TipText: Eyes-Free Text Entry on a Fingertip Keyboard

Zheer Xu[1*], Pui Chung Wong[1,2*], Jun Gong[1], Te-Yen Wu[1], Aditya Shekhar Nittala[3], Xiaojun Bi[4], Jürgen Steimle[3], Hongbo Fu[2], Kening Zhu[2], Xing-Dong Yang[1]

Dartmouth College[1], City University of Hong Kong[2], Saarland University[3], Stony Brook University[4]
{zheer.xu.gr, jun.gong.gr, te-yen.wu.gr, xing-dong.yang}@dartmouth.edu,
tony.wong.pc@my.cityu.edu.hk, {nittala, steimle}@cs.uni-saarland.de, xiaojun@cs.stonybrook.edu,
{hongbofu, keninzhu}@cityu.edu.hk

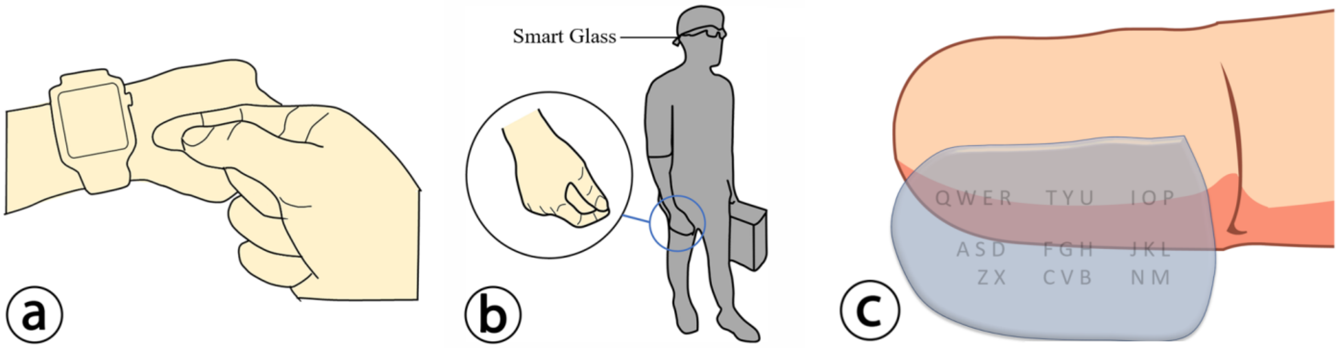

Figure 1(a-b) One-handed text entry using thumb-tip tapping on the index finger in wearable applications; (c) TipText keyboard layout.

ABSTRACT
In this paper, we propose and investigate a new text entry technique using micro thumb-tip gestures. Our technique features a miniature QWERTY keyboard residing invisibly on the first segment of the user's index finger. Text entry can be carried out using the thumb-tip to tap the tip of the index finger. The keyboard layout was optimized for eyes-free input by utilizing a spatial model reflecting the users' natural spatial awareness of key locations on the index finger. We present our approach of designing and optimizing the keyboard layout through a series of user studies and computer simulated text entry tests over 1,146,484 possibilities in the design space. The outcome is a 2×3 grid with the letters highly confining to the alphabetic and spatial arrangement of QWERTY. Our user evaluation showed that participants achieved an average text entry speed of 11.9 WPM and were able to type as fast as 13.3 WPM towards the end of the experiment.

Author Keywords
Micro thumb-tip gesture; text entry; wearable

CSS Concepts
• Human-centered computing~Text Input;

ACM ISBN 978-1-4503-6816-2/19/10...$15.00.
http://dx.doi.org/10.1145/3332165.3347865

INTRODUCTION
As computing devices are being tightly integrated into our daily living and working environments, users often require easy-to-carry and always-available input devices to interact with them in subtle manners. One-handed micro thumb-tip gestures offer new opportunities for such fast, subtle, and always-available interactions especially on devices with limited input space (e.g., wearables) [3]. Very much like gesturing on a trackpad, using the thumb-tip to interact with the virtual world through the index finger is a natural method to perform input. This has become increasingly practical with the rapid advances in sensing technologies, especially in epidermal devices and interactive skin technologies [71, 72]. While many mobile information tasks (e.g., dialing numbers) can be handled using micro thumb-tip gestures [32], text entry is overlooked, despite that text entry comprises of approximately 40% of mobile activity [10].

Using the thumb-tip for text entry on the index finger has several unique benefits. **First**, text input can be carried out using one hand, which is important in mobile scenarios, as the other hand can be occupied by a primary task. **Second**, text input can be carried out unobtrusively, which can be useful in social scenarios, such as in a meeting, where alternative solutions, like texting on a device (e.g., smartphone or watch) or using speech may be socially inappropriate or prone to exposing the users' privacy. **Third**, text input can be carried out without looking at the keyboard (referred to as "eyes-free" in this paper). This can lead to better performance than eyes-on input [82] and save screen real estate for devices with a limited screen space.

* The first two authors are equally contributed.

Despite all these benefits, eyes-free text entry using the thumb-tip and the index finger is challenging because of the lack of input space, missing proper haptic feedback, and lack of a flat and rigid surface on the index finger. A QWERTY keyboard can barely be laid out on the index finger and the keys can be too small to type. Unlike a physical keyboard, typing on the index finger offers little useful haptic feedback to inform the user about which key was selected, making it more difficult for eyes-free typing. Finally, the tip of the index finger is curved and soft, which may impact tapping accuracy on those already small keys.

In this paper, we present TipText, a one-handed text entry technique designed for enabling thumb-tip tapping on a miniature fingertip keyboard on the index finger. TipText features a QWERTY keyboard, familiar to most of today's computer users, in a 2×3 grid layout residing invisibly on the first segment of the index finger (Figure 1c). The design of the grid layout was optimized for eyes-free input by utilizing a spatial model reflecting the users' natural spatial awareness of key locations on the index finger. The efforts of learning to type with eyes-free is largely minimized.

We explored the design space of this new text entry technique in a wide spectrum of design options, ranging from the default QWERTY layout with 26 keys to layouts with a lower number of keys that are larger keys to facilitate tapping. Among the choices of 1,146,484 possibilities, we struck a balance between layout learnability, key size, and word disambiguation introduced by associating the keys with more than one letter. Through a number of user studies and computer simulated typing tests, we compared the performance of various design options and identified an optimized design for TipText. Lastly, we conducted a controlled experiment to evaluate the speed and accuracy of TipText using a proof-of-concept interactive skin overlay placed on the tip of participants' index fingers. Our results revealed that participants could achieve an average of 11.9 (s.e. = 0.5) WPM with 0.30% uncorrected errors.

In summary, our contributions are: (1) a spatial model workable with thumb-tip tapping on fingertip surface (e.g. interactive skin); (2) an optimized keyboard layout design for TipText; and (3) a user study demonstrating the effectiveness of TipText.

RELATED WORK

In this section, we present existing literature in enabling microgesture interaction and text entry on wearables.

Microgesture Interaction

There have been a number of techniques proposed to facilitate input performed with hand and finger gestures. Various sensing approaches have been introduced for input recognition. Camera-based approaches [12, 28, 38, 45, 62, 63], bio-acoustic approaches [6, 17, 29, 40, 64, 79], and electromyography-based approaches [37, 57, 58] have shown effective detection of hand gestures (e.g. fist, hand waving, finger tap on skin) and pinch (e.g. thumb touching other fingers). Hand gestures can also be sensed using electrical impedance tomography [80] and pressure sensor [16] on wrist and arm.

Research improvements in precise sensing [13, 23, 32, 35, 43, 67, 70] can better facilitate microgestures recognition, which provides natural, subtle, and private interaction. Sharma et al. [60] showed that single-hand microgestures are useful in hand-busy conditions via an elicitation study. Soli [43] uses millimeter-wave radar to detect accurate microgestures beside a smartwatch without instrumenting the user. Huang et al. [32] proposed a one-handed and eyes-free thumb-to-fingers interface and revealed that people could locate 16 positions on fingers with skin sensation. ThumbRing [67] calculates the relative angles between two inertial measurement unit (IMU) worn on the thumb and back of the hand to select items with subtle and natural thumb-to-fingers tapping and sliding. FingerPad [13] turns the index finger into a touchpad through magnetic tracking by attaching a Hall sensor grid on the index fingernail, and a magnet on the thumbnail. Skin surface of the index finger is directly used to preserves natural haptic feedback for microgestures. More recent, Pyro [23] enables micro thumb-tip gesture recognition with a pyro-electric passive infrared (PIR) sensor to sense changes in thermal infrared signals emitted from user's finger. These gestures allow subtle, fast, natural, and private interactions in wearable, mobile, and ubiquitous computing applications.

On the other hand, existing work also proposes fabrication processes for thin and flexible interactive skin [36, 44, 69, 71, 72, 74]. iSkin [71] introduced digital fabrication to realize stretchable skin-mounted touch sensors based on biocompatible silicone. SkinMarks [72] provide precise input on fine body landmarks and could be leveraged to detect microgestures on fingers. Recently, Wang et al. [69] demonstrated a soft fluidic user interface by fluidic actuation to achieve dynamic shape change for better fitting on the skin surface. Interactive skin has the potential to facilitate more natural and comfortable microgestures.

Gestural and Non-Visual Text Entry

One of the common approaches for text-entry is using gestures. For example, a continuous stroke can be used to enter a letter (e.g., Graffiti [11], MDITIM [33], EdgeWrite [75]) or a word (Shark^2[39]). Alternatively, a single letter can be entered using several discrete strokes (e.g., H4-Writer [47], QuikWriting [54]). Another commonly applied technique is the non-visual text entry where the user has no visual access to the keyboard. However, most of this work has been focused on enabling novel text entry schemes for visually impaired users [9, 52, 56] or for touch-screen devices [34, 66, 81, 82] where the screen real-estate is considerably larger than that of the finger-tip.

Text Entry on Wearable Devices

Text entry on wearable devices is very challenging since the input space is too small for a QWERTY keyboard with 26 keys. Two-step key selection method is a common approach used in the existing literatures [4, 14, 15, 21, 30, 41, 53, 59, 61]. For example, ZoomBoard [53] expands the size of the QWERTY keyboard by first zooming into a region containing the desired key then select. DualKey [27] and ForceBoard [31] group two letters into one key, thus users could select different letters within a key by using the index or middle finger tap or by force touch. However, most of these two-step selection approaches require two hands and use finger touch as input modality.

Meanwhile, there are various techniques supporting one-handed text entry in wearables. Yu et al. [78] proposed 1D handwriting with a unistroke gesture and SwipeZone [26] adopted a two-step typing method on smartglass touchpad. WrisText [22] allows smartwatch text entry on the watch-wearing hand through wrist whirling on a circular keyboard. FingerT9 [76] leverages thumb-to-fingers touch on a T9 keyboard mapped on finger segments providing haptic feedback. Kim et al. [43] introduced ThumbText, a touch-slide-lift typing method on a ring-sized touchpad for wearable input.

More recent research, including this work, proposes to use statistical keyboard decoder with a spatial model and a language model to facilitate multi-letter keyboard typing [2, 18, 20, 51, 55, 77]. Spatial model treats touch points as noisy signals with a probability distribution over multiple keys. The probability inferred from touch points is then combined with the probability from the language model via the Bayes' Rule to determine the probability of a word candidate. Bi et al. derived FFitts law [8] to model finger touch location with bivariate Gaussian distribution model, providing better approximation of touch model for text entry. Qin et al. [55] optimized a T9-like keyboard by considering word clarity, speed, and learnability besides preserving advantages over the standard QWERTY layout.

Text entry on wearable devices can benefit from eyes-free input in many situations. While QWERTY layout is widely adopted and dominant in daily usage, users might have strong memory on key location. Lu et al. [46] and Zhu et al. [81] explored eyes-free typing on QWERTY keyboard with statistical keyboard decoder and showed promising performance. Furthermore, wearable text entry methods, such as SwipeBoard [14], SwipeZone [26], and 1D handwriting [78] could support eyes-free usage after training and expert users could type without reliance on the visuals. Besides, FingerT9 [76] and WrisText [22] demonstrated the feasibility of eyes-free typing with similar performance as normal typing condition. Speech input is an alternative method for eyes-free text entry while it suffers from inaccuracy in noisy environment, privacy concerns, and is socially inappropriate when used in quiet situation. Thus, other methods for eyes-free text entry are necessary.

Within the existing research, ThumbText [43] is the most relevant work to ours, which proposed a touch-slide-lift text entry approach on a 2×3 grid keyboard for thumb input. ThumbText, however, is not designed for eyes-free text entry and it works on a capacitive touchpad rather than soft fingertip. ThumbText keyboard was designed via two experiments considering accuracy and letter frequency while TipText identified an optimized keyboard layout via iterative simulations with learnability, precision, efficiency, and eyes-free usage as considerations.

DESIGN CONSIDERATIONS

We considered the following factors for designing a text entry method using micro thumb-tip gestures.

Learnability

We considered three types of learnability: learnability of input technique, learnability of keyboard layout, and learnability of eyes-free text entry.

Input techniques for text entry are varying, including tap [53], directional swipe [14], and whirling the wrist [22]. Learnability also varies among different input techniques. For example, tapping keys is easy to learn but swiping directional marks requires more effort. In general, letter-based text entry methods (e.g., Zoomboard [53]) require less learning effort than word-based methods (e.g., WatchWriter [25]) but trade-offs may exist between learnability and efficiency. For example, letter-based input methods can be slower in entering text. In our current exploration, we focus on key tapping for letter-based text entry for the sake of learning.

Various types of keyboard design exist, including the ones following an alphabetical order [22] or a QWERTY layout. With respect to the learnability of keyboard layout, QWERTY is relatively easy to learn due to its wide adoption. Therefore, we used QWERTY in this work. In our design, we also considered preserving the spatial layout of the letters to minimize learning.

Eyes-free typing also requires learning. The adoption of tapping and QWERTY layout minimizes the user's learning efforts in this regard. When typing in an eyes-free context, the user's imaginary location of the desired key, based on his/her spatial awareness, can be different from the actual location of the key. Thus, the user needs to learn the mapping and practice to develop corresponding kinesthetic memory. We minimized the user's learning efforts of eyes-free typing through a system that adopts a spatial model of collected eyes-free input on the index finger.

Eyes-Free Input

We considered two types of eyes-free conditions: typing without looking at the finger movement and typing without looking at the keyboard. Since the user's input space is different from the output space, it is important to free the user's visual attention on fingers because regularly switching attention between where they type and where the output is may introduce significant cognitive overhead and lead to reduced performance. The visual appearance of the keyboard should also be avoided since the screen, if it exists on a very small wearable device (e.g., smartwatch or head-worn display) is tiny. The screen real estate should be dedicated to the text entered by the user rather than the keyboard. On devices without a screen, the entered text can be provided via audio using a wireless headphone. Finally, eyes-free input can facilitate common mobile scenarios, such as walking with the hand hanging along the body.

In general, precise eyes-free input is challenging especially on the small fingertip. We overcame this challenge through a careful design of keyboard layout, which took into consideration the models of both input language and people's natural spatial awareness.

Accuracy and Efficiency

We considered two types of accuracy: accuracy of input technique and accuracy of text entry method. With respect to the accuracy of input technique (e.g., tapping precision), it is hard to locate precisely on the small input space of the index finger because of the "fat finger" issue [68]. However, input does not have to be 100% accurate as the modern text entry systems can tolerate certain level of tapping errors using a statistical decoder [25, 46, 77, 81].

The efficiency of a letter-based text entry method is mostly related to word disambiguation. This issue appears when more than one letters are associated with an enlarged key (like T9) because it is hard to tell which letter the user wants to enter. Therefore, a balance needs to be struck between key size and word disambiguation.

TIPTEXT

With the consideration of these factors, we designed our thumb-tip text entry technique. It comprises a miniature QWERTY keyboard that resides invisibly on the first segment (e.g. distal phalanx) of the index finger. When typing in an eyes-free context, a user selects each key based on his/her natural spatial awareness of the location of the desired key. The system searches in a dictionary for words corresponding to the sequence of the selected keys and provides a list of candidate words ordered by probability calculated using a statistical decoder (see below). The user then swipes the thumb right to enter the selection mode, in which the first word is highlighted. If it is not the desired word, the user swipes the thumb right again to move to the next word in candidate list. The word will be committed automatically upon the user typing the next word (e.g. tapping the first letter of the next word). Additionally, a space will be inserted automatically after the committed word. Auto-complete was implemented following the algorithm described in [77] (see details later). With auto-complete, the user can pick the desired word from the candidate list without having to input all the letters. Finally, the user can swipe the thumb left to delete the last letter.

Principle of Statistical Decoding

TipText uses a statistical decoder [25, 81], which relies on a *spatial model* (SM), describing the relationship between a user's touch locations and the location of the keys, and a *language model* (LM), providing probability distributions of a sequence of words for a certain language (English in our case). Upon a user's entry of a series of letters, the decoder combines probabilities from these two models and generates an overall probability of a word according to Bayes' theorem. In this way, the decoder can provide the user with a list of candidate words ranked by the overall probability. The higher the user's target word is ranked in the candidate list, the less ambiguation issue the corresponding keyboard design has.

In particular, for a given set of touch points on the keyboard $S = [1\ ...,\ sn\ ...,\ sn]$, the decoder finds a word W^* in lexicon L that satisfies:

$$\boldsymbol{W^* = arg\max_{W \in L} P(W|S)} \tag{1}$$

From the Bayes' rule,

$$\boldsymbol{P(W|S) = \frac{P(S|W)P(W)}{P(S)}} \tag{2}$$

Since $P(S)$ is an invariant across words, Equation (1) can be converted to:

$$\boldsymbol{W^* = arg\max_{W \in L} P(S|W)P(W)} \tag{3}$$

where $P(W)$ is obtained from the language model (LM) and $P(S|W)$ is from a spatial model (SM), which can be calculated using the following method.

Assuming that W is comprised of n letters: $c_1, c_2, c_3, ..., c_n$, S has n touch points, and each tap is independent[46], we have:

$$\boldsymbol{P(S|W) = \prod_{i=1}^{n} P(s_i|c_i)} \tag{4}$$

We assumed that touch points for text entry using TipText follows a similar pattern as text entry on a touchscreen. So if the coordinates of s_i is (x_i, y_i), $P(s_i \mid c_i)$ can be calculated using a bivariate Gaussian distribution [7]:

$$\boldsymbol{P(s_i|c_i) = \frac{1}{2\pi\sigma_{ix}\sigma_{iy}\sqrt{1-\rho_i^2}} exp\left[-\frac{z}{2(1-\rho_i^2)}\right]} \tag{5}$$

where

$$\boldsymbol{z \equiv \frac{(x_i-\mu_{ix})^2}{\sigma_{ix}^2} - \frac{2\rho_i(x_i-\mu_{ix})(y_i-\mu_{iy})}{\sigma_{ix}\sigma_{iy}} + \frac{(y_i-\mu_{iy})^2}{\sigma_{iy}^2}} \tag{6}$$

in which (μ_{ix}, μ_{iy}) is the center of the touch point distribution aimed on key c_i ; $\sigma_{ix}\ and\ \sigma_{iy}$ are standard deviations; ρ_i is the correlation.

For auto-complete, our system assumes that users generate no insertion and omission errors and each key is tapped independently [24, 46]. Thus, we can extend (4) and have:

$$\boldsymbol{P(S|W) = \prod_{i=1}^{n} P(S_i|W_i) \times a^{m-n}} \tag{8}$$

where S_i refers to the ith letter of word entered by the user, and W_i refers to the ith letter of W in the dictionary with length between S and $S+8$, in which 8 is determined based on our test. Finally, a refers to the penalty preventing long words with high frequency to be ranked high, and m is the length of W, where $m \geq n$. We set $a = 0.7$ which yields the best compromise between the aggressiveness and candidate coverage for TipText.

Miniature QWERTY Keyboard Design Options

In designing a usable keyboard layout for TipText, we considered two options. First option is to directly adopt a layout with 26 keys. Although keys will be extremely hard to select correctly, the intuition is that the statistical decoder may tolerate many, if not all, the tapping errors, as shown on larger devices like smartwatches [25] and smartphones [81]. The second option is to incorporate larger size but smaller number of keys in a grid layout, like T9 [55] and 1line keyboard [42]. The benefit of this option is that keys are larger, thus easier to acquire but ambiguity may become an issue as each key is associated with more than one letter. It is thus unclear which option is better.

Next, we present a study to explore the feasibility of the first option, in which we collected the data reflecting eyes-free typing behaviors on a miniature QWERTY keyboard.

USER STUDY 1: UNDERSTANDING TYPING ON A MINIATURE QWERTY KEYBOARD WITH 26 KEYS

The goal of this study was to collect data to understand eyes-free typing using the thumb-tip on a keyboard with 26 keys to inform our final keyboard design. We were also interested in knowing whether it is feasible for users to perform text entry based on their natural spatial awareness of a QWERTY layout without practicing ahead of time on locations of keys.

Participants

We recruited 10 right-handed participants (4 female) aged between 20 and 26.

Apparatus

The study was conducted with the Vicon motion tracking system for finger tracking with 1mm accuracy and the Unity 2018.3.5f1 game engine for real-time physical touch estimation. It was a careful decision that we didn't use interactive skin to sense user input because we wanted to minimize sensor influence on users' spatial awareness of key locations. Studies have found that user's spatial acuity and sensitivity can be affected by the presence of the epidermal sensor [19].

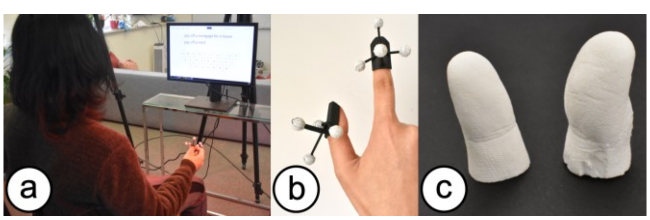

Figure 2. Study setup (a) the participant was typing in front of a monitor surrounded by 5 Vicon cameras in eyes-free condition; (b) markers attached on the fingers; (c) clay models of a participant's fingertips used for 3D scanning.

To obtain precise thumb-tip touch locations on the index finger, we attached markers on the nail of the thumb and index finger (Figure 2b) for the Vicon to track the movement and orientation of the first segments of these two fingers. The data from Vicon was then used to control the movement of the fingers' 3D virtual representation in Unity. The virtual fingers were high-resolution 3D meshes of participants' fingers, which were obtained by scanning clay models of each participant's fingers using a Roland Picza LPX-250RE laser scanner. We used these meshes for real-time physical simulation during the study.

It was observed that people used different thumb regions (e.g. thumb tip, side of the thumb) to perform touch input on the index finger. We thus allowed participants to tap using different regions of the thumb to preserve a natural and comfortable interaction. When the thumb was in contact with the index finger, a collision of the 3D finger meshes was detected in Unity. Ideally, the 3D meshes should deform accordingly to reflect the deformation of the skin of the fingertips. We allowed them to penetrate each other for the sake of simplicity. The user's touch point in a 3D space was estimated using the center of the meshes' contact area, calculated using a mesh intersection algorithm [5]. A touch event was registered upon the size of the intersection exceeding a threshold value. The 3D touch point was then projected to a virtual plane perpendicular to the index finger surface, representing a 2D keyboard. Since participants' fingers were different in size and shape, we manually measured their fingers and transformed the plane for each participant to fit the first segment of the index finger. The projection point, captured using the local coordinate system of that plane, was used as participants' input (Figure 3). Note that while our estimation of tap location may not reflect the real sensor data from the interactive skin, it provided a reasonable estimate to inform the design of our keyboard layout, which was shown effective in our final user study.

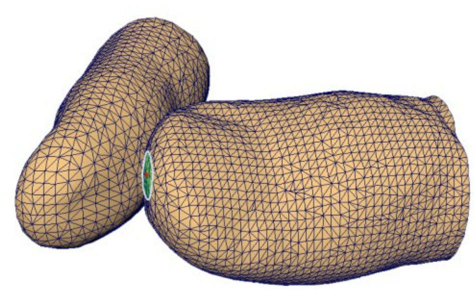

Figure 3. 3D touch simulation of two intersected fingers: green contour refers to contact area of the index finger surface and red dot refers to the input point.

Task and Procedure

Eyes-free thumb-tip text entry tasks were performed with 4 blocks of 10 phrases using a Wizard of Oz keyboard (e.g., no real keyboard was involved [81]). The phrases were picked randomly from the MacKenzie' s phrase set [39, 57, 68, 81]. The same set of 40 phrases was used for all the participants. For each letter, participants tapped on an imaginary key location on the first segment of the index finger using the thumb-tip of their dominant hand based on their natural spatial awareness. They were asked to perform the task using their dominant hand as naturally as possible and assume that the keyboard would correct input errors. Our system always displayed the correct letters no matter where they tapped. In a few cases, however, users accidentally touched the input area on the finger before they were ready to input a new letter, so we designed a left-swipe gesture to delete the last letter to allow users to correct these errors. After entering a phrase, participants pressed a "Done" button to proceed to the next phrase. This process was repeated until they completed all phrases. Participants were encouraged to take a short break between blocks. During the study, a monitor was placed in front of the participant to display the task. A static image of a QWERTY keyboard was also shown on the monitor to remind participants about the positions of keys. Participants sat in a chair with their dominant hand placed on the armrest below participants' sight. Their finger could face any comfortable orientation. An experimenter sat beside them to ensure that their attention was on the monitor.

Prior to the study, the system was calibrated for each participant to ensure that the fingers and their virtual representations in the 3D space were well aligned with each other. Before the study, participants were given 5 to 10 minutes to get familiar with the system without practicing locations of keys. The remaining of the study procedure was similar to that used in [46].

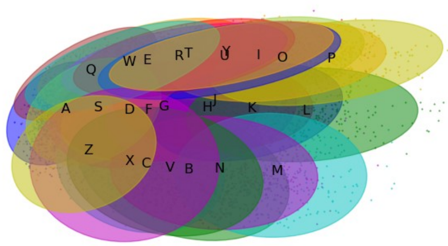

Figure 4. Scatter plots with 95% confidence ellipses of touch points in a 26 key QWERTY keyboard.

Result

Touch points recorded in the study used the local coordinates of a 2D planes, which varied from user to user. Thus, we normalized these touch points to obtain a general distribution. Figure 4 shows all touch points from 10 participants. The touch locations for different keys are shown in different colors. The corresponding letters are shown at centroids of the touch points along with a 95% confidence ellipse. It is obvious that touch locations are noisy with considerable overlaps among different ellipses. This suggests that eyes-free typing on a miniature fingertip keyboard with 26 keys is imprecise. However, it is still observable that centroids of the users' touch points for 26 keys form a QWERTY layout, except that some keys do not clearly separate apart from each other. For example, "Y" and "U" almost overlap. We saw it less of an issue at this moment as a language model can perhaps be helpful in this case. The result of this study is surprising but also very interesting in the sense that despite how small the keys are, there is still a chance that participants might be able to type on a keyboard of 26 keys on the tip of the finger with the help of a statistical decoder. With the collected data, we were able to derive a general spatial model for this keyboard. It was used in a later study to compare this keyboard with other design options.

DESIGN A MINIATURE QWERTY KEYBOARD WITH A GRID LAYOUT OF LESS THAN 26 KEYS

Next, we explored the second option, where a keyboard design incorporates a grid layout. In this layout, keys are larger in size to facilitate tapping but smaller in quantity to fit themselves into the same rectangular input space of the QWERTY keyboard. T9 is an example of this approach. The outcome of this approach was compared against the layout with 26 keys.

Note that larger keys mean that each key may be associated with more than one letter. As such, user input may become ambiguous as it is unclear which letter is the user's target. Therefore, the major challenge of this approach is to find a keyboard layout that can *balance tapping precision and input ambiguation best*. However, there are 1,146,484 possibilities counting all different ways the rectangular space of a keyboard can be divided into a grid and how the 26 letters can be assigned to each key per grid design. It is thus not possible to run user studies to compare the performance of all keyboard designs.

We took an approach similar to Qin et al.'s work [55], where we compared the theoretical performance of all different designs. For each candidate keyboard design, our simulation first calculated the key entries per target word and then found a list of words exactly matched the key entries due to input ambiguation. If the list existed with more than one word, it was ordered by word frequency. No spatial information was involved at this step. The system recorded whether the target word appeared in the top three entries of the list. This approach repeated until it finished all the test words picked from a corpus, which in our case, was the top 15,000 words from the American National Corpus [1], which covers over 95% of common English words [77]. The percentage of times when the target word appeared in the top three entries of the list was calculated as word disambiguation scores for the given keyboard design.

As mentioned above, we only used the language model in our simulation test since the spatial model of a statistical decoder cannot be acquired without a user study. This is fine because the best candidate keyboard design can perform is bounded by $P(W)$ as the spatial mode $P(S|W)$ is 1 at best, in which case, no tapping error occurs. So, the assumption of this comparison test is that tapping errors do not exist regardless how small the keys are. This could be fixed by incorporating heuristics, where top ranked candidates also needed to have large keys.

Result

After the simulator traversed all the possible keyboard designs that confined to the QWERTY's alphabetical arrangement, we selected the ones, which received a word disambiguation score higher than 90%. These designs included keyboard ranging from one to three rows, among which, we pick the ones with the least number of keys. This was to strike a balance between key size and word disambiguation. The remaining 162,972 candidates had a keyboard design in one of 1×5 (132,300), 2×3 (30,240), or 3×2 (432) grid. Figure 5 shows the layout of the top ranked design, which received a word disambiguation score of 94.6%. This score represents the theoretical upper bound of all possible designs in these three grids.

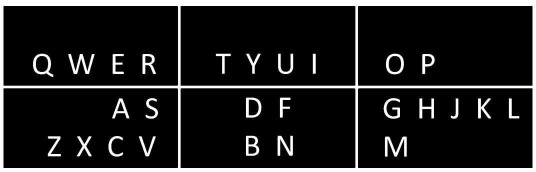

Figure 5. keyboard layout which received highest score in language model.

Note that an issue with this design is that many letters are shifted away from their original locations. For example, "G" and "V" are both in the horizontal center of a smartphone keyboard, but now neither of them resides inside the middle key in the second row. This is due to the result of maximizing word disambiguation. The trade-off is learnability as people can no longer rely on their existing knowledge of the layout of a smartphone keyboard. Instead, they will have to learn new letter locations before they can start eyes-free typing. We thus investigated an extra design criterion, which restricted letter assignments to follow their original locations strictly unless the letter resides at the boundary of two keys (e.g., "G" originally resides on the boundary of the two keys in the second row under a 3×2 grid). In this case, we considered the possibilities for the letter to be assigned to either key. By applying this rule, only 50 qualified out of all 162,972 candidates. This included 16 for the 1×5 grid, 32 for the 2×3 grid, and 2 for the 3×2 grid (see appendix).

Our next step was to obtain an understanding of potential users' natural spatial awareness of key locations in these three grid layouts. We were interested in knowing how grids differ in terms of tapping precision. The answer to these questions helped us to derive a spatial model for each of the three candidate grid layouts, which could be used to form a complete statistical decoder with the language model to estimate the performance of the different keyboard designs associated with these grids in a more realistic setting.

USER STUDY 2: UNDERSTANDING TAPPING PRECISION ON MINIATURE GRID LAYOUTS

The goal of this study was to derive a spatial model for each of these three grid layouts. Note that at this stage, the assignment of 26 letters to grid keys was yet to be determined so we had to replace the text entry task by a target acquisition task in which participants were instructed to acquire cells in a grid. So, the spatial models obtained from our study served as a close approximation of the spatial models for acquiring keyboard keys, which in our case were identical in size and location as the grid cells.

Participants

We recruited 12 right-handed participants (4 female) aged from 20 to 26 for counterbalancing grid conditions.

Apparatus

We used the same apparatus as in Study 1.

Task and Procedure

The task required participants to select a target cell in one of the three tested grid layouts by tapping somewhere on the first segment of the index finger using the thumb-tip of dominant hand. Since letter assignment was not considered in this study, targets were generated in a random order instead of following a corpus.

The grid layouts were introduced to participants by an experimenter describing the number of rows and columns. During the study, no visual grid layout was shown to the user. Instead the target was indicated by row and column number to avoid influencing participants' tapping behavior. Participants were asked to perform the task using their dominant hand as fast and as accurately as possible without looking at the fingers. Upon the end of a trial, a new target appeared. This process was repeated until participants completed all trials. Prior to the study, participants were given 5 to 10 minutes to get familiar with the system and the representation of location in row and column number.

Study Design

This study employed a within subject design with three grid layout conditions: 1×5, 2×3, and 3×2. The order of three conditions were counter-balanced among participants and the target location was presented randomly. Each target in a grid repeated 50 times.

Result

Figure 6 shows the distributions of all touch points from 12 participants for each of three grid layouts. The touch locations for different cells are shown in different colors. The centroids of points for all cells and the 95% confidence ellipses are also shown.

As expected, the touch locations are less noisy than on the layout with 26 keys. There is still overlap among the ellipses for all three grids. This suggests that tapping on the tested grid layouts based on participants' imagination and spatial awareness is still inaccurate. However, the touch points are better separated. Among the three tested grids, we observed less overlap on the 2×3 and 3×2 grids than on the 1×5 grid. This is because the cells are wider in these grids. It is promising that centroids of touch points are all well separated for all three grids. They all follow the same geometry of the three tested grids. This suggests that participants were able to identify the location of the grid

Figure 6. Scatter plots with 95% confidence ellipses of touch points in three grid layouts.

cells using their spatial awareness without looking at their fingers. Although tapping precision tended to be low, we expected that a keyboard decoder could tolerate the errors.

We derived a general spatial model per grid layout using the data collected in this study. It was then used along with the language model to form a statistical decoder, which was used in the next simulation test to help us identify the most suitable keyboard design for TipText.

DETERMINE TIPTEXT KEYBOARD LAYOUT

With the general statistical decoders obtained for the keyboard with 26 keys (default keyboard) and the three grid layouts, we were able to conduct another simulation, in which we simulated text entry on the default keyboard by the 10 participants from Study 1 and on the 50 grid candidates by the 12 participants from Study 2.

We assumed that typing using TipText is similar to typing on a soft keyboard in that users' touch locations follow a bivariate Gaussian distribution [7]. Therefore, the location of the user's touch input was generated based on the bivariate Gaussian distribution of the individual spatial model. For each target word, the generated touch points served as input for the statistical decoder and the simulation checked whether the target word appeared in the top 3 entries of the list. The process was repeated like the first simulation. Since the touch points generated from different participants' models are different, the word disambiguation scores for each candidate keyboard layout differed among participants. Therefore, an average score was calculated to represent the performance of each candidate.

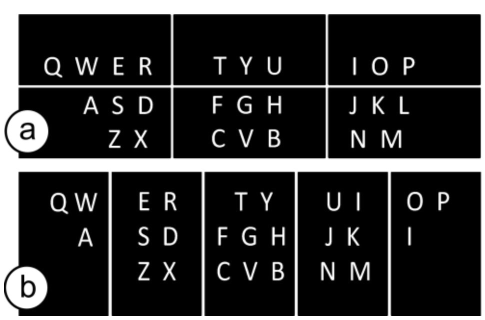

Figure 7. (a) The final keyboard layout; (b) The layout with lowest score.

Result

The default keyboard received an average score of 71.6%. On the other hand, among the 50 grid layout candidates, 10 had a disambiguation score above 80%. All of them were in a 2×3 grid. The top ranked layout (Figure 7a) scored an average of 82.38%. It was also the one scored the highest by 9 out of 12 participants. The winning layout outperformed the one ranked the lowest (Figure 7b) by 45.83%. It also outperformed the default layout by 10.78%. We thus use this layout for TipText.

TIPTEXT HARDWARE

We developed an interactive skin overlay for TipText. The thin and flexible device measures ~2.2 × 2.2cm. It contains a printed 3×3 capacitive touch sensor matrix. The sensor features diamond shaped electrodes of 5 mm diameter and 6.5mm center-to-center spacing.

Our sensor development went through an iterative approach. We first developed a prototype using conductive inkjet printing on PET film using a Canon IP100 desktop ink-jet printer filled with conductive silver nanoparticle ink (Mitsubishi NBSIJ–MU01) [65]. Once the design was tested and its principled functionality on the finger pad confirmed, we created a second prototype with a flexible printed circuit (FPC), which gave us more reliable reading on sensor data (Figure 8b). It is 0.025 – 0.125 mm thick and 21.5mm × 27mm wide. Finally, we developed a highly conformal version on temporary tattoo paper (~30-50 µm thick). We screen printed conductive traces using silver ink (Gwent C2130809D5) overlaid with PEDOT: PSS (Gwent C2100629D1). A layer of resin binder (Gwent R2070613P2) was printed between the electrode layers to isolate them from each other. Two layers of temporary tattoos were added to insulate the sensor from the skin.

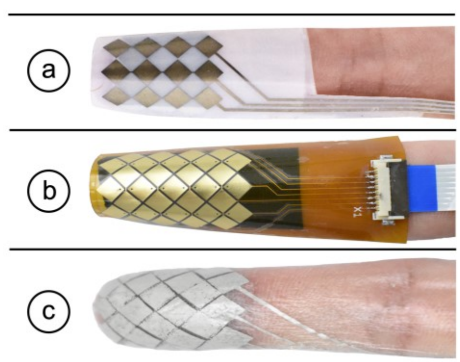

Figure 8. (a) first prototype with PET film; (b) second prototype with FPC; (c) third prototype on temporary tattoo paper.

The finished sensors were controlled using an Arduino Nano with a MPR121 touch sensing chip. The raw capacitive data from each channel was transmitted at a frequency of 100Hz. Software that interpolates the electrode data was implemented in C# based on the algorithm described in the touch controller spec sheet.

We tested TipText on the FPC and tattoo version and decided to use the FPC version for our user study due to its mechanical robustness and durability. We used the tattoo version only for demonstration in this paper.

USER STUDY 3: PERFORMANCE EVALUATION

We conducted a user study to evaluate the performance of TipText. We were also interested in measuring how well our keyboard design worked on a current state-of-the-art micro thumb-tip gesture sensor.

Participants

We recruited 12 right-handed participants (2 female) aged between 20 and 27. All the participants are familiar with the QWERTY keyboard.

Apparatus

The study was conducted using the interactive skin prototype developed using FPC. During the study, participants sat in a chair and placed their hands as the same as study one. An experimenter sat beside the participant to ensure that the participant's attention was on the monitor. Test phrases and top three candidates were shown on a monitor, placed at a comfortable distance from the participant which simulated the situation where a near-eye display is available to the user. Swipe gestures were used to allow participants to navigate the candidate list and delete the last entered letter. A static image of the TipText keyboard was shown on the monitor to remind participants about the positions of keys during training while it was hidden during the study.

Procedure and Experimental Design

The sensor was calibrated for each participant prior to the study by having them tap three edge locations on the first segment of the index finger (e.g., tip and the two ends of the edge of the segment). This was to ensure that the sensor readings of the skin overlay was largely aligned with the spatial model obtained in the previous study. Prior to the experiment, participants were asked to practice for as long as they wanted. During the study, participants transcribed 4 blocks, each containing 10 phrases picked randomly from the MacKenzie's phrase set [50]. The same set of 40 phrases was used for all participants. No phrase was repeated. After entering a phrase, participants pressed the button of a mouse placed on a table with their non-wearing hands to proceed to the next phrase. This process was repeated until they completed all the phrases. The experimental session lasted around 40 minutes, depending on participant speed. We collected 480 phrases (12 participants × 4 blocks × 10 phrases) in the study.

Result and Discussion

We analyzed the data using a one-way repeated measures ANOVA and Bonferroni corrections for pair-wise comparisons. For violations to sphericity, we used a Greenhouse-Geisser adjustment for degrees of freedom.

Text-Entry Speed

ANOVA yielded a significant effect of Block ($F(3) = 20.529$, $p < 0.001$). The average text entry speed was 11.9 WPM (s.e. = 0.5). Figure 9 shows the mean WPM by block, which demonstrates a performance improvement over practice. Post-hoc pair-wise comparisons showed a significant difference between first and second block ($p < 0.05$). Participants achieved 10.5 WPM (s.e. = 0.6) in the first block and the speed increased to 13.3 WPM (s.e. = 0.5) in the last block with an improvement of 27%. It is exciting that participants were able to achieve a fairly good speed even in the first block, suggesting that participants were able to pick up TipText relatively quickly.

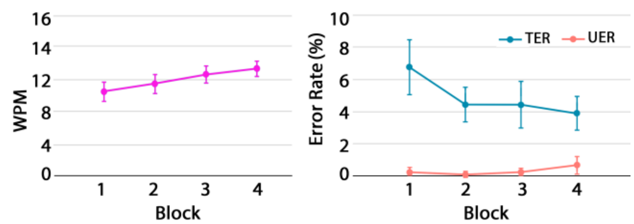

Figure 9. Text entry speed, mean UER and TER across 4 blocks.

Error Rate

Error rate is reported based on uncorrected error rate (UER) and total error rate (TER). Uncorrected errors were the errors found in the final input phrases whereas total errors included both corrected and uncorrected errors.

ANOVA yielded a significance effect of Block on TER ($F(3) = 4.986$, $p < 0.01$). Typing speed increased with the decrease in errors. This suggests that correcting errors was the major source that prevented participants from typing faster but participant were mostly able to identify errors and correct them as there was no significant effect of Block on UER ($F(3) = 2.396$, $p > 0.05$).

Overall, the average TER and UER was 4.89% (s.e. = 0.66%) and 0.30% (s.e. = 0.33%) respectively. Figure 9 shows TER and UER by block. The average TER in the first block was 6.75% (s.e. = 0.85%), and it improved significantly in the last block (3.88%, s.e. = 0.53%). The average UER was 0.30%. (s.e. = 0.33%), which did not change significantly across all blocks. Our observation suggests that when a target word fell outside of the top three suggestions, participants tended to delete the word and retype instead of exploring further down the list even the candidate was sometimes only a swipe away. This is an interesting behavior as it suggests that three might not be the optimized number for showing candidates.

Auto-Complete Rate

We calculated auto-complete rate of a word by dividing the number of automatically filled letters by the length of that word. The overall auto-complete rate was thus the mean of the auto-complete rate of all tested words.

Overall, auto-complete rate was 14.91% (s.e. = 2.39%) for all the input words across all four blocks. We found that text entry speed without auto-complete on Block 4 was 13.3 × (100% - 14.91%) = 11.3 WPM. There was no significant effect of Block on auto-complete ($F(3) = 2.406$, $p > 0.05$) Over the four blocks, the mean standard deviation was 0.74%. This suggested that participants used auto-complete consistently throughout even getting more familiar with the keyboard layout.

DISCUSSION, LIMITATIONS, AND FUTURE WORK

In this section, we discuss the insights gained from this work, the limitations, and propose future research.

Text entry speed and error rate. The average speed of TipText was 11.9 WPM but participants were able to achieve 13.3 WPM in the last block. This is faster than the existing finger-based one-handed text-entry technique, FingerT9 (5.42 WPM) , which uses the entire body of all four fingers as the input space for a keypad. The performance of TipText is also comparable with DigiTouch [73], a bimanual text entry technique using the fingers of both hands (avg. 13 WPM). In the context of mobile scenarios, TipText has the advantage of freeing the other hand of the user for other tasks, such as carrying shopping bags. Note that our observation suggested that participants were able to pick up TipText fast even without seeing a keyboard. This is promising in the sense that TipText might be a good option for ultra-small devices without a screen. Our result shows a trend for this speed to continue growing, which suggests that expert performance could be even higher, warranting a longer-term study. Future research will investigate the upper boundary of TipText input speed.

Number of suggestions. Research [48, 49] showed that the number of suggestions could affect the layout performance because searching through the candidate word list requires extra cognitive effort and visual attention. We chose three suggestions in this work to save screen real estate. However, since TipText was designed to avoid showing an on-screen keyboard on a small computing device (e.g., a smartwatch or smart glasses), it is thus possible that more than three candidate words can be shown to the user. We see it an important future research to investigate how many suggestions may affect typing performance and whether an optimal number of suggestions exist for general population.

Statistical decoder. Our current method uses a statistical decoder derived from the general spatial data collected from twelve participants. The bivariate Gaussian distributions vary among different users and a personalized keyboard decoder would theoretically improve individual's typing performance. Future work will focus on developing an adaptive algorithm that can effectively shift the model from general to personal. Additionally, we observed that users' tapping behaviors may vary with different hand postures and contexts such as standing and walking. Therefore, it is important to further investigate adaptive algorithms that can dynamically update the keyboard decoder according to users' instantaneous and historical input.

User study. TipText introduces a new way for people to perform text entry and we see it a promising method in many different mobile and social scenarios. Apart from the sitting condition tested in our study, other scenarios also warrant careful investigation in the future. For example, we plan to investigate the effectiveness and usability of TipText in a walking condition with hand hanging alongside the body, or even with the same handholding an object while performing text entry. We also plan to evaluate the performance of TipText with users with long nails.

Effect of tactile feedback. Unlike existing mobile text entry techniques on a touchscreen device, TipText users rely on the haptic feedback from both thumb and input device (the index finger) to locate a touch. This is important especially when the keys are rather small. However, the FPC-based interactive skin overlay used in the study reduced haptic feedback on the index finger because of its thickness. We expect that TipText could be even easier and faster to use on an ultra-thin overlay, like the third prototype we implemented. Our study showed that the statistical decoder generated using the Vicon was also effective on our skin sensor. We expect that the performance of the decoder could be further exploited with our third prototype.

CONCLUSION

we discuss our approach of designing a micro thumb-tip text entry technique based on a miniature invisible keyboard residing invisibly on the first segment of the index finger. Our design was informed by an iterative design process, involving a series of user studies and computer simulated text entry tasks that explored a wide spectrum of design options of 1,146,484 possibilities, ranging from default QWERTY layout with 26 keys to layouts with larger sized keys to facilitate tapping and a smaller quantity to fit the keys into the same input space on the index finger. We struck a balance between layout learnability, key size, and word disambiguation and came up with a design, which has a 2×3 grid layout with the letters highly confining to the alphabetic and spatial arrangement of QWERTY. The design of this keyboard was optimized for eyes-free input by utilizing a spatial model reflecting users' natural spatial awareness of key locations on the index finger so the user does not need to look at the keyboard when typing. We foresee that micro finger gesture typing has many applications, ranging from mobile, wearable, and AR. Our techniques serves as important groundwork for future investigation in this field.

ACKNOWLEDGEMENTS

We thank Charles J. Carver, Weihao Chen, Jiehui Luo, Sicheng Yin and Dongyang Zhao for video production.

REFERENCES

[1] 2016. American National Corpus. Retrieved March 3, 2019 from http://www.anc.org/

[2] 2015. TouchOne Keyboard. Retrieved March 3, 2019 from http://www.touchone.net/

[3] Aditya Shekhar Nittala, Klaus Kruttwig, Jaeyeon Lee, Roland Bennewitz, Eduard Arzt, Jürgen Steimle. 2019. Like a Second Skin: Understanding How Epidermal Devices Affect Human Tactile Perception. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '19), Paper 380, 16 pages. DOI=http://dx.doi.org/10.1145/3290605.3300610

[4] Sunggeun Ahn, Seongkook Heo and Geehyuk Lee. 2017. Typing on a Smartwatch for Smart Glasses. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces* (ISS '17), 201-209. DOI=http://dx.doi.org/10.1145/3132272.3134136

[5] Pierre Alliez, Stéphane Tayeb, Camille Wormser and reference manual. 2019. 3D Fast Intersection and Distance Computation. In *CGAL User and Reference Manual*, CGAL Editorial Board, 4.14 edition.

[6] Brian Amento, Will Hill and Loren Terveen. 2002. The sound of one hand: a wrist-mounted bio-acoustic fingertip gesture interface. In *CHI '02 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '02), 724-725. DOI=http://dx.doi.org/10.1145/506443.506566

[7] Shiri Azenkot and Shumin Zhai. 2012. Touch behavior with different postures on soft smartphone keyboards. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services* (MobileHCI '12), 251-260. DOI=http://dx.doi.org/10.1145/2371574.2371612

[8] Xiaojun Bi, Yang Li and Shumin Zhai. 2013. FFitts law: modeling finger touch with fitts' law. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '13), 1363-1372. DOI=http://dx.doi.org/10.1145/2470654.2466180

[9] Syed Masum Billah, Yu-Jung Ko, Vikas Ashok, Xiaojun Bi and IV Ramakrishnan. 2019. Accessible Gesture Typing for Non-Visual Text Entry on Smartphones. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '19), Paper 376, 12 pages. DOI=http://dx.doi.org/10.1145/3290605.3300606

[10] Barry Brown, Moira McGregor and Donald McMillan. 2014. 100 days of iPhone use: understanding the details of mobile device use. In *Proceedings of the 16th international conference on Human-computer interaction with mobile devices and services* (MobileHCI '14), 223-232. DOI=http://dx.doi.org/10.1145/2628363.2628377

[11] Steven J Castellucci and I Scott MacKenzie. 2008. Graffiti vs. unistrokes: an empirical comparison. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '08), 305-308. DOI=http://dx.doi.org/10.1145/1357054.1357106

[12] Liwei Chan, Yi-Ling Chen, Chi-Hao Hsieh, Rong-Hao Liang and Bing-Yu Chen. 2015. CyclopsRing: Enabling Whole-Hand and Context-Aware Interactions Through a Fisheye Ring. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology* (UIST '15), 549-556. DOI=http://dx.doi.org/10.1145/2807442.2807450

[13] Liwei Chan, Rong-Hao Liang, Ming-Chang Tsai, Kai-Yin Cheng, Chao-Huai Su, Mike Y. Chen, Wen-Huang Cheng and Bing-Yu Chen. 2013. FingerPad: private and subtle interaction using fingertips. In *Proceedings of the 26th annual ACM symposium on User interface software and technology* (UIST '13), 255-260. DOI=http://dx.doi.org/10.1145/2501988.2502016

[14] Xiang 'Anthony' Chen, Tovi Grossman and George Fitzmaurice. 2014. Swipeboard: a text entry technique for ultra-small interfaces that supports novice to expert transitions. In *Proceedings of the 27th annual ACM symposium on User interface software and technology* (UIST '14), 615-620. DOI=http://dx.doi.org/10.1145/2642918.2647354

[15] Hyeonjoong Cho, Miso Kim and Kyeongeun Seo. 2014. A text entry technique for wrist-worn watches with tiny touchscreens *Proceedings of the adjunct publication of the 27th annual ACM symposium on User interface software and technology* (UIST '14), 79-80. DOI=http://dx.doi.org/10.1145/2658779.2658785

[16] Artem Dementyev and Joseph A. Paradiso. 2014. WristFlex: low-power gesture input with wrist-worn pressure sensors. In *Proceedings of the 27th annual ACM symposium on User interface software and technology* (UIST '14), 161-166. DOI=http://dx.doi.org/10.1145/2642918.2647396

[17] Travis Deyle, Szabolcs Palinko, Erika Shehan Poole and Thad Starner. 2007. Hambone: A bio-acoustic gesture interface. In *2007 11th IEEE International Symposium on Wearable Computers*, IEEE, 3-10. DOI=http://dx.doi.org/10.1109/ISWC.2007.4373768

[18] Mark D. Dunlop, Andreas Komninos and Naveen Durga. 2014. Towards high quality text entry on smartwatches. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '14), 2365-2370. DOI=http://dx.doi.org/10.1145/2559206.2581319

[19] Rachel Freire, Cedric Honnet and Paul Strohmeier. 2017. Second Skin: An Exploration of eTextile Stretch Circuits on the Body. In *Proceedings of the Eleventh International Conference on Tangible, Embedded, and Embodied Interaction* (TEI '17), 653-658. DOI=http://dx.doi.org/10.1145/3024969.3025054

[20] Markus Funk, Alireza Sahami, Niels Henze and Albrecht Schmidt. 2014. Using a touch-sensitive

wristband for text entry on smart watches. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '14), 2305-2310. DOI=http://dx.doi.org/10.1145/2559206.2581143

[21] Timo Götzelmann and Pere-Pau Vázquez. 2015. InclineType: An Accelerometer-based Typing Approach for Smartwatches. In *Proceedings of the XVI International Conference on Human Computer Interaction* (Interacción '15), 1-4. DOI=https://doi.org/10.1145/2829875.2829929

[22] Jun Gong, Zheer Xu, Qifan Guo, Teddy Seyed, Xiang 'Anthony' Chen, Xiaojun Bi and Xing-Dong Yang. 2018. WrisText: One-handed Text Entry on Smartwatch using Wrist Gestures. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (CHI '18), 1-14. DOI=http://dx.doi.org/10.1145/3173574.3173755

[23] Jun Gong, Yang Zhang, Xia Zhou and Xing-Dong Yang. 2017. Pyro: Thumb-Tip Gesture Recognition Using Pyroelectric Infrared Sensing *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (UIST '17), 553-563. DOI=http://dx.doi.org/10.1145/3126594.3126615

[24] Joshua Goodman, Gina Venolia, Keith Steury and Chauncey Parker. 2002. Language modeling for soft keyboards. In *Proceedings of the 7th international conference on Intelligent user interfaces* (IUI '02), 194-195. DOI=http://dx.doi.org/10.1145/502716.502753

[25] Mitchell Gordon, Tom Ouyang and Shumin Zhai. 2016. WatchWriter: Tap and Gesture Typing on a Smartwatch Miniature Keyboard with Statistical Decoding. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (CHI '16), 3817-3821. DOI=http://dx.doi.org/10.1145/2858036.2858242

[26] Tovi Grossman, Xiang Anthony Chen and George Fitzmaurice. 2015. Typing on Glasses: Adapting Text Entry to Smart Eyewear *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services* (MobileHCI '15), 144-152. DOI=http://dx.doi.org/10.1145/2785830.2785867

[27] Aakar Gupta and Ravin Balakrishnan. 2016. DualKey: Miniature Screen Text Entry via Finger Identification. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (CHI '16), 59-70. DOI=http://dx.doi.org/10.1145/2858036.2858052

[28] Chris Harrison, Hrvoje Benko and Andrew D. Wilson. 2011. OmniTouch: wearable multitouch interaction everywhere. In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (UIST '11), 441-450. DOI=http://dx.doi.org/10.1145/2047196.2047255

[29] Chris Harrison, Desney Tan and Dan Morris. 2010. Skinput: appropriating the body as an input surface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '10), 453-462. DOI=http://dx.doi.org/10.1145/1753326.1753394

[30] Jonggi Hong, Seongkook Heo, Poika Isokoski and Geehyuk Lee. 2015. SplitBoard: A Simple Split Soft Keyboard for Wristwatch-sized Touch Screens *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (CHI '15), 1233-1236. DOI=http://dx.doi.org/10.1145/2702123.2702273

[31] Min-Chieh Hsiu, Da-Yuan Huang, Chi An Chen, Yu-Chih Lin, Yi-ping Hung, De-Nian Yang and Mike Chen. 2016. ForceBoard: using force as input technique on size-limited soft keyboard. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct* (MobileHCI '16), 599-604. DOI=http://dx.doi.org/10.1145/2957265.2961827

[32] Da-Yuan Huang, Liwei Chan, Shuo Yang, Fan Wang, Rong-Hao Liang, De-Nian Yang, Yi-Ping Hung and Bing-Yu Chen. 2016. DigitSpace: Designing Thumb-to-Fingers Touch Interfaces for One-Handed and Eyes-Free Interactions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (CHI '16), 1526-1537. DOI=http://dx.doi.org/10.1145/2858036.2858483

[33] Poika Isokoski and Roope Raisamo. 2000. Device independent text input: A rationale and an example. In *Proceedings of the working conference on Advanced visual interfaces* (AVI '00), 76-83. DOI=http://dx.doi.org/10.1145/345513.345262

[34] João Guerreiro, André Rodrigues, Kyle Montague, Tiago Guerreiro, Hugo Nicolau, and Daniel Gonçalves. 2015. TabLETS Get Physical: Non-Visual Text Entry on Tablet Devices. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (CHI '15), 39-42. DOI=http://dx.doi.org/10.1145/2702123.2702373

[35] Hsin-Liu Kao, Artem Dementyev, Joseph A. Paradiso and Chris Schmandt. 2015. NailO: Fingernails as an Input Surface. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (CHI '15), 3015-3018. DOI=http://dx.doi.org/10.1145/2702123.2702572

[36] Hsin-Liu Kao, Christian Holz, Asta Roseway, Andres Calvo and Chris Schmandt. 2016. DuoSkin: rapidly prototyping on-skin user interfaces using skin-friendly materials. In *Proceedings of the 2016 ACM International Symposium on Wearable Computers* (ISWC '16), 16-23. DOI=http://dx.doi.org/10.1145/2971763.2971777

[37] Frederic Kerber, Pascal Lessel, Antonio Kr, #252 and ger. 2015. Same-side Hand Interactions with Arm-placed Devices Using EMG. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing System* (CHI EA '15), 1367-1372.

DOI=http://dx.doi.org/10.1145/2702613.2732895

[38] David Kim, Otmar Hilliges, Shahram Izadi, Alex D. Butler, Jiawen Chen, Iason Oikonomidis and Patrick Olivier. 2012. Digits: freehand 3D interactions anywhere using a wrist-worn gloveless sensor. In *Proceedings of the 25th annual ACM symposium on User interface software and technology* (UIST '12), 167-176. DOI=http://dx.doi.org/10.1145/2380116.2380139

[39] Per-Ola Kristensson and Shumin Zhai. 2004. 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 '04), 43-52. DOI=http://dx.doi.org/10.1145/1029632.1029640

[40] Gierad Laput, Robert Xiao and Chris Harrison. 2016. ViBand: High-Fidelity Bio-Acoustic Sensing Using Commodity Smartwatch Accelerometers. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (UIST '16), 321-333. DOI=http://dx.doi.org/10.1145/2984511.2984582

[41] Luis A. Leiva, Alireza Sahami, Alejandro Catala, Niels Henze and Albrecht Schmidt. 2015. Text Entry on Tiny QWERTY Soft Keyboards. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (CHI '15), 669-678. DOI=http://dx.doi.org/10.1145/2702123.2702388

[42] Frank Chun Yat Li, Richard T. Guy, Koji Yatani and Khai N. Truong. 2011. The 1line keyboard: a QWERTY layout in a single line. In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (UIST '11), 461-470. DOI=http://dx.doi.org/10.1145/2047196.2047257

[43] Jaime Lien, Nicholas Gillian, M. Emre Karagozler, Patrick Amihood, Carsten Schwesig, Erik Olson, Hakim Raja and Ivan Poupyrev. 2016. Soli: ubiquitous gesture sensing with millimeter wave radar. ACM Trans. Graph. 5, 4, Article 142 (July 2016), 19 pages. DOI=http://dx.doi.org/10.1145/2897824.2925953

[44] Joanne Lo, Doris Jung Lin Lee, Nathan Wong, David Bui and Eric Paulos. 2016. Skintillates: Designing and Creating Epidermal Interactions. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems* (DIS '16), 853-864. DOI=http://dx.doi.org/10.1145/2901790.2901885

[45] Christian Loclair, Sean Gustafson and Patrick Baudisch. 2010. PinchWatch: a wearable device for one-handed microinteractions. In *Proc. MobileHCI*.

[46] Yiqin Lu, Chun Yu, Xin Yi, Yuanchun Shi and Shengdong Zhao. 2017. BlindType: Eyes-Free Text Entry on Handheld Touchpad by Leveraging Thumb's Muscle Memory. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 1, 2, Article 18 (June 2017), 24 pages. DOI=http://dx.doi.org/10.1145/3090083

[47] I Scott MacKenzie, R William Soukoreff and Joanna Helga. 2011. 1 thumb, 4 buttons, 20 words per minute: Design and evaluation of H4-Writer. In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (UIST '11), 471-480. DOI=http://dx.doi.org/10.1145/2047196.2047258

[48] I. Scott MacKenzie, Hedy Kober, Derek Smith, Terry Jones and Eugene Skepner. 2001. LetterWise: prefix-based disambiguation for mobile text input. In *Proceedings of the 14th annual ACM symposium on User interface software and technology* (UIST '01), 111-120. DOI=http://dx.doi.org/10.1145/502348.502365

[49] I. Scott MacKenzie and R. William Soukoreff. 2003. Phrase sets for evaluating text entry techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing System* (CHI EA '03), 754-755. DOI=http://dx.doi.org/10.1145/765891.765971

[50] I. Scott MacKenzie and Shawn X. Zhang. 1999. The design and evaluation of a high-performance soft keyboard. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (CHI '99), 25-31. DOI=http://dx.doi.org/10.1145/302979.302983

[51] Aske Mottelson, Christoffer Larsen, Mikkel Lyderik, Paul Strohmeier and Jarrod Knibbe. 2016. Invisiboard: maximizing display and input space with a full screen text entry method for smartwatches. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services* (MobileHCI '16), 53-59. DOI=http://dx.doi.org/10.1145/2935334.2935360

[52] João Oliveira, Tiago Guerreiro, Hugo Nicolau, Joaquim Jorge and Daniel Gonçalves. 2011. BrailleType: unleashing braille over touch screen mobile phones. In *IFIP Conference on Human-Computer Interaction*, Springer, 100-107. DOI=http://dx.doi.org/ 10.1007/978-3-642-23774-4_10

[53] Stephen Oney, Chris Harrison, Amy Ogan and Jason Wiese. 2013. ZoomBoard: a diminutive qwerty soft keyboard using iterative zooming for ultra-small devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '13), 2799-2802. DOI=http://dx.doi.org/10.1145/2470654.2481387

[54] Ken Perlin. 1998. Quikwriting: continuous stylus-based text entry. In *ACM Symposium on User Interface Software and Technology* (UIST '98), 215-216. DOI=http://dx.doi.org/10.1145/288392.288613

[55] Ryan Qin, Suwen Zhu, Yu-Hao Lin, Yu-Jung Ko and Xiaojun Bi. 2018. Optimal-T9: An Optimized T9-like Keyboard for Small Touchscreen Devices. In *Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces* (ISS '18), 137-146.

DOI=http://dx.doi.org/10.1145/3279778.3279786

[56] Mario Romero, Brian Frey, Caleb Southern and Gregory D. Abowd. 2011. BrailleTouch: designing a mobile eyes-free soft keyboard. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services* (MobileHCI '11), 707-709. DOI=http://dx.doi.org/10.1145/2037373.2037491

[57] T Scott Saponas, Desney S. Tan, Dan Morris and Ravin Balakrishnan. 2008. Demonstrating the feasibility of using forearm electromyography for muscle-computer interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '08), 515-524. DOI=http://dx.doi.org/10.1145/1357054.1357138

[58] T. Scott Saponas, Desney S. Tan, Dan Morris, Ravin Balakrishnan, Jim Turner and James A. Landay. 2009. Enabling always-available input with muscle-computer interfaces. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology* (UIST '09), 167-176. DOI=http://dx.doi.org/10.1145/1622176.1622208

[59] Yuan-Fu Shao, Masatoshi Chang-Ogimoto, Reinhard Pointner, Yu-Chih Lin, Chen-Ting Wu and Mike Chen. 2016. SwipeKey: a swipe-based keyboard design for smartwatches. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services* (MobileHCI '16), 60-71. DOI=http://dx.doi.org/10.1145/2935334.2935336

[60] Adwait Sharma, Joan Sol Roo and Jürgen Steimle. 2019. Grasping Microgestures: Eliciting Single-hand Microgestures for Handheld Objects. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (CHI '19), Paper 402, 13 pages. DOI=http://dx.doi.org/10.1145/3290605.3300632

[61] Tomoki Shibata, Daniel Afergan, Danielle Kong, Beste F. Yuksel, I. Scott MacKenzie and Robert J.K. Jacob. 2016. DriftBoard: A Panning-Based Text Entry Technique for Ultra-Small Touchscreens *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (UIST '16), 575-582. DOI=http://dx.doi.org/10.1145/2984511.2984591

[62] Mohamed Soliman, Franziska Mueller, Lena Hegemann, Joan Sol Roo, Christian Theobalt, and Jürgen Steimle. 2018. FingerInput: Capturing Expressive Single-Hand Thumb-to-Finger Microgestures. In *Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces* (ISS '18), 177-187. DOI=http://dx.doi.org/10.1145/3279778.3279799

[63] Srinath Sridhar, Anders Markussen, Antti Oulasvirta, Christian Theobalt and Sebastian Boring. 2017. WatchSense: On- and Above-Skin Input Sensing through a Wearable Depth Sensor. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (CHI '17), 3891-3902. DOI=http://dx.doi.org/10.1145/3025453.3026005

[64] Ke Sun, Yuntao Wang, Chun Yu, Yukang Yan, Hongyi Wen and Yuanchun Shi. 2017. Float: One-Handed and Touch-Free Target Selection on Smartwatches. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (CHI '17), 692-704. DOI=http://dx.doi.org/10.1145/3025453.3026027

[65] Tung Ta, Masaaki Fukumoto, Koya Narumi, Shigeki Shino, Yoshihiro Kawahara and Tohru Asami. 2015. Interconnection and double layer for flexible electronic circuit with instant inkjet circuits. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (UbiComp '15), 181-190. DOI=http://dx.doi.org/10.1145/2750858.2804276

[66] Hussain Tinwala and I. Scott MacKenzie. 2010. Eyes-free text entry with error correction on touchscreen mobile devices. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries* (NordiCHI '10), 511-520. DOI=http://dx.doi.org/10.1145/1868914.1868972

[67] Hsin-Ruey Tsai, Cheng-Yuan Wu, Lee-Ting Huang and Yi-Ping Hung. 2016. ThumbRing: private interactions using one-handed thumb motion input on finger segments. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct* (MobileHCI '16), 791-798. DOI=http://dx.doi.org/10.1145/2957265.2961859

[68] Daniel Vogel and Patrick Baudisch. 2007. Shift: a technique for operating pen-based interfaces using touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '07), 657-666. DOI=http://dx.doi.org/10.1145/1240624.1240727

[69] Yanan Wang, Shijian Luo, Hebo Gong, Fei Xu, Rujia Chen, Shuai Liu and Preben Hansen. 2018. *SKIN+*: Fabricating Soft Fluidic User Interfaces for Enhancing On-Skin Experiences and Interactions. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems* (CHI EA '18), 1-6. DOI=http://dx.doi.org/10.1145/3170427.3188443

[70] Martin Weigel and Jürgen Steimle. 2017. DeformWear: Deformation Input on Tiny Wearable Devices. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 1, 2, Article 28 (June 2017), 23 pages. DOI=http://dx.doi.org/10.1145/3090093

[71] Martin Weigel, Tong Lu, Gilles Bailly, Antti Oulasvirta, Carmel Majidi and Jürgen Steimle. 2015. iSkin: Flexible, Stretchable and Visually Customizable On-Body Touch Sensors for Mobile Computing. In *Proceedings of the 33rd Annual ACM*

Conference on Human Factors in Computing Systems (CHI '15), 2991-3000. DOI=http://dx.doi.org/10.1145/2702123.2702391

[72] Martin Weigel, Aditya Shekhar Nittala, Alex Olwal and Jürgen Steimle. 2017. SkinMarks: Enabling Interactions on Body Landmarks Using Conformal Skin Electronics. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (CHI '17), 3095-3105. DOI=http://dx.doi.org/10.1145/3025453.3025704

[73] Eric Whitmire, Mohit Jain, Divye Jain, Greg Nelson, Ravi Karkar, Shwetak Patel and Mayank Goel. 2017. Digitouch: Reconfigurable thumb-to-finger input and text entry on head-mounted displays. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 1, 3, Article 113 (September 2017), 21 pages. DOI=https://doi.org/10.1145/3130978

[74] Anusha Withana, Daniel Groeger and Jürgen Steimle. 2018. Tacttoo: A Thin and Feel-Through Tattoo for On-Skin Tactile Output. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (UIST '18), 365-378. DOI=http://dx.doi.org/10.1145/3242587.3242645

[75] Jacob O Wobbrock, Brad A Myers and John A Kembel. 2003. 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 technolog* (UIST '03), 61-70. DOI=http://dx.doi.org/10.1145/964696.964703

[76] Pui Chung Wong, Kening Zhu and Hongbo Fu. 2018. FingerT9: Leveraging Thumb-to-finger Interaction for Same-side-hand Text Entry on Smartwatches. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (CHI '18), Paper 178, 10 pages. DOI=http://dx.doi.org/10.1145/3173574.3173752

[77] Xin Yi, Chun Yu, Weijie Xu, Xiaojun Bi and Yuanchun Shi. 2017. COMPASS: Rotational Keyboard on Non-Touch Smartwatches *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (CHI '17), 705-715. DOI=http://dx.doi.org/10.1145/3025453.3025454

[78] Chun Yu, Ke Sun, Mingyuan Zhong, Xincheng Li, Peijun Zhao and Yuanchun Shi. 2016. One-Dimensional Handwriting: Inputting Letters and Words on Smart Glasses. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (CHI '16), 71-82. DOI=http://dx.doi.org/10.1145/2858036.2858542

[79] Cheng Zhang, AbdelKareem Bedri, Gabriel Reyes, Bailey Bercik, Omer T. Inan, Thad E. Starner and Gregory D. Abowd. 2016. TapSkin: Recognizing On-Skin Input for Smartwatches. In *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces* (ISS '16), 13-22. DOI=http://dx.doi.org/10.1145/2992154.2992187

[80] Yang Zhang and Chris Harrison. 2015. Tomo: Wearable, Low-Cost Electrical Impedance Tomography for Hand Gesture Recognition. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology* (UIST '15), 167-173. DOI=http://dx.doi.org/10.1145/2807442.2807480

[81] Suwen Zhu, Tianyao Luo, Xiaojun Bi and Shumin Zhai. 2018. Typing on an Invisible Keyboard. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (CHI '18), 1-13. DOI=http://dx.doi.org/10.1145/3173574.3174013

[82] Suwen Zhu, Jingjie Zheng, Shumin Zhai and Xiaojun Bi. 2019. i'sFree: Eyes-Free Gesture Typing via a Touch-Enabled Remote Control. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (CHI '19), Paper 448, 12 pages. DOI=https://doi.org/10.1145/3290605.3300678.

APPENDIX

Candidate layouts with restricted letter assignments

QW AS Z	ER DF XC	TY GH VB	UI JK NM	OP L

QW AS Z	ER DF XC	TY GH VB	UI J N	OP KL M

QW AS Z	ER DF XC	TY G V	UI HJK BNM	OP L

QW AS Z	ER DF XC	TY G V	UI HJ BN	OP KL M

QW AS Z	ER D X	TY FGH CVB	UI JK NM	OP L

QW AS Z	ER D X	TY FGH CVB	UI J N	OP KL M

QW AS Z	ER D X	TY FG CV	UI HJK BNM	OP L

QW AS Z	ER D X	TY FG CV	UI HJ BN	OP KL M

QW A	ER SDF ZXC	TY GH VB	UI JK NM	OP L

QW A	ER SDF ZXC	TY GH VB	UI J N	OP KL M

QW A	ER SDF ZXC	TY G V	UI HJK BNM	OP L

QW A	ER SDF ZXC	TY G V	UI HJ BN	OP KL M

QW A	ER SD ZX	TY FGH CVB	UI JK NM	OP L

QW A	ER SD ZX	TY FGH CVB	UI J N	OP KL M

QW A	ER SD ZX	TY FG CV	UI HJK BNM	OP L

QW A	ER SD ZX	TY FG CV	UI HJ BN	OP KL M

QWER ASD	TYU FGHJ	IOP KL
ZX	CVBN	M

QWER ASD	TY FGHJ	UIOP KL
ZX	CVBN	M

QWE ASD	RTYU FGHJ	IOP KL
ZX	CVBN	M

QWE ASD	RTY FGHJ	UIOP KL
ZX	CVBN	M

QWER ASD	TYU FGH	IOP JKL
ZX	CVB	NM

QWER ASD	TY FGH	UIOP JKL
ZX	CVB	NM

QWE ASD	RTYU FGH	IOP JKL
ZX	CVB	NM

QWE ASD	RTY FGH	UIOP JKL
ZX	CVB	NM

QWE ASD	RTYU FGHJ	IOP KL
Z	XCVBN	M

QWER AS	TY DFGHJ	UIOP KL
Z	XCVBN	M

QWE AS	RTYU DFGHJ	IOP KL
Z	XCVBN	M

QWE AS	RTY DFGHJ	UIOP KL
Z	XCVBN	M

QWER AS	TYU DFGH	IOP JKL
Z	XCVB	NM

QWER AS	TY DFGH	UIOP JKL
Z	XCVB	NM

QWE AS	RTYU DFGH	IOP JKL
Z	XCVB	NM

QWE AS	RTY DFGH	UIOP JKL
Z	XCVB	NM

QWER	TYU	IOP
ASD ZX	FGHJ CVBN	KL M

QWER	TY	UIOP
ASD ZX	FGHJ CVBN	KL M

QWE	RTYU	IOP
ASD ZX	FGHJ CVBN	KL M

QWE	RTY	UIOP
ASD ZX	FGHJ CVBN	KL M

QWER	TYU	IOP
ASD ZX	FGH CVB	JKL NM

QWER	TY	UIOP
ASD ZX	FGH CVB	JKL NM

QWE	RTYU	IOP
ASD ZX	FGH CVB	JKL NM

QWE	RTY	UIOP
ASD ZX	FGH CVB	JKL NM

QWER	TYU	IOP
AS Z	DFGHJ XCVBN	KL M

QWER	TY	UIOP
AS Z	DFGHJ XCVBN	KL M

QWE	RTYU	IOP
AS Z	DFGHJ XCVBN	KL M

QWE	RTY	UIOP
AS Z	DFGHJ XCVBN	KL M

QWER	TYU	IOP
AS Z	DFGH XCVB	JKL NM

QWER	TYU	IOP
AS Z	DFGH XCVB	JKL NM

QWE	RTYU	IOP
AS Z	DFGH XCVB	JKL NM

QWE	RTY	UIOP
AS Z	DFGH XCVB	JKL NM

QWERT	YUIOP
ASDF	GHJKL
ZXC	VBNM

QWERT	YUIOP
ASDFG	HJKL
ZXCV	BNM