OPTIMIZING HUMAN PERFORMANCE IN MOBILE TEXT ENTRY

STEVEN JOHN CASTELLUCCI

A DISSERTATION SUBMITTED TO
THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

GRADUATE PROGRAMME IN COMPUTER SCIENCE
YORK UNIVERSITY
TORONTO, ONTARIO

DECEMBER 2014

ABSTRACT

Although text entry on mobile phones is abundant, research strives to achieve desktop typing performance “on the go”. But how can researchers evaluate new and existing mobile text entry techniques? How can they ensure that evaluations are conducted in a consistent manner that facilitates comparison? What forms of input are possible on a mobile device? Do the audio and haptic feedback options with most touchscreen keyboards affect performance? What influences users’ preference for one feedback or another? Can rearranging the characters and keys of a keyboard improve performance? This dissertation answers these questions and more.

The developed TEMA software allows researchers to evaluate mobile text entry methods in an easy, detailed, and consistent manner. Many in academia and industry have adopted it. TEMA was used to evaluate a typical QWERTY keyboard with multiple options for audio and haptic feedback. Though feedback did not have a significant effect on performance, a survey revealed that users’ choice of feedback is influenced by social and technical factors.

Another study using TEMA showed that novice users entered text faster using a tapping technique than with a gesture or handwriting technique. This motivated rearranging the keys and characters to create a new keyboard, MIME, that would provide better performance for expert users. Data on character frequency and key selection times were gathered and used to design MIME. A longitudinal user study using TEMA revealed an entry speed of 17 wpm and a total error rate of 1.7% for MIME, compared to 23 wpm and 5.2% for QWERTY. Although MIME's entry speed did not surpass QWERTY's during the study, it is projected to do so after twelve hours of practice. MIME's error rate was consistently low and significantly lower than QWERTY's. In addition, participants found MIME more comfortable to use, with some reporting hand soreness after using QWERTY for extended periods.

ACKNOWLEDGEMENTS

I would like to convey my sincerest gratitude to my supervisor, Scott MacKenzie, for his knowledge, guidance, and support. His generous funding of me via his NSERC grant made this research possible. I especially appreciate co-presenting his course *Empirical Research Methods for Human-Computer Interaction* at CHI for the past three years. Thank you to Wolfgang Stuerzlinger for your research advice, your advice on the design of MIME, and for suggesting the communication channel to log IME data in TEMA. Thank you to Burton Ma for serving on my supervisory committee on short notice.

Thank you to Rob Allison, Xiangji (Jimmy) Huang, and Mark Green for taking time out of their busy schedules to read this dissertation, provide feedback, and serve on my examination committee. I would also like to thank Hamzeh Roumani for his support of my teaching skills, Ouma Jaipaul Gill for her guidance with administrative procedures, and Susan Cameron for ensuring I received reimbursement for conference and user study expenses as soon as possible.

Thank you, Andrew Speers, for encouraging me to apply for the faculty teaching award. Thank you, Ilnaz (Elle) Ameli, for your generous help and guidance with statistics. Thank you, Michelle Brown, for your invaluable contributions to my user studies. Thank you, Kathy Weaver, for your expertise in box plots. Thank you, Robert

Teather, for being a conference guru. Thank you, Victoria McArthur, for selecting me as a CHI Student Volunteer. I would also like to thank Andriy Pavlovych, Ahmed Arif, Shumon Zaman, and Hoda Dehmeshki for their constructive feedback on my research over the years.

Most importantly, I would like to thank my parents, Betty and Sandro, for their love, encouragement, and support.

DISSIMINATION OF THIS DISSERTATION

The following chapters (in whole or in part) of this dissertation have been previously published:

Chapter 3

Castellucci, S. J. and MacKenzie, I. S., Gathering text entry metrics on Android devices. In *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems – CHI 2011*, ACM (2011), 1507-1512.

Castellucci, S. J. and MacKenzie, I. S., Encouraging consistency in mobile text entry evaluations. In *CHI 2012 Workshop on Designing and Evaluating Text Entry Methods*, ACM (2012), 43-46.

Castellucci, S. J. and MacKenzie, I. S., Gathering Text Entry Metrics on Android Devices. In *Proceedings of the International Conference on Multimedia and Human Computer Interaction – MHCI 2013*, International ASET Inc. (2013), 119.1-119.8.

Chapter 4

Castellucci, S. J. and MacKenzie, I. S., Touch Me, Hear Me, Feel Me: Feedback Preferences for Mobile Text Entry. In *Proceedings of the International Conference on Multimedia and Human Computer Interaction – MHCI 2013*, International ASET Inc. (2013), 123.1-123.6.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

ANOVA .... Analysis of variance

ART... Aligned rank transform

CER... Corrected error rate

cpm... Characters per minute

IME .... Input method editor

IPC .... Inter-process communication

KOP... Keyboard optimization problem

KSPC... Keystrokes per character

MIME.. My Input Method Editor

MSD.. Minimum string distance

OTT... Over-the-top

QAP... Quadratic assignment problem

SMS... Short Message Service

TED... Text entry distribution

TEMA .... Text Entry Metrics on Android

TER .... Total error rate

TLX... (NASA) Task Load Index

UER...Uncorrected error rate

UUID..Universally unique identifier

wpm...Words per minute

Chapter 1
Introduction and Motivation

For more than a century, people have interacted with machines to enter text. Starting with typewriters and transitioning to computers, typing is now the primary method of preparing reports and writing correspondences in both academia and industry. Compared to hand-written text, typed text is consistently legible and (with practice) can be produced more quickly. Typing is so prolific that some elementary schools are teaching typing skills to students in kindergarten [53].

Computer users young and old are accustomed to entering text using a keyboard while seated at a desk. Mobile devices are now facilitating text entry in more diverse environments and situations. Mobile touchscreen devices, such as smartphones and tablets, are now pervasive in contemporary society and are often used for SMS text messaging and social networking. However, mobility comes at a cost. Instead of using all ten fingers to type on a stationary desktop keyboard, mobile users often balance holding a mobile device with other items (e.g., purse, briefcase, umbrella, or coffee cup). This often leaves only one or two fingers for entering text. In addition, being mobile (or even stationary in a constantly changing, possibly crowded environment) can negatively affect a user's attention and accuracy when entering text. Thus, investigating methods for optimizing mobile text entry is an important research topic.

Many mobile devices use a touchscreen and a soft keyboard (also known as a software or onscreen keyboard) for text input. Compared to devices with a physical keyboard, touchscreen devices have a larger screen and are lighter and smaller. Compared to physical keyboards, soft keyboards are easier and less expensive to develop and deploy. Although current soft keyboards lack the preferred tactile feedback of physical keys, emerging technology may eliminate this disadvantage [112]. Soft keyboards can change appearance depending on context (e.g., numerical versus alphanumeric entry), previous user input (e.g., word completion), or disappear completely when text entry is not needed. This last benefit allows other content (e.g., pictures, text, or video) to occupy the precious screen space previously reserved for text entry. In addition, touchscreens allow user interaction beyond simple "button" presses and allow users to draw gestures using a finger or stylus. This has led to a variety of soft keyboard designs that attempt to replicate the performance of desktop typing in a mobile environment.

1.1 Established Performance Metrics

Two performance metrics are predominantly used to evaluate and compare text entry techniques: entry speed and accuracy. Entry speed, as the name suggests, represents how quickly a user can enter (i.e., transcribe) text, and is typically measured in words per minute (wpm). Sometimes, it is measured in characters per minute (cpm). Regardless of the actual text entered, a "word" (for the purposes of calculating entry speed) is deemed a consecutive sequence of five characters, including spaces [143]. The following equation calculates entry speed, in wpm, given the number of transcribed characters, C, and the entry time (in seconds), t:

$$EntrySpeed = \frac{C}{5} \times \frac{60}{t}$$

Equation 1. Entry speed calculated from the number of entered characters, *C*, and entry time (in seconds), *t*.

The interpretation of C is surprisingly convoluted. Measuring t starts when the user enters the first character, c_0. Because this approach ignores the mental and physical preparation time to enter c_0, entry speed calculations should not include it among the transcribed characters. Thus, a value of C-1 replaces C in Equation 1 [64, 130 (p. 49)]. However, if a hidden character (e.g., a newline or a carriage return) is used to terminate entry and timing, it should be added to the number of transcribed characters [64] (thus restoring the original value of C in Equation 1).

At first, measuring accuracy seems straightforward – an error rate simply reflects the amount of wrong input relative to all input. In user studies, participants are presented with a phrase to enter. They then enter that phrase using the technique under evaluation. Accuracy is measured by comparing the transcribed input to the presented text. But how does one classify "wrong input"? As evident in Figure 1, one might use a character-wise comparison to claim that the six characters ("xck br") are incorrect, as they do not correspond with the presented text.

Presented text: `the quick brown fox`
Transcribed text: `the qui`**`xck br`**`wn fox`

Figure 1. An example illustrating different interpretations of accuracy [105, 106].

An error rate might then be calculated as follows:

$$CharacterWiseErrorRate = \frac{CharacterWiseErrors}{\max(|A|,|B|)} \times 100\%$$

Equation 2. Calculation of a character-wise error rate, where *A* and *B* are the presented and transcribed text, respectively.

However, one might posit that the insertion of "x" and omission of "o" are the only two errors made. The Minimum String Distance (MSD) [105] function returns the minimum number of operations required to convert one string (i.e., the transcribed text) to another string (i.e., the presented text). The considered operations are as follows: insertion of a character, deletion of a character, and substitution of one character for another. Given this, MSD error rate is calculated as follows [105]:

$$MSDErrorRate = \frac{\mathrm{MSD}(A, B)}{\max(|A|,|B|)} \times 100\%$$

Equation 3. Calculation of MSD error rate [105].

Another approach analyses the user's entire input stream to better represent actual text entry interaction. Uncorrected Error Rate (UER), Corrected Error Rate (CER), and Total Error Rate (TER) [106] divide user input into the following values and are calculated according to Equation 4, Equation 5, and Equation 6, respectively:

Correct (*C*): Correctly transcribed characters.

Incorrect Not Fixed (*INF*): Incorrect or missing characters that occur in the transcribed text. This value equals the MSD for the presented and transcribed text.

Incorrect Fixed (*IF*): Incorrect characters that were corrected (and therefore do not appear in the transcribed text).

$$UncorrectedErrorRate = \frac{INF}{C + INF + IF} \times 100\%$$

Equation 4. The calculation of Uncorrected Error Rate [106].

$$CorrectedErrorRate = \frac{IF}{C + INF + IF} \times 100\%$$

Equation 5. The calculation of Corrected Error Rate [106].

$$TotalErrorRate = \frac{INF + IF}{C + INF + IF} \times 100\%$$

Equation 6. The calculation of Total Error Rate [106].

Another measure for characterizing accuracy is the keystrokes per character (KSPC) metric [60]. It represents the number of keystrokes used to enter all the transcribed characters, including those keystrokes used to correct errors. The observed KSPC measure can be compared to the technique's inherent KSPC to gauge accuracy. Considering only lowercase letters, the ubiquitous QWERTY keyboard has a KSPC of 1, as each letter of the alphabet can be entered with a single key press. Conversely, using the multi-tap method to enter text using a 12-key keypad requires multiple presses of a key to enter most letters. Observing a KSPC measure much higher than the inherent value would indicate many input errors being committed (and possible corrected), while an observed KSPC close to the inherent one would indicate accurate text entry.

Performance metrics are often calculated by running user studies (experiments) and recording empirical data, based on the actual performance of participants. Although this reflects actual performance, it can be costly and time-consuming. Another evaluation technique involves the use of mathematical models, such as Fitts' Law [29], and a language corpus to calculate expert entry speed [14, 103].

1.2 A Priori and Post Hoc Power Calculations

Determining the ideal sample size of participants for a user study can be accomplished using a priori power calculation. The calculation requires specifying an effect size and the likely standard deviation of the results [55]. In HCI, this calculation is rarely performed, as researchers simply want to investigate whether or not a statistically significant effect size exists [61 (p. 172)]. Typically, the sample size is chosen to mimic the size used in published research [61 (p. 171), 75 (p. 234)].

Post hoc power calculations appear in Chapter 6 and Chapter 7 to compare the statistical power of the parametric and non-parametric tests used. However, post hoc power calculation is controversial [35, 56, 117] and discouraged [54, 110, 117], as it simply a restatement of the p-value [35, 56, 110, 117] and leads to flawed logic about rejecting the null hypothesis [35, 54, 56, 110, 117]. Consequently, post hoc power calculations are usually not included in HCI research.

1.3 Dissertation Contributions

One of the major contributions in this dissertation is the design of a new soft keyboard for optimized mobile text entry. It is called My Input Method Editor (MIME). An "input method editor" is an Android developer term for a text input method. After conducting a thorough examination of the benefits and drawbacks of existing text entry methods, a design direction is identified. The MIME input method has three overarching characteristics: one-handed operation using one thumb for text entry, the absence of autocorrect, and easily accessible special characters.

The QWERTY layout is suitable for two-handed (or even two-thumb) input, but mobile users often have only one hand available for text entry. Some tablets have detachable physical keyboards that facilitate typing, while others are large enough to type using multiple fingers on both hands. Smartphones can be turned sideways to use a QWERTY keyboard wide enough for comfortable two-thumb use, but this layout obstructs the underlying user interface. MIME targets one handed, one thumb text entry on smartphones held in portrait orientation. To facilitate this, MIME employs a novel layout that places frequent characters in easy to select locations. This requires building a corpus to determine character frequency and gathering movement time data for onscreen key locations. Previous research gathered movement time data for stylus input, but one's grip on a smartphone makes thumb movement more restrictive than that of a stylus. Additionally, the trend towards larger smartphone screens makes traversing the width of the QWERTY keyboard burdensome.

The inaccuracy often associated with mobile text entry on a soft keyboard can be mitigated using techniques to automatically replace non-dictionary words. These techniques assume that non-dictionary words are incorrectly spelled dictionary words. However, this assumption can be wrong and the inserted words can lead to frustrating and embarrassing conversations. Such instances have become infamous in pop culture[1]. To avoid these situations, MIME does not implement autocorrect techniques.

Sending text messages and social networking often involves entering smilies or emoticons – a sequence of alphanumeric and punctuation characters used to convey one's emotions. Entering these characters on a mobile soft keyboard usually involves navigating numerous levels of submenus. MIME aims to use simple gestures and the option to "long press" (i.e., press and hold for a very short duration, rather than tap) a button to enter more characters than QWERTY, without the need for submenus.

Another major contribution is a software framework for evaluating mobile text entry methods. The software tool was used in this dissertation, and is in current use by others in academia and industry to ensure consistency in mobile text entry user studies.

The contributions of this dissertation are summarized as follows:

- Examination of the benefits and drawbacks of existing mobile text entry methods, including:
    - Character recognition
    - Menu navigation
    - Mid-air gesture recognition
    - Optimized layouts
- Development of software to facilitate text entry research on Android devices

---

[1] http://www.damnyouautocorrect.com/

- Introduction of a new methodology for conducting mobile text entry user studies
- Exploration of users' mobile text entry feedback preferences and their effect on performance
- Investigation of easily-selectable key locations on a mobile touchscreen
- Development of a new optimized mobile text entry technique

1.4 Dissertation Organization

This dissertation is organized as follows: Chapter 2 reviews existing text entry techniques that could be used on mobile and touchscreen devices. The chapter also discusses these techniques to justify preliminary design decisions for MIME. Chapter 3 presents software to evaluate text entry techniques on mobile devices, while Chapter 4 investigates the role and effect of aural and haptic feedback when typing. Evaluating text entry techniques requires a set of phrases that reflect realistic input. Chapter 5 describes the creation of this corpus to evaluate MIME. Designing the MIME character layout will require empirically determined movement times. Chapter 6 gathers these values, and Chapter 7 generates and evaluates the MIME layout. Conclusions and future work are presented in Chapter 8.

Chapter 2
Reviewing Touch-Based Mobile Text Entry

A digitizing tablet (also known as a digitizer or graphics tablet) is a surface on which a user can draw with a pen-like stylus. The advent of digitizers in 1956 sparked interest in handwriting recognition [79, 115] – computer-based text entry using handwritten gestures. However, several issues impede accurate, real-time recognition of handwriting [79, 114, 115].

Handwriting Variation: Handwriting varies between individuals to such an extent that it is used as a forensic tool to identify the writer of a document with 95% confidence [109]. Additionally, factors such as stress, carelessness, or fatigue can also result in handwriting that is too sloppy for even a human to understand.

Segmentation: Strokes that occur too closely in time and/or space are difficult to discern as separate gestures. This is especially true for cursive (script) handwriting, where an entire word could be written with a single, continuous stroke.

Semantics: Gestures could map to similar characters in the language (e.g., “O” (oh) and “0” (zero), “I” (eye) and “l” (el) and “1” (one), etc.).

These problems can be alleviated by placing restrictions on the user. A common restriction is the use of a gesture alphabet, to which characters (i.e., letters, numbers, and sometimes punctuations) are mapped. The gesture alphabet specifies the shape of handwritten strokes, their direction, and order of any intermediate strokes. Alternatively, dynamic writing information, such as the number, order, direction, and speed of intermediate strokes, can help identify the written gesture [114].

A shortcoming of gesture recognition is that English text entry is much faster using a standard QWERTY keyboard [115]. Typical handwriting speed in English is about 18-30 words-per-minute (wpm) [5 (p. 287), 22 (p. 61), 115, 143 (p. 196)], while a proficient touch typist is about twice as fast [22]. So why use gesture recognition at all for text entry?

Gesture recognition is feasible in situations where using a full sized keyboard is impractical. For example, with mobile computing (e.g., with cell phones, PDAs), users often hold the device with two hands and type on mini or onscreen keypads with their thumbs. Alternatively, they hold the device with one hand and enter text with the other. In both circumstances, the speed advantage of touch typing is minimized or eliminated. When a digitizer is integrated into the device's display (often called a touchscreen), input can be performed over the user interface. The elimination of a physical keypad can improve portability and the elimination of an onscreen keypad relinquishes valuable screen space.

Text entry using gestures is not limited to drawing characters on a digitizer. With some techniques, gestures drawn on a digitizer represent navigation of menus; the selection of a menu item corresponds to entry of a character or word. Other techniques employ sensors to recognize movement in mid-air. Computer vision technology has also been used to capture and analyse a user's hand pose [71, 82], which is mapped to entry of a character or word. However, computer vision techniques for text entry will not be covered in this review. This literature review details and compares English gestural text entry techniques in the categories of character recognition (a.k.a. "symbolic keyboards" [73]), menu navigation (a.k.a. "target keyboards" [73]), and mid-air movement. It then presents techniques that use a touchscreen to recognize user taps on an onscreen keyboard and techniques that use a combination of input to facilitate text entry.

2.1 Character-Based Recognition

First introduced in 1993, *Unistrokes* is a gesture alphabet for stylus-based text entry [31, 32] (Figure 2). The single-stroke nature of each gesture allows entry without the user attending to the writing area [32] and simple segmentation of characters. Additional gestures change modes to allow entry of uppercase letters, numbers, and punctuation [31]. Furthermore, the alphabet's strokes are well distinguished in "sloppiness space" [32], allowing for accurate recognition of not-so-accurate input.

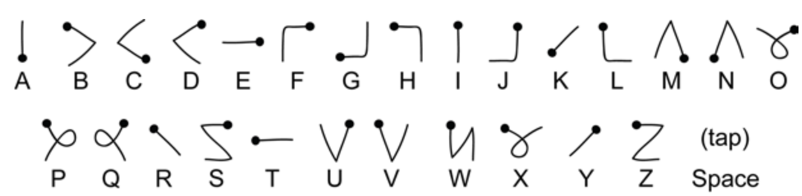


Figure 2. The *Unistrokes* alphabet [12]. The dot indicates the start of a gesture.

Unistrokes gestures bear little resemblance to Roman letters. However, each letter is assigned a short stroke, with frequent letters (e.g., E, A, T, I, R) associated with a straight line. *Unistrokes* is analogous to touch-typing with a keyboard, as practice will result in high-speed, “eyes-free” input [32]. A small pilot study of three participants showed initial text entry performance of 6.2 wpm, increasing to 13.0 wpm after a week of practice [32].

In 1996, Palm, Inc. released its PDAs with the *PalmOS* operating system. These devices allowed text entry using the *Graffiti* gesture alphabet (Figure 3) [7]. Strokes are recognized as lowercase letters, uppercase letters, or numbers, based on the location of input.

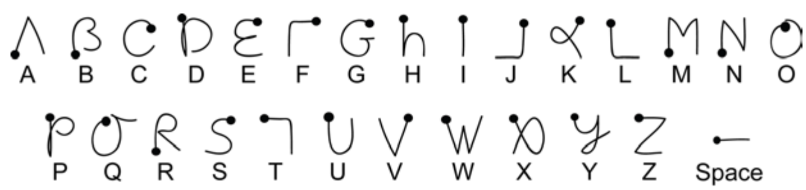


Figure 3. The *Graffiti* alphabet [12].

Like *Unistrokes*, each gesture is a single stroke. However, unlike the *Unistrokes* alphabet, *Graffiti* gestures resemble their corresponding Roman letter. This is intended to facilitate learning. Support for this was found in a previous study, where users demonstrated 97% accuracy after only five minutes of practice [68]. However, a longitudinal study spanning twenty, fifteen-phrase sessions compared *Graffiti* to *Unistrokes*. Initially, entry speed was similar between the two alphabets at 4 wpm. With practice, entry with *Graffiti* reached 11 wpm, but was surpassed by *Unistrokes* at 16 wpm [12].

Because input methods can vary greatly with device, researchers developed *Minimal Device Independent Text Input Method* (*MDITIM*) [42]. *MDITIM* maps combinations of the four compass directions to character input. Movement in the compass directions can be associated with joystick, mouse, or trackball movement, key presses on a keyboard, or gestures on a touchpad (as in Figure 4). As with *Unistrokes*, *MDITIM* was designed for robust recognition. To that end, its gestures represent prefix codes – no gesture represents the beginning of another gesture in the alphabet. This allows multiple characters to be written with a single, continuous gesture without hindering recognition [42].

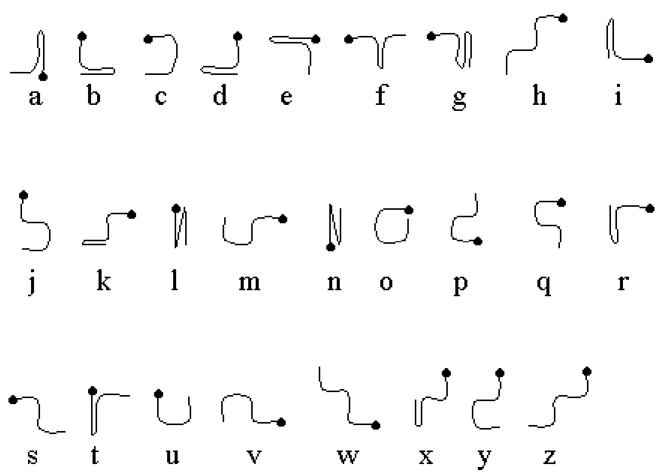


Figure 4. The *MDITIM* alphabet [64].

In a user study, participants practiced *MDITIM* text entry using a touchpad over 10 sessions. During that time, entry speed increased from 2.5 wpm to 7.6 wpm and error rates dropped from 15% to 6%. To test the device independence of *MDITIM*, participants then performed *MDITIM* entry using a joystick, keyboard, mouse, and trackball. Text entry was fastest with the touchpad and slowest with the keyboard. Joystick input had the highest accuracy, while the trackball yielded the lowest [42].

As with handwriting, stylus text entry typically requires a high degree of motor coordination. *EdgeWrite* [140] was introduced in 2003 to facilitate text entry for users with motor impairments (e.g., Cerebral Palsy, Muscular Dystrophy, etc.) [129]. *EdgeWrite* gestures (Figure 5) resemble Roman letters, but are drawn along the edges of the writing area. A raised border is placed around the writing area to guide movement. Gestures are recognized based solely on the order in which corner regions are hit [140]. Thus, hand tremors are less likely to result in recognition errors.

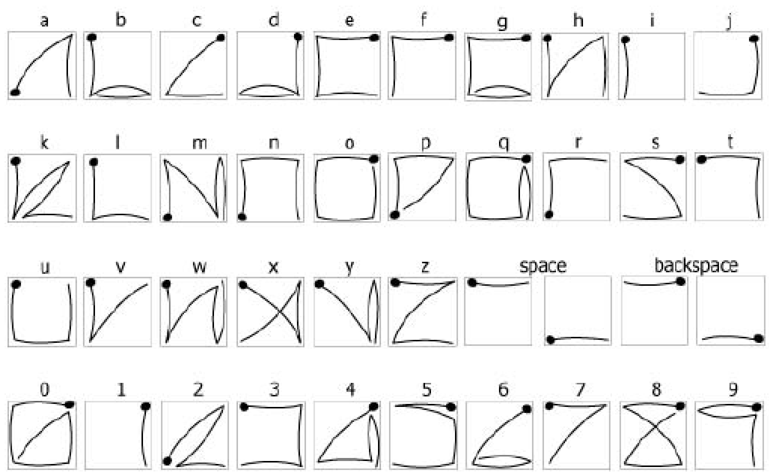


Figure 5. The *EdgeWrite* alphabet [137].

To improve recognition further, the regions defined as "corners" in the writing area change based on the handedness of the user and after hitting an initial corner (Figure 6) [140].

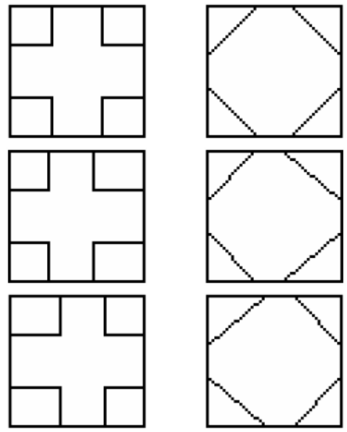


Figure 6. After the first corner hit, the areas recognized as corners transition from the layout on the left to the one on the right [140]. The top pair of layouts represents an ambidextrous layout. The middle pair caters to right handed input, while the bottom pair favours left handed input.

A small study involving four participants with differing motor impairments compared *EdgeWrite* to *Graffiti*. Each participant performed their individual task (specific to their motor skill) more accurately with *EdgeWrite* than with *Graffiti* [140]. A study of ten able-bodied participants yielded entry speeds of 6.6 wpm for *EdgeWrite* and 7.2 wpm for *Graffiti*. Error rates were 0.34% for *EdgeWrite* and 0.39% for *Graffiti*. These values were averaged over eight sentences, after twelve practice sentences [140].

The *EdgeWrite* technique has been adapted for use with multiple input devices (Figure 7), including gaming joysticks [136, 137], wheelchair joysticks [136, 138], isometric joysticks [18, 134, 136], trackballs [132, 136], and four-button keypads [136]. *EdgeWrite*'s simple corner-based input scheme makes it viable in many devices and precludes the need for an onscreen menu or keyboard. Furthermore, its robust recognition algorithm is well-suited for motor impaired users and "situationally impaired able-bodied user who are on-the-go" [136].

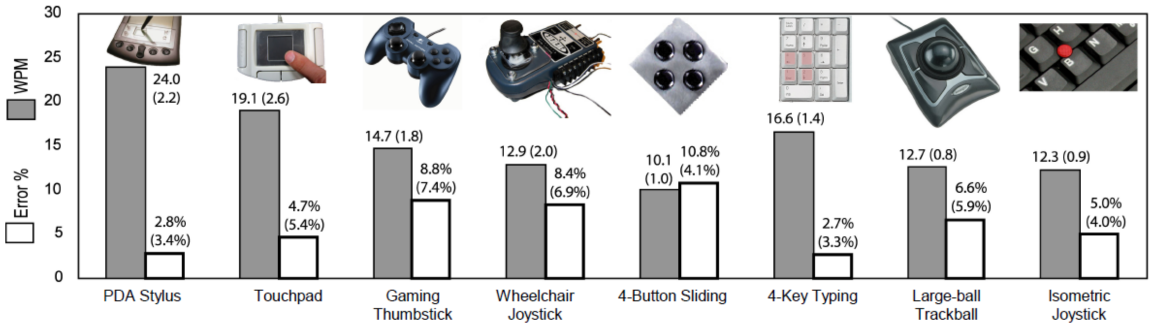


Figure 7. A summary of devices used with *EdgeWrite* [136].

The initial *EdgeWrite* implementation used a touchscreen or touch pad. Character segmentation would occur when the user's stylus or finger lifted from the input surface. With wheelchair and gaming joystick input, the limits of stick movement provide the input boarders; character segmentation occurs when the stick returns to its rest position. With isometric joystick, trackball, and keypad input, segmentation is triggered by a timeout (i.e., duration of no input). Input via a four-button keypad simply associates corner hits to button presses. However, the boundless input of an isometric joystick and trackball require more sophisticated recognition.

With *Trackball EdgeWrite* [132], gestures are not continuous motions, but rather a series of "pulses" (i.e., strokes) that determine movement between the four *EdgeWrite* corners. Angular thresholds distinguish between vertical, horizontal, and diagonal transitions. Two additional features maintain robust recognition of gestures: non-recognition retry and slip detection. With non-recognition retry, an incorrect gesture can be restarted without triggering character segmentation. If the recognition algorithm cannot resolve the input as a valid gesture, pulses are trimmed from the start of the sequence until the gesture is validated. Slip detection uses the speed of input to determine if a corner was inadvertently hit. The recognized character is determined using a binary decision tree and digraph probabilities [132]. *EdgeWrite* input with an isometric joystick uses the *Trackball EdgeWrite* technique, but with different thresholds and timeouts [134].

2.2 Menu Navigation

Pie menus [38] are radial menus in which a selection is made by drawing from the center of the on-screen menu to the desired option on the menu's outer edge. By placing characters as the menu options, pie menus can be used to enter text.

Published in 1994, *T-Cube* [119] initially presents a single pie menu to the user (centre of Figure 8). The options along its edge include whitespace characters and modifiers for uppercase letters and commands. Beginning a stroke at the center of this initial menu can select its options. However, beginning a stroke at the edge of the initial menu displays additional menus containing lowercase letter, numbers, and punctuations. The options available in the additional menus are not visible beforehand and must be memorized by the user. A longitudinal study of eleven users over nine, thirty-minute sessions shows a linear increase in entry speed from about 18 cpm (3.6 wpm) to a maximum of 106 cpm (21.2 wpm) [119].

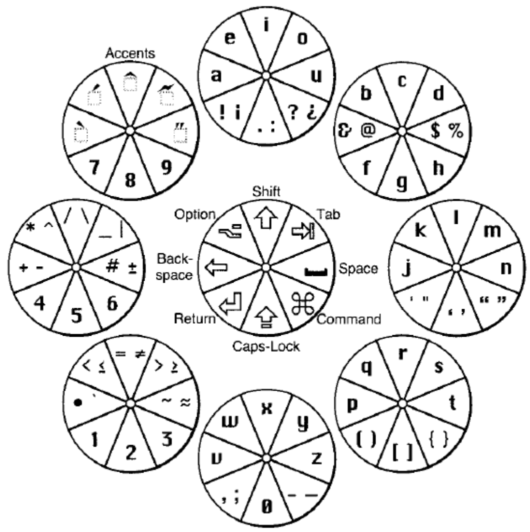


Figure 8. Inputs available via *T-Cube* [119].

Two techniques published in 1998 use pie menus to enter multiple characters with a single stroke: *Cirrin* [72] and *Quikwriting* [89]. *Cirrin* presents English alphabet characters in a pie menu. The layout of characters was chosen to minimize the average length of a word's gesture, based on an English corpus [72]. To enter a word, the user moves the pointer (controlled by a stylus, mouse, etc.) from the center of the menu to the first letter of the word, then to subsequent letters. Letters are selected if the pointer enters its region. The pointer can move from one letter's region to the region of the subsequent letter in the word if the regions are adjacent (e.g., "fin" in "finished", as depicted in Figure 9). Otherwise, the pointer must pass through the center of the menu (e.g., "ish" in "finished", as depicted in Figure 9). The gesture is completed upon pen/mouse up; the word is entered, and appended with a space. A user with two months of *Cirrin* experience was able to enter text at 20 wpm [72].

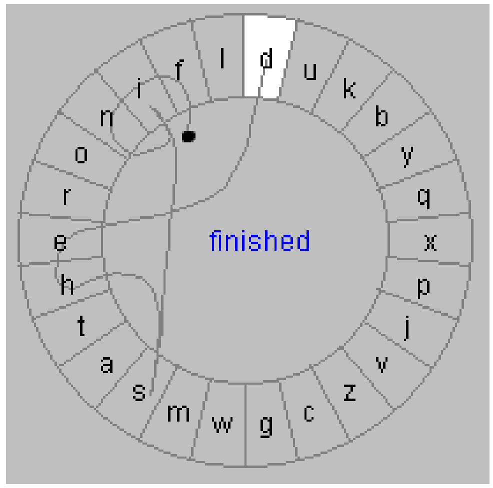


Figure 9. A path used to enter the word "finished" using *Cirrin* [72]. The pen-down location is indicated with a dot.

Quikwriting [89] divides the pie menu into a 3×3 grid, where the center zone (zone 5) is called the "resting zone" and the other zones (zones 1-4 and 6-9) are each called a "major zone". Each major zone is similarly subdivided into a 3×3 grid, with the center zone left unused and the other zones each called a "minor zone". Each minor zone represents, at most, one character (Figure 10).

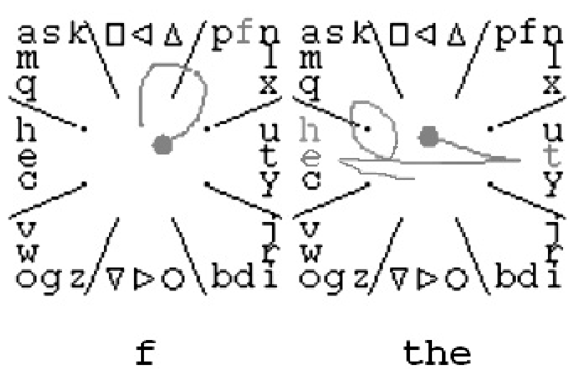


Figure 10. An example of entering "f" (left) and the word "the" (right) using *Quikwriting* [89].

A character is represented by its [*i*, *j*] coordinates in the pie menu, where *i* represents its major zone, and *j* is its minor zone. To enter a character, the pointer must leave the resting zone via major zone *i*. Once a major zone is entered, zones 1-4 and 6-9 become minor zones. The pointer must travel around the pie menu (if necessary), and re-enter the resting zone from minor zone *j*. If *i*=*j* (as is the case with “t”), then traversal around the pie menu is not necessary – the pointer would travel from the resting zone to zone 6, and immediately back to the resting zone. Character segmentation is indicated by egress and ingress of the resting zone. *Quikwriting* was designed to allow continuous writing without lifting the stylus (or releasing a button) and without halting pointer movement [89] – an entire sentence could be written with a single gesture!

A separate, longitudinal study compared using *Quikwriting* with a stylus to using *Quikwriting* with a gamepad joystick (a.k.a. thumbstick) [43]. By the end of twenty sessions, totalling five hours per device, the twelve participants averaged an entry speed of 16 wpm using the stylus and 13 wpm using the joystick [43]. Interestingly, entry rates are similar to those in a longitudinal study of *Unistrokes* and *Graffiti* input [12]. In both studies, the initial entry for all conditions is about 4 wpm. After twenty sessions, totalling about five hours per technique, *Unistrokes* entry speed reached 16 wpm, while *Graffiti* reached 11 wpm [12].

8pen (8pen.com) is a commercial variant of *Quikwriting* for Android devices. It uses a resting zone in the center of the input area, but only four major and minor zones. Each major zone represents eight characters, with two characters per minor zone. The characters are disambiguated using the movement direction of the pointer, whether clockwise or counter-clockwise. Figure 11 illustrates entering “the” using *8pen*. The pointer (i.e., the user's finger or thumb) enters the left major zone, loops once clockwise to the next zone (representing the minor zone for "t"), then re-enters the resting zone. Without stopping, the pointer then enters the left major zone, loops counter-clockwise to the second minor zone (because "h" is second from the centre), passes through the resting zone to the bottom major zone, loops clockwise to the next minor zone, and ends in the resting zone. Although the *8pen* website claims text entry speeds of 30 wpm, there are no published papers to substantiate the claim.

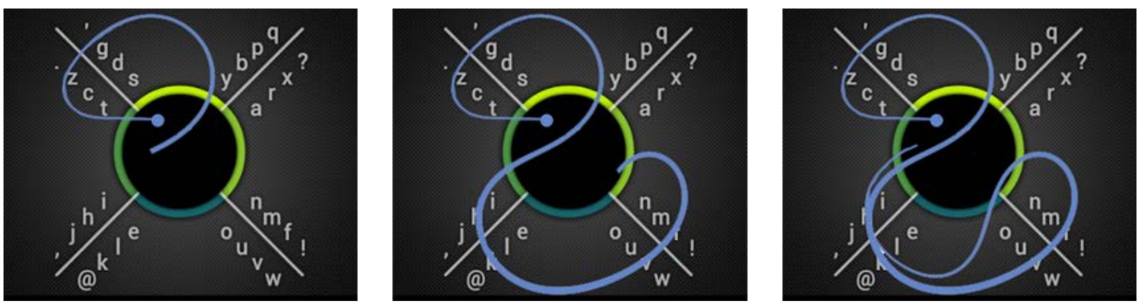


Figure 11. An example of entering "t" (left), "h" (centre), and "e" (right) using *8pen* (8pen.com).

Like *Quikwriting* with a gamepad joystick, *SonicTexting* [96] maps characters to the continuous movement of the thumbstick (Figure 12). However, with *SonicTexting*, there is no onscreen menu. Instead, characters are presented aurally. As the user moves the thumbstick along a path, he or she hears the corresponding letter, looped continuously. A character is entered by returning the thumbstick to its rest position. Entry can be cancelled by pressing down on the thumbstick, which also acts as a button. The creator of *SonicTexting* describes it as "an attempt to tap into the sources of audio-tactile gratification", citing "the addictive qualities of puncturing bubble-wrap" [96]. Although this audio-tactile gratification would not translate to text entry on mobile devices, thumbstick movement could be mapped to gestures on a touchscreen. Characters could be entered upon a finger- or thumb-up event.

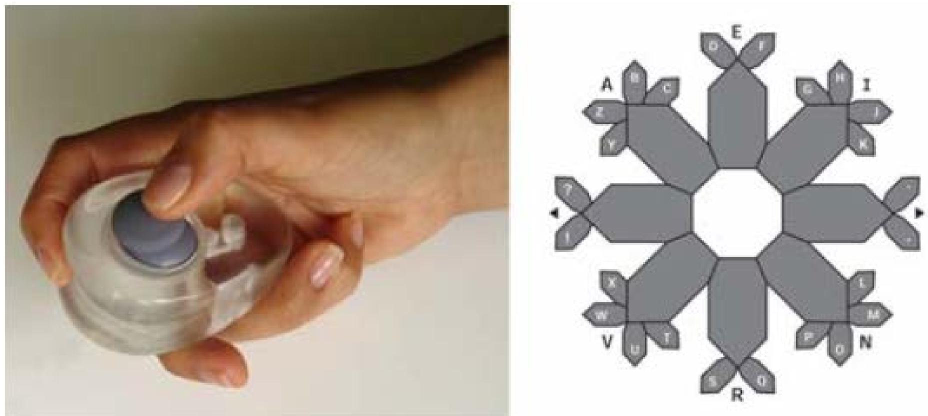


Figure 12. The input device and gesture map associated with *SonicTexting* [96].

Quikwriting was also the inspiration for *TwoStick* [47]. *TwoStick* uses a dual-joystick game controller to enter text. Users are presented with a nine-by-nine onscreen grid divided into nine 3×3 zones. Each unit in the grid represents one character, though some are empty (Figure 13). To facilitate "walk-up usability", the characters are arranged alphabetically [47]. To input a character, one thumbstick selects a zone, while the other selects a character within the zone. Returning the character-selection thumbstick to its rest position enters the character. A longitudinal study yielded text-entry rates of 4.3 wpm initially, increasing to 14.9 wpm after five hours of practice. During the same time, error rates dropped from 13.3% to 8.2% [47]. Videogames for mobile devices often provide virtual thumbsticks on the touchscreen. A similar approach could be used to implement *TwoStick*.

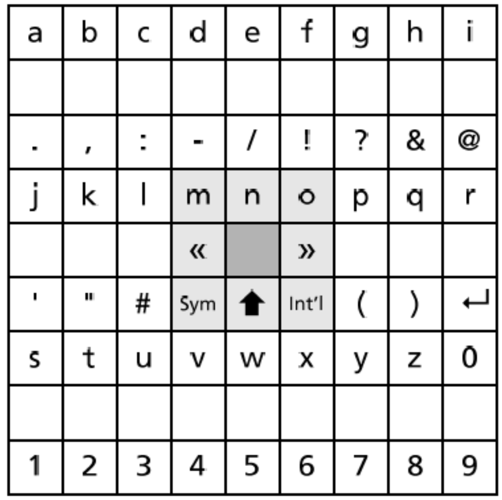


Figure 13. The grid layout for *TwoStick* [47].

While the menu-based text entry techniques usually involve selection of stationary menu entries, *Dasher* [123] presents the user with a moving menu. Pointer movement manipulates the speed and direction of the menu entries, which generally move from the right of the input area to the left. Intersecting an entry with the crosshair at the middle of the input area selects that entry. The initial, top-level menu presents the 26 letters of the English alphabet and the space character (Figure 14). For mobile devices, input could come from either the touchscreen or an integrated accelerometer that measure device tilt.

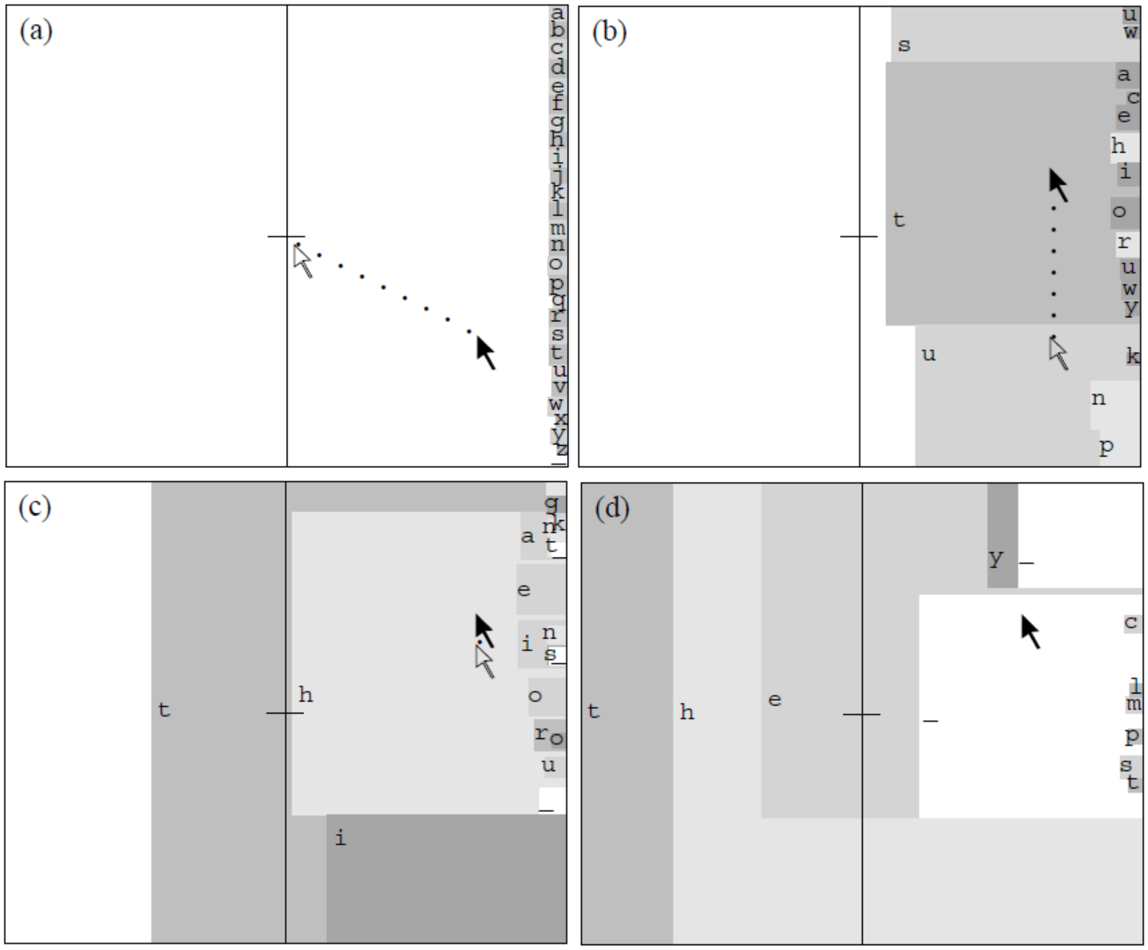


Figure 14. Entering "the" using *Dasher* [123]. The space character is shown as "_".

Menu entries increase in size as they approach the crosshairs. This way, likely selections are larger and unlikely selections are smaller, or not visible. As one menu level approaches the crosshair, the next level appears along the right edge of the input area. Subsequent levels are populated with only those letters that can form an English word, given the previously selected letters. Furthermore, the relative size of the menu entries reflects the probability of selection. Both menu characteristics are based on a model of the English language. An empirical study revealed an average text entry rate of about 18 wpm after an hour of practice, with an error rate of less than 5% [123].

Like *Dasher*, *VirHKey* presents a moving menu of characters to facilitate text entry [73]. Characters are presented on a grid of pentagons (i.e., a "pentagrid") on a hyperbolic plane (Figure 15). The direction of the user's stroke indicates the direction of the desired character. With each stroke, the view of the pentagrid is rotated so that the desired character moves closer to the center of the grid. Lifting the stylus (or releasing a button) completes the gesture, triggers character segmentation, and enters the current character at the center of the pentagrid. Because strokes serve to rotate the view of the pentagrid, they need not occur over the pentagrid itself. In an evaluative study, the pentagrid appeared on a display, while participants drew gestures with a stylus on a separate digitizer [73]. Participant initially entered 6.6 wpm, but increase speed to 22.9 wpm after twenty sessions, totalling about seven hours [73].

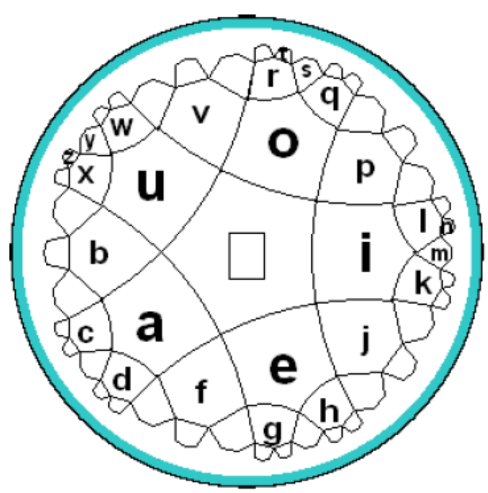


Figure 15. The *VirHKey* pentagrid layout [73].

SHARK [145] and *SHARK*^{2} [50] recognize word-gestures using an onscreen keyboard. They combine the single-stroke nature of *Unistrokes* with the continuous input characteristic of *Cirrin* and *Quikwriting*. With *SHARK*, a word's gesture starts from the key of the first letter and continues in a straight line to subsequent letters. Upon pen/mouse up, the gesture is then compared to a lexicon of recognized words. If a word is not in the lexicon, the user can resort to the point-and-tap input typically associated with onscreen keyboards. *SHARK*^{2} does not provide point-and-tap input, but compensates with an enlarged lexicon of recognized words [50]. To simplify gestures and improve recognition, these techniques use an optimized, non-QWERTY keyboard layout (Figure 16). A limited study of two participants using *SHARK*^{2} reached a speed of 70 wpm by repeatedly entering "The quick brown fox jumps over the lazy dog". The authors of the study do not mention the amount of training given to the participants and admit that the result is only an indication "of what *SHARK*^{2} could potentially achieve" [50].

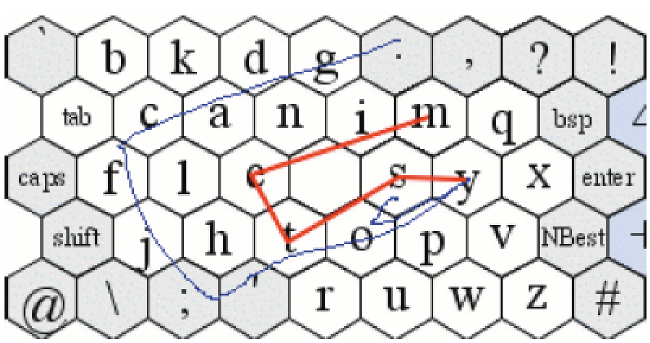


Figure 16. The word "system" being entered using *SHARK*^{2} [50]. The bold red path represents the correct gesture, while the blue path represents a sloppy gesture that is still correctly recognized. Both paths start on the S-key.

EdgeWrite was presented earlier as a character-based input technique. However, the use of an onscreen menu can provide an "integrated help" system [74]. The integrated help system displays the *EdgeWrite* input area on the screen and guides the user through each gesture. Initially, characters are grouped according to the first corner in their gesture sequence; each group is displayed in its respective corner (Figure 17). Once a corner is hit, its characters are grouped according to the next corner in sequence. In the "static" version of the system, the new groups instantly appear in their respective corners. In the "dynamic" version of the system, movement to the next corner is animated [74]. A four-session study compared using a paper chart, the static system, and the dynamic system. The dynamic system was consistently faster, peaking at about 6.7 wpm. Paper and static conditions peaked at 5.7 wpm and 5.0 wpm, respectively. Error rates were less than 1.5% in all conditions. The dynamic system initially yielded the highest error rate, but dropped to the lowest error rate by session four [74].

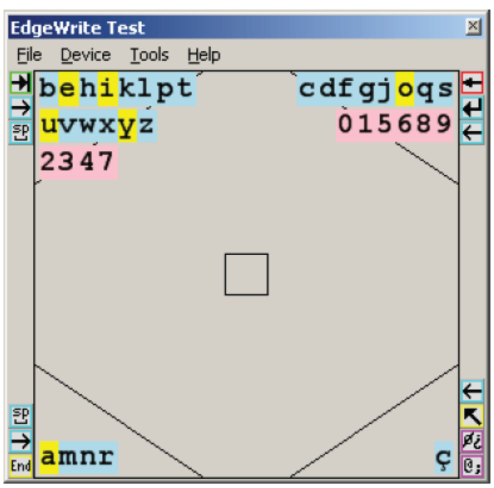


Figure 17. *EdgeWrite* with an "integrated help system" [74].

The *Hex* technique [127] uses seven hexagonal regions to facilitate text entry. One hexagon is the rest area, while the others are arranged around its perimeter. The surrounding hexagons each have six characters associated with them. When the pointer crosses from the rest area to one of the other hexagons, it becomes the rest area and its six characters are redistributed to the surrounding hexagons. Crossing into another hexagon enters its character and layout returns to the original arrangement (Figure 18). The benefit of this technique is that every character can be entered by navigating to only two regions. Unfortunately, this limits the number of characters to 36.

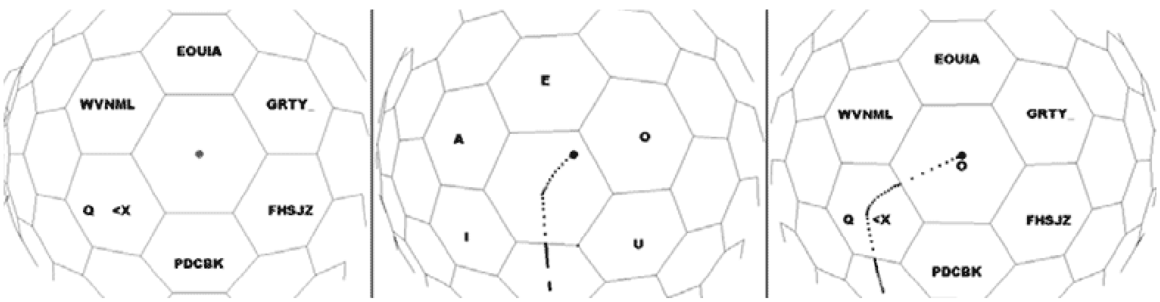


Figure 18. Entering the character "o" with *Hex* [127].

The *Hex* paper states that the pointer can be controlled using a mouse or by the orientation of a mobile device. One of the Hex authors achieved entry speeds of 10-12 wpm after approximately 30 hours of practice [127]. However, it is not clear what method of input was used.

Instead of the six directions used by *Hex*, the *LURD-Writer* technique [26] uses only four: *L*eft, *U*p, *R*ight, and *D*own. Users enter characters by moving the mouse to select one of four keys (Figure 19). The characters associated with that key are then redistributed and the process continues until a key with a single character is selected.

Clicking a mouse button enters a character. The left mouse button is used for uppercase letters and numbers, while the right mouse button is used for lowercase letters and special characters.

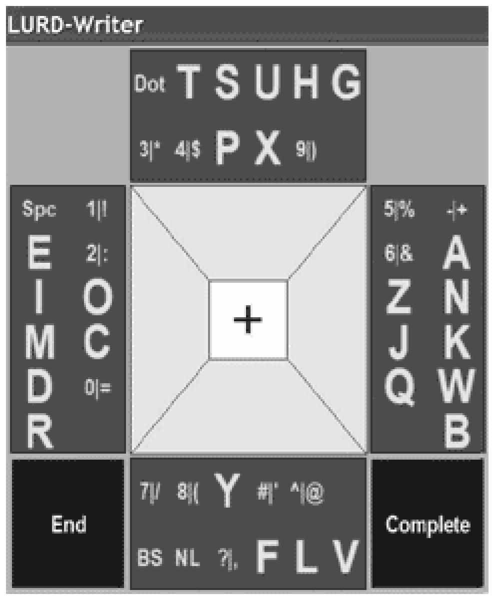


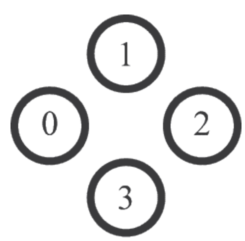
Figure 19. The character arrangement for *LURD-Writer* [26].

LURD-Writer was designed for motor-impaired users [26]. To reduce pointer movement, the pointer is re-centered after each key selection. The number of selections for characters varies, but frequent characters require fewer than infrequent ones. An evaluation with a single, motor-impaired user yielded entry speeds of 8 cpm using a mouse for input.

Distributing characters to one of four keys results in each character having an encoding sequence – a sequence of key selections used to input the character. The technique *H4-Writer* (often abbreviated to "*H4*") [66] uses Huffman codes [39] for character encodings. Huffman codes have two valuable properties: Firstly, no code forms a prefix to another code, so unlike *EdgeWrite*, no input event (e.g., a finger-up event) is required to segment character input. This allows a continuous stream of inputs to be unambiguously parsed. Secondly, using a letter frequency model guarantees that encoded messages are of minimum average length. Consequently, the KSPC value for *H4* (2.3) is lower than that of *MDITIM* (3.0), *LURD-Writer* (3.3), and *EdgeWrite* (4.4). The character encodings for *H4* use the four symbols ‘0’, ‘1’, ‘2’, and ‘3’, and maps these symbols to four gamepad keys. Table 1 shows the encodings, while Figure 20 depicts the *H4* keyboard.

Table 1. The *H4* character encodings and mapping to gamepad keys.

Character	Code
Space	33
e	11
t	22
a	23
o	20
i	13
n	12
s	31
h	10
r	322
l	300
d	321
c	303
u	302
f	301
m	323
w	213
y	212
p	211
g	210
b	3203
v	3202
k	3201
x	32003
j	32002
q	32001
z	32000



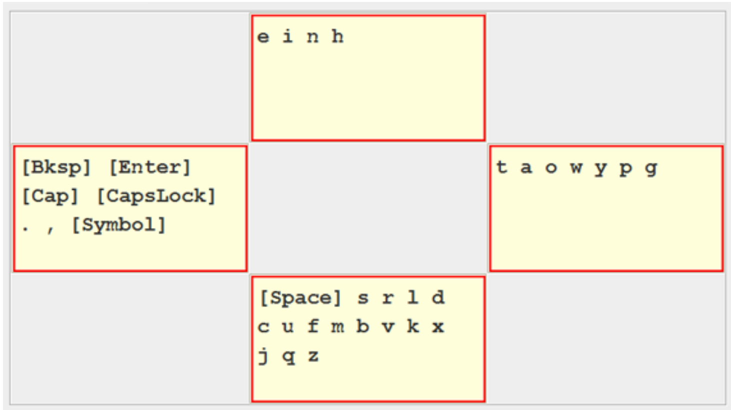


Figure 20. The onscreen *H4* keyboard in its initial arrangement.

In the default arrangement (shown), characters are assigned to the key that represents the first encoding symbol. Once a key is activated, characters are assigned to the key that represents the second encoding symbol. All non-activated characters are removed from the arrangement. This reassignment continues until a key with only one character is activated, thus completing that character's encoding. The character is entered and the character arrangement returns to the default one.

In a longitudinal study, participants reached 20.4 wpm, with an error rate of only 0.69% after 400 minutes of practice. Although the onscreen keyboard was always visible, the researchers state that participants stopped referring to it at approximately the midpoint of the study and text entry became "eyes-free" [66]. *H4* has also been used with other input methods. Mapping *H4* "keys" to directional gestures on a touchpad yielded 6.6 wpm and a total error rate of 9.2%, while mapping to mid-air gestures yielded only 5.3 wpm and a total error rate of 10.8% [16].

2.3 Mid-Air Techniques

Unigesture [101] uses an accelerometer to determine device orientation in midair. Its designers wanted to facilitate one-handed mobile text entry on small devices without the need for buttons or a digitizer [101]. *Unigesture* combines features of *Quikwriting* and *T9*. Like *Quikwriting*, each letter of the English alphabet is assigned to one of seven zones arranged in a 3×3 grid. The space character is assigned its own zone, and the middle zone is designated the rest zone and left empty (Figure 21). To input the first letter of a word, the user tilts the device in the direction of that letter's zone, and then returns the device to its rest orientation. This continues for each subsequent letter in the word, resulting in a sequence of zone selections. Entering the space character terminates entry of a word.

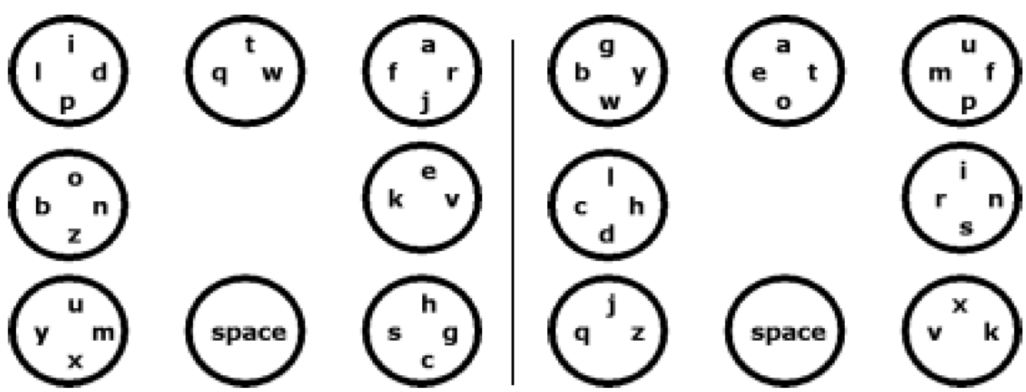


Figure 21. The "spread-out" (left) and "clustered" (right) layouts of *Unigesture* [87].

With *T9*, each key represents three or four letters; a word is represented by a sequence of key presses. A disambiguation algorithm relies on a corpus of the target language to map the sequence to a valid word. Similarly, the sequence of *Unigesture* zone selections is passed to an "inference engine" that produces the corresponding word.

In *T9* collisions occur when a sequence of key presses maps to multiple words. The user selects the intended word with presses of a “next” button. Similarly, the prototype *Unigesture* system relied on a “try again” button to traverse the list of possible words. The designers admit that design changes would be needed in a full-featured system [101].

An evaluation of the *Unigesture* system evaluated a mock hand-held device (containing the accelerometer) connected to a PC (providing visual feedback and data recording). Two letter layouts were tested: “spread-out” reduced the number of possible collisions; “clustered” simplified selection of frequent letters. In addition, two interaction styles were compared: “deep tilt” required substantial tilting to register selection, but was resistant to accidental movement; “slight tilt” registered more motion, but allowed faster selection [101].

Based on reported data [101], participants entered text at approximately 2 wpm using both letter layouts. Furthermore, the slight-tilt interaction resulted in more errors than the deep-tilt technique; specific error rates were not available. By the end of the study, two participants experienced wrist fatigue and one experienced wrist pain. A quarter of participants reported that diagonal tilts were especially difficult.

The creators of *Unigesture* went on to design *TiltType* [87], which combines tilting and button presses to enter letters, numbers, and punctuation. Three buttons are used to enter English letters. Each button is associated with a 3×3 grid of letters. Letters are arranged alphabetically and occupy all nine cells in the grid (i.e., there is no rest zone). The contents of each grid are not displayed, but mnemonic labels appear on the border of the prototype device [87]. A user types a letter by pressing the button corresponding to the letter's grid and tilting (if necessary) in the direction of the letter. At this point, the letter corresponding to the button press and tilt appear on the display. The user can change the tilt to select a different letter, or release the currently pressed button to confirm entry of the displayed letter. Unlike *Unigesture*, character input is deterministic; no disambiguation is needed. The designers did not perform an evaluation study, but informal use did not result in the wrist fatigue that was reported with *Unigesture* [87].

TiltText [126], like *TiltType*, uses device orientation to facilitate deterministic character input. Keys on a standard telephone keypad are associated with multiple letters. For example, the 2-key represents the first three letters of the English alphabet. Using a technique called *Multi-Tap*, the user would enter "a", "b", or "c" by pressing the 2-key one, twice, or trice in rapid succession, respectively. *TiltText* augments a cell phone with an accelerometer. It determines the desired letter using the tilt of the cell phone at the time of the key press. Continuing the 2-key example, users would tilt the phone left for "a", forward for "b", and right for "c". If the key had a fourth letter (as do the 7- and 9-keys), it would be selected by tilting the cell phone towards the user. The designers also considered tilting during a key press to perform entry. However, during a pilot study, they found this method to be much slower than *Multi-Tap* [126]. By the end of a 16-session study, participants were 22.9% faster with *TiltText* than with *Multi-Tap*. This improvement was statistically significant [126].

An unnamed technique places accelerometers and buttons on a glove, worn by the user [128]. The "data glove" recognizes motions upwards and downwards, rolls to the left and right, as well as button presses. The data glove has a button on the inside of the index, middle, and ring finger. However, the ring finger button was deemed too difficult to press, and is not used for text entry. Text input is done by chording – performing one or two motions in parallel with one or two button presses. Two recognition alphabets (Figure 22) were proposed and evaluated. Method 1 associates eighteen one-motion chords with two characters each. The characters are designated "Map1" and "Map2". A button press toggles between the two states. For example, performing an upwards gesture when Map1 is active enters the letter "t". Performing the same gesture when Map2 is active, results in the letter "g". Method 2 uses up to two motions per chord, but assigns a unique chord to each character. The study showed the entry rate of Method 1 to be significantly faster than that of Method 2. Method 1 was also the most accurate technique, but the difference was slight and no indication of statistical significance was given [128].

Method 1			
Gesture	Button	Map1	Map2
rotLeft		DEL	DEL
rotRight		SPC	SPC
	B1	CHG	CHG
	B2	e	m
Up		t	g
Down		o	f
Up	B1	a	y
Down	B1	RET	.
rotLeft	B1	r	b
rotRight	B1	n	w
Up	B2	i	k
Down	B2	s	v
rotLeft	B2	h	x
rotRight	B2	l	z
Up	B1+B2	d	j
Down	B1+B2	c	q
rotLeft	B1+B2	u	
rotRight	B1+B2	p	

with B1 = Button1
and B2 = Button2

Method 2			
Gesture1	Gesture2	Button	Char.
		B1	e
		B2	t
Up			o
Down			a
rotLeft			DEL
rotRight			SPC
rotLeft	Up		h
rotLeft	Down		l
rotRight	Up		d
rotRight	Down		c
Up		B1	n
Down		B1	r
rotLeft		B1	u
rotRight		B1	p
rotLeft	Up	B1	f
rotLeft	Down	B1	y
rotRight	Up	B1	b
rotRight	Down	B1	w
Up		B2	i
Down		B2	s
rotLeft		B2	m
rotRight		B2	g
rotLeft	Up	B2	k
rotLeft	Down	B2	v
rotRight	Up	B2	x
rotRight	Down	B2	z
Up		B1+B2	j
Down		B1+B2	q
rotLeft		B1+B2	.
rotRight		B1+B2	RET

Figure 22. The input maps used with the data glove [128].

Like the data glove, the *Wii Remote* (a.k.a. *Wiimote*) provides button and motion-sensing input via accelerometer and/or gyroscope. The *Wiimote* is the controller for Nintendo's *Wii U* video game system (www.nintendo.com). *Unigest* [13] facilitates text entry by mapping combinations of vertical, horizontal, and rolling motions to letters of the English alphabet (Figure 23). Frequent characters (i.e., SPACE, BACKSPACE, E, T, A, O, and I) are each associated with a single motion. The remaining letters are mapped to gestures that combine up to two motions. The gestures attempt to mimic their assigned Roman letter (in lower case) to aid its memorization [13]. This technique could be adapted for mobile text entry by using the onboard accelerometer and buttons (or touchscreen taps) for input.

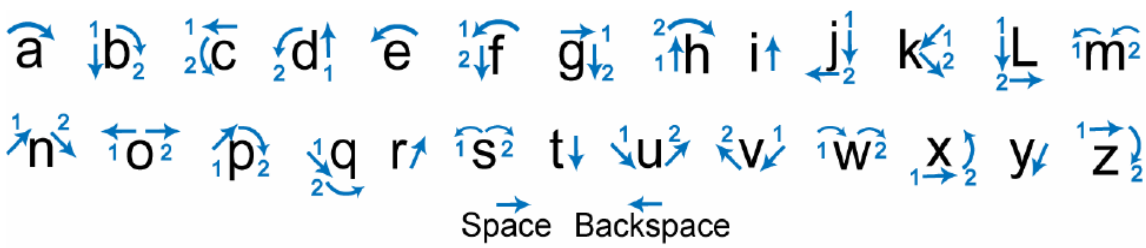


Figure 23. The *Unigest* alphabet.

Although no text entry user study has yet been performed with *Unigest*, its upper-bound entry rate is predicted to be 27.9 wpm. This prediction is based on predictive models for cellphone [103] and *Unistrokes* [32] text entry. First, a user study gathered movement times for each of the ten input motions (i.e., up, down, left, right, up-left, up-right, down-left, down-right, roll left, roll right). These movement times were used to predict the entry time for each letter. Finally, a words-per-minute speed was calculated using a letter-frequency distribution for the English language.

Text entry using a *Wiimote* has also been attempted with onscreen keyboards on a large video wall [102]. The interaction employed one of three onscreen keyboards: Circle, QWERTY, and Cube (Figure 24). With the Circle onscreen keyboard, users would select letters from a circular, alphabetical arrangement. The Cube keyboard is described as a “3D extension” of *T-Cube* [102]. Users would draw a gesture along the surface of the cube to select the desired letter. Unfortunately, Cube text entry yielded the highest error rate (7.0%), the lowest performance (7.6 wpm), and the lowest participant ratings [102].

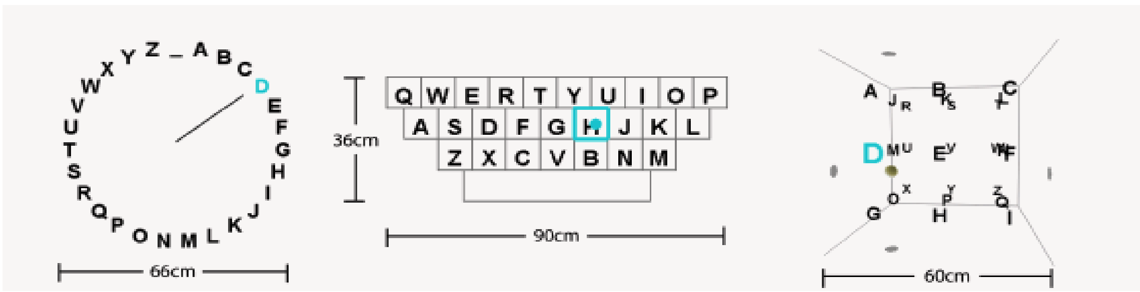


Figure 24. The three onscreen keyboards used with the video wall: Circle (left), QWERTY (middle), and Cube (right) [102].

The accelerometer in tablets has also been used to enter text [28]. Teenage participants tilted a tablet to control the position of a ball (i.e., cursor) on an onscreen keyboard. When the ball remained on a key for 500 ms, the corresponding letters was entered. Text entry was performed with one hand and with two hands, while sitting and while walking. The task involved selecting 50 letters. Performance was best when seated and gripping the tablet with two hands. Walking increased task completion time and error rate, and the one-handed grip resulted in slower entry and more errors.

2.4 Tapping and Hybrid Techniques

Instead of using gestures for text entry, some techniques provide a soft keyboard for users to simply tap (with a finger, thumb, or stylus) the desired characters. Instead of using the QWERTY layout that is ubiquitous in desktop computing, the *Opti* [69] and *Fitaly* (www.fitaly.com) layouts rearrange characters so that frequent ones are in the centre of the keyboard and frequent digrams (i.e., letter pairs) are adjacent to one another. Because of the high occurrence of the space character, *Opti* provides four space keys (each twice the size of a letter key) instead of *Fitaly*'s two. The layouts appear below in Figure 25. The *Opti II* layout represents an optimization based on the frequency of trigrams (i.e., letter triplets) to minimize movement across the keyboard [64].

z	v	c	h	w	k
f	i	t	a	l	y
space		n	e	space	
g	d	o	r	s	b
q	j	u	m	p	x

q	f	u	m	c	k	z
space		o	t	h	space	
b	s	r	e	a	w	x
space		i	n	d	space	
j	p	v	g	l	y	

q	k	c	g	v	j
spc	s	i	n	d	spc
w	t	h	e	a	m
spc	u	o	r	l	spc
z	b	f	y	p	x

Figure 25. The *Fitaly* (left), *Opti* (centre), and *Opti II* (right) layouts, after [64].

Expert entry speeds (using a single stylus) for the *Fitaly*, *Opti* (*I*), and *Opti II* keyboards are 41.96 wpm, 42.16 wpm, and 42.37 wpm, respectively. A longitudinal empirical study compared stylus-based text entry performance between *Opti* and QWERTY. Session 1 speeds favoured QWERTY with 28 wpm over *Opti* with 17 wpm. However, by the tenth session (after 200 minutes of practice), *Opti* surpassed QWERTY. Session 20 speeds were 40 wpm for QWERTY and 45 wpm for *Opti*. Character-wise error rates rose during the study, with higher error rates for QWERTY (Session 1: 3.21%, Session 20: 4.84%) than *Opti* (Session 1: 2.07%, Session 20: 4.18%) [69].

The KALQ keyboard [83] was designed with one interaction in mind: a user grasping a tablet in landscape orientation and typing with both thumbs. The researchers identified keys that can be easily selected and assigned characters to benefit from these keys and by alternating input between thumbs. After an average of 16.8 hours of training on KALQ, participants reached an entry speed of 37.1 wpm with an error rate of 5.2%. This was an improvement over the baseline QWERTY condition with an entry speed of 27.7 wpm and an error rate of 9.0%.

Figure 26. The KALQ keyboard [83] uses a split, modified layout.

Other tapping techniques use QWERTY layout to exploit users' familiarity with the layout. However, the techniques are often hybrids, facilitating input using taps, gestures, or a combination of the two. *UniKeyb* [41] allows users to combine taps on a keyboard with *Unistroke* gestures drawn over the keyboard. For example, entering the word "the" could be accomplished by three taps, three *Unistroke* gestures, or a combination of the two (e.g., a tap on "t", another tap on "h", and a horizontal gesture to enter "e"). This provides the user with many optimization opportunities (i.e., "Do I move my stylus across the keyboard to tap "e", or do I just draw its gesture here?"). A simulation suggests that augmenting tapping with *Unistroke* gestures can improve input speed by 28%. A longitudinal study was conducted using a soft keyboard with the AZERTY layout (the French equivalent of QWERTY). Session 1 speeds were only 7 wpm, but that rose to 51 wpm by Session 36 (after 3 hours of practice). *UniKeyb* entry speed surpassed that of strictly tapping by Session 17 (after 85 minutes of practice). Unfortunately the benefit in speed came at the expense of accuracy. The Session 36 MSD error rate was approximately 4% for *UniKeyb*, but only about 1% for strictly tapping. Other research [2] augments a QWERTY keyboard with gestures mapped to space, backspace, shift, and enter functionalities. The addition of gestures did not have a significant benefit when entering lower case text. There was a significant benefit in entry speed (approximately 3 wpm over ordinary QWERTY), but not error rate, when entering mixed case text.

The *KeyScretch* [30] keyboard allows users to type a single letter at a time, or to press and hold a key to enter multiple letters with a single gesture. Upon holding a key, a popup menu appears with four characters along its borders. These characters are chosen to be the most likely subsequent letters. Dragging one's finger to those characters enters them in succession. This allows input of multiple letters with a single gesture. Though, gesture input is limited to the letters in the popup menu. After 6 hours of training, participants in a user study entered text at 37.4 wpm (31.8 wpm for QWERTY) with an error rate of 3.8% (3.47% for QWERTY).

Figure 27. The KeyScretch keyboard [30] combines tapping with multi-letter gesture input on a popup menu.

The word-gestures of the *SHARK* [50, 145] input technique were adapted to use a QWERTY layout. The resulting method is called *ShapeWriter* [51]. Users are able to tap keys or draw a path starting from the first letter of a word and passing through each successive letter of the word. An empirical study compared two-thumb typing on a physical QWERTY keyboard ("thumb keyboard") with *ShapeWriter*. Participants used each technique for 40 minutes, in eight 5-minute blocks. Overall uncorrected error rate was identical for both at 1.1%. Entry rate was 27.7 wpm for thumb keyboard and 20.9 wpm for *ShapeWriter*. Some participants' *ShapeWriter* speed matched or surpassed thumb keyboard after 30 minutes of practice. This is especially interesting, considering that thumb keyboard input used both thumbs, but *ShapeWriter* input used only one stylus [51].

ShapeWriter was available for Android devices, and even iPhones [146]. However, Nuance Communications acquired *ShapeWriter* in 2010 and the technique is no longer officially available. Since then, other techniques have been released that mimic *ShapeWriter*'s input technique. These include *SlideIT* (www.mobiletextinput.com), *TouchPal Curve* (www.touchpal.com), and *Swype* (www.swypeinc.com). Nuance Communications acquired *Swype* in 2011, but unlike *ShapeWriter*, *Swype* is still commercially available.

Castellucci and MacKenzie [15] evaluated *Swype*, as well as one-finger typing on a QWERTY soft keyboard, two-thumb typing on a QWERTY soft keyboard, and a *Graffiti*-like technique, called *DioPen* (www.diodict.com). The entry speed for the one-finger and two-thumb QWERTY techniques were very similar, at 20.9 wpm and 20.8 wpm, respectively. *Swype* had an entry speed of 16.7 wpm and *DioPen* had only 7.0 wpm. *Swype* had the lowest total error rate at 7.0%. This was followed by one-finger QWERTY (7.1%), two-thumb QWERTY (13.8%), and *DioPen* (30.4%). Cuaresma and MacKenzie [21] evaluated *TouchPal Curve* and measured and entry speed of 35.3 wpm and a character-wise error rate of 5.4%. However, this evaluation was very brief, with participants entering only nine identical phrases.

Cuaresma and MacKenzie also evaluated the *Octopus* keyboard (ok.k3a.me), which mimics the proprietary keyboard on the BlackBerry *Z10*. With *Octopus*, users type on a QWERTY keyboard, but frequent words appear above some keys. Those words can be entered with an upward gesture on the respective key. Entry speed with *Octopus* started at about 25 wpm with the first phrase and rose to approximately 70 wpm by the ninth phrase [21]. However, the same phrase was entered each time and the *Octopus* word suggestions quickly adapted to the repetition. Consequently, participants were able to enter entire phrases with just a few short gestures. Still, this technique represents a merging of tapping and gesture input.

Although the *EdgeWrite* technique is not a tapping technique, its character gestures have been augmented with *Fisch* [139], which uses gesture suffixes to enter entire words. At the end of a gesture, but before character segmentation occurs, the user can draw a “pigtail loop” [139]. This loop indicates the end of character entry. The character is recognized, and four probable words are presented at each corner (Figure 28). The words are determined by the characters entered thus far, and a corpus. The user can enter a word by terminating the gesture in its corner. Ending the gesture in the center of the writing area enters only that character.

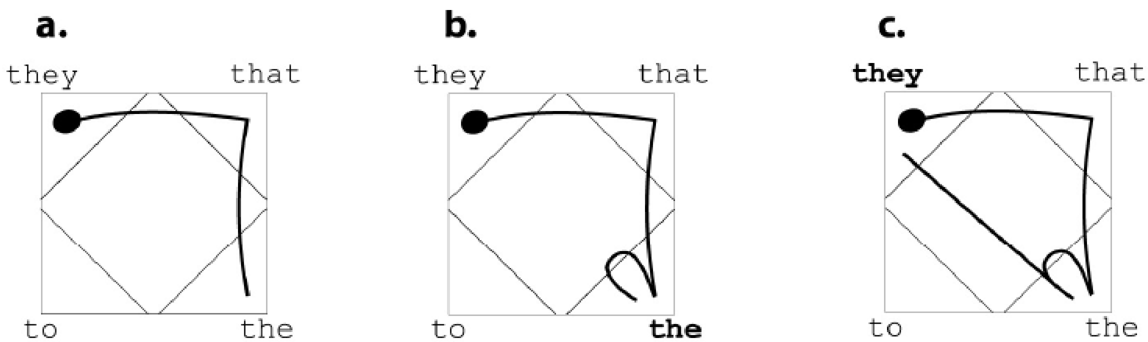


Figure 28. After entering “t” (a), a “pigtail loop” can select “the” (b) or “they” (c) [139].

Fisch was integrated with *Trackball EdgeWrite* [132]. A motor-impaired user averaged evaluated the technique and reached 12.09 wpm with *Fisch*, and 8.22 wpm without *Fisch*. His error rate was just under 4% in both conditions [132]. *Fisch* was also integrated with *EdgeWrite* in mobile phones using an isometric joystick [134]. A study compared character-level (*EdgeWrite* and *Multi-Tap*) and word-level (*EdgeWrite* with

Fisch and *T9*) text entry on a mobile phone. No statistically significant differences were found in either comparison [134].

2.5 Non-English Text Entry

Although this dissertation focuses on English text entry, it is also important to mention text entry in other languages. The prevalence of English keyboards (both desktop and mobile) has influenced writing in other alphabetic languages. When chatting or writing SMS messages in Arabic, Hebrew, or Indian languages, users typically write phonetically, from left-to-right, using Roman letters in the ASCII character set. If users wish to write in the original script on a mobile device, the letters of the corresponding alphabet are entered from right-to-left on a phone keypad or a soft keyboard. With the phone keypad, characters are entered using a multi-tap technique, similar to entering English text. With the soft keyboard, the characters overlay a QWERTY-like layout. In either case, text is often entered without vowels (Arabic and Hebrew) or vowel signs (Indian languages). Omitting the vowels completely is common even in non-electronic Arabic and Hebrew writing. If desired, vowels can be inserted using the input and a dictionary-based word completion technique, as used with English text entry [100]. For Indian languages, written text always includes the vowel signs. Based on the context of the input symbols, the vowels are inserted as needed [33]. Mobile text entry in English can also involve omitting the vowels from text (e.g., “tmrw” instead of “tomorrow”). However, this practice is informal and not universal.

1	2 ب ت ة ث	3 ا ء
4 س ش ص ض	5 د ذ ر ز	6 ج ح خ
7 ن ه و ي	8 ف ق ك ل م	9 ط ظ ع غ

Figure 29. A phone keypad for Arabic text entry [100].

Figure 30. A soft keyboard for Hebrew text entry on a PDA [100].

ँ ः 1	अ-ऊ 2 abc	ए-औ 3 def
क-ङ 4 ghi	च-ञ 5 jkl	ट-ण 6 mno
त- न 7 pqrs	प-म 8 tuv	य-ह 9 wxyz

Figure 31. A phone keypad used to enter text in Indian languages [33].

Chinese text entry is much more complex, as it is a logographic system that uses ideograms (i.e., symbolic characters) to communicate concepts. A single word can be composed of multiple characters, and each character can be composed of numerous strokes. In addition, Chinese characters are also used in Japanese and Korean writing. The early keyboards for Chinese text entry were huge, and allowed for direct, non-predictive entry of thousands of characters. Current keypad-based methods for mobile text entry assign a stroke to each key. The user enters the first four strokes (in order) for the desired character, followed by the last stroke for the character. A shape prediction algorithm then outputs the corresponding character, or a candidate list of matching characters [113].

Figure 32.A phone keypad for Chinese text entry [113].

Pinyin is an official system for entering Chinese characters using phonetics and Roman letters. It allows Chinese characters to be written using a QWERTY keyboard. While performing phonetic input would simplify the input of complex Chinese character, such a technique would likely not benefit English text entry, as the input sequence would be approximately the same length. However, with the proliferation of touchscreens, users are now able to draw characters on the screen and have them recognized as text input. A dissertation on mobile Chinese text entry was written by Liu [57] and examines these techniques in greater detail. The recognition techniques used for Chinese text entry might help improve English handwriting or word-gesture recognition, but it would still require a disambiguation technique and not help with discrete, deterministic text entry techniques.

2.6 Design Rationale for MIME

With such a variety of text entry methods, what form should an optimized, high-performance mobile text entry technique take? The *Graffiti* versus *Unistrokes* comparison by Castellucci and MacKenzie [12] demonstrates that gestures resembling corresponding handwritten characters are not necessarily easier to learn. The *Graffiti* gestures were so different from the participants' own handwriting that both gesture alphabets were equally novel. An evaluation of *DioPen* [15] shows even a gesture alphabet that accommodates multiple handwriting styles can hinder performance due to recognition errors. In an effort to reduce visual demand on the user, MacKenzie and Castellucci augmented the *Graffiti* input method with a corpus to automatically correct unrecognized and misrecognized gestures [62]. However, another revelation from the *DioPen* evaluation is that participants describe handwriting as too slow. Even without having to pause for the recognizer, human handwriting speed is estimated to be approximately 18-30 wpm [5 (p. 287), 22 (p. 61), 115, 143 (p. 196)], well below the theoretical upper limit of some tapping text entry techniques.

The *Unigest* technique [13] and a variation of *H4* [16] by Castellucci and MacKenzie, as well as the unpublished *TiltWriter* by MacKenzie and Castellucci all use mid-air gestures to enter text. *Unigest* maps gesture pairs to characters, *H4* codes are mapped to gestures, and *TiltWriter* uses the tile of a mobile device to select characters on a soft keyboard. Unfortunately, performing input with these techniques might be difficult in a mobile environment, such as on a bus. The motion of the bus (e.g., starting or stopping abruptly or turning sharply) could interfere with recognition of the user's intended gesture. An alternative would be to have a second sensor (e.g., accelerometer, gyroscope, etc.) on the user. This second sensor would measure the motion of the environment and its readings could be subtracted from the device's measurements to determine device-specific motions. Another alternative would be to use the device's front-facing camera to determine motion using computer vision techniques. However, both of these approaches might be too complicated and resource-intensive for mobile text entry.

Word path gestures (e.g., *ShapeWriter*) provide fluid input, but the complexity of the gestures might hinder performance. For example, the touch-based *H4* implementation by Castellucci and MacKenzie [16] performed poorer than tapping *H4* keys on a gamepad. Consequently, tapping on a soft keyboard seems like a preferred method for fast text entry. In particular, input does not require a corpus of dictionary words.

Although the *Opti* layout optimizes character arrangement, it localizes frequent characters at the centre of the keyboard. Users typically hold a mobile device and tap or swipe with one or two thumbs. The static position of the hand(s) along the side(s) of a device restricts the movement of the thumb(s) and makes some keys easier to tap than others. Optimizing a keyboard layout that accommodates this restriction might significantly improve mobile text entry performance. With practice, two-thumb entry would be superior, but situational restrictions might prevent the user from using two-thumbs (e.g., holding a cup of coffee in one hand).

To determine preferred key locations, participants in a user study would be asked to tap a highlighted key as quickly and accurately as possible. The task competition time for each key would be inversely proportional to its ease of selection. Using a language model, frequent letters would be mapped to keys that are easy to select. In a longitudinal study, one thumb text entry using this keyboard could be compared with using the established QWERTY layout.

Chapter 3
Gathering Text Entry Metrics on Android Devices

Mobile devices present countless opportunities for text entry research. Many users are accustomed to rapid, accurate touch typing on a spacious desktop keyboard while seated. While mobile devices are small, often lack a physical keyboard, and are used “on the go”, they have touchscreen digitizers, cameras, microphones, and motion sensors. These features allow for more text entry modalities then just pressing a key, and allow researchers to explore whether mobile text entry can approach desktop performance.

3.1 Motivation

The Android operating system by Google is very popular on mobile devices. As of 2013, over 1 billion Android devices have been activated [92]. Furthermore, 62% of tablets and 79% of smartphones sold in 2013 ran the Android OS [58, 97]. Android is currently the only popular mobile OS to allow third-party text entry methods[2]; anyone can easily develop and freely distribute an Android Input Method Editor (IME). IMEs can be used system-wide, without modifying installed applications, and can use any hardware resource on the device. For example, an IME could use the touchscreen to recognize handwriting (see Section 2.1), the camera for eye tracking [24], the microphone for voice recognition [147], or the motion sensors to determine device orientation (see Section 2.3).

[2] Apple’s iOS 8 will support third-party keyboards by the end of 2014 [84].

With so many IME options, it was necessary to develop an application to facilitate evaluation and comparison of existing and future IMEs – a mobile equivalent of *TextTest* [131] for the PC. To satisfy this need, Text Entry Metrics on Android (TEMA) was created.

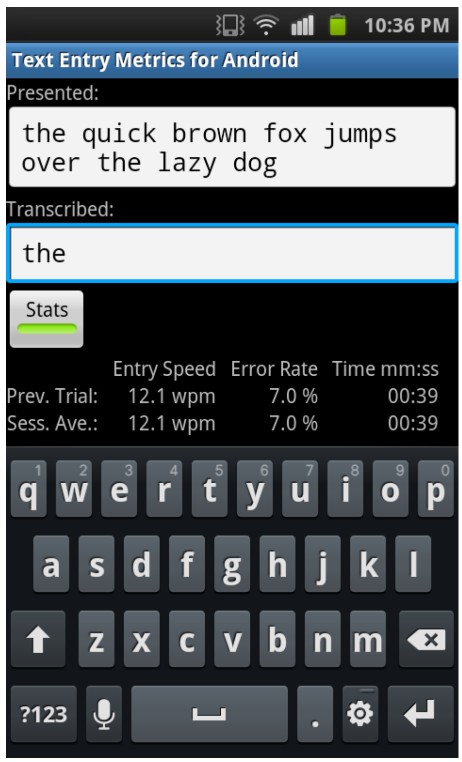


Figure 33. The TEMA application (above) is available at http://www.eecs.yorku.ca/~stevenc/tema/.

3.2 TEMA Features

TEMA is a small (less than 250 kB) ready-made application to aid researchers gathering text entry metrics on Android devices. It presents a phrase for the user to transcribe. Once transcribed, performance is calculated and logged, and another phrase is presented for transcription.

TEMA measures performance using established measurements for entry speed and accuracy. Entry speed is calculated by dividing the length of the transcribed text by the entry time (in seconds), multiplying by sixty (seconds in a minute), and dividing by five (the accepted word length, including spaces [143]). The result is reported in words-per-minute (wpm). Accuracy is evaluated according to the total error rate (TER), corrected error rate (CER), and uncorrected error rate (UER) metrics [107]. TER characterizes general input accuracy and is the sum total of CER and UER. CER reflects the errors that the participant corrected during transcription, while UER reflects the errors that the participant did not correct. All three error rates are reported as a percent. These performance measurements can appear on screen during text input to provide feedback to the user. This could be helpful if participants are instructed to reach a specific performance threshold. However, they are hidden by default to prevent distractions during typical evaluations.

TEMA records user actions in three logs: stats, events, and IME. They are saved to the device's internal storage and can be transferred to a PC via a USB or wireless connection. The logs are in tab-delimited format and can be opened by most spreadsheet applications. Each log begins with the date and time it was created, and ends with the date and time it was closed. The use of "[#]" in the logs is to easily identify comments intended for human consumption rather than for data analysis. The "stats" log summarizes entry speed, the accuracy metrics mentioned above, and intermediate measurements for each trial (i.e., phrase), with one trial per line. The "input time" is the time to transcribe the phrase. Timing begins with the first input: For a character based IME (e.g., QWERTY), this would be the first character, and for a word based IME (e.g., *Swype*), this would be the first word. Input timing ends with the input of the last transcribed character; it does not include the time to enter the terminating newline character (e.g., pressing the Enter key). Although interruptions are not recommended during evaluation sessions, TEMA measures the duration of interruptions (e.g., an incoming phone call, etc.) as "pause time"; it is not included in input time. The "total time" is the time from input of the first character to input of the terminating newline character, including pause time.

	A	B	C	D
1	[#]Log opened: Sun May 18 00:16:05 GMT 2014			
2	[#]presented	transcribed	presented_characters	transcribed_characters
3	the collapse of the Roman empire	the collapse of the Roman empier	32	32
4	[#]Log closed: Sun May 18 00:17:52 GMT 2014			
5				

E	F	G	H	I	J	K
input_time(sec)	pause_time(sec)	total_time(sec)	wpm	msd	numBksp	numDelChars
38.95	0	39.934	9.550706033	2	1	1

L	M	N
total_error	cor_error	uncor_error
0.090909091	0.03030303	0.060606061

Figure 34. An example of the stats log generated by TEMA.

TEMA also records the Minimum String Distance (MSD) [105] between the presented and transcribed phrases. The MSD value represents the minimum number of character operations (i.e., insertion, deletion, or substitution) required to convert the transcribed text to the presented one. The "numBksp" value represents the number of times the backspace button was triggered, while the “numDelChars” value represents the number of characters deleted during transcription. These values could be different when using IMEs that delete entire words with a single (long) press of a button.

	A	B	C	D	E
1	[#]Log opened: Sun May 18 00:16:05 GMT 2014				
2	[#]the collapse of the Roman empire				
3	0	t	pos@0		
4	730	h	pos@1		
5	1754	e	pos@2		
6	2547	<Sp>	pos@3		
7	4710	c	pos@4		
8	5891	o	pos@5		
9	6918	l	pos@6		
10	7321	l	pos@7		
11	8697	a	pos@8		
12	10303	p	pos@9		
13	10746	s	pos@10		
14	11470	e	pos@11		
15	12125	<Sp>	pos@12		
16	14073	o	pos@13		
17	14939	f	pos@14		
18	15677	<Sp>	pos@15		
19	16876	t	pos@16		
20	17687	e	pos@17		
21	18958	<Bksp>	pos@17		
22	20337	h	pos@17		
23	21311	e	pos@18		
24	22147	<Sp>	pos@19		
25	26180	R	pos@20		
26	27063	o	pos@21		
27	27890	m	pos@22		
28	28713	a	pos@23		
29	29450	n	pos@24		
30	30309	<Sp>	pos@25		
31	32281	e	pos@26		
32	33130	m	pos@27		
33	34589	p	pos@28		
34	36290	i	pos@29		
35	38104	e	pos@30		
36	38950	r	pos@31		
37	39735	<Entr>	pos@32		
38	[#]the collapse of the Roman empier				
39	[#]Log closed: Sun May 18 00:17:52 GMT 2014				

Figure 35. An example of the event log generated by TEMA.

The “event” log records low-level input. Each line contains the timestamp of the event, the input, and its position in the transcribed string. The timestamps allow for verification and analysis of events across logs. For example, the last character of the phrase was entered at 38 950 ms, which corresponds to an input time of 38.95 sec. In addition, the nearly 200 ms between the user pressing Enter and trial termination might give insight into device responsiveness. The block of events for a trial begins with a comment containing the presented phrase and ends with a comment containing the transcribed phrase.

The “IME” log records information sent directly from the IME being used. IME developers can include a provided Java file in their IME package to facilitate this functionality. The logged date could be high-level, such as new or technique-specific metrics, or very low-level, such as the screen location of individual touch events. In the following example the IME being evaluated uses vertical and horizontal gestures to trigger the Space, Shift, Backspace, and Enter keys. The developer is logging the triggered key and the x- and y-component of the gesture, in pixels. According to the log, it seems that the user’s gestures were quite straight, not angled. Again, the timestamps are synchronized across the logs. Thus, it might be interesting to note that events were logged from the IME, and then the corresponding character was sent to the text field 30-60 ms later.

	A	B	C	D	E
1	[#]Log opened: Sun May 18 00:16:05 GMT 2014				
2	[#]the collapse of the Roman empire				
3	2515	Space	263.3261	3	
4	12076	Space	377.6982	39.15094	
5	15623	Space	180	6	
6	18899	Bksp	121.4192	37.4032	
7	22115	Space	203.7564	48	
8	23478	Shift	15	107	
9	30262	Space	200.4553	0.891052	
10	39701	Enter	9	221.5792	
11	[#]Log closed: Sun May 18 00:17:52 GMT 2014				
12					

Figure 36. This IME log generated by TEMA contains the x- and y-components of the gestures used to enter the indicated character.

When TEMA is started, it presents an options dialog. Here, the user can specify experiment parameters. The participant number, session number, and technique code are used to name the log files. In the example below, the created log files would be named “1_2_A_stats.tema”, “1_2_A_events.tema”, and “1_2_A_ime.tema”. These values help easily identify logs, especially when there are many participants and/or sessions. It also simplifies sorting the log files as input for other programs or for later analysis.

From the dialog, the user can select the IME to use for the session. This is handy if the same participants are evaluating multiple IMEs in a session (e.g., a user study with within-subjects design). For security reasons, Android does not allow TEMA (or any application) to have direct access to an IME or for an IME to be selected programmatically. Doing so would allow malicious code to swap the default IME with an identical-looking one that logs and transmits passwords (for example). For the same reason, no program can alter IME options, such as auto-correct, audio feedback, or haptic feedback. Instead, TEMA can only trigger a separate dialog for the user to select the IME and configure options manually.

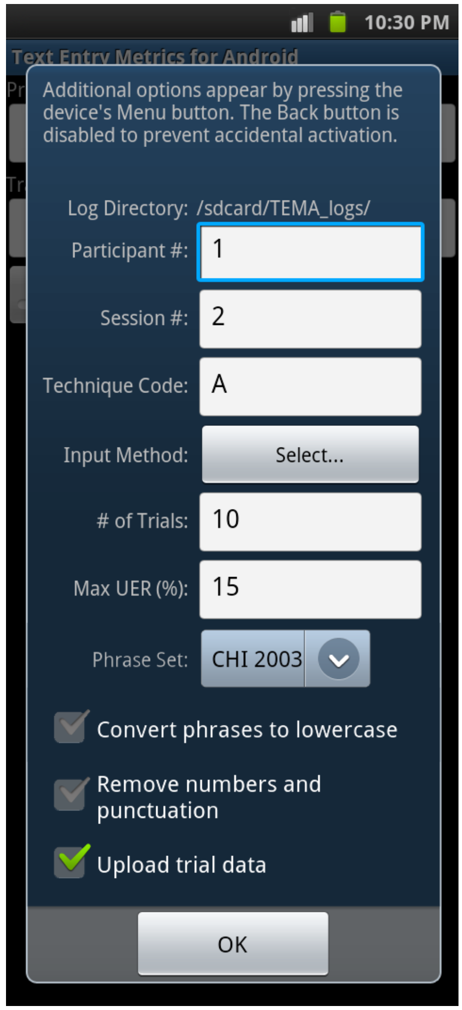


Figure 37. Users can specify study parameters in this dialog.

The user can specify the number of trials to be administered in this session and the maximum allowed UER. Sometimes, participants in a user study will rush to complete a session without paying enough attention to accuracy. This setting will reset a trial if its UER exceeds the set value. The trial results will still appear as a comment in the logs, but the trial counter will not be incremented. TEMA will also display a message to the participant encouraging increased attention to accuracy.

Presented phrases are taken from one of five phrase sets. Details about the phrase sets are presented in Chapter 5. Users also have the option of converting all letters to lowercase and/or removing numbers and punctuation from presented phrases. These options are useful when evaluating entry of only words or when evaluating input methods that provide input of a limited number of characters.

There is also the option to upload the session data to a server for logging. The uploaded data is similar to the contents of the stats log, but also includes the mobile device's unique identifier (UUID), the Android version, the display resolution and density, and the package name of the IME used. The contents of the events and IME logs are not uploaded. Currently, the server used is the Department's web server. However, the target Perl script can be copied to another server and its URL can be modified in the TEMA source code.

Once the session has begun, the user can reveal an options menu. The trial can be refreshed with a new phrase or reverted (i.e., reset). The Help option presents users with instructions to transcribe the presented phrase as quickly and accurately as they can. The Info option presents copyright information about TEMA. Finally, the Exit option allows the user to terminate the session prematurely.

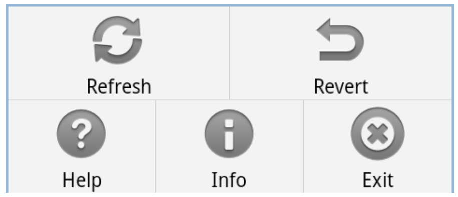


Figure 38. The menu provides additional options.

3.3 TEMA Design

Android's huge install base and acceptance of third-party IMEs means TEMA can be used to evaluate a vast number of IMEs, running on a variety of mobile devices and form factors. In particular, Android can change IMEs on-demand without requiring changes to (or even restarting) installed applications. Consequently, TEMA remains IME agnostic. It monitors the transcribed text field and records the time and position of character insertions and deletions.

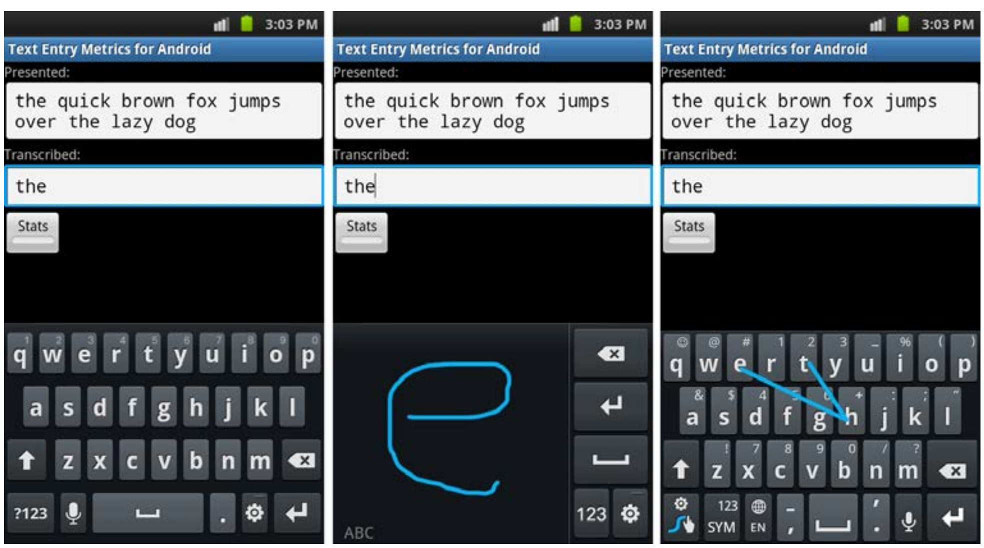


Figure 39. TEMA can be used to evaluate a variety of text entry methods.

Unfortunately, the strict separation between application and IME means that TEMA cannot directly communicate with just any preinstalled IME. Low-level, intermediate events that serve to input text (e.g., key presses to select a character for input or the path of a stylus gesture) are not available to TEMA for logging or analysis, but are often necessary to evaluate a text entry method. To address this deficiency, IME developers (i.e., ones with access to the IME's source code) can add the `TemaImeLogger` class to their package and use it to facilitate communication with TEMA.

```
...
public class TemaImeLogger
{
   ...

   /** Initializes this object. */
   public TemaImeLogger(Context c)
   {
      context = c;
   }

   /** Writes the passed String to TEMA's IME log. The string
    *  can contain multiple fields or represent a comment.
    *  Non-comments will be prefixed with a timestamp.
    *  @param s the String to write
    *  @param isComment if true, prefixes with [#]
    */
   public void writeToLog(String s, boolean isComment)
   {
      Intent i = new Intent(BROADCAST_TEMA);
      i.putExtra(KEY_1, s);
      i.putExtra(KEY_2, isComment);
      i.putExtra(KEY_3, System.currentTimeMillis());
      context.sendBroadcast(i);
   }
}
```

The TemaImeLogger class allows IME developers to send logging data to TEMA.

The class, which is provided to all TEMA users, implements a public method that bundles a string, a Boolean flag, and a timestamp, then broadcasts that data using Android's intent method of inter-process communication (IPC). The string can contain any textual data, including multiple, tab-delimited fields. The Boolean flag indicates whether or not the log entry should be treated as a comment. Finally, the timestamp represents the moment the event was logged. In the TEMA application a broadcast receiver is initialized to receive data from the `TemaImeLogger` class, decompose the bundled data, and write the transferred string to the IME log. More complex (e.g., bi-directional) communication could be implemented, but this simplicity ensures that logging does not unnecessarily burden the system, especially if the IME developer decides to log multiple events per character input.

```
...
intentFilter = new IntentFilter(BROADCAST_TEMA);
brdcstRec = new BroadcastReceiver()
{
   @Override
   public void onReceive(Context context, Intent intent)
   {
      String s = intent.getStringExtra(KEY_1);
      boolean isComment = intent.getBooleanExtra(KEY_2, false);
      long time = intent.getLongExtra(KEY_3,
         System.currentTimeMillis());
      String prefix = isComment ? "" : "" + (time - startTime) +
         log.DELIM;
      log.logIME(prefix + s, isComment);
   }
};
registerReceiver(brdcstRec, intentFilter);
...
```

Broadcasts sent to TEMA are logged. Timestamps are synchronized across all logs.

3.4 Encouraging Consistency in Mobile Text Entry Evaluations

The ability to evaluate text entry techniques in a consistent manner is very important. Such evaluations allow for meaningful comparisons between input methods and between studies. Unfortunately, some published user studies differ on how entry speed and accuracy are measured. Some [41, 74, 83] used the MSD metric [105]; some [21, 69, 73] used a character-wise metric; and some [36, 37, 78] reported the percentage of phrases entered correctly. Sometimes, accuracy measures are omitted completely [32, 119, 127]. One of TEMA's goals is to facilitate the consistent gathering and reporting of established entry speed and accuracy metrics.

This methodology has already been adopted by numerous researchers in both academia and industry: Amanda Smith used TEMA in her dissertation to evaluate and compare how young and old adults use smartphones [104]; Anju Thapa used TEMA in her thesis to compare novice mobile text entry performance using *MessagEase* and QWERTY [116]; advisors, such as Poika Isokoski, Erno Makinen, Janet Read, and Robert Teather, are using TEMA in text entry research projects with their graduate students; and Curtis Ray, Vice-President of Engineering at Tactus Technology Inc. is interested in using TEMA to evaluate the *Tactus Keyboard* – a touchscreen display that morphs into physical keys for text entry [112]. Discussions are on-going. Representatives from Motorola and Sprint were also interested in using TEMA to evaluate and market mobile phones. They were each very impressed with TEMA, but no mutually beneficial agreement was reached with their respective accounting departments regarding licensing. For a full list of TEMA users, see Appendix B.

A consistent methodology for mobile text entry evaluation should be adopted and TEMA facilitates this. In situations where an established convention might not suffice (e.g., non-prose or non-alphabetical text entry), researchers should detail how metrics were calculated, so that their methods are reproducible.

3.5 Method

A user study was conducted to demonstrate TEMA's utility and establish entry speed and accuracy measurements for the evaluated techniques.

3.5.1 Participants

Sixteen paid participants (ten male, six female) were recruited from the local university campus. Ages ranged from 18 to 31 years (mean = 23; SD = 3.53). Two participants were left-handed. Although participants were familiar with the QWERTY layout, none was an expert in onscreen QWERTY keypads, handwriting, or word-gesture techniques. Therefore, the results are characteristic of novice, not expert, performance.

3.5.2 Apparatus

The TEMA application ran on a Samsung *Galaxy S Vibrant* (GT-I9000M) smartphone running Android 2.1. The touchscreen measured 4.0 inches diagonally and had a resolution of 480×800 pixels. The phone was held in portrait orientation throughout the study. The phone's wireless radios were disabled to eliminate disruptions due to incoming calls or text messages.

Three of the IMEs included with the phone were evaluated with TEMA: the default QWERTY keypad, *DioPen* (handwriting, Figure 39, centre), and *Swype* (word-gesture, Figure 39, right). For each IME, the input language was set to English (US) and options for auto-spacing, auto-capitalization, and word prediction were deactivated. All other options were kept at default values.

3.5.3 Procedure

Participants entered ten phrases in each condition. The phrases were chosen randomly from the "CHI 2003" phrase set [65]. They were instructed to enter text as quickly as possible, to correct errors if noticed immediately, but to ignore errors made two or more characters back. This was to prevent deletion of many correct characters to correct an early mistake, which would unnecessarily increase the measured CER. A study by Arif and Stuerzlinger [3] showed that recommending or requiring error correction significantly increased TER, but the choice of error correction strategy did not significantly affect entry speed.

In the "QWERTY-thumbs" condition, the phone was held with two hands and participants typed with both thumbs (Figure 40, left). In the *DioPen*, *Swype*, and QWERTY-finger conditions, participants held the device in their non-dominant hand and used a finger on their dominant hand to perform input (Figure 40, right). Before each condition, participants were instructed on how to use the corresponding technique. For the *DioPen* condition, a chart with the gesture alphabet[3] was provided. A practice

---

[3] http://help.diotek.com/data/diopen/android/10/page42.html

session followed, consisting of three random phrases. Study sessions typically lasted 50 minutes and took place in a quiet office, with participants seated at a desk.

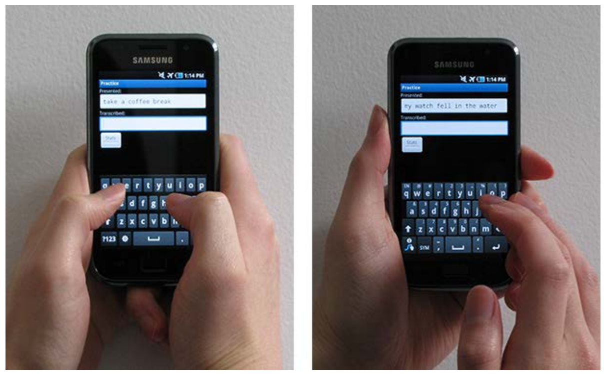


Figure 40. The above images demonstrate participants' hand positions during the study conditions.

3.5.4 Design

The experiment employed a within-subjects factor, technique, with four levels: QWERTY-thumbs, QWERTY-finger, *DioPen*, and *Swype*. The two-thumb QWERTY input condition encapsulates a popular method of mobile text entry. The single-finger QWERTY condition represents an alternative QWERTY input method and allows comparisons with the single-finger handwriting and word-gesture input techniques.

The order of testing was counterbalanced using a balanced Latin Square. The dependent variables were entry speed and accuracy. They were measured by TEMA (as detailed previously) and averaged over the ten phrases.

3.6 Results and Discussion

3.6.1 Accuracy

The TER of *Swype* was the lowest, at 7.0% (Figure 41). Interestingly, an evaluation of *ShapeWriter* on a tablet PC revealed a similar TER value of 6.7% [51 (pp. 65-66)]. The TER of the QWERTY-finger condition was slightly higher at 7.1%. Surprisingly, the QWERTY-thumbs condition was almost double that, at 13.8%. This is considerably greater than the 10.4% TER measured using two thumbs on the *iPhone*'s QWERTY keypad [1]. *DioPen* had the worst TER, at 30.4%. In comparison, a *Graffiti* study revealed an error rate of only 19.4% [46].

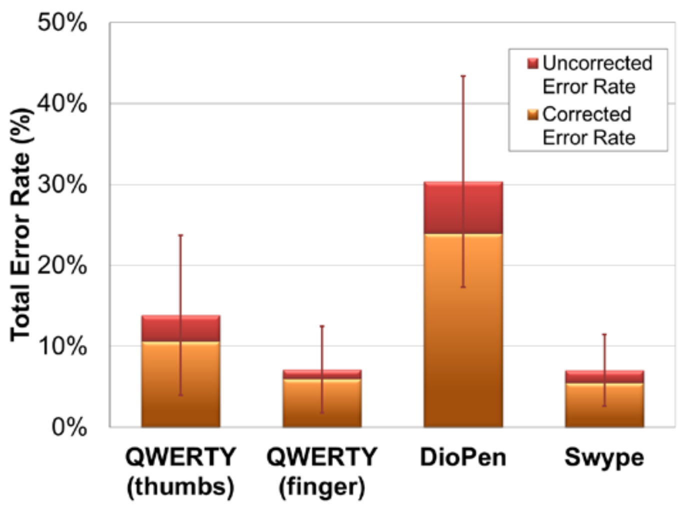


Figure 41. Accuracy values gathered by TEMA. Error bars represent ±1 standard deviation of TER. A box plot representation appears in Appendix F.

An Analysis of Variance (ANOVA) revealed a significant effect of technique on TER ($F_{3,36} = 41.66$, $p < .0001$). However, Scheffé post hoc analysis indicated a significant difference only between *DioPen* and all other conditions. Tukey, LSD, and

Bonferroni post hoc tests also indicated significant difference between QWERTY-thumbs condition and all other conditions. Condition order had no significant effect on TER ($F_{3,12} = 0.83$, ns).

DioPen's UER of 6.4% indicates participants missed (or ignored) many errors. The corresponding event logs revealed multiple attempts to enter characters (i.e., participants entered an incorrect character, backspaced, entered the same incorrect character, backspaced, etc.). This suggests participants could not reliably draw the required gestures. Considering the *DioPen* gesture alphabet, the errors generally fall under three categories: incomplete loops (e.g., "c" inputted instead of "o"), incorrect proportions (e.g., "h" or "r" inputted instead of "n"), and poor timing (e.g., "l." inputted instead of "i"). The frequency of these errors would likely decrease with practice, as users perfect the accuracy of their gestures. Alternatively, research by Arif and Stuerzlinger [4] suggests that, over time, users would likely switch to alternate input gestures if available (e.g., inputting gestures that resemble cursive script, instead of printing). However, in this study, participants typically deleted the incorrect character and performed input again. Most participants were frustrated by *DioPen*'s unreliable input. One participant mentioned that *DioPen* was difficult to use because its gesture alphabet did not resemble his own handwriting. Another participant stated, "It's just easier to type [rather than write]."

3.6.2 Entry Speed

The QWERTY-finger entry rate of 20.9 wpm is the fastest in this study (Figure 42). The QWERTY-thumbs entry rate was just slightly lower at 20.8 wpm. Both values exceed the 15.9 wpm reported for two-thumb text entry on the *iPhone*'s QWERTY keypad [1]. *DioPen* was slowest at 7.0 wpm. This is probably due to the high gesture misrecognition. A *Graffiti* study yielded a rate of 9.2 wpm [46]. The *Swype* entry speed of 16.7 wpm is consistent with a *ShapeWriter* study that reported 15 wpm [51 (pp. 65-66)].

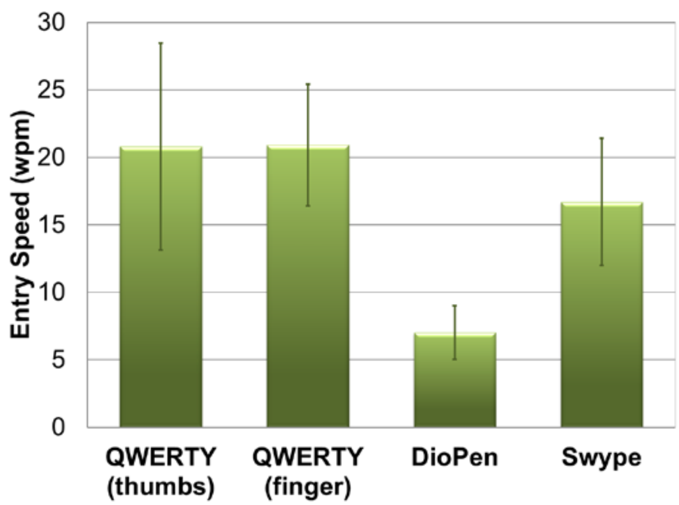


Figure 42. Entry speed values gathered by TEMA. Error bars represent ±1 standard deviation. A box plot representation appears in Appendix F.

There was a significant effect of technique on entry speed ($F_{3,36} = 71.17$, $p < .0001$). However, there was no significant difference between the two QWERTY conditions. This is surprising, as many believe two-thumb input to be a faster method of text entry. This study focused on novice performance. Perhaps expert users learn to better coordinate input with two thumbs, resulting in faster input. Every other pairwise comparison of techniques satisfied the 5% threshold for significance. Again, counterbalancing proved effective ($F_{3,12} = 2.34$, $p > .05$).

3.7 Conclusion

The conducted study demonstrated TEMA's utility. Despite the perceived advantage of two-thumb input, there was no statistically significant difference between the two QWERTY conditions with respect to novice entry speed. Word-gesture input was slightly slower, but not significantly less accurate. Handwriting was both slow and error-prone.

In addition to gathering performance metrics, TEMA's logs helped identify the source of participants' handwriting errors. By examining the event logs, researchers were able to determine that the high occurrence of character deletions was a symptom of spatially and temporally malformed gestures. If the IME developer had used the `TemaImeLogger` class to record each gesture's sample points, further analysis could quantify the deviation of the erroneous input gestures from the accepted gestures for each problematic character.

TEMA will aid in the evaluation of the MIME keyboard. It provides a consistent platform for mobile text entry research on Android devices, includes thousands of phrases for text entry, measures timings, calculates performance metrics, and generates easily viewable log files for post-study analysis. Furthermore, it has been recognized by corporations in industry and is used by researchers in academia. TEMA may be downloaded from the following URL: http://www.eecs.yorku.ca/~stevenc/tema/.

Chapter 4
Determining Feedback Preferences for Mobile Text Entry

This chapter aims to answer three questions regarding aural and haptic feedback options during text entry: 1) What feedback (or combination of feedback) do users prefer and why? 2) Does the type of feedback affect users' performance? 3) Does the type of feedback affect users' perception of performance?

This chapter first summarizes other research related to aural and haptic feedback during text entry. Then, the survey and user study used to investigate the above questions are detailed. Finally, the results are presented and discussed.

4.1 Motivation

Many mobile devices use touchscreens and soft keyboards instead of physical keyboards. This allows for a larger display without increasing the size of the device. Furthermore, soft keyboards change their layout based on user input and disappear when not needed. To compensate for the lack of tactile feedback provided by physical keys, soft keyboards can include aural and haptic feedback. The feedback takes the form of audible clicks from a speaker and device vibration, respectively. However, it is important to investigate how feedback will affect users. This is especially true for a commercial product, where success depends on user acceptance. The use of feedback might annoy users, or cause decreased performance.

4.2 Related Work

Existing research has investigated the effect of haptic feedback on text entry performance. Some use vibration to indicate key presses, erroneous input, or to alert the user to input options. Koskinen et al. [49] evaluated the effect of various forms of haptic feedback when entering numbers via a soft keypad. Although the effect was not statistically significant, participants found vibrations of 16 ms the most pleasant. However, the authors state that preferences are not necessarily generalizable and might vary between devices. This might explain why other studies evaluating haptic feedback yield conflicting results.

Dunlop and Taylor [23] used a 75 ms vibration to indicate “helpful” word completions during text entry and a 150 ms vibration to signal entry of a non-dictionary word. The feedback significantly improved entry speed by 3 wpm.

McAdam and Brewster [78] also found that haptic feedback significantly benefitted entry speed. A vibration of 30 ms signaled a correct key press, while 500 ms signaled a key slip. The vibrations were delivered to one of six locations on the participant, with the upper arm and wrist performing the best. They did not find any significant effect of vibration on accuracy.

Brewster et al. [8] used a “smooth” vibration to indicate correct input and a “rough” one to signal errors. Both were 800 ms in duration. Though this gave participants a perceived increase in performance, the feedback had no significant effect on entry speed or total error rate. However, it significantly improved accuracy in the form of fewer uncorrected errors.

Hoggan et al. [36] used 30 ms and 500 ms vibrations to signal correct input and errors, respectively. They found a significant effect on both speed and accuracy. Furthermore, Hoggan et al. [37] used both audio and haptic feedback individually in noisy and moving environments and found they each improved speed and accuracy over the condition with no feedback. The effect of each mode depended on the environment. Haptic feedback improved performance in noisy environments, while audio was better in high vibration environments.

Mobile devices use less sophisticated haptic actuators than those used in the aforementioned research. This is perhaps to minimized size, weight, or cost. Thus, evaluating the effect of haptic feedback using an actual mobile device is valuable.

4.3 Method 1 (Survey)

Mobile users were polled on their feedback preferences when typing on touchscreen devices. To reach a large sample of users, the following question was posted on various online forums that cater to mobile technology:

> *Smartphones allow feedback when typing. This feedback could be audio (e.g., a "tick" sound from the speaker), vibration (i.e., the device shakes a little), or a combination of the two. What feedback do you prefer when typing (e.g., texting, emailing, etc.)?*

Participants were able to select only one of the following responses: "Audio", "Vibration", "Audio and Vibration", or "None". They were also allowed to post comments elaborating on their choice. Although the context of text entry was not specified, it is believed that users resist changing their audio or haptic feedback settings based on their environment or situation – they "set it and forget it". This is often demonstrated by phones that ring during movies or lectures.

4.4 Method 2 (User Study)

In addition to the survey, a user study was conducted to determine the effect of a combination of feedback modes on mobile text entry performance.

4.4.1 Participants

Twelve participants (two females, ten males) with an age range of 20 to 31 years (mean = 26; SD = 4.1) entered text on a mobile phone. The number of participants is consistent with related studies [8, 36, 37]. All participants were fluent in English and frequently typed on a touchscreen device.

4.4.2 Apparatus

The phone used for the study was a Samsung *Galaxy S Vibrant* (GT-I9000M), running Android 2.3.3. The touchscreen measured 4.0 inches diagonally and had a resolution of 480×800 pixels. The audio feedback was the default key "click" sound, as defined by `AudioManager` in the Android API. The phone's volume was set to provide feedback that was clearly audible, but not intense. An audio recording of the phone's haptic feedback was created and analyzed. The vibration was measured to be about 80 ms in duration.

For text entry, the default QWERTY keyboard was used with auto-spacing and auto-correction options disabled. TEMA was used to administer phrases from the CHI 2003 phrase set, record participant input, and calculate text entry metrics.

4.4.3 Procedure

Participants entered 30 phrases in each condition. However, the first 5 phrases served as a warm-up and were not included in the analysis. To eliminate variability in the task, all participants held the phone in a portrait orientation and entered text using their thumbs. Furthermore, they were instructed to enter text as quickly as possible, to correct errors if noticed immediately, but to ignore errors initially missed (i.e., to prevent deletion of many correct characters to correct an early mistake).

Study sessions typically lasted 30 minutes and took place in a quiet office, with participants seated at a desk. Participants also completed a questionnaire to elicit their text entry preferences and to gather demographic information.

4.4.4 Design

The study employed a within-subjects factor, feedback mode, with four levels: Audio, Vibration, Both (audio and vibration), and None. The order of testing was counterbalanced using a balanced Latin Square. Each participant entered 30 phrases (5 warm-up, 25 experimental) in each condition, which is consistent with previous text entry research [36, 78]. Analysis was based on the resulting 1200 (12×25×4) trials.

The dependent variables were entry speed and accuracy, as calculated by TEMA. Entry speed was reported in words-per-minute and accuracy was measured according TER, CER, and UER metrics.

4.5 Results and Discussion

4.5.1 Survey Results

The results of the survey appear in Figure 43 and include responses from participants in the user study. A total of 92 people cast a vote indicating their preferred feedback mode when typing on a mobile touchscreen device. While just over one third of respondents opt for only haptic feedback, almost half prefer no aural or haptic feedback at all. The margin of error is 9.6% with a confidence level of 95%.

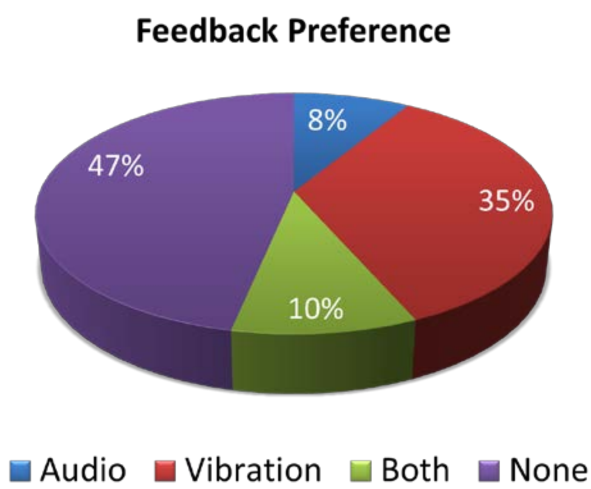


Figure 43. Survey participants' feedback preference when typing on a mobile touchscreen device (n = 92).

4.5.2 *Entry Speed and Accuracy*

Entry speeds for the Audio and None conditions were identical at 29.9 wpm, with the Both condition being slightly higher at 30.3 wpm and Vibration being slightly lower at 28.7 wpm (Figure 44). Dunlop and Taylor [23] used a 12 key phone keypad for input and recorded a speed of 23 wpm when using vibration. However, McAdam and Brewster [78] and Hoggan et al. [37] both used touchscreen keyboards and reported speeds of approximately 30 wpm, consistent with these results. Unfortunately, the other studies measured entry speed in "time to enter phrases" or "number of lines entered", thus preventing direct comparisons.

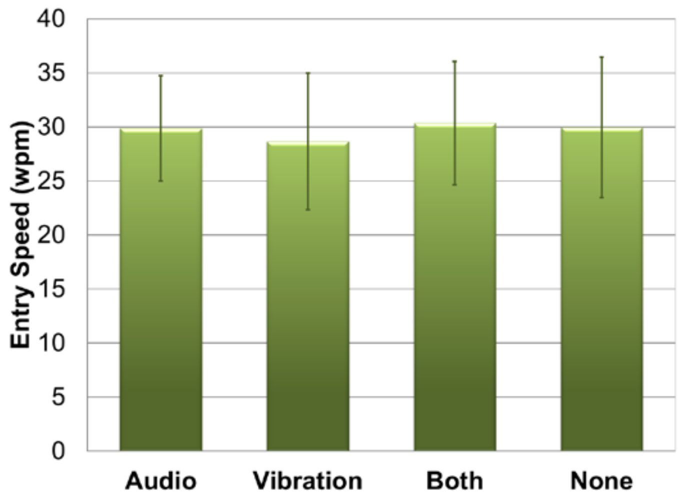


Figure 44. Entry speed values gathered from this user study. Error bars represent ±1 SD. A box plot representation appears in Appendix F.

The difference in entry speed between the four conditions was not statistically significant ($F_{3,24} = 1.25$, $p > .05$). This is consistent with the findings of Brewster et al. [8], but differs from the findings of Dunlop and Taylor [23] and McAdam and Brewster

[78]. In addition, the ANOVA indicates that counterbalancing worked, as the order of the conditions was not significant ($F_{3,8} = 1.53$, $p > .05$).

Accuracy results appear in Figure 45. The None condition was the most accurate, as it yielded the lowest CER (7.0%) and TER (9.7%), respectively. Participants committed (and corrected) more errors in the conditions that provided feedback. Evidently haptic feedback motivated participants to correct their errors. The Vibration condition had the highest CER (8.1%) and the lowest UER (2.1%). Surprisingly, the combination of haptic and aural feedback resulted in the highest UER (3.3%) and TER (10.7%). Participants committed the most errors in the Both condition and did not correct them. Unfortunately, the effect of feedback was not statistically significant for TER ($F_{3,24} = 0.69$, ns), CER ($F_{3,24} = 0.94$, ns), or UER ($F_{3,24} = 1.15$, $p > .05$). As with entry speed, the group effect on accuracy was not significant ($F_{3,8} = 3.83$, $p > .05$).

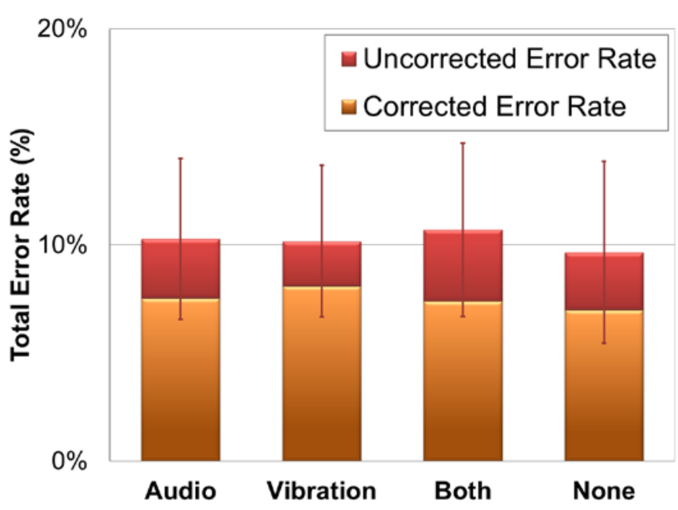


Figure 45. Accuracy values gathered from this user study. Error bars represent ±1 SD of TER. A box plot representation appears in Appendix F.

Unfortunately, the use of different accuracy metrics in related studies prevents accurate comparison with these results. One study [8] measured accuracy as "total errors" and "number of errors uncorrected", suggesting analogs to TER and UER metrics, respectively. However, the accuracy measurements appear on the same chart as entry speed, with an "average score" on the *y*-axis rather than the expected error rate.

Other studies reported the number of phrases entered correctly. Unfortunately, this metric does not convey how many errors appeared in incorrect phrases, nor the number of errors corrected during input. McAdam and Brewster [78] reported 75% to 80% of phrases were entered correctly, with vibration having no statistically significant effect on accuracy. In comparison Hoggan et al. [36, 37] reported accuracy rates from 55% to 90% and found that feedback had a significant effect on accuracy; vibration improved accuracy in noisy environments, but audio was better in high vibration environments.

4.5.3 Users' Perception of Performance

After the study sessions, participants were asked to select the feedback mode they felt resulted in the fastest typing and which resulted in the most accurate typing. This was to investigate whether or not feedback mode had any effect on perceived performance.

The majority of participants' selections were evenly split between the Audio and None conditions for both speed and accuracy. However, the results show that most participants typed fastest in the None condition, but typed most accurately in the Vibration condition. Half the participants correctly identified the fastest feedback mode, while only a quarter of participants correctly identified the most accurate feedback mode. Thus, there was a significant divide between actual and perceived performance.

4.5.4 Users' Preferences

What contributes to users' preference for one feedback mode over another? Comments received by survey respondents provide insight and are summarized below. (Respondents' usernames appear in parentheses.)

University instructors understand that some mobile phone users mute their devices *in an attempt* to hide text entry activities during lecture. However, survey comments suggest that, in social settings, an ethic of reciprocity might also influence the preference for no audio feedback. Some users are bothered by other people's noisy devices. Thus, they choose to disable audio feedback on their own device to not disturb people around them.

> *"Audio feedback annoys me a little when using it, and it annoys me A LOT when the person next to me is using it!" (Big Ang)*

> *"i [sic] prefer silence, no audio, no vibration, because audio will influence other people, while vibration will make me uncomfortable." (jean2012)*

Seven respondents stated specifically that the sound clips used for audio feedback in mobile devices are "annoying". Others commented that the audio feedback seems unnatural.

> *"I can't stand that fakey clicking sound." (synaesthetic)*

> *"The audio feedback is often annoying; this isn't the nineteenth century anymore [in reference to typewriters?], and often the noises devices choose are silly." (primetechv2)*

While most respondents dislike aural feedback, many appreciate having haptic feedback to indicate that input to the mobile device was received.

> *"I activate the haptic feedback, because it give [sic] me a sense that the phone is really typing." (Felimenta97)*

Finally, one respondent turns off audio and vibration feedback in an effort to conserve battery power.

> *"I prefer no audible or haptic feedback what so ever. To me, they are pointless and help eat battery life that I can be better used for programs I use." (moonzbabysh)*

4.6 Conclusion

In this user study, feedback mode had no significant effect on typing speed or accuracy. Some related studies conclude that feedback significantly effects speed, but not accuracy, while other studies that use similar feedback conclude the opposite. These results highlight the disagreement on the effect of feedback on performance and raise the question, "Why do users prefer one feedback over another?"

To that end, this chapter also provides insight into mobile users' feedback preferences. Almost half of users surveyed prefer no aural or haptic feedback during text entry. Thus, the cost to create, evaluate, and deploy new feedback techniques might outweigh the benefit if few users adopt it. Survey comments indicated that other issues, such as power consumption and social etiquette, also influence preferences. To cater to user preferences, MIME will be designed with options to provide haptic and audio feedback, but not with the expectation that these features will necessarily affect text entry performance.

Chapter 5
Compiling Phrase Sets for Mobile Text Entry

When evaluating a text entry method (mobile or otherwise), participants in a user study use the method to enter presented text from a phrase set. While copying presented text is not a typical real-world scenario, it serves three important purposes: 1) the duration to compose text is eliminated from the entry speed measurement; 2) input errors are easy to identify and quantify; and 3) the phrases can be selected to reflect specific, real-world input [67 (p. 81)]. This last point emphasizes the importance of selecting phrases from a corpus (i.e., a body of text used for analysis) that represents typical usage. This chapter details the development of four phrase sets from three corpora that characterize mobile text input.

5.1 Motivation

Twenty years ago, SMS text messaging was just becoming popular, with an estimated 3 million messages being sent annually worldwide [125], and text entry was limited to the 12-key telephone keypad on mobile phones. The introduction of smartphones and the availability of mobile Internet access has led to mobile users also writing emails, engaging in social networking, and using other forms of instant messaging. Today, analysts estimate the annual worldwide number of SMS messages sent to be 6.5 trillion and the number of over-the-top (OTT) instant messages sent over the Internet (e.g., using applications like *WhatsApp*[4]) to be 10.3 trillion [99]. In addition, Facebook has 1.01 billion active mobile users monthly [25] and Twitter has 198.9 million active mobile users monthly [118].

Research by Lyons and Clawson [59] and Kristensson and Vertanen [52] compare the effect of phrase sets on text entry speed. They found only subtle differences in performance, but the authors emphasized that phrase sets should be "representative of the text that end-users are likely to write" [52]. While email communications typically have a grammatically correct, business-like tone, text messages and tweets are very informal and contain many special characters. The established CHI 2003 phrase set for text entry research [65] does not capture such diversity.

5.2 Corpora Sources

The following corpora were selected to represent the popular mobile tasks of emailing, texting, and tweeting:

EnronSent Corpus [111]: This is a corpus representing over 96 000 email messages sent by Enron employees and made public by the United States Federal Energy Regulatory Commission during their investigation of Enron. It contains approximately 13.8 million words.

[4] http://www.whatsapp.com/

NUS SMS Corpus v2012.04.30 [19]: This corpus represents over 51 000 English SMS messages sent by volunteers at the National University of Singapore. Personal information, such as names, emails, and phone numbers were removed to protect the privacy of those involved.

Illocution 10% Twitter Corpus v.2013[5]: Illocution provides a free sub-sample of their *Twitter Stratified Random Sample* corpus. This version contains approximately 1.2 million English tweets made in 2013. There was a Twitter Corpus [91] published in 2010. It comprised 97 million tweets, occupying 14 GB of text. However, the authors are no longer distributing it, due to a request from Twitter.

5.3 Digram Frequencies

The twelve most frequent digrams in each of the corpora are listed in Table 2. The relative ordering of many digrams (e.g., "e·", "t·", and "·t") is consistent between the corpora, but some digrams are far more popular with specific text entry tasks. The double-space is the most popular digram in the email corpus, ranks 128th in the SMS corpus, and does not appear at all in the Twitter corpus. The digram "·@" ranks 7th in the Twitter corpus, 891st in the SMS corpus, and 1311th in the email corpus. The digram

---

[5] http://www.illocutioninc.com/site/products-data.html

“..” ranks 12th in the SMS corpus, 79th in the Twitter corpus, and 267th in the email corpus.

Table 2. The twelve most frequent digrams in each of the corpora. The “·” character is used to represent the space character.

Email	
··	4.57%
e·	2.48%
·t	1.88%
th	1.43%
s·	1.37%
·a	1.37%
t·	1.27%
in	1.20%
he	1.15%
d·	1.07%
on	1.02%
an	1.01%

SMS	
e·	2.36%
t·	1.84%
·t	1.73%
.·	1.63%
in	1.24%
·a	1.20%
·s	1.11%
ha	1.10%
s·	1.09%
o·	1.07%
n·	1.07%
..	1.04%

Twitter	
e·	1.71%
t·	1.13%
s·	1.06%
·t	1.05%
in	1.01%
er	0.87%
·@	0.82%
th	0.81%
an	0.80%
·a	0.78%
he	0.72%
n·	0.71%

To create a corpus representative of mobile text entry, it is not sufficient to simply identify mobile text entry tasks (i.e., email, texting, social networking). One must also estimate the proportion of one’s time allocated to each task. A marketing report [27] serves to construct a text entry distribution (TED). It estimates that text entry tasks are divided as follows: 44% texting, 36% social networking, and 20% email. Given this TED, the digram frequencies of the three corpora were merged to calculate the digram and letter frequencies for the mobile corpus. These are summarized in Table 3.

Table 3. The twelve most frequent digrams and characters in the Mobile corpus. The "·" character is used to represent the space character.

Mobile		Mobile (ignore case, space)		Mobile (character frequency)	
e·	2.16%	in	1.22%	·	15.97%
·t	1.52%	th	1.19%	e	7.74%
t·	1.48%	he	1.04%	t	6.26%
in	1.16%	ha	0.98%	a	6.24%
s·	1.14%	an	0.95%	o	6.05%
·a	1.09%	er	0.88%	i	5.00%
.·	1.07%	re	0.84%	n	4.76%
th	1.03%	on	0.79%	s	4.22%
··	1.00%	ou	0.74%	r	3.97%
he	0.95%	at	0.73%	h	3.75%
n·	0.93%	to	0.65%	l	3.34%
an	0.89%	ng	0.65%	u	2.42%

These values will aid in the development of MIME. The digram frequencies will be used to calculate the upper bound entry speed of any prospective layout. Because shift and space functionality will be mapped to gestures, it is also necessary to determine the most frequent digrams (ignoring letter case) that do not contain the space character. This data will determine which letter pairs have priority in the character arrangement process. Character frequencies will also be used to determine prospective keyboard layouts using an alternative process.

5.4 Selection of Phrase Sets

For the email, SMS, and Twitter corpora, wrapped lines were concatenated, phrases were extracted (one per line), and metadata (e.g., message date, time, and location) was discarded. To achieve an average phrase length close to that of the CHI 2003 phrase set (28.6 characters), phrases longer than 35 characters were discarded. In addition, non-English phrases were discarded. This was accomplished by filtering out any phrase that contained characters not available on a standard US QWERTY keyboard.

The phrases were also checked for inappropriate content (e.g., profanities, sexual content). The phrases were filtered against a list[6] of "bad words" provided to web administrators to filter content. However, several iterations of filtering were required. After each iteration, the filtered phrases would be manually scanned for inappropriate content. Words deemed inappropriate would be added to the list for the next iteration. A subsequent, manual scan was then performed to remove any remaining non-English phrases and to remove phrases with personal information (e.g., "Amy's number is 415-555-1234"). In addition, personal Twitter handles (i.e., usernames) were truncated to protect the person's online identity. As a result of this filtering, each corpus was reduced to 700-1000 phrases.

For each corpus, tens of thousands of 500-phrase sets were randomly generated and analyzed for their letter frequency distribution. Again, that value was chosen to mimic the CHI 2003 phrase set. The set with the highest correlation with the original, unfiltered corpus was selected as its representative. All three of the resulting phrase sets are highly correlated with the original corpora, with coefficients greater than .98. In addition, the average length of a phrase ranges from 28 to 30 characters. For more details of the phrase sets and a sample of the phrases, see Appendix C.

---

[6] http://code.google.com/p/badwordslist/downloads/detail?name=badwords.txt

The Enron corpus has been previously used to generate phrase sets. Paek and Hsu [85] created phrase sets from Enron, Facebook, and Twitter sources, but only the Enron-based phrase set was made publicly available. Although it too contains 500 phrases, they all contain just four words[7] each. Vertanen and Kristensson [120] also used the Enron corpus. Like the email phrase set described in this chapter, phrases were chosen to be representative of the corpus's digram frequency. However, the set contains fewer entries, totalling 320 phrases.

The Email, SMS, and Twitter phrase sets described in this chapter were added to TEMA, joining the existing CHI 2003 entry. These new sets provide phrases for transcription that include the punctuation, numerals, and special characters typically used in mobile text entry scenarios. The Mobile "phrase set" is also an option, but is generated dynamically. Phrases are selected from the other three mobile text entry phrase sets with the weighted probabilities in the TED. This allows future updates to TEMA to adjust the percentages within the TED to reflect changing mobile usage habits.

5.5 Conclusion

In addition to augmenting the features of TEMA, the generation of the mobile corpus has the added benefit of providing digram frequency data for the development of MIME. The data summarized in this chapter will be used to determine an optimized character

[7] Here, a "word" represents a tokenized string of characters which vary in length, not the accepted definition of five characters including spaces.

arrangement. The inclusion of typical punctuation, numerals, and special characters is important, as the placement of those characters (as well as the letters of the alphabet) will be optimized.

Chapter 6
Evaluating One-Handed Target Selection Times on a Mobile Touchscreen

In order to optimize mobile text entry, it is important to capture both the mental and physical components of the task. The previous chapter provides phrase sets that model text composition. This chapter provides a model of the motor actions performed.

6.1 Motivation

Grasping a mobile device anchors the position of the user's hand(s). The user's range of input is further restricted by the physiology and range of motion of our thumbs. A patent filed by Microsoft describes using sensors in a tablet's bezel to determine the user's grip [76]. The position of UI elements, such as a keyboard, could then move to be easily reachable. Similarly, a patent filed by Samsung describes the dynamic rearrangement of UI elements to fit the reach of user's thumb [77]. As with the MIME keyboard, both of these approaches aim to make UI elements (specifically keyboard keys) easier to select. However, to determine the location for easy-to-select keys, selection time data is needed.

Previous research has gathered selection time data on mobile devices, but under different conditions. Perry and Hourcade [90] conducted a study to gather selection time data for 25 targets placed in a grid encompassing the entire screen. Their study seems more applicable to general icon or button selection, rather than typing in particular. Twenty-five targets are insufficient to discretely map all the characters of the English alphabet on a keyboard. In addition, a keyboard typically would not occupy the entire screen. A methodology more focused towards text entry is required.

Hughes et al. [90] gathered selection time data for a 30-key grid representing the text entry region of a mobile device. However, selections were made using a stylus. Thumb input will be the intended interaction for the MIME keyboard. Although a stylus allows for greater precision than the tip or pad of one's thumb or finger, using one's thumb facilitates peripheral-free text entry. This chapter gathers selection time data using a modified version of Hughes et al.'s methodology.

6.2 Method

A study was conducted to determine the speed with which participants can perform swipe gestures and select targets in various regions of the touchscreen. This data will facilitate the assignment of frequent characters to keys that can be selected the fastest.

6.2.1 Participants

Twenty-four paid participants (10 male, 14 female) were recruited from the local university campus. Ages ranged from 18 to 34 years (mean = 23.5, SD = 4.5) and two participants were left-handed. Participants had to be frequent users of a mobile touchscreen device and send more than the average of five texts per day [11]. As a group, the participants sent an average of 62 messages per day.

6.2.2 Apparatus

The study used a Nexus 4 smartphone running Android 4.4.2 and custom software to administer trials and capture user touch events. The display measured 4.7 inches and had a resolution of 768×1280 pixels. The Home and Recent Apps navigation buttons at the bottom of the screen could not be disabled, so they were covered with heavy-weight paper secured with tape to prevent accidental activation during study sessions. In addition, the phone's wireless radios were disabled to eliminate disruptions due to incoming calls, text messages, or network activity.

Two applications were written specifically for the study. The first (Figure 46, left) presented an arrow indicating the direction participants should swipe. The second (Figure 46, right) presented buttons in a 6×5 grid that reached the left, right, and bottom borders of the screen. This region represents the typical location for an IME. A button near the top of the screen allowed participants to pause the session on demand (e.g., due to distraction or fatigue). Pressing the button again would restart the last trial, discarding any data from the interrupted trial. The software logged all touch events on the button grid, as well as timing data for all successful target selections.

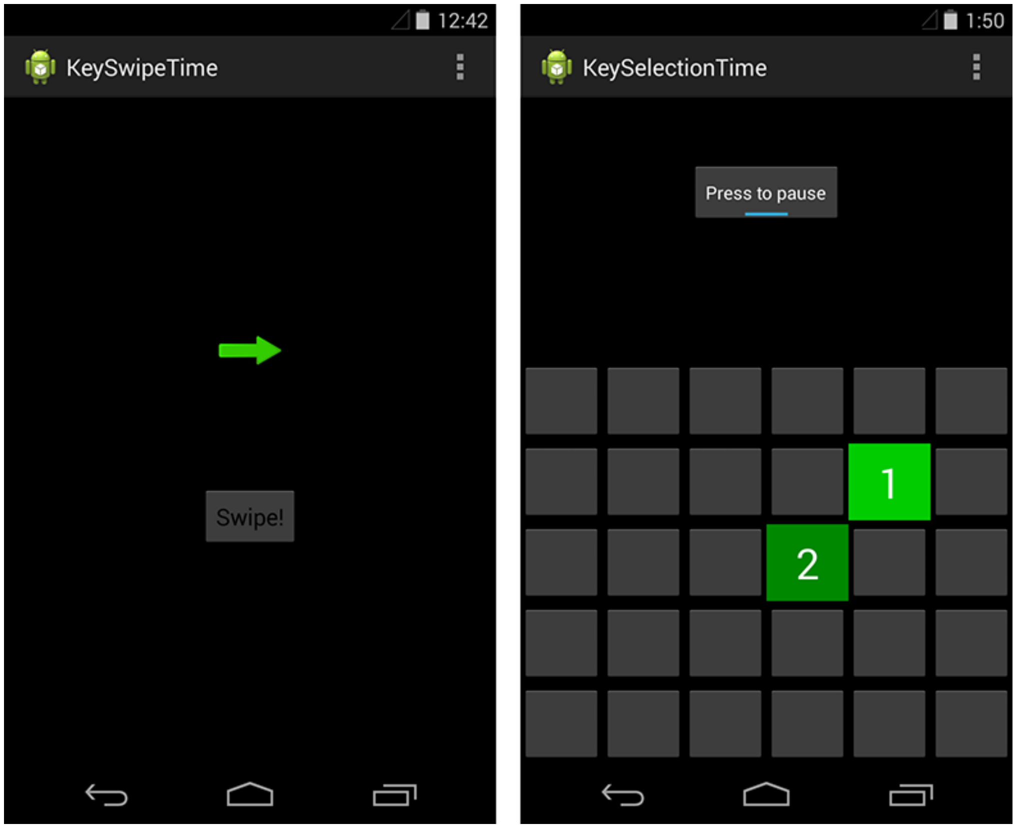


Figure 46. The applications used to measure swipe (left) and selection (right) times.

6.2.3 Procedure

Participants were first asked to perform swipes in the indicated direction as quickly as possible. The order of the directions was random, without replacement. After ten swipes in the correct direction, there was a one-second pause before the next direction was presented. The task was repeated for the other hand.

Participants were then presented with an array of 30 targets, similar to Hughes et al. [40]. In each trial, a pair of targets was highlighted. The first target was highlighted light green with the label "1", while the second target was highlighted dark green with the label "2". If the first and second target represented the same button, it was highlighted light green with the label “*”. Participants selected the first target, followed immediately by selecting the second target. Once successfully selected, a target would return to its normal appearance. Each trial was separated by a 200 ms pause and the application automatically paused and prompted the participant to “take a break” after every 100 trials. All pairwise combinations of the 30 keys were administered in a random order without replacement, leading to a total of 900 trials in a block. A block was administered once for each hand. Participants held the Nexus 4 in portrait orientation using one hand and selected targets using the thumb of the same hand (Figure 47). The study sessions took place in an office setting, with the participant seated at a desk. Study sessions lasted approximately 50 minutes.

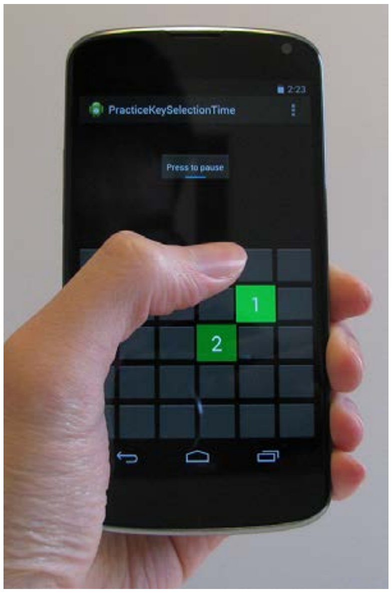


Figure 47. The above image demonstrates the hand position used during the study.

Participants completed a questionnaire to gather demographic information, feedback about the interaction task, and measurements for each hand. In keeping with other studies, measurements included hand length from the tip of the third finger (middle finger) to the wrist crease [6, 44], thumb length from the tip to the joint at the base [6, 44, 45, 90], thumb circumference around the joint closest to the tip [6, 44, 45, 90], and thumb width at the base of the finger nail. Participants' hands were measured using the figure-of-eight method [9] and also categorized "XS", "S", "M", "L", or "XL" based on their measured unisex glove size.[8] Measurements were taken with a plastic tape measure.

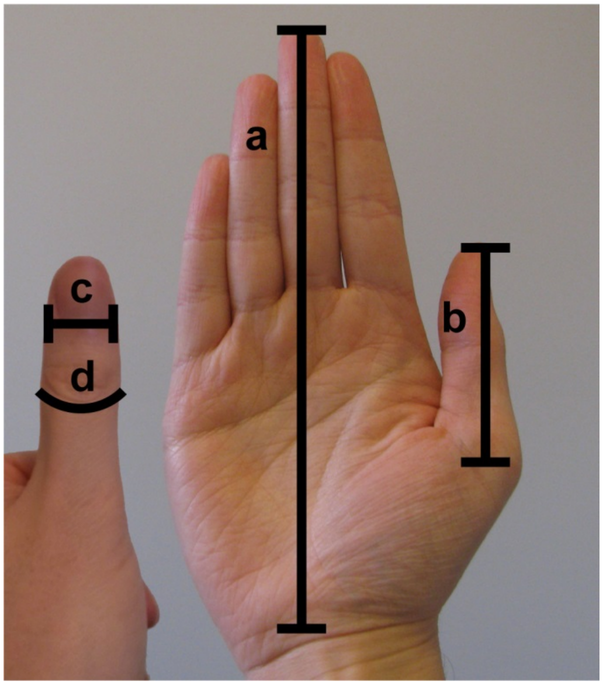


Figure 48. Measurements were taken for: a) hand length; b) thumb length; c) thumb width; and d) thumb circumference.

---

[8] http://www.glove.org/Modern/glovemeasure.php

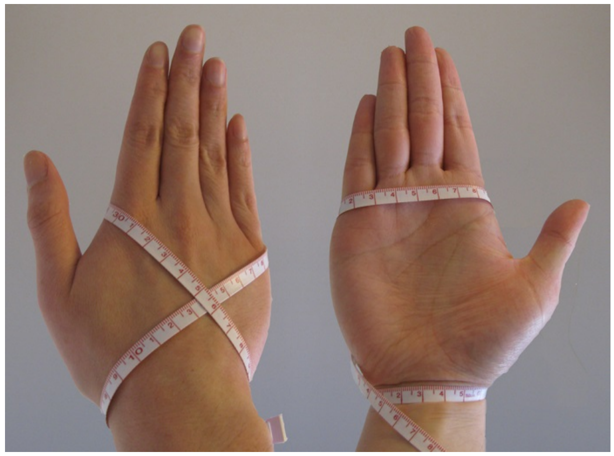


Figure 49. Specifically, the Pellecchia [88] technique for the figure-of-eight measurement was used, which starts at the ulnar side of the wrist (away from the thumb). The Maihafer et al. [70] technique is similar, but starts at the radial side of the wrist (near the base of the thumb) and follows a mirrored path.

6.2.4 Design

This study employed a within-subjects factor design, input hand, with levels left and right. Hand size was a between-subjects factor, with levels XS, S, and M. The number of participants with larger hand sizes was insufficient for statistical analysis. In addition, the order of testing was counterbalanced, with one left-handed participant in each group. For the first task, the dependent variables were swipe duration (in milliseconds) and swipe length (i.e., the length of the gesture, in pixels). For the second task, the dependent variable was selection time (in milliseconds). This was calculated as the duration between selection of the first target and selection of the second target.

6.3 Results and Discussion

Table 4 summarizes the hand measurements of all the participants, averaged by hand size. The effects of different hand sizes are discussed subsequently.

Table 4. This table summarizes mean (and SD) measurements in millimeters for a) hand length, b) thumb length, c) thumb width, d) thumb circumference, and e) figure-of-eight. *The L and XL groups had only one participant each.

Size	Left: a	b	c	d	e	Right: a	b	c	d	e
XS	170.1 (4.4)	56.0 (2.7)	15.9 (1.7)	55.1 (2.7)	371.9 (8.4)	169.2 (4.2)	56.3 (3.3)	16.0 (1.7)	55.0 (2.8)	378.7 (7.9)
S	180.0 (3.9)	58.1 (3.6)	17.3 (1.3)	58.1 (2.7)	397.9 (8.0)	179.0 (5.5)	58.4 (4.5)	17.7 (1.3)	59.4 (2.3)	401.6 (6.6)
M	188.7 (7.4)	60.5 (2.9)	19.3 (1.0)	64.0 (3.7)	427.3 (13.4)	189.5 (7.9)	65.2 (3.4)	20.2 (0.8)	65.5 (3.2)	433.7 (15.0)
L*	193	68	21	67	445	191	68	20	65	445
XL*	210	71	23	75	484	203	77	22	74	498

6.3.1 Selection Time and Patterns

ANOVA showed the effect of input hand on selection time was significant ($F_{1,22} = 11.39$, $p < .005$) with an observed power of .90. On average, right hand input was 14.7% faster than left hand input. Unfortunately, the small number of left-handed participants ($n = 2$) made it impractical to determine if handedness had an effect. However, one cannot assume that input is necessarily faster with one's dominant hand. Selection times for three right-handed participants were faster in the left-hand condition, while the two left-handed participants were faster in the right-hand condition. In addition, counter balancing worked ($F_{1,22} = 3.15$, $p > .05$), as there was no apparent asymmetric skill transfer.

There was only one participant in each of the L and XL hand size groups. Because this is insufficient for statistical analysis, they were identified as outliers. Thus, the remaining 22 participants were used to analyze the effect of hand size. ANOVA requires that data approximately match a normal distribution. The Shapiro-Wilk test for normality was chosen because of its power with small sample sizes [95]. It showed the data in the M group significantly deviated from a normal distribution ($p < .05$). To correct this, the data in all three groups was transformed using a logarithmic function and the ANOVA was conducted on the transformed data. The effect of hand size on selection time was significant ($F_{2,19} = 4.88$, $p < .05$) with an observed power of .74.[9] Tukey, Scheffé, LSD, and Bonferroni post hoc tests all indicated significance only between the XS and S groups. On average, participants in the XS group performed 43.3% slower than those in the S group. This is interesting, as the difference in hand measurements between the XS and S groups is smaller than the difference in hand measurements between the S and M groups (Table 4). In particular, the average difference in thumb length between the XS and S groups is only 2.1 mm for both hands. Between the S and M groups, the difference is 6.8 mm for the right hand, and 2.4 mm for the left hand. Perhaps the S group measurements represent the lower-bound hand size for easy operation of a smartphone of that size. It would be interesting to investigate how many smartphone users deem that size too big. Subsequent popular smartphone models are larger than the Nexus 4, so it would seem that device manufacturers see a sufficient market for smartphones with touchscreens exceeding 5 inches in size. Still, the new smartphones are drawing the ire of technology writers, who see the new large sizes as too big for the average person's hand (approximately 180 mm) [122].

---

[9] Performing the analysis on the original data yielded a similar result ($F_{2,19} = 4.71$, $p < .05$), power .72.

The selection times for each key were averaged. A visualization of this data appears in Figure 50. As with Perry and Hourcade's study [90], the corner regions were selected the slowest. Average selection times ranged from 197 ms to 789 ms, which is slower than the 147-330 ms range from Hughes et al. [40]. Participants were required to successfully select each highlighted key before the next target was made active (i.e., ready to be selected). However, all touch events made when a target was active were logged for analysis. Figure 51 shows two interesting examples of key selection.

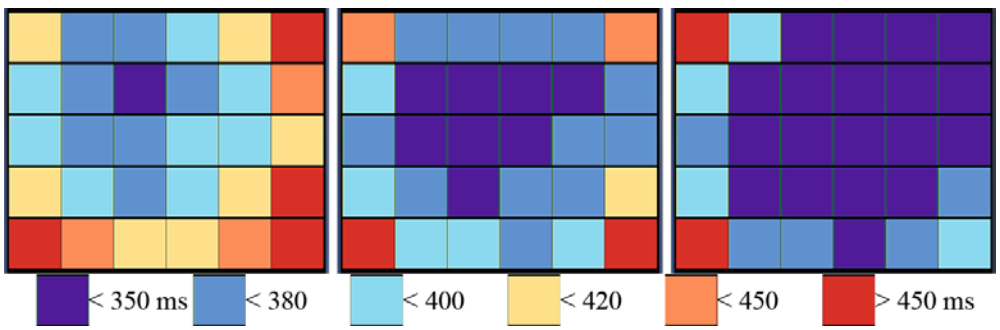


Figure 50. Keys are coloured to show the average selection time for left-hand input (left) and right-hand input (right). The centre image represents the average of the two conditions.

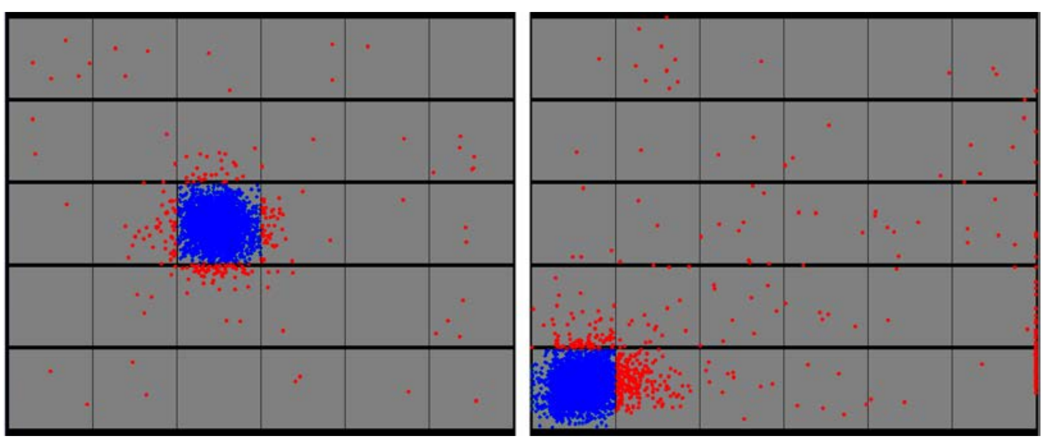


Figure 51. Blue dots represent successful selections of that key; red dots represent misses.

In the left image, almost all of the touch events are on the key. Misses mainly occur vertically or horizontally adjacent to the key, close to the middle of its border. All touch events on or around the key tend to slightly favour the bottom edge. These findings are similar those of Henze et al. [34], who gathered touch events on a QWERTY soft keyboard. In their study though, the keyboard keys were narrower and touch events were more concentrated towards the centre of the key. However, touch events for the keys on the bottom row tend to favour the top edge of the key. Events for corner keys favour the edges towards the centre of the keyboard, as depicted in the right image. Again, misses usually occur vertically or horizontally, but not diagonally. Participants often remarked that keys along the edges and at the corners were the most difficult to select. In the right image, one can also see a collection of touch events along the right edge of the screen. This is likely evidence of the base of one's thumb touching the screen as one stretches to reach the target. Both images use touch events from both left-hand and right-hand input conditions.

6.3.2 Swipe Time and Length

Data gathered from the swipe task will be used to model gesture input of shift, space, backspace, and enter keys. Swipe durations will be used to calculate an entry speed prediction, while swipe lengths will be used to determine minimum thresholds for the gestures. A summary of both duration and length data appears in Table 5.

Table 5. The mean (and SD) of swipe duration and gesture length for left- and right-hand input.

Direction	Duration (in ms)		Length (in pixels)	
	Left	Right	Left	Right
Down	141.3 (71.9)	115.7 (44.4)	136.8 (97.9)	122.5 (88.8)
Left	134.8 (56.4)	131.6 (53.8)	166.1 (100.9)	145.0 (81.4)
Right	152.5 (76.2)	114.3 (43.1)	135.6 (74.1)	140.4 (101.4)
Up	140.3 (68.9)	122.5 (65.7)	132.0 (81.1)	115.5 (75.5)

The effect of input hand on swipe duration was significant ($F_{1,22} = 7.43$, $p < .05$), with an observed power of .74. Furthermore, the effect of hand size on swipe duration was also significant ($F_{2,19} = 3.73$, $p < .05$), with an observed power of .61. Tukey, Scheffé, LSD, and Bonferroni post hoc tests were conducted, but only the LSD procedure showed significance, and that was between XS and both other groups. The XS group was 40.3% slower than the S group and 47.2% slower than the M group. Participants would likely find the Nexus 4 and most new smartphones too big and cumbersome. As with the selection task, the order of the conditions was not significant ($F_{1,22} = 2.63$, $p > .05$). With regards to swipe length, input hand ($F_{1,22} = 2.49$, $p > .05$), hand size ($F_{2,19} = 1.19$, $p > .05$), and condition order ($F_{1,22} = 1.20$, $p > .05$) had no significant effect.

6.4 Conclusion

Participants in the XS group took significantly longer to perform the swipe and selection tasks, likely due to the size of the device. In general keys at the corners of the layout were the most difficult to select, but this was expected. The gathered swipe and selection times represent the final components required to design the MIME keyboard.

Chapter 7
Optimizing a Keyboard Layout for Mobile Text Entry

This chapter presents the generation and evaluation of the MIME keyboard. Characters are assigned to keys based on their digram frequency in a corpus of mobile text entry activities (Chapter 5) and key selection times (Chapter 6). In addition, the input method will provide options for haptic and aural feedback to accommodate users' preferences (Chapter 4). Finally, MIME will be evaluated using TEMA (Chapter 3).

7.1 Motivation

The QWERTY layout lends itself well to two-handed text input on tablets and smartphones held in landscape orientation, as the division of responsibility for the left and right hands carries over from desktop touch typing. However, one handed (one thumb) text entry is needed when the user's other hand is occupied (e.g., holding a coffee, bag, or umbrella). If the user is mobile, one handed operation of the smartphone allows the other arm to swing freely, which benefits locomotion efficiency and gait stability [80].

Existing research [83] has looked at optimizing two-handed typing on a tablet, and word gesture techniques (e.g., *Swype*) have improved one-thumb text entry using the QWERTY layout. Related work (detailed subsequently) have non-QWERTY layouts optimized for stylus input, but unlike MIME, they do not consider the ergonomics of thumb movement.

7.2 Related Work

The general layout of the MIME keyboard is 30 keys arranged in six columns and 5 rows. This layout is also shared by other soft keyboards. The *Opti II* [64] layout rearranges characters so that frequent trigrams are located on adjacent keys in the centre of the keyboard. Because of the high frequency of the space character in text entry, four keys on the layout can input that character. *Fitaly* (www.fitaly.com) also arranges frequent characters in the centre of the keyboard, but uses two oversized keys for the space character. The unnamed layout designed by Hughes et al [40], (herein referred to as "Hybrid Ant") gathered movement time measurements for stylus tapping and mapped frequent digrams to quickly selectable key pairs.

q	k	c	g	v	j
spc	s	i	n	d	spc
w	t	h	e	a	m
spc	u	o	r	l	spc
z	b	f	y	p	x

z	v	c	h	w	k
f	i	t	a	l	y
space		n	e	space	
g	d	o	r	s	b
q	j	u	m	p	x

	k	g	i	c	z
	f	n	t	h	w
q	o	s	spc	a	y
j	u	r	e	d	v
	p	m	l	b	x

Figure 52. The *Opti II* (left), *Fitaly* (centre), and "Hybrid Ant" (right) layouts, after [40].

7.3 Layout Generation

The challenge of optimizing a keyboard layout is a difficult one to solve. Optimization problems are common in Computer Science. One called the quadratic assignment problem (QAP) [48] is described informally as follows: A set of *n* facilities must be built in *n* locations. Each pair of locations has a "distance" between them, and each pair of facilities has a "flow" of materials between them. Consequently, the cost of transporting goods between two facilities is a function of their distance and flow. The problem is to build facilities in locations, such that the overall cost of transporting goods is minimized.

Similarly, the keyboard optimization problem (KOP) can be described as a reduction of the QAP: A set of *n* characters must be assigned to *n* keys. Each pair of keys has a movement time between them, and each pair of characters (digram) has a frequency (within the corpus). The cost of entering a digram is the movement time between the two corresponding keys, multiplied by the digram's frequency in the corpus. The problem is to assign characters to keys, such that the time to enter the corpus is minimized. Indeed, approaches to the QAP have been used to generate optimized layouts for typewriters [10].

Unfortunately, Sahni and Gonzalez showed that the QAP is NP-hard[10] [98]. Thus, so too is the KOP. Evaluating every permutation of character assignment has a running-time complexity of O(*n*!). For the English alphabet (26 letter, plus space), that equates to more than 10.9 octillion character arrangements!

However, there are alternatives to an exhaustive search. The derivation of OPTI is described as "trial-and-error" using key-to-key movement time [69], while the Metropolis keyboard is based on a method used to search for the minimum energy state

[10] Some describe the QAP (or similar optimization problems) as NP-complete. However, an optimal solution to the QAP cannot even be verified in polynomial time, so the problem is NP-hard, not NP-complete [93].

in statistical physics [144]. Fitts' law [29] was used to predict movement times for OPTI and was used as an energy analogue for Metropolis. A genetic algorithm was used to produce slightly optimized variations of OPTI and Metropolis [94]. In a genetic algorithm, a set of candidate solutions are evaluated. The candidates that perform better are combined to form the next set of candidates. This process continues until a candidate achieves the desired performance, or until repeated generation of new candidates fails to produce candidates with significantly increased performance. The Hybrid Ant keyboard used an ant colony optimization algorithm, which searches for a solution by simulating how an ant colony uses pheromones to search for food. As ants find paths to food, more successful paths have higher levels of pheromones and are more likely to be followed by other ants. In this metaphor, an ant's successful path to find food represents a fast path across a keyboard to enter the corpus.

For MIME, it was decided to solve the KOP for a subset of the English alphabet, representing the most frequent letters, and then incrementally add characters in an optimal way to complete the layout. Sixty-nine characters were considered for the layout. This corresponds to the number of unique characters in the Mobile corpus, ignoring letter case (the shift modifier is mapped to an upward swipe gesture) and the space character (which is mapped to a rightward swipe gesture). The general design for the MIME keyboard involves 60 characters, assigned to 30 keys. Thirty primary characters appear on the *Alpha* layout and are inputted by tapping a key. Thirty secondary characters appear on the *Beta* layout and are inputted by pressing a key for short (e.g., 500 ms) duration. Consequently, nine characters would be left out of the layout. Character priority was based on digram (not letter) frequency in the corpus, as selection times are based on key pairs. To ensure the layout accommodates the most frequent digrams, both characters in each digram must be included in the layout. Layouts were evaluated by calculating an entry speed based on the corpus and the gathered key selection times, averaged for both left- and right-hand conditions. The results from both conditions were averaged, because some users might perform faster with their non-dominant hand (as some participants did).

The first subset of characters, *S1*, consisted of the 12 letters comprising the 16 most frequent digrams of the corpus. Those letters (i, n, t, h, e, a, r, o, u, g, m, and s) represent 54.5% of the corpus. They were assigned to the 12 keys in the middle of the keyboard (which also had the fastest selection times) by calculating an entry speed for each of the approximately 479 million permutations. On a whim, the same process was performed using the 12 most frequent letters, based on letter (not digram) frequency. However, none of the approximately 479 million permutations yielded a better solution.

Initially, a greedy algorithm was used to assign the next 18 most frequent characters (subset *S2*). That is, each subsequent character was systematically evaluated on every available key and assigned to the one that corresponded to the highest entry speed. However, a greedy approach sometimes fails to find a solution that is globally optimal. To address this, approximately 422 million random permutations of S2 were evaluated. The one with the highest entry speed (higher than the greedy solution) was selected, thus completing the Alpha layout.

Some of the remaining characters have a semantic association. This is especially true for numerals (subset *S3*), as they have an associated order. First, the numeral zero was optimally assigned to the Beta layout. The numerals 1-9 were then assigned, mimicking layout of the numeric keypad found on most QWERTY keyboards. The next 20 most frequent characters (subset *S4*) were arranged using a greedy algorithm. Interestingly, the resulting layout had associated characters (e.g., <>, ()) in adjacent or nearby keys. Although the + character occurs slightly more frequently than ^ in the corpus, the ^ character was included in S4 to facilitate entry of smilies in the SMS and Twitter phrase sets. The detriment to entry speed was negligible, at less than 0.01 wpm. The remaining characters in the corpus comprise subset *S5*.

Table 6. The subsets of characters in the MIME layout.

Subset	Characters	Location
S1	i n t h e a r o u g m s	Alpha
S2	. l c v y w b f d k p ! : / j ' x ,	Alpha
S3	0-9	Beta
S4	q * _ - ?) @ z (> = < ~ # & " % $ ^ ;	Beta
S5	| \ + { } [ ]	“Sym” submenu

Approximately 422 million random permutations for the Beta layout were evaluated and compared to the greedy solution, but none yielded a faster layout. The completed MIME keyboard is the result of evaluating over 1.8 billion layout permutations and appears in Table 7.

Table 7. The generated MIME character arrangement.

Alpha Layout	Beta Layout
, y d l b x	^ < > ? = %
j u o e r .	~ q * 7 8 9
f m n t s p	& @ _ 4 5 6
' g i a h /	" z – 1 2 3
! v k c w :	; # (0) $

Using the Mobile corpus and the data from the key selection time experiment, the estimated entry speed of the MIME layout is 30.3 wpm. At first glance, this seems to be lower than the entry speeds quoted for *Opti II*, *Fitaly*, and Hybrid Ant. However, these input methods provide only for input of far fewer characters. By evaluating input of only 27 characters (alphabet, plus space) and using the selection time data from Chapter 6 to evaluate the other techniques, the estimated entry speed of MIME becomes the fastest (Table 8). The discrepancy between estimated and quoted speeds can be attributed to the authors using theoretical timing values, a specific input phrase, and different empirical timing values.

Table 8. A comparison of entry speeds.

Technique	Entry Speed
MIME	30.3 wpm
MIME (27 characters)	34.9 wpm
OPTI II (27 characters)	34.4 wpm (42.4 wpm quoted [64])
Fitaly (27 characters)	33.7 wpm (58.9 wpm quoted[11])
Hybrid Ant (27 characters)	33.5 wpm (65.3 wpm quoted [40])

The fully-implemented MIME IME appears in Figure 53. Keyboard keys are slightly smaller than those in the key selection experiment. There, the keys were

[11] www.fitaly.com/domperignon/domperignon4.htm

designed to cover a large area of the screen and measured 128×120 pixels. With MIME, the keys are made smaller (approximately 115×102 pixels[12]) to move input away from the difficult-to-reach edges and corners. Specifically, the base of the user's thumb is less likely to activate a key by accident (see Figure 51, right).

Figure 53. The fully-implemented MIME IME.

The decrease in size also allowed an added row of keys. Based on preliminary feedback, it was decided to add keys to supplement the gestures for shift, space, backspace, and enter by providing explicit keys for this otherwise hidden functionality. These would provide more explicit input options to those completely unfamiliar with MIME's gesture recognition functionality. The bottom row also contains keys for IME preferences, such as settings for audio and haptic feedback (see Chapter 4), the colour of the Beta layout characters, and the hold duration threshold to activate the Beta layout.

---

[12] To accommodate various screen sizes, a key measures 15% of screen width, by 8% of screen height.

The default hold duration threshold is set at 300 ms to mimic that of Google's Android (QWERTY) keyboard. The "Sym" key displays a popup menu to select additional characters for input. Currently, this menu contains *S5*, the subset of characters not included in the Alpha or Beta layout. This menu can be augmented with additional characters (e.g., accented letters, emoji) to facilitate additional input.

7.4 Method

A longitudinal study was conducted to determine how users' text entry performance using MIME improves over time.

7.4.1 Participants

Six paid participants (4 male, 2 female) were recruited from the local university campus. This number of participants is consistent with (or exceeds) other longitudinal text entry studies [41, 42, 81, 82, 121, 133]. Ages ranged from 24 to 33 years (mean = 29.7, SD = 3.4) and all participants were right-handed. Participants had to be frequent users of a mobile touchscreen devices and send more than the average of five texts per day [11]. As a group, the participants sent an average of 15 messages per day.

7.4.2 Apparatus

A series of identically configured Nexus 4 smartphones running Android 4.4.2 were used for this study. The MIME keyboard was installed on them and TEMA was used to administer trials from the Mobile phrase set and gather data. Google's standard QWERTY keyboard was pre-installed on all the devices.

For the QWERTY keyboard, options for spell checking, auto-capitalisation, auto-correction, word-suggestions, and word gesture typing were all disabled. Options for audio and haptic feedback were disabled for both the QWERTY and MIME IMEs.

7.4.3 Procedure

For each condition, participants entered five blocks of ten phrases. The first block served as a warm-up, which is consistent with previous text entry research [66]. This continued for ten sessions. Participants held the smartphone in their right hand and entered text using their right thumb. They were instructed to enter text as quickly as possible, to correct errors if noticed immediately, but to ignore errors made two or more characters back (i.e., to prevent deletion of many correct characters to correct an early mistake). Furthermore, participants were encouraged to take a break between phrases, if needed.

Participants' demographic information and hand measurements were recorded prior to the first session. At the end of the first and last sessions, participants completed NASA TLX questionnaires to gather feedback about the input techniques.

Participants completed sessions at their convenience, subject to the following restrictions: 1) Wait at least two hours between sessions. 2) Do not exceed two sessions per day. 3) Do not exceed two days between sessions. These restrictions are consistent with those of other longitudinal text entry studies [20, 41, 141].

7.4.4 Design

The study employed a within-subjects factor, technique, with two levels: MIME and QWERTY. Session was a between-subjects factor, with levels 1 to 10. Similar to a previous study [69], the order of the techniques was counterbalanced and also alternated between sessions. For odd-numbered participants, odd-numbered session proceeded QWERTY-MIME, while even-numbered sessions proceeded MIME-QWERTY. For even-numbered participants, the order was reversed. All participants received a chart to track their progress and remind them of condition order. Each session consisted of two conditions. Each condition consisted of five blocks of ten phrases. The first block served as a warm-up, so analysis was based on the resulting 4800 (6×10×2×4×10) trials.

The dependent variables were entry speed and accuracy, as calculated by TEMA. Entry speed was reported in words-per-minute and accuracy was measured according TER, CER, and UER metrics.

7.5 Results and Discussion

The participants had a mean (and SD) hand length of 188.0 mm (14.5), thumb length of 61.0 mm (4.5), thumb width of 18.8 mm (0.8), thumb circumference of 61.0 mm (6.3), and figure-of-eight measurement of 418.7 mm (36.6). One participant was classified as an XS glove size, two as S, two as M, and one as L. The entry speed and error rate values are summarized and illustrated in this section. The values for each participant, including means and standard deviations appear in Appendix E.

7.5.1 Entry Speed

ANOVA requires that the data being analysed not diverge significantly from a normal distribution. A Shapiro-Wilk test showed that the entry speed data (Figure 54) satisfied this requirement ($p < .05$). ANOVA showed layout had a significant effect on entry speed ($F_{1,4} = 103.84$, $p < .0005$), with QWERTY performing faster than MIME. Although the MIME entry speed did not surpass that of QWERTY during the ten sessions, its performance (Figure 54) shows substantial improvement and potential. In addition, session had a significant effect on entry speed ($F_{9,36} = 10.08$, $p < .0001$) and the layout × session interaction effect was also significant ($F_{9,36} = 11.21$, $p < .0001$). The QWERTY entry speed remained relatively steady, averaging 23.3 wpm throughout the study, while MIME entry speed started at 10.0 wpm and increased to 17.2 wpm. Counter balancing worked, as the group effect was not significant ($F_{1,4} = 0.29$, ns).

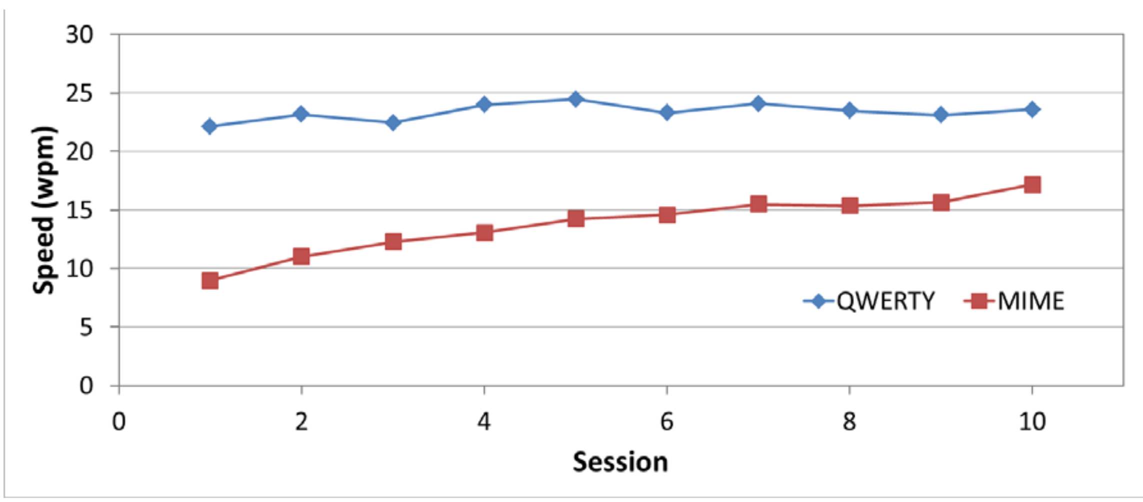


Figure 54. Entry speed measured in wpm for the ten user study sessions.

The duration of the QWERTY component ranged from 18.6 minutes in session one to 15.1 minutes in session ten, for an average of 16.4 minutes. Because of the novelty of MIME, participants initially took far longer to complete a session. The duration of the MIME component started at 84.1 minutes in session one and quickly decreased, ending at 22.4 minutes in session ten, for an average of 33.0 minutes. Figure 55 shows QWERTY and MIME entry speed performance extrapolated to 55 sessions. MIME performance crosses over and surpasses QWERTY performance by the 45th session. By also extrapolating the duration of the MIME component for each session, one can estimate that the crossover will occur shortly after 12 hours of practice.

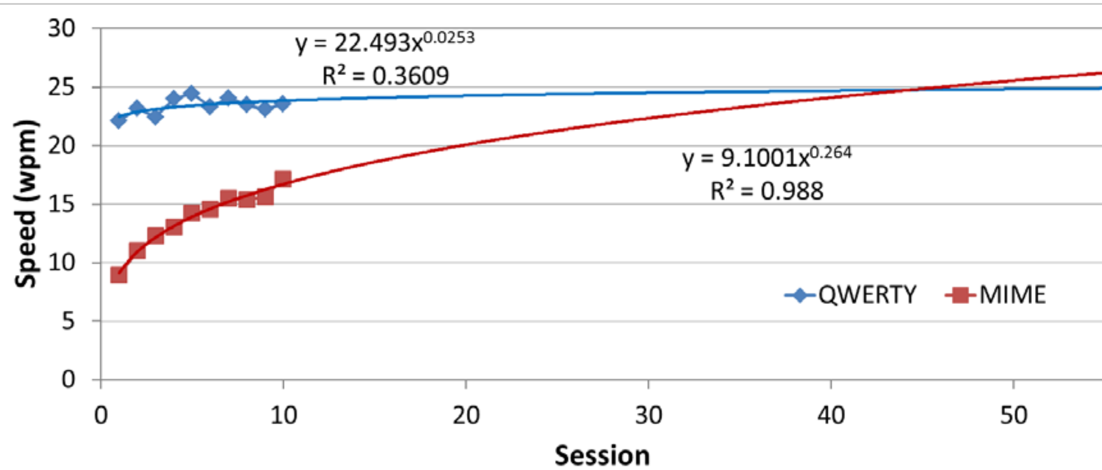


Figure 55. QWERTY and MIME entry speeds, extrapolated to 55 sessions. The crossover occurs by session 45.

After approximately 330 minutes (5.5 hours) of practice, MIME's entry speed trails far behind the 45 wpm that *Opti* participants achieved after 400 minutes of practice [69] and the 23 wpm that *VirHKey* participants reached after 7 hours of use [73]. However, MIME performs better than *Hex*, with which participants reached 12 wpm after

30 hours of use [127]. The technique *KALQ* achieved 37 wpm, but required using two hands and 13-19 hours of practice.

7.5.2 Accuracy

A Shapiro-Wilk test showed the error rate data (Figure 56) for some sessions did not satisfy the requirement or normality ($p < .05$). Logarithmic and polynomial transformations failed to remedy this. The Aligned Rank Transform (ART) procedure [135] does not have a requirement of normality and was used to analyze the data.[13]

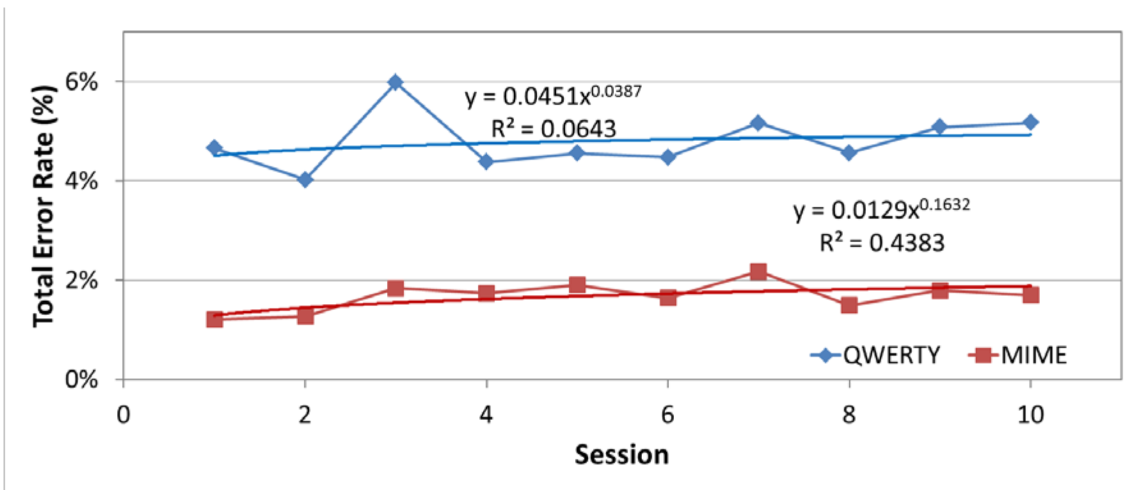


Figure 56. Total error rates for the two conditions evaluated over the ten sessions.

---

[13] Because ART transforms the error rate values into ranks, the original empirical data is lost and statistical power is decreased. In human-computer interaction research, it is common to proceed with ANOVA, even if the data diverge from a normal distribution [61 (p. 223)]. ANOVA on the original data shows only the main effect of layout to be significant on TER ($F_{1,4} = 46.45$, $p < .001$), UER ($F_{1,4} = 7.00$, $p < .05$), and CER ($F_{1,4} = 34.80$, $p < .005$).

TER was 1.7% for MIME and 5.2% for QWERTY in session ten. As with entry speed, the group effect was not significant for TER ($F_{1,4} = 2.58$, $p > .05$), UER ($F_{1,4} = 0.96$, ns), and CER ($F_{1,4} = 3.51$, $p > .05$). Layout had a significant effect on TER ($F_{1,4} = 50.16$, $p < .001$), UER ($F_{1,4} = 7.38$, $p < .05$), and CER($F_{1,4} = 46.24$, $p < .001$). Entry using MIME was consistently more accurate than using QWERTY; the slower entry speed benefitted accuracy.

Session also had a significant effect on TER ($F_{1,4} = 2.33$, $p < .05$). Unfortunately, it seems that TER tends to increase with technique familiarity. Figure 57 illustrates TER values in the first and last session for both conditions.

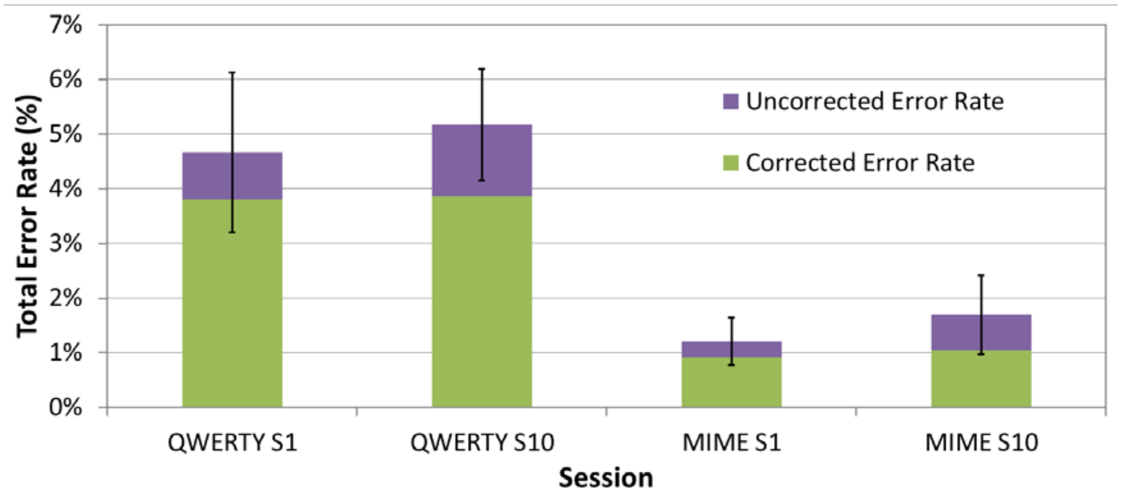


Figure 57. A comparison of error rates between the first and last sessions for each condition. Error bars represent ±1 SD of TER. A box plot representation appears in Appendix F.

The discrepancy between the two techniques is clearly evident. Though TER for MIME is significantly lower than for QWERTY, another pattern emerges: for each technique, TER is higher at the end of the study than at the beginning. Increased familiarity with a technique might lead to increased carelessness and decreased accuracy, which may benefit entry speed. However, the increase in TER from session one to session ten is not statistically significant ($p > .05$).

MIME's low error rate is lower than that of other techniques, but the use of various error rate metrics make meaningful comparisons impossible. *KALQ* yielded a 5.2% MSD error rate [83]. The error rate in the *VirHKey* study was 5.9% in the final session [73] and the error rate in the *Opti* study was 4.2% in the last session [69]. Both studies used a character wise error rate metric. Unfortunately, the MSD metric does not capture errors that were committed and corrected during the trial, and a character wise metric does not accurately represent corrected or uncorrected errors.

With a new technique (i.e., MIME), participants might see correcting errors as too costly with respect to performance, so they slow their entry speed. With a familiar technique (i.e., QWERTY), participants might be willing to make and correct errors, believing that their overall performance would be heightened. This speed-accuracy trade-off [108, 142] could represent participants' predisposition or a confounding variable. To eliminate the effect, a user study could be designed to ensure equal error rates in both conditions. Participants could be required to perform flawless text entry – a trial would be repeated if an error were made. This would ensure a TER of 0%, but would not reflect actual usage at all.

7.5.3 Participant Feedback

The NASA TLX scores are based on a discrete, non-continuous scale. Additionally, the Shapiro-Wilk test showed the participants' feedback does not represent a normal distribution. Thus, the non-parametric Wilcoxon signed rank test is appropriate for statistical analysis of a pair of conditions, administered within-subjects [61 (p. 214), 86 (p. 475)]. The feedback scores are summarized in Figure 58.

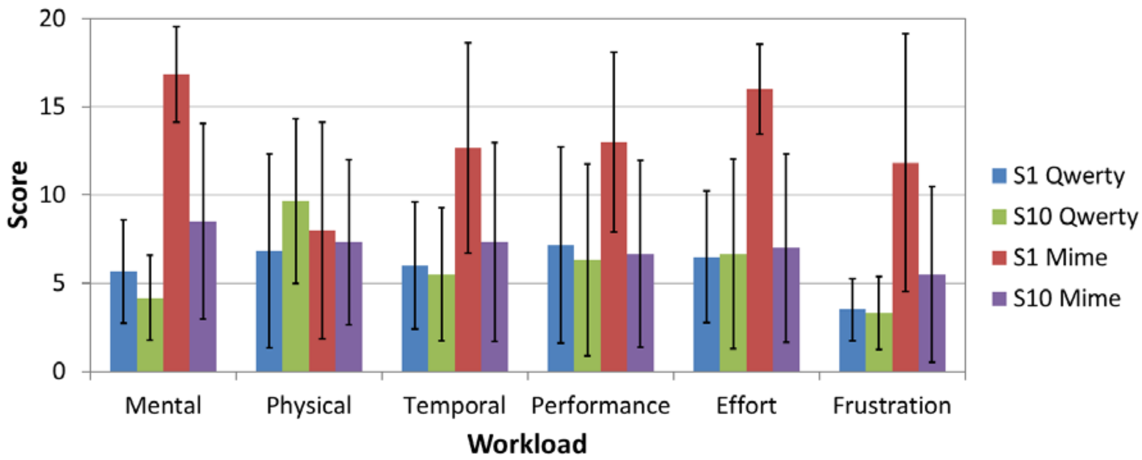


Figure 58. Participant feedback using NASA TLX workload scores.
Error bars represent ±1 SD. A box plot representation appears in Appendix F.

Not surprisingly, there is no statistically significant difference in workload scores between sessions one and ten using QWERTY ($p > .05$). This is likely the result of participants' familiarity with the layout. However, there was a slight worsening in the physical workload. The session ten score for QWERTY exceeded even that of MIME in session one. Specifically, three participants noted discomfort in their right hands when using QWERTY, but not MIME. These participants had XS and S glove sizes and owned mobile devices smaller than the Nexus 4. The discomfort is likely attributable to frequent thumb movement across the screen, necessitating a shift in grip for those participants.

There was a significant difference using MIME between session one and ten, resulting in an improvement in mental (z = -2.201, $p < .05$), temporal (z = -1.992, $p < .05$), performance (z = -2.023, $p < .05$), effort (z = -2.201, $p < .05$), and frustration (z = -2.201, $p < .05$) workload. There was no significant difference in physical workload (z = -0.548, ns). By session ten, there was no significant difference in any of the workloads between MIME and QWERTY ($p > .05$). This is promising, as it suggests that MIME's workload is similar to that of the established norm.

7.6 Conclusion

One handed, one thumb text entry is valuable for mobile users. MIME was designed for one thumb input, using the knowledge researched and presented in previous chapters. The character arrangement was determined using an exhaustive evaluation of a subset of the most frequent characters, representing a majority of the mobile corpus. Other characters were then added in a locally optimal manner. The MIME layout accommodates special character, numerals, and punctuations. It also moves keys away from the left and right edges to prevent unintentional selections.

MIME entry speed increased significantly during the study, with a projection that it would surpass QWERTY after 12 hours of practice. Error rates were consistently low and significantly lower than QWERTY. By the end of the study, participant feedback for MIME was positive, especially from three participants who found it painful to use the QWERTY keyboard.

Chapter 8
Concluding Remarks

This dissertation highlighted and summarized text entry research from over the past 21 years. Combined with new research that has been published and presented at several academic conferences, this dissertation presents the design and evaluation of a novel technique for mobile text entry.

8.1 Summary of Contributions

Examination of Techniques for Mobile Text Entry

The digitizers and motion sensors in mobile devices allow for numerous forms of input. Character recognition was found to be error prone due to variation in users' handwriting, and handwriting in general is slower than typing. Mid-air gestures were deemed inappropriate for mobility, as it would be too difficult to distinguish the user's movement of the device from the user's movement in the environment. Finally, a comparison of tapping and menu navigation (i.e., word gesture) text entry showed tapping techniques were faster. The general design of MIME used an optimized layout whose characters were entered using taps, long taps, and simple gestures.

Software to Evaluate Mobile Text Entry Techniques on Android Devices and Encourage Consistent Methodology

TEMA facilitates evaluating text entry methods on Android devices. It presents phrases for transcription and gathers metrics on the text inputted and low level events directly from the input method. The review of exiting text entry research revealed a variety of incongruent accuracy metrics, preventing meaningful comparison. A goal of TEMA was to bring consistency to evaluation methodology. In support of this, many researchers in academia and industry are now using TEMA. A user study conducted to illustrate TEMA's utility showed that a tapping text entry method can be faster than a menu navigation technique, thus reaffirming the design decision for MIME.

Explanation of Users' Feedback Preferences

Soft keyboards often have audio and haptic feedback options to compensate for the lack of feedback when typing. The results of a user study did not show feedback had a significant effect on text entry performance. However, a user survey revealed the reason for some users' preference for one form of feedback over others. Many users shun audio feedback out of courtesy for those around them. Others disable both audio and haptic feedback to save precious battery power.

Mobile Corpus

A corpus of mobile text entry was created to provide a phrase set to evaluate mobile text entry methods and to calculate digram frequencies for mobile text entry. Mobile text entry is often very informal and involves emailing, texting, and posting to social networks. A mobile corpus was created that mimicked the typical proportion of those tasks.

Investigation of Easily-Selectable Key Locations on a Mobile Touchscreen

With smartphones increasing in size, some users find it difficult to reach some areas of the screen. Even with hands large enough to reach all areas of the screen, the ergonomics of one's thumb makes selection in some regions of the screen easier than others. The conducted user study gathered gesture duration, gesture length, and target selection times for input using each thumb. The target selection data represents which areas of the screen are easier or harder to select. It also provides information on target selection tendencies (e.g., touches favouring the lower half of a target, or the edge towards the middle of the keyboard). Interestingly, participants did not always perform the fastest with their dominant hand. In addition, participants with hand lengths of approximately 170 mm performed significantly slower in selection tasks and likely required shifting their grip of the smartphone in order to reach some targets. This difficulty validated the design decision to map the frequent space character to a gesture that could be entered anywhere on the screen.

Development of a New Optimized Mobile Text Entry Technique

The results from previous chapters were used to develop MIME: input is performed using taps, long presses, and simple gestures; characters were assigned to keys based on their frequency in the mobile corpus and the key selection times; and the resulting input method was evaluated in a user study using TEMA. MIME's error rate was consistently low and significantly lower than QWERTY's. MIME's entry speed did not surpass QWERTY's during the ten sessions, but is expected to do so after approximately 12 hours of practice. Most importantly, MIME's layout places frequent character in easy to reach locations and this alleviated the difficulty and physical discomfort experience by participants using QWERTY.

8.2 Future Improvements to MIME

Participants' relatively slow entry speed with MIME benefitted their accuracy – the time used to scan for the next character allowed for more precise selection of the intended key. As a user's proficiency with MIME increase, his or her thumb movements will quicken, increasing the likelihood of committing an error. To alleviate this, a touch model or language model could be integrated to correct ambiguous selections. A similar technique was recently investigated with tablets [124].

The data from Chapter 6's user study could be used to determine a participant's actual point of selection, relative to the intended selection (i.e., the highlighted target). Figure 51 illustrates many near misses occurred along the bottom border of a key located towards the middle of the layout, and along the inner boarders of a key along the edge of the layout. To improve accuracy, a user's actual touch event could be translated to an intended location in real-time.

However, consistently translating the point of a touch event could result in an unintentional, adjacent key always being selected. This would result in user frustration similar to that for autocorrect. Augmenting the touch model with a language model would alleviate this issue. Digrams would be formed from the previous character and the character of the selected key, and likely adjacent keys (as determined by the touch model). The frequencies of those digrams would be compared and the most likely key selected.

Performance might also be improved by implementing word suggestion or completion. However, care must be taken not to increase the user's focus of attention [63]. Like the keyboard on the BlackBerry *Z10*, candidate words could appear above the next letter in the word – close to the user's next target for selection. With MIME, the user could accept the candidate word with a gesture (e.g., a checkmark).

8.3 Future Improvements to TEMA

TEMA has garnered a lot of attention from researchers around the world. An often-requested feature is the ability to use custom phrase sets with TEMA. Two researchers have already offered phrase sets for Chinese and German text entry. The upcoming acceptance of third-party keyboards on iOS devices also opens a new platform for development of TEMA-like software. (Though, a name change to TEMi or TEMO would be appropriate to reflect the iOS platform.) There are many avenues for further development of TEMA and also opportunities for collaborations in both academia and industry.

Appendix A
Summary of Existing Research

Table 9. A chronological summary of research involving onscreen and mobile techniques for English text entry. Unless otherwise noted, "error rate" refers to the uncorrected error rate. For comparison, the first entry evaluates a physical mini-QWERTY keyboard.

Authors	Input Type	Description	Technique (input device)	Speed (wpm)		Error Rate (%)		Practice	Notes
				Initial	Skilled	Initial	Skilled		
Clarkson et al. (2005) [20]	Tap	An evaluation of two physical QWERTY keyboards for mobile devices; error rate was averaged over both keyboards	QWERTY (Dell) QWERTY (Targus)	29.12 34.33	58.61 61.44	6.12	8.32	20 sess. (400 min.)	4
Goldberg and Richardson (1993) [32]	Chars	Single-stroke gestures representing English letters	*Unistrokes* (stylus)	6.2	13.0	*NR*	*NR*	1 week	1
Venolia and Neiberg (1994) [119]	Menu	Characters are selected from a pie menu	*T-Cube* (stylus)	3.6	21.2	*NR*	*NR*	9 sessions	1, 3
MacKenzie and Zhang (1997) [68]	Chars	Single-stroke gestures representing English letters	*Graffiti* (stylus)	*NR*	*NR*	14.5	3.1	5 min.	
Mankoff and Abowd (1998) [72]	Menu	Words are entered by selecting characters, in sequence, from a pie menu by drawing a single stroke	*Cirrin* (stylus)	*NR*	20	*NR*	*NR*	2 months	

Authors	Input Type	Description	Technique (input device)	Speed (wpm)		Error Rate (%)		Practice	Notes
				Initial	Skilled	Initial	Skilled		
Perlin (1998) [89]	Menu	Characters are entered based on the trajectory of egress from and entry into a central resting zone; input can be one continuous gesture	*Quikwriting* (stylus)	*NR*	*NR*	*NR*	*NR*	*NA*	
MacKenzie and Zhang (1999) [69]	Tap	Characters are arranged in decreasing frequency extending out from the centre of the keyboard	*Opti* QWERTY (soft keyboard)	17 28	45 40	2.07 3.21	4.18 4.84	20 sess. (400 min.)	6
Isokoski and Raisamo (2000) [42]	Chars	Compass directions are used to create optimal prefix codes for gestures; participants were trained using the touchpad, then tested on other devices	*MDITIM* (touchpad) *MDITIM* (joystick) *MDITIM* (keyboard) *MDITIM* (mouse) *MDITIM* (trackball)	2.5 *NR* *NR* *NR* *NR*	15 *NR* *NR* *NR* *NR*	7.6 5.5 4.9 6.5 6.5	6 3 3 5 7.2	10 sess. (5 hrs)	3
Ward et al. (2000) [123]	Menu	Pointer movement steers a target through a moving, hierarchical, language-model-based menu of characters	*Dasher* (mouse) QWERTY typing (keyboard)	7 50	18 55	7 8	3 6	60 min. 30 min.	3
Partridge et al. (2002) [87]	Midair	Text is entered by tilting a wrist-watch-sized device and pushing buttons	*TiltType* (prototype device)	*NR*	*NR*	*NR*	*NR*	*NA*	
Sazawal et al. (2002) [101]	Midair	Tilting selects a zones of characters; a word is inferred using *T9*-like disambiguation	*Unigesture* (prototype device)	2.0	*NR*	*NR*	*NR*	*NA*	1
Wigdor and Balakrishnan (2003) [126]	Midair	*Multi-Tap* input is disambiguated by tilting the cell phone to indicate the desired letter	*Multi-Tap* (phone keypad) *TiltText* (phone keypad)	7.53 7.42	11.04 13.57	4 22	3 9	16 blocks	3,6

Authors	Input Type	Description	Technique (input device)	Speed (wpm) Initial	Speed (wpm) Skilled	Error Rate (%) Initial	Error Rate (%) Skilled	Practice	Notes
Wobbrock et al. (2003) [140]	Chars	*EdgeWrite* gestures are drawn along the raised boarder of the writing area	*EdgeWrite* (stylus) *Graffiti* (stylus)	*NR* *NR*	6.6 7.2	*NR* *NR*	0.34 0.39	12 sentences	5
Isokoski and Raisamo (2004) [43]	Menu	*Quikwriting* characters are entered based on the trajectory of egress from and entry into a central resting zone; input can be one continuous gesture	handwriting (pen) *Quikwriting* (thumbstick) *Quikwriting* (stylus) typing (QWERTY keyboard)	26 4 4 39	*NA* 13 16 *NA*	*NR* 0.2 0.8 *NR*	*NR* 0.4 0.3 *NR*	20 sess. (5 hrs)	3,5
Kristensson and Zhai (2004) [50]	Menu	Word-gestures on an onscreen keyboard represent the sequential path between letters spelling the word	*SHARK*^{2} (stylus)	*NR*	70	*NR*	*NR*	*NR*	
Wobbrock et al. (2004) [137]	Chars	*EdgeWrite* gestures are drawn using a gamepad joystick (a.k.a. thumbstick)	date stamp (thumbstick) *EdgeWrite* (thumbstick) onscreen typing (ABC keyboard)	*NR* *NR* *NR*	4.43 6.40 6.17	*NR* *NR* *NR*	5.24 10.85 3.32	15 min.	4
Wobbrock et al. (2004) [138]	Chars	*EdgeWrite* gestures are drawn using a joystick and touchpad; WiViK is an onscreen keyboard; participants were motor-impaired	*EdgeWrite* (joystick) *EdgeWrite* (touchpad) *WiViK* (joystick)	*NR* *NR* *NR*	0.77 1.00 0.84	*NR* *NR* *NR*	29.56 25.40 5.11	35 min.	4
Martin (2005) [73]	Menu	Characters are arranged in a pentagrid; flicking gestures are used to navigate the pentagrid and select characters	*VirHKey* (stylus)	6.60	22.89	2.80	5.87	20 sess. (7 hrs)	6
Rinott (2005) [96]	Menu	*SonicTexting* associates thumbstick movements to characters; feedback is given aurally, not visually; thumbstick device called “Keybong”	*SonicTexting* (thumbstick)	*NR*	*NR*	*NR*	*NR*	*NA*	

Authors	Input Type	Description	Technique (input device)	Speed (wpm) Initial	Speed (wpm) Skilled	Error Rate (%) Initial	Error Rate (%) Skilled	Practice	Notes
Williamson and Murray-Smith (2005) [127]	Menu	*Hex* distributes characters into one of six groups; a character is entered by selecting the group and then the character within the group; the user study involved only one participant	*Hex*	*NR*	12	*NR*	*NR*	30 hours	
Wobbrock and Myers (2005) [136]	Chars	*EdgeWrite* is used with multiple devices; participants are "able-bodied experts"	*EdgeWrite* (button sliding)	*NR*	10.1	*NR*	10.8	*NR*	
			EdgeWrite (iso. joystick)	*NR*	12.3	*NR*	5.0		
			EdgeWrite (joystick)	*NR*	12.9	*NR*	8.4		
			EdgeWrite (key typing)	*NR*	16.6	*NR*	2.7		
			EdgeWrite (stylus)	*NR*	24.0	*NR*	2.8		
			EdgeWrite (thumbstick)	*NR*	14.7	*NR*	8.8		
			EdgeWrite (touchpad)	*NR*	19.1	*NR*	4.7		
			EdgeWrite (trackball)	*NR*	12.7	*NR*	6.6		
Chau et al. (2006) [18]	Chars	*EdgeWrite* is used with a thumb-operated isometric joystick on the front of a mobile phone and with a finger-operated isometric joystick on the back of a mobile phone	*EdgeWrite* (iso. joy.; back)	7.2	*NR*	2.2	*NR*	*NR*	
			EdgeWrite (iso. joy.; front)	12.0	*NR*	2.3	*NR*		
Felzer and Nordmann (2006) [26]	Menu	*LURD-Writer* arranges character in menus that are accessed using the directional inputs left, up, right, and down; the user study employed one motor-impaired participant and the *Ha*nds-free *M*ouse *Co*ntrol *S*ystem (HaMCoS)	*LURD* (mouse)	1.6	*NR*	*NR*	*NR*	*NA*	1
			LURD (HaMCoS)	1.0	*NR*	*NR*	*NR*		

Authors	Input Type	Description	Technique (input device)	Speed (wpm)		Error Rate (%)		Practice	Notes
				Initial	Skilled	Initial	Skilled		
Wobbrock and Myers (2006) [132]	Chars	*EdgeWrite* is used with a trackball; participants consisted of four able-bodied and one motor-impaired users	*EdgeWrite* (trackball; able) *EdgeWrite* (trackball; impaired)	*NR* *NR*	9.87 5.28	*NR* *NR*	3.75 11.80	45 min. 8 sessions	4
Wobbrock et al. (2006) [139]	Menu	*EdgeWrite* is augmented with *Fisch*, an in-stroke word completion technique	*EdgeWrite* (iso. joy.) *EdgeWrite* (iso. joy.; *Fisch*)	9.39 12.81	*NR* *NR*	1.01 0.54	*NR* *NR*	*NR*	
Költringer et al. (2007) [47]	Menu	*TwoStick* uses a dual-joystick gamepad controller; one joystick selects one of nine zones, while the other joystick selects a character from that zone	onscreen typing (QWERTY) *TwoStick* (thumbstick x2)	6.32 5.10	8.58 13.34	12.90 14.87	5.35 8.21	20 sess. (5 hrs)	4
Kristensson (2007) [51]	Menu	Word-based gestures on an onscreen keyboard represent the sequential path between letters spelling the word; *SHARK* using a QWERTY layout	*ShapeWriter* QWERTY (two-thumb, physical)	20.9 27.7	*NR* *NR*	1.1 1.1	*NR* *NR*	*NA*	
Witt and Janssen (2007) [128]	Midair	A data glove is used to capture hand gestures, which are mapped to input; Method 1 uses modifier keys, Method 2 uses only gestures	Method 1 (data glove) Method 2 (data glove)	1.5 1.5	2.5 2.0	42 47	29 34	5 sessions	3

Authors	Input Type	Description	Technique (input device)	Speed (wpm) Initial	Speed (wpm) Skilled	Error Rate (%) Initial	Error Rate (%) Skilled	Practice	Notes
Wobbrock et al. (2007) [134]	Menu	*EdgeWrite* is augmented with *Fisch*, an in-stroke word completion technique; in the "input blind" conditions, users held the device under a table but could view entered text on desktop display	*EdgeWrite* (iso. joy.; front)	7.74	*NR*	1.34	*NR*	*NA*	
			Multi-Tap (phone keypad)	8.83	*NR*	0.52	*NR*		
			EdgeWrite (iso joy; front; *Fisch*)	13.65	*NR*	0.35	*NR*		
			T9 (phone keypad)	14.63	*NR*	0.28	*NR*		
			EdgeWrite (iso. joy.; back)	6.34	*NR*	4.38	*NR*		
			EdgeWrite (iso joy; back; *Fisch*)	11.11	*NR*	0.20	*NR*		
			Input Blind:						
			EdgeWrite (iso. joy.; front)	8.09	*NR*	2.96	*NR*		
			Multi-Tap (phone keypad)	3.09	*NR*	1.58	*NR*		
Castellucci and MacKenzie (2008) [12]	Chars	A stylus is used to draw simple, single-stroke gestures representing English letters	*Unistrokes* (stylus)	4.1	15.8	43.4	16.3	20 sess. (5 hrs)	2
			Graffiti (stylus)	4.0	11.4	26.2	26.2		
Castellucci and MacKenzie (2008) [13]	Midair	Text is entered by combining vertical, horizontal, and rolling motions with the *Wiimote*	*Unigest* (*Wiimote*)	*NR*	*NR*	*NR*	*NR*	*NA*	
Martin and Isokoski (2008) [74]	Char, Menu	*EdgeWrite* gestures are entered with the help of onscreen characters; characters are displayed in the corner associated with the next motion; hints are static, dynamic (i.e., animated), or shown on paper	*EdgeWrite* (joystick; dynamic)	2.5	6.7	1.3	0.4	1 hour	3,5
			EdgeWrite (joystick; paper)	2.1	5.7	1.0	0.5		
			EdgeWrite (joystick; static)	1.5	5.0	1.1	0.8		

Authors	Input Type	Description	Technique (input device)	Speed (wpm) Initial	Speed (wpm) Skilled	Error Rate (%) Initial	Error Rate (%) Skilled	Practice	Notes
Shoemaker et al. (2009) [102]	Midair	Remote pointing is used to input text using onscreen keyboards/menus (distance in feet); the "Cube" condition is a 3D extension of the *T-Cube* technique for gestural text entry	Circle (*Wiimote*, 8') Cube (*Wiimote*, 8') QWERTY (*Wiimote*, 8') Circle (*Wiimote*, 9') Circle (*Wiimote*, 18') QWERTY (*Wiimote*, 9') QWERTY (*Wiimote*, 18')	10.2 7.6 18.9 11.6 10.0 14.5 10.3	*NR* *NR* *NR* *NR* *NR* *NR* *NR*	6.3 7.0 2.4 8.9 14.1 8.5 19.0	*NR* *NR* *NR* *NR* *NR* *NR* *NR*	*NA*	
Arif et al. (2010) [1]	Tap	An evaluation of the *iPhone*'s QWERTY soft keyboard	QWERTY (two thumbs) QWERTY (two thumbs, haptic feedback)	15.92 16.27	*NR* *NR*	*10.38* *9.46*	*NR* *NR*	*NA*	4
Isokoski et al. (2010) [41]	Tap, Char	Tapping on a soft keyboard is augmented with *Unistroke* shortcuts	*UniKeyb*	7	51	*NR*	4	36 sess. (3 hrs)	3,5
MacKenzie et al. (2011) [66]	Menu	Huffman codes are generated using a language model and mapped to four gamepad keys	*H4* (gamepad)	7.7	20.4	0.34	0.89	10 sess. (400 min.)	3,5
Castellucci and MacKenzie (2013) [15]	Tap, Char, Menu	An evaluation of four mobile text entry techniques using a novel application to gather metrics	QWERTY (*Android*, finger) QWERTY (*Android*, thumbs) *DioPen* (*Graffiti*-like) *Swype* (*ShapeWriter*-like)	20.9 20.8 7.0 16.7	*NR* *NR* *NR* *NR*	7.1 13.8 30.4 7.0	*NR* *NR* *NR* *NR*	*NA*	4
Castellucci and MacKenzie (2013) [16]	Menu, Midair	H4 encodings are mapped to touchpad and midair regions	*H4* (touchpad) *H4* (*Wiimote*)	6.6 5.3	*NR* *NR*	9.2 10.8	*NR* *NR*	*NA*	4

Authors	Input Type	Description	Technique (input device)	Speed (wpm) Initial	Speed (wpm) Skilled	Error Rate (%) Initial	Error Rate (%) Skilled	Practice	Notes
Castellucci and MacKenzie (2013) [17]	Tap	The effect of audio and haptic feedback are investigated using a QWERTY soft keyboard	QWERTY (*Android*, two thumbs) QWERTY (audio feedback) QWERTY (haptic feedback) QWERTY (audio and haptic)	29.9 29.9 28.7 30.3	*NR* *NR* *NR* *NR*	9.7 10.3 10.2 10.7	*NR* *NR* *NR* *NR*	*NA*	4
Cuaresma and MacKenzie (2013) [21]	Tap, Menu	An evaluation of four QWERTY-like soft keyboards; *Octopus* adds shortcut gestures for frequent words; *Curve* is *ShapeWriter*-like; *T+* groups two letters to a key	QWERTY (*iOS*, two thumbs) *Octopus* (two thumbs) TouchPal *Curve* (one thumb) TouchPal *T+* (two thumbs)	54.0 54.7 35.3 38.7	*NR* *NR* *NR* *NR*	4.6 1.9 5.4 4.1	*NR* *NR* *NR* *NR*	*NA*	6
Fitton et al. (2013) [28]	Menu, Midair	Tilting a tablet moves a ball cursor; letters are entered by dwelling over them	Sitting, one-handed Sitting, two-handed Walking, one-handed Walking, two-handed	9.0 10.0 7.5 8.0	*NR* *NR* *NR* *NR*	4.5 3.0 7.0 5.0	*NR* *NR* *NR* *NR*	*NA*	1,3,5
Oulasvirta et al. (2013) [83]	Tap	Characters are arranged on two halves of a tablet keyboard; one half at the bottom-left corner, the other at the bottom-right corner	QWERTY KALQ	27.7 *NR*	*NR* 37.1	9.0 *NR*	*NR* 5.2	13-19 hrs	5
Arif et al. (2014) [2]	Tap, Char	Tapping on a soft keyboard is augmented with gestures for space, backspace, shift, and enter	QWERTY lowercase QWERTY mixed case *New* lowercase *New* mixed case	20.78 14.51 20.10 17.35	*NR* *NR* *NR* *NR*	12.0 8.21 12.0 8.98	*NR* *NR* *NR* *NR*	*NA*	4
Fuccella et al. (2014) [30]	Tap, Menu	Long-holding a key enters its letter and brings up four additional letters around it; gesturing to those letters enters them in succession	QWERTY *KeyScretch*	*NR* *NR*	31.8 37.4	*NR* *NR*	3.47 3.80	25 sess. (6 hrs)	4

Authors	Input Type	Description	Technique (input device)	Speed (wpm)		Error Rate (%)		Practice	Notes
				Initial	Skilled	Initial	Skilled		

Notes:
1. Converted to wpm assuming 5-character words (including spaces) [143].
2. Error rate based on reported correction rate.
3. Values approximated from graphs.
4. Total Error Rate metric used.
5. MSD error rate metric used.
6. Character-wise error rate metric used.

NA: Value not applicable.
NR: Value not reported.

Appendix B
List of TEMA Users

Industry

Naveen Durga, KeyPoint Technologies Ltd.

Aidan Kehoe, Logitech, Inc.

Motamedi Nima, Motorola, Inc.

Curtis Ray, Tactus Technology Inc.

Philip Strain, Google Inc.

Donnelle R. Weller, Sprint Nextel Corporation

Academia

Nikola Banovic, Carnegie Mellon University, USA

Mike Clarke, University of Washington, USA

James Clawson, Georgia Institute of Technology, USA

Mark Dunlop, University of Strathclyde, UK

Vittorio Fuccella, University of Salerno, Italy

Michael Geary, Colorado Technical University, USA

Mayank Goel, University of Washington, USA

Jibo He, Wichita State University, USA

Niels Henze, University of Stuttgart, Germany

Kheng Hui, University of Osnabrück, Germany

Poika Isokoski, University of Tampere, Finland

Anirudha Joshi, Indian Institute of Technology Bombay, India

Erno Mäkinen, University of Tampere, Finland

Benoît Martin, University of Lorraine, France

Alexander Ng, University of Glasgow, UK

Janet C. Read, University of Central Lancashire, UK

Amanda Smith, Wichita State University, USA

Robert Teather, McMaster University, Canada

Anju Thapa, University of Tampere, Finland

Sandy Tran, University of Toronto, Canada

Simon Whatley, University College London, UK

Hui-Shyong Yeo, Korea Advanced Institute of Science and Technology, South Korea

Appendix C
Samples from the Mobile Phrase Set

C.1 Email Phrase Set

Phrase length (in characters)	Min: 20	Max: 35	Ave: 30.0
Number of tokens:	2951 (1098 unique)		
Average tokens per phrase:	5.9		
Average token length (in characters):	4.3		

C.1.1 Frequencies (Top 10)

`··`	4.57%	`·`	17.70%
`e·`	2.48%	`e`	8.65%
`·t`	1.88%	`t`	6.43%
`th`	1.43%	`o`	5.75%
`s·`	1.37%	`a`	5.73%
`·a`	1.37%	`i`	5.24%
`t·`	1.27%	`n`	5.19%
`in`	1.20%	`r`	4.39%
`he`	1.15%	`s`	4.35%
`d·`	1.07%	`l`	2.97%

C.1.2 Samples

Hope things are going well.
I'm not sure it's your style.
Would you mind to handle this one?
You cannot change the past.
Sorry, but we're swamped.
Here's a simple first draft.
Aah, thanks for the clarification.
Do you need anything else?
The cost of the seminar is $397.
Can we change the arrival to 10/21?
Andy: Just checking on the options.
Comments due by November 22, 2000.
That will make 562,003 to 1.

Mark: Here's the other email.
Can we meet from 10am-11am instead?
The P&L showed this deal at $40.
The term is 4/1/00 through 3/31/01.
Am I taking care of you or what??
You're right; I can't open the doc.
Folks: Here are my edits.
I don't have any comments to add.
I really need to get a life.
Let's try to get this thing signed.
Let's shoot for lunch next Tuesday.
It is presently trading around $30.
They are filing lawsuits.
How should I handle this?
What did you have in mind, John?
I appreciate your efforts.
Thanks for your assistance.

C.2 SMS Phrase Set

Phrase length (in characters)	Min: 25	Max: 35	Ave: 29.6
Number of tokens:	3152 (1366 unique)		
Average tokens per phrase:	6.3		
Average token length (in characters):	3.9		

C.2.1 Frequencies

`e·`	2.36%	`·`	17.96%
`t·`	1.84%	`e`	7.65%
`·t`	1.73%	`a`	6.73%
`.·`	1.63%	`o`	6.33%
`in`	1.24%	`t`	5.89%
`·a`	1.20%	`i`	4.96%
`·s`	1.11%	`n`	4.94%
`ha`	1.10%	`h`	4.57%
`s·`	1.09%	`s`	4.02%
`o·`	1.07%	`r`	3.68%

C.2.2 Samples

i have no money 4 steve mate! !
yeah excited! whachudoin?
thanks for loving me so. you rock
i thought it was a box of cutlery
good nite. i thought u'l talk 2 me.
hahaha where are you now?
hey, thanx for helping me today.
hey tmr can save an extra seat?
haha how much will it cost?
i don't know the next line
how cheap is your cheapness?
wts. sick. tis year trackers own
go home safe!!(: thanks for today!
ok. but i shld b doing hmwk le.
yes ok. will do in a bit.
cn i gv u a cl at ur lnd ph?
vry gud mornin.. hav a g8 day :-)
call me when u get a chance
u done let me noe, my sis is back.
what happened to calling me back?
cool no problem.. cya :-)
nope but i'll be going next week!(:
can i giv this it @ 2mrw eveng .
thanks for the quick reply. :-)
yes! i'm already losing my hair :(
ah._. i just got your sms ._.
stupid auto correct on my phone
ok lor. msg me b4 u call.
lol u believe meh hahaha.
yup wat time r they going?

C.3 Twitter Phrase Set

Phrase length (in characters)	Min: 17	Max: 35	Ave: 28.0
Number of tokens:	2641 (1344 unique)		
Average tokens per phrase:	5.3		
Average token length (in characters):	4.5		

C.3.1 Frequencies

`e·`	1.71%	`·`	12.57%
`t·`	1.13%	`e`	7.34%
`s·`	1.06%	`t`	6.62%
`·t`	1.05%	`a`	5.93%
`in`	1.01%	`o`	5.87%
`er`	0.87%	`i`	4.92%
`·@`	0.82%	`s`	4.38%
`th`	0.81%	`n`	4.31%
`an`	0.80%	`r`	4.09%
`·a`	0.78%	`l`	3.31%

C.3.2 Samples

and every day i love you more
@ziemniak_ follow back ;*
@chagreyson ahw :'(xd
howhowhow. things stressing me out!
fake it till you make it.
@sydney4 luv you more sunshine
@hannah love you:*
i may act dumb, but i'm not dumb
you pick me up when i fall down
@bangbang dont be lazy
@kolby thanks, means a lot
dont worry, bae you got me <3
touch my snacks #waystogetslapped
#bestmanholiday awesome movie!
my mommys' coming home! ! ^_^
@brooksbeau you do beau and ily <3
i plan to relax all weekend. #yay
i trusted you, my mistake (;
@emma tell matt he's pretty
i'm still up. #breakingbad
163 emma watson #forbes30
#lrt amen! say it one more time!
i'm gonna find another youu
i want chocolate milk brb
ahahah i cant breathing!!!! #music

tamar & vince @ 9 and scandal @10
#np my morning jacket - rocket man
maybe its a good day for me.
i don't know how to feel right now
feeling proud i have a 3.8 gpa

Appendix D
Key Selection Times

The tables on the subsequent pages contain the key selection times from the user study described in Chapter 6. The times (reported in milliseconds) represents the duration between selecting the first target (designated *T1*) and the second one (*T2*), averaged for both left and right hand input. The following figure illustrates the location of each key and its identifying number.

Figure 59. The layout and ID for each key in the MIME layout.

Table 10. The mean (and SD) of selection times from key *T1* to *T2*, measured in milliseconds. This table covers T1 = 1..15, T2 = 1..16.

Mean (SD)	T1: 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
T2:1	275	368	394	462	458	611	365	349	432	449	482	522	365	422	459
	(147)	(297)	(186)	(210)	(155)	(331)	(247)	(162)	(420)	(232)	(202)	(249)	(154)	(263)	(437)
2	305	245	287	349	425	467	322	293	309	376	390	499	400	338	371
	(134)	(171)	(100)	(145)	(267)	(221)	(158)	(121)	(113)	(172)	(164)	(265)	(230)	(152)	(268)
3	321	320	234	279	453	436	399	299	287	305	407	455	411	351	312
	(117)	(182)	(186)	(114)	(392)	(249)	(247)	(141)	(135)	(222)	(233)	(230)	(194)	(179)	(108)
4	443	421	309	209	282	410	430	396	291	287	284	404	459	358	363
	(235)	(403)	(184)	(63)	(94)	(221)	(219)	(232)	(115)	(166)	(94)	(252)	(202)	(138)	(169)
5	498	467	377	384	219	300	515	402	382	327	281	312	445	425	426
	(246)	(417)	(296)	(386)	(89)	(118)	(314)	(187)	(243)	(209)	(97)	(127)	(181)	(206)	(319)
6	565	518	420	429	317	232	604	481	455	441	357	368	522	503	532
	(284)	(338)	(247)	(331)	(167)	(103)	(320)	(247)	(294)	(309)	(224)	(289)	(249)	(282)	(321)
7	323	361	505	497	534	637	239	316	431	511	552	527	319	347	385
	(171)	(206)	(392)	(281)	(255)	(592)	(86)	(119)	(369)	(437)	(433)	(217)	(171)	(236)	(177)
8	378	300	320	369	505	584	298	208	267	316	456	484	323	273	313
	(265)	(135)	(150)	(170)	(240)	(374)	(132)	(54)	(81)	(97)	(280)	(337)	(147)	(96)	(149)
9	417	375	348	324	413	437	357	318	210	298	341	435	360	308	257
	(276)	(290)	(291)	(179)	(304)	(191)	(177)	(216)	(79)	(195)	(151)	(213)	(150)	(183)	(80)
10	432	429	342	314	326	426	401	413	308	236	306	420	483	344	313
	(231)	(259)	(221)	(171)	(148)	(231)	(249)	(310)	(182)	(140)	(137)	(229)	(362)	(148)	(354)
11	510	516	469	336	345	322	461	390	320	314	205	321	478	397	358
	(221)	(391)	(335)	(185)	(192)	(174)	(217)	(170)	(133)	(200)	(48)	(172)	(231)	(186)	(207)
12	564	453	490	397	334	339	472	487	409	376	316	219	511	440	430
	(333)	(183)	(339)	(199)	(179)	(186)	(237)	(252)	(157)	(179)	(156)	(67)	(208)	(189)	(225)
13	446	388	411	459	562	528	322	371	430	478	458	539	255	325	378
	(408)	(185)	(143)	(218)	(350)	(206)	(189)	(218)	(331)	(321)	(177)	(298)	(159)	(181)	(216)
14	418	330	377	391	501	497	304	347	301	392	475	510	342	212	298
	(223)	(117)	(165)	(157)	(234)	(235)	(116)	(297)	(148)	(282)	(291)	(311)	(273)	(71)	(146)
15	429	388	455	391	492	481	432	326	288	313	401	439	390	291	221
	(227)	(248)	(297)	(247)	(395)	(213)	(232)	(200)	(151)	(153)	(250)	(195)	(218)	(146)	(82)
16	432	441	391	349	358	447	553	432	351	298	341	397	447	364	293
	(188)	(269)	(255)	(167)	(167)	(212)	(376)	(300)	(351)	(154)	(208)	(227)	(268)	(199)	(123)

Table 11. The mean (and SD) of selection times from key *T1* to *T2*, measured in milliseconds. This table covers T1 = 16..30, T2 = 1..16.

Mean (SD)	T1: 16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
T2:1	424 (147)	548 (340)	571 (310)	499 (399)	437 (273)	504 (276)	483 (288)	567 (561)	585 (299)	559 (389)	632 (470)	520 (231)	524 (258)	514 (169)	600 (282)
2	406 (193)	425 (172)	495 (215)	472 (390)	388 (208)	387 (172)	414 (223)	499 (266)	519 (192)	525 (327)	434 (189)	437 (151)	478 (225)	531 (233)	503 (170)
3	358 (182)	421 (227)	437 (170)	486 (367)	384 (163)	387 (190)	406 (223)	395 (191)	490 (239)	501 (277)	487 (381)	417 (193)	478 (298)	443 (162)	535 (280)
4	326 (146)	376 (241)	431 (216)	475 (344)	416 (240)	393 (210)	396 (218)	360 (138)	422 (167)	506 (294)	461 (222)	409 (152)	430 (222)	427 (147)	584 (297)
5	374 (255)	349 (192)	344 (136)	485 (181)	452 (233)	445 (244)	358 (107)	399 (210)	385 (143)	533 (283)	516 (289)	437 (134)	469 (293)	442 (176)	550 (329)
6	434 (251)	377 (187)	370 (188)	538 (206)	538 (295)	453 (210)	455 (206)	600 (834)	430 (218)	527 (183)	549 (255)	479 (182)	567 (331)	525 (284)	523 (267)
7	407 (156)	455 (134)	479 (132)	336 (144)	388 (233)	401 (214)	501 (351)	490 (217)	566 (216)	439 (214)	447 (267)	419 (143)	503 (240)	499 (185)	578 (373)
8	388 (232)	392 (135)	494 (241)	371 (174)	349 (219)	374 (229)	380 (157)	409 (155)	528 (263)	406 (181)	405 (234)	426 (297)	490 (446)	466 (230)	532 (253)
9	294 (127)	353 (184)	452 (317)	372 (148)	389 (322)	307 (106)	349 (190)	391 (193)	467 (203)	429 (225)	406 (221)	406 (267)	416 (247)	412 (183)	460 (183)
10	287 (154)	313 (142)	349 (131)	439 (204)	385 (210)	315 (112)	357 (308)	323 (123)	373 (163)	473 (245)	398 (169)	433 (275)	352 (136)	389 (182)	440 (161)
11	305 (194)	336 (227)	325 (157)	466 (227)	421 (207)	388 (158)	326 (113)	332 (148)	373 (205)	510 (279)	422 (172)	396 (130)	417 (224)	394 (201)	554 (520)
12	340 (124)	312 (134)	284 (118)	516 (233)	492 (303)	423 (191)	426 (268)	390 (239)	388 (256)	552 (234)	514 (239)	442 (153)	483 (253)	446 (193)	485 (267)
13	425 (219)	453 (203)	551 (341)	284 (98)	310 (112)	391 (204)	504 (660)	474 (184)	528 (182)	395 (273)	390 (186)	384 (113)	460 (319)	457 (153)	581 (320)
14	343 (154)	413 (190)	475 (219)	323 (175)	269 (116)	290 (98)	348 (126)	430 (227)	479 (203)	408 (254)	339 (232)	322 (124)	446 (304)	451 (162)	535 (226)
15	279 (109)	422 (281)	443 (241)	390 (194)	323 (220)	256 (73)	283 (86)	359 (155)	486 (263)	465 (310)	378 (216)	329 (159)	350 (144)	412 (278)	548 (390)
16	197 (53)	303 (168)	367 (187)	451 (238)	417 (246)	314 (170)	266 (111)	305 (126)	423 (235)	499 (313)	417 (221)	314 (120)	295 (99)	355 (189)	447 (229)

Table 12. The mean (and SD) of selection times from key *T1* to *T2*, measured in milliseconds. This table covers T1 = 1..15, T2 = 17..30.

Mean (SD)	T1: 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
T2: 17	528	451	387	372	384	411	467	418	407	359	314	331	454	398	370
	(215)	(222)	(139)	(189)	(223)	(206)	(203)	(214)	(270)	(263)	(153)	(171)	(197)	(177)	(208)
18	546	465	512	411	432	375	505	449	464	406	325	322	508	472	403
	(271)	(174)	(280)	(161)	(271)	(167)	(206)	(182)	(216)	(225)	(139)	(158)	(231)	(243)	(167)
19	499	432	542	484	593	625	400	421	439	453	496	564	354	369	441
	(389)	(208)	(300)	(232)	(358)	(327)	(235)	(296)	(232)	(197)	(263)	(218)	(312)	(266)	(333)
20	426	396	427	456	464	513	374	331	355	456	498	491	327	324	322
	(203)	(165)	(189)	(209)	(201)	(164)	(163)	(138)	(168)	(314)	(305)	(193)	(149)	(173)	(150)
21	473	435	455	400	477	535	408	404	350	330	403	419	409	346	275
	(240)	(191)	(236)	(183)	(340)	(299)	(193)	(256)	(239)	(127)	(217)	(142)	(222)	(266)	(112)
22	507	496	410	470	411	516	503	430	415	317	354	456	454	380	358
	(194)	(282)	(238)	(277)	(206)	(266)	(280)	(227)	(252)	(149)	(170)	(222)	(248)	(204)	(328)
23	513	480	444	441	438	466	525	468	399	405	318	386	462	487	343
	(187)	(211)	(186)	(271)	(228)	(293)	(275)	(264)	(194)	(288)	(120)	(194)	(185)	(448)	(133)
24	535	525	488	453	435	486	577	496	485	454	492	389	533	512	419
	(186)	(187)	(243)	(156)	(184)	(337)	(261)	(164)	(258)	(225)	(395)	(182)	(204)	(321)	(208)
25	678	646	789	633	668	771	554	488	531	546	608	681	521	592	559
	(560)	(644)	(1313)	(319)	(340)	(792)	(701)	(367)	(272)	(227)	(336)	(535)	(638)	(992)	(590)
26	538	512	514	496	561	576	504	441	469	467	463	548	391	373	372
	(379)	(473)	(292)	(212)	(311)	(259)	(436)	(252)	(303)	(228)	(165)	(193)	(191)	(155)	(149)
27	534	462	492	482	493	562	467	456	390	409	479	494	402	406	388
	(302)	(170)	(208)	(181)	(207)	(300)	(212)	(271)	(153)	(222)	(209)	(209)	(158)	(225)	(199)
28	522	474	446	476	466	519	502	483	423	470	396	495	503	393	388
	(228)	(192)	(136)	(212)	(206)	(237)	(213)	(292)	(225)	(366)	(163)	(284)	(336)	(166)	(246)
29	563	547	496	474	484	523	597	548	443	446	445	498	560	456	446
	(228)	(260)	(213)	(203)	(240)	(283)	(365)	(327)	(175)	(207)	(298)	(273)	(243)	(191)	(285)
30	671	590	648	518	540	555	630	554	526	644	573	555	601	504	465
	(312)	(265)	(489)	(200)	(221)	(375)	(330)	(260)	(336)	(629)	(365)	(388)	(309)	(189)	(183)

Table 13. The mean (and SD) of selection times from key *T1* to *T2*, measured in milliseconds. This table covers T1 = 16..30, T2 = 17..30.

Mean (SD)	T1: 16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
T2: 17	315	233	346	447	425	311	289	284	359	532	424	411	359	360	429
	(249)	(89)	(184)	(210)	(275)	(97)	(108)	(204)	(235)	(407)	(170)	(257)	(202)	(212)	(255)
18	395	291	266	507	432	402	405	319	275	539	450	476	382	354	387
	(208)	(89)	(192)	(181)	(180)	(142)	(216)	(165)	(84)	(229)	(148)	(269)	(165)	(135)	(257)
19	466	535	617	257	304	386	504	569	529	305	341	452	469	504	540
	(213)	(313)	(805)	(128)	(98)	(164)	(471)	(654)	(186)	(166)	(159)	(475)	(220)	(231)	(208)
20	375	413	520	318	225	283	338	400	538	348	294	335	408	505	582
	(169)	(188)	(264)	(228)	(127)	(96)	(154)	(148)	(386)	(223)	(152)	(203)	(241)	(324)	(313)
21	305	420	531	371	316	272	306	362	503	415	323	292	311	379	463
	(130)	(255)	(349)	(174)	(201)	(197)	(173)	(183)	(394)	(216)	(162)	(162)	(160)	(162)	(230)
22	290	322	466	453	417	284	213	295	408	441	362	341	297	326	401
	(181)	(189)	(370)	(232)	(351)	(135)	(78)	(147)	(265)	(217)	(135)	(249)	(173)	(278)	(182)
23	322	327	375	478	452	375	281	225	346	531	458	406	307	300	336
	(200)	(198)	(220)	(220)	(235)	(219)	(137)	(105)	(168)	(289)	(238)	(265)	(134)	(166)	(167)
24	395	336	449	500	440	446	399	320	268	556	470	424	387	358	304
	(243)	(174)	(432)	(196)	(150)	(218)	(261)	(150)	(184)	(255)	(195)	(162)	(154)	(206)	(132)
25	524	658	638	492	454	552	577	689	603	308	502	528	567	569	656
	(292)	(838)	(471)	(1116)	(455)	(659)	(510)	(1025)	(405)	(286)	(784)	(391)	(493)	(387)	(584)
26	472	486	531	340	318	368	387	555	592	321	267	330	397	491	559
	(459)	(266)	(306)	(171)	(137)	(224)	(181)	(467)	(348)	(196)	(150)	(181)	(215)	(224)	(254)
27	393	416	470	386	328	318	402	427	501	414	398	228	308	422	514
	(257)	(162)	(192)	(144)	(158)	(160)	(317)	(218)	(211)	(200)	(261)	(127)	(120)	(346)	(270)
28	350	404	420	462	409	368	322	339	400	462	416	319	224	314	465
	(152)	(194)	(214)	(235)	(198)	(297)	(167)	(172)	(175)	(243)	(256)	(251)	(91)	(152)	(443)
29	409	460	375	525	427	419	337	303	375	522	454	405	321	266	375
	(243)	(352)	(149)	(363)	(185)	(219)	(137)	(117)	(209)	(238)	(208)	(217)	(180)	(149)	(271)
30	479	425	404	567	631	497	432	380	388	561	624	452	467	368	307
	(226)	(214)	(196)	(364)	(568)	(270)	(217)	(188)	(310)	(233)	(457)	(174)	(348)	(273)	(232)

Appendix E
MIME Evaluation Results

Table 14. The entry speed measurements (in wpm) from each participant for both techniques and each of the ten sessions.

Tech.	QWERTY										MIME									
Sess.	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
1	18.48	20.06	20.93	19.84	21.94	21.14	21.90	19.54	14.80	21.52	8.03	10.18	12.26	12.02	14.66	14.20	15.25	14.92	12.69	16.52
2	21.04	23.63	21.86	26.32	24.48	25.56	26.14	26.71	25.24	24.77	10.41	12.45	14.87	16.43	16.03	16.62	17.32	15.94	16.56	19.01
3	23.29	23.41	24.78	24.63	24.76	19.83	23.21	23.21	25.98	24.93	7.68	8.53	9.71	10.70	10.55	10.77	12.63	14.13	13.61	15.68
4	21.33	20.66	23.29	23.62	24.31	22.94	23.71	23.76	22.41	22.08	10.34	11.94	13.69	14.23	16.09	15.50	16.91	16.86	18.75	18.53
5	26.53	28.01	22.86	26.89	26.81	26.43	28.51	27.34	27.15	25.29	8.98	13.14	12.72	12.53	14.24	15.77	15.79	16.27	17.45	16.98
6	22.09	23.31	20.94	22.68	24.42	23.85	21.02	20.46	23.16	22.90	8.43	9.81	10.44	12.52	13.84	14.51	15.00	14.02	14.75	16.32
Mean	22.13	23.18	22.44	24.00	24.45	23.29	24.08	23.50	23.12	23.58	8.98	11.01	12.28	13.07	14.23	14.56	15.49	15.36	15.64	17.17
SD	2.68	2.82	1.50	2.58	1.55	2.53	2.79	3.17	4.44	1.62	1.16	1.77	1.94	2.00	2.03	2.05	1.67	1.18	2.34	1.32

Table 15. The total error rate (in %) from each participant for both techniques and each of the ten sessions.

Tech.	QWERTY										MIME									
Sess.	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
1	3.59	3.03	3.99	3.66	3.63	2.44	3.54	1.43	4.90	4.69	1.80	1.21	1.07	2.09	1.51	2.14	1.50	0.96	2.89	2.00
2	5.12	3.05	5.48	4.18	4.97	4.85	3.91	5.12	4.50	5.33	1.11	0.88	1.59	0.93	1.49	1.54	2.26	1.30	1.40	1.87
3	3.07	3.64	2.39	2.98	2.66	3.66	4.05	2.84	3.90	3.50	1.29	1.35	2.39	2.68	3.88	2.14	2.82	1.38	2.62	1.47
4	7.24	5.60	6.82	5.99	7.26	7.51	8.58	4.01	9.36	6.37	1.56	2.05	2.11	2.34	1.18	1.95	3.88	1.71	1.21	2.77
5	4.26	4.50	9.85	4.37	3.76	4.48	5.29	3.70	4.74	6.03	0.72	1.22	1.70	1.41	2.09	1.13	1.42	1.29	1.32	1.48
6	4.69	4.34	7.36	5.11	5.06	3.91	5.65	10.30	3.11	5.10	0.74	0.94	2.17	0.96	1.27	0.91	1.16	2.32	1.33	0.60
Mean	4.66	4.03	5.98	4.38	4.56	4.48	5.17	4.57	5.09	5.17	1.20	1.28	1.84	1.74	1.90	1.64	2.17	1.49	1.80	1.70
SD	1.46	0.99	2.63	1.06	1.60	1.70	1.86	3.07	2.20	1.02	0.44	0.42	0.48	0.74	1.02	0.53	1.04	0.47	0.75	0.72

Table 16. The uncorrected error rate (in %) from each participant for both techniques and each of the ten sessions.

Tech.	QWERTY										MIME									
Sess.	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
1	0.88	0.63	2.30	1.76	2.68	1.40	2.19	0.85	2.39	3.90	0.39	0.83	0.82	1.31	0.75	1.17	0.53	0.32	2.08	1.50
2	0.90	0.23	0.99	1.09	1.04	0.93	1.12	1.32	1.17	1.55	0.32	0.18	0.40	0.10	0.08	0.36	0.72	0.24	0.39	1.03
3	0.23	0.49	0.78	0.15	0.10	0.27	0.58	0.62	0.69	0.36	0.42	0.35	0.78	0.35	0.76	0.39	0.90	0.08	0.64	0.00
4	0.53	0.82	1.18	0.65	1.58	0.41	0.71	0.29	0.56	0.72	0.21	1.31	0.66	0.53	0.37	0.59	1.74	0.44	0.69	0.44
5	1.08	1.32	0.73	1.67	1.23	1.69	2.84	1.32	0.82	1.02	0.26	0.35	0.76	0.88	0.72	0.64	0.35	0.35	0.27	0.85
6	1.45	1.10	0.45	0.51	0.84	0.40	0.58	1.63	0.26	0.35	0.10	0.27	0.71	0.42	0.42	0.18	0.43	0.82	0.23	0.08
Mean	0.84	0.77	1.07	0.97	1.25	0.85	1.33	1.01	0.98	1.32	0.28	0.55	0.69	0.60	0.52	0.55	0.78	0.38	0.72	0.65
SD	0.42	0.40	0.65	0.65	0.86	0.59	0.96	0.50	0.75	1.35	0.12	0.44	0.15	0.43	0.28	0.34	0.51	0.25	0.70	0.58

Table 17. The corrected error rate (in %) from each participant for both techniques and each of the ten sessions.

Tech.	QWERTY										MIME									
Sess.	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
1	2.71	2.40	1.69	1.90	0.95	1.05	1.35	0.58	2.50	0.79	1.41	0.38	0.25	0.79	0.76	0.97	0.97	0.63	0.81	0.51
2	4.22	2.82	4.49	3.08	3.93	3.92	2.79	3.80	3.34	3.78	0.79	0.70	1.19	0.83	1.42	1.18	1.54	1.05	1.02	0.84
3	2.84	3.15	1.61	2.83	2.56	3.39	3.47	2.22	3.21	3.15	0.87	1.01	1.61	2.33	3.11	1.76	1.92	1.30	1.98	1.47
4	6.70	4.78	5.63	5.34	5.68	7.10	7.87	3.72	8.80	5.65	1.35	0.74	1.45	1.82	0.81	1.36	2.14	1.27	0.52	2.33
5	3.18	3.18	9.12	2.70	2.54	2.79	2.45	2.38	3.92	5.02	0.46	0.87	0.93	0.53	1.37	0.49	1.07	0.94	1.05	0.62
6	3.25	3.24	6.92	4.60	4.22	3.51	5.07	8.68	2.85	4.75	0.64	0.68	1.46	0.54	0.85	0.73	0.73	1.51	1.09	0.51
Mean	3.81	3.26	4.91	3.41	3.31	3.63	3.83	3.56	4.10	3.86	0.92	0.73	1.15	1.14	1.39	1.08	1.39	1.12	1.08	1.05
SD	1.51	0.81	2.96	1.29	1.65	1.98	2.33	2.77	2.35	1.75	0.38	0.21	0.50	0.75	0.89	0.45	0.56	0.31	0.49	0.72

Appendix F
Box Plots of Performance Results

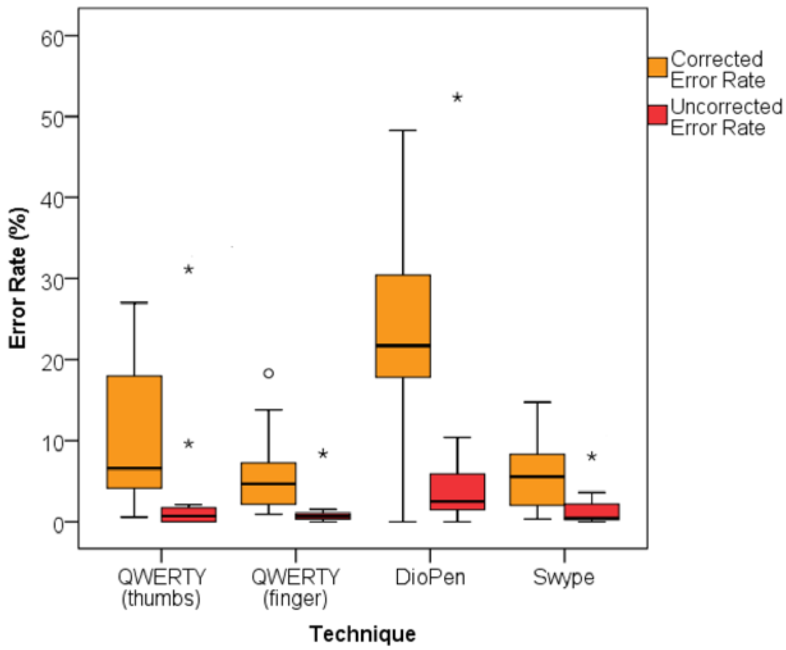


Figure 60. Accuracy values gathered by TEMA, corresponding to Figure 41.

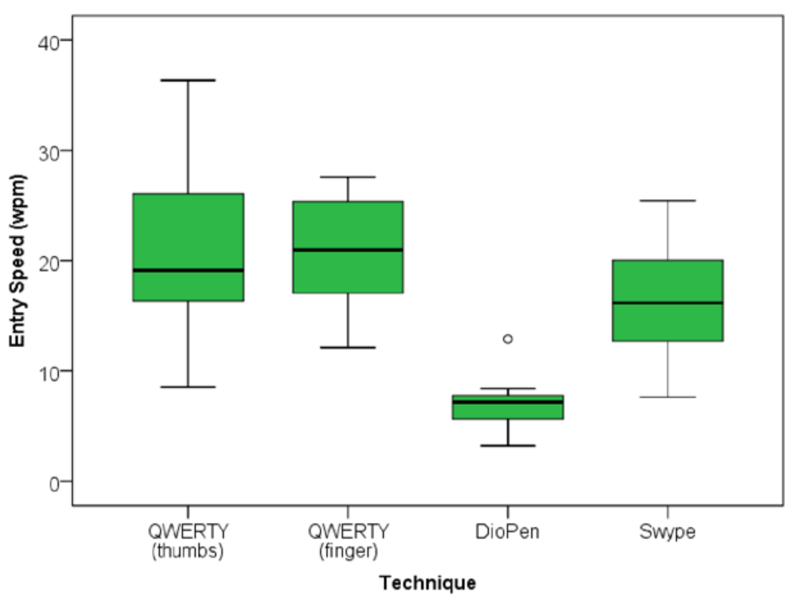


Figure 61. Accuracy values gathered by TEMA, corresponding to Figure 42.

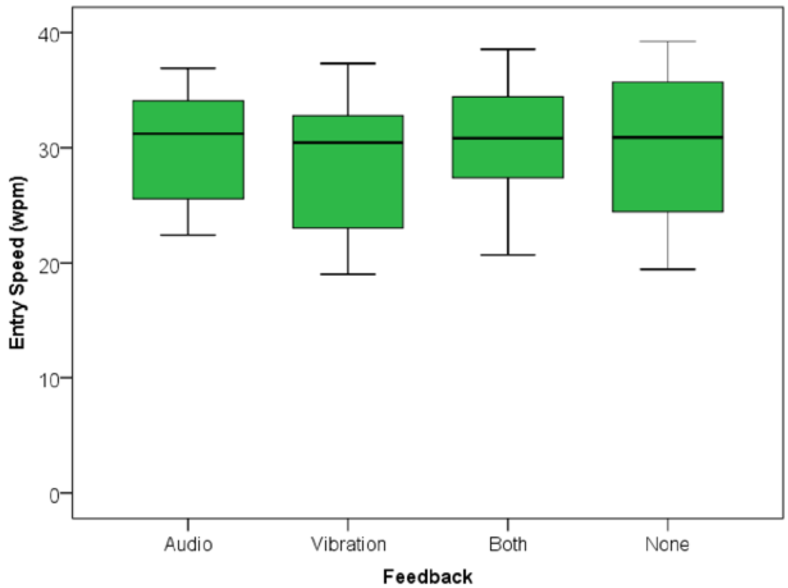


Figure 62. Entry speed values for the feedback user study, corresponding to Figure 44.

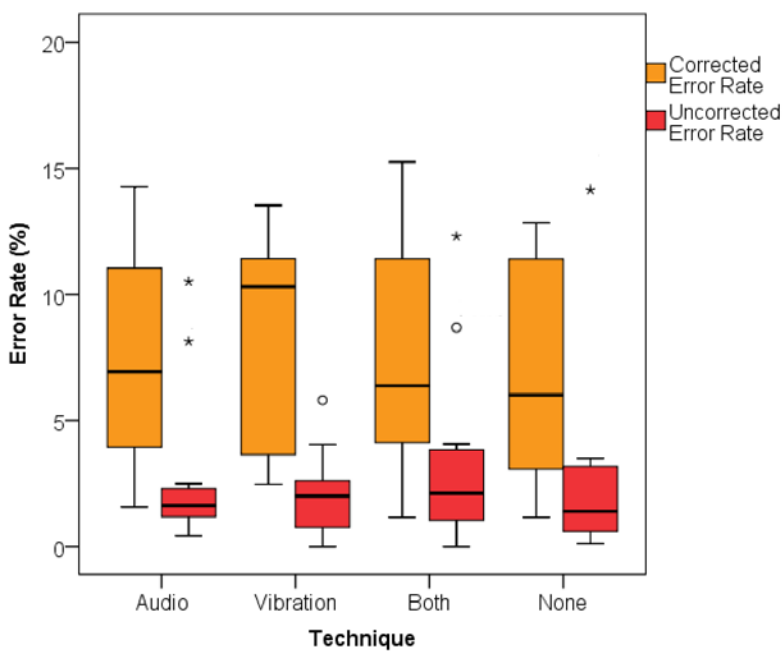


Figure 63. Entry speed values for the feedback user study, corresponding to Figure 45.

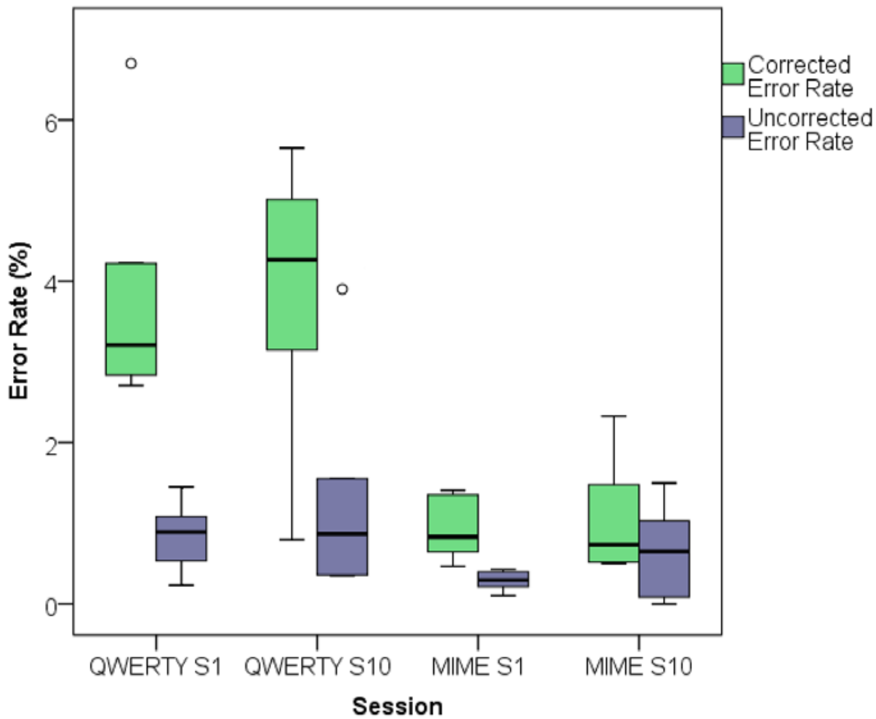


Figure 64. Error rates for the first and last MIME sessions, corresponding to Figure 57.

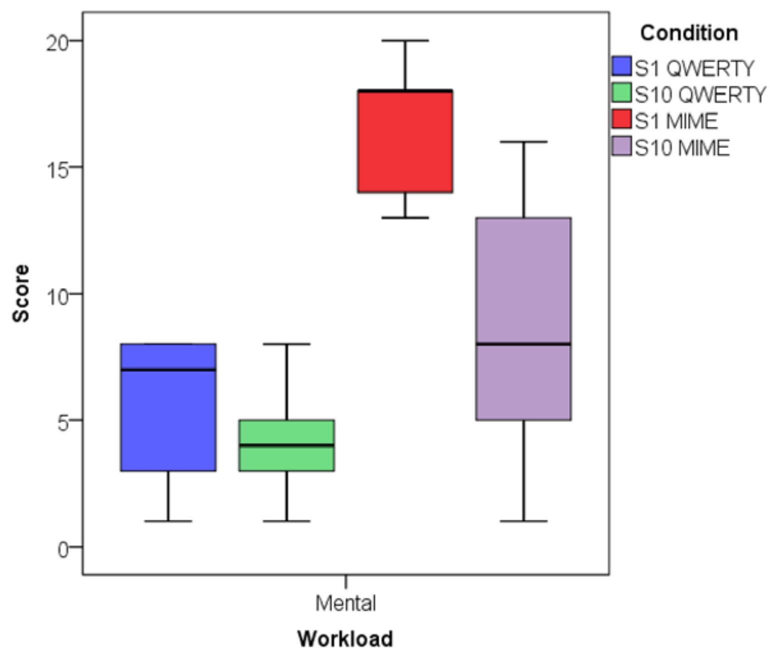


Figure 65. Mental workload scores for the MIME user study, corresponding to Figure 58.

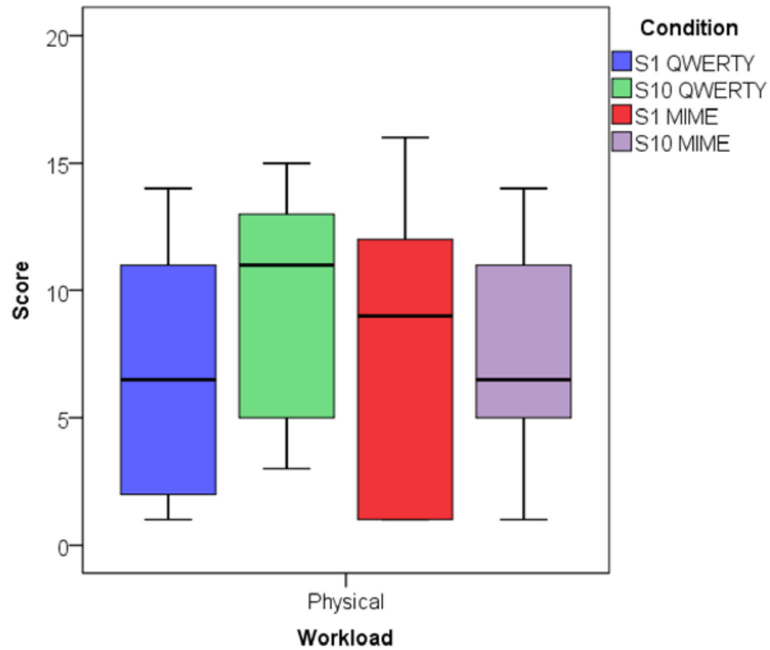


Figure 66. Physical workload scores for the MIME user study, corresponding to Figure 58.

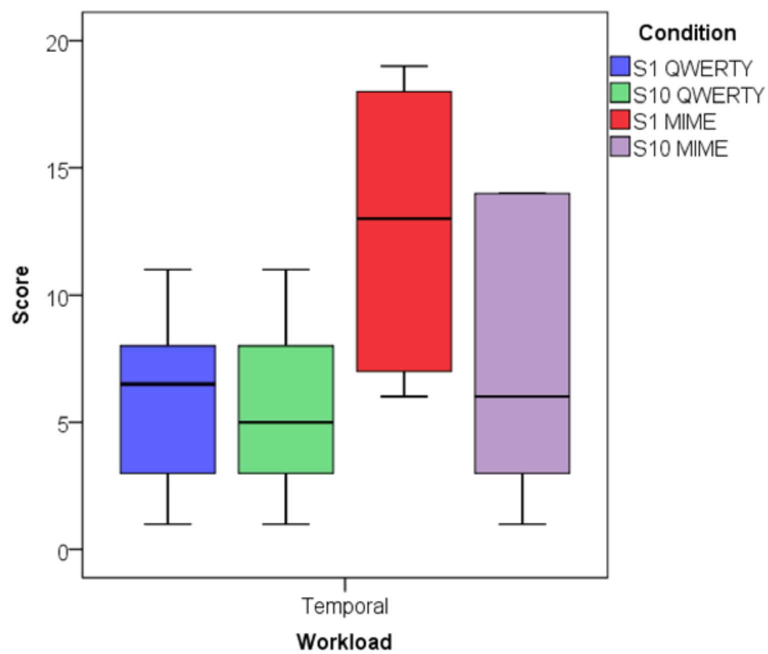


Figure 67. Temporal workload scores for the MIME user study, corresponding to Figure 58.

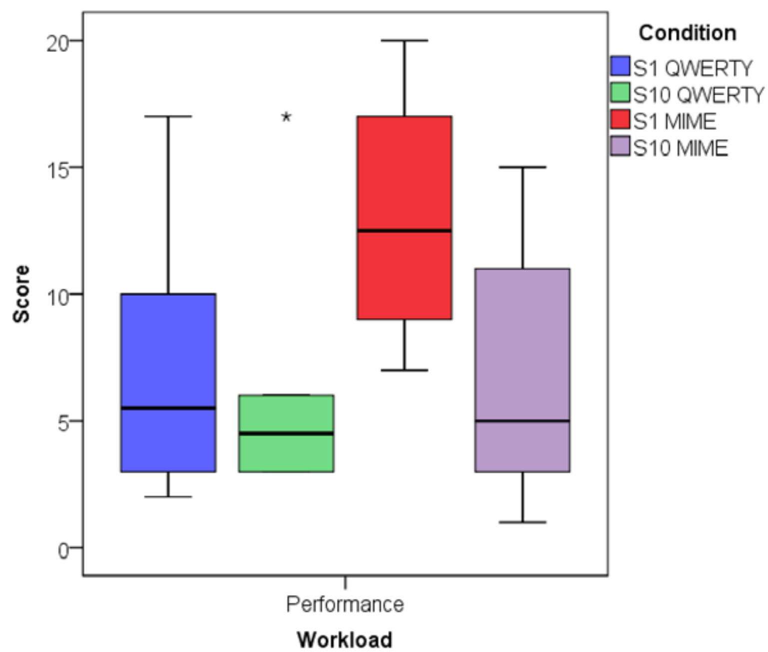


Figure 68. Performance workload scores for the MIME user study, corresponding to Figure 58.

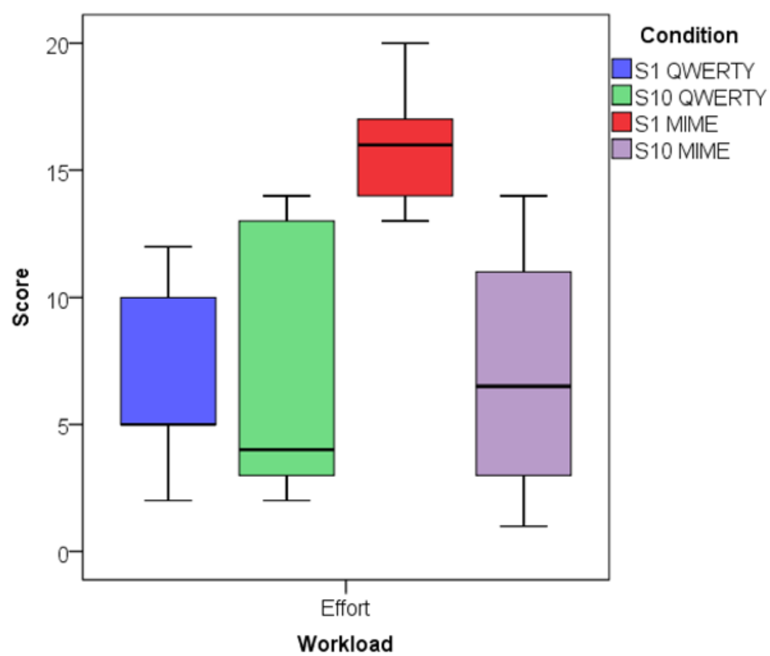


Figure 69. Effort workload scores for the MIME user study, corresponding to Figure 58..

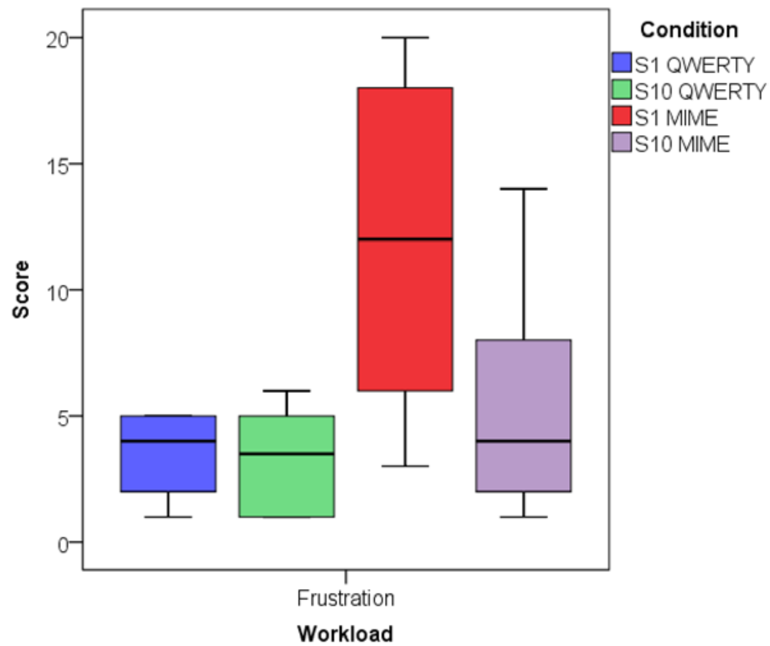


Figure 70. Frustration workload scores for the MIME user study, corresponding to Figure 58.

Bibliography

1. Arif, A. S., Lopez, M. H., and Stuerzlinger, W., Two new mobile touchscreen text entry techniques. In *Proceedings of Graphics Interface 2010 – GI 2010*, Toronto: Canadian Information Processing Society (2010), 22-23.

2. Arif, A. S., Pahud, M., Hinckley, K., and Buxton, W., Experimental study of stroke shortcuts for a touchscreen keyboard with gesture-redundant keys removed. In *Proceedings of Graphics Interface 2014 – GI 2014*, Toronto, Canada: Canadian Information Processing Society (2014), 43-50.

3. Arif, A. S. and Stuerzlinger, W., Analysis of text entry performance metrics. In *Science and Technology for Humanity (TIC-STH), 2009 IEEE Toronto International Conference*, IEEE (2009), 100-105.

4. Arif, A. S. and Stuerzlinger, W., User adaptation to a faulty unistroke-based text entry technique by switching to an alternative gesture set. In *Proceedings of Graphics Interface 2014 – GI 2014*, Toronto, Canada: Canadian Information Processing Society (2014), 183-192.

5. Bailey, R. N., *Human performance engineering: Designing high quality, professional user interfaces for computer products, applications, and systems*, 3rd ed. Prentice Hall, 1996.

6. Balakrishnan, V. and Yeow, H. P., Hand-size variations effect on mobile phone texting satisfaction, *Ubiquitous Computing and Communication Journal*, *2*(5), 2007, 1-8.

7. Blickenstorfer, C. H., Graffiti: Wow!, in *Pen Computing Magazine* 1995, 30-33.

8. Brewster, S., Chohan, F., and Brown, L., Tactile feedback for mobile interactions. In *Proceedings of the SIGCHI conference on Human factors in computing systems – CHI '07*, New York: ACM (2007), 159-162.

9. Brown, M. A. and MacKenzie, I. S., Evaluating video game controller usability as related to user hand size. In *Proceedings of the International Conference on Multimedia and Human Computer Interaction – MHCI 2013*, Ottawa, Canada: International ASET Inc. (2013), 114.1-114.9.

10. Burkard, R. E. and OfferMann, J., Entwurf von schreibmaschinentastaturen mittels quadratischer zuordnungsprobleme, *Mathematical Methods of Operations Research*, *21*(4), 1977, B121-B132.

11. Casey, B., *OTT messaging traffic will be twice volume of P2P SMS traffic this year*, Informa Telecoms & Media, 2013. http://www.informa.com/Media-centre/Press-releases--news/Latest-News/OTT-messaging-traffic-will-be-twice-volume-of-P2P-SMS-traffic-this-year/.

12. Castellucci, S. J. and MacKenzie, I. S., Graffiti vs. Unistrokes: An empirical comparison. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems – CHI 2008*, New York: ACM (2008), 305-308.

13. Castellucci, S. J. and MacKenzie, I. S., UniGest: Text entry using three degrees of motion. In *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems – CHI 2008*, New York: ACM (2008), 3549-3554.

14. Castellucci, S. J. and MacKenzie, I. S., TnToolkit: A design and analysis tool for ambiguous, QWERTY, and on-screen keypads. In *Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems – EICS 2009*, New York: ACM (2009), 55-60.

15. Castellucci, S. J. and MacKenzie, I. S., Gathering text entry metrics on Android devices. In *Proceedings of the International Conference on Multimedia and Human Computer Interaction – MHCI 2013*, Ottawa, Canada: International ASET Inc. (2013), 119.1-119.8.

16. Castellucci, S. J. and MacKenzie, I. S., Gestural text entry using Huffman codes. In *Proceedings of the International Conference on Multimedia and Human Computer Interaction – MHCI 2013*, Ottawa, Canada: International ASET Inc. (2013), 120.1-120.7.

17. Castellucci, S. J. and MacKenzie, I. S., Touch me, hear me, feel me: Feedback preferences for mobile text entry. In *Proceedings of the International Conference on Multimedia and Human Computer Interaction – MHCI 2013*, Ottawa, Canada: International ASET Inc. (2013), 123.1-123.6.

18. Chau, D. H., Wobbrock, J. O., Myers, B. A., and Rothrock, B., Integrating isometric joysticks into mobile phones for text entry. In *Extended Abstracts of the ACM Conference on Human Factors in Computer Systems – CHI 2006*, New York: ACM (2006), 640-645.

19. Chen, T. and Kan, M.-Y., Creating a live, public short message service corpus: the NUS SMS corpus, *Language Resources and Evaluation*, *47*(2), 2013, 299-335.

20. Clarkson, E., Clawson, J., Lyons, K., and Starner, T., An empirical study of typing rates on mini-QWERTY keyboards. In *CHI '05 extended abstracts on Human factors in computing systems*, New York: ACM (2005), 1288-1291.

21. Cuaresma, J. and MacKenzie, I. S., A study of variations of Qwerty soft keyboards for mobile phones. In *Proceedings of the International Conference on Multimedia and Human-Computer Interaction - MHCI 2013*, Ottawa, Canada: International ASET Inc. (2013), 126.1-126.8.

22. Dix, A., Finlay, J., Abowd, G., and Beale, R., *Human-Computer Interaction*, 2nd ed. Prentice Hall, 1998.

23. Dunlop, M. D. and Taylor, F., Tactile feedback for predictive text entry. In *Proceedings of the 27th international conference on Human factors in computing systems – CHI '09*, New York: ACM (2009), 2257-2260.

24. Dybdal, M. L., San Agustin, J., and Hansen, J. P., Gaze input for mobile devices by dwell and gestures. In *Proceedings of the Symposium on Eye Tracking Research and Applications – ETRA '12*, New York: ACM (2012), 225-228.

25. Facebook, *Company Info*. (Accessed on May 20, 2014.) http://newsroom.fb.com/company-info/.

26. Felzer, T. and Nordmann, R., Alternative text entry using different input methods. In *Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility – ASSETS 2006*, New York: ACM (2006), 10-17.

27. Fetto, J., *American spend 58 minutes a day on their smartphones*, Experian, Marketing Forward, May 28, 2013. http://www.experian.com/blogs/marketing-forward/2013/05/28/americans-spend-58-minutes-a-day-on-their-smartphones/.

28. Fitton, D., MacKenzie, I. S., Read, J. C., and Horton, M., Exploring tilt-based text input for mobile devices with teenagers. In *Proceedings of the 27th International British Computer Society Human-Computer Interaction Conference – HCI 2013*, London: British Computer Society (2013), 1-6.

29. Fitts, P. M., The information capacity of the human motor system in controlling the amplitude of movement, *Journal of Experimental Psychology*, *47*(6), 1954, 381-391.

30. Fuccella, V., Rosa, M. D., and Costagliola, G., Novice and expert performance of KeyScretch: A gesture-based text entry method for touch-screens, *IEEE Transactions on Human-Machine Systems*, (99), 2014, 1-13.

31. Goldberg, D., Unistrokes for computerized interpretation of handwritimg, in *http://www.freepatentsonline.com/5596656.pdf*, U. S. Patent, Ed. USA: Xerox Corporation, Stamford, Conn., 1997.

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

33. Gupta, R. and Sornlertlamvanich, V., Text entry in South and Southeast Asian languages, in *Text entry systems: Mobility, accessibility, universality*, (I. S. MacKenzie and K. Tanaka-Ishii, Eds.). Morgan Kaufmann, 2007, 227-249.

34. Henze, N., Rukzio, E., and Boll, S., Observational and experimental investigation of typing behaviour using virtual keyboards on mobile devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems – CHI 2012*, New York: ACM (2012), 2659-2668.

35. Hoenig, J. M. and Heisey, D. M., The abuse of power: The pervasive fallacy of power calculations for data analysis, *American Statistical Association*, *55*(1), 2001, 19-24.

36. Hoggan, E., Brewster, S., and Johnston, J., Investigating the effectiveness of tactile feedback for mobile touchscreens. In *Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems – CHI '08*, New York: ACM (2008), 1573-1582.

37. Hoggan, E., Crossan, A., Brewster, S., and Kaaresoja, T., Audio or tactile feedback: Which modality when? In *Proceedings of the 27th international conference on Human factors in computing systems – CHI '09*, New York: ACM (2009), 2253-2256.

38. Hopkins, D., The design and implementation of pie menus, *Dr. Dobb's Journal*1991.

39. Huffman, D. A., A method for the construction of minimum-redundancy codes, *Proceedings of the IRE*, *40*(9), 1952, 1098-1101.

40. Hughes, D., Warren, J., and Buyukkokten, O., Empirical bi-action tables: A tool for the evaluation and optimization of text-input stystems. Application I: Stylus keyboards, *Human-Computer Interaction*, *17*(2/3), 2002, 271-309.

41. Isokoski, P., Martin, B., Gandouly, P., and Stephanov, T., Motor efficiency of text entry in a combination of a soft keyboard and unistrokes. In *Proceedings of the*

6th Nordic Conference on Human-Computer Interaction – NordiCHI 2010, New York: ACM (2010), 683-686.

42. Isokoski, P. and Raisamo, R., Device independent text input: A rationale and an example. In *Proceedings of the Working Conference on Advanced Visual Interfaces – AVI 2000*, New York: ACM (2000), 76-83.

43. Isokoski, P. and Raisamo, R., Quikwriting as a multi-device text entry method. In *Proceedings of the 3rd Nordic Conference on Human-Computer Interaction – NordiCHI 2004*, New York: ACM (2004), 105-108.

44. Jain, S. and Pathmanathan, G., Importance of anthopometry for designing user-friendly devices: Mobile phones, *J. Ergonomics*, *2*(4), 2012.

45. Katre, D., One-handed thumb use on smart phones by semi-literate and illiterate users in India, in *Human Work Interaction Design: Usability in Social, Cultural and Organizational Contexts*, (D. Katre, R. Orngreen, P. Yammiyavar, and T. Clemmensen, Eds.). Springer, 2010, 189-208.

46. Költringer, T. and Grechenig, T., Comparing the immediate usability of Graffiti 2 and virtual keyboard. In *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems – CHI 2004*, New York: ACM (2004), 1175-1178.

47. Költringer, T., Isokoski, P., and Grechenig, T., TwoStick: Writing with a game controller. In *Proceedings of Graphics Interface 2007 – GI 2007*, New York: ACM (2007), 103-110.

48. Koopmans, T. C. and Beckmann, M., Assignment problems and the location of economic activities, *Econometrica*, *25*(1), 1957, 53-76.

49. Koskinen, E., Kaaresoja, T., and Laitinen, P., Feel-good touch: Finding the most pleasant tactile feedback for a mobile touch screen button. In *Proceedings of the 10th international conference on Multimodal interfaces – ICMI '08*, New York: ACM (2008), 297-304.

50. Kristensson, P.-O. and Zhai, S., SHARK^2: A large vocabulary shorthand writing system for pen-based computers. In *Proceedings of the 17th annual ACM Symposium on User Interface Software and Technology – UIST 2004*, New York: ACM (2004), 43-52.

51. Kristensson, P. O., *Discrete and continuous shape writing for text entry and control*. Unpublished doctoral dissertation, Linköping University, Linköping, Sweden, 2007.

52. Kristensson, P. O. and Vertanen, K., Performance comparisons of phrase sets and presentation styles for text entry evaluations. In *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*, New York: ACM (2012), 29-32.

53. Layton, L., Elementary students learn keyboard typing ahead of new Common Core tests, in *The Washington Post*, October 13, 2013. http://www.washingtonpost.com/local/education/elementary-students-learn-keyboard-typing-ahead-of-new-common-core-tests/2013/10/13/d329ba66-3289-11e3-9c68-1cf643210300_story.html.

54. Lenth, R. V., *Two sample-size practices that I don't recommend*. (Accessed on October 31, 2014.) http://homepage.stat.uiowa.edu/~rlenth/Power/2badHabits.pdf.

55. Lenth, R. V., Some practical guidelines for effective sample size determination, *The American Statistician*, *55*(3), 2001, 187-193.

56. Levine, M. and Ensom, M. H. H., Post hoc power analysis: An idea whose time has passed?, *Pharmacotherapy*, *21*(4), 2001, 405-409.

57. Liu, Y., *Chinese Text Entry with Mobile Devices*. Unpublished University of Tampere, Tampere, Finland, 2010.

58. Llamas, R., Reith, R., and Shirer, M., *Android and iOS continue to dominate the worldwide smartphone market with Android shipments just shy of 800 million in 2013*, IDC, Feb. 12, 2014. http://www.idc.com/getdoc.jsp?containerId=prUS24676414.

59. Lyons, K. and Clawson, J., Analytical evaluation of the impact of phrase set on text entry rates. In *CHI 2012 Workshop on Designing and Evaluating Text Entry Methods*, New York: ACM (2012).

60. MacKenzie, I. S., KSPC (keystrokes per character) as a characteristic of text entry techniques. In *Proceedings of the 4th International Symposium on Mobile Human-Computer Interaction – Mobile HCI 2002*, Heidelberg, Germany: Springer-Verlag (2002), 195-210.

61. MacKenzie, I. S., *Human-computer interaction: An empirical research perspective*. Morgan Kaufmann, 2013.

62. MacKenzie, I. S. and Castellucci, S. J., Reducing visual demand for gestural text input on touchscreen devices. In *Extended Abstracts of the ACM Conference on*

Human Factors in Computing Systems – CHI 2012, New York: ACM (2012), 2585-2590.

63. MacKenzie, I. S. and Castellucci, S. J., Eye on the message: Reducing attention demand for touch-based text entry, *International Journal of Virtual Worlds and Human-Computer Interaction*, *1*(1), 2013, 1-9.

64. MacKenzie, I. S. and Soukoreff, R. W., Text entry for mobile computing: Models and methods, theory and practice, *Human-Computer Interaction*, *17*(2&3), 2002, 147-198.

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

66. MacKenzie, I. S., Soukoreff, R. W., and Helga, J., 1 thumb, 4 buttons, 20 words per minute: Design and evaluation of H4-Writer. In *Proceedings of the ACM Symposium on User Interface Software and Technology – UIST 2011*, New York: ACM (2011), 471-480.

67. MacKenzie, I. S. and Tanaka-Ishii, K., *Text entry systems: Mobility, accessibility, universality*. Morgan Kaufmann, 2007.

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

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

70. Maihafer, G. C., Llewellyn, M. A., Jr., W. J. P., Scott, K. L., Marino, D. M., and Bond, R. M., A comparison of the figure-of-eight method and water volumetry in measurement of hand and wrist size, *J. of Hand Therapy*, *16*(4), 2003, 305-310.

71. Malik, S. and Laszlo, J., Visual touchpad: A two-handed gestural input device. In *Proceedings of the 6th International Conference on Multimodal Interfaces – ICMI 2004*, New York: ACM (2004), 289-296.

72. Mankoff, J. and Abowd, G. D., Cirrin: A word-level unistroke keyboard for pen input. In *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology – UIST 1998*, New York: ACM (1998), 213-214.

73. Martin, B., VirHKey: A VIRtual Hyperbolic KEYboard with gesture interaction and visual feedback for mobile devices. In *Proceedings of the 7th International Conference on Human Computer Interaction with Mobile Devices & Services – MobileHCI 2005*, New York: ACM (2005), 99-106.

74. Martin, B. and Isokoski, P., EdgeWrite with integrated corner sequence help. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems – CHI 2008*, New York: ACM (2008), 583-592.

75. Martin, D. W., *Doing psychology experiments*, 6th ed. Wadsworth, 2004.

76. Mathews, L., *Future Surface tablets may have touch-sensitive bezels and rear casing*. (Accessed on November 18, 2013.) http://www.geek.com/tablets/future-surface-tablets-may-have-touch-sensitive-bezels-and-rear-casing-1577480/.

77. Mathews, L., *Samsung patent dynamically adjusts a smartphone UI to fit your thumb's reach*. (Accessed on December 4, 2013.) http://www.geek.com/android/samsung-patent-dynamically-adjusts-a-smartphone-ui-to-fit-your-thumbs-reach-1578696/.

78. McAdam, C. and Brewster, S., Distal tactile feedback for text entry on tabletop computers. In *Proceedings of the 23rd British HCI Group Annual Conference on People and Computers – BCS-HCI '09*, London: British Computer Society Swinton (2009), 504-511.

79. Meyer, A., Pen computing: A technology overview and a vision, *ACM SIGCHI Bulletin*, *27*(3), 1995, 46-90.

80. Meyns, P., Bruijn, S. M., and Duysens, J., The how and why of arm swing during human walking, *Gait & Posture*, *38*(4), 2013, 555-562.

81. Morimoto, C. H. and Amir, A., Context switching for fast key selection in text entry applications. In *Proceedings of the 2010 Symposium on Eye-Tracking Research and Applications – ETRA 2010*, New York: ACM (2010), 271-274.

82. Ni, T., Bowman, D., and North, C., AirStroke: Bringing unistroke text entry to freehand gesture interfaces. In *Proceedings of the 2011 annual conference on Human factors in computing systems – CHI 2011*, New York: ACM (2011), 2473-2476.

83. Oulasvirta, A., Reichel, A., Li, W., Zhang, Y., Bachynskyi, M., Vertanen, K., and Kristensson, P. O., Improving two-thumb text entry on touchscreen devices. In

Proceedings of the SIGCHI Conference on Human Factors in Computing Systems – CHI 2013, New York: ACM (2013), 2765-2774.

84. Padilla, R., *iOS 8 to include support for system wide third-party keyboards*. (Accessed on June 3, 2014.) http://www.macrumors.com/2014/06/02/ios-8-third-party-keyboards/.

85. Paek, T. and Hsu, B.-J. P., Sampling representative phrase sets for text entry experiments: A procedure and public resource. In *Proceedings of the SIGCHI conference on Human factors in computing systems – CHI '11*, New York: ACM (2011), 2477-2480.

86. Pagano, R. R., *Understanding statistics in the behavioral sciences*, 8th ed. Thomson-Wadsworth, 2007.

87. Partridge, K., Chatterjee, S., Sazawal, V., Borriello, G., and Want, R., TiltType: Accelerometer-supported text entry for very small devices. In *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology – UIST 2002*, New York: ACM (2002), 201-204.

88. Pellecchia, G. L., Figure-of-eight method of measuring hand size: Reliability and concurrent validity, *J. of Hand Therapy*, *16*(4), 2003, 300-304.

89. Perlin, K., Quikwriting: Continuous stylus-based text entry. In *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology – UIST 1998*, New York: ACM (1998), 215-216.

90. Perry, K. B. and Hourcade, J. P., Evaluating one handed thumb tapping on mobile touchscreen devices. In *Proceedings of Graphics Interface 2008 – GI 2008,* Toronto: Canadian Information Processing Society (2008), 57-64.

91. Petrović, S., Osborne, M., and Lavrenko, V., The Edinburgh Twitter corpus. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media*, Stroudsburg, PA, USA: Association for Computational Linguistics (2010), 25-26.

92. Pichai, S., *#AndroidKitKat*. (Accessed on December 1, 2013.) http://plus.google.com/+SundarPichai/posts/NeBW7AjT1QM.

93. Puget, J. F., *No, the TSP isn't NP complete*. (Accessed on March 7, 2014.) https://www.ibm.com/developerworks/community/blogs/jfp/entry/no_the_tsp_isn_t_np_complete?lang=en.

94. Raynal, M. and Vigouroux, N., Genetic algorithm to generate optimized soft keyboard. In *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems – CHI 2005*, New York: ACM (2005), 1729-1732.

95. Razali, N. M. and Wah, Y. B., Power comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors and Anderson-Darling tests, *Journal of Statistical Modeling and Analytics*, *2*(1), 2011, 21-33.

96. Rinott, M., SonicTexting. In *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems – CHI 2005*, New York: ACM (2005), 1144-1145.

97. Rivera, J. and van der Meulen, R., *Gartner says worldwide tablet sales grew 68 percent in 2013, with Android capturing 62 percent of the market*, Gartner, March 3, 2014. http://www.gartner.com/newsroom/id/2674215.

98. Sahni, S. and Gonzalez, T., P-complete approximation problems, *Journal of the ACM*, *23*(3), 1976, 555-565.

99. Sale, S., *OTT messaging volumes will nearly double in 2014*, Analysys Mason, January 28, 2014. http://www.analysysmason.com/About-Us/News/Insight/OTT-messaging-volumes-Jan2014-RDMV0/.

100. Sasaki, T. and Tanaka-Ishii, K., Text entry in Hebrew and Arabic scripts, in *Text entry systems: Mobility, accessibility, universality*, (I. S. MacKenzie and K. Tanaka-Ishii, Eds.). Morgan Kaufmann, 2007, 251-268.

101. Sazawal, V., Want, R., and Borriello, G., The Unigesture approach: One-handed text entry for small devices, in *Human Computer Interaction with Mobile Devices*, vol. 2411/2002, *Lecture Notes in Computer Science*, (F. Paternò, Ed.). Springer Berlin / Heidelberg, 2002, 485-496.

102. Shoemaker, G., Findlater, L., Dawson, J. Q., and Booth, K. S., Mid-air text input techniques for very large wall displays. In *Proceedings of Graphics Interface 2009 – GI 2009*, Toronto: Canadian Information Processing Society (2009), 231-238.

103. Silfverberg, M., MacKenzie, I. S., and Korhonen, P., Predicting text entry speed on mobile phones. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems – CHI 2000*, New York: ACM (2000), 9-16.

104. Smith, A. L., *Smartphone input method performance, satisfaction, workload, and preference with younger and older novice adults*. Unpublished doctoral dissertation, Wichita State University, Wichita, Kansas, USA, 2013.

105. Soukoreff, R. W. and MacKenzie, I. S., Measuring errors in text entry tasks: An application of the Levenshtein string distance statistic. In *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems – CHI 2001*, New York: ACM (2001), 319-320.

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

107. Soukoreff, R. W. and MacKenzie, I. S., Recent developments in text-entry error rate measurement. In *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems – CHI 2004*, New York: ACM (2004), 1425-1428.

108. Soukoreff, R. W. and MacKenzie, I. S., An informatic rationale for the speed-accuracy tradeoff. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics – SMC 2009*, New York: IEEE (2009), 2890-2896.

109. Srihari, S. N., Sung-Hyuk, C., Arora, H., and Sangjik, L., Individuality of handwriting: A validation study. In *Proceedings of the 6th International Conference on Document Analysis and Recognition – ICDAR 2001*, New York: IEEE (2001), 106-109.

110. Statistical_Consulting_Group-UCLA, *Statistical computing seminars: Introduction to power analysis*. (Accessed on October 31, 2014.) http://www.ats.ucla.edu/stat/seminars/Intro_power/default.htm.

111. Styler, W., *The EnronSent Corpus*, University of Colorado at Boulder Institute of Cognitive Science, Boulder, CO, Technical Report 01-2011, 2011. https://verbs.colorado.edu/enronsent/.

112. Tactus_Technology, *"Appearing & disappearing" haptic user interface has first public demonstration*. (Accessed on June 20, 2013.) http://www.tactustechnology.com/release_120605.html.

113. Tanaka-Ishii, K., Zhou, M., and Kim, J.-D., Text entry in East Asian languages, in *Text entry systems: Mobility, accessibility, universality*, (I. S. MacKenzie and K. Tanaka-Ishii, Eds.). Morgan Kaufmann, 2007, 203-225.

114. Tappert, C. C. and Cha, S.-H., English language handwriting recognition interfaces, in *Text entry systems: Mobility, accessibility, universality*, (I. S. MacKenzie and K. Tanaka-Ishii, Eds.). Morgan Kaufmann, 2007, 123-137.

115. Tappert, C. C., Suen, C. Y., and Wakahara, T., Online handwriting recognition—a survey. In *Proceedings of the Ninth International Conference on Pattern Recognition – ICPR 1988*, New York: IEEE (1988), 1123-1132.

116. Thapa, A., *Beginners performance with MessagEase and QWERTY*. Unpublished Masters Thesis, University of Tampere, TamPub Institutional repository, 2013.

117. Thomas, L., Retrospective power analysis, *Conservation Biology*, *11*(1), 1997, 279-280.

118. Twitter, *About Twitter, Inc.* (Accessed on May 20, 2014.) https://about.twitter.com/company.

119. Venolia, D. and Neiberg, F., T-Cube: A fast, self-disclosing pen-based alphabet. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems – CHI 1994*, New York: ACM (1994), 265-270.

120. Vertanen, K. and Kristensson, P. O., A versatile dataset for text entry evaluations based on genuine mobile emails. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services – MobileHCI 2011*, New York: ACM (2011), 295-298.

121. Vertanen, K. and MacKay, D. J. C., Speech Dasher: Fast writing using speech and gaze. In *Proceedings of the 28th international conference on Human factors in computing systems – CHI 2010*, New York: ACM (2010), 595-598.

122. Wallen, J., *Android smartphones: Oversizing trend disregards usability*. (Accessed on May 1, 2014.) http://www.techrepublic.com/article/android-smartphones-oversizing-trend-disregards-usability/.

123. Ward, D. J., Blackwell, A. F., and MacKay, D. J. C., Dasher – A data entry interface using continuous gestures and language models. In *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology – UIST 2000*, New York: ACM (2000), 129-137.

124. Weir, D., Pohl, H., Rogers, S., Vertanen, K., and Kristensson, P. O., Uncertain text entry on mobile devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems – CHI 2014*, New York: ACM (2014), 2307-2316

125. White, J., *Happy birthday SMS!* (Accessed on December 8, 2012.) http://www.portioresearch.com/blog/2012/12/happy-birthday-sms!.aspx.

126. Wigdor, D. and Balakrishnan, R., *TiltText*: Using tilt for text input to mobile phones. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology – UIST 2003*, New York: ACM (2003), 81-90.

127. Williamson, J. and Murray-Smith, R., Hex: Dynamics and probabilistic text entry, in *Switching and Learning in Feedback Systems*, vol. 3355, (R. Murray-Smith and R. Shorten, Eds.). Springer-Verlag, 2005, 333-342.

128. Witt, H. and Janssen, T., Comparing two methods for gesture based short text input using chording. In *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems – CHI 2007*, New York: ACM (2007), 2759-2764.

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

130. Wobbrock, J., Measures of text entry performance, in *Text entry systems: Mobility, accessibility, universality*, (I. S. MacKenzie and K. Tanaka-Ishii, Eds.). Morgan Kaufmann, 2007, 47-74.

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

132. Wobbrock, J. and Myers, B., Trackball text entry for people with motor impairments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems – CHI 2006*, New York: ACM (2006), 479-488.

133. Wobbrock, J., Myers, B., and Rothrock, B., Few-key text entry revisited: Mnemonic gestures on four keys. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems – CHI 2006*, New York: ACM (2006), 489-492.

134. Wobbrock, J. O., Chau, D. H., and Myers, B. A., An alternative to push, press, and tap-tap-tap: Gesturing on an isometric joystick for mobile phone text entry. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems – CHI 2007*, New York: ACM (2007), 667-676.

135. Wobbrock, J. O., Findlater, L., Gergle, D., and Higgins, J. J., The Aligned Rank Transform for nonparametric factorial analyses using only ANOVA procedures. In *Proceedings of the SIGCHI conference on Human factors in computing systems – CHI '11*, New York: ACM (2011), 143-146.

136. Wobbrock, J. O. and Myers, B. A., Gestural text entry on multiple devices. In *Proceedings of the 7th International ACM SIGACCESS Conference on Computers and Accessibility – ASSETS 2005*, New York: ACM (2005), 184-185.

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

138. Wobbrock, J. O., Myers, B. A., Aung, H. H., and LoPresti, E. F., Text entry from power wheelchairs: EdgeWrite for joysticks and touchpads. In *Proceedings of the 6th International ACM SIGACCESS Conference on Computers and Accessibility – ASSETS 2004*, New York: ACM (2004), 110-117.

139. Wobbrock, J. O., Myers, B. A., and Chau, D. H., In-stroke word completion. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology – UIST 2006*, New York: ACM (2006), 333-336.

140. Wobbrock, J. O., Myers, B. A., and Kembel, J. A., EdgeWrite: A stylus-based text entry method designed for high accuracy and stability of motion. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology – UIST 2003*, New York: ACM (2003), 61-70.

141. Wobbrock, J. O., Rubinstein, J., Sawyer, M. W., and Duchowski, A. T., Longitudinal evaluation of discrete consecutive gaze gestures for text entry. In *Proceedings of the 2008 symposium on Eye tracking research and applications – ETRA 2008*, New York: ACM (2008), 11-18.

142. Woodworth, R. S., The accuracy of voluntary movement, *The Psychological Review*, *3*(3), 1899, 1-114.

143. Yamada, H., A historical study of typewriters and typing methods: From the position of planning Japanese parallels, *Journal of Information Processing*, *2*(4), 1980, 175-202.

144. Zhai, S., Hunter, M., and Smith, B. A., The metropolis keyboard – An exploration of quantitative techniques for virtual keyboard design. In *Proceedings of the 13th*

Annual ACM Symposium on User Interface Software and Technology – UIST 2000, New York: ACM (2000), 119-128.

145. Zhai, S. and Kristensson, P.-O., Shorthand writing on stylus keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems – CHI 2003*, New York: ACM (2003), 97-104.

146. Zhai, S., Kristensson, P. O., Gong, P., Greiner, M., Peng, S. A., Liu, L. M., and Dunnigan, A., ShapeWriter on the iPhone: From the laboratory to the real world. In *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems – CHI 2009*, New York: ACM (2009), 2667-2670.

147. Zhang, D., Chou, H.-M., and Zhou, L., Hype or ready for prime time?: Speech recognition on mobile handheld devices (MASR), *International Journal of Handheld Computing Research*, *3*(4), 2012, 40-55.