

# An Empirical Evaluation of Graspable User Interfaces: towards specialized, space-multiplexed input

George W. Fitzmaurice <sup>2,1</sup>

<sup>1</sup> Alias\Wavefront  
 110 Richmond St. East  
 Toronto, Ontario,  
 CANADA M5C 1P1  
 Tel: +1 (416) 362-9181  
 E-mail:{gf, buxton}@aw.sgi.com

William Buxton <sup>1,2</sup>

<sup>2</sup> Dynamic Graphics Project, CSRI  
 University of Toronto  
 Toronto, Ontario,  
 CANADA M5S 1A4  
 Tel: +1 (416) 978-1531

## ABSTRACT

This paper reports on the experimental evaluation of a Graspable User Interface that employs a "space-multiplexing" input scheme in which each function to be controlled has a dedicated physical transducer, each occupying its own space. This input style contrasts the more traditional "time-multiplexing" input scheme which uses one device (such as the mouse) to control different functions at different points in time. A tracking experiment was conducted to compare a traditional GUI design with its time-multiplex input scheme versus a Graspable UI design having a space-multiplex input scheme. We found that the space-multiplex conditions out perform the time-multiplex conditions. In addition, we found that the use of specialized physical form factors for the input devices instead of generic form factors provide a performance advantage. We argue that the specialized devices serve as both visual and tactile functional reminders of the associated tool assignment as well as facilitate manipulation due to the customized form factors.

**KEYWORDS:** input devices, graphical user interfaces, graspable user interfaces, haptic input, two-handed interaction, augmented reality, ubiquitous computing

## INTRODUCTION

Graspable User Interfaces [2, 4] advocate providing users concurrent access to multiple, specialized input devices which can serve as dedicated physical interface widgets, affording physical manipulation and spatial arrangements. These physical artifacts are essentially a collection of input devices that can be tightly coupled or "attached" to virtual objects for manipulation or for expressing action (e.g., to set parameters or for initiating processes).

Permission to make digital/hard copies of all or part of this material for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication and its date appear, and notice is given that copyright is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires specific permission and/or fee.

CHI 97, Atlanta GA USA

Copyright 1997 ACM 0-89791-802-9/97/03 ...\$3.50

## Space-multiplex versus Time-multiplex input

Input devices can be classified as being *space-multiplexed* or *time-multiplexed*. With space-multiplexed input, each function to be controlled has a dedicated transducer, each occupying its own space. For example, an automobile has a brake, clutch, throttle, steering wheel, and gear shift which are distinct, dedicated transducers each controlling a single specific task. Each transducer can be accessible independently but also possibly simultaneously. A space-multiplex input style affords the capability to take advantage of the shape, size and position of the multiple physical controllers to increase functionality and decrease complexity. It also means that the potential persistence of attachment of a device to a function can be increased.

In contrast, time-multiplexing input uses one device to control different functions at different points in time. Hence, the device is being repeatedly attached and unattached to the various logical functions of the GUI. For instance, the mouse uses time multiplexing as it controls functions as diverse as menu selection, navigation using the scroll widgets, pointing, and activating "buttons."

Consider the task of text entry with a space- versus time-multiplex input style. The time-multiplex input scheme could, for example, employ a Morse code button where a single key is used to encode each alpha numeric character. In contrast, the space-multiplex input scheme could use a QWERTY keyboard where there is one key per character.

## Graspable functions not devices

Beyond a space-multiplexing input style, we are proposing a conceptual shift in thinking about physical input devices not as graspable *devices* but instead as graspable *functions*. In the traditional sense, almost all physical input devices are "graspable" in that one can physically touch and hold them. However, we are exploring the utility of designing the physical devices as graspable functions. This can best be shown in Figure 1. With traditional GUIs there are often three phases of interaction: (1) acquire physical device, (2) acquire logical device (e.g., a UI widget such as a scrollbar

or button) and (3) manipulate the virtual device. Note, however, that in some cases, acquiring the physical device in a GUI can be trivial (e.g., there is only one mouse) or can be eliminated if the hand remains on the device. Alternatively, with Graspable UIs, we can often reduce the phases of interaction to: (1) acquire physical device and (2) manipulate the logical device directly. This is possible because the physical devices can be persistently attached to a logical device. Thus, the devices serve as dedicated graspable functions.

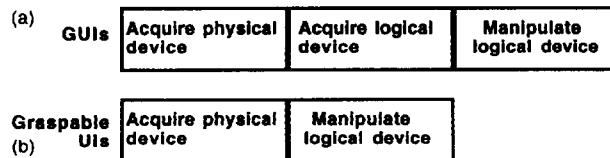


Figure 1. Phases of interaction for (a) traditional GUIs and (b) Graspable UIs.

Having a dedicated physical input device for every function can be costly and potentially inefficient. Figure 2 shows an example of two input configuration styles: the time-multiplexed mouse and the space-multiplexed audio mixing console. The mouse is a generic all-purpose pointing device which is constantly attached and detached to logical devices. In contrast, the audio mixing console has hundreds of physical transducers (e.g., sliders, dials, buttons) each assigned a function. Which input configuration is more desirable, more direct or more manipulable? We believe the ultimate benefits lie somewhere in between these two extremes.

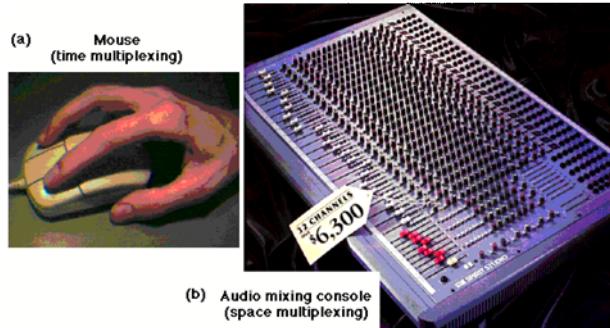


Figure 2. The mouse, a time multiplexed design (a) and an audio mixing console, a space multiplexed design (b).

## RELATED WORK

The origins of a Graspable UI are rooted in many systems and research initiatives aimed at improving the quality of interaction while at the same time reducing complexity by leveraging off of people's understanding of physical objects and physical manipulations [2, 10].

The passive interface props approach [7] is an example of using real objects with embedded 6 DoF sensors to capture the natural dialog a user has with physical objects. In their neurosurgical visualization application, doctors hold and manipulate two objects: a doll's head and a rectangular plate, used for specifying the cutting-plane for a patient's

head data. Each object has a specialized form factor and functional role in the interaction: manipulating the doll's head specifies the camera view for the 3D model while the plate specifies the cutting-plane. The 3-Draw system [13] has a similar set-up but in the context of a CAD and modeling package where it uses a plate and stylus each being tracked by the computer using an embedded 6 DoF sensor.

The LegoWall prototype (developed by Knud Molenbach of Scaitech and LEGO) consist of specially designed blocks that fasten to a wall mounted panel composed of a grid of connectors and electronics to track the position and identity of each brick. In a shipping application, bricks are used to represent objects (e.g., ships) and actions (e.g., print or display schedule). The wall panel is divided up into spatial regions where a column represents a shipping port. Users interact with the system by moving bricks on the wall board as the ships travel to different ports and also to execute commands by placing action bricks next to ship bricks (e.g., to print the schedule of a ship, place the "print" brick next to the ship brick). This system illustrates the Graspable UI philosophy of physically instantiating components of the UI to tap into our skills at physical manipulations and spatial layout.

Wacom Technologies Inc. has explored the concept of having specialized "character devices," what they call electronic stationary, in which devices have a unique shape and a fixed, predefined function associated with it [5]. The idea is that the form or shape of the device reveals or describes the function it offers. Three character devices were defined: (1) eraser, which functioned to erase electronic ink, (2) ink pot which served to select from a color palette and (3) a file cabinet which brought up a file browser to retrieve and save files (see Figure 3).



Figure 3. Wacom character devices

A number of systems are being investigated that serve to bridge and blend interactions that span both the physical and virtual mediums. These systems often are characterized within the augmented reality or ubiquitous computing fields. The Bricks prototype [4] uses physical bricks as handles of controls for manipulating virtual objects or for expressing actions within a simple 2D drawing program. The Chameleon [3] serves as a spatially-aware palmtop device that provides a virtual window onto physical

artifacts. Wellner's DigitalDesk [11] merges our everyday physical desktop with paper documents and electronic documents. This is achieved by projecting a computer display down onto a desk and pointing video cameras at the desk which use image-analysis techniques to sense what the user is doing. All of these system use physical artifacts as input devices but strive to blend the UI components (some physical some virtual) to take advantage of the strengths and affordances of both mediums. The Graspable UI approach advocates this same philosophy.

Finally, the design goals of Graspable UIs are guided by research in areas such as 2-handed interactions [6], the use of physical artifacts as cognitive aids [12, 15] and the intelligent use of space and spatial arrangements to simplify choice, perception and internal computation [8, 9].

### THE EXPERIMENT

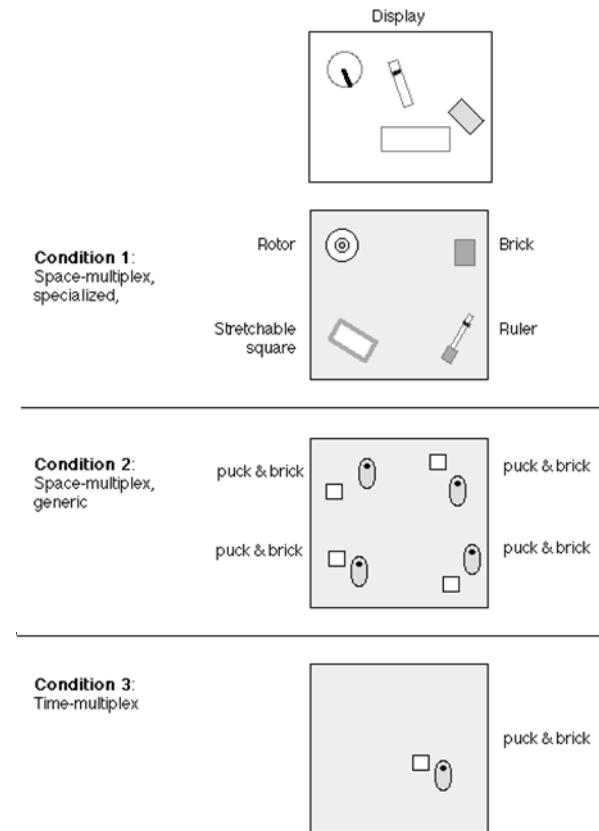
In this experiment we focus on the issue of space-multiplexed versus time-multiplexed input and examine the inter-device transaction phase of interactions. That is, the experiment is designed to study the relative costs of acquiring physical devices (in the space-multiplex conditions) versus acquiring virtual logical controllers (in the time multiplex condition). Moreover, we investigate the utility of specialized physical form factors (where the physical shape of the device roughly matches the shape and manipulation characteristics of the virtual logical controller) versus generic form factors for input devices.

The experiment has subjects continuously track four randomly moving targets on the computer screen (see Figure 4). The four targets can be considered four user interface widgets which a user manipulates during a compound task or workflow. Two of the targets (rotor and brick) require position and rotation adjustments while the other two targets (stretchable square and ruler) require position, rotation and scale adjustments. The continuous pursuit tracking task was chosen to emphasize the inter-device transaction phase, not the manipulation phase (as was explored in a previous experiment [2]).

With the space-multiplexed conditions, the physical input devices are permanently assigned and attached to a virtual, logical widget. Thus, to manipulate an on-screen widget, the subject directly manipulates the physical device. In contrast, the time-multiplex condition uses only one set of input devices which must be attached and detached to each logical widget before it is manipulated. Thus, subjects never need to release the physical input devices in the time-multiplex condition. Condition 1 uses specialized input devices (the rotor, brick, stretchable square and ruler) while condition 2 uses a generic puck and brick pair for each logical widget (thus a total of 4 pucks and 4 bricks are used).

All of the input devices operate on Wacom digitizing tablets and thus are spatially-aware and are untethered. Subjects were encouraged to use as much concurrency as

possible. Finally, note that the multiple target tracking task was designed as a two handed task.



**Figure 4.** Three experimental conditions.

**Hypothesis 1.** *Subjects perform better with space-multiplex than time multiplex input conditions.*

We predict that subjects will have superior performance for the space-multiplexed conditions over the time-multiplexed input condition. This is primarily due to the persistence of attachment between the physical input devices and the assigned virtual, logical widgets. We speculate that the physical input devices are easier to acquire than the corresponding virtual handles in the time-multiplex condition. Moreover, the space-multiplex conditions offer a greater potential for concurrent access and manipulation of virtual widgets by providing continuous access to the physical handles. Part of this investigation tests whether subjects utilize this extra concurrency capability.

**Hypothesis 2.** *In space-multiplex conditions, subjects perform better with specialized than generic devices.*

Within the space-multiplex conditions, we predict that the specialized input devices will allow for superior task performance compared to the generic devices. Again, the specialized form factor should serve to remind the subject what virtual widget is attached to the device as well as facilitate the manipulation of the widget.

## METHOD

### Subjects

Twelve right-handed subjects participated in the experiment. All subjects except two had minimal exposure to operating a tablet device. Ten of the subjects use a computer on a daily basis. Finally, all subjects were naive to the purpose and predictions of the experiment.

### Equipment

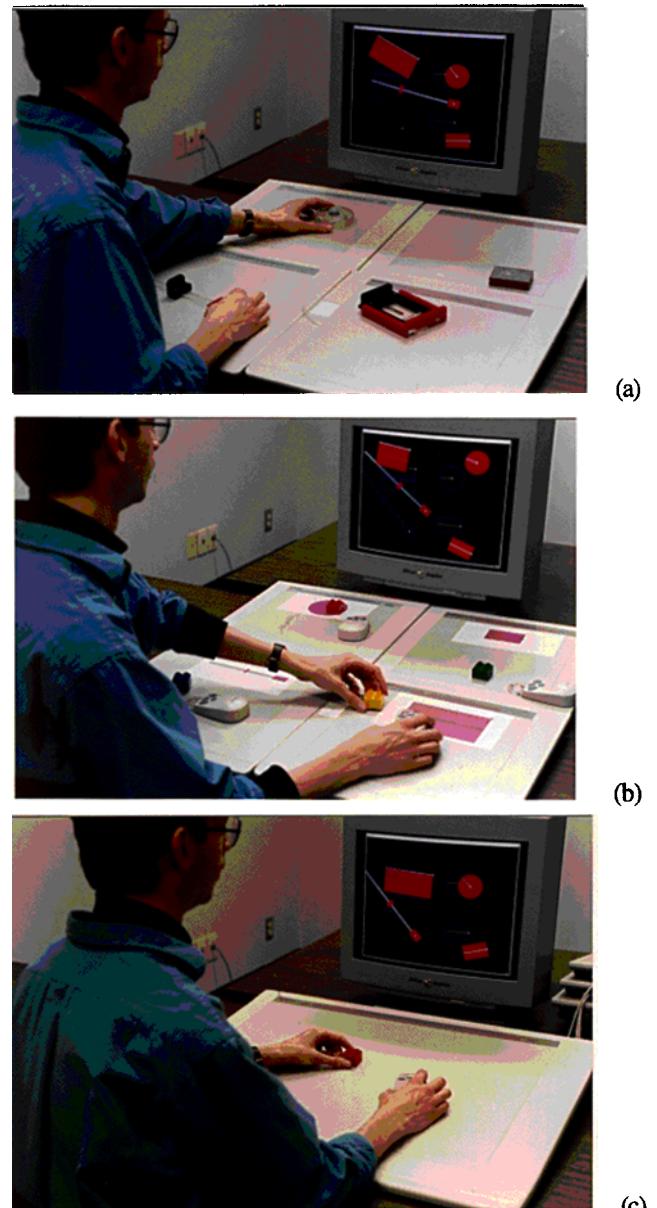
The task was performed on a Silicon Graphics Indigo2 workstation computer using four 12"x12" Wacom tablets arranged in a 2x2 grid for the space-multiplex conditions and a single 18"x25" Wacom tablet for the time-multiplex condition (see Figure 5a-c). A SpecialiX serial expander was used to attach the four Wacom tablets simultaneously to the computer and all accessed the same X11 device driver. The program was written in C using a mixed-model of OpenGL (graphics library) and X11 (for window and event-based input handling). The 2x2 grid of Wacom tablets was necessary due to the fact that the tablets can only support two sensors (puck and stylus) on them while operating in "multimode." Alternatively, we would have run all conditions of the experiment on one large Wacom tablet if it could support multiple sensors (e.g., 8 or more). Each of the tablets map onto a full screen dimension. All input devices operated in absolute mode. Thus, moving a device to the bottom left of a tablet would have the corresponding effect of moving the virtual widget to the bottom left of the computer screen.

Four specialized input devices were used in the space-multiplex, specialized devices condition consisting of the stretchable square, ruler, brick and rotor (see Figure 5a). All four devices sense both position and orientation as they have 2 sensors on the bottom side. The rotor consists of an inner core and a floating outer disk. One puck sensor is positioned in the core to provide positional information while the second stylus sensor is housed in the outer disk to orbit around the core providing rotation information. The stretchable ruler measures 11 inches long with a thin knob at one end (for the non-dominant hand) and a slider on a track that extends to the opposite end. The ruler is approximately 1.5 inches wide. The puck sensor is housed in the knob end while the stylus sensor is housed in the physical slider. The stretchable square has a more compact design in that its length dimension ranges from 4.25 inches to 8 inches. It has a constant width of 3.25 inches. The puck sensor is at the left edge while the stylus sensor is at the right edge.

Four pairs of a brick and puck were used in the space-multiplexed, generic devices condition. The puck is a standard 4 button Wacom digitizing puck while the brick is a LEGO brick measuring 1.25 inches in width and length and having a height of approximately 0.75 inches. Inside the brick was a Wacom stylus sensor which is small, wireless and batteryless providing as accurate position information as a regular stylus device. Note that both the

pucks and bricks have felt on the bottom surface for a consistent smooth feel. For this condition only, each of the four tablets were labeled using a graphic picture to indicate the virtual widget which was permanently attached to the brick and puck pairing (see Figure 5b).

The time-multiplex condition used one puck and brick device on a single 18"x25" Wacom tablet (see Figure 5c).



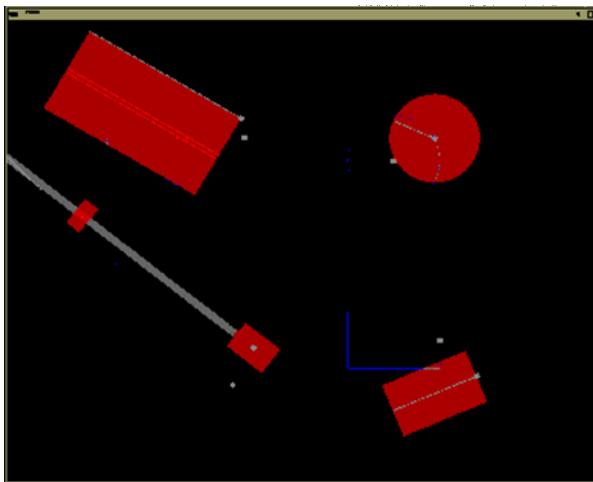
**Figure 5.** Experimental set-up: (a) space-multiplex with 4 specialized devices using 4 tablets, (b) space-multiplex with 4 puck and brick pairs of generic devices using 4 tablets and (c) time-multiplex with one puck and brick device operating on a large tablet.

### Task

Subjects used the three input conditions on a multi target tracking task. A trial consisted of a 90 second pursuit tracking session. Before the trial begins, subjects must

align their 4 widgets on top of the 4 computer targets. When the trial begins, the 4 computer targets begin to move on their pseudo-random track. Each target position is updated approximately every 1/20th of a second having a total of 1800 tracking steps. The targets can make up to 4 adjustments (x, y, rotate, scale) per update. However, to minimize a jittering effect, a direction and a minimum duration were chosen to have a target adjust along one dimension for a period of time before possibly switching to a new direction. The duration was approximately 0.5 seconds. In addition, periodically (approximately every 4 seconds), one target would "dart off" (i.e., make much larger incremental adjustments). Thus, the targets have a non-uniform adjustment. This design encourages the subject to service the dominant deviants in order to achieve the best score as opposed to sequentially or randomly servicing each widget regardless of assessing the scene.

In terms of visual representations, the computer targets were drawn in a blue outline while the user's widgets were drawn in a solid, transparent red color (see Figure 6). The transparency was used to allow for computer and user target overlaps. Transparency was achieved using alpha-blending with a value of 0.60. The shape of the targets roughly matched the shape of the specialized input devices (stretchable square, ruler, brick and rotor).



**Figure 6.** Snapshot of multi target tracking task. Computer targets are outlined in blue while the user's targets are in transparent red color.

For the space-multiplex conditions subjects could move their targets by physically acquiring the associated input device(s) and manipulating the device(s).

During the time-multiplex condition two graphical cursors are visible on the screen. The puck (used in the dominant, right hand) is represented by an "arrow" cursor while the brick is represented by a "cross" cursor. Before manipulating a user widget, the subject first must acquire the widget by moving towards the widget's selection "handle" and selecting it with the puck cursor. This is achieved by pressing and holding any one of the four puck buttons. Once pressed, the user's widget becomes attached

to the puck and automatically attached to the brick device. Subjects manipulate the widget and once the puck button is released, the widget is detached.

At the end of each trial, subjects were presented with a score of their trial. The score represents the average root-mean-square (RMS) Euclidean distance off-target for all 4 targets (along all dimensions: translation, rotation and scale).

#### *Design and procedure*

All twelve subjects used the three input conditions: space-multiplex, specialized devices (SpaceS), space-multiplex generic devices (SpaceG), and time-multiplex (Time). During each trial, four device targets were required to be tracked. Six trials lasting 90 seconds were conducted in each of the three input conditions. A total of six, 90 second, multi-target, pseudo-random tracking path stimuli were predefined. The ordering of the stimuli were randomly shuffled for each condition. Thus, all subjects experienced the same 6 track stimuli a total of three times (once per input condition). Subjects were assigned the sequence of input device conditions based on a Latin-square counterbalancing scheme to minimize ordering effects. For each new input device condition, subjects were given a maximum of one 90 second trial to acquaint themselves with the device and interaction technique. After the experiment, subjects were presented a questionnaire to obtain their subjective preferences for each condition. In summary, this experiment is a three factor  $3 \times 4 \times 6$  (MultiplexCond x Device x Trials) within subjects, repeated measures, Latin-square design.

**Experimental biasing.** In general, the technology constraint of using four tablets biases the conditions in favor of the time-multiplex conditions. With the time-multiplex condition, a stronger stimulus-response (SR) compatibility exists with the input control space and the computer display space. That is, subjects move their devices and limbs in the direction they wish to acquire or manipulate a widget. In contrast, the 2x2 grid of tablets has a stimulus-response incompatibility. First, the input devices always remained on their designated tablet. In order for subjects to manipulate a virtual, logical widget, they must remember or visually search the 2x2 grid of tablets to acquire the proper physical input device. For example, the ruler logical widget may currently be in the top right of the computer display. However, the physical ruler device is located on the bottom left tablet. We believe this mismatch places an extra cognitive burden on the subject. In addition, the space multiplex conditions were susceptible to infrequent system lags due to the multiple tablet configuration. In pilot studies, the lag was only observable in the space-multiplex, specialized device condition which generates more tablet data due to the inherent concurrency of having two sensors built into one physical device. Again, this lag phenomena was very infrequent and biases in favor of the time-multiplex control conditions. Finally note that one positive outcome of using four tablets is the reduction of device collisions. We predict that the phenomena we wish to detect is strong enough to overcome the negative biasing effects.

## RESULTS

Traditional tracking experiments define the tracking error at any moment as the distance between the center point of the user and computer targets. This is not sufficient for our tracking experiment that varies multiple dimensions and has multiple targets. An overall single measure of the tracking quality is necessary for feedback to the subject as well as for manageable data analysis [14]. Thus, we have defined a single main dependent variable of interest, the "score," to reflect the overall tracking error of the user's 4 targets from the computer's 4 targets. Specifically, the score is defined in equations 1-8 as the root-mean-square (RMS) Euclidean distance off-target for all four targets along all three dimensions: translation, rotation and scale (see equation 1).

$$\text{Score} = \sqrt{\text{RMS}_{\text{ruler}}^2 + \text{RMS}_{\text{brick}}^2 + \text{RMS}_{\text{rotor}}^2 + \text{RMS}_{\text{ssquare}}^2} \quad (1)$$

$$\text{RMS}_{\text{ruler}} = \sqrt{\text{RMS}_{\text{trans}}^2 + \text{RMS}_{\text{arcLen}}^2 + \text{RMS}_{\text{scale}}^2} \quad (2)$$

$$\text{RMS}_{\text{ssquare}} = \sqrt{\text{RMS}_{\text{trans}}^2 + \text{RMS}_{\text{arcLen}}^2 + \text{RMS}_{\text{scale}}^2} \quad (3)$$

$$\text{RMS}_{\text{rotor}} = \sqrt{\text{RMS}_{\text{trans}}^2 + \text{RMS}_{\text{arcLen}}^2} \quad (4)$$

$$\text{RMS}_{\text{brick}} = \sqrt{\text{RMS}_{\text{trans}}^2 + \text{RMS}_{\text{arcLen}}^2} \quad (5)$$

For each trial (90 seconds, 1800 tracking steps) overall tracking performance was calculated by root mean square (RMS) error for each dimension (see equations 6-8).

$$\text{RMS}_{\text{trans}} = \sqrt{\frac{\sum_{k=1}^N \text{errorTrans}_k^2}{N}} \quad (eq. 6)$$

$$\text{RMS}_{\text{arcLen}} = \sqrt{\frac{\sum_{k=1}^N \text{errorAng}_k^2}{N}} \quad (eq. 7)$$

$$\text{RMS}_{\text{scale}} = \sqrt{\frac{\sum_{k=1}^N \text{errorScale}_k^2}{N}} \quad (eq. 8)$$

At any tracking instant  $k$ , the translation tracking error  $\text{errorTrans}(k)$  is defined as the Euclidean distance between the user and computer target. The  $\text{errorAng}(k)$  is defined as the arc length ( $\text{arcLen} = \phi \cdot \text{length}$ ) between the user and computer target where  $\phi$  ranges from 0 to  $\pi$  and  $\text{length}$  is the current length of the computer target. Finally, the  $\text{errorScale}(k)$  is defined as the difference between the user and computer target lengths.

An analysis of variance (ANOVA) was conducted on the RMS score data and we now revisit the experimental hypotheses. Both of our hypotheses were supported (see Figure 7). We found that input multiplex condition has an effect on RMS score ( $F(2,22) = 103.7, p < .001$ ). Specifically, the space-multiplex specialized devices condition performs best followed by the space-multiplex generic devices followed by the time-multiplex condition. A significant difference was found when a pairwise means comparison was conducted between the conditions (SpaceS and SpaceG:  $F(1,22) = 96.9, p < .001$ ; SpaceS and Time:  $F(1,22) = 196.8, p < .001$ ; and SpaceG and Time:  $F(1,22) = 17.5, p < .001$ ).

Further analysis of the data revealed how the 90 seconds worth of trial activity varied between each of the input multiplexing conditions (see Figure 8).

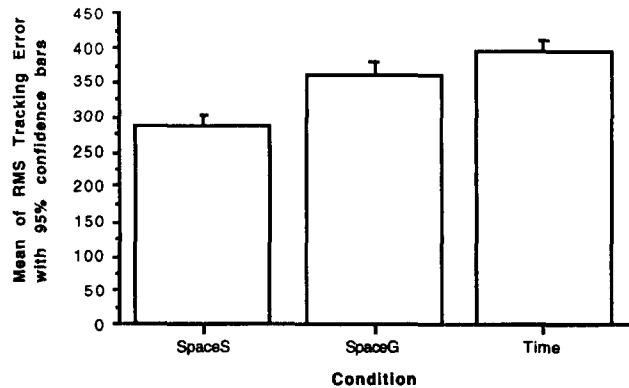


Figure 7. Mean RMS tracking error as a function of input device configuration.

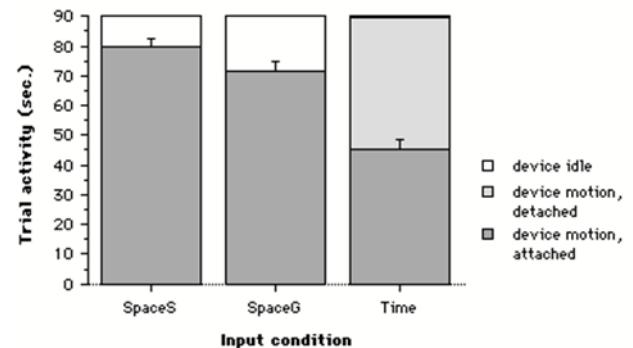


Figure 8. Trial activity breakdown between input device conditions.

With the time-multiplex condition, 45.2 seconds of the trial activity was accountable to logical widget manipulation. That is, the time when a subject has the input devices attached to a logical widget and the device is in motion (i.e., manipulating a widget). The majority of the remaining time (44.2 seconds) of the trial was dedicated to device motion without a widget attached. The bulk of this time can be considered the "switching cost" for acquiring different widgets. The remaining 0.6 seconds of the trial had no device motion. In contrast, we found that subjects in the space-multiplex, specialized device condition had 80.0 seconds of the trial accountable to device motion while the space-multiplex generic devices had only 71.6 seconds. This difference is significant (pairwise means comparison between SpaceS and SpaceG:  $F(1,22) = 22.75, p < .001$ ). We argue that the specialized physical form factors contribute to the reduced switching costs compared to the generic form factors (SpaceS vs. SpaceG). Moreover, the trial activity analysis for the time multiplex conditions shows a significant switching cost compared to both of the space-multiplex conditions.

If we examine the data by individual input device, we see a consistent trend for all four input devices across the three conditions (see Figure 9). This implies that our conclusions are generalizable. All of our specialized devices had superior performance over the generic devices in both the space and

time multiplex conditions. However, a significant interaction exists between the input devices and input condition ( $F(6,66) = 3.42$ ,  $p < .005$ ). One explanation for this difference could be that some specialized devices perform better than others compared to the generic devices. For example subjects performed slightly better with the rotor and brick devices compared to the stretchable square and ruler devices. This suggests that beyond tactile mnemonics, some devices have physical affordances that facilitate the operation of the task.

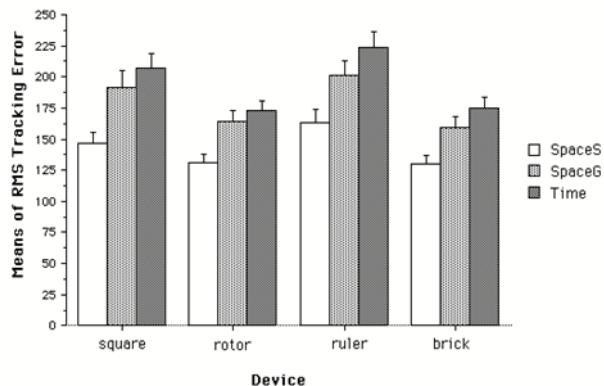


Figure 9. RMS tracking error by input device and condition

We were also interested in measuring learning effects across the six trials per input condition. A significant learning effect was found across the trials ( $F(5,22) = 4.8$ ,  $p < .001$ ). There was no significant interaction between learning and input conditions. Thus, we cannot conclude that subjects exhibited different learning rates between the space or time multiplex conditions.

After the experiment, subjects were asked to quantify their preferences for each of the input device configurations. They were asked to rate the physical comfort (i.e., how fatiguing) each condition was ranging from extreme discomfort to extreme comfort) as well as the ease at which they could solve the task (very difficult to very easy). A continuous scale from -2 to +2 was used for both ratings.

The space-multiplex with specialized devices was considered significantly more comfortable than the space-multiplex generic devices ( $t(22) = 3.42$ ,  $p < .0025$ ) or the time-multiplex conditions ( $t(22) = 4.33$ ,  $p < .0003$ ). No significant difference exists between the space-multiplex with generic devices compared to the time-multiplex condition for physical comfort.

As well for ease of use, the space-multiplex, specialized devices was viewed as significantly easier to use than both the space-multiplex, generic devices ( $t(22) = 5.65$ ,  $p < .0001$ ) and the time-multiplex condition ( $t(22) = 6.65$ ,  $p < .0001$ ). A pairwise means comparison indicates no significant differences between the space-multiplex, generic and time-multiplex conditions for ease of use.

## DISCUSSION

In general, a variety of strategies was observed throughout the experiment. The majority of the subjects used one hand to operate the specialized devices. The ruler and stretchable square were more difficult to operate than the rotor and brick. Some subjects keep their left hand on the ruler device and used their right hand to service the remaining three devices. It was not clear if this offered any improvement in performance. Nevertheless, all the subjects managed to operate the rotor and brick with one hand. Only one subject complained about grabbing the wrong input device.

In contrast, the space-multiplex, generic device conditions for the most part had subjects using two hands (one for the brick and the other for the puck) to manipulate each widget. However, at least two of the subjects used one hand to operate both the puck and brick simultaneously. We observed one subject who used one hand on the puck and drove the puck into the brick to move both of them. The graphic overlays on the tablets were designed to aid the subject in remembering what virtual widget could be controlled with a given brick and puck pair. It is not clear how frequently, if ever, the subjects used the graphic overlays. Questioning the subjects after the experiment, they claimed to make very little use of the graphic overlays. Two of the subjects reported looking down at the tablets (i.e., graphic overlays) if they were confused. Five of the subjects complained at least once during this condition of grabbing the wrong device pairings.

In the time-multiplex condition, some subjects would occasionally attempt to select a computer target instead of the corresponding user target. This cannot be easily explained except that the multi-target tracking task is difficult. Subjects must constantly assess the scene and watch the moving targets to make a decision when to stop servicing the current widget and determine which target to service next. In contrast, the space-multiplex conditions does not suffer from mistakenly selecting a computer target instead of the corresponding user target. By using the physical devices, it is only possible to select user targets.

Moreover, we believe that target acquisition is easier with physical targets than virtual targets. Physical targets can often be larger than virtual targets. Moreover, tactile feedback and mnemonics can facilitate the physical target acquisition and confirmation process. Note that for virtual target acquisition, the selection handles appear as 15 pixel wide squares on the widget. This handle size is slightly larger than what most SGI applications employ. In practice, using a larger handle size begins to compete with the application data and to clutter the scene. While we believe slight variations in target size would have a minimal overall performance effect, this was not part of the experimental design.

One could argue that Fitts law [1] could serve as a model to predict our performance results of this experiment. This, however, would be misleading. In general, Fitts law defines the time to acquire a target as a function of the distance

traveled (between the starting position and final target position) divided by the target size. While this has been shown to be true for rapid reciprocal target tapping tasks, our experimental task has a number of different features: (1) requires more high level cognitive reasoning (e.g., to assess the scene and determine which and when to switch devices), (2) consists of device acquisition for the space-multiplex conditions, as well as (3) requires not only target acquisition but a significant portion of the task deals with manipulating the device to perform a target tracking task.

While we have shown performance advantages for using multiple, specialized devices, there are some possible drawbacks for this approach including (1) cost of buying multiple devices, (2) learning the association between the physical and virtual user interface components, and (3) the overall management of multiple devices (e.g., they take up space and could get misplaced). Nevertheless, many of these issues are common and manageable in other disciplines such as tools being used for carpentry or kitchen gadgets used for cooking.

Finally note that the chosen generic experimental task of switching between controllers (physical or virtual) was designed to reflect the common behavior of switching between a set of user interface components in any application domain. Nevertheless, we have applied some of these Graspable UI concepts in the character keyframe animation domain [2] in which dedicated, space-multiplexed physical controllers are used for: (1) the selection of objects and commands, (2) three-dimensional view controls, (3) time controls and (4) character pose controls. We have received positive feedback from our initial trials among animators.

## CONCLUSIONS

Our experiment provides some initial evidence that a space-multiplex input scheme with specialized devices can outperform a time-multiplex (e.g., mouse-based) input design for certain situations. The inter-device switching cost may not be as costly as originally anticipated. That is, it may be faster to acquire an attached device that is *out of hand* than to attach to virtual controls with a device *in hand*.

We notice that today an accountant, animator and graphic designer, all use the same input device set-up (i.e., a keyboard and mouse) for performing their very diverse activities. This "universal set-up" seems inefficient for users who work in a specific domain. The mouse is a general all-purpose weak device; it can be used for many diverse tasks but may not do any one fairly well. In contrast, strong specific devices can be used which perform a task very well but are only suited for a limited task domain. The ultimate benefit may be to have a collection of strong specific devices creating a strong general system.

What these results suggest is that we may want to design systems that employ a Graspable User Interface—allowing for space-multiplexed, rapidly reconfigurable, specialized, input devices that are spatially-aware.

## ACKNOWLEDGMENTS

This work was conducted in the User Interface Research Group at AliasWavefront as well as the Input Research Group at the University of Toronto. Special thanks to Thomas Baudel for his assistance in getting 4 Wacom tablets working on one SGI computer.

## REFERENCES

1. Fitts, P. and Peterson, J. (1964). Information Capacity of Discrete Motor Responses. *Journal of Experimental Psychology*, 67, 103-112.
2. Fitzmaurice, G. W., (1996). Graspable User Interfaces, Ph.D. Thesis, Dept. of Computer Science, Univ. of Toronto.
3. Fitzmaurice, G. W. (1993). Situated information spaces and spatially aware palmtop computers. In *Communications of the ACM*. 36(7), 38-49.
4. Fitzmaurice, G. W., Ishii, H. and Buxton, W. (1995). Bricks: Laying the Foundations for Graspable User Interfaces, In *Proceedings of CHI'95*, 432-449.
5. Fukuzaki, Y. (1993). Electronic pen according to the BTRON guideline and its background. *Tronware*, vol. 4, 49-62, Japan: Personal Media publishers.
6. Guiard, Y. (1987). Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model. In *Journal of Motor Behavior*. 19(4), 486-517.
7. Hinckley, K., Pausch, R., Goble, J. C. and Kassell, N. F. (1994). Passive Real-World Interface Props for Neurosurgical Visualization. In *Proceedings of CHI'94*, 452-458.
8. Kirsh, D. (1995a). The intelligent use of space. *Journal of Artificial Intelligence*, 73(1-2), 31-68.
9. Kirsh, D. and Maglio, P. (1994). On Distinguishing Epistemic from Pragmatic Action, *Cognitive Science*, 18, 513-549.
10. MacKenzie, C. L. and Iberall, T. (1994). *The Grasping Hand*. Amsterdam: North-Holland, Elsevier Science.
11. Newman, W. and Wellner, P. (1992). A desk supporting computer-based interaction with paper documents. *Proceedings of CHI'92*, 587-592.
12. Norman, D. A. (1993). *Things that make us smart: defending human attributes in the age of the machine*. Reading, Massachusetts: Addison-Wesley.
13. Sachs, E., Roberts, A. and Stoops, D. (1990). 3-Draw: A tool for the conceptual design of three-dimensional shapes. *CHI'90 Technical Video Program, ACM SIGGRAPH Video Review*, Issue 55, No. 2.
14. Zhai, S. (1995). Human Performance in Six Degree Of Freedom Input Control, Ph.D. Thesis, Dept. of Computer Science, University of Toronto.
15. Zhang, J. and Norman, D. A. (1994). Representations in Distributed Cognitive Tasks. In *Cognitive Science*, 18(1), 87-122.