engineering. They were familiar with structural analysis and architectural design tools and used these tools daily.

The procedure consisted of two sessions. In the first session, subjects were given two ETS standard psychology paper tests [13]. The first test was the perceptual speed test, defined as "speed in comparing figures or symbols, scanning to find figures or symbols, or carrying out other very simple task involving visual perception". Participants were asked to compare a group of numbers, two in each group and mark those that are different. The second test was the Figure Fluency test, used to test ability to produce new figures from a given set of elements. Participants were asked to think of different ways to add details to a design. This section lasted about 15 minutes. These tests were chosen not because they were related to VEs, but because they were related to human cognition in general. We used the tests because we wanted to learn if there was any correlation between users' aptitudes/abilities and users' task performance or preferences. We then may be able to design user interfaces that accommodate their needs [25].

During the second session participants completed a background questionnaire, were given instructions and completed two sets of trials (one with each interface). We then administered a written survey and oral follow-up questions. Participants were allowed to take a break any time. Interfaces were counter-balanced between participants. Tasks were not and executed in the order specified previously. Participants were told to do the tasks as fast as they could while avoiding errors.

The pre-questionnaire recorded participants' background in architecture, engineering, work habits and game and computer experiences. During the practice tasks the participants could familiarize themselves with the system and the different interaction techniques. During the experiment, they were allowed to finish task 1, but were given time constraints (one minute) for tasks 2 and 3 (one and half minutes). We limited the time of the tasks in order to reduce the fatigue that may be associated with the Naïve HOMER user interface (which may require about 50 copy and paste operations for some of the tasks).

5.2 Study I: Results and discussion

All participants completed the experiment. The experiment lasted about 1.5 hours. Participants performance and subjective ratings suggested that, as predicted, the cloning interface overall produced a better performance than the non-cloning interface. This conclusion was drawn from the following analyses: (1) For task 1, a single-factor Analysis of Variance (ANOVA) on interaction techniques was performed. Interaction technique was not significant ($F(1,15)=2.64$, $p=0.14$). However, the overall task completion time for PORT-C (mean=68.4, stdev=20.2) was faster than Naïve HOMER (mean=82.3, stdev=26.5). (2) For task 2, participants' performance was measured by the number of objects created in a given time period divided by the total number of objects required upon completion (50). PORT-C (mean=93%) was about four times more effective than the naïve HOMER interface (mean=23%) and this difference was significant ($F(1, 15)=76$, $p<0.0001$); (3) the same calculation was used for task 3. PORT-C (mean=95%) was 1.7 times faster than naïve HOMER (mean=56%) and this difference was also significant ($F(1, 14)=11.7$, $p<0.01$).

A single factor ANOVA on the participants' rating indicated that the difference in participants' preference was significant only for the numeric input task ($F(1, 15)=57$, $p=0.0001$; mean = 6.5 for PORT-C; mean = 2.25 for naïve-HOMER). The difference was not significant for task 1 (mean=6 for PORT-C and mean=4 for naïve HOMER) and task 3 (mean=5 for PORT-C and mean=4.5 for naïve HOMER). The perceived usefulness and perceived ease of use were not significant for any of the task conditions. Participants did rate the PORT-C interface higher, though two out of eight participants preferred Naïve HOMER to perform the building and matching tasks.

Participants' comments matched what we expected: naïve HOMER is intuitive, but objects placement is difficult. PORT-C is suitable for building large structures. Also, there is no need to place objects directly with PORT-C, which can be difficult when objects are faraway and movement is sensitive to hand movement.

From the cognitive and perceptual ability tests given to each subject, the perceptual speed test scores showed high correlation with task completion time with the PORT-C interface ($r=-0.82$, $p=0.01$) but not with naïve HOMER. This may have confirmed that people with higher perceptual speed had better performance [1]. No correlation was found with naïve HOMER might be that participants spend time on object placement or travel which did not require perceptual speed.

Participants' self-rated game experience was also correlated with task completion time ($r=-0.69$, $p=0.05$). Their self-rated computer experience did not correlate with task performance ($r=0.58$, $p=0.13$). No correlation was found significant between task completion time and the figural test score.

5.3 Study II: Comparison of cloning techniques

5.3.1 Experimental Design

The goal of this experiment is to compare the performance of the five cloning user interfaces (sections 4.2-4.6) and to illustrate some strengths and weaknesses of each technique for the AEC domain and for particular tasks within that domain.

The study used a 2x5 (Groups: G1, G2 x Interface: PORT-C, space metaphor, D-slider, copy-by-example, keypad) mixed design (Table 2). Group 1 participants used the PORT-C, space metaphor and keypad interfaces, while group 2 used the space metaphor, D-slider and copy-by-example interfaces. Each group was a within-subjects design, but group was a between-subjects factor. It would be ideal to use a complete within-subjects design for comparison purposes, but we sacrificed this to avoid the aftereffects associated with prolonged use. The third task (picture matching task) was not used with the keypad interface for the same reason and also because the keypad was designed as complementary tool rather than a design tool.

Table 2 Experimental design

	PORT-C	Space metaphor	D-slider	Copy-by-example	Keypad
G1	8	8			8
G2		8	8	8	

Participants were assigned to one of the two groups randomly. The order in which participants used the interfaces was randomized within each group to minimize any order effects. For each user interface, participants performed three tasks in each block with the same order as in Study I.

5.4 Procedure

Sixteen student participants volunteered for this study (age 20-32 years, 15 male and one female, 12 in architecture and 6 in engineering (two were double-major)). All students used structural analysis and/or architectural design tools daily. Two of them had used a wand previously for navigation in a CAVE environment. The experiment procedure was the same as Study I but all participants in this study were requested to finish all three tasks and were asked to finish each as fast and as accurately as they could.

5.5 Results and discussion

The experiment lasted about 2 to 2.5 hours. All participants finished the experiment except one who dropped due to motion sickness. Again, two sets of trials were performed, but the data from the first trial was not used for statistical analysis.

5.5.1 Performance

The results in terms of the overall task completion time and cloning time for the second trial using five user interfaces are shown in Figure 10. The horizontal axis corresponds to three tasks participants performed. The vertical axis is the task completion time on overall performance or cloning time. The time for cloning is the accumulation of all cloning operations that started at the time participants clicked a button to indicate cloning until the time another command was issued to finish the cloning operation. We first performed a between groups comparison on task completion time. Results did not show significant difference between these two groups (F(1, 112)=3.06, p=0.1)

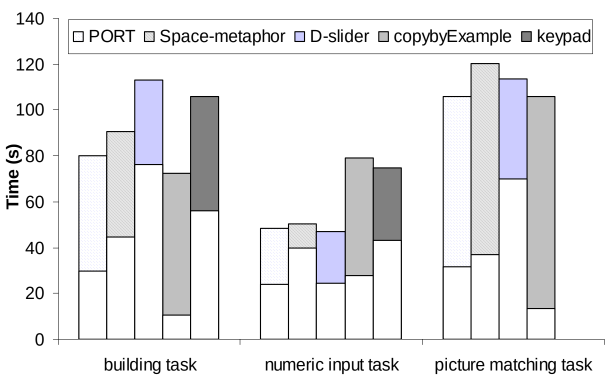


Figure 10. Task completion time (Bottom bar is cloning time; full bar is overall task completion time)

We performed a two-factor general linear model (GLM) procedure on interaction techniques and tasks. For overall task completion time, (1) interface was not significant (F(4, 123) = 0.77, p=0.83); (2) task was significant (F(2, 123) = 9.4, p=0.0042); and (3) The two-way interaction of interaction technique * task was not significant (F(4, 123)=1.17, p=0.34).

For cloning time, (1) interaction technique was significant (F(4, 123)=9.05, p<0.0001); (2) task was not significant (F(1, 123)=0.17, p=0.68); (3) the two-way interaction was not significant (F(4, 123)=0.55, p=0.69).

We then performed a post-hoc analysis using Tukey test on task completion time and cloning time for each task. All significant factors are listed in Table 3 and detailed below in this section. In this table, symbols starting with A represent overall task completion time and symbols starting with C represent cloning time. Each symbol followed by a number, which is the task number (1 is the building task, 2 is the numeric input task, and 3 is the picture matching task.) The cell location indicates the two interfaces that were significantly different, with the interface labeling the column being significantly faster than the interface labeling the row. For example, C3 in the fourth column, second row means that copy-by-example was significantly faster than space-metaphor on cloning time for task 3.

Table 3. Significant factors – (SM: space-metaphor, cByE: clone-by-example)

	PORT-C	SM	D-Slider	CbyE	Keypad
PORT-C					
SM				C1, C3	
D-Slider	A1, C1 C3	C1		A1, C1,C3	
CbyE	A2	A2			
Keypad	C1, A2	A2		C1	

For task 1 on overall completion time, interaction technique was significant (F(4, 25)=3.64, p=0.018). A post-hoc analysis using Tukey test on differences of Least Squares Means showed that copy-by-example was significantly better than the slider interface (t=2.34, p=0.02) and PORT-C was significantly better than slider (t=2.03, p=0.04).

For task 1 on cloning time, interaction technique was significant (F(4, 26)=11.9, p<0.0001). A post-hoc analysis using Tukey test showed that PORT-C was significantly faster than D-slider (t=3.85, p=0.0004) and keypad (t=2.27, p=0.03). PORT-C and copy-by-example also significantly outperformed keypad and space-metaphor (t=3.63, p=0.0008 and t=3.04, p=0.004 accordingly).

For task 2 on overall task completion time, interaction technique was significant (F(4, 26)=6.35, p=0.001). A post-hoc analysis using Tukey test indicated that all user interfaces with specific consideration of the cloning task outperformed the other interfaces. This was based on the following observations: (1) PORT-C was significantly better than copy by example (t=2.5, p=0.02) and keypad interface (t=2.17, p=0.03); (2) space-metaphor was significantly better than copy-by-example (t=2.58, p=0.01) and keypad (t=2.28, p=0.02); (3) slider was also significantly better than copy-by-example (t=2.5, p=0.02) and keypad (t=2.19, p=0.03). There was no significant difference among the PORT-C, space-metaphor or slider user interfaces.

For task 2 on cloning time, interaction technique was not significant (F(4, 44)=1.52, p=0.2).

For task 3 on overall completion time, interaction techniques was not significant (F(4, 17)=1.6, p=0.2).

For task 3 on cloning time, interaction technique was significant (F(3, 17)=7.3, p=0.002). A post-hoc analysis using Tukey test indicated that space-metaphor, PORT, and copy-by-example were significantly better than the slider interface (t=3.27, p=0.003; t=3.37, p=0.002; t=4.39, p=0.0001 accordingly.) Copy-by-example was significantly better than the space metaphor and D-slider techniques (t=2.07, p=0.046; t=4.39, p=0.0001 accordingly).

5.5.2 Discussion

Performance data indicates that each user interface has its merits for specific task conditions. Those interfaces that are more specifically designed for cloning (such as PORT-C, space-metaphor and D-slider) outperformed those more generic ones (such as copy-by-example and keypad) for tasks that require the creation of a large number of elements (e.g., task 2). Otherwise, copy-by-example is the best fit for creating a small number of objects, such as adding two missing columns to a structure in task

1. These results further confirmed the hypothesis made in experiment I that cloning is an effective way to model structures when large repetitive patterns exist.

The PORT-C interface, in most cases, outperformed other user interfaces, perhaps because of its simplicity and directness. Hollan, Hutchins and Norman [15] coined the term "directness", and characterized it into two aspects: distance and engagement. Engagement in direct manipulation means the user is engaged with the objects themselves. So PORT-C may have high engagement while being used for cloning and lower turn-around time because no operating on widgets is required.

Widgets were implemented as part of the pen-and-tablet user interface for the D-slider, keypad and space-metaphor techniques. Using the pen-and-tablet metaphor added at least two extra time costs: a "transitional time" between the tablet and the 3D world and "operational time" on widgets. The reason that D-slider was slow might be the high cost of "operational time." This is at least partially due to the input device (a wand) we used and its lack of support for the pen-and-tablet style. A stylus or pen-like device would clearly be better. A few participants held the wand like a pen during the experiment when operating on the tablet.

The disadvantage of the pen-and-tablet metaphor, however, seemed diminished in the space-metaphor user interface. Cloning time was significantly shorter than D-slider for task 1. This might be because the operational time was minimal so that the only tablet-related cost associated with the space-metaphor interface was the transition time. The space-metaphor interface used large buttons that were easy to select and click and avoided precise manipulation. In fact, one participant mentioned that he felt like he was using a touchscreen on the tablet while changing modes.

The "touchscreen" interface also differs from D-slider and keypad in how users switched their attention. We observed that participants "causally" clicked on the large buttons and switched back to the world to continue their work. This also had a negative effect, however, because participants were unlikely to check the current states of the interface, producing high error rates.

PORT-C and space-metaphor use two distinct styles of direct manipulation: PORT-C is modeless and the space-metaphor used modes. The lack of significant differences between these two was likely due to the operational time. However, PORT-C could keep participants' focus on the task because of the intuitive transition of button functions and its ease of use. Participants learned the automatic mode transitions during training and felt the mode changes were natural.

Interestingly, copy-by example outperformed most other interfaces under task conditions 1 and 3. The reason might be the ease of cloning. By our observation, participants quickly placed the first copy (with snap) and then clicked the button to create more. With other cloning user interfaces, participants were very careful while editing numbers with joystick to avoid overshooting. It took longer to make fine adjustments. This is confirmed by the performance data with significant differences in cloning time for task 3.

Participants also relied on counting the number of button clicks combined with the sound cue to confirm their operation, rather than constantly tracking every copy when creating large number of objects. This was confirmed by our observation. Most participants with the copy-by-example interface immediately confirmed their operation after cloning objects rather than counting the objects to make sure.

Examining copy-by-example carefully, we found that the bottleneck for its low performance on task 2 was the larger number of selections participants had to perform. All other interfaces required selecting objects once for this task, but copy-by-example required three selections. We evaluated the selection time with a two-factor GLM procedure across all interfaces on task 2. Results confirmed that interface was significant ($F(4, 44)=4.64$, $p=0.004$). A post-hoc Tukey test indicated that copy-by-example was significantly slower than all other user interfaces for selection time: (PORT-C: $t=2.43$, $p=0.02$, space-metaphor: $t=3.82$, $p=0.0005$, D-slider: $t=3.72$, $p=0.0006$ and keypad: $t=2.09$, $p=0.04$.) No other groups were significant.

5.5.3 Users' behavior

A behavior we did not anticipate was that participants changed their modeling strategies while using different user interfaces. This behavior may have lowered task performance using copy-by-example. Two strategies emerged for task 2: three out of eight participants made up to six single-object creations then replicated the second level to form the structure while using copy-by-example user interface; however they replicated the initial one column and two beams to form a 5x2x2 structure and then deleted extra copies while using PORT-C. The six object placements in the first strategy were time consuming. The short cloning time seen in figure 9 for copy-by-example on this task was because placement time was not counted as part of the cloning time.

This behavior was confirmed with participants' comments: "copy-by-example facilitated direct grasp and drop, so I was unwilling to delete objects, while the other cloning interfaces made object creation so easy that I did not mind deleting objects later".

Fatigue was observed due to prolonged use. Most participants asked if they could sit down and work. They also avoided looking around with the HMD on and put down the tablet or changed to a different grasp. They reported gesture-associated with fatigue with PORT. These results suggested that we may need to consider the design of HMD interfaces suitable for seated use.

5.5.4 Subjective ratings, comments, and others

There were no significant differences for preference rank and perceived usefulness among the interfaces. However, participants did prefer and therefore ranked the PORT-C and the space metaphor interfaces for cloning highly (PORT: 6.1, Space-metaphor: 5.4, slider 3.2, copy-by-example: 5, keypad: 3.3.)

Participants commented that they mostly preferred copy-by-example for simple tasks and PORT or space-metaphor for large modeling work. This could also be explained by saying that copy-by-example represents simple tasks and domain objects well while the other interfaces represent complex cloning tasks well. When participants created one element, they perceived that the task required a single object copy and placement rather than a cloning operation. Participants did not like the keypad because it required too many clicks, which may suggest that handwriting, such as [24], should be used for numeric input.

Participants preferred the smaller tablet compared to the larger one (used for slider widgets interface) unanimously because it was light and easy to hold with a flexible grasping posture. These results may imply that we can even go further and use the hand rather than a tablet as the interactive device [18]

We again found a strong correlation of self-rated game experience with task 1 completion time ($r=-0.3$, $p=0.03$), but game experience had no effects on their preferences for user interfaces ($r=-0.12$, $p=0.4$). This would suggest that the game player may not care very much about the style of the user interface since they could master most interfaces quickly. Computer experience was correlated ($r=-0.47$, $p=0.006$) with task 2 completion time, but game play was not ($r=0.23$, $p=0.19$). Additionally, no correlation was found for task 3 with regard to either computer experience ($r=-0.1$, $p=0.5$) or game experience ($r=0.2$, $p=0.2$). We may infer that that computer and game experiences were associated with performance under certain task conditions only.

5.5.5 Summary

A major take away lesson was that there may be no single best technique for real-world applications because real world applications include many different tasks. Different techniques match well with different task situations. Our experiment also suggested that we should use simple user interfaces like copy-by-example for simple tasks (e.g., place one piece of element.) However, user interfaces should be specifically designed for certain task conditions, such as cloning, if applicable to a domain. The transfer of 2D WIMP user interface needs a careful consideration, especially for the sliders and keypad user interface. One way to improve the design of keypad, however, is to map keys to the wand buttons.

6 CONCLUSION AND FUTURE WORK

We designed several new useful and usable techniques for the cloning task. There were several interesting tradeoffs. We have made progress in bridging the gap between the requirements of complex real-world VE applications and the design of 3D interaction techniques through designing interaction techniques for domain-specific tasks. We believe the task space can be used to expand the taxonomy at generic level and techniques are reusable to solve problems with similar characteristics.

One key point to make is that the design of interaction techniques could extend across the traditional boundaries of generic tasks. Tasks could be formed in the context of an application by integrating domain knowledge into the formation of tasks. We can draw the conclusion from the experimental studies that designing for the cloning task was different from designing for the object manipulation task.

This work, however, was limited by our understanding of domain-specific design. Other domain knowledge has been considered but needs to be further investigated for its applicability to AEC. We are currently working on categorizing domain knowledge, studying empirically the impact of domain knowledge on design and integrating domain-specific techniques in a real-world application.

Many interesting questions raised during this process are proposed as future work:

- What is a logical design process to develop domain-specific tasks or techniques?
- How can we organize the domain knowledge in a useful manner to benefit the design of interfaces?
- Could domain-specific interaction reduce the cost of integration and reduce conflicts between techniques?
- How would domain-specific design affect existing taxonomies and design spaces?

ACKNOWLEDGEMENT

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.

REFERENCE

[1] Allen, B., "Individual differences and the counundrums of user centered design: two experiments," *Journal of the american society for information science*, vol. 51, 508-520, 2000.

[2] Angus, I. and Sowizral, H., "Embedding the 2D Interaction Metaphor in a Real 3D Virtual Environment," Stereoscopic Displays and Virtual Reality Systems, 2409, 282-293, 1995.

[3] Bier, E., "Snap-dragging in three dimensions," Symposium on Interactive 3D Graphics, 193-204, 1990.

[4] Bowman, D.A., "Conceptual Design Space - Beyond Walkthrough to Immersive Design," *Designing Digital Space*. New York: John Wiley & Sons, 1996, pp. 225-236.

[5] Bowman, D.A., "Survey results on productoin VR/AR applications," 2005.

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

[7] Bowman, D.A., Kruijff, E., LaViola, J.J. and Poupyrev, I."*3D User Interfaces: theory and practice"*, 1st ed. Boston: Addison-Wesley, 2004.

[8] 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," IEEE Virtual Reality, 243-250, 2003.

[9] Brooks, F.P., "Walkthrough - a Dynamic Graphics System for Simulating Virtual Buildings," Symposium on Interactive 3D Graphics, 9-21, 1986.

[10] Brooks, F.P., "What's real about virtual reality," *IEEE Computer Graphics & Applications*, vol. Nov/Dec, 16-27, 1999.

[11] Butterworth, J., Davidson, A., Hench, S. and Olano, M.T., "3DM: a Three Dimensional Modeler Using a Head-Mounted Display," Symposium on Interactive 3D Graphics, 135 - 138, 1992.

[12] *Covered for blind review*

[13] Ekstrom, R.B., French, J.W. and Harman, H.H."*Manual for kit of factor referenced cognitive tests"*. Princeton, NJ: Educational testing service, 1976.

[14] Hix, D., II, J.E.S., Gabbard, J.L., McGee, M., Durbin, J. and King, T., "User-Centered Design and Evaluation of a real-time battlefield visualization virtual environment," IEEE Virtual Reality, 96-103, 1999.

[15] Hutchins, E.L., Hollan, J.D. and Norman, D.A., "Direct manipulation interfaces," *Human-computer interaction*, vol. 1, 311-338, 1985.

[16] Jacob, R., Siber, L.E., McFarlane, D.C. and Muller, M.P., "Integrality and Separability of Input Devices," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 1, 3-26, 1994.

[17] Kitamura, Y., Yee, A. and Kishino, F., "A Sophisticated Manipulation Aid in a Virtual Environment using Dynamic Constraints among Object Faces," *Presence: Teleoperators and Virtual Environments*, vol. 7, 460-477, 1998.

[18] Kohli, L. and Whitton, M., "The haptic hand: providing user interface feedback with the non-dominant hand in virtual environments," Graphics interface, 2005.

[19] Liang, J. and Green, M., "JDCAD: A Highly Interactive 3D Modeling System," 1994.

[20] Lucas, J.F., Kim, J. and Bowman, D.A., "Resizing beyond widgets: object resizing techniques for immersive virtual environments," ACM CHI, 1601-1604, 2005.

[21] Mine, M., "Virtual environment interaction techniques," UNC Chapel Hill CS Dept., Technical Report TR95-018, 1995.

[22] Mine, M., "ISAAC: A Meta-CAD System for Virtual Environments," *Computer-Aided Design*, vol. 29, 547-553, 1997.

[23] Oh, J.-Y. and Stuerzlinger, W., "Moving objects with 2D input devices in CAD systems and desktop virtual environments," Graphics interface, 2005.

[24] Poupyrev, I., Tomokazu, N. and Weghorst, S., "Virtual Notepad: Handwriting in Immersive VR," VRAIS, 126-132, 1998.

[25] Stanney, K.M., Mourant, R.R. and Kennedy, R.S., "Human factors issues in virtual environments: a review of the literature," *Presence: Teleoperators and Virtual Environments*, vol. 7, 327-351, 1998.

[26] Sutcliffe, A.G., Benyon, D. and van Assche, F."*Domain knowledge for interactive system design"*: Chapman & Hall, 1996.