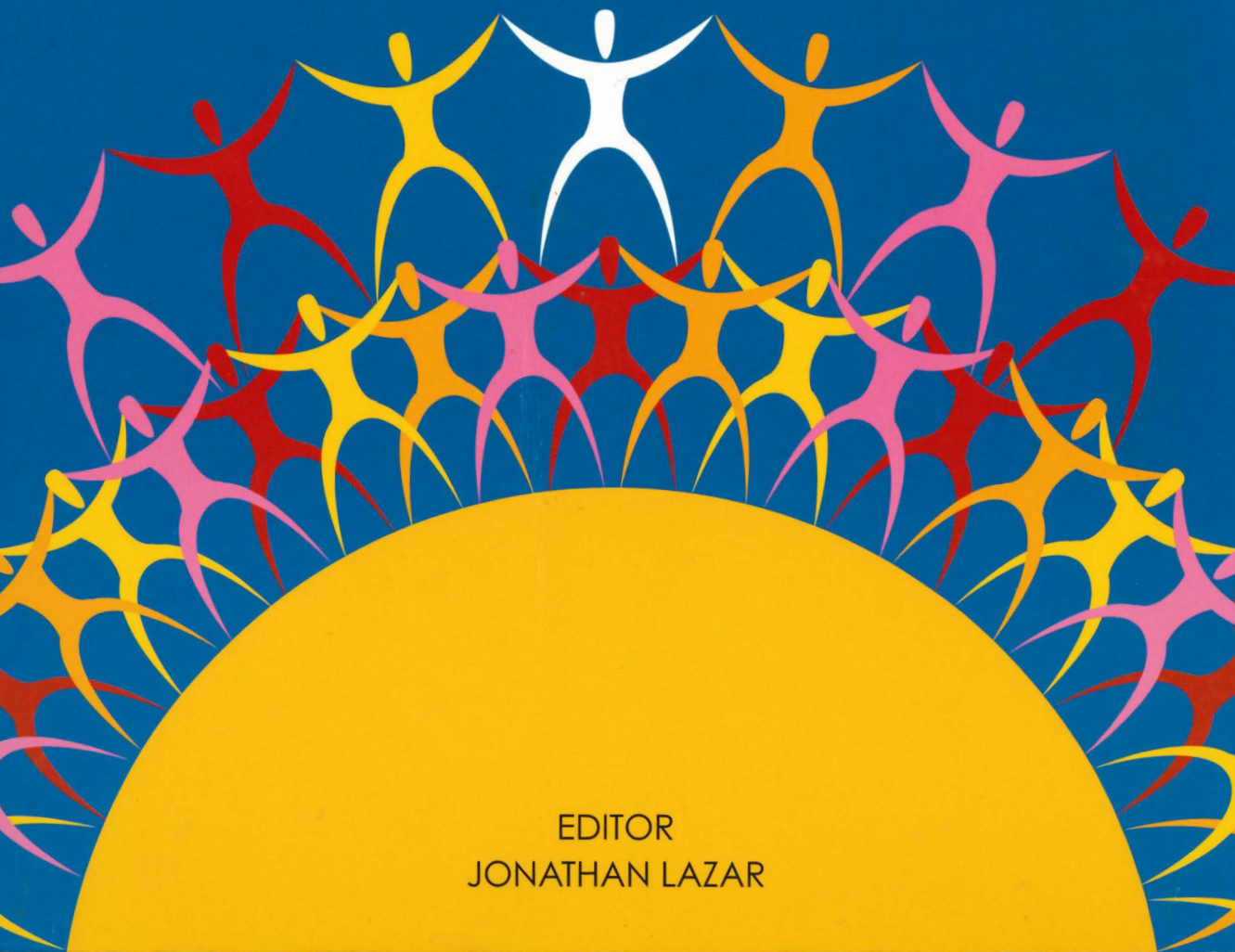
UNIVERSAL USABILITY

Designing Computer Interfaces for Diverse Users



EDITOR
JONATHAN LAZAR

Universal Usability

Designing Computer Interfaces for Diverse User Populations

Edited by Jonathan Lazar

John Wiley & Sons, Ltd

John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester,
West Sussex PO19 8SQ, England

Telephone (+44) 1243 779777

Email (for orders and customer service enquiries): cs-books@wiley.co.uk
Visit our Home Page on www.wiley.com

Other Wiley Editorial Offices

John Wiley & Sons Inc., 111 River Street, Hoboken, NJ 07030, USA

Jossey-Bass, 989 Market Street, San Francisco, CA 94103-1741, USA

Wiley-VCH Verlag GmbH, Boschstr. 12, D-69469 Weinheim, Germany

John Wiley & Sons Australia Ltd, 42 McDougall Street, Milton, Queensland 4064, Australia

John Wiley & Sons (Asia) Pte Ltd, 2 Clementi Loop #02-01, Jin Xing Distripark, Singapore 129809

John Wiley & Sons Canada Ltd, 6045 Freemont Blvd, Mississauga, ONT, L5R 4J3

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

Anniversary Logo Design: Richard J. Pacifico

British Library Cataloguing in Publication Data

A catalogue record for this book is available from the British Library

ISBN 978-0-470-02727-1

Typeset in 10/13 pt Sabon by Thomson Digital
Printed and bound in Great Britain by Bell & Bain, Glasgow
This book is printed on acid-free paper responsibly manufactured from sustainable forestry in which at least two trees are planted for each one used for paper production.

Adding Gestural Text Entry to Input Devices for People with Motor Impairments

14

Jacob O. Wobbrock[a, b] and Brad A. Myers[b]

[a] The Information School, Box 352840, University of Washington, Seattle, WA 98195, USA
[b] Human-Computer Interaction Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA

Parts of this chapter appeared in Wobbrock, J.O., Aung, H.H., Myers, B.A., and LoPresti, E.F. (2005) Integrated text entry from power wheelchairs. Journal of Behavior and Information Technology 24(3): 187–203, published by Taylor & Francis. Other parts are extensions to the work in Wobbrock, J.O. and Myers, B.A. (2006) Trackball text entry for people with motor impairments. Proceedings of ACM CHI 2006, pp. 479–488, published by ACM Press.

14.1 Introduction

People with motor impairments, such as those caused by muscular dystrophy, cerebral palsy, Parkinson's disease, or spinal cord injuries, often cannot use a conventional mouse and keyboard. They may lack sufficient mobility to reach for these devices, sufficient dexterity to operate them or switch between them, or sufficient endurance to use them for more than a few minutes. However, many of these users have other input devices already at hand. Examples are joysticks and touchpads, which are found on many power wheelchairs, and trackballs, which are often preferred to desktop mice by people with low dexterity or limited motion (Fuhrer and Fridie, 2001). This chapter presents our work on adding text entry capabilities to joysticks, touchpads, and trackballs so that they may be used as integrated devices for both mouse control and text entry. In so doing, we aim to reduce the need for multiple devices, free a person's environment from device clutter, and remove the need to switch among different devices when controlling a computer. We specifically focus on providing gestural text entry in an effort to move away from the tedious and visually taxing experience of using an on-screen keyboard.

Motor impairments caused by neuromuscular disorders, injuries, or diseases manifest themselves in different ways. Accordingly, they affect people's ability to

use computers differently. For example, users with muscular dystrophy (MD), who number about 250 000 in the United States, suffer from degenerative voluntary muscle control (Myers *et al.*, 2002). Accordingly, MD results in reduced strength, limited endurance, reduced flexibility, and slowed movements (Sears and Young, 2003). However, it does not often result in tremors or spasms. Thus, many users with MD remain relatively accurate in their fine motor skills, although they are slow and prone to rapid fatigue.

In contrast, people with cerebral palsy (CP) may suffer from spasms or 'intention tremors' that often magnify as they reach for objects or attempt to acquire targets (e.g. with a mouse or stylus). They may have a more difficult time using their fine motor skills than people with MD, and may also suffer from a compromised ability to speak, which reduces the utility of voice recognition systems for these users.

Other sufferers of tremors are people with Parkinson's disease (PD), which affects over 1% of the population aged 65 and over (Sears and Young, 2003). Resting tremors are common, as are stiff muscles and limited flexibility. Difficulty controlling voluntary movements often sets in as the disease progresses. Like with CP, speech impairments are also common, limiting the extent to which voice recognition software may be used.

People with spinal cord injuries (SCI) are also targets for this work. Spinal cord injuries differ depending on the position and severity of the injury. Paraplegics with SCI, for example, may be able to use computer input devices (e.g. trackballs) with their hands, but often with reduced dexterity, strength, and endurance. These users often cannot use a desktop mouse, since shuttling the mouse across the surface of a desk requires the elevation and suspension of the forearm and wrist. They also may not be able to use a physical QWERTY keyboard, since the ability to control the fingers is compromised. Usually SCIs of this sort are the result of trauma at the C6 level or lower (i.e. C6–C8, or T1) (Sears and Young, 2003).

Other motor impairments relevant to this work include repetitive stress injury (RSI) and arthritis. These ailments may benefit from alternative forms of text entry with devices that do not exacerbate pain, discomfort, or fatigue. For example, trackballs are popular among sufferers of RSI because they do not aggravate the forearm and wrist muscles in the same way that desktop mice do.

As this chapter will show, our approach to adding text entry to joysticks, touchpads, and trackballs takes into account the limitations of these user groups by providing a more accurate, physically stable means of entering text. The core idea is to use edges and corners to provide stability and higher accuracy during movement. The result is feasible gestural text input on commonly available input devices.

14.2 Motivation

14.2.1 Joystick and touchpad text entry

An estimated 1.4 million people in the United States depend on wheelchairs for mobility (Kraus *et al.*, 1996). Of these, about 10% are in power wheelchairs, about half of whom require more than one assistive technology to participate in daily activities (Cook and Hussey, 1995). A computer access solution that works with an existing power wheelchair input device, rather than one which adds to the mix of encumbering devices, would be valuable (Guerette and Sumi, 1994). Such solutions have previously been termed 'integrated control systems' (Spaeth *et al.*, 1998).

Commercial technology already exists for enabling mouse cursor control from a power wheelchair joystick (e.g. Switch-It Inc., 2005a). But mouse control is only part of a computer access solution. The ability to enter text is also a cornerstone of successful human–computer interaction. However, an integrated text entry method to accompany joystick mouse control is not readily available. Instead, text entry with a power wheelchair joystick is mouse-driven, taking the form of point-and-click or point-and-dwell on an on-screen keyboard.

On-screen keyboards can be dissatisfying for many reasons. They require precise pointing over small targets. They also consume precious screen real-estate. When editing a document, they exacerbate the need for mouse travel between the document and the on-screen keyboard. They also impose a secondary focus-of-attention, the first being where the text is being entered, the second being the on-screen keyboard itself. This requires users to repeatedly look back and forth between their document and the on-screen keyboard, which can be visually fatiguing. In short, using a joystick-driven mouse cursor to enter text from a power wheelchair with an on-screen keyboard leaves much to be desired. By comparison, a gestural text entry method for joysticks, one which allows users to 'write' by moving the joystick in meaningful letter patterns, does not suffer from the aforementioned drawbacks. This is because gestures can be done by feel. However, the main drawback with gestures is, of course, that they must be learned.

Though less common than joysticks, touchpads can also be used to control power wheelchairs (e.g. Switch-It Inc., 2005b). Touchpads usually require less strength to operate than joysticks and little or no calibration. Like joysticks, they can be used proportionally: the further a finger is from the center of the touchpad, the faster the wheelchair turns or moves. Although touchpads have been used extensively for mousing (e.g. Hinckley *et al.*, 1998; Rekimoto, 2003), they have not generally been considered text entry devices. People in power wheelchairs might benefit from an integrated device that controls their chair, mouse, and text entry. This requires a versatile text entry technique for touchpads.

As more public information terminals (i.e. kiosks) appear in building lobbies and public libraries, on streets, in subway stations, and in community centers, the ability to access these terminals becomes more important. Just as the Americans with Disabilities Act requires that many buildings have access ramps, future terminals may be required to be accessible electronically via Bluetooth or other wireless technologies. It would be advantageous to have an integrated control system where power wheelchair joysticks or touchpads could be used as input devices for mousing and text entry on these terminals. Wheelchair users could then approach public terminals with confidence, knowing that their own wheelchair will be their gateway for access.

14.2.2 Trackball Text Entry

Although they are not used to control power wheelchairs, trackballs are often preferred by individuals with motor impairments for desktop mousing. Trackballs have many desirable properties that make them more suitable than conventional mice for people with limited dexterity, strength, or endurance in their hands or arms. The amount of effort required to roll a ball is generally less than that required to shuttle a mouse across the surface of a desk. Trackballs do not require reaching or the suspension of the forearm, as do conventional mice. Further, trackballs do not suffer from the 'edge of the pad' problem, unlike mice, which sometimes must be lifted and recentered when pushed beyond a certain range. (This action is called 'clutching.') Finally, trackballs have a very small 'desktop footprint' (Card *et al.*, 1990), making them suitable for space-constrained surfaces like the kinds of trays attached to many power wheelchairs.

Not surprisingly, many of the people who prefer trackballs due to motor impairments cannot use a conventional physical keyboard. For these people, an alternative to touch-typing is required. One option is to use a trackball with an on-screen keyboard. But on-screen keyboards have the drawbacks mentioned in the previous section. Therefore, a gestural means of 'writing' with a trackball may be a desirable alternative, allowing users to accomplish mousing and text entry with the same device.

14.3 Overview of Our Approach

Clearly, joysticks, touchpads, and trackballs are very different from one another. Adding gestural text entry capabilities to them requires leveraging their individual strengths while remaining consistent across devices. Simply providing users with a handwriting recognizer and asking them to 'write' with these devices by moving a mouse cursor around would not work very well, as these devices are not well suited

to making the kind of smooth, curved letters that one makes with a pencil or stylus (Sperling and Tullis, 1988; Accot and Zhai, 1999). Instead, these devices are best used for short, ballistic straight-line movements. Thus, our approach to adding gestural text entry to them must take this into account.

As a part of the Pittsburgh Pebbles PDA Project (http://www.pebbles.cs.cmu.edu), we are investigating how handheld devices, including personal digital assistants (PDAs), mobile phones, joysticks, touchpads, trackballs, and other devices, can be used to interact in novel ways with desktop computers (Myers, 2001). In our previous work (Myers *et al.*, 2002), we showed that a Palm PDA connected to a desktop PC could be effective for computer access for some people with motor impairments. This is because while many people with motor impairments, particularly MD, lack gross motor control, strength, and endurance, they retain enough finger dexterity to negotiate the small expanse of a PDA screen.

In addition, we invented a new text entry technique called EdgeWrite (Wobbrock *et al.*, 2003b). Originally developed for use with a stylus and Palm PDA, EdgeWrite uses a custom unistroke alphabet whose letters are composed entirely of straight-line segments within a square region, making EdgeWrite suitable for a variety of devices, including joysticks, touchpads, and trackballs. Specifically, we adapted EdgeWrite to work on an Everest & Jennings power wheelchair joystick, a Synaptics touchpad, and any trackball. We conducted three studies of these prototypes, comparing their gestural text entry performance to that of on-screen keyboards for real power wheelchair and trackball users.

In the first study, seven power wheelchair users, six with CP and one with multiple sclerosis (MS), used a power wheelchair joystick to enter EdgeWrite letters and letters with the on-screen keyboard WiViK from Prentke Romich, Inc. (Shein *et al.*, 1991). They also used a touchpad to enter EdgeWrite letters. The study was only a single session, and subjects had no prior experience with EdgeWrite letters, unlike with the WiViK keyboard, which may of them used daily. Nonetheless, with only 30 minutes of practice, touchpad EdgeWrite was the fastest and most preferred, joystick WiViK was second, and joystick EdgeWrite was a close third.

The second study was a follow-up to the first. Two subjects from the first study, both with CP, used the joystick and touchpad with EdgeWrite and WiViK over 10 sessions. The goal was to discover a 'crossover point' (MacKenzie and Zhang, 1999) where EdgeWrite overtakes WiViK for each device. Our results showed that the touchpad methods were faster than the joystick methods, and that EdgeWrite overtook WiViK on both devices after an initial learning period. This supports the notion that gestural text entry may be more satisfying for extended use than selection-based methods like on-screen keyboards.

In the third study, we developed trackball EdgeWrite over a series of nine participatory design sessions with a 15-year trackball veteran with spinal cord injury. The sessions were spread by at least a week and consisted of short 'checkpoint studies' in which we compared trackball EdgeWrite to our subject's preferred on-screen keyboard, the Microsoft Accessibility Keyboard, which he used with his favorite trackball. By the second session his speed with EdgeWrite was better than with the on-screen keyboard, and over the nine sessions he was significantly faster with EdgeWrite without leaving more errors. After 15 years, our subject no longer uses an on-screen keyboard, preferring instead to use trackball EdgeWrite.

In the remainder of this chapter, we tour the related work, describe the three EdgeWrite prototypes (i.e. the joystick, touchpad, and trackball), and discuss our studies and their results. We end with implications for stakeholders and some conclusions.

14.4 Related Work

Many devices exist for computer access, some of which can be used from a power wheelchair. Alternative physical and on-screen keyboards, head or hand switches, sip-and-puff devices, voice recognition systems, and augmentative or alternative communication (AAC) devices are just a few of the options available for computer access (Anson, 1997). But there are often obstacles to effective deployment. Many devices are prohibitively expensive. Others require extensive configuration or maintenance (Dawe, 2006). Some might be unwieldy, even on a power wheelchair. These and other reasons may be why prior research has found that less than 60% of people who indicate they need adaptations actually use them (Fichten *et al.*, 2000). Our aim in this work, by providing text entry techniques for existing power wheelchair and trackball devices, is to lower barriers to computer access by using devices already present.

EdgeWrite is related to other gestural text entry methods, most notably Unistrokes (Goldberg and Richardson, 1993) and Graffiti (Palm Inc., 1996). Numerous efforts have been made both in research and industry at improving text entry for mobile devices, many of which are relevant to EdgeWrite. For an in-depth treatment, see the article by MacKenzie and Soukoreff (2002).

A few methods besides EdgeWrite have been devised for 'writing' with a joystick (Wobbrock *et al.*, 2004). One is myText by Co-operwrite Ltd. (1997), which is meant for miniature mobile phone joysticks. Another is Weegie (Coleman, 2001), a dual-stick method for use on the X Window System, the graphical user interface shell that runs on UNIX.

As will be seen, joystick and touchpad EdgeWrite use physical edges and corners to provide stability of motion. Previous work has explored using edges in interaction techniques, such as placing buttons along screen edges for easier target acquisition

(e.g. Farris *et al.*, 2001; Wobbrock *et al.*, 2003a). The classic Lisa and Macintosh user interfaces placed their menus along the top of the screen for easy target acquisition using the screen's edge (Ludolph and Perkins, 1998).

Mouse cursor control using a power wheelchair joystick has been studied previously (Romich *et al.*, 2002; LoPresti *et al.*, 2004), but not with an integrated text entry technique. Like the current work, the study by LoPresti *et al.* (2004) also used the on-screen keyboard WiViK for mouse-based text entry.

Touchpad interaction techniques have existed for some time, but few text entry techniques have been developed for touchpads. Two limited exceptions are a touchpad used for a television remote control (Enns and MacKenzie, 1998) and for numeric entry using a clock-face metaphor (Isokoski and Kaki, 2002). Neither of these, however, is a generic touchpad text entry technique like EdgeWrite. Most touchpad techniques focus on control and selection tasks (e.g. Hinckley *et al.*, 1998; MacKenzie and Oniszczak, 1998; Rekimoto, 2003). Similar to touchpad EdgeWrite, templates have been used before on touch surfaces to help guide finger motion (Buxton *et al.*, 1985).

Trackballs have also been evaluated as mousing devices, but not for text entry. Studies have generally found them to be slower and less accurate than conventional mice for able-bodied users in pointing, dragging, and steering tasks, but comparable to touchpads for steering (MacKenzie *et al.*, 1991; Accot and Zhai, 1999). One study noted that a particular problem with trackballs is that users often accidentally move the ball when trying to click (MacKenzie *et al.*, 2001). These findings suggest that using trackballs with on-screen keyboards is not optimal, as one must perform repeated target acquisitions. Studies have shown, however, that trackballs are well suited to short straight-line movements (Accot and Zhai, 1999). As will be seen, trackball EdgeWrite uses this type of motion.

14.5 EdgeWrite for Joysticks, Touchpads, and Trackballs

14.5.1 Stylus EdgeWrite for PDAs

EdgeWrite was originally designed to address the problem of text input on PDAs. The built-in methods of text entry on PDAs usually consist of either a small 'soft' stylus keyboard displayed on-screen with tiny 'keys,' or a stroke alphabet like Graffiti, Graffiti 2, or Jot. As one might imagine, these built-in methods can be very difficult for people with motor impairments (Wobbrock *et al.*, 2003b). Tremor and poor strength, in particular, affect a person's ability to acquire small stylus targets or smoothly stroke letter gestures.

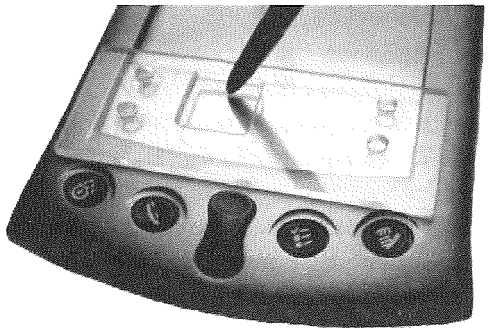


Figure 14.1 Stylus EdgeWrite on a Palm PDA.

In contrast, EdgeWrite enables people with tremor and reduced strength to write on a PDA with over 90% accuracy, even if they are unable to write Graffiti. The core concept behind stylus EdgeWrite is that of physical edges, which are placed around the input area of a PDA in the form of a square hole (Figure 14.1). These edges provide physical stability and high accuracy (Wobbrock, 2003) as a user moves his or her stylus along the edges and into the corners according to letter patterns that mimic the feel of Roman hand-printed letters. For a thorough discussion of stylus EdgeWrite for PDAs, see Wobbrock *et al.* (2003b).

The EdgeWrite letter patterns (Figure 14.2) were devised through a formal guessability study in which subjects lacking prior knowledge of EdgeWrite were asked to guess letter patterns by connecting the four corners of a square (Wobbrock *et al.*, 2005a). Although the figure shows only one form for each letter, multiple alternate forms exist

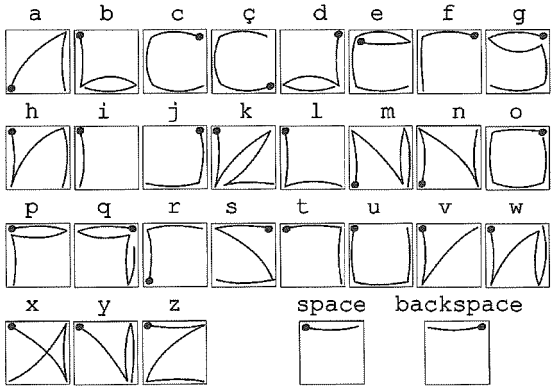


Figure 14.2 The EdgeWrite alphabet. Bowed segments are for illustrative purposes only; all motion is ideally in straight lines along edges or diagonals and into corners.

for most letters, which increases guessability. Subsequent comparisons to Graffiti show that the EdgeWrite alphabet is as guessable and immediately usable as Graffiti (MacKenzie and Zhang, 1997; Wobbrock *et al.*, 2005a).[1]

A key feature of EdgeWrite is that stroke recognition is not based on the overall path of motion, but on the order that the corners of the square are hit. This provides tolerance to wiggle and tremor since conventional path-based recognizers can be thwarted by jagged or tremulous strokes. Corner-based recognition also enables fast recognition, even on weak processors, as 'template' gestures can be encoded as simple integers representing corner sequences instead of more elaborate geometric shapes. It is this corner-based stroking and recognition scheme that allows EdgeWrite to easily adapt to other devices like joysticks, touchpads, or trackballs.

Prior results for stylus EdgeWrite show that it is significantly more accurate, over 18%, than Graffiti for able-bodied novices after 15 minutes of practice, and upwards of 3 times more accurate than Graffiti for some motor-impaired individuals with CP, MD, PD, or SCI (Wobbrock *et al.*, 2003b). Moreover, it is not significantly slower than Graffiti in speed. Able-bodied novices can write at about 7 words per minute (wpm), while experts can reach 24 wpm with only 2.8% errors (Wobbrock and Myers, 2005).

14.5.2 Joystick EdgeWrite

We implemented a version of EdgeWrite in C++ for an Everest & Jennings 1706–5020 power wheelchair joystick, which we removed from its chair (Figure 14.3). We attached wires to the joystick outputs in order to access the voltage signals corresponding to the absolute (x, y) position of the stick and the state of a switch. We used a National Instruments 6024E DAQCard to read the voltage signals and make them available to our software.

The joystick is polled for its position every 5 ms by the software. When the (x, y) position of the stick enters one of the four EdgeWrite corners, a character trace begins. When the joystick is briefly released and snaps back to the center, the trace is deemed complete and recognition of the stroke trace occurs. Using this snap-to-center approach for segmenting between letters worked very well. It was based on one of our previous studies (Wobbrock *et al.*, 2004), where able-bodied users of a video game joystick version of EdgeWrite committed no segmentation errors for thousands of letters.

The (x, y) position of the joystick was initially very noisy, in essence containing a great deal of electronic 'tremor' (Figure 14.4, left). To filter out this noise, we took

[1] A full chart with numbers, punctuation, accents, and extended characters is available at http://www.edgewrite.com.

Figure 14.3 Joystick EdgeWrite on a power wheelchair joystick. The plastic template provides a square boundary within which EdgeWrite letters can be made.

the last n points and computed a running average, treating the result as a single point (Figure 14.4, right). Trial and error yielded $n = 12$ as a value that removed sufficient noise while decreasing the inevitable stroke lag introduced by a running average.

It is important to note that the joystick is not used to drive a mouse cursor. That is, the joystick is not being used as a relative position device. Instead, the absolute position of the stick within the physical plastic square is read, so that when a user feels an edge or corner, their digital position is in agreement. For baseline comparisons, it is worth noting that able-bodied experts can write at about 12.9 wpm with this version with about 8.4% errors (Wobbrock and Myers, 2005).

Joystick EdgeWrite is a prototype system that at this time requires us to use our specific joystick. In the future, we would like to cooperate with wheelchair manufacturers

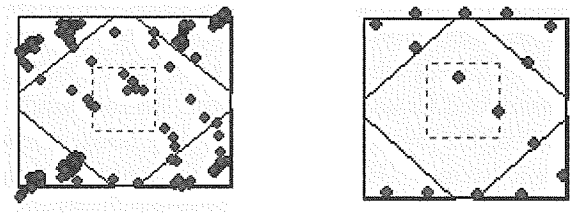


Figure 14.4 (left) A trace of the letter 's' using joystick EdgeWrite. The absolute (x, y) position of the stick was initially quite noisy. (right) A simple running average produced a clean trace.

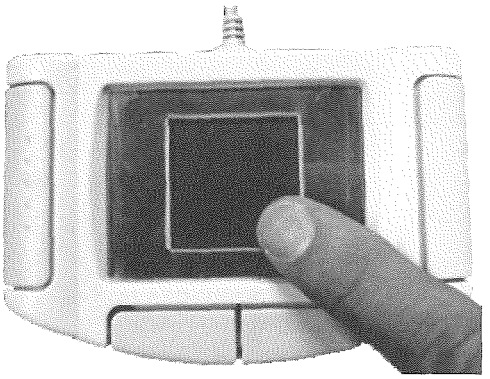


Figure 14.5 Touchpad EdgeWrite on a Synaptics touchpad with a plastic template.

to interface with their joysticks and allow our software to receive joystick inputs. EdgeWrite simply needs to know the (x, y) position of the stick in order to work, but this information is not available from most wheelchair joysticks. The EdgeWrite software receives these inputs 'in the background,' allowing the input focus to remain on any other application. EdgeWrite strokes generate key events as if they were typed on the computer keyboard, allowing joystick EdgeWrite to send text to any Windows application.

14.5.3 Touchpad EdgeWrite

We built a version of EdgeWrite in C# for use with a Synaptics touchpad (Figure 14.5). Like the stylus and joystick versions, the touchpad version used a plastic template to provide a square boundary for stability and passive haptic feedback. The edges of the touchpad's plastic template aid tremulous finger movement in the same way that physical edges aid stylus motion on a PDA. Users can feel the smooth plastic edges as they move, exerting pressure against them for stability. The touchpad surface is a capacitive sensor that detects human skin, so putting pressure on the plastic template itself does not interfere with the touchpad.

Touchpad EdgeWrite is similar to stylus EdgeWrite in that segmentation between letters is accomplished when the finger (or in the case of a PDA, the stylus) is lifted. Before a finger goes down on the touchpad, the corners are rectangular. Once a finger enters a corner, however, the corners 'deflate,' changing from rectangles into triangles. This deflation is to prevent diagonal strokes from accidentally hitting unintended corners (Figure 14.6).

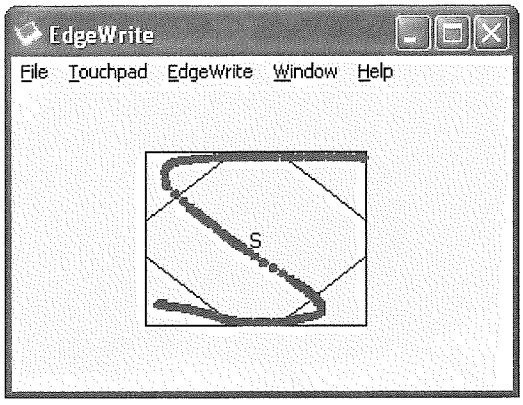


Figure 14.6 An 's' trace within the touchpad EdgeWrite square. Note the 'deflation' of the corners as triangles, which helps the diagonal part of the 's' avoid unintended corners.

Like the stylus and joystick versions, the touchpad version of EdgeWrite does not drive a mouse cursor but receives input as the absolute position of a finger within the EdgeWrite square. Synaptics drivers allow the software to read the absolute position of the finger on the touchpad's surface. With this version, able-bodied experts are able to write at about 19.1 wpm with about 4.7% errors (Wobbrock and Myers, 2005).

Touchpad EdgeWrite can send text to any application running on Microsoft Windows. With help from the Synaptics drivers, it receives touchpad events 'in the background,' so the input focus can remain on any other application window and yet EdgeWrite can receive touchpad input. EdgeWrite strokes generate key events as if they were typed on the computer keyboard, allowing touchpad EdgeWrite to send text to any Windows application.

14.5.4 Trackball EdgeWrite

Many people with motor impairments prefer trackballs to conventional mice because of the relatively low strength and dexterity required to roll a ball (Fuhrer and Fridie, 2001). Accordingly, many of these same users cannot touch-type on a conventional physical keyboard. We therefore built a version of EdgeWrite for trackballs. Unlike the stylus, joystick, and touchpad versions described above, trackball EdgeWrite does not work by reading the absolute position of a stylus, stick, or finger within a square. That's because trackballs do not report absolute position or even absolute rotation, but only changes in rotation (Card *et al.*, 1990). Trackball EdgeWrite therefore works

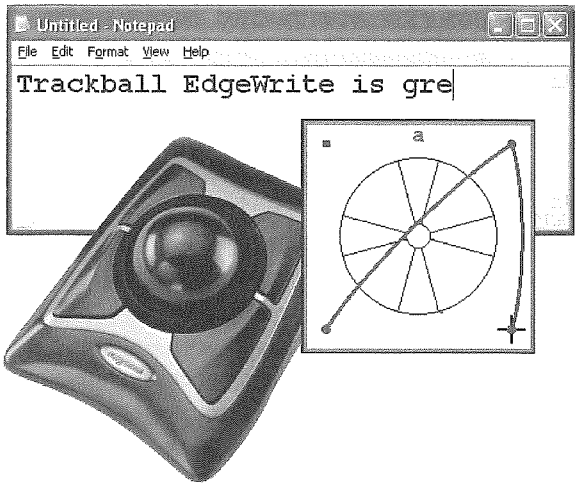


Figure 14.7 Trackball EdgeWrite works within a virtual EdgeWrite square. The cursor itself is hidden during writing, and the input focus remains on the active window (e.g. Notepad).

by observing the motion of a mouse cursor within a virtual EdgeWrite square that appears on-screen (Figure 14.7).

Although trackball EdgeWrite is based on the underlying mouse cursor, the cursor is not moved about the square as one would move a stylus, joystick, or finger in the other versions.[2] Instead, the cursor is hidden during writing, and users make short 'pulses' with the trackball toward the corners they desire. The angle (or vector) made by the cursor determines the intended corner, and an idealized stroke trace is drawn to that corner. Theoretically, this can be modeled as a 'crossing task,' which has been shown to be more accurate than traditional pointing (Accot and Zhai, 2002). Thus, subjects are not using the trackball as they would on an on-screen keyboard to point to targets. Instead, they are using the trackball to 'pulse' toward intended corners. Thus, a tight coupling exists between the corner users *feel* they are in, and the corner their stroke trace is *actually* in. Segmentation occurs after the trackball is left stationary for a brief amount of time, usually about 250 ms. Figure 14.8 depicts the process of writing an 's' and the underlying crossing tasks performed.

In Figure 14.8, the user's mouse, which is invisible, moves a short distance r from the top-right corner of the EdgeWrite square. The angle θ indicates the top-left corner,

[2]Our early prototypes were designed this way and were slow and dissatisfying. They did not 'feel like' writing because there was no correspondence between the corner the user felt he was in, and the corner the mouse cursor was actually in.

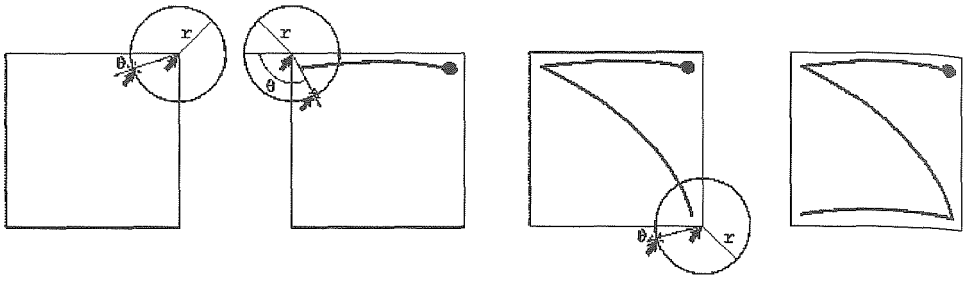


Figure 14.8 The conceptual stages of writing an 's' with trackball EdgeWrite.

and an idealized stroke is drawn to the top-left, where the mouse now appears. Next, the angle departing the top-left indicates the bottom-right corner, to where a stroke is subsequently drawn. Then the final angle departing the bottom-right indicates the bottom-left should be entered. This process is not unlike object pointing with only four objects (Guiard *et al.*, 2004).

Note that the stroke trace that is drawn is not a literal trace of mouse movement, but a smooth idealized trace directly connecting corners. The subjective experience for users is that they are 'pulsing' the trackball with subtle but distinct movements toward the corners they desire. Able-bodied users can write at about 12.7 wpm and 6.6% errors (Wobbrock and Myers, 2005).

Trackball EdgeWrite has been robustly engineered in C# to run on Microsoft Windows operating systems. Although we call it trackball EdgeWrite, it actually will work with any cursor-control device. DirectInput is used to poll the mouse position 'in the background' so that the input focus can remain on the active application (e.g. Notepad, Microsoft Word, Microsoft Outlook) and yet trackball EdgeWrite can still receive mouse movements.[3] EdgeWrite strokes generate key events as if they were typed on the computer keyboard, allowing trackball EdgeWrite to send text to any Windows application.

14.6 Study 1: Novice Use of Joystick and Touchpad EdgeWrite

This section describes our first study with seven subjects. Throughout this study we worked closely with real power wheelchair users to improve joystick and touchpad EdgeWrite.

[3]Normally, only the active application, that is the window with the current focus, receives mouse and keyboard events.

14.6.1 Mouse Control and The WiViK Keyboard

In order to compare EdgeWrite to a currently available means of text entry with a power wheelchair joystick, we compared joystick and touchpad EdgeWrite to the on-screen keyboard WiViK (Shein *et al.*, 1991) in conjunction with a power wheelchair joystick, the Everest & Jennings 1706–5020 (Figure 14.3). In order to allow subjects to use the WiViK on-screen keyboard, we implemented proportional mouse control for the wheelchair joystick (LoPresti *et al.*, 2004). We also enabled a switch on the joystick to simulate a mouse click. When the switch was pressed, it acted as a mouse-down. When the switch was released, it acted as a mouse-up.

We used the WiViK keyboard with the default settings, which included no 'dead space' between the virtual keys, no word prediction, and click-triggering of keys rather than hover/dwell-triggering. The keyboard consumed the entire width and about a third of the height of a 1024 × 768 screen. We chose the WiViK keyboard because of its familiarity to subjects as a mouse-driven on-screen keyboard.

14.6.2 Subjects

We improved the three techniques that we evaluated – joystick and touchpad EdgeWrite, and joystick WiViK – with the help of seven power wheelchair users. (We initially had eight subjects, but one was too impaired to perform any of the techniques.) Six of the seven were from the United Cerebral Palsy Center of Pittsburgh and had CP. One subject had MS. The average age of the subjects was 25.9 years, with a low of 21 and a high of 67. Subjects had been in wheelchairs for an average of 14 years, with a low of 3 and a high of 30. Two of the seven subjects were male. Four were right-handed. None of the subjects had ever used EdgeWrite, but six of seven had used WiViK. Thus, subjects were very familiar with the technique to which we would compare EdgeWrite.

14.6.3 Procedure

In order to involve subjects in the design of the techniques, we had them practice each technique before entering a single test phrase (~30 letters). Practice consisted of entering each letter four times in a row with a given technique (e.g. 'aaaa bbbb . . . yyyy zzzz'). This took about 30 minutes with the EdgeWrite techniques, and about 15 minutes with WiViK, since there was no gestural alphabet to learn. The whole test duration did not exceed 2 hours. All seven subjects used joystick WiViK and joystick EdgeWrite, but only four subjects used touchpad EdgeWrite due to time constraints. This first study did not involve using WiViK with the touchpad.

An EdgeWrite character chart was visible during the test. With the slow pace of practice and the limited endurance of subjects, we did not want to unduly burden subjects with memorizing the EdgeWrite letters. Instead, we taught them how to read the chart and observed their need to do so. Reading the chart greatly slowed them compared with their use of WiViK, which required no chart. The inter-character time – the time from the end of one character to the start of the next – gives us some idea of the delay caused by reading the EdgeWrite character chart. The average inter-character time was 6.23 seconds for the EdgeWrite methods. With more practice, this value drops dramatically, since subjects become familiar with the letters. Our second study (described below) confirms that after 10 sessions, the inter-character time drops to 3.74 seconds.

All text input was logged on the PC by a text entry test program that we wrote (Wobbrock and Myers, 2006). It was later analyzed with recently developed measures (Soukoreff and MacKenzie, 2003), which allow subjects to enter text in an unconstrained, real-world fashion, where they can choose to fix errors or not. According to this testing paradigm, subjects are merely instructed to 'proceed quickly and accurately' (Soukoreff and MacKenzie, 2003).

We solicited responses from subjects in between text entry phrases and more formally using questionnaires. In addition, many subjects offered ideas while practicing with the techniques.

14.6.4 Results

In this first study, slow performance and rapid fatigue during the practice phase meant that only one test sentence could be entered by each subject for each method. Thus, we do not have sufficient data for statistical significance. Speeds are reported in wpm and error rates in per cent. Standard deviations are reported in parentheses.

For text entry speed, touchpad EdgeWrite proved fastest at 1.00 wpm (0.72), joystick WiViK second at 0.84 wpm (0.36), and joystick EdgeWrite a close third at 0.77 wpm (0.57). See Figure 14.9.

Uncorrected errors are those left remaining in the transcribed string (Soukoreff and MacKenzie, 2003).[4] It is common for gestural methods to make more errors than selection-based methods like on-screen keyboards, particularly for novices

[4]In contrast to uncorrected errors, *corrected errors* are any letters backspaced during entry. Although corrected errors should be minimized, it is uncorrected errors that warrant real concern, since these are at odds with speed – that is, the more errors one leaves in the transcribed text, the faster one can proceed. Corrected errors, on the other hand, are subsumed in speed, since it takes time to make corrections. Thus we focus on uncorrected errors in these analyses.

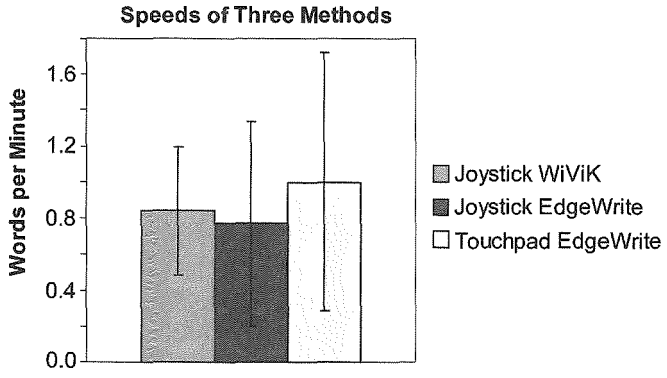


Figure 14.9 Mean speeds of three methods after minimal practice. Error bars represent ±1 st. dev.

(e.g. Költringer and Grechenig, 2004). This was indeed the case with EdgeWrite and WiViK. For uncorrected errors, joystick WiViK had the lowest rate at 0.46% ($\sigma = 1.21$), followed by joystick EdgeWrite at 6.05% (7.28), and touchpad EdgeWrite at 6.70% (8.24). These results are shown in Figure 14.10.

Clearly, subjects left more errors with joystick and touchpad EdgeWrite than with joystick WiViK. This is to be expected, since subjects often perform gestures incorrectly when learning them. On the other hand, to make an error with WiViK, a subject had to place the mouse cursor over the wrong key and still choose to press the switch, a lengthy perceptual-motor task that is easily avoided.

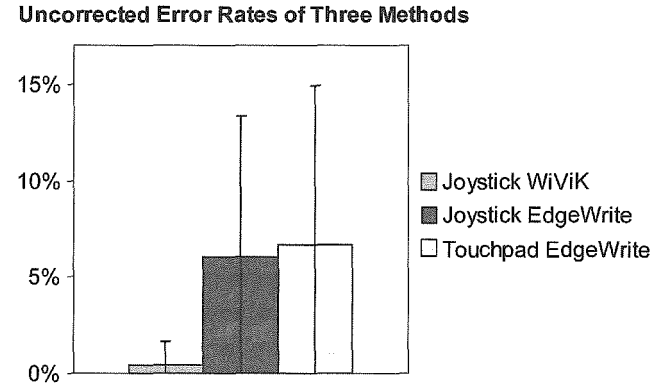


Figure 14.10 Mean uncorrected error rates of three methods after minimal practice. Error bars represent ±1 st. dev.

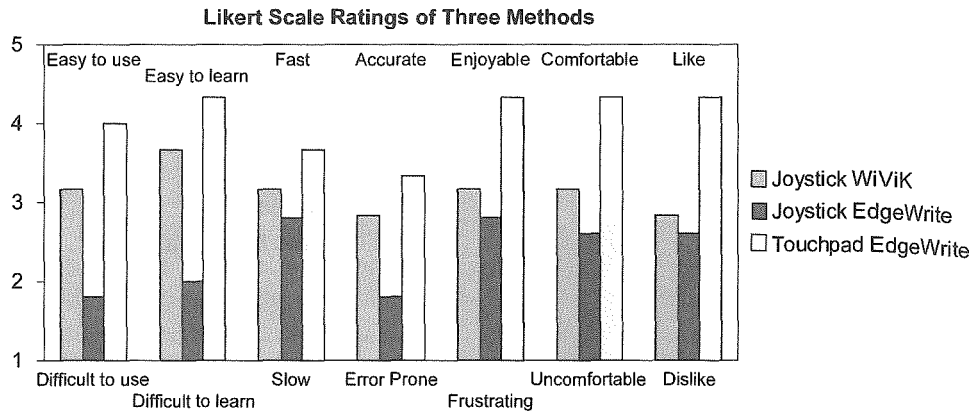


Figure 14.11 Mean Likert scale ratings for subjective measures (1–5, worst–best).

The questionnaire results showed that, of the three methods, subjects felt that touchpad EdgeWrite was the easiest to use, easiest to learn, fastest, most accurate, most enjoyable, most comfortable, and most liked. They rated joystick WiViK second in these categories, and joystick EdgeWrite third (Figure 14.11).

14.6.5 Lessons Learned from Subjects

Subject #1 was a 67-year-old retired school teacher with MS. He was notable for two reasons: he was the only person without CP, and he was only one of two subjects who was faster with joystick EdgeWrite than with joystick WiViK (1.91 vs. 1.22 wpm). The other was subject #8, who was a 22-year-old female with good fine motor control. She was only slightly better with joystick EdgeWrite than WiViK (0.52 vs. 0.50 wpm). Subject #1 showed us that the plastic template should be thicker to prevent the exposed spring on the joystick post from catching the template's edge. After using WiViK for a few minutes he said, 'It takes the patience of Job to do this.' Upon switching from WiViK to EdgeWrite he said, 'I'm much faster with this; don't you think I'm much faster?' indicating his first impression of joystick EdgeWrite.

Subject #2 was a 21-year-old student. She initially had trouble with the diagonal strokes required by joystick EdgeWrite because she would move too slowly through the center, and EdgeWrite would try to recognize her stroke to that point. She motivated us to change the center dwell time required for segmentation. If a polled joystick point falls outside the center area before the dwell time has elapsed, the dwell time counter resets. The time that worked well for subject #2 was 500 ms. However, a long

dwell time was not sufficient for subject #4, a 40-year-old volunteer. She moved inconsistently with joystick EdgeWrite, sometimes making letters very quickly, other times pausing for many seconds to think. For her we added the ability to trigger segmentation explicitly with the switch, removing the need for implicit snap-to-center segmentation. She enjoyed touchpad EdgeWrite because she said it was the easiest method with which to fix mistakes. She said, 'Once you understand what you are doing, it goes completely well.' Subject #7 echoed this when she said, 'If you got used to it, you'd be really fast I suppose.'

While the females tended to interact too gingerly with the joystick, the males, subjects #1 and #3, were too forceful at first. For them, discovering the right speed and pressure to exert against the joystick template was a large part of learning joystick EdgeWrite.

A common problem was that subjects did not always start in the corner of the plastic template before making their gestures with the joystick. This was less of a problem with the touchpad. The reason may be that the joystick must be pushed from the center to reach the starting corner, whereas a finger can begin in the corner of the touchpad.

Subject #4 gave us an important insight into the design of the touchpad's plastic template. We originally beveled the edge of the touchpad template so that it was slightly rounded. But this caused subject #4's finger to slip up onto the template's surface, actuating a 'finger lift' and prematurely triggering letter segmentation. This insight led to the fabrication of a thicker touchpad template, the edges of which we left vertical and unbeveled. We also added a settable lift-tolerance for the second study (described below).

Subject #6 highlighted the importance of end-user customizability. While using touchpad EdgeWrite, this subject's finger did not always press against an edge of the physical square. Having defined the software square for ourselves by tracing along the plastic edges, we later saw that her fingers moved inside this square, and that the actual square in which she moved was smaller than the one we had defined. When we had *her* redefine the software square by tracing around the physical square herself, her accuracy improved remarkably.

Finally, the diagonal strokes were difficult for many users of joystick EdgeWrite. This is not surprising, because it is along the diagonals that the user does not have an edge to press against. The letter 'k' was particularly problematic because of its two diagonals in a row. For our second study, we designed a new form of 'k' that is still reminiscent of a Roman 'k' but without the diagonal. This new 'k' proved much easier to perform and has become a permanent part of the alphabet in all versions of EdgeWrite. In fact, all letters except 's,' 'v,' 'x,' and 'z' now have alternate forms that contain no diagonals.

14.7 Study 2: Extended Use of Joystick and Touchpad EdgeWrite

The first study largely represents 'immediate usability' (MacKenzie and Zhang, 1997; Wobbrock *et al.*, 2005a). Its findings showed that subjects needed more practice, warranting a second investigation over multiple sessions. Such a study can identify a 'crossover point' where EdgeWrite overtakes WiViK in speed or accuracy. Subjects #2 and #4 from the first investigation agreed to partake in a 10-session study over consecutive days (except weekends). About 3 months passed between the end of the first study and the beginning of the second.

14.7.1 Subjects

Subject #2 is female and 22 years old. She uses a computer more than a few hours a day, largely for email, surfing the web, and word processing. She reports being able to slowly use a standard physical QWERTY keyboard for up to 1 hour, after which she switches to an alternative method, usually a WiViK on-screen keyboard accessed with a standard mouse. She is able to write her name with a pen, though it takes her many seconds, and it is legible to others only about 80% of the time. The joystick on her power wheelchair has a short stick with a plastic ball at the top. She is right-handed.

Subject #4 is female and 41 years old. She uses a computer only about once a week for email, surfing the web, or word processing. She, too, reports being able to use a standard physical QWERTY keyboard for up to 1 hour, after which she either stops using the computer or uses the WiViK on-screen keyboard with a standard mouse. She can write her name legibly with a pen but it takes her many seconds (if not minutes), and it is legible to others about 80% of the time. The joystick on her power wheelchair is about twice as long as the one we used in the study. It also has a softer spring. Subject #4 was much weaker than #2, which affected her ability to move the joystick snugly into the corners while using joystick EdgeWrite. Presumably this would be remedied by using a joystick more like her own. This subject was also right-handed.

14.7.2 Design Improvements

Before conducting the second investigation, we made some changes to the joystick and touchpad techniques based on observations from our first study. For example, we added more 'accidental lift tolerance' to touchpad EdgeWrite. Previously, when a finger was lifted the stroke was immediately segmented. The new tolerance, which took the

form of a customizable lift-delay, allowed subjects to briefly lift from the surface while stroking, thereby avoiding premature stroke segmentation. Both subjects preferred a tolerance of 275 ms.

The triangular corner regions in touchpad EdgeWrite were also reduced slightly from 47.5% of the square's width and height to 42.5%. This was because subjects would sometimes hit unwanted corners when making diagonals, particularly when moving from the bottom-left to the upper-right corners.

The area considered the 'center' for joystick EdgeWrite letter segmentation was reduced by about 11% to make accidental segmentations less common, since subjects would often move too slowly through the center region while making a diagonal. In the first study, this caused the software to think the joystick had snapped to center for segmentation between letters. We also found that subject #2 required a 500-ms center dwell segmentation time, while subject #4 required 1000 ms.

The joystick used with WiViK was rate-controlled, so the farther it was moved from its center, the faster the mouse cursor moved. The acceleration transfer function was linear from the center of the joystick to its extremes. For joystick WiViK, we reduced this acceleration from a maximum of 1.2 pixels/ms to 0.8 pixels/ms. We made this change because of some occasional target overshooting while subjects tried to acquire keys on the WiViK keyboard during the first study. With the reduced acceleration, target overshoots in the second study were rare.

Finally, for this second study we included a condition for touchpad WiViK. Thus, the study contained four input techniques, two devices crossed with two text entry methods.

14.7.3 Procedure

The experiment was a $2 \times 2 \times 10$ within-subjects factorial design with factors for method (EdgeWrite or WiViK), device (joystick or touchpad), and session (10 sessions). With only two subjects, the experiment was aimed less at achieving statistical significance and more at observing how long it takes subjects to become proficient at EdgeWrite.

Each subject performed all four techniques (method $\times$ device) during each session. Technique order was assigned randomly by the software for each session. Subjects warmed up with each technique before testing by entering about 10 letters. During the test, subjects transcribed two phrases of about 30 letters each for a total of about 60 letters. Subjects were quite slow, so this process took about 20 minutes per technique. Test phrases were drawn randomly from the published test corpus of 500 phrases by MacKenzie and Soukoreff (2003).

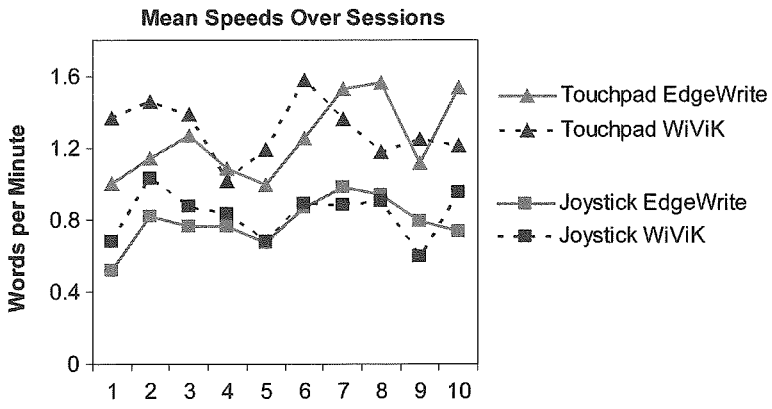


Figure 14.12 Mean speeds over 10 sessions for two subjects.

Unfortunately, subject #2 was unable to finish all 10 sessions due to intervening commitments. She finished six sessions with joystick WiViK due to technical problems during the seventh and eighth sessions. She finished eight sessions with the other three techniques. These limitations are taken into account in our analyses.

14.7.4 Results

There was a significant speedup for the two EdgeWrite methods over sessions, but no significant speedup for WiViK. This was expected, since there is more to learn with a gestural method than an on-screen keyboard already familiar to the subjects. Figure 14.12 depicts the mean session speeds for each of the four method × device combinations.

The touchpad was generally faster than the joystick for both methods at 1.27 vs. 0.81 wpm, respectively. There was no significant difference between EdgeWrite and WiViK overall (1.02 vs. 1.06 wpm), since EdgeWrite was slower in the early sessions but faster toward the end for both devices.

Overall uncorrected errors were less than 1% for all methods excluding joystick EdgeWrite in session 1 (14.6%) due to subjects refamiliarizing themselves with the technique. Average uncorrected errors for sessions 2 through 10 were 0.68% (1.41) for joystick WiViK, 0.31% (0.37) for joystick EdgeWrite, 0.28% (0.41) for touchpad WiViK, and 0.26% (0.39) for touchpad EdgeWrite. Clearly, subjects were very conscientious in correcting any errors made during entry, so speeds can be compared fairly.

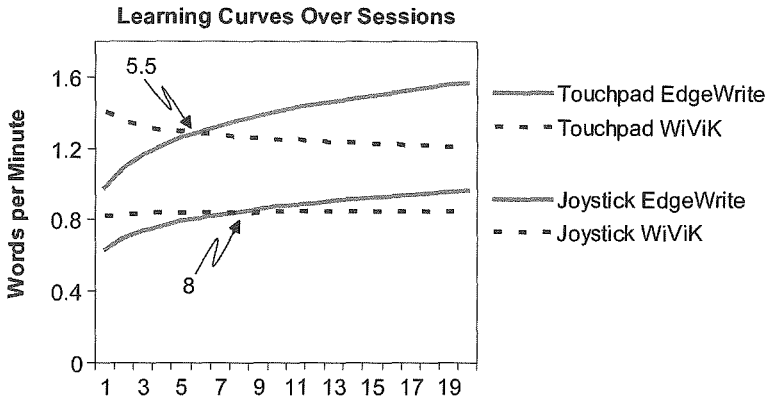


Figure 14.13 Learning curves extended to 20 sessions for our two subjects.

14.7.5 Learning Rates

It is customary in input studies to fit regression curves to speed data based on the power law of learning (Card *et al.*, 1978; MacKenzie and Zhang, 1999). Such curves are of the form $y = bx^c$, where y is speed and x is session, and b and c are regression coefficients. These performance models allow us to predict how a subject might perform in future sessions. Fitting these curves is speculative for the current data, however, since we only have two subjects. Nonetheless, the curves give a sense of how their performance may proceed beyond session 10.

The regression curves for the four method × device combinations are shown in Figure 14.13. The data is highly varied, so obtaining high correlations is not possible. But the learning curves show clear upward trends for the two EdgeWrite methods. The flat or downward slopes for WiViK are due to subjects' prior familiarity with WiViK from extended use. In addition, on-screen keyboards require little learning and offer little room for improvement. Thus, the WiViK curves are governed more by fatigue than by learning.

The power law equations and R^2 values are as follows:

$y = 0.978x^{0.1574}$	$R^2 = 0.3755$	Touchpad EdgeWrite
$y = 1.397x^{-0.0492}$	$R^2 = 0.0873$	Touchpad WiViK
$y = 0.632x^{0.1406}$	$R^2 = 0.3609$	Joystick EdgeWrite
$y = 0.819x^{0.0110}$	$R^2 = 0.0025$	Joystick WiViK

Crossover points for the touchpad and joystick techniques occur at about sessions 5.5 and 8.0, respectively, with EdgeWrite overtaking WiViK in both cases. While these

curves are speculative, they do give a sense of the overall trends. The higher R^2 values for EdgeWrite suggest that more learning took place for these methods than for the WiViK methods, and would likely continue to do so with further practice.

Overall, the results for speed and accuracy confirm both the challenge of learning a gestural text input method and the potential benefits. The initially poor accuracy of EdgeWrite, particularly the joystick version, is not surprising, and could be mitigated with further design. For example, both subjects' personal power wheelchair joysticks were longer and had much weaker springs than the one used in the study. Optimizing design parameters such as these might be one way to improve users' experiences. The general advantage of the touchpad over the joystick points to this device for future inclusion in computer access solutions.

A post-test questionnaire showed similar results for the two subjects as from the first study (see Figure 14.11). Both subjects preferred touchpad EdgeWrite overall, followed by touchpad WiViK, joystick EdgeWrite, and joystick WiViK. For both devices, the WiViK methods were considered easier to learn but the EdgeWrite methods were preferred for their perceived speeds.

14.8 Study 3: Participatory Design of Trackball EdgeWrite

Our third study compared trackball EdgeWrite and on-screen keyboards. Since the requirements for the design of a trackball version were not well understood, we iterated over nine participatory design sessions with a single trackball user, obtaining feedback and incorporating it into the next version of the software. At this early stage of design, it would have been premature to run multiple subjects in a single session; so instead, we opted to run one subject over multiple sessions.

14.8.1 'Jim'

Our participatory design subject, who we will call 'Jim,' has a spinal cord injury and is in a power wheelchair. He lacks fine motor control in his hands and arms. He is 46 years old, has used trackballs for over 15 years, and prefers the Stingray trackball from CoStar Corporation (Figure 14.14).

Jim has two methods for entering text. For long sections of prose, like lengthy emails, class papers, and formal documents, Jim uses Dragon Naturally Speaking, a voice recognition program. However, he says it often 'acts up,' failing to recognize his speech and becoming frustrating to him. Sometimes fatigue or medications alter his voice and make speech recognition problematic. He also says that it is tiresome to put

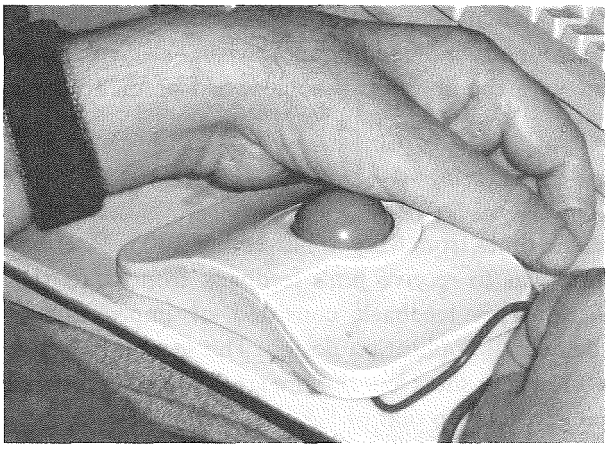


Figure 14.14 Jim's hand upon his Stingray trackball.

on and take off his microphone headset whenever he wants to use speech recognition, especially for writing quick replies to emails. Furthermore, Jim works in a lab where others sit nearby, and speaking aloud can disturb them. Finally, Jim says that certain computer-related tasks don't work well with voice recognition, like filling out web forms, writing search queries, naming or renaming files, entering email addresses, editing spreadsheet formulae, working with command-line interfaces, and retrieving contacts in an address book. Thus, Jim has always relied on a secondary method of text entry: point-and-dwell with a mouse cursor on an on-screen keyboard. He has tried numerous on-screen keyboards but has found the Microsoft Accessibility Keyboard to be his favorite. He uses it with the minimum hover-time setting of 0.5 seconds. This means that after his mouse cursor remains over a virtual key for half a second, the key is 'pressed' and the letter is sent to the active application.

Jim has three main complaints about on-screen keyboards. The first is that they are extremely visually intense and therefore fatiguing. He finds himself constantly looking back and forth between the on-screen keyboard and his document. He also must locate each letter on the on-screen keyboard in a visually taxing 'hunt and peck' manner. Jim's second complaint is that the on-screen keyboard is slow. He must have the dwell time set long enough that he does not trigger unwanted keys as he moves over them, but short enough that once he arrives at his target key, it does not take forever to actuate. This inherent trade-off makes typing with point-and-dwell laborious, but point-and-click is not viable for Jim. Jim's third complaint is that performing repeated letter-by-letter selections is, in his words, 'mind numbing.'

Jim says that word prediction can reduce the letter-by-letter doldrums, but that he feels it often slows him down by adding more visual elements to inspect. This sentiment is consistent with the findings of some word prediction studies (e.g. Koester and Levine, 1996).

14.8.2 Procedure

We met with Jim nine times from May to November 2005. Meetings consisted of two objectives. The first was to receive Jim's feedback from using the latest version of trackball EdgeWrite. This feedback consisted of improvements to the writing mechanics and the overall application design. The second objective was to measure Jim's speed and accuracy in short 'checkpoint studies' that compared his performance with his on-screen keyboard to that of EdgeWrite in a series of five test phrases with each. This was to ensure that the design changes we were making were beneficial, and also to collect detailed data about Jim's performance.

Jim permitted us to log his use of EdgeWrite at all times, providing us with valuable information on which to base our changes. Over the course of the study, Jim used EdgeWrite sporadically, mainly for short text entry tasks. His weekly usage varied from a few minutes to a few hours. Over time, he became proficient with EdgeWrite and began to use it more. By the end of the study, Jim no longer used his on-screen keyboard, but preferred to use EdgeWrite instead. After 15 years, he said he once again could feel himself 'writing.'

14.8.3 Results

Jim's mean speeds over the weekly checkpoints are shown in Figure 14.15. After the first week, Jim's EdgeWrite speed (mean = 5.61, $\sigma = 1.40$, max = 8.25) remains consistently higher than his on-screen keyboard speed (mean = 4.86, $\sigma = 1.17$, max = 6.90). A Wilcoxon sign test shows that his EdgeWrite speeds were significantly faster than his on-screen keyboard speeds ($z = -19.5$, $p < 0.02$).

Jim's uncorrected errors varied but were generally low for both methods. Uncorrected errors for EdgeWrite were 1.12% (1.39) and for the on-screen keyboard were 2.03% (2.45). These are shown over sessions in Figure 14.16. Although EdgeWrite was lower on average, a Wilcoxon sign test is not quite significant ($z = 8.0$, $p = 0.22$).

Jim's learning curves are shown extended to week 50 in Figure 14.17. Although such an extension is speculative, it gives us an idea of the possible trends for Jim's data. As in the second study, we see that the use of on-screen keyboards, particularly by those familiar with them, is not as well-modeled by the power law of learning as EdgeWrite is.

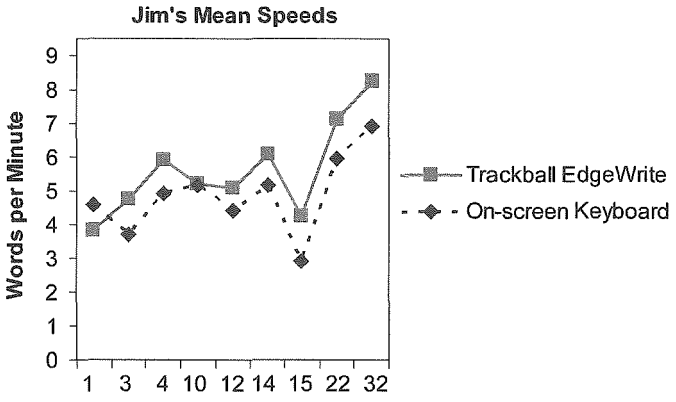


Figure 14.15 Jim's average speeds over nine checkpoint studies spread over 32 weeks.

The power law equations and R^2 values for Jim's curves are:

$y = 3.713x^{0.1850}$	$R^2 = 0.5274$	Trackball EdgeWrite
$y = 3.796x^{0.1127}$	$R^2 = 0.2144$	On-screen Keyboard

Jim no longer uses an on-screen keyboard with his trackball, but keeps trackball EdgeWrite running at all times, able to be called up at a moment's notice. To begin writing, Jim simply places the mouse cursor in the top-left corner of his screen, waits a

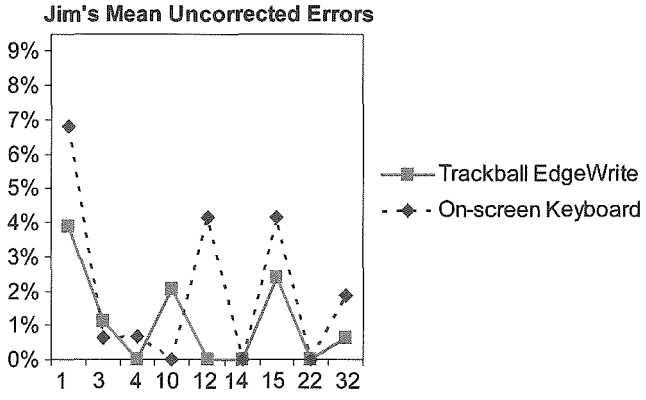


Figure 14.16 Jim's average uncorrected errors over nine checkpoint studies spread over 32 weeks.

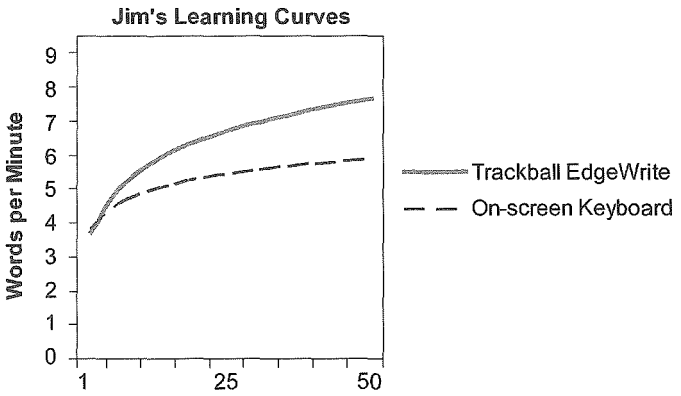


Figure 14.17 Jim's learning curves as wpm extended to week 50.

moment, and then watches as the cursor is 'captured' automatically within the EdgeWrite square for writing. When he is done writing, he makes a dedicated stroke to 'release' his mouse cursor and dismiss the EdgeWrite window. Thus, he can operate trackball EdgeWrite without ever clicking a mouse button. The idea for this button-less design was a contribution Jim himself made during our participatory design sessions. He also helped us improve the 'look and feel' of the software with suggestions for altering the on-screen depiction and stroke feedback while writing.

Jim's main reasons for preferring trackball EdgeWrite over an on-screen keyboard are, in his words:

- 'The on-screen keyboard is so terribly boring. EdgeWrite is fun, like a video game. The on-screen keyboard is not fun. I don't care which is faster.'
- 'With EdgeWrite, you can keep your eyes on your document and just write as you would holding a pencil. I don't feel disabled when I'm using EdgeWrite.'
- 'The on-screen keyboard requires too much visual scanning and concentration. In EdgeWrite, if you know the letter, you can just bang it out by feel.'
- 'I feel like I can write again.'

When Jim began working with trackball EdgeWrite, he had trouble using a PDA because he could not enter text. A neat benefit for Jim was that his trackball EdgeWrite knowledge transferred directly to the PDA for use with stylus EdgeWrite (Wobbrock *et al.*, 2003b). We compared his stylus EdgeWrite accuracy to his accuracy with Graffiti 2, the built-in Palm OS alphabet, and found that Jim was 99.0% accurate with

EdgeWrite compared to just 54.8% accurate with Graffiti 2. This comparison did not require Jim to recall strokes from memory, but was simply a comparison of motor performance while Jim looked at a character chart.

14.9 Implications for Stakeholders

The current work has implications for a variety of stakeholders. This section describes some of those implications, highlighting areas of importance for future work.

14.9.1 Users

People with motor impairments who wish to access computers are faced with a variety of challenges in acquiring, configuring, and maintaining access technologies (Dawe, 2006). Accordingly, less than 60% of people who indicate they need adaptations to use computers actually use them (Fichten *et al.*, 2000). The current work attempts to lower the barriers to access for users by providing text entry solutions for use with devices that are readily available, inexpensive, robust, and easy to configure and maintain. So-called 'commodity' devices can be purchased at everyday electronics and computer stores, and do not require a therapist or specialist to configure. This is particularly true of our touchpad and trackball EdgeWrite solutions. Our joystick solution, on the other hand, was prototyped specifically for use with the Everest & Jennings 1706–5020 power wheelchair joystick. In order for joystick EdgeWrite to become widely available, wheelchair manufacturers would have to make wheelchair joystick inputs available to third-party software. One can imagine future wheelchairs equipped with USB or serial ports, making it possible for users to benefit from add-on software solutions like joystick EdgeWrite.

All of the software discussed in this chapter is available for free download at http://www.edgewrite.com.

14.9.2 Developers

This work brought to light issues relevant to developers of both software and hardware. One issue is that access technologies must achieve a high level of robustness in order to be genuinely useful. Our trackball version, for example, was not 'academicware,' but commercial-grade software capable of running stably in a user's System Tray for weeks or even months. This level of robustness and reliability was necessary for the software to become adopted into actual daily use. Furthermore, if prototypes are successful, disabled users may come to rely heavily on them. The developer therefore has

an additional responsibility to his subjects to provide them with stable software that will support their real-world needs.

Another consideration for developers is to prioritize their application's interoperability with other software. Too often, research software is engineered to run in a stand-alone test bed where its integration with other software, if appropriate, is left for later (or never). High fidelity prototypes, in particular, should strive to integrate well into users' actual computing environments. For example, joystick, touchpad, and trackball EdgeWrite could have been built as stand-alone applications where the text destination was contained within the prototype software itself. Instead, extra care was taken to make these versions able to send text to any other Windows application, allowing them to support real-world text entry.

Developers can download an EdgeWrite library (a Windows DLL) and create their own versions in any .NET language (C#, J#, VB .NET) in under 10 lines of code. The library is fully documented with examples and code snippets. Visit http://www.edgewrite.com to obtain the developer's library.

14.9.3 Researchers

As assistive technology researchers, we learned three important lessons from conducting this work:

- *Small changes can make big differences.* The success of input systems depends on a tight perceptual-motor loop that is quite fickle. Small adjustments, from beveling edges to adjusting timeouts by milliseconds, can have a large effect on the resulting user experience of such systems. Similar sentiments were expressed by Ted Selker in his development of the Trackpoint isometric joystick found on many IBM laptops (Ehrlich, 1997).
- *Logging is essential.* When designing input systems, 'intuition' is generally not reliable for making quantitative adjustments like segmentation timeouts or the size of input regions. Input systems must be outfitted with log-writing capabilities so that adjustments can be made based on quantitative data of actual user performance. Similarly, log data can capture extended field use over days, weeks, or months, and yield insights not available in laboratory settings.
- *Real users are invaluable.* All of this work relied heavily upon multiple sessions with real power wheelchair and trackball users. Our best attempts at designing apart from them produced solutions that were lacking in many ways. Early deployment, continual testing, and rapid iteration are keys to successful input technique development. Designers and engineers should sit with users during

these sessions and not completely offload this work onto usability specialists. Only then will they see how their innovations fare. In assistive technology and human–computer interaction, if a design does not succeed with real users, it does not succeed.

14.9.4 Policymakers

Those who make policy concerning assistive technologies may be able to contribute in meaningful ways to the spirit motivating this work. Policies that help reduce the cost of specialized assistive technologies and make it easier for special populations to acquire computer access devices will help lower barriers to access. Standards committees also can influence the interoperability of hardware and software devices. Certainly our work on joystick EdgeWrite would have been made much easier, and potentially much more successful, if wheelchair joysticks had standardized ports from which computers could receive data. Then subjects would have been able to use their own wheelchair joysticks instead of the one in our prototype.

In addition, American and European populations are aging rapidly. By 2030, over 20% of the population in the United States will be 65 or over; in Europe this number will exceed 23.5% (Kinsella and Phillips, 2005). Many more people will have motor impairments of some kind, particularly those associated with aging (e.g. arthritis). These users must have adequate computer access technologies if they are to continue to participate in an information society. Policies must be upheld or enacted that put real emphasis on the need for handheld and desktop computing technologies to become more accessible, lest we risk creating another digital divide.

14.10 Conclusion

In this chapter, we described three means of integrating text entry into pre-existing input devices for power wheelchair and trackball users. All three devices – joysticks, touchpads, and trackballs – are small, light, inexpensive, and require minimal configuration, giving them significant practical advantages as integrated control systems over dedicated computer access technologies. We described our design and implementation of EdgeWrite and the participatory role real power wheelchair and trackball users played in our development process. We presented results from three separate studies of real-world users in which EdgeWrite was compared to various on-screen keyboards. Although these techniques still have much room for improvement, this work has opened the way for their future refinement, and ultimately, better computer access.

Acknowledgments

The authors thank the following people for their input and contributions: Edmund F. LoPresti, Htet Htet Aung, Janis Thoma-Negley, Steve Hayashi, John A. Kembel, Ryan S. Baker, Richard Simpson, Elaine Wherry, and John SanGiovanni.

This work was supported in part by grants from the NEC Foundation of America, NISH, General Motors, Microsoft Corporation, and the National Science Foundation under Grant No. UA-0308065. Any opinions, findings and conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect those of any supporting person or institution.

References

Accot, J. and Zhai, S. (1999) Performance evaluation of input devices in trajectory-based tasks: an application of the Steering Law. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '99),* ACM Press, New York, pp. 466–472.

Accot, J. and Zhai, S. (2002) More than dotting the I's: foundations for crossing-based interfaces. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '02),* ACM Press, New York, pp. 73–80.

Anson, D.K. (1997) *Alternative Computer Access.* F.A. Davis, Philadelphia.

Buxton, W., Hill, R. and Rowley, P. (1985) Issues and techniques in touch-sensitive tablet input. *Proceedings of the ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '85),* ACM Press, New York, pp. 215–224.

Card, S.K., English, W.K. and Burr, B.J. (1978) Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for text selection on a CRT. *Ergonomics,* 21, 601–613.

Card, S.K., Mackinlay, J.D. and Robertson, G. (1990) The design space of input devices. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '90),* ACM Press, New York, pp. 117–124.

Coleman, M. (2001) *Weegie.* `http://weegie.sourceforge.net/.`

Cook, A.M. and Hussey, S.M. (1995) *Assistive Technologies: Principles and Practice.* Mosby Press, St. Louis, MO.

Co-operwrite Ltd. (1997) *myText.* `http://www.my-text.com/.`

Dawe, M. (2006) Desperately seeking simplicity: how young adults with cognitive disabilities and their families adopt assistive technologies. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '06),* ACM Press, New York, pp. 1143–1152.

Ehrlich, K. (1997) A conversation with Ted Selker. *Interactions,* 4(5), 34–47.

Enns, N.R.N. and MacKenzie, I.S. (1998) Touchpad-based remote control devices. CHI 1998 Conference Summary, ACM Press, New York, pp. 229–230.

Farris, J.S., Jones, K.S. and Anders, B.A. (2001) Acquisition speed with targets on the edge of the screen: an application of Fitts' Law to commonly used web browser controls. *Proceedings of the Human Factors and Ergonomics Society 45th Annual Meeting,* Santa Monica, CA, Human Factors and Ergonomics Society, pp. 1205–1209.

Fichten, C.S., Barile, M., Asuncion, J.V. and Fossey, M.E. (2000) What government, agencies, and organizations can do to improve access to computers for postsecondary students with disabilities: recommendations based on Canadian empirical data. *International Journal of Rehabilitation Research,* 23(3), 191–199.

Fuhrer, C.S. and Fridie, S.E. There's a mouse out there for everyone. *Proceedings of CSUN's 16th Annual Conference on Technology and Persons with Disabilities,* California State University Northridge, available at `http://www.csun.edu/cod/conf/2001/proceedings/0014fuhrer.htm`

Goldberg, D. and Richardson, C. (1993) Touch-typing with a stylus. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '93),* ACM Press, New York, pp. 80–87.

Guerette, P. and Sumi, E. (1994) Integrating control of multiple assistive devices: a retrospective review. *Assistive Technology,* 6(1), 67–76.

Guiard, Y., Blanch, R. and Beaudouin-Lafon, M. (2004) Object pointing: a complement to bitmap pointing in GUIs. *Proceedings of Graphics Interface 2004,* Waterloo, Ontario, Canadian Human-Computer Communications Society, pp. 9–16.

Hinckley, K., Czerwinski, M. and Sinclair, M. (1998) Interaction and modeling techniques for desktop two-handed input. *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '98),* ACM Press, New York, pp. 49–58.

Isokoski, P. and Kaki, M. (2002) Comparison of two touchpad-based methods for numeric entry. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '02),* ACM Press, New York, pp. 25–32.

Kinsella, K. and Phillips, D.R. (2005) Global aging: the challenge of success. Population Bulletin, Washington, DC, Population Reference Bureau, 60(1).

Koester, H.H. and Levine, S.P. (1996) Effect of a word prediction feature on user performance. *Augmentive and Alternative Communications,* **12**(3), 155–168.

Költringer, T. and Grechenig, T. (2004) Comparing the immediate usability of Graffiti 2 and virtual keyboard. Extended Abstracts of the ACM Conference on Human Factors in Computing Systems (CHI '04), ACM Press, New York, 1175–1178.

Kraus, L.E., Stoddard, S. and Gilmartin, D. (1996) *Chartbook on Disability in the United States,* Washington, DC, National Institute on Disability and Rehabilitation Research.

LoPresti, E.F., Romich, B.A., Hill, K.J. and Spaeth, D.M. (2004) Evaluation of mouse emulation using the wheelchair joystick. *Proceedings of the 27th Annual Conference of the Rehabilitation Engineering and Assistive Technology Society of North America (RESNA '04),* RESNA Press, Washington, DC.

Ludolph, F. and Perkins, R. (1998) The Lisa user interface. CHI 1998 Conference Summary, ACM Press, New York, 18–19.

MacKenzie, I.S. and Oniszczak, A. (1998) A comparison of three selection techniques for touchpads. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '98),* ACM Press, New York, pp. 336–343.

MacKenzie, I.S. and Soukoreff, R.W. (2002) Text entry for mobile computing: models and methods, theory and practice. *Human Computer Interaction,* 17(2), 147–198.

MacKenzie, I.S. and Soukoreff, R.W. (2003) Phrase sets for evaluating text entry techniques. Extended Abstracts of the ACM Conference on Human Factors in Computing Systems (CHI '03), ACM Press, New York, 754–755.

MacKenzie, I.S. and Zhang, S.X. (1997) The immediate usability of Graffiti. *Proceedings of Graphics Interface 1997,* Toronto, Canadian Information Processing Society, pp. 129–137

MacKenzie, I.S. and Zhang, S.X. (1999) The design and evaluation of a high-performance soft keyboard. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '99),* ACM Press, New York, pp. 25–31.

MacKenzie, I.S., Sellen, A. and Buxton, W. (1991) A comparison of input devices in elemental pointing and dragging tasks. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '91),* ACM Press, New York, pp. 161–166.

MacKenzie, I.S., Kauppinen, T., and Silfverberg, M. (2001) Accuracy measures for evaluating computer pointing devices. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '01),* ACM Press, New York, pp. 9–16.

Myers, B.A. (2001) Using hand-held devices and PCs together. Communications of the ACM, 44(11), 34–41.

Myers, B.A., Wobbrock, J.O., Yang, S. et al. (2002) Using handhelds to help people with motor impairments. *Proceedings of the ACM SIGCAPH Conference on Assistive Technologies (ASSETS '02),* ACM Press, New York, pp. 89–96.

Palm Inc. (1996) Graffiti text input system. `http://www.palm.com/us/products/input/ Palm_Graffiti.pdf.`

Rekimoto, J. (2003) ThumbSense: automatic input mode sensing for touchpad-based interactions. Extended Abstracts of the ACM Conference on Human Factors in Computing Systems (CHI '03), ACM Press, New York, 852–853.

Romich, B.A., LoPresti, E.F. and Hill, K.J. (2002) Mouse emulation using the wheelchair joystick: preliminary performance comparison using four modes of control. *Proceedings of the 25th Annual Conference of the Rehabilitation Engineering and Assistive Technology Society of North America (RESNA '02),* RESNA Press, Washington, DC, pp. 106–108.

Sears, A. and Young, M. (2003) *Physical disabilities and computing technologies: an analysis of impairments.* In Jacko, J. & Sears, A. (Eds.). *The Human-Computer Interaction Handbook,* Lawrence Erlbaum Associates, Mahwah, NJ, 482–503.

Shein, F., Hamann, G. and Brownlow, N. (1991) WiViK: a visual keyboard for Windows 3.0. *Proceedings of the 14th Annual Conference of the Rehabilitation Engineering and Assistive Technology Society of North America (RESNA '91),* RESNA Press, Washington, DC, pp. 160–162.

Soukoreff, R.W. and MacKenzie, I.S. (2003) Metrics for text entry research: an evaluation of MSD and KSPC, and a new unified error metric. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '03),* ACM Press, New York, pp. 113–120.

Spaeth, D.M., Jones, D.K., and Cooper, R.A. (1998) Universal control interface for people with disabilities. *Saudi Journal of Disability Rehabilitation,* 4(3), 207–214.

Sperling, B.B. and Tullis, T.S. (1988) Are you a better 'mouser' or 'trackballer'? A comparison of cursor-positioning performance. *SIGCHI Bulletin,* 19(3), 77–81.

Switch-It Inc. (2005a) Mouse driver. `http://www.switchit-inc.com/mouse_driver.htm.`

Switch-It Inc. (2005b) Touch drive. `http://www.switchit-inc.com/touch_drive.htm.`

Wobbrock, J.O. (2003) The benefits of physical edges in gesture-making: empirical support for an edge-based unistroke alphabet. Extended Abstracts of the ACM Conference on Human Factors in Computing Systems (CHI '03), ACM Press, New York, 942–943.

Wobbrock, J.O., Aung, H.H., Rothrock, B. and Myers, B.A. (2005a) Maximizing the guessability of symbolic input. Extended Abstracts of the ACM Conference on Human Factors in Computing Systems (CHI '05), ACM Press, New York, 1869–1872.

Wobbrock, J.O. and Myers, B.A. (2005) Gestural text entry on multiple devices. *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '05),* ACM Press, New York, pp. 184–185.

Wobbrock, J.O. and Myers, B.A. (2006) Analyzing the input stream for character-level errors in unconstrained text entry evaluations. *ACM Transactions on Computer-Human Interaction (TOCHI),* 13(4), 458–489.

Wobbrock, J.O., Myers, B.A. and Hudson, S.E. (2003a) Exploring edge-based input techniques for handheld text entry. *Proceedings of the 23rd IEEE Conference on Distributed Computing Systems Workshops (ICDCSW '03),* Los Alamitos, CA, IEEE Computer Society, pp. 280–282.

Wobbrock, J.O., Myers, B.A. and Kembel, J.A. (2003b) EdgeWrite: a stylus-based text entry method designed for high accuracy and stability of motion. *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '03),* ACM Press, New York, pp. 61–70.

Wobbrock, J.O., Myers, B.A. and Aung, H.H. (2004) Writing with a joystick: a comparison of date stamp, selection keyboard, and EdgeWrite. *Proceedings of Graphics Interface 2004,* Waterloo, Ontario, Canadian Human-Computer Communications Society, pp. 1–8.