

A Flick in the Right Direction: A Case Study of Gestural Input

Michael Moyle and Andy Cockburn

Human-Computer Interaction Lab
Department of Computer Science
University of Canterbury
Christchurch, New Zealand

Tel: +64 3 364 2987 x7768

Fax: +64 3 364 2569

michael.moyle@alliedtelesyn.co.nz
andy@cosc.canterbury.ac.nz

Abstract

This paper describes the design and evaluation of a gesture-based scheme for issuing the back and forward commands in web browsers. In designing our gesture recogniser we conducted several experiments to determine metrics associated with the magnitude, timing and stereotypical errors of ‘natural’ linear flick gestures using stylus and mouse input devices. These low-level metrics are important to software designers who must implement algorithms that discriminate between gestures and other actions such as mouse clicks and drags. As well as empirically characterising gestures, the metrics provide various insights into stereotypical behaviour with gestures, including the facts that angular errors are larger in the left and right directions with the pen, that vertical gestures are ‘awkward’ with the mouse, and that downwards gestures are slower than other directions. An evaluation of gestures for web browsing shortcuts shows that they enhance navigation efficiency, and that participants were extremely enthusiastic about them.

Keywords: Marking menus, gestures, web navigation, browser design, evaluation.

1 Introduction

In 2001, as part of a larger project on understanding and easing user problems in web navigation, we decided to experiment with gesture-based input for issuing the very common ‘back’ and ‘forward’ actions in web browsing. Similar gesture-based controls appeared in the commercial Opera5.11¹ browser released in April 2001—the user makes a rapid ‘flick’ dragging action leftwards to issue the ‘back’ command, and other commands are issued through similar gesture-based shortcuts. When we began implementing our gesture-recognition software, we were surprised to find little prior research characterising the physical properties of gestures. In order to distinguish between different gesture commands and between other mouse-driven actions, gesture recognition software must set constraints on the timing, direction, and magnitude of gestures. If the gesture-recognition constraints are unnatural, then users must adapt to the system’s requirement. Ideally, however, the software should be designed around the natural properties of human gesture input. By gathering data on the characteristics of natural and unconstrained gestures, we aim to provide data that answers questions such as ‘Can the left mouse button be used for gestures, given that it is already overloaded with actions for clicking on links and for selecting text?’, and many more.

This paper describes a series of experiments that investigate the natural properties of linear flick gestures. In particular, we investigate three measures: gesture magnitude (or length), the time taken to produce gestures, and the angular error or degree to which gestures deviate from the ideal direction. These values are determined across two different settings for mouse velocity/acceleration and across mouse and stylus input devices. We also briefly analyse text selection metrics to determine whether software can reliably discriminate between text selections and gestures. Finally, the paper describes an evaluation of gesture shortcuts in a realistic and practical setting: issuing the back/forward commands in web browsers. Results show that the technique is both efficient and extremely popular, with several participants stating that ‘all browsers should support this’.

The next section describes related work on gesture-based interfaces and on web navigation. We then describe our experiments to create low-level empirical characterisations of gestures and text selections. Section 4 describes our web browser implementation, which runs under any standard unaltered browser, and its evaluation. Results are then presented and discussed, followed by the conclusions.

2 BACKGROUND

This section reviews related work on gesture-based interfaces. Gesture input is used to control a wide range of user interfaces, from simple mouse-based flicks in marking menus (see below), through to rich free-form hand and body gestures used to control and interact within 3D virtual worlds—Wexelblat (1995), for instance, describes a system for gesture control of a virtual environment, and Paradiso (2003) describes recent work on using free-form gestures to interact with a large displays. This review focuses on gestures created using standard mouse and stylus input devices. The section also describes related work on web navigation, which is our target application domain for gestural input.

¹ www.opera.com.

2.1 *Gesture Systems*

Mouse-based gesture systems parse the direction or rate of mouse movement, and map the motion onto a desired user action. One of the first systems to feature mouse-gestures was Broderbund's 'Shufflepuck Café' (released in 1989), in which the user slides a puck towards an opponent with the puck's speed and direction determined by a rapid mouse-drag gesture. Kurtenbach and Buxton (1991) were among the first to investigate mouse gestures for issuing everyday commands in user interfaces through their 'marking menus', which were a specialisation of pie-menus (Callahan, Hopkins, Weiser and Shneiderman 1988). Pie-menus minimise item selection time by arranging menu items around a circle centred on the user's cursor. The Fitts' Law (1954) time-to-target requirements are minimised because a movement of one-pixel is sufficient to reach any of the menu items, and further movements result in the target effectively becoming larger. Marking menus enhance the efficiency of pie-menus by allowing the user to select items simply by flicking the mouse in the appropriate direction—the user need not wait for the menu to be displayed. If the user hesitates during the gesture (a delay of more than approximately half a second) then the pie-menu is displayed to assist learning the gesture set. Kurtenbach and Buxton (1994) found that marking menus were heavily used once users learned the gesture direction. Further evaluation showed that performance with marking menus deteriorates as the number of items in the menu increases (Kurtenbach, Sellen and Buxton 1993).

Early mouse-based marking menus systems used the left mouse button to issue gesture commands. The left button, however, is often used for other tasks such as text selection. For this reason, some systems (such as Opera, see section 2.2) use the right mouse button to reduce the problems of overloading the interface semantics associated with each button, yet users are unfamiliar with dragging actions using the right button. The evaluations reported in this paper include an examination of the differences between gestures that are created using the left and right mouse-buttons.

In selecting one item from a single marking menu, the recognition software need only compare the total distance travelled on the X and Y coordinates to determine the direction of the gesture. By extending the marking menu concept to non-linear gestures, users can access much larger command sets, but the recognition software needs greater sophistication to distinguish between differently shaped gestures. The Unistrokes gesture alphabet, for example, allows users to express all letters in the Roman alphabet with gestures (Goldberg and Richardson 1993). Several other character sets have been implemented using similar gesture techniques, for example, T-CUBE (Venolia and Neilberg 1994) and Graffiti (www.palm.com). Beyond text input, non-linear gesture input has been used for a wide range of application areas including air traffic control (Chatty and Lecoanet 1996). The GRANDMA toolkit allows gesture recognition to be added to interfaces by having the system developer provide examples of gestures and their associated interface actions (Rubine 1995).

Despite the wide range of gesture systems developed, there has been relatively little work on characterising the physical properties of gestures used by recognition software. Dulberg, Amant and Zettlemoyer (1999) compared simple linear flick gestures with normal button clicks and keyboard shortcuts. In tasks that involved flicking towards abstract targets, they showed gestures to be 26% faster than button selection, but not reliably faster than key-bindings. Users also found the gestures easy to learn and accurate with only 4 errors from 3300 trials. In their six-subject informal study of flick gestures for window-management tasks (redirecting keyboard focus to items on the Microsoft Windows desktop) subjects reported no problems with learning the technique, and five of the six participants said they would use it if available.

Goldberg and Richardson (1993) measured the median times that a single subject took to input various characters using Unistrokes. They found the dot gesture used to specify a space was the fastest at 90 milliseconds, and that the ‘ α ’ shaped gesture used for ‘q’ was the slowest at approximately 330 milliseconds. Overall, the median stroke time was approximately 150 milliseconds. These values result from observations of only a single user. It remains unclear how these values will differ between users.

2.2 *Web Navigation*

Several prior studies have shown that navigating back to previously visited pages is an extremely common activity in web use. In a study of three-weeks of client-side web logs, Catledge & Pitkow (1995) found that, on average, each user visited 58% of URLs more than once. In a more recent four-month study, Cockburn & McKenzie (2001) found a revisitation rate of 81%: four out of five page visits are to pages previously seen by the user.

Web browsing applications support many mechanisms for revisiting web pages, including ‘favorites’ (or bookmarks), history tools, and the back button. The back button is a dominant source of page requests, with Catledge & Pitkow (1995) and Tauscher & Greenberg (1997) reporting that it accounts for 41% and 30% of requests respectively. The forward button, in contrast, was lightly used, accounting for only 2%.

In recognising the importance of ‘back’, all commercial browsers provide several shortcuts for issuing the command. Microsoft Internet Explorer supports two keyboard accelerators for back: Backspace and ‘Alt+left-arrow’. Netscape Navigator also uses the ‘Alt+left-arrow’ key binding. The main limitation of key-bindings for web browsing is that the mouse is the main mechanism for accessing the links on the page (Tauscher and Greenberg (1997) showed that link selection accounts for approximately 50% of user actions), and keyboard shortcuts therefore incur an overhead in homing the hands between the mouse and keyboard. The context menu, accessed through the right mouse-button, provides another shortcut for ‘back’, but only if the menu is not posted over an image on the page. Accessing back through the context menu also incurs overheads in waiting for the menu to be posted and in the Fitts’ Law limitations of pointing to the menu item. Through our experience, studies, and subjects’ comments, it seems that most users seldom issue the back command using any technique other than by directly clicking on the button.

Gesture controls for web navigation appeared in commercial browsers at approximately the same time as we started working with them. Opera 5.11², released in April 2001, provides facilities similar to those evaluated in this paper. The Mozilla Optimoz project³ released gesture navigation late in 2001. Sensiva⁴ is a commercial front-end to the Windows, Mac and Linux operating systems that provides around a dozen gesture-based shortcuts for commonly issued commands, including cut, copy, paste, and back. To our knowledge none of these systems have been formally evaluated, and details of the metrics used in their gesture-recognition engines are unavailable.

² www.opera.com.

³ www.optimoz.org.

⁴ www.sensiva.com.

3 EMPIRICAL CHARACTERISATION OF NATURAL FLICK GESTURES

In most graphical user interfaces the left mouse button is overloaded with several actions associated with different types of manipulation. The traditional tasks assigned to the mouse buttons can be classified into one of two groups, depending on the physical actions they require:

1. *Point, click and release* (PCR). These actions are characterised by moving the pointer to an item on the screen, then rapidly pressing and releasing the mouse button, with little or no mouse movement between the press and release. The left mouse button is heavily used in this group for actions such as selecting icons and targets such as web links in a web browser. The right mouse button is normally used to pop-up a context menu. Double clicking is a special case of this group of actions.
2. *Point, click, drag and release* (PCDR). These actions are characterised by moving the pointer to an item, pressing and holding the mouse button down, then moving (or ‘dragging’) the cursor to a new location prior to releasing the mouse. Normally the location of the release action carries interface semantics (such as the new location for an object, or identifying the desired item in a menu). The left mouse-button is heavily used in this mode, for actions such as dragging icons, menu selections, and text selections (for cut, copy and paste actions). The right button is seldom used for PCDR actions, with the result that many users report a sense of awkwardness when dragging with the right button held down (shortcut creation in Microsoft Windows is a relatively rare exception).

The physical characteristics of a flick gesture are similar to the PCDR group, requiring a mouse down action with a quick movement in a certain direction, followed by a mouse release. This poses a problem when implementing gesture recognition systems, due to the potential conflict between the gesture commands and other actions from the PCDR group. The major difference between the flick and the PCDR group is the accuracy of the drag movement. Traditional dragging tasks require a level of precision to select a portion of text, select a menu item, or place an icon in the desired position. In comparison, the flick requires a relatively imprecise movement—the user simply flicks the pointer in the general direction of the target. It therefore seems reasonable to suspect that the time taken to issue a command in the PCDR group will be longer than that required to issue a flick. If the time distributions do not overlap, a gesture recognition system could distinguish between PCDR tasks and flick gestures by examining the time taken to complete the task. Actions lasting longer than a specified threshold would be classified as dragging actions, and those lasting less than the threshold would be classified as flick gestures. If, however, the distributions overlap significantly, time could not be used to distinguish between each action. In this case an alternative to mouse button overloading would be necessary, such as using different mouse buttons.

This section describes an experiment to determine empirical characterizations of gesture actions. These characterizations will answer questions such as the PCDR/flick ambiguity, and many others, to aid the implementation of gesture recognition systems.

To maximize the generality and utility of our results, we studied several conditions beyond the minimum requirements for our needs in implementing gestures for web navigations. As well as comparing the physical properties of gestures created using the left mouse button with the PCDR actions of text selection, we also compare metrics for gestures created under the following conditions:

- *Different directions.* We were interested to see whether gestures in different directions display stereotypical differences to each other. To reduce the number of tasks participants would have to perform, we limited the analysis to four simple directions: left, right, up and down.
- *Left and right mouse buttons.* Systems such as Opera by-pass the potential problem of overloading gestures on the left mouse button by using the right mouse button instead. Determining metrics using both mouse buttons will aid the generality of our results.
- *With and without mouse acceleration.* Mouse motion is normally controlled by one or two user-configurable ‘control-display gain’ (MacKenzie and Riddersma 1994) parameters that determine the mapping between movement of the physical mouse and the cursor. These values are normally termed ‘acceleration’ and ‘threshold’. When the mouse moves slowly, a base mapping between physical mouse-motion and cursor movement applies. Normally the default value for base movement is approximately one to four, meaning that the cursor moves four centimetres for each centimetre of physical mouse motion. The acceleration setting determines the maximum mapping between mouse movement and screen distance. This mapping applies during rapid mouse movement and is normally around double the base value, meaning that during rapid motion the cursor moves $\approx 8\text{cm}$ for each centimetre of mouse motion. The threshold value determines the mouse-movement rate (distance per unit time) that must be reached before the accelerated mouse mapping applies.

We measure gesture metrics with both accelerated and unaccelerated mouse settings. Although accelerated values are indicative of real system use (mouse acceleration is the default behaviour in operating systems), they are strongly susceptible to differences in control-display gain settings. With unaccelerated gestures we translate screen pixel distances (which vary across hardware platforms) into physical measurements at the mouse.

- *Mouse and pen input devices.* Finally, to aid transfer of our results to mobile computing technology, we also inspect gesture metrics as generated using mouse and stylus input devices.

3.1 Method

The twenty-nine participants were all right-handed post-graduate Computer Science students. Although these participants have substantially more computing experience than most, we believe that their general motor skills in generating natural flicks will be similar to other subject groups. None of the participants had prior experience with gesture-based interfaces.

The participants were assigned to one of three gesture-input conditions:

1. *Mouse input, no acceleration.* All gestures were created with mouse acceleration turned off (constant control-display gain), providing a constant linear mapping between physical movement of the mouse and corresponding cursor movement.
2. *Pen input.* Gestures were created using a pressure sensitive tablet placed flat on a desktop. There was a one-to-one mapping between physical movement of the pen on the screen surface and the resultant gesture size.
3. *Accelerated mouse input.* The gestures were created using a common default setting for mouse motion. This X-Windows default is termed ‘two-to-one, threshold four’, and corresponds to an approximately four-to one mapping

between physical and cursor motion during slow mouse movement, and an eight-to-one mapping during rapid movement.

The experimental procedure was the same in each of the three conditions. The participants were informed that we were interested in the natural properties of flick gestures with the mouse (or with the pen). Mouse users were told that ‘flicks are quick motions with the mouse, during which the mouse button is pressed and released’. Pen users were told that ‘flicks are a quick motion with the pen’. Pen gestures did not require the stylus button to be pressed.

The participants were asked to practice issuing several gestures in each direction prior to beginning the experiment. Through the experiment, they were asked to generate either twenty-five or fifty gestures in each of four directions: up, right, down, and left. Participants assigned to one of the two mouse conditions were asked to repeat the set of gestures using the right mouse button (the first set was issued using the left button). Participants were told which direction to gesture and which button to use by the experimenter prior to each block of gestures.

The twenty-five or fifty repetitions in four directions gave us either one- or two-hundred gestures for each pen user, and either two- or four-hundred gestures for each mouse user (half each with the left and right buttons). We changed from fifty to twenty-five gestures approximately half way through the study because some of the participants mentioned that they started to feel repetitive strain problems with ‘stiffness’ and ‘cramping’ in their hands and wrists. The reduction in number of gestures has a minimal impact on our data analysis, because only one mean sample time in each direction is calculated per user.

3.2 Apparatus

All of the gestures were issued using a Logitech mechanical mouse to drive the cursor within in a large window (500x400 pixels) created by a Tcl/Tk program. The window was completely blank except for a counter at the bottom which showed the number of gestures generated. The program was equipped to log coordinates and time when the gesture began and finished. For mouse actions, the gesture began when the left/right mouse button was pressed, and finished when the button was released. For pen gestures, the gesture began when the mouse touched the screen surface and finished when it left the surface.

The mouse experiments were run on a 32x24cm display, with a 1280x1024-pixel resolution, giving 40 horizontal pixels per cm. The pen experiments ran on a stylus computer with a 19x14.25cm display, running at a 640x480-pixel resolution, giving 33.7 horizontal pixels per cm. The pen computer only allowed time to be measured to the nearest 55 milliseconds. For this reason, time information for the pen gestures was discarded. For input using the stylus or non-accelerating mouse conditions, the pixels per centimetre measure allows us to translate the logged pixel values into millimetre measurements of physical device movement.

3.3 Data Analysis

Three dependent variables were measured, as follows:

- *Gesture magnitude*—the distance between the x/y cursor coordinates when the gesture begins and when it finishes. The data values were logged in pixel coordinates, but were also translated into the corresponding millimetre motion values at the physical device.

- *Gesture timing*—the time, measured in milliseconds, between the start and finish of each gesture. Timing values were not measured for the pen condition in analysis one because of the low timing resolution supported by the tablet computer.
- *Angular error*—the per-gesture offset between the intended gesture direction (up, right, down or left) and the actual direction. Measuring angular error allows us to detect stereotypical biases towards particular angular errors for each gesture direction for each input device. Dulberg et al. (1999) reported an overall median error of 5.73 degrees for gestures, but they did not report the difference between errors in specific directions.

Data from the experiment were analysed in three ways, as described in Sections 4.1, 4.2, and 4.3.

3.4 *Gesture Metrics — Results*

All participants completed the full set of gestures extremely rapidly, with all training and two hundred gestures typically taking less than five minutes to complete. Considering that the participants were given minimal training and instruction, there was surprisingly little variation in the magnitude, timing, and angular error of gestures produced by each participant and between participants. Across the total 5200-gesture set, the mean gesture size (distance from mouse-button/pen down position to mouse-button/pen up position) was 6.6mm (standard deviation $\sigma=3.2$), the mean gesture time was 151 milliseconds (σ 53), and the mean angular error was 4.2 degrees (σ 5.3).

As mentioned earlier, the participants began to suffer repetitive strain symptoms very rapidly. We doubt that commercial use of gesture commands will cause similar problems because the experiment required the subjects to generate artificially large numbers of gestures in an extremely short period of time. It is unlikely that commercial use would require equivalently dense patterns of use.

3.4.1 *Analysis one: pen versus non-accelerating mouse directional comparison:*

The first analysis compares gesture magnitude and angular error in a mixed 2x4 analysis of variance (ANOVA) for factors ‘input device’ and ‘gesture direction’. Input device is a between-subjects factor with two levels: pen (stylus) and non-accelerating mouse. ‘Gesture direction’ is a within-subjects factor with four levels: up, right, down, and left. This analysis illuminates differences between pen and mouse gestures, and between gestures in different directions. If reliable differences exist between input devices or between directions, then gesture recognition software should account for them. Seven participants were randomly assigned to the mouse condition, eight to the pen.

Pen gestures were substantially larger than mouse gestures, with the mean magnitudes of physical movement for the pen and mouse of 18.9mm (σ 9.5) and 7.0mm (σ 2.2): $F_{1,13}=10.84$, $p < .01$. The gesture magnitudes in the four directions (left, right, up and down) were not reliably different: $F_{3,39}=1.73$, $p=.18$. There was no interaction between input device and gesture direction ($F_{3,39}=1.47$, $p=.24$). The difference between the magnitude of pen and mouse gestures is unsurprising. Pen gestures begin as soon as the pen makes contact with the pen-computer display: the gesture size is directly equivalent to that created with a pencil on paper. With the mouse, however, the gesture size is the subset of the physical mouse movement that occurs between the button being depressed and released—the users began sweeping the mouse prior to depressing the button, and finished the sweep after releasing the button.

[Figure 1 around here.]

Analysing angular errors reveals an interesting contrast between gestures created with the pen and the mouse. Figures 1a and 1b show the magnitude and direction of gestures created in the four directions using the pen (left) and the mouse (right). The ‘brush-like’ effect when using the pen (figure 1a), particularly for the rightwards gesture, indicates a higher degree of angular error. Analysis of variance showed no significant difference between the mean angular errors using the pen (6.5° , σ 6.7) and the mouse (3.6° , σ 1.4): $F_{1,13}=2.38$, $p=.15$. It also showed no significant difference between the angular errors in the four gesture directions: $F_{3,39}=1.64$, $p=.2$. There is, however, a significant and surprising interaction between the input device and direction: $F_{3,39}=3.15$, $p<.05$. Figure 2 reveals the cause of the interaction. It shows the mean angular errors for the pen and mouse gestures across the four directions, and reveals that angular errors are relatively large in the left and right directions when using the pen but small when using the mouse. Conversely, the angular error for the down gesture is relatively large when using the mouse but small when using the pen.

[Figure 2 around here.]

The directional biases of angular errors are shown as percentage values on either side of each direction in figure 1. These values show that when using the pen to make horizontal (left or right) gestures, there is a strong tendency to err upwards on right gestures (83%), and downwards on left gestures (71%). The bias works in the same direction for mouse gestures, but the effect is much less pronounced, with 64% of right gestures erring up, and 51% of left gestures erring down. There were no clear biases in angular errors for the up and down gestures for either the pen or the mouse. The physical mechanisms used to create gestures partially explain these stereotypical error biases, as discussed in section 3.4.4.

All of the participants were right-handed. We suspect that the direction of the angular bias partially depends on the orientation of the user’s body to the device, and would be reversed for left-handers.

3.4.2 *Analysis two: left versus right mouse button flicks with a non-accelerating mouse*

The second analysis examines the three dependent variables—magnitude, time, and angular error—across gestures created with the left and right mouse-buttons. A non-accelerating mouse was used. The analysis design is a repeated-measures 2x4 ANOVA, with factors for mouse button used (left or right) and gesture direction. This comparison is relevant because user interfaces seldom require right-button dragging, yet some commercial gesture systems (such as Opera) use a right button gesture. We were interested to see whether there are performance and preference differences between the two mouse buttons. The same seven participants assigned to the mouse condition in analysis one created assigned to both the left and right mouse button conditions.

The mean displacements in the left, right, up and down gestures were 7.15 (σ 2.2), 7.25 (σ 2.37), 6.68 (σ 2.34) and 6.76 (σ 2.62) millimetres respectively (the millimetre displacement values are one sixteenth of the pixel values, equating to 114, 116, 107 and 108 pixels in each direction). Analysis of variance showed no significant difference between gesture distances in the four directions: $F_{3,18}=0.49$, $p=0.7$.

The timing and angular error dependent measures also showed no significant differences between the means for the two buttons. The overall mean angular error was 3.5° (σ 1.7), and the mean time was 169ms. Furthermore, the means and standard deviations for all dependent measures indicated that use of different mouse buttons has a negligible impact on the

performance of gesture input. The participants' comments, however, revealed a marked preference for creating gestures with the left button (discussed below).

3.4.3 *Analysis three: impact of mouse acceleration*

The third analysis compares gestures created using a non-accelerating mouse with those created using a standard acceleration setting. The experimental design is a 2x4 mixed ANOVA for factors 'mouse acceleration' (between-subjects, with levels non-accelerating and accelerating) and 'gesture direction' (within-subjects, left, right, up, down). The metrics derived will help characterise the values that a gesture recognition system can expect to receive on user input of linear gestures. Data from the seven participants used for the non-accelerating mouse in analysis one was re-used, but a further fourteen participants generated gestures in the accelerated mouse condition.

Surprisingly, there was no significant difference between the mean magnitude of gestures created in the non-accelerating (111, σ 34.9 pixels) and accelerating (127, σ 41 pixels) conditions: $F_{1,19}=0.8$, $p=.37$. The similarity in magnitude of non-accelerating and accelerating gestures is almost certainly due to their short duration. Mouse motion causes accelerated movement only when the threshold movement per unit time is exceeded. When the threshold time is not substantially larger than the gesture time, the accelerated mapping applies for only a small subset of the gesture.

ANOVA revealed an unexpected significant difference between the mean times taken to issue gestures in different directions: $F_{3,57}=5.9$, $p<0.01$. The mean gesture times in the left, right, up and down directions were 160 (σ 39), 161 (σ 43), 159 (40) and 173 (σ 42) milliseconds. The down gesture therefore took approximately 8% longer than the others to issue, discussed further below. In post-hoc comparison, the Tukey Honest Significant Difference is 10.1, confirming the difference between downward gestures and other directions. Finally, there was no interaction between factors acceleration and direction: $F_{3,57}=0.25$, $p=0.86$.

3.4.4 *Observations and comments*

Many participants commented that gestures felt 'awkward' and tiring with the mouse. Equivalent statements were not made about the pen interface. Several participants using the mouse commented that vertical (up and down) gestures were 'no where near as natural' as horizontal (left, right) ones. In comparing left and right gestures, those that expressed a preference preferred the leftward direction. Although mouse-users found horizontal movement preferable, pen participants generally found up and down gestures to be more natural than left and right. The preferences for different directions when using the pen and mouse can be attributed to the physical characteristics of gesture generation. When using the mouse to make left and right gestures, the hand was moved by laterally flexing the wrist with almost no finger movement. In contrast, when making left and right gestures with the pen, lateral and rotational wrist movement was combined with small amounts of finger extension (left) or contraction (right) was used. The participants used two different methods to create vertical gestures with the mouse. The less commonly used technique was to move the whole arm with minimal movement in the fingers or wrist. More often, however, the participants kept their hand and wrist still, and moved the mouse by extending (up) or contracting (down) their thumb and fourth/little fingers. With the pen, vertical gestures were made by extending and contracting the fingers and thumb.

There was a strong preference for creating gestures using the left mouse button rather than the right one. All but one of the participants used their middle finger to press the button when generating flicks with the right mouse button. The remaining participant used his third-finger. The left finger was always used on the left mouse button. The participants' lack of experience with dragging using the right mouse button is the most likely explanation for the left-button preference.

3.5 *Metrics for PCDR Text Selection*

Having established empirical characterizations of natural gestures, we need to characterize PCDR text selections in order to determine whether they populate a different temporal distribution from gestures.

In previous work Card, English and Burr (1978) measured expert performance of text selection using four different devices: a mouse, a joystick, step keys (or arrow keys), and text keys (such as 'home' and 'end'). Their participants' tasks included the time to home the hands to the device starting from the space bar, and the time required to position the cursor at the appropriate position in the text. The mouse was found to be the most rapid and accurate input device, but their results did not isolate measurements for the PCDR portion of the selection action. In another related study, Gillan, Holden, Adam, Rudisill and Magee (1990) examined the relationship between Fitts' Law and elementary dragging tasks such as text selection. Their experiment showed that text selections are quickest when the dragging distance is small; their minimum value was 0.25cm, or a single character. The mean time for such a selection was approximately 600ms (extracted from figure 3 in their paper). Given that their focus was on determining whether Fitts' Law predicted dragging performance, it is unsurprising that their study did not report minimum times taken for PCDR actions.

3.5.1 *Method*

Seven right-handed post-graduate Computer Science students participated in the study. Due to their substantial computer experience the evaluation results should err towards the lower-bound timing of expert performance.

[Figure 3 around here.]

All text selections were made within a window of text with line numbers to the left, as shown in figure 3. The text content of the window was the same throughout the evaluation, and was displayed in a monospaced font. Each participant selected nineteen different pieces of text, ranging from a single character to an entire paragraph, with the data from the first five selections discarded as training tasks. Each task was verbally presented to the participant: for example, "select the phrase 'As part of the' at the end of line 7." Participants were asked to point with their finger at the text to ensure that they understood the target. Software logged all user actions, and if the selection was incorrect, the data was discarded and the participant repeated the selection until successful. Errors were not measured as the primary purpose of the experiment was to determine representative minimum text selection times.

The experiment was run on a 32x24cm display at a 1280x1024 pixel resolution. Mouse control-display gain was set to the X-Windows default value (two-to-one, threshold four). A Tcl/Tk program was used to log all timing, mouse movement, and button-press data.

[Table 1 around here.]

Table 1 lists the fourteen selection tasks, and figure 3 shows the location of the items in the interface. The selection tasks were grouped into related sets of tasks, such as selecting word groups at the start of the line. Many of the selection tasks were chosen to represent rapid actions, following Gillan et al.'s observation that the horizontal size of the drag is not necessarily the main determining factor in selection time, but that the precision of the start/stop regions must also be considered. It should be noted that some text environments allow selections through multiple clicks. This type of selection task was not evaluated as the focus of the evaluation was to measure mouse dragging movements rather than stationary clicks.

3.5.2 Text Selection: Results and Comparison with Gestures

The overall mean for the text selection tasks was 1054ms (σ 354.8), with a range from 294ms to 2682ms. Table 1 shows the results for each of the six different task groups. As predicted, single letter selections were fastest (mean 581ms, σ 319ms, range 294–1785ms), and the 'middle of words' group were slowest due to the dual precision required at the start and end of the selection. Although many of the single letter tasks required precise targeting, the short distance helped to offset the cost of targeting. The direction of text selection was heavily influenced by the direction subjects read the text.

Combining the results of the small text selection experiment with the larger gesture analysis experiment, it is clear that a gesture recognition system can disambiguate between rapid flick gestures and dragging actions for text selection. Over 99.7% of flick gestures were completed within 310ms, yet even the most rapid text selection (selecting a single character, which is an unlikely action) has a mean completion time of 581ms, with only 1% completed in less than 310ms. This indicates that PCDR gesture tasks inhabit a distinct temporal population from PCDR selection tasks, as indicated in figure 1a.

4 GESTURE NAVIGATION—A PRACTICAL EVALUATION OF GESTURE SHORTCUTS

Our analysis of gestures and selections, reported above, was motivated by our desire to construct and evaluate gesture-based shortcuts for web navigation. This section reports on the implementation and evaluation of gesture navigation. This evaluation was first reported in Moyle and Cockburn (2003).

To support gesture navigation, we constructed a web-site where each page contained a Javascript program that interacts with any unaltered browser. The program records information on the movement of the mouse whenever the mouse button is pressed, and when it detects a valid gesture command it instructs the browser to execute the appropriate action (navigating back or forward). To issue a back or forward command the user flicks the mouse (with the left button held down) left for back or right for forward. The requirements for a valid gesture are as follows.

Gesture size. To prevent the script from recognising a simple mouse click (such as a link selection) as a gesture command, the mouse coordinates must change by at least 35 pixels between the button being pressed and released. This value was selected to include 95% of the natural gestures created in the previous study, while minimising the likelihood of misinterpretation with a link-selection (PCR) action in which the user accidentally displaces the cursor.

Gesture duration. To prevent the script from accidentally recognising text-selection PCDR actions as gesture commands, gestures had to be completed within 250ms. This value was selected to include 99% of gesture actions, while excluding all of the selections observed in our text selection experiment (section 3.5.2).

Gesture direction. To detect the back or forward gesture, the absolute mouse coordinate change between the button being pressed and released must be greater on the X-axis than on the Y-axis, allowing for a 45% angular error. This extremely large sloppiness factor is possible because our gesture recogniser need only discriminate between two gestures. A richer gesture language could safely use a much smaller error level, according to our analysis of angular errors.

Page links. Gestures are not recognised when the mouse button is pressed while the cursor is over a page link. This property is undesirable, but was beyond our control within our implementation due to the way in which browsers intercept low-level interface events. We explained this feature to subjects as an unfixed usability bug.

4.1 *Evaluation*

Twenty participants, all volunteer postgraduate Computer Science students, took part in the experiment, which lasted approximately twenty minutes. The participants completed two web-browsing tasks that were based on common types of backtracking behaviour in web-use: depth-first search and back, and breadth-first ‘hub-and-spoke’ (Catledge and Pitkow 1995) browsing. All trials in both tasks involved selecting a total of four page links and issuing the equivalent of four independent ‘back’ commands. The experimental procedure was consistent across all trials in both tasks. First, the participants were shown the precise navigational path to follow, and they were required to rehearse it at least twice (and more times if they wished). They were then asked to follow the path *as quickly as possible* using the normal back button. Having completed all of the trials in tasks one and two, they were given one minute of training on issuing gestures, and they then repeated exactly the same paths for the two tasks using the gesture interface. We intended that the rehearsal of the navigation tasks prior to timed performance would minimise the impact of learning effects between the normal back button and gesture interfaces. Essentially, we wished to measure expert performance of routine tasks.

The experiment was conducted using Internet Explorer version 5.50 running on Windows 2000. The browser window was sized at 1152x864 pixels on a 17inch-monitor display running at a resolution of 1280x1024 pixels. The implementation of the gesture system was transparent to users, with many commenting on the seamless integration with the browser. The Javascript program was automatically added to every page served by an IIS 5.0 web server. All pages were held in the browser’s cache prior to the experiment, ensuring that download speeds did not affect the results. Task completion times were measured using a stopwatch.

The web pages used in the evaluation were selected from the University of Canterbury web-site. All pages had a consistent format, with a banner across the top, and an index menu on the left (see figure 4). This format is familiar to the participants who were all students at the university.

[Figure 4 around here.]

4.1.1 *Task one*

Task one examined the effectiveness of the two interfaces in depth-first navigation. Figure 5a depicts the path, which involved following four links on subsequent pages then backtracking with four successive back commands. This task represents a directed search style of web-use: for instance, searching for a faculty member’s web page starting from their university’s home-page. The data were analysed using a paired T-Test to compare task performance using the back button and the gesture system.

[Figure 5 around here.]

There are two reasons for hypothesising that the gesture system might not provide significant performance benefits in this task. First, the user need only make one movement to the back button, minimising the time-to-target overheads predicted by Fitts' Law. Having moved the cursor to the back button, the task can be completed by clicking four times with no additional cursor motion. Second, once the pointer is over the back button, the users can short-cut back to the top-level page using the back menu. Using the back menu adds one more cursor-positioning task in pointing to and selecting the desired menu item. It is unclear that the back-menu technique will be more efficient than issuing four discrete clicks of the back button without the overhead of cursor movement.

4.1.2 Task two

Task two examined the effectiveness of the two interfaces in breadth-first navigation, also called 'hub-and-spoke' navigation (Catledge and Pitkow 1995). Hub-and-spoke navigation involves visiting a series of links (or 'spokes'), one at a time, off a central 'hub' page: for example, visiting the pages for several members of faculty, one at a time, by selecting a series of links on the 'Faculty' page.

Figure 5b depicts the navigation path used. It shows that, beginning at a 'start' page, the user follows a link to a main 'hub' page, then navigates to three 'spoke' links off that page, pressing 'back' to return to the hub each time. Finally, the user issues a back command to return to the start page.

When using the back button, the mouse-pointing requirements of this task are much higher than task one. After selecting each link to a spoke page the user must point to the back button, and then point to the next page link. Fitts' Law predicts that the back button technique will result in slower task performance than the gesture technique. For this reason, a second factor for 'amplitude' was introduced to allow us to analyse the degradation of performance as the distance between the back button and the links on the hub-page increased. The links on the hub-page were vertically aligned immediately above one another at one of three corners of the web page: top-left, bottom-left, and bottom-right of the page (see figure 4). The location of the links gives three levels of amplitude 'low', 'medium' and 'high' with mean distances between the back button and the group of links of 8.5cm, 14cm, and 20cm.

The performance data in this task were analysed using a 2x2 analysis of variance with repeated measures for factors 'interface type' with two levels (back button and gesture system) and 'amplitude' with three levels (low, medium and high).

4.1.3 Subjective measures

Seven questions were presented to the participants during the tasks. We wished to measure their satisfaction with the back button (both before and after using the gesture system) and with the efficiency and learnability of the gesture system. All questions were answered on a five-point Likert-scale from one (disagree) to five (agree). The questions are summarised in table 2.

[Table 2 around here.]

Question 1, 'The back button is an effective means of navigation', was presented at the start of the experiment, before any tasks or training had been conducted. Question 2, 'The back button allowed me to quickly navigate the pages', was presented

after completing both tasks using the back button but before using the gesture system. Participants responded to Question 3, 'The gesture system will allow me to navigate faster', after their first practice session with the gesture system. Questions 4, 'The gesture system did allow me to navigate faster' and 5/6, 'The (back button/gesture system) is an effective means of page navigation' were presented after all tasks were complete. Question 7, 'The gesture system was easy to learn', completed the evaluation.

4.2 Results

As explained in the preceding section, to reduce the learning effect in solving the tasks using the normal back button first, it was necessary for the participants to perform their tasks in an expert manner, without instruction from the experimenter, and without hesitation in accessing each page. From our observations we believe that the repeated rehearsal of the navigation paths were successful in inducing expert performance. All but one of the participants completed all of the tasks without hesitation, and mean task completion times were low (the one non-expert user's data was discarded). Across both interfaces and both task sets, the mean task completion time was 6.63 (σ 0.97) seconds. Considering that both tasks involved displaying a total of nine pages this task completion time may seem unrealistically low. Prior research, however, indicates that web browsing is a surprisingly rapid activity (Cockburn and McKenzie 2001), with a high percentage of page visits lasting less than one second.

Although most participants had little to no difficulty in learning to make valid gestures, one participant persisted in making slow and deliberate gestures, even when the importance of rapid flicks was stressed. His mean time for gesture tasks was three and a half standard deviations from the mean for the remaining nineteen subjects. All his data is removed from the evaluation. Almost all other participants were extremely enthusiastic about the gesture system, and many expressed their desire that all browsers should support the technique.

4.2.1 Task one

Task one involved depth-first navigation through a series of links, followed by backtracking to the starting page. The mean task completion times for the back button and gesture systems were 6.1 (σ 1.1) seconds and 5.4 (σ 0.96) seconds, showing a reduction of 11% in the mean task time when using the gesture system. This is a significant difference: two tailed T-Test, $t_{18}=2.68$, $p<0.05$.

The result (in favour of the gesture system) is surprising given that relatively little mouse-motion is necessary for the task, and that the back-menu can be used as a shortcut (one menu selection rather than four clicks of the button). Six of the participants did use the back-menu as a shortcut for backtracking to the start page. Four of these were still faster using the gesture system. One participant incorrectly identified the title of the start page in the back menu, and had to issue one further back command.

Two participants used the backspace key to issue the back command when using the normal browser. Both solved the task extremely rapidly (4.9 and 4.7 seconds). Interestingly, one of these participants was faster still when using gestures (3.1 seconds).

4.2.2 Task two

The second task compared the efficiency of the two interfaces for breadth-first ‘hub-and-spoke’ browsing. Additionally, it examined the relative performance of the two interfaces as the distance (amplitude) between the back button and the links on the page increased. The mean time for the back button was 7.44 (σ 1.12) seconds compared to a mean of 6.09 (σ 1.19) seconds for the gesture system, giving a significant main effect: $F_{1,18}=82.2$, $p<.001$. The gesture system reduced the mean task time by 18%. The mean times for the three levels of amplitude were not significantly different ($F_{2,36}=1.17$, $p=0.32$), with all means approximately 6.7 seconds. Considering that amplitude has no effect on task completion time in the gesture system, it is unsurprising that the means are not reliably different.

As expected, the interaction between factors ‘interface type’ and ‘amplitude’ is significant: $F_{2,36}=6.78$, $p<0.01$. Figure 6 shows the cause of the interaction. As the amplitude increases, the mean completion times for the back button increase, but the task times for the gesture system remain relatively constant.

[Figure 6 around here.]

4.2.3 Comments and Subjective Measures

The participants’ comments go far beyond the quantitative results in demonstrating the effectiveness of the gesture system. Half of the participants were extremely enthusiastic about the system, making comments such as ‘Fantastic’, ‘That’s amazing’, ‘Really really nice’ and ‘Brilliant!’

There were, however, some negative comments that frequently occurred. Six participants complained that the implementation did not treat gestures as first class actions. This problem arose from two specific types of user action: first, when starting a gesture over a link on the page; and second, when starting a gesture over non-link text on the page. As mentioned in the implementation details section, gestures are not recognised when initiated over a link. Consequently, the participants had to put some care into the location of their gestures over the browser window. This placement care is inconsistent with the informal and ‘sloppy’ flick gesture. Participants felt that the gesture should work at any location over the page. If the gesture scheme was implemented in a commercial browser, it would be relatively straightforward to overcome this problem.

The second situation causing problems with gestures—initiating gestures over text—did not cause gesture recognition to fail, but did cause momentary concern for a few participants. Normally dragging the mouse over text causes text selection, but in our implementation, mouse motion over text is overloaded. This overloading meant that when the user initiated a gesture over text, the text was highlighted as normal. The brief ‘flash’ of selected text prior to navigating back (or forward) was disconcerting to some of the participants who complained of ‘having to find a blank bit of the page’. Overcoming this problem in a commercial implementation of the gesture scheme requires thought. One option is to overload the right mouse button for gestures as well as the context menu. Disambiguating gestures from menu requests could be based on coordinate change per unit time. We could not use the right-button technique in our Javascript implementation because Internet Explorer always pops up the menu on a right-button release, regardless of the coordinate change between button press and release. Another problem with the right-button approach is the strong subjective preference against dragging with the right-button in our gesture characterisation experiment (section 3.4.4).

The participants' responses to the Likert-scale questions showed several interesting effects. Mean responses to the questions are summarised in table 2. Before beginning either experiment the participants were asked to rate the effectiveness of the back button for navigation (Q1). The mean response was 2.95 (σ 1.13), yet after using the gesture system the participants were again asked the same question (Q5) and their ratings dropped to a mean of 2.42 (σ 1.12). Eighteen of the twenty participants rated the efficiency of the gesture system more highly than the back button (questions 5 and 6), with the remaining two giving the same rating for both interfaces. The mean responses for the effectiveness of the back button and gesture system were 2.42 (σ 1.12) and 4.26 (σ 0.81), giving a reliable difference: Wilcoxon Signed Ranks test, $N=17$, $z=3.46$, $p<0.01$. Finally, participants rated the gesture system highly for ease of learning (Q7): mean 4.26 (σ 0.93).

4.2.4 Discussion

To summarise the results, gesture shortcuts significantly reduced the mean task completion times in both tasks. In the depth-first browsing task, the gesture system reduced mean task times by approximately 11%, and in the hub-and-spoke browsing task the reduction was approximately 18%. Considering that backtracking activity is completed hundreds of millions of times every day, it appears that gesture navigation has the potential to enhance the overall efficiency of web navigation. The potential of gesture navigation is further demonstrated by the minimal training that our participants required and by their positive comments and subjective measures.

There are, however, several experimental concerns that might have affected the results. Overall, we do not believe that removing or controlling any of these issues would alter the primary result.

- *Sub-optimal implementation.* The Javascript implementation of the gesture system was limited in that participants had to ensure that their flicks were not initiated over a link, and some participants were disturbed by brief flash of text-selection that occurred while generating gestures. These issues are likely to have adversely affected performance with the gesture system. As discussed in the Results section, overcoming this limitation should be relatively straightforward in a commercial system.
- *Experimental tasks.* The tasks were based on expert navigation through rehearsed page sequences. This was necessary to remove highly variable performance factors such as searching for links, navigating to incorrect links, and so on. Prior studies have shown that users' patterns of web use are highly repetitive and that a high percentage of pages are visited for surprisingly short periods of time (Catledge and Pitkow 1995; Tauscher and Greenberg 1997; Cockburn and McKenzie 2001).
- *Browser preference.* All participants used Microsoft Internet Explorer for the experiment, although approximately half of the participants normally used Netscape Navigator. This is unlikely to have affected the results because the back button's location and behaviour is similar in both interfaces.
- *Participant pool.* All of the participants were post-graduate Computer Science students. Although this subject pool has more experience with mouse-use than most, we do not believe that their motor skills in generating the flick gesture were significantly different to other user groups.
- *Training.* Although most participants learnt the marking menu concept with ease, two had problems during initial training. Rather than flicking to make a gesture, they used slow and precise movements that took more than 250ms

to complete. Once the importance of making rapid gestures was stressed, one participant dramatically improved, but another persisted with slow precise gestures (and his data was discarded). In a commercial deployment of gesture navigation there may be problems with users learning the new features because they have no visual representation in the interface and because direct training is impractical. It seems likely however that awareness of the gesture features would propagate through informal discussions and observations of colleagues' web use.

- *Measurement tool.* All tasks were timed using a stopwatch rather than software logging. There are obvious inaccuracies associated with stopwatch use. Initial trials of the experiment used server logs to measure navigation times, but these provided page access times at one-second granularity, which is a coarser measure than can be reliably attained with a stopwatch.

5 Conclusions

Gesture-based shortcuts, issued by rapid flick drags of the mouse, are becoming increasingly common in user interfaces. Despite this increased deployment, the fundamental metrics of gestures are unavailable in prior research, meaning that developers must generate their own data on which to base their gesture recognition algorithms, or they must take a best-guess.

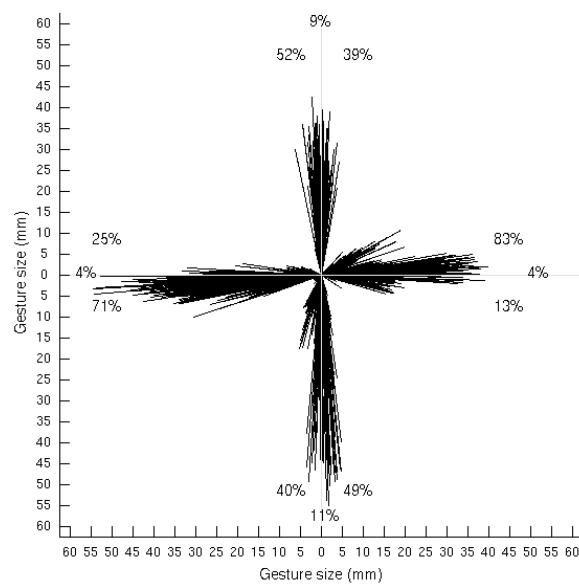
This paper described a set of experiments to empirically characterise mouse and stylus generated linear flick gestures. In addition, we analysed the metrics associated with the very common text-selection activity, which displays similar characteristics to mouse gestures. Finally, we used the empirical characterisations to implement gesture-based shortcuts for the back/forward actions in web browsing.

Results of the empirical characterisations revealed many interesting metrics and stereotypical properties of gesture input that should be of use to future implementers of gesture recognition systems. These results include the facts that properties of text-selections and gestures are temporally distinct (allowing the same controls to be overloaded with semantics for both actions without risk of misinterpretation), that down gestures are relatively slow with the mouse, and that horizontal pen gestures display stereotypical angular errors. Results of the evaluation of gesture shortcuts for back/forward in web-navigation show that gestures can enhance efficiency, that they are easily learned, and that people are extremely enthusiastic about them.

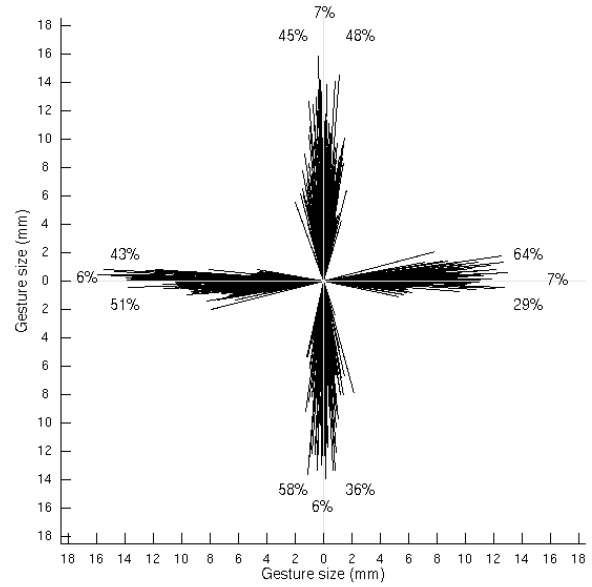
In further work we will examine performance and learnability issues in accessing a wide range of controls through mouse-based gesture interfaces.

6 REFERENCES

- Callahan, J., D. Hopkins, M. Weiser and B. Shneiderman (1988). An Empirical comparison of Pie Versus Linear Menus. Proceedings of CHI'88 Conference on Human Factors in Computing Systems. 95-100.
- Card, S., W. English and B. Burr (1978). "Evaluation of Mouse, Rate-Controlled Isometric Joystick, Step Keys, and Text Keys for Text Selection on a CRT." *Ergonomics* **21**(8): 601-613.
- Catledge, L. and J. Pitkow (1995). Characterizing Browsing Strategies in the World Wide Web. Computer Systems and ISDN Systems: Proceedings of the Third International World Wide Web Conference, 10--14 April, Darmstadt, Germany. 1065-1073.
- Chatty, S. and P. Lecoanet (1996). Pen Computing for Air Traffic Control. Proceedings of CHI'96 Conference on Human Factors in Computing Systems, Vancouver, April 13--18. 87-94.
- Cockburn, A. and B. McKenzie (2001). "What Do Web Users Do? An Empirical Analysis of Web Use." *International Journal of Human-Computer Studies* **54**(6): 903-922.
- Dulberg, M., R. Amant and L. Zettlemoyer (1999). An Imprecise Mouse Gesture for the Fast Activation of Controls. Proceedings of INTERACT'99. 375-382.
- Fitts, P. (1954). "The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement." *Journal of Experimental Psychology* **47**: 381-391.
- Gillan, D., K. Holden, S. Adam, M. Rudisill and L. Magee (1990). How Does Fitts' Law Fit Pointing and Dragging? Proceedings of CHI'90 Conference on Human Factors in Computing Systems, Seattle, April. 183-191.
- Goldberg, D. and C. Richardson (1993). Touch-Typing With a Stylus. Proceedings of INTERCHI'93 Conference on Human Factors in Computing Systems, Amsterdam, April 24--29. 80-87.
- Kurtenbach, G. and W. Buxton (1991). Issues of Combining Marking and Direct Manipulation Techniques. Proceedings of ACM UIST'91. 137-144.
- Kurtenbach, G. and W. Buxton (1994). User Learning and Performance with Marking Menus. Proceedings of CHI'94 Conference on Human Factors in Computing Systems, Boston, April 24--28. 258-264.
- Kurtenbach, G., A. Sellen and W. Buxton (1993). "An Empirical Evaluation of Some Articulatory and Cognitive Aspects of Marking Menus." *Human-Computer Interaction* **8**(1): 1-23.
- MacKenzie, I. and S. Riddersma (1994). "Effects of Output Display and Control-Display Gain on Human Performance in Interactive Systems." *Behaviour and Information Technology* **13**(5): 328-337.
- Moyle, M. and A. Cockburn (2003). The Design and Evaluation of a Flick Gesture for 'Back' and 'Forward' in Web Browsers. Proceedings of the Fourth Australasian User Interface Conference (AUIC2003). Adelaide, Australia, 4-7 February: 39-46.
- Paradiso, J. (2003). "Tracking Contact and Free Gestures Across Large Interactive Surfaces." *Communications of the ACM* **46**(7): 62-68.
- Rubine, D. (1995). Specifying Gestures by Example. Proceedings of ACM Conference on Computer Graphics. 329--337.
- Tauscher, L. and S. Greenberg (1997). "How People Revisit Web Pages: Empirical Findings and Implications for the Design of History Systems." *International Journal of Human Computer Studies*, Special Issue on World Wide Web Usability **47**(1): 97-138.
- Venolia, D. and F. Neilberg (1994). T-Cube: A Fast, Self-Disclosing Pen-Based Alphabet. Proceedings of CHI'94 Conference on Human Factors in Computing Systems, Boston, April 24--28. 265-270.
- Wexelblat, A. (1995). "An Approach to Natural Gesture in Virtual Environments." *ACM Transactions on Computer Human Interaction* **2**(3): 170-200.



(a) Pen gestures.



b) Mouse gestures.

Figure 1: Magnitude and angular errors of gestures (left, up, right, down) using the pen and mouse. Percentage values show the proportion of flicks that deviate to either side of the desired direction.

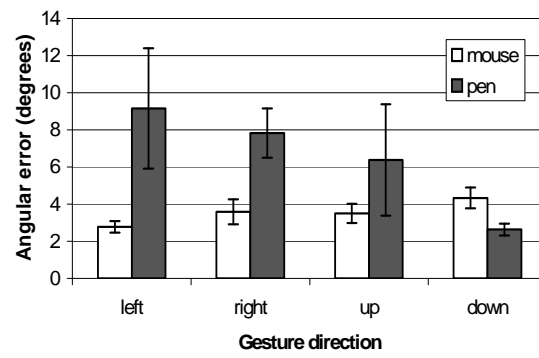


Figure 2: Mean angular errors in each of the four directions using the pen and mouse devices. Error bars show one standard error above and below the mean.

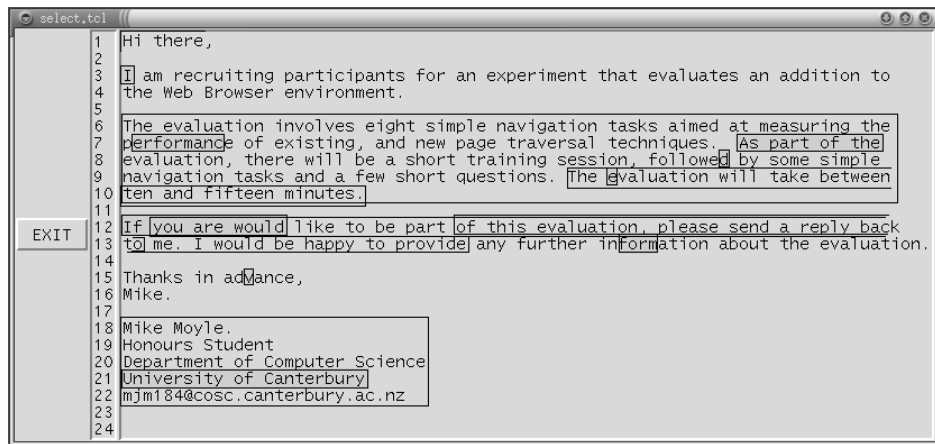


Figure 3: Interface used for the text selection timing tasks.

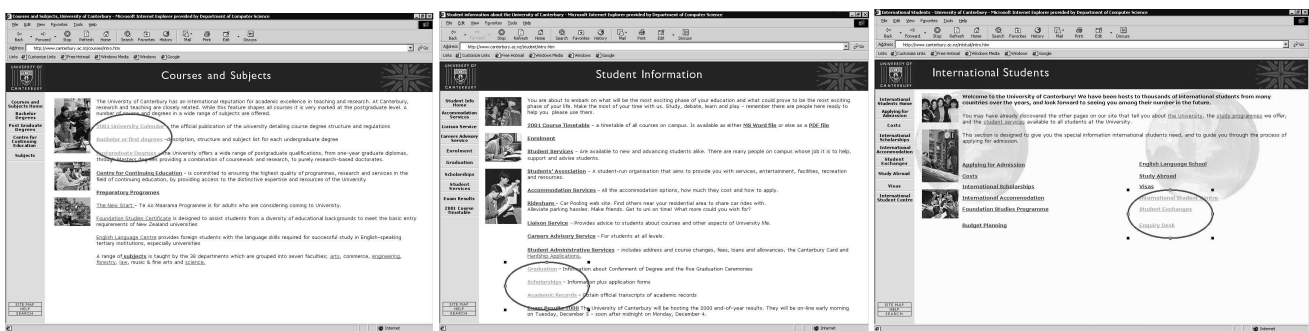


Figure 4: The three ‘hub’ pages used in task two. From left to right are the low, medium and high levels of amplitude, with the spoke links in the top-left, bottom-left and bottom-right of the page. The location of the spoke links is shown by

Start page
→follow link
→follow link
→follow link
→follow link
←back
←back
←back
Start page

a) Depth-first navigation path used in task one.

Start page
→follow link to hub page
→follow link to spoke page
←back to hub page
→follow link to spoke page
←back to hub page
→follow link to spoke page
←back to hub page
←back
Start page

b) Breadth-first 'hub-and-spoke' navigation (task 2).

Figure 5: Navigational paths used in tasks one and two.

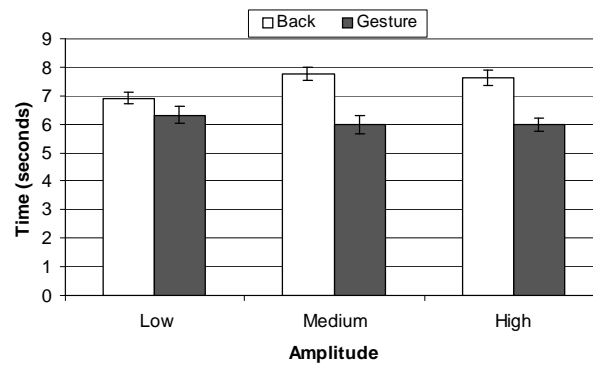


Figure 6: Mean task completion times in task two for the normal and gesture interfaces across the three levels of amplitude. Error bars show one standard error above and below the mean.

Table 1: Tasks and results for the text selection tasks. All times in milliseconds.

Group	Task	Description	Mean	St. dev	Min	Max
Single Letters	1	'v' in "advance" on line 15.	581	319	294	1785
	2	'd' in "followed" on line 8.				
	3	'e' in "evaluation" on line 9.				
	4	'I' on line 3 without the space.				
Multi-line word groups	5	The last sentence of second paragraph, starting on line 9.	1217	595	539	2501
	6	From "you" on line 12 to first "to" on 13.				
	7	From "of" on line 12 to "provide" on 13.				
Paragraphs	8	The signature paragraph from line 18 to 22.	1283	576	628	2456
	9	The second paragraph that starts on line 6.				
Middle of word	10	"form" from "information" on line 13.	1295	469	718	2024
	11	The word "performance" less the first 'p' and last 'e' on line 7.				
Word Group – Left	12	"If you are would" on line 13.	1291	682	473	2682
	13	"University of Canterbury" on line 21.				
Word Group – Right	14	"As part of the" on line 7.	1200	527	545	1899
Overall			1055	570	294	2682

Table 2: Subjective satisfaction questions, and mean and standard deviation of responses (5-point Likert scale questions).

Question	Mean	SD
Q1. The back button is an effective means of page navigation.	3.0	1.12
Q2. The back button allowed me to quickly navigate pages.	3.3	1.17
Q3. The gesture system will allow me to navigate faster.	4.05	0.83
Q4. The gesture system did allow me to navigate faster.	4.55	0.83
Q5. The back button is an effective means of page navigation.	2.5	1.15
Q6. The gesture system is an effective means of page navigation.	4.25	0.79
Q7. The gesture system was easy to learn.	4.25	0.91