A New Approach to Haptic Augmentation of the GUI

Thomas N. Smyth
Simon Faser University
8888 University Drive
Burnaby, BC, Canada, V5A 1S6
tsmyth@cs.sfu.ca

Arthur E. Kirkpatrick
Simon Faser University
8888 University Drive
Burnaby, BC, Canada, V5A 1S6
ted@cs.sfu.ca

ABSTRACT

Most users do not experience the same level of fluency in their interactions with computers that they do with physical objects in their daily life. We believe that much of this results from the limitations of unimodal interaction. Previous efforts in the haptics literature to remedy those limitations have been creative and numerous, but have failed to produce substantial improvements in human performance. This paper presents a new approach, whereby haptic interaction techniques are designed from scratch, in explicit consideration of the strengths and weaknesses of the haptic and motor systems. We introduce a haptic alternative to the tool palette, called Pokespace, which follows this approach. Two studies (6 and 12 participants) conducted with Pokespace found no performance improvement over a traditional interface, but showed that participants learned to use the interface proficiently after about 10 minutes, and could do so without visual attention. The studies also suggested several improvements to our design.

ACM Classification H5.2 [Information interfaces and presentation]: User Interfaces. - Haptic I/O.

General Terms Design, Human Factors, Performance

Keywords

Haptic interface, haptic feedback, 3D interaction, multimodal interface, tool palette, rehearsal, visual attention

1. INTRODUCTION

Many skills in everyday life are so well practiced that they are highly efficient and require little cognitive effort on the part of the user. For example, an experienced motorist is able to adjust the radio volume, steer, and brake, all at the same time. However, many computing tasks that are eligible for this kind of fluent use do not attain it.

For example, no matter how many times a user has adjusted a slider control or selected a value with a combo box, the procedure still requires a shift of gaze and a precise mouse movement. Keyboard shortcuts and some gestural techniques provide a more fluent alternative for some tasks, but become difficult to learn as the set of potential operations grows. Furthermore, keyboard shortcuts require a conscious switch to a different technique than the one the user has already practiced.

We suggest that these limits on interaction with computers, relative to the fluency of skills in daily life, are partly a consequence of unimodal interaction. This work investigates the potential of combining visual and haptic feedback to enable more fluent and efficient interaction for a rich set of tasks, and supporting a smooth transition from novice to skilled interaction.

The common tool-based interaction paradigm, used in applications such as Adobe Photoshop or Microsoft Visio, appears to be a good candidate for such enhancement. In most tool-based applications, interaction follows a sequence: a tool (such as a text tool) is selected from a palette; its parameters (such as its font and point size) may be adjusted using a set of GUI widgets; and the tool is then used to modify the object of interest (such as adjusting text on a figure).

While this technique is proven, we feel that the precision of the movements required and the exclusive reliance on visual feedback prevent it from becoming second nature to the user. We designed Pokespace as a multimodal alternative to the tool palette, supporting both fluent expert use and smooth learnability. The Pokespace technique combines graphical display and force display, effectively 'giving users a feel' for the tools. The forces are rendered using a Sensable Phantom haptic device, which is held in the user's non-dominant hand. Users select tools and modify tool parameters using short, simple gestures, and receive rich haptic feedback in the form of constraints (walls or corners), detents (notches or grooves) and other haptic effects. The tool space is also displayed graphically, allowing novice users to smoothly transition from visual to haptic use.

This paper presents two initial studies of the Pokespace design. Among the questions we set out to answer were:

- Will users tend to ignore visual feedback once haptic feedback becomes familiar? What if we force them to rely on haptic feedback?
- Will users learn Pokespace easily? Will performance continue to improve over time?
- How will Pokespace compare to an interface composed of traditional GUI widgets?

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ICMI'06, November 2–4, 2006, Banff, Alberta, Canada.
Copyright 2006 ACM 1-59593-541-X/06/0011 ...$5.00.

- Will the multimodal feedback be comfortable or overwhelming?

We first review previous work in adding haptic feedback to everyday computing tasks. Next, we list some salient properties of the haptic system and a set guidelines for visual-haptic interface design. We then describe Pokespace in detail before proceeding to the experimental design, results, and implications of the two user studies.

2. HAPTIC INTERACTION TECHNIQUES

Previous work has typically proceeded by adding haptics to traditional WIMP interaction techniques, in particular techniques for moving a cursor to a target in 2D or 3D space. Haptic feedback was used to indicate when the cursor had reached the target, draw the cursor towards the target, or keep the cursor on the target once it had been reached.

Akamatsu, MacKenzie, and Hasbroucq [1] used a mouse that provided tactile feedback when users had entered a target. They found that users required less time to verify that they had reached the target, but that overall pointing time was not significantly improved. Engel, Goossens, and Haakma [4] reported similar findings with a modified trackball. Oakley, McGee, Brewster, and Gray [13] implemented several haptic pointing enhancements. They found that errors were reduced, but pointing time was not significantly improved by any of the enhancements.

Dennerlein, Martin, and Hasser [3] found that force display improved performance on a steering task, moving a cursor down a 'tunnel' to a target. For this task, where the path is more restricted than general pointing, force constraints improved performance times by 52%.

The above research dealt with pointing and steering tasks with only one potential target. Oakley, Adams, Brewster, and Gray [11] argued that results from studies with single targets may not generalize well to actual interfaces, which feature many possible targets. In this case, forces associated with unintended targets can disturb the movement of the user, reducing performance. Oakley, Brewster, and Gray [12] addressed this by modulating the magnitude of the haptic effects according to the speed of the user's movements to a menu item. They reported a 48% reduction in error rates with no significant difference in speed, compared to a visual-only technique.

In another example of selecting from multiple targets, Komerska and Ware [7] added haptic effects to 2D menus in a 3D virtual reality environment. The effects included constraining the cursor to the plane containing the 2D menu, constraining the cursor within the menu boundaries, and snapping the cursor to the centre of a menu item. They found that the haptic features only improved performance 0–4%. Their participants did indicate a subjective preference for the snap-to effect, though.

Other researchers have added haptic effects to more complex techniques. Miller and Zeleznik [10] added haptic effects to GUI features, such as window borders, buttons, and checkboxes. In a similar vein, Komerska and Ware [6] extended the 3D interaction techniques of their GeoZUI3D. In both projects, forces were added to either pull the user's pointer towards a target or keep the pointer on a target once acquired. Neither of these papers reports empirical evaluations of their designs.

Bernstein, Lawrence, and Pao [2] developed a bimanual interaction technique featuring haptic feedback to the dominant hand. The haptic feedback provided contact cues and snap-to-grid effects for a 3D object editor. No results for these effects were reported.

Grosjean, Burkhardt, Coquillart, and Richard [5] added tactile feedback to a technique for selecting one of 27 commands in a virtual reality environment. Users felt a vibration every time they crossed a threshold from one command to another. Unfortunately, the tactile feedback produced slower performance than the same technique with none.

Although creative and well-executed, only one of the above haptic techniques was found to increase speed of human performance. Several others reduced errors, a result of practical value, but not an indication of substantially higher fluency. Essentially, the techniques were haptic decorations of existing techniques. We suggest that high-performance haptic interaction techniques will only be achieved when new techniques are designed from scratch, in explicit consideration of the strengths and weaknesses of the haptic and motor systems.

In a previous study [14], we reported a first step in this direction — a simple bimanual command selection and location technique. We focus here on the technique developed for the non-dominant hand, which used a Sensable Phantom to select one of eight possible commands. In the haptic form of the interaction, participants could move the cursor until it hit a haptic wall. The wall functioned as a *haptic backstop*, providing a strong, clear indication that the cursor had moved far enough to activate the desired command. In the non-haptic version, participants had to rely on their proprioception to estimate when they had moved far enough. Human proprioception is inaccurate, and the version with haptic backstops supported 25% faster performance than the version relying on proprioception.

2.1 Designing for haptic capabilities

Based on the previous work, initial versions of Pokespace, and our informal observations of many haptic interaction techniques, we propose the following features of the haptic system as potentially relevant to the design of interaction techniques:

- The sense of proprioception (the awareness of the positions of the limbs in space) lacks precision and is neither well suited to guiding a limb to a precise location, nor to keeping track of a limb's precise position over time.
- It is difficult to seperate the movement of one's limbs along the dimensions of Euclidean space.
- Muscle memory, while not precisely defined, is an undeniable phenomenon. Repeated sequences of actions tend to become integrated into units, and are performed with great accuracy and efficiency.

These properties are tentative. Studies such as the ones reported in this paper will be required to verify and refine them. These properties in turn suggest an initial set of design principles for haptic interaction techniques:

- Design for rehearsal [8]: Reward the spontaneous learning that comes with practice. Do not require users to consciously choose a new strategy to improve performance (as required, for example, when switching from a menu selection to a keyboard shortcut). Instead, let them capitalize on learning gained through repeated performance of the skill.

- Use vision for controlling novel tasks, and haptics for controlling routine tasks. Maintain sufficient consistency in the environment that haptic feedback is effective and muscle memory is allowed to develop.
- Use haptic constraints, forces that guide movement along a restricted path, to compensate for the inaccuracy of proprioception. Constraints effectively increase the size of the target, reducing the accuracy required to reach it.
- Promote visual attention on the object of interest, not the interface controls.

Application of these design principles led to the Pokespace interaction technique.

3. POKESPACE

Pokespace maps a set of tools and their parameters to a three dimensional dynamic force field, displayed and manipulated with a Phantom haptic device. The Phantom is held in the non-dominant hand, and is used to select the current tool, and modify its parameters (such as brush size or font style). Meanwhile, the mouse remains in the dominant hand and specifies the point on the canvas where the current tool should be applied (such as where a line should be drawn or a flood fill should be initiated), just as it does in the familiar tool palette technique.

3.1 Basic haptic rendering

Within the space, each tool is mapped to a plane perpendicular to the z axis (parallel to the screen). A light force is rendered in the z direction, gently forcing the Phantom tip toward the nearest plane. The user can easily change from plane to plane, and thus from tool to tool, with a push strong enough to overcome this force. Additionally, stiff walls are rendered at the front and back of the set of planes.

The overall effect of these forces can be described as a series of haptic *detents* in the z direction, similar to the detents on some electronic device knobs or mouse scroll wheels, with hard stops at both ends. A tool is selected by moving the Phantom tip to the corresponding plane/detent. In our implementation, planes are spaced 10 mm apart.

Within each plane, the Phantom tip is constrained to a square region with sides of 20 mm. The magnitude of this constraint force is given by the following piecewise linear function, graphed in figure 1:

$$f(x) = \begin{cases} 0 & x < 0 \\ k_b x & 0 < x < x_b \\ k_b \frac{(x_w - x)}{x_w - x_b} & x_b < x < x_w \\ k_w(x - x_w) & x > x_w \end{cases}$$

where x is the distance of penetration of the Phantom tip beyond the boundary, x_b is the distance over which an initial semi-stiff spring force with spring constant k_b is displayed, and x_w is the distance after which a second, stiffer spring force with constant k_w is displayed. Note that between x_b and x_w, the force tapers to zero. The resulting haptic effect is that, with a light push, the Phantom tip *pops* into a channel just inside the boundary, and with a light pull, it pops out again.

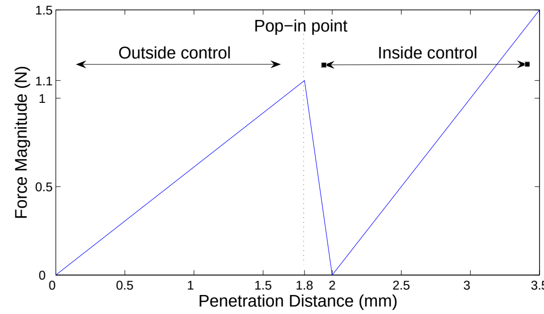

Figure 1: The magnitude of the force displayed at the constraint boundary as a function of penetration distance. In this example, $x_b = 1.8$ and $x_w = 2.0$. Pop-in occurs when the penetration distance exceeds x_b. The rightmost line rising from (2,0) theoretically extends to infinity, and represents the stiff wall encountered after popping in.

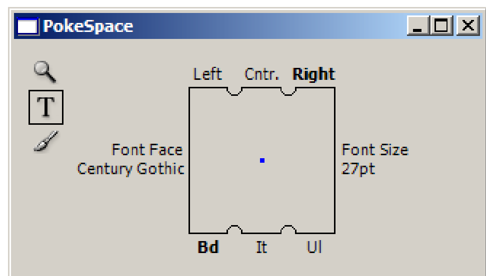

Figure 2: The Pokespace graphical display for the text tool. The small square in the center of the constraint boundary is the Phantom tip marker.

Each edge of the square is mapped to one or more parameters of the active tool. Edges mapped to more than one parameter are divided by small haptic bumps. Figure 2 shows a graphical display of the boundary for the text tool. The bottom edge of the boundary is divided into three segments, one each for bold, italic, and underline. The top edge is similarly divided into segments for left, right, and centre alignment.

A parameter is manipulated by popping into its boundary segment. For boolean parameters (such as *bold*) simply popping in toggles their value. For a parameter with a range of possible values (such as *font size*), moving the Phantom tip along its boundary segment after popping in changes its values according to a specified segment-to-value mapping.

Parameters are manipulated in this fashion so that the user can haptically explore the constraint boundary without inadvertently changing a parameter's value. Values are only changed when the user exerts enough force on the boundary to pop in.

3.2 Graphical display

The haptic rendering described above is accompanied by a graphical display similar in appearance to the mini-dialog

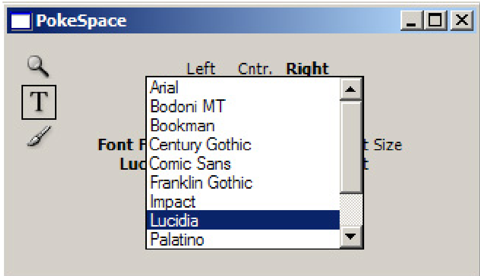

Figure 3: The Pokespace graphical display for the text tool, with the font face parameter's pop-up list box visible. The text box aids in the selection of non-boolean parameters with non-numerical value ranges or non-linear segment-to-value mappings.

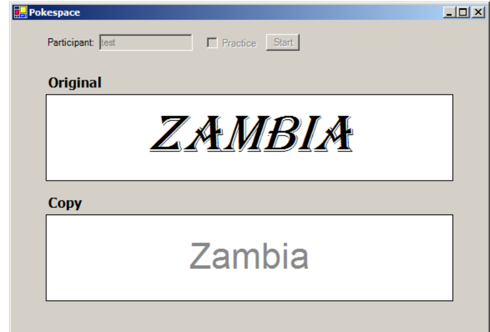

Figure 4: The window displaying the words in the text matching task. Participants had to modify the formatting of the bottom word to match the top one.

windows found in several well-known design applications such as Adobe Photoshop.

Figure 2 shows the display with the text tool active. The icons on the left indicate the currently selected tool. The main part of the window shows a visualization of the constraint boundary, as described above, for the currently active tool. Each segment of the boundary is labeled with the name and current value of the parameter it controls. The labels of boolean parameters that are set to *true* are shown in boldface.

A blue marker dot displays the position of the Phantom tip in the plane. When the Phantom tip pops into a boundary segment of a non-boolean parameter, the label for that segment changes to boldface to indicate its active state. As the Phantom tip moves along the segment, the displayed value is continuously updated, much like a conventional slider.

For parameters with a non-numeric range or a non-linear segment-to-value mapping, the consequences of moving the Phantom tip a given distance may not be clear. In such cases, the display provides a pop-up list box containing the set of possible parameter values. The list box is displayed only while the parameter's segment is active. Figure 3 shows such a pop-up list box for the text tool's font face parameter.

A light force attracting the Phantom tip to the center of the boundary square is rendered when no boundary segments are active. This force is light enough to permit free movement within the square, but strong enough to center the Phantom tip when it is allowed to drift freely. The recentering force is intended to provide an eyes-free cue as to the relative position of the Phantom tip within the boundary square, much as the raised bumps on the F and J keys of a standard keyboard help users re-orient their hands without looking.

3.3 Usage Scenarios

In order to reveal some of the features of Pokespace, consider the task of drawing a moustache on a digital photograph of a friend (or foe).

In order to create a moustache of just the right thickness and texture, the user may want to fine tune the size, hardness, and opacity parameters of the brush tool through repeated experimentation. With the conventional interaction technique, this would require many mouse movements between the canvas and the parameter controls, constantly testing one setting after another. With Pokespace, a well practiced user could start drawing a test stroke with the mouse and alternately manipulate several brush parameters with the Phantom while keeping their attention on the canvas as the parameters are changing.

In the course of drawing the moustache, the user is bound to make mistakes which must be erased. A typical eraser tool has parameters similar to those of a brush. If the eraser tool's plane were near (ideally adjacent to) the brush tool's plane, through rehearsal the user would quickly become accustomed to the movements between the two tools and into and out of their respective parameters. This cannot be said for the mouse movements between the canvas and a conventional tool palette, and between the palette and the parameter controls.

One of the most commonly used tools in many design programs is the selection tool. For example, in drawing the moustache, the user may wish to select a mask region to avoid accidentally painting the person's nose or lips with the brush tool, then select a different region to trim with the eraser, then select a third region to fill with the paint bucket. Common tools like the selection tool can be placed at the extreme front or rear walls of Pokespace, so that activating them requires only an imprecise forward or backward thrust with the Phantom tip. While the movements to each of the brush, eraser, and paint bucket tools depend upon the starting plane, the movement to the selection tool is independent of starting point, and is easily performed without looking.

4. OVERVIEW OF STUDIES 1 AND 2

We conducted two user studies to evaluate the benefits of some of the Pokespace design choices. For these first studies, we wanted to determine the learnability and experienced user performance for a single tool. We therefore focused on interaction with only the text tool. Future studies will consider multiple tools (requiring multiple planes) and tasks requiring the combined use of both hands.

In this section we describe the characteristics common to both studies.

4.1 Task

Participants were shown two copies of the same word, one underneath the other. A screenshot of the word display is shown in figure 4. The words were chosen randomly from a

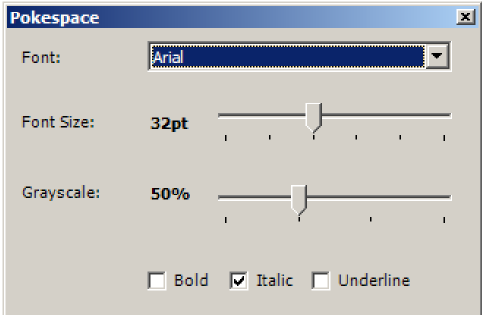

Figure 5: The traditional GUI controls used in Study 2. A combo box controls the font, sliders control the font size and grayscale, and checkboxes control the bold, italic, and underline.

list of countries. At the start of each trial, the bottom word was displayed in 50% gray 32 pt plain roman Arial font. The top word was displayed in one of four fonts (Algerian, Comic Sans MS, Stencil, and Times New Roman), chosen to be easily distinguished from each other and Arial; with size 16 pt, 24 pt, 40 pt, 48 pt, or 56 pt; in 25%, 75%, or 100% gray; and exactly one of bold, italicized, or underlined. The properties for a given trial were randomly selected from these sets. Participants were asked to modify the bottom word's properties so that it typographically matched the top word, and then to press the 'N' key on the keyboard, advancing to the next trial.

Modification of the word's properties was accomplished with two different interaction techniques. The first was a simplified version of Pokespace in which there was only one tool (a single plane). The graphical display showing the configuration of Pokespace for the studies was similar to that shown in figure 2, with the three controls on the top boundary segment replaced with a single control for grayscale. For this technique, the Phantom was held in the participant's non-dominant hand, as per the intended use of Pokespace.

The second technique consisted of a mini-dialog window containing a set of traditional GUI controls—one combo box, two sliders, and three check boxes. These controls are shown in figure 5. Participants operated the controls with their dominant hand using the mouse, while the non-dominant hand completed the trial by pressing 'N'.

The graphical display for Pokespace and the traditional control dialog window were located at the top left of the screen, while the word pair was displayed at the center of the screen. The distance from the center of the word display to the center of the Pokespace display or traditional controls was about 625 pixels or 15 cm.

4.2 Participants

Study 1 had 6 participants (5 male, 1 female), aged 23 to 27 years, with a median age of 24.5. Study 2 had 12 participants (10 male, 2 female), aged 23 to 29 years, with a median age of 24.5. No participants performed both studies. All participants were graduate students from the schools of computing science and engineering science at Simon Fraser University. No participant had more than trivial experience with haptic interfaces. Seventeen participants reported using a computer 14 or more hours per week, and one reported between 7 and 14 hours. Participants were each paid $20.

4.3 Apparatus

The studies were run on a machine with dual Intel Xeon 3.06GHz processors and 2GB of RAM, running Microsoft Windows XP 2002. The haptic device was a Phantom Premium 1.0, driven at a refresh rate of 1000 Hz. The mouse was a Microsoft IntelliMouse Optical 1.1A. Pokespace and the experimental interface were developed using Microsoft Visual C++ .NET 2003. The display was a 17" TFT LCD monitor with a resolution of 1280x1024. A Tobii 1750 eye tracker and Tobii Clearview software ran on the same machine, collecting the gaze data and recording screen contents. No other applications were running during the experiment. A stack of books was placed in front of the Phantom to provide an arm rest for participants.

5. STUDY 1

Study 1 was designed to identify and examine trends in usage of Pokespace over a moderately long period (approximately 30 minutes).

Design. Study 1 had only one condition, using Pokespace for the text matching task. Trial times and eye gaze coordinates were recorded, and users completed subjective evaluations and background questionnaires.

Hypotheses. We expected that participants would learn to rely on haptic feedback for monitoring the modification of parameters, so that eye gaze would seldom stray from the words in the centre of the screen. We also expected response time to improve dramatically during the first two blocks of trials, after which improvement would continue at a lower rate.

Protocol. Participants first read a written instruction sheet summarizing the experiment, and completed a consent form.

Next, the workspace was adjusted to the participant, and the eye tracker was calibrated (participants were thus aware that their eyes were being tracked). Participants were then introduced to the Phantom. A simple demo program which graphically and haptically renders a set of polyhedra was run, and participants were shown how to grip the Phantom stylus and poke at the objects.

Once apparently comfortable with the Phantom, verbal instructions were issued to participants describing the task and the interaction technique. Thirteen blocks of 10 trials were then completed. There were no practice trials. Short breaks were allowed between blocks.

The subjective evaluations and background questionnaires were completed at the end of the session. Sessions lasted approximately 50 minutes.

5.1 Results

The Tobii ClearView software's fixation detection algorithm was used to analyze gaze fixations with a 30 pixel fixation radius and a 100 ms minimum fixation duration. Two regions were defined—one containing the word display in the center of the screen, and one containing the Pokespace graphical display in the top left corner.

Participants usually fixated several times consecutively in one region before shifting their gaze to the other. We were primarily interested not in how many individual fixations occurred, but how often participants' gaze shifted to the Pokespace display. We therefore defined a *glance* as an incidence of one fixation on the word display immediately followed by a fixation on the Pokespace display.

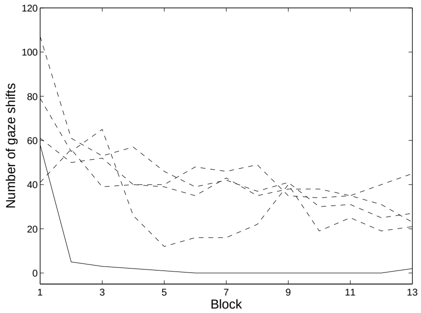

Figure 6: Glance counts per block for participants in Study 1. One participant almost never glanced after the first few blocks, while the other participants glanced frequently throughout the session.

Data for one participant was improperly recorded and had to be discarded. A graph of each of the remaining 5 participants' glance counts per block is shown in figure 6. While 1 participant made almost no glances after the second block, the other 4 participants glanced regularly throughout the session.

Of the 4 who continued to glance at Pokespace, 1 glanced markedly less. Examination of the screen recordings for that participant revealed that most of their glances occurred when the bold, italic, or underline parameters were modified.

Mean completion times per block improved sharply for the first 3 to 4 blocks, and then began to level, reaching approximately 11 s per trial by block 6.

Three issues with Pokespace's design were identified by participants in their subjective questionnaires. First, several participants stated that the hardest controls to manipulate eyes-free were the bold/italic/underline controls, and 1 of those stated that this was because they found it hard to know when the Phantom tip was in the centre of the square. This agrees with the gaze data described above. Second, several participants were distracted by the sudden change in parameter value that occurs when the Phantom tip first pops in to a boundary segment. Finally, several participants stated that they were prone to inadvertently change a parameter as they pulled the Phantom tip away from the boundary.

6. STUDY 2

In Study 2, we compared Pokespace to a variant with no graphical feedback, and to the traditional interaction technique described above.

Design. Study 2 used a single-factor within-subjects design, in which interaction technique was a factor with three conditions. The first condition used standard Pokespace with graphical display (PSG); the second used Pokespace, but with no graphical display (PSNG); and the third condition used the traditional GUI controls (TR) described above.

PSNG had no practice period; instead it was always performed directly after PSG. While this was a non-standard experimental design, we felt that it allowed the most direct measurement of the effect of removing the graphical display, and ensured participants had enough practice with the combined haptic/graphic display (30 trials) to allow the transition to pure haptic use. The order of PSG/PSNG and TR was counterbalanced.

Trial times and eye gaze coordinates were recorded, and participants completed NASA TLX workload questionnaires, open-ended subjective questionnaires, and background questionnaires.

Hypotheses. We expected participants to shift their gaze to the graphical display far more frequently for TR than for PSG. As a result, we predicted that PSG times would be faster than TR times. We also expected that after performing the technique over 30 times in the PSG condition, participants would have little difficulty when the graphical display was removed, making performance times for PSNG similar to or faster than PSG.

We expected no significant difference between subjective workloads for PSG and PSNG but, given participants' familiarity with traditional interfaces, expected that workloads for TR would be slightly lower than for PSG.

Protocol. The instructions, calibration, and workspace adjustments were the same as in Study 1.

For the PSG and TR conditions, participants first completed five minutes of practice. In this time, participants completed an average of 17 trials (range 11–25). The first few practice trials were monitored, and any confusion about the technique was clarified. No advice on strategy was given. Participants then completed 3 blocks of 10 trials. Short breaks were allowed between blocks.

Workload questionnaires were completed after each technique. The subjective and background questionnaires were completed after all techniques. Sessions lasted about an hour.

6.1 Results

ANOVA was strongly non-significant for mean block completion times for PSG ($F(2, 10) = 0.06, p = .94$) and TR ($F(2, 10) = 0.55, p = .58$), indicating that performance was similar for all blocks in those conditions.

However, for the PSNG condition, ANOVA showed a significant difference for block ($F(2, 10) = 3.79, p = .034$). A post-hoc Tukey HSD test found no difference between blocks at the .05 significance level, but at the .065 significance level found differences between block 1 and block 2 and between block 1 and block 3 . Block means over all participants were 17.3 s, 12.9 s, and 13.1 s. The greater mean for block 1 is likely due to the lack of practice time for PSNG—participants needed time to habituate to the absence of the graphical display, but their performance stabilized in blocks 2 and 3. For this reason, we used only data from blocks 2 and 3 of the PSNG condition in the analysis below.

Trial time data for one participant was improperly recorded and had to be discarded.

Quantile-normal plots of trial times revealed that the time data was non-normal, but log-transformed data was approximately normal. Thus, we use the log-transformed values for the remainder of the analysis.

Performance in TR was about 15% faster than performance in PSG ($t(10) = 2.32, p = .043$), contradicting our first hypothesis.

Performance in PSNG was significantly faster (14%) than in PSG ($t(10) = 3.56, p = .005$), confirming our second hypothesis. However, this difference must be treated with caution, as it confounds the effect of the two techniques and the effect of greater practice before PSNG (since PSG always

Table 1: Glance counts for all participants in Study 2 for the PSG condition. Rows represent blocks in descending order. The data suggest two groups of users: those who glanced frequently throughout the session, and those who learned to hardly glance at all. Participants are arranged according to group in the table.

Frequent								Seldom			
1	2	3	4	7	8	11	12	5	6	9	10
21	53	64	58	60	59	48	48	7	41	3	35
18	38	57	58	51	54	45	74	0	31	0	20
25	33	49	61	50	54	46	61	0	7	1	3

came before PSNG). Note again that the comparison does not include block 1 of PSNG, which we treated as a practice block and which had higher trial times than blocks 2 and 3.

No significant difference in subjective workload was found between either PSG and PSNG ($t(11) = 1.63, p = .13$) or PSG and TR ($t(11) = 0.16, p = .88$). This confirms our third hypothesis but contradicts our fourth.

Examination of screen recordings revealed that participants glanced at the controls frequently for the TR condition, as expected. We felt that quantitative analysis was not necessary to confirm this result. Glance counts per block for each participant for the PSG condition are given in table 1. These data confirm our observation in Study 1 that there are two distinct groups of participants: those who glance for nearly every parameter modification, and those who learn to hardly ever glance. Examination of the screen recordings of participant 1, who glanced throughout but had markedly lower glance counts than the other frequent glancers, revealed that they usually glanced only at the bold/italic/underline controls.

In subjective questionnaires, many participants identified the same three issues with Pokespace's design as in Study 1. Nonetheless, 7 out of 12 participants preferred Pokespace to the traditional technique, with some describing it as "natural", "smooth", "convenient", and "almost instinctive".

7. DISCUSSION

The results of the two studies indicate that two of our design principles will have to be modified. First, reducing the amount of visual saccades did not improve performance time. Although the PSNG condition of Study 2 demonstrated that haptics can provide sufficiently strong feedback for users to perform coarse selection tasks without visual feedback, the users performed the same task more quickly using a traditional interface. This does not mean that there is no benefit from reducing gaze switches, however. Allowing the user to maintain focus on the object of interest may be less disruptive for the higher level task, such as drawing a figure, that is their actual goal.

Second, we found that for most users, rehearsing with the combined graphic-haptic interface did not smoothly lead to exclusive use of haptic feedback. Even after using Pokespace for 130 trials, many users continued to glance at the graphical display. However, practice with Pokespace did produce an important rehearsal effect: Although disoriented when graphical feedback was removed, users had learned the location of the text controls well enough that they quickly accommodated and could locate the controls relying only on haptic cues.

As described in section 5.1, three issues with Pokespace's design were identified: trouble with bold/italic/underline controls; unexpected parameter changes on pop-in; and pull-away errors.

The first of these issues is likely due to poor proprioception, which makes it difficult to accurately keep track of the position of one's limbs in space. It was hoped that the centering force described in section 3.2 would counteract this limitation by providing a spatial cue, but it appears this was not the case. In future, we plan to experiment with increasing the magnitude of this force, and with adding textures to the square's area to act as an additional spatial cue.

We believe the problems on pop-in and pull-away are due in part to the absolute mapping of boundary segment position to parameter value. In future, we plan to experiment with a relative mapping instead. This would allow the use of more dynamic, velocity based techniques, such as those used by Oakley et al. [12], to reduce unwanted sensitivity in the interface.

8. CONCLUSION

Pokespace suggests a promising approach to using multimodal interaction techniques to increase the fluency of human computer interaction. Users learned to use it quickly and found it comfortable. After only about 30 trials, their performance was only slightly less than performance with the same task with a traditional GUI interface, and they could use it without visual attention. Their performance continued to improve as they practiced the technique for a half hour. Given that the design of Pokespace is still in its infancy, we find this encouraging.

We plan several paths for future work. First, we plan to refine the implementation of Pokespace. Work by MacLean and Enriquez [9] on *hapticons* suggests that users can readily learn mappings of vibrotactile impulses of varying frequency, amplitude, and waveform to arbitrary entities. We plan to experiment with adding hapticons to our rendering in an effort to increase discriminability of tools, parameters, and parameter values. Some other refinements have been described elsewhere in this paper.

Second, we plan to perform more extensive evaluations of Pokespace, using tasks involving multiple tools and both hands. Such evaluations will give us stronger evidence of how Pokespace might perform in actual contexts.

Lastly, we hope that work such as Pokespace inspires researchers to reconsider the familiar desktop. Although the HCI community has recently emphasized mobile and ubiquitous applications, we suggest that many interesting forms of interaction remain to be discovered for desktop computing.

9. ACKNOWLEDGMENTS

Jon Kavanagh, Dylan Tisdall, and Steven Bergner helped with pilot studies. Stella Atkins provided the Tobii eye tracker. This work was funded in part by an NSERC Postgraduate scholarship, an NSERC Discovery grant, and a CFI New Opportunities grant.

10. REFERENCES

[1] M. Akamatsu, I. S. MacKenzie, and T. Hasbrouq. A comparison of tactile, auditory, and visual feedback in a pointing task using a mouse-type device. *Ergonomics*, 38:816–827, 1995.

[2] N. L. Bernstein, D. A. Lawrence, and L. Y. Pao. Design of a uniactuated bimanual haptic interface. In *Proceedings of IEEE HAPTICS 2003*, pages 310–317, 2003.

[3] J. T. Dennerlein, D. B. Martin, and C. Hasser. Force-feedback improves performance for steering and combined steering-targeting tasks. In *Proceedings of ACM CHI 2000*, pages 423–429, 2000.

[4] F. L. Engel, P. Goossens, and R. Haakma. Improved efficiency through I-and E-feedback: A trackball with contextual force feedback. *International Journal of Human-Computer Studies*, 41(6):949–974, 1994.

[5] J. Grosjean, J. M. Burkhardt, S. Coquillart, and P. Richard. Evaluation of the Command and Control Cube. In *Proceedings of ICMI 2002*, pages 473–478, 2002.

[6] R. Komerska and C. Ware. Haptic task constraints for 3D interaction. In *Proceedings of IEEE HAPTICS 2003*, pages 270–277, 2003.

[7] R. Komerska and C. Ware. A study of haptic linear and pie menus in a 3D fish tank VR environment. In *Proceedings of IEEE HAPTICS 2004*, pages 224–231, 2004.

[8] G. Kurtenbach, T. P. Moran, and W. Buxton. Contextual animation of gestural commands. *Computer Graphics Forum*, 13(5):305–314, 1994.

[9] K. MacLean and M. Enriquez. Perceptual design of haptic icons. In *Proceedings of Eurohaptics 2003*, Dublin, Ireland, 2003. 13 pages.

[10] T. Miller and R. Zeleznik. An insidious haptic invasion: Adding force feedback to the X desktop. In *Proceedings of ACM UIST 1998*, pages 59–64, 1998.

[11] I. Oakley, A. Adams, S. A. Brewster, and P. D. Gray. Guidelines for the design of haptic widgets. In *Proceedings of British HCI 2002*. British Computer Society, 2002. 16 pages.

[12] I. Oakley, S. A. Brewster, and P. D. Gray. Solving multi-target haptic problems in menu interaction. In *Extended Abstracts of ACM CHI 2001*, pages 357–358, 2001.

[13] I. Oakley, M. McGee, S. Brewster, and P. Gray. Putting the feel in 'look and feel'. In *Proceedings of ACM CHI 2000*, pages 415–422, 2000.

[14] T. N. Smyth and A. E. Kirkpatrick. Haptic constraints for a bimanual command and location selection task. In *Proceedings of Eurohaptics 2006*, 2006. 6 pages.