Towards Keyboard Independent Touch Typing in VR

Falko Kuester, Michelle Chen, Mark E. Phair and Carsten Mehring
Calit2 Center of GRAVITY
University of California Irvine
Irvine, CA 92697
{fkuester,mwchen,mphair,cmehring}@uci.edu

ABSTRACT

A new hand- or finger-mounted data input device, called KITTY, is presented. KITTY is designed for keyboard independent touch typing and supports traditional touch-typing skills as a method for alphanumeric data input. This glove-type device provides an ultra-portable solution for "quiet" data input into portable computer systems and full freedom of movement in mobile VR and AR environments. The KITTY design follows the concept of the column and row layout found on traditional keyboards, allowing users to draw from existing touch typing skills, easing the required training time.

Categories and Subject Descriptors

H.5.1 [**Multimedia Information Systems**]: Artificial, augmented, and virtual realities; H.5.2 [**User Interfaces**]: Input devices and strategies, Prototyping—*Data gloves*

General Terms

Human Factors, Design, Experimentation

Keywords

Wearable devices for augmented reality, keyboard independent touch-typing, glove input, interaction techniques

1. INTRODUCTION

With recent advances in portable computing and in particular the design of pocket PCs [18] and eye-glass displays [19], the development of new augmented reality (AR) environments has been made possible. However, one of the current shortcomings in these AR systems is the lack of intuitive input devices that provide users with complete control over their workspace. In particular, in environments where voice input is undesirable or infeasible, touch-typing capabilities have to be available to allow for intuitive access to possibly complex data.

Figure 1: Prototype glove system illustrating initial contact placement.

The size, layout and weight of traditional keyboards make them poor candidates for use in mobile AR applications. In particular, if integration with mobile computing devices, such as personal digital assistants (PDAs) or pocket personal computers (PocketPCs) is desired, a new generation of natural handheld input devices is required.

Today's mobile computing devices frequently include a physical or digital miniature keyboard, that may be integrated with the device, worn on the user's wrist or simply be part of the display area. Unfortunately, these keyboards generally require the use of both hands, one holding the device and the other for finger-based or stylus-based data input. The selection of individual keys on a hand held display-based keyboard, for example, is a rather inefficient process and requires the user to look at the keyboard device. Handwriting recognition can increase input rates after extensive training but is still very limiting. This performance gap is problematic if the portable computing devices are to be operated at the level suitable for today's computational platforms.

Other devices for data input into portable computing units, include chording devices, such as a hand-mounted keypads that require the user to press different key combinations to generate the characters found on a standard keyboard. Further alternatives include microphones in combination with voice recognition software. The former input devices typically do not meet the data input rates desired for full-scale interaction and typically require the user to learn a chording/coding language particular to the specific device. On the other hand, privacy concerns or ambient noise frequently limit the use of voice recognition devices. Possible scenarios include input of confidential or sensitive data via voice recognition in conference settings, meetings, or crowded public places, such as airports.

2. RELATED WORK

With a demand for decreased keyboard size and increased portability, while maintaining the traditional keyboard layout, new device concepts are required. A first attempt was the development of foldable keyboards that allowed users to carry familiar input devices with reduced physical footprint. While these devices are certainly more compact than traditional full-size keyboards during transport, a folding keyboard is still relatively large in the unfolded state in which its surface requirements match that of a traditional full-size device.

Other devices, such as the "Wrist-PC" keyboard [8], have tried to reduce the number of keys rather than reducing the overall size of the keys in order to achieve size reductions. For example, one-handed keyboards use only half of a traditional keyboard by mapping two characters on each key. This requires the addition of a switch (keyboard identification) key, enabling the user to toggle between two sets of characters, each representing one-side of a full-size keyboard. Even though "half-keyboards" may have a very small footprint, they commonly still require a support surface.

The "body-wearable keyboard" and similarly the so-called "Gesture-Pad", which is under development by Sony Inc., place "keyboard-keys" on the users clothing or a skeletal structure worn on the users body. Even though this increases portability, it greatly restricts the users arm-movement and flexibility for data input.

Many of the existing keyboard- or keypad-less data input devices for portable computing applications, use finger or hand mounted sensors in order to generate the signals that would normally be generated by pressing a keyboard key. The "finger-mounted computer input device" [16], the "fingerless glove for interacting with a data processing system" [20] and the "Virtual Keyboard" by Senseboard Technologies [17] use pressure sensors on the users fingers or sensors (mounted on gloves) measuring flexure of the fingers in order to determine data input via finger positions with respect to a physical or virtual reference surface. Generally, these devices assume that the reference surface functions as virtual touch-typing environment. As for folding keyboards, the need of a physical support surface greatly limits the mobile computing capabilities of these devices.

In response to this limitation, hand- or finger-mounted input devices have been proposed, such as the "FingeRing" [3], "Chording Glove" [22], "PinchGlove" [2] and "Twiddler" [5]. These three input devices use a combination of finger presses (chord), such that several "keys" are pressed simultaneously to generate a specific keyboard event. Although chording devices remove the need for a support surface, their use requires substantial training as the user has to learn a new coding language. The "FingeRing" is a continuous signal device that will limit freedom of movement by locking the fingers in the neutral position to prevent accidental keystrokes.

Pratt [15] describes a device-independent digital sign language ("Thumbcode"), using keys on the users fingers that are activated by the thumbs of the same hand. In order to generate different input signals with the same key, the user has to pair neighboring fingers according to a prescribed coding language. As with chording, thumbcoding requires a user to essentially learn a new coding language in order to enter data into the computing device.

Figure 2: KITTY (left) and Twiddler (right) in comparison.

The functionality of the so-called "Lightglove" by Howard [9, 7], is based on the generation of a "light-matrix" below the users hands. Location and penetration of this matrix by the users fingertips is detected and electronically mapped to an overlay assigning each position a character according to the layout of a traditional keyboard. This operation of the device involves simultaneous scanning of the matrix with various light emitting diodes or lasers, evaluation of the scattered electromagnetic waveforms by various light sensitive detectors and signal filtering in a bandpass filter in order to reject non-correlated ambient signals. Generation of the optical matrix and detection of finger position in this matrix is fairly complicated. Furthermore, the system has to be calibrated (possibly in real-time) for specific ambient conditions, such as strong ambient illumination or electromagnetic interference. The lack of tactile feedback (providing confirmation of data input as is the case for touch-typing on a keyboard) may also result in lower data input rates. The use of an optical reflection matrix to detect signal input by predicting finger position with respect to the optical matrix below the user's hands does not allow for free finger motion without the risk of data input. Instead, the user has to hold the hands and fingers in a more-or-less stretched out position in order to omit interference with the optical matrix. The involuntary generation of signals is a common problem with these devices and has to be addressed.

A highly portable but still limiting device is the VKB [21] laser projection keyboard. The electronics are small enough to fit into a pen but can project a full-size keyboard onto a flat surface. This device does provide fast and intuitive input, however requiring a flat surface confines the user to a desktop area which often is not available for VR/AR environments or laptop situations.

In general, many of the devices described in this section are large, counter-intuitive, or too cumbersome for fast data input. Most users when confronted with having to learn an entirely new coding language in order to enter information will likely decide against using such a device. This is particularly true for good VR or AR applications that should require only very limited user training. In addition, for VR environments implementing head-mounted displays, the user's real environment becomes inaccessible, and devices that require visual reference will be rendered ineffectual. Similarly, handheld devices are not easily put down, reacquired, and repositioned, and continuously holding one in the hand may interfere with hand-tracking.

For VR/AR and mobile computing applications, intuitive, flexible, and efficient, keyboardless input devices are urgently needed. KITTY, a device prototype that addresses these objectives is presented in this paper (Figure 2).

3. KITTY INTERFACE

The presented KITTY interface overcomes the limitations described in the previous sections by providing intuitive access to common touch-typing skills. To accomplish this, the device provides multiple contacts on each finger that are only activated when the appropriate finger and thumb contacts are combined. The traditional QWERTY keyboard layout was chosen to allow anyone with previous touch-typing experience to make a smoother transition to a foreign input device. While mentally visualizing a QWERTY keyboard is difficult for most, it must be remembered that typing speed is not acquired from visualization but through muscle memory of the layout. Relying on this sensory-motor skill will allow the user to "feel" how their fingers should be placed for each key. Another significant advantage of this device is that arbitrary finger motion is allowed without data input, as long as finger contacts and thumb contacts on a single hand do not meet. In other words, KITTY allows the user complete freedom for hand movements while only generating input when finger- and thumb-contacts are combined.

3.1 Finger-Thumb Touch-Typing

Two different key-maps were proposed for the finger mounted KITTY interface. Section 3.2 discusses the alphabetic key-mapping as employed in the wired prototype shown in Figure 2. The mapping of alphabetic keys is described in the framework of a comprehensive mapping procedure for the entire set of keys/commands found on a standard keyboard. Note that the wired prototype shown in the photograph employs a commercially available keyboard by GrandTec and provides alphabetic data-input functionality as well as punctuation marks and a limited number of essential commands such as "Delete", "Space", "Return", and "Shift". Section 3.3 describes a slightly modified key-map procedure for alphanumeric character input. This mapping is used for the recently completed wireless KITTY interface.

Figure 3 illustrates a standard QWERTY keyboard, which was used as the reference for the design of the KITTY keyboard mapping. While the QWERTY layout was chosen for the initial implementation other key mappings such as Dvorak or the half-keyboard can be easily supported. Using the standard QWERTY keyboard (Figure 3) via touch-typing, a user's fingers are normally in a home position that includes placing the left pinky on the A-key, the ring finger on the S-key, the middle finger on the D-key and the left index finger on the F-key. The home position for the right hand includes placement of the right index finger on the J-key, the right middle finger on the K-key, the index finger on the L-key and the right pinky on the ;-key. This row (ASDF ...), which is annotated in Figure 3 with a solid line, may be referred to as the base row. The row above the base row (QWERTY ...) is annotated with a long dashed line and is referred to as the top row and the row below the base row (ZXCV ...) is annotated with a short dash line and referred to as the bottom row. Figure 3 also illustrates the enumeration of left-hand fingers (1-4), right-hand fingers (5-8) and thumbs with thumb contacts (A,A0), (B,B0), (C,C0), (D,D0), (E,E0), (F,F0) employed for the key-map of the wired prototype as discussed in Section 3.2.

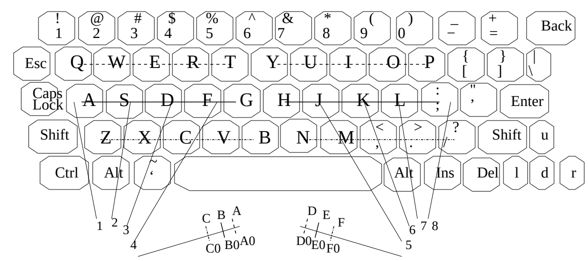


Figure 3: Standard US-keyboard (HP Omnibook 500 C) with division of alphabetical characters into 'base' row (solid), top row (dashed) and bottom row (dash-dot) characters.

3.2 Key-Map Wired Prototype

Figure 4 illustrates the location of electrical contacts on the user's hands as employed within the present wired prototype and following the key layout shown in Figure 3. Both finger- and thumb-contacts are employed. An electric circuit is closed and a signal is generated upon closing of one finger-contact with one thumb-contact as discussed below in detail.

As shown in Figure 4 by dashed lines, finger contacts 1-8 are located on the palm-side of the user's hand near the tops of the fingers or the fingertips. There is one finger contact on the pinky, ring, middle and index finger of each hand (1-8). Also, there are six contacts A-F, A0-F0 on each thumb; three contacts on each inner thumb and three contacts on each outer thumb. As described in more detail below, signals for letters A-Z are generated by contacting one of the finger contacts 1-8 with one of the six thumb contacts A-F, A0-F0 on the same hand.

As shown in Figure 4, there are supplemental finger contacts located along the sides of the user's fingers nearest to the user's thumbs. Any given finger may have three supplemental contacts located proximate to the first, second and third segment of the user's finger. These supplemental contacts are used to generate signals representing special characters. For example, the supplemental contacts located

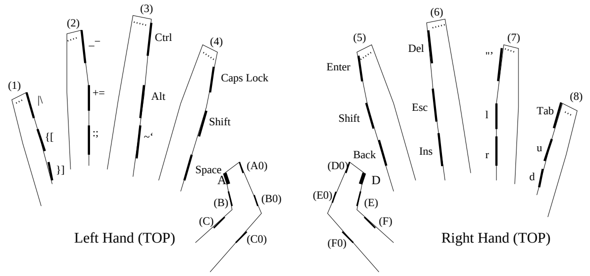


Figure 4: Wired prototype: Location of a) fingertip contacts (1-8) (dashed lines indicate location on the inside of each hand), b) thumb contacts (A(0) - D(0)) and c) 'side-finger' contacts for 'special' character input.

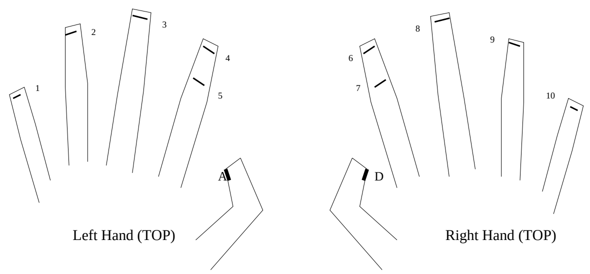


Figure 5: Wired Prototype: Additional contacts 1–10 located on the fingernails and/or neighboring finger segments of each hand for input of numbers 0–9 and (upon using 'Shift') several 'special' characters. Data input is achieved by closing contacts 1–5 with thumb contact A and contacts 6-10 with thumb contact B, respectively.

near the base and middle of the ring and pinky fingers of the right hand are used to generate the same signals that would be generated by the arrow (*l*eft, *r*ight, *u*p and *d*own) keys. A signal is generated by contacting a particular supplemental contact with the thumb contact A or D located on the inside tip of the thumb of the same hand. In other words, "Thumb-coding" is used for these special characters due to the infrequent use of these characters.

As illustrated in Figure 5, additional contacts (denoted by numbers 1-10) may be located on the fingernails (for the index fingers also on the neighboring finger segment) for input of digits 0-9. Upon using the "Shift" contacts, several special characters can be generated using these additional contacts. Data input for digits 1-5 is achieved by closing finger contacts 1-5 in Figure 3 with thumb contact A and data input for digits 6-9 and 0 is achieved by Closing finger contacts 6-10 in Figure 3 with thumb contact D. Using contacts 1-10 (of Figure 5) for input of digits 0-9 still uses fingering or finger movement analogous to the fingering on a standard keyboard; however, now signal input is not achieved by combining a contact on the inside tip of the fingers with a thumb contact but rather by combining a contact on the outside or nail-side of the finger tips with the thumb contact located on the palm-side tip of the thumb of the same hand. In other words, when using contacts 1-10, the thumb contacts A, D represent the upper-most or fourth row of keys on a standard keyboard carrying the number keys.

Tables 1(a) through 1(f) illustrate finger-thumb contact combinations employed for the wired prototype in order to

(a) Left-hand base-line characters.

Finger-Thumb Contact	Character
1-B	A
2-B	S
3-B	D
4-B	F
4-B0	G

(b) Left-hand top-line characters.

Finger-Thumb Contact	Character
1-A	Q
2-A	W
3-A	E
4-A	R
4-A0	T

(c) Left-hand bottom-line characters.

Finger-Thumb Contact	Character
1-C	Z
2-C	X
3-C	C
4-C	V
4-C0	B

(d) Right-hand base-line characters.

Finger-Thumb Contact	Character
5-E0	H
5-E	J
6-E	K
7-E	L
8-E	Z

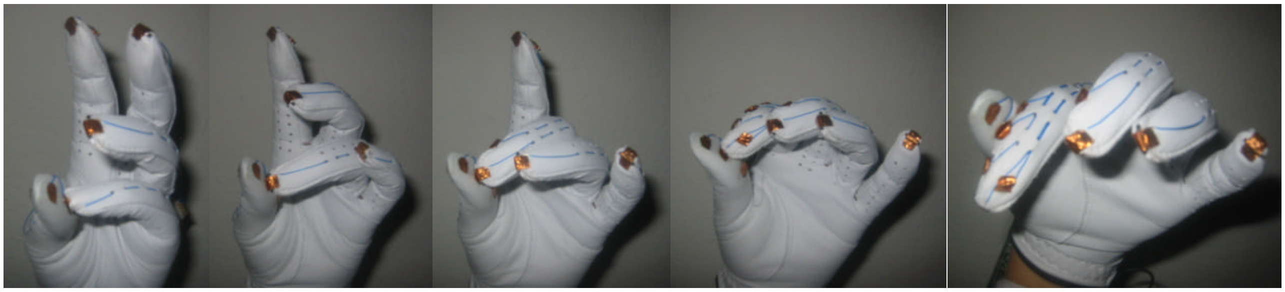
(e) Right-hand top-line characters.

Finger-Thumb Contact	Character
5-D0	Y
5-D	U
6-D	I
7-D	O
8-D	P

(f) right-hand bottom-line characters.

Finger-Thumb Contact	Character
5-F0	N
5-F	M
6-F	<,
7-F	> .
8-F	? /

Table 1: Prototype Key-Map for KITTY device.



(a) (b) (c) (d) (e)

Figure 6: Key mapping example corresponding to Table 1(a). (a) ”a” (b) ”s” (c) ”d” (d) ”f” and (e) ”g”.

	LP	LR	LM1	LI1	LI2	LM2	LI3	LI4	RI4	RI3	RM2	RI2	RI1	RM1	RR	RP
T1	1	2	3	4									7	8	9	0
T2	Q	W	E	R	T	Ctrl	Shift	5	6	Shift	Bksp	Y	U	I	O	P
T3	A	S	D	F	G	Alt	Space			Enter	Up	H	J	K	L	” ′
T4	Z	X	C	V	B	Tab	Left			Right	Dwn	N	M	< ,	> .	? /

Table 2: The KITTY keyboard to hand mapping based on four contacts per thumb (T1-T4) and eight additional finger contacts each for the left (L) and right hand (R).

generate the signals that would be generated by pressing the keys on a standard QWERTY keyboard. The middle thumb contacts B, E are used for characters in the base row. The top thumb contacts A, D are used for characters in the top row and the bottom thumb contacts C, F are used for characters in the bottom row. For example, on a QWERTY keyboard the character “a” is the character that would be typed using the fifth finger of the left hand in the home (base row) position. With the present method, the signal representing the character “a” is generated when the finger contact 1 on the pinky finger of the left hand (see Figure 4) is closed with the thumb contact B (representing the base row). Since the left pinky finger is used to press the “q” character located on the upper row of a keyboard, the signal representing the character “q” is generated by closing the contact between the pinky finger contact 1 and the top thumb contact A (see Figure 4). As can be seen in Figure 3 and Table 1(b), the combination of finger contact 1 and thumb contact A is used to generate the signal representing the character “q”. Since the index finger is used to reach two characters on a given row, an additional thumb contact for each row is located on the back or outside of the user’s thumbs. See contacts A0-F0 in Figure 4. For example, to generate the signal representing the character “f”, the finger-thumb-contact combination 4-B is used. The contact combination used to generate the character “g” is 4-B0 which implies contacting the left index finger 4 with the center contact on the outside of the user’s left thumb B0 (see Figure 4). Note that, due to ergonomic reasons, finger-thumb contact 1-C is not used to generate “z” but rather 8-E. Accordingly, “;” and “ :” are generated by one of the supplemental contacts located on the sides of the fingers as shown in Figure 4.

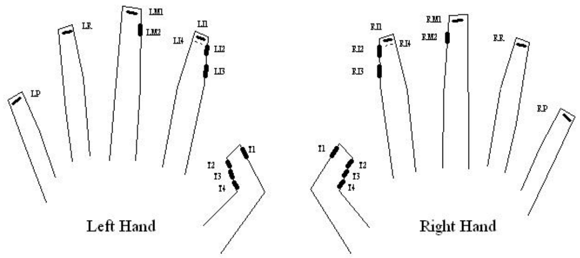
The previously described method to generate numeric data input (using contacts 1-10 in Figure 5), avoids the need of placing four thumb contacts on the inside of each thumb in order to mimica total of four rows on a standard keyboard. The latter is problematic considering the finite size of the electrical contacts and the limited area on the inside thumb which is easily accessible by the finger contacts on the palm-side tips of the fingers.

3.3 Key-Map Wireless Interface

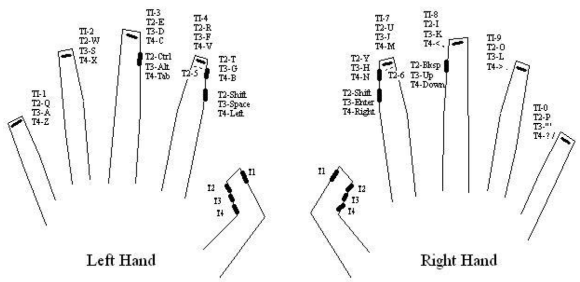
Table 2 illustrates an alternate keyboard mapping based on four thumb contacts (T1-T4) and six finger contacts (LP, LR, LM1-LM2, LI1-LI3) placed on the pinky, ring, middle and index fingers as shown in Figure 7(a). Contact T1 is placed on the thumb nail, T2 on the thumb pad, T3 on the center of the thumb pad and T4 on the bottom of the thumb pad (just below the knuckle). As for the prototype key-map described in Section 3.2, finger-contacts represent rows and thumb-contacts represent columns on a standard keyboard. Here, a new labeling is employed to allow the reader to easily distinguish between both mapping schemes.

According to the alphanumeric mapping summarized in Table 2 and illustrated in Figure 7(b) each finger contact is linked to the column of characters it would press on a keyboard using touch-typing. For example, the middle finger of the left hand, LM1, is linked to “e”, “d”, and “c”, the same keys this finger would normally press on a keyboard using touch-typing. The three thumb contacts match the three letter rows on a normal keyboard. The contact at the tip of the thumb, T2, corresponds to the top row of the keyboard, T3 to the base row, and T4 to the bottom row. Using the left middle finger again as an example, LM1 would yield “e” when touched to T2, “d” when touched to T3, and “c” when touched to T4.

In this modified key-map, each index finger has two more contacts on the side of the finger, LI2, LI3, RI2, and RI3, spaced above and below the top knuckle. The contact above the top knuckle, LI2 or RI2, corresponds to the second col-



(a)



(b)

Figure 7: (a) Finger contacts for wireless KITTY implementation and (b) corresponding Keyboard to glove mapping.

umn of keys (“t”, “g”, and “b” for the left index and “y”, “h”, and “n” for the right index) that the index finger presses on the keyboard during touch-typing. The location of this contact follows the sequential feel of moving from finger to finger and column to column. LM2, LI3, RM2, and RI3 are used for non-alphanumeric keys. Both LI3 and RI3 touched to T2 produce the “Shift” key. The rest of the non-alphanumeric keys are placed as described in Table 2 and cover the most necessary of these keys.

Most of the numeric input is provided by simply placing another contact, T1, on the back of the thumb on the nail. The finger contacts are touched to T1 consecutively in keeping with the traditional keyboard assignments. This configuration for numbers works also well for non-touch-typists since it follows an intuitive method of finger counting. Starting at the left pinky, LP touches T1 to generate the number 1, and then LR, LM1, and LI1 follow for numbers 2 to 4. The number 5 is the exceptional number that uses the LI4 (contact on nail of index finger) to T2 combination. Although an odd combination, it still follows the flow of the series as the tip of the thumb is used for number 5 like the finger pads for the other numbers. The right hand picks up number 6 in a mirror series to the left hand, with 6 to 0 moving through RI3, RI1, RM1, RR, and RP in that order.

As with the key-map discussed in the previous section, a drawback to the described key-to-glove mapping is the bottom row key on the pinky. Touching LP or RP to T4 has an “awkward feel” to it. A momentary slight strain is felt in stretching the pinky towards the lower pad of the thumb. Like learning an awkward chord on a guitar, it is possible that the strain will lessen with prolonged use as the hand adjusts towards this unaccustomed position. One could argue that the keys “Z” and “? /” are used so infrequently that there is no concern regarding continuous straining of the hand.

4. HARDWARE IMPLEMENTATION

Two different implementation strategies were investigated supporting a wired and wireless configuration of the KITTY device.

4.1 Wired Prototype

Initially, the KITTY interface was implemented with minimal effort (Figure 8(a)) using the “Virtually Indestructible” foldable keyboard [4]. The mentioned keyboard was chosen due to the 100% correspondence between the circuit layout on the two membranes of the keyboard and the key-mapping described in Section 3.2, with finger-contacts representing keyboard columns and thumb-contacts representing keyboard rows. Wires were attached to the corresponding locations on the keyboard membrane circuitry using self-adhesive copper tape. Wirewrap was used to keep the design sleek. The wire ends were then collected and attached to two golf gloves according to the layout shown in Figure 4.

Clearly, the wiring or printed circuitry on the two membranes of the “Virtually Indestructible Keyboard” by Grand-

(a)

(b)

Figure 8: The KITTY prototype based on a (a) traditional roll-up keyboard as the keyboard encoder and (b) wireless, Bluetooth enabled encoder.

Tec perfectly matches (without modifications) the mapping proposed in Section 3.2. The GrandTec keyboard uses two membranes, membrane A onto which the wires for the finger contacts are attached and membrane B onto which the wires for the thumb contacts are attached. The contacts for the letters on membrane A are connected in a column format and the contacts on membrane B are connected in a row format.

4.2 Wireless Interface

The latest KITTY prototype incorporates Bluetooth technology for a wireless transfer of key input. Thus, the KITTY may be used with any Bluetooth enabled device including cell phones, PDAs, and PCs with the proper software support. A Wireless Futures RS232 Bluetooth transceiver is connected to a miniature keyboard encoder by Motion Encoders. The electronics are mounted on the wrist as pictured in Figure 8(b). The platform of the enclosure box is large due to prototyping constraints and is expected to be reduced to at least half its current size. The Bluetooth transceiver consumes 1.8mA when in idle mode and 60.9mA while transferring data. Due to the relatively large power requirements of the transceiver, a minimum of four AA batteries are required to provide a battery life of approximately eight hours with average use. The enclosure will increase in height by twice its current size with the addition of a power pack of four AA batteries. The higher platform is awkward and cumbersome, but with low-profile custom batteries as seen on cell phones, it is possible to keep the enclosure small while providing acceptable battery life.

4.3 Design Concepts

Various designs have been considered for implementing the described KITTY interface. The employed prototype design shown in Figure 1 and Figure 9 uses contacts attached to gloves (only one glove is shown in the figures carrying only contacts for alphabetic character input). The contacts are connected by wires to the encoder and (for the wireless implementation) transfer electronics. Preferably, the gloves are fairly thin and lightweight, thereby allowing the user to easily manipulate individual fingers.

An alternative configuration also shown in Figure 9 has contacts mounted on clips or half-rings. The clips are attached to a skeletal structure. The keyboard encoder and transfer electronics (e.g., wireless transmission electronics) are also mounted to the skeletal structure. Figure 9 shows an exemplary skeletal structure for the left hand. The skeletal structure rests against the user's palms (or alternatively on the back of the user's hand) and the contacts are positioned on the user's fingers by placing the clips (on which the contacts are mounted) around the user's fingers and thumbs. The clips can be attached to the skeletal structure in a way that allows adjustment of the positions of the clips in order to accommodate different finger and thumb lengths. The skeletal structure can also be adjustable in order to accommodate hands of varying sizes. The main purpose of the skeletal structure is to preserve the integrity of the overall system.

The designs discussed here can also be configured to be used on one of the user's hands to simulate a "Half-Keyboard". The location of contacts on the hand being used (either right or left) would be the same as in the configuration of contacts when using finger/thumb based touch-typing in the two-handed configuration described above.

5. USER STUDY AND TEST RESULTS

The purpose of these tests was to measure the intuitiveness of the KITTY mapping as well as the comfort of the design. Because of ongoing development and the cost of prototyping the Bluetooth version of the KITTY, a cheaper version of the wireless design was built for user testing. The test prototypes were made from RF controller boards that were stripped from pre-existing keyboards. The contact mapping of the test prototypes were the same as those pictured in Figure 7(b). A custom python program provided a localized remap and test environment with a GUI similar to that of a typing tutor.

5.1 User Study Framework

Testing parameters were established to evaluate three major aspects of KITTY performance. These categories included typing accuracy, speed, and learning curve, as well as some subjective questions about user demographics and general user opinion about the device itself. Even though the QWERTY layout is far from optimal, a broad user community is familiar with it. Therefore, the QWERTY key mapping was selected to create a balance between intuitiveness and typing speed to facilitate ease of use. To validate the strength of the QWERTY layout, several alternative key mappings were, including an alphabetic and optimized layout.

A group consisting of 15 users was first tested on a standard keyboard with a test that included 585 alpha and commonly used punctuation characters. Subjects were tested for typing speed and accuracy and categorized as either an expert touch-typist, average touch-typist, non-touch-typist, or hybrid-typist (someone who uses touch-typing some of

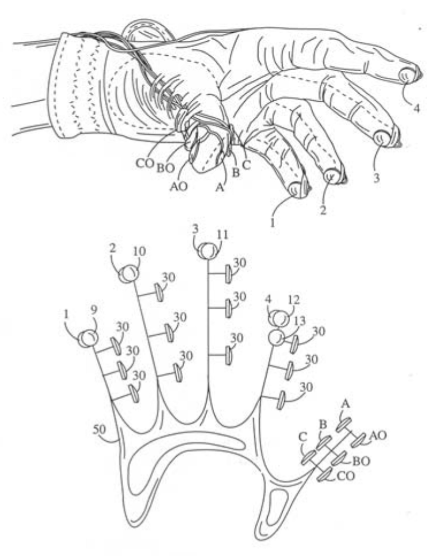
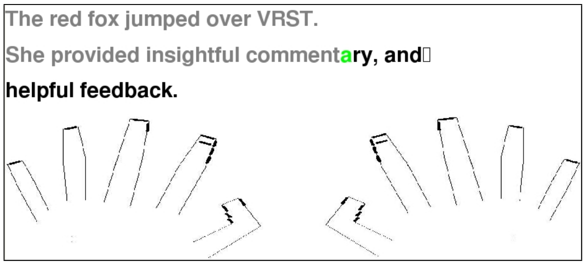
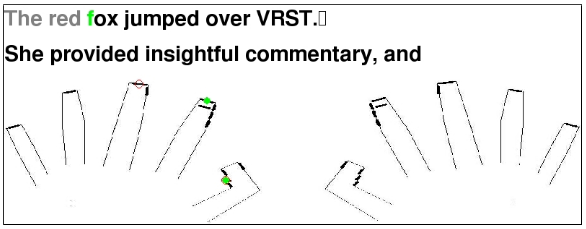


Figure 9: Top: Glove-based prototype design. Bottom: "Click-on" skeletal-structure design.



(a)



(b)

Figure 10: The Kitty training and evaluation interface, (a) the user is asked to press the 'a' key, as indicated by the highlighting of that letter in the text and (b) the user has typed an incorrect key, resulting in the used versus proper contact locations being displayed.

the time). The KITTY test included a test of 93 alpha and punctuation characters to balance the test complexity and not to overwhelm the new user. While the smaller sample size that was selected for the preliminary tests does not reveal the clear advantages of the developing muscle memory observed on a two person test group, a general trends emerge fairly rapidly and are discussed next. The outcomes are expressed as a comparison of the user's KITTY results to the standard keyboard results. Considering that the QWERTY layout may be ideal for the expert touch-typist, a non-touch-typist might have difficulty adapting to the KITTY layout. In order to analyze this premise, the user's ingrained typing skills were matched to the results of their speed and accuracy.

As part of the subjective testing, the most critical question was how quickly the user felt comfortable typing with the KITTY device. Users were questioned about the intuitiveness of the contact mapping, the comfort of the finger-to-thumb placements, and possible applicability to mobile and virtual environments. General comments about the device and test setup were also encouraged.

For the user study, an automated test system was developed to allow careful analysis of the different aspects of each subject's typing style. This testing harness and its data acquisition and logging component were designed to acquire all device events with associated time stamps. The data logs where kept as generic as possible allowing the test metrics to be re-evaluated and improved. In support of this objective a keypress specific logging mechanism was established. For each keystroke three records were acquired, including (1) system time (in milliseconds), (2) desired key ID and (3) pressed key ID. From this data, performance statistics were derived including the mean time between key press events, total amount of time between the first and last key press event, total number of mistakes (comparison of desired and actual key press). In addition, more generic questions could be answered from these logs, including: "How often did the user mis-capitalize a letter?" or "How many times did the user mis-type the letter 'M'?" The interface itself is shown in Figures 10(a) and 10(b). Figure 10(a) depicts the interface as it normally appears. The user has already typed a number of characters, and those are now displayed in a greyed-out color to indicate this fact. The text that the user still must type appears in a second color, and the current key that the user must type is displayed in a third color. Text scrolls upwards through the window as the user types, with text from previously typed lines eventually disappearing at the top of the screen. When the user types an incorrect key, as has been done in Figure 10(b), the system provides a reminder to which fingers the current key corresponds, and which fingers the user had just pressed.

5.2 Test Results

The preliminary tests involved fifteen test subjects. Of these subjects, two were undergraduate students at the senior level, eleven were graduate students, and the last two were working professionals. All of these subjects worked with computers daily and therefore had extensive experience with the QWERTY keyboard. Most of the subjects were full touch-typists with adequate skill, speed, and accuracy. The other subjects also exhibited adequate skill but used a hybrid version of touch-typing that involved four fingers per hand instead of five. Those with the so-called hybrid style of typing looked at the keyboard for reference fairly often in order to find numbers, punctuation, and some alpha characters. Full touch-typists rarely looked at the keyboard except to find rarely used punctuation characters and some numbers.

Results were calculated from eleven out of the fifteen subjects. Four of the data sets were unreliable due incomplete data records and were discarded from the test set. In addition, due to problems with the test prototypes, the KITTY data had to be filtered. Items filtered out of the data sets included repeated key strokes due to a combination of the user's hands shaking and the sensitivity of the contacts. Re-

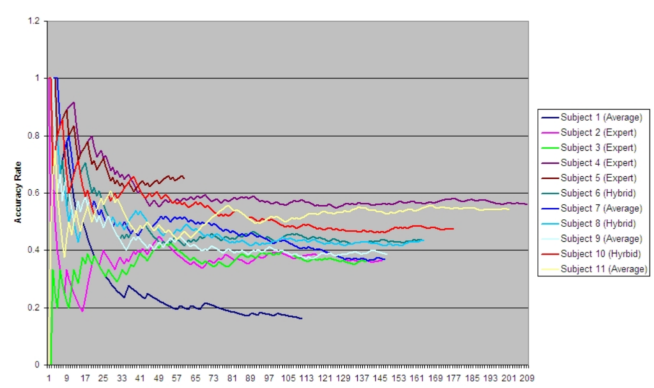
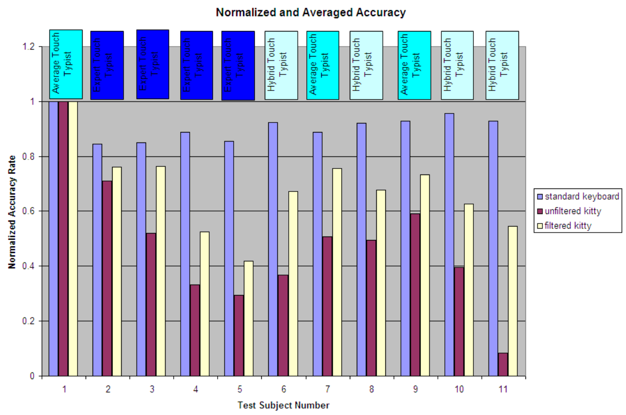
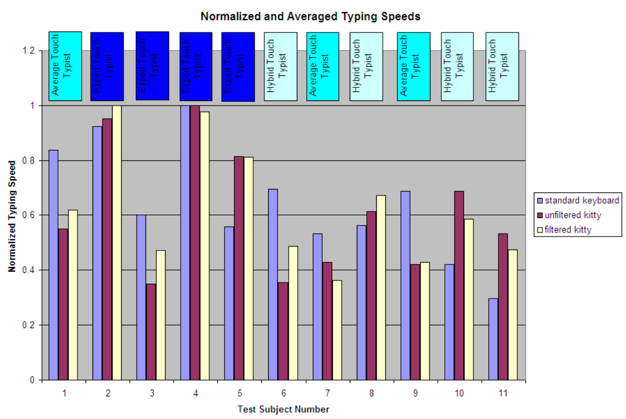


Figure 11: Over a short period of time, all of the test subjects showed some improvement in accuracy on the KITTY.



(a)



(b)

Figure 12: (a) Correlation between accuracy on a standard keyboard and accuracy on the KITTY and (b) correlation between typing speed on a standard keyboard and accuracy on the KITTY.

peat keystrokes similar to those introduced by holding a key for an extended time were also found within the data set. The results shown in Figure 12(a) and Figure 12(b) provide and overview of both the unfiltered and filtered KITTY data.

As can be seen in Figure 11, nearly all of the subjects showed improvement over time. The improvement appears to be logarithmic with a sharp improvement in the first quarter of the test as the user starts to comprehend how the KITTY mapping relates to a QWERTY keyboard. Oddly enough, the users' self-categorized level of expertise did not seem to relate to the level of accuracy. While a correlation may exist, it is possible that this cannot be pictured within our abbreviated test setup. In general, Figure 11 illustrates a quick learning curve for the KITTY within a short period of time.

Figures 12(a) and 12(b) give a comparison of the users' typing skills on both the standard keyboard and KITTY. In order to correlate the results, both the filtered and unfiltered KITTY data records are displayed. Comparison between the two figures shows that those who typed faster on the KITTY usually decreased their accuracy. This is a common problem with typing tests where the assumed primary focus tends to be speed rather than accuracy. As with Figure 11, there does not appear to be a definitive correlation between the users' interpreted skill level and measured performance.

The true typing speeds for the standard keyboard measured between 106-358 strokes/minute with a mean of 240 strokes/minute and an average accuracy of 89%. The typing speeds for the KITTY fell between 14-39 strokes/minute at a mean of 20 strokes/minute and accuracy of 21% for the unfiltered data. For the filtered data the typing speeds were 9-25 strokes/minute with a mean of 13 strokes/minute and an accuracy of 43%. Thus for untutored and unpracticed initial use of the KITTY, users were able to type a character about every 2-7 seconds. All test subjects were first time KITTY users.

All fifteen of the subjects were also evaluated based on subjective data. Users were questioned about how effective their previous touch-typing skills mapped to their skill with the KITTY. One expert user felt it was very effective, while six found it somewhat effective, and four found it not very effective. Two of those who found it not effective were hybrid touch-typists. The rest failed to answer. Eleven users felt at varying degrees that they became more familiar with the KITTY, while two users disagreed. Almost all of the users who were familiar with virtual environments felt that the KITTY would work well as a VR input device. Some problems with the design that users noted were the lack of enough haptic feedback from the flat contacts and difficulty in reaching certain finger combinations. These combinations included the pinky-to-thumb contact and index finger-to-thumb contact. A side note about the awkward contacts comes from discussion with these users about their level of manual dexterity. It was found that those who played musical instruments which required increased stretching and dexterity of the fingers had less difficulty with these awkward combinations. Many users also noted that a short typing tutorial would have been very beneficial in giving users a quicker start on the KITTY. Overall, comments from the users were encouraging and favorable.

6. CONCLUSIONS

With recent advances in portable computing and in particular the design of pocket PCs and eye-glass displays, efforts aimed at the design of portable input devices and techniques have increased. Many portable CHI devices have been and are being developed for intuitive data input on mobile devices. Some of these devices use simple contacts and pressure sensors while others employ more innovative technology such as accelerometers, optic fibers, voice recognition, and laser projection. Although these last four implementations offer fascinating possibilities, their continuous signal processing can have undesirable consequences. Continuous signal devices are open to inaccuracies from unintended finger gestures or other such anomalies. Other devices that need working surfaces, such as those that use pressure sensors, limit the portability of the device to stationary use alone. Contacts seem less impressive a design in their simplicity but offer speed and accuracy with a discrete signal input that is continuously ready. In addition to technology implementations, different systems of key layout such as chording offer smaller designs, but they require the average user to learn a new touch typing technique.

The wireless KITTY prototype attempts to create a simple and easy to use design that is highly portable and ideal for VR/AR environments. The choice of Bluetooth for the wireless implementation employs the latest technology for cross platform needs. The use of contacts fitted to a QWERTY layout gives the typical touch typist a familiar typing environment for easy transition from a standard QWERTY keyboard. The results of the user study compare favorably with this assumption. While the user study test is abbreviated and only tests a short instance of usage, the results show a fast enough acclimation that almost 50% accuracy was achieved after typing only one sentence and without a tutorial. KITTY is currently going through an extended testing period with a selected target group, to evaluate the aforementioned test parameters as well as the user's opinions on the comfort and fatiguing qualities of the gloves when used on a daily basis.

7. ACKNOWLEDGEMENTS

This research was supported in part by the National Science Foundation under Grant Number EIA-0203528 and the California Institute for Telecommunications and Information Technology (Calit2) Graduate Student Fellowship program. The above support is greatly appreciated.

REFERENCES

[1] D. A. Bowman, C. J. Rhoton, and M. S. Pinho. Text input techniques for immersive virtual environments: An empirical comparison. In *Proceedings of the Human Factors and Ergonomic Society Annual Meeting*, pages 2154–2158, 2002.

[2] D. A. Bowman, C. A. Wingrave, J. M. Campbell, and V. Q. Ly. Using pinch gloves; for both natural and abstract interaction techniques in virtual environments. In *Proceedings of the Ninth International Conference on Human-Computer Interaction*, volume 1, pages 629–633, 2001.

[3] M. Fukumoto and Y. Tonomura. Body coupled fingering: Wireless wearable keyboard. In *Proceedings of ACM CHI'97 Conference on Human Factors in Computing Systems*, pages 147–154, 1997.

[4] GrandTec. http://www.grandtec.com/vik.htm, 2005.

[5] Handykey Corporation. www.handykey.com, 2003.

[6] M. E. Horn. Single-hand mounted and operated keyboard. U.S. Patent 5,552,782, September 1996.

[7] R. B. Howard. Wrist-pendent wireless optical keybaord. U.S. Patent 6,097,374, August 2000.

[8] L3 Systems. www.l3sys.com, 2004.

[9] Lightglove. www.lightglove.com, 2003.

[10] MachineBrain.com. www.machinebrain.com, 2003.

[11] E. Matias, I. S. MacKenzie, and W. Buxton. Half-QWERTY: Typing with one hand using your two-handed skills. In *Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems*, volume 2 of *Interactive Experience*, pages 51–52, 1994.

[12] E. Matias, I. S. MacKenzie, and W. Buxton. One-handed touch typing on a QWERTY keyboard. *Human-Computer Interaction*, 11(1):1–27, 1996.

[13] Matias Corporation. www.halfkeyboard.com, 2003.

[14] R. P. McCarty. Wearables central: Softwear. wearables.blu.org, 2003.

[15] V. R. Pratt. Thumb-code: A device-independent digital sign language, July 1998.

[16] K. R. Prince. Finger-mounted computer input device. U.S. Patent 5,581,484, December 1996.

[17] Senseboard Technologies. www.senseboard.com, 2003.

[18] Stanford Wearables. wearables.stanford.edu, 2001.

[19] The MicroOptical Corporation. www.microopticalcorp.com, 2003.

[20] S. Vance, L. Migachyov, W. Fu, and I. Hajjar. Fingerless glove for interacting with a data processing system. U.S. Patent 6,304,840, October 2001.

[21] VKB Limited. www.vkb-tech.com, 2004.

[22] A. Wozniak. Pic-key chorded keyboard. http://mudlist.eorbit.net/ adam/pickey/, 1999.