BiTipText: Bimanual Eyes-Free Text Entry on a Fingertip Keyboard

Zheer Xu[1], Weihao Chen[1,2], Dongyang Zhao[1,3], Jiehui Luo[1], Te-Yen Wu[1], Jun Gong[1], Sicheng Yin[1,2], Jialun Zhai[1,4], Xing-Dong Yang[1]

Dartmouth College[1], Tsinghua University[2], Fudan University[3],
University of Science and Technology of China[4]
{zheer.xu.gr, jiehui.luo.ug, jun.gong.gr, te-yen.wu.gr, xing-dong.yang}@dartmouth.edu,
{chenwh16, ysc16}@mails.tsinghua.edu.cn, dyzhao15@fudan.edu.cn, jlzhai@mail.ustc.edu.cn

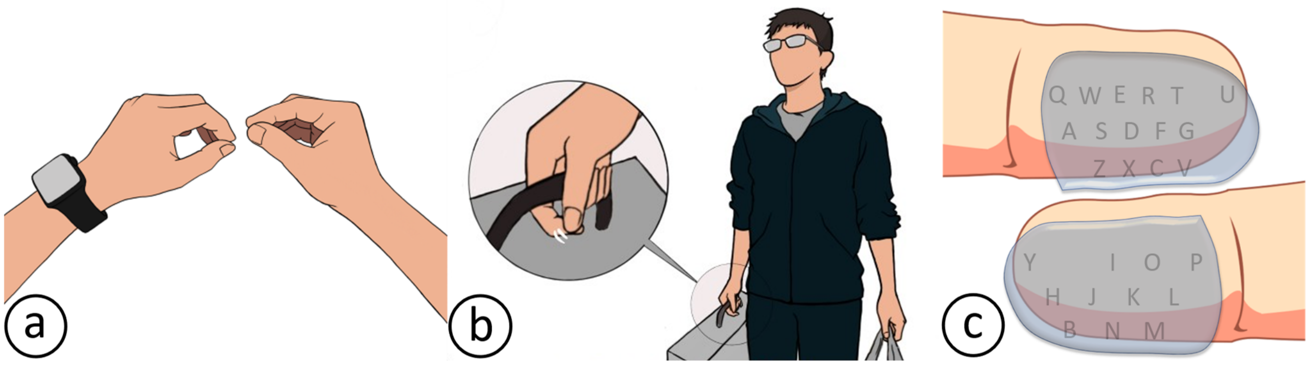

Figure 1. (a - b) Bimanul text entry using the thumb-tip tapping on a miniature keyboard residing invisibly on the first segment of an index finger (c) BiTipText keyboard layout.

ABSTRACT
We present a bimanual text input method on a miniature fingertip keyboard, that invisibly resides on the first segment of a user's index finger on both hands. Text entry can be carried out using the thumb-tip to tap the tip of the index finger. The design of our keyboard layout followed an iterative process, where we first conducted a study to understand the natural expectation of the handedness of the keys in a QWERTY layout for users. Among a choice of 67,108,864 design variations, we identified 1295 candidates offering a good satisfaction for user expectations. Based on these results, we computed an optimized bimanual keyboard layout, while considering the joint optimization problems of word ambiguity and movement time. Our user evaluation revealed that participants achieved an average text entry speed of 23.4 WPM.

Author Keywords
Micro finger gesture, text entry, wearable, bimanual input.

CSS CONCEPTS
• **Human-centered Computing~Text Input**;

CHI '20, April 25–30, 2020, Honolulu, HI, USA

ACM ISBN 978-1-4503-6708-0/20/04…$15.00
https://doi.org/10.1145/3313831.3376306

INTRODUCTION
As computing becomes ubiquitous, the need to provide users with a fast, subtle, and always-available mechanism for text entry has grown significantly. Micro thumb-tip gestures can deliver on this promise, by allowing a user to type by tapping on a miniature keyboard residing invisibly on the first segment of an index finger, using the thumb (TipText [38]). Thus, the text input can be carried out unobtrusively and even without the user looking at the keyboard (referred to as "eyes-free" in this paper). This can lead to better performance when compared with eyes-on input [44] and can also save screen real estate for devices with limited screen space. However, the existing technique is exclusively unimanual [38], despite typing often being a two-handed activity.

In this paper, we propose a keyboard design for bimanual thumb-tip text input. With two index fingers, the size of the input space doubles, thus the keys are larger and less crowded, which is helpful for reducing tapping errors. Additionally, keys residing on different index fingers use two separate input spaces, thus they can no longer be confused with each other by the system. This largely mitigates the ambiguity issue that is inevitable on a miniature fingertip keyboard. Further, the ***handedness*** of the keys on a layout (i.e. which index finger a certain key resides on) determines how typing alternates between the two hands (e.g., left → left → left → right → left). For words with a unique order of handedness, they cannot be typed incorrectly as long as the user types the words in the correct order of handedness, even if the keys are not tapped precisely. All of these benefits can make typing more accurate and faster.

Despite these benefits, identifying an optimized design for a bimanual layout for eyes-free typing is challenging for several reasons. First, it is unclear how to optimize the keyboard layout for an improved typing speed to get the most out of the key handedness unique to the bimanual text input. Second, numerous design options exist for a bimanual thumb-tip keyboard, but it is unlikely to conduct user studies to test every possible layout variation to find an optimized design. Third, the performance of the layouts may vary significantly even with small changes but there is a lack of a mechanism that can effectively measure how well a certain design may work in comparison to millions of other alternatives.

To explore the design space of this new text entry technique (which we call BiTipText), we took an iterative design approach, where we first conducted a study to understand a users' natural expectation of the handedness of the keys in a QWERTY layout. Among the choice of 67,108,864 possibilities, we identified 1295 candidates offering good satisfaction on user expectation. The results were used for our layout optimization, where we performed a stepwise search for optimized layout variations and identified one that improves the ***movement time*** (target acquisition speed) and ***word ambiguity*** (Figure 1c). Finally, we optimized this design for eyes-free input by utilizing a spatial model reflecting a users' natural spatial awareness of key locations on the tip of the index fingers. To evaluate our technique, we conducted a study with 10 participants to evaluate the speed and accuracy of BiTipText in a controlled experiment. Our results revealed that participants could achieve an average of 23.4 WPM with 0.03% uncorrected errors.

Our contribution is two-fold: an optimized keyboard layout design for bimanual thumb-tip text input and a user study demonstrating the effectiveness of BiTipText.

RELATED WORK

We summarize the literature in micro-gesture interactions, text entry on small devices, and keyboard optimization.

Micro-Gesture Input

Gesturing with the thumb-tip against the tip of the index finger offers opportunities for natural, subtle, and unobtrusive interactions in ubiquitous computing applications, such as navigation [5], triggering commands [12, 20, 32], or performing text input [38]. This style of input has become increasingly prevalent with the advance in sensing techniques [5, 12, 20, 31, 32, 34]. For example, FingerPad [5] detects 2D touch input on the tip of the index finger using a finger-worn device with electromagnetic sensing. Huang et al.'s work [17] uses a similar approach but enables one-handed and eyes-free input using a thumb-to-fingers interface. Soli [20, 32] uses an external sensor to track very small finger movements using 60 GHz radar signals. Pyro [12] detects thumb-tip gestures drawn on the tip of the index finger based on the thermal radiation emitting from the user's fingers.

Recent developments for interactive skin technologies enable a new way to sense micro gestural input [18, 21, 25, 33, 35, 36]. For example, iSkin [35] is a thin, flexible, and stretchable skin overlay, made of biocompatible materials capable of sensing touch input on the skin. The sensor can be made in different shapes and sizes and used in many different locations on the body, such as the fingertip. DuoSkin [18] is similar in that it can detect touch input on the skin using an interactive overlay made of gold leaf. SkinMarks [36] are interactive tattoos that can detect touch, squeeze, and bend on the skin. Finally, in recent work by Nittala et al. [25], high-resolution multi-touch input is now feasible on interactive skin overlays.

Text Entry on Small Devices

Text entry on small devices is challenging primarily due to the lack of input space. A large body of research has been carried out in this space to improve user experience [1, 6, 11, 13-16, 26, 29, 40]. Within the existing work, many have employed a two-step operation, requiring a keyboard to be expanded before a user can select the desired key [1, 6, 7, 16, 26, 28, 29]. For example, with Zoomboard [26], a user needs to zoom into a region containing the desired key before the key can be selected. Splitboard [16] takes a different approach by using a keyboard beyond the size of the screen of a smartwatch. When the desired key is off-the-screen, the user must scroll the keyboard to bring the key inside the view. Both DualKey [15] and ForceBoard [42] associate a keyboard key with two letters. The user must use different levels of pressure or different fingers to specify the desired letter.

Note that text input does not need to be carried out on a touchscreen. For example, Yu et al.'s work [41] allows users to type on a one-dimensional touch sensor using unistroke gestures. With WrisText [11], a user can perform text input by whirling the wrist. FingerT9 [37] maps the keys of a T9 keyboard onto the different segments of the fingers for the thumb to tap. ThumbText [19] allows a user to perform text input using a ring-sized touchpad worn on the index finger. TipText [38] features a miniature QWERTY keyboard residing invisibly on the first segment of the user's index finger. The user can type using the thumb-tip to tap the tip of the index finger. However, the technique was designed exclusively for one-thumb use, a limitation we address in this work.

Keyboard Optimization for One vs. Two-Thumb Use

While QWERTY is the standard keyboard layout for both physical and virtual keyboards, it can also be suboptimal in many scenarios, including gestural typing. Smith, et al. proposed a set of QWERTY variations to optimize the performance of the keyboard for gestural typing based on gesture clarity, gesture speed, and similarity to QWERTY

[30]. Their approach requires keys to be shifted away from their original locations, making users learn a layout before they can begin typing. To address this issue, Bi and Zhai [4] proposed IJQwerty, a layout that is different from QWERTY by one pair of keys but still capable of achieving a significant improvement on typing accuracy and speed.

The keyboard layout can also be optimized for bimanual text input. For example, Oulasvirta et al.'s work [27] demonstrated that rearranging the location of the keys in a split keyboard encouraged more alternate-side taps, thus leading to faster typing speed. Bi, et al. developed a method for gestural typing to be performed using two thumbs. They found bimanual typing was more comfortable and less physically demanding [2]. Recent work from Lu et al. allows bimanual text entry to be performed on a tablet using the user's peripheral vision [22]. This way, the user's primary attention can be focused on the output text.

DESIGN CONSIDERATIONS

We considered several factors for a bimanual thumb-tip keyboard.

Eyes-Free Input

Similar to TipText, a user types using BiTipText without looking at their finger movement or the keyboard (or "eyes-free"). As the user's input space is different from the output space, their imaginary location of the desired key, based on their spatial awareness, can be different from the actual location of the key. Certain levels of tapping errors can be tolerated using a statistical decoder [13, 43], which considers the models of input language and people's natural spatial awareness of the location of the desired key.

Learnability

Learning needs to be reduced, especially when the keyboard layout is invisible when typing. Our layout is based on QWERTY, as it is relatively easy to learn due to its widespread adoption. Note that for bimanual input, keys need to be distributed across two index fingers, which creates millions of possibilities for layout designs. To facilitate learning, the spatial relationship of the letters needs to remain the same as QWERTY. Thus, keys remain in their original locations in relation to their neighbors. This ensures that users can rely on their existing knowledge of a smartphone keyboard layout without the need to learn new letter locations. In the example shown in Figure 2, our design requires the center of "E" to remain to the north-east of the center of "S" (Figure 2a) rather than shifting to the north-west (Figure 2b) even when space is available on the right side of "S". This is to ensure strings like "SE" can be typed based on a user' existing spatial awareness of key locations. Although keys near each other can be hard to select, we saw it less of an issue as a statistical decoder can likely be helpful in this case. We used this spatial relationship to infer the key center in the estimation of word ambiguity and movement time later in our optimization. Finally, in our design, keys are not duplicated on both sides.

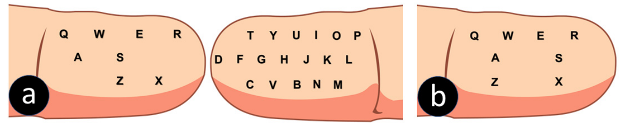

Figure 2. (a) Example of a possible bimanual keyboard layout (b) an alternative design of left keyboard which divides the space evenly for large keys.

Typing Speed

In the bimanual condition, typing speed is affected by many factors. For example, the task distribution across the dominant vs. nondominant hand may affect the speed. Some layouts may be more optimal for word ambiguity while others may be more optimal for movement time. The mixture of different factors makes it difficult to find an effective strategy for keyboard optimization. We used a program to estimate how fast users can type for all the candidate layouts using a quantitative measurement of movement time and word ambiguity.

Handedness Expectation

People's natural expectation of the handedness of a layout may follow their bimanual typing experience on a soft or hardware keyboard. As such, an intuitive solution for our work can be a split keyboard with the QWERTY layout divided from somewhere in the middle, like those found on tablets. However, it is unclear if such a layout is optimal for typing performance. While some layouts may not fully satisfy user expectation, the tradeoff can be improved typing efficiency. Thus, a balance can be drawn. Our strategy was to iterate all the possible layout variations and find the ones promising relatively good user satisfaction as input for our layout optimization.

BITIPTEXT

By considering these factors, we designed our thumb-tip text entry technique as a bimanual version of TipText [38], with new features designed exclusively for bimanual text input. The keyboard involves keys distributed across the two index fingers with a layout optimized for typing speed, without significantly impacting usability. This allows the users to type fast and efficiently. When typing with BiTipText in an eyes-free context, a user selects the keys based on their natural spatial awareness of the location of the keys on the two index fingers. The system generates a list of candidate words in response to the selected keys, ranked based on the probability calculated using a statistical decoder. The user swipes the thumb right to enter the selection mode, which highlights the top-ranked candidate. The user can swipe right again to navigate to the next candidate. The word will be committed automatically upon the user typing the next word (e.g. tapping the first letter of the next word). Space will be inserted automatically after the committed word. The user can swipe left to delete the last letter. Auto-complete was implemented by following the algorithm described in [40].

BiTipText Hardware

We developed an interactive skin overlay, similar to the one presented in TipText [38]. Our sensor measures ~2.2cm × 2.2cm and contains a printed 3×3 capacitive touch sensor matrix with diamond-shaped electrodes of 5 mm diameter and 6.5mm center-to-center spacing. Our prototype was developed using a flexible printed circuit (FPC). The sensor is 0.025 – 0.125 mm thick and 21.5mm × 27mm wide. The sensor was controlled using an Arduino microcontroller with a MPR121 touch sensing chip. The raw capacitive data from each channel was transmitted at a frequency of 100Hz. Software that interpolates the touch events was implemented in C#.

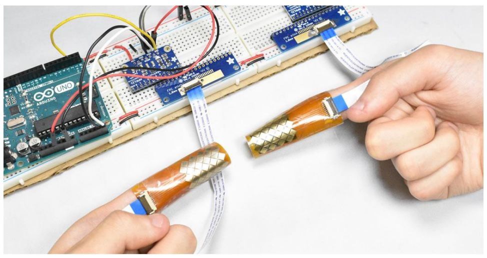

Figure 3. BiTipText prototype, composed of a capacitive sensor developed using FPC.

STUDY 1: HANDEDNESS EXPECTATION

To begin our research, we sought to understand users' natural expectation of the handedness of the 26 letters, which we described as the percentage of times that a letter is selected using the left or right hand when typing bimanually on index fingers. We used the study result to identify a set of candidate layouts that offer a good match between the handedness of the layout and user expectation.

Participants and Apparatus

Ten participants (3 females) aged from 20 to 25 participated in the study. All of them were familiar with the QWERTY layout on a smartphone keyboard. Participants wore the BiTipText hardware on the first segment of the index finger on both hands. A 24-inch monitor was used to show the test phrases and user input. Participants sat in a chair with their hands placed below their sight.

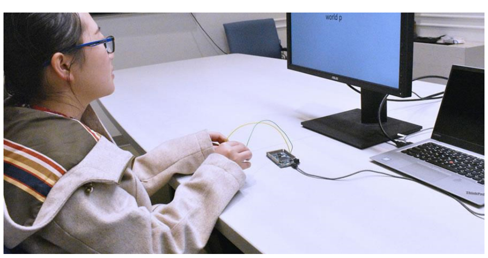

Figure 4. Setup of Study 1.

Task and Procedure

Participants were asked to type 8 blocks of 10 test phrases, among which, 64 were randomly picked from MacKenzie's phrase set [24]. The rest were randomly selected pangrams, presented as every fifth phrase. Like previous work [9, 10], we used the pangrams to ensure that every letter had a minimum presence of 30 times each. Participants were informed that the keys in the QWERTY layout were distributed across the two index fingers in any way they liked, and that the system was smart enough to determine what letter they wanted to type, upon a touch. There was no restriction on which hand to use for a certain letter. The same letter was allowed to be typed using different hands throughout the study. Participants were asked to type as fast and naturally as possible. Our system always displayed the correct letters no matter where they touched on the sensor. Participants understood that both hands needed to be involved in the study, thus none completed the task using only one hand.

Result

In total, we collected 20490 data points (2049 letters in the 80 test phrases × 10 participants) to analyze participants' handedness expectation. Figure 5 shows the percentage of each letter typed using the left hand (reverse for the right hand). Our result shows that even though participants were free to use either hand for any letter, there is a clear consistency in their expectation about the handedness of each letter. Participants tended to type the keys on the left side of the QWERTY layout using their left hand, and vice versa. In contrast, there was no clear consistency in the handedness of the keys residing in the middle of the keyboard (e.g., "G", "V", "B", etc.). As expected, the observed behaviors are likely due to the users' bimanual typing experience on existing soft or physical keyboards. There were instances where a sided letter was typed using the opposite-side finger. For example, 11 out of 300 "Q"s were typed using the right index finger, among which, 82% were typed after two or more successive taps on the left index finger (e.g., "whizzed quickly"). This suggests that long successive taps on one side may introduce side errors.

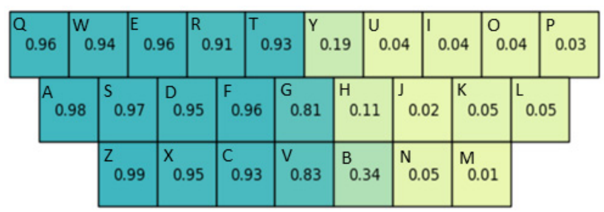

Figure 5. Users' natural expectation of the handedness of the 26 letters, described using the percentage of each letter typed using the left hand (reverse for the right hand).

FILTERING OUT LAYOUTS WITH LOW USER SATISFACTION ON KEY HANDEDNESS

Using the data from Study 1, we were able to gauge how well a given layout satisfies the user's expectation. Depending on the handedness of each key in that layout, we calculated a *Satisfaction Score* for each word. In particular, we simulated the handedness sequence of a word for 10 times using the percentage data shown in Figure 5. The satisfaction score of a word was calculated as the probability of a sequence matched exactly the handedness distribution of the layout. The satisfaction score of that layout is thus the sum of the scores of all the words in the top 3500 words in American National Corpus (80% of word usage) weighted by their frequency. The higher the better. We iterated and scored all 67,108,864 (2^{26}) layout variations (each key can be placed either on the left or right index finger) using a computer program. The overall score for each layout was averaged among participants. The layouts were then sorted in a descending order based on satisfaction score. We used the score of the top ranked layout as a reference and chose the candidates scoring 85% and higher for our optimization step. This included 1295 variations. The top-ranked layout is the one commonly seen on touchscreen devices, which separates the keyboard in the middle with more keys on the left hand (Figure 6a). This is consistent with our finding in Study 1. The ones ranking near the bottom of the list followed a similar pattern, except that the letters with a low frequency in the corpus (e.g., “U”, “P”, “J”, “K”, or “B”) were placed on the opposite-side hand. This makes sense as the penalty for violating user expectation is low on the low-frequency letters.

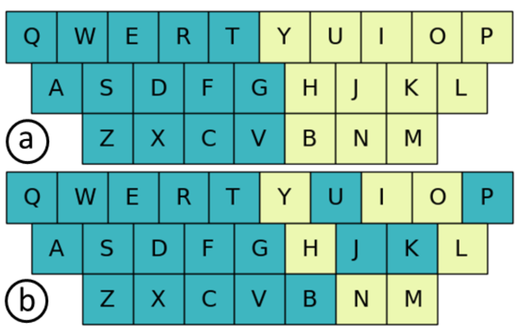

Figure 6. An illustration of (a) top and (b) bottom-ranked layouts in our candidate list. Keys in cyan are on the left hand. Keys in yellow are on the right hand.

GENERAL OPTIMIZATION APPROACH

With the list of candidates for our optimization, we sought to find those that can perform well for typing speed. Here we present our optimization approach inspired by Smith et al.’s work [30].

Optimization Metrics

Designing the layout for a bimanual fingertip keyboard can be framed as an objective optimization problem, where the objectives are to improve (1) word ambiguity and (2) movement time.

Word Ambiguity. Word ambiguity is a measurement describing how much a word is likely to confuse with other words in a layout. On a miniature fingertip keyboard, tapping errors can hardly be avoided because the keys are too small to type. As such, a series of key entries may map to a set of neighboring keys and thus different words. Therefore, a user needs to spend time on searching and navigating a list of candidate words for the target word.

Movement Time. Movement time is related to the time needed to type the desired keys. Different layouts may permit different movement time for typing due to the handedness and density of the keys.

Optimization Procedure

To maximize the objectives based on the metrics, a stepwise search can be employed by iterating all the possible layout variations based on these metrics and their weights. We explain our optimization procedure here.

Calculating Word Ambiguity Score

The word ambiguity of a layout can be measured based on how likely words are confused with each other by the statistical decoder. To gauge ambiguity, we define the likelihood of the target word A to be misinterpreted as another word B using a *Confusion Score* between 0 and 1:

$$Confusion(A,B) = \begin{cases} \min\{\frac{P(B|S)}{P(A|S)}, 1\}, & same\ OH \\ 0, & different\ OH \end{cases} \quad (1)$$

where $S = \{s_1, s_2, s_3, \ldots, s_n\}$ denotes a set of touch points intended to type the word A, $P(A|S)$ and $P(B|S)$ are the probability of A and B calculated by the statistical decoder using S (see equation details in TipText [38]). In the case when A and B are typed in the same order of handedness (OH), the closer $P(A|S)$ and $P(B|S)$ are, the more likely the word B will be ranked higher than A in the candidate list. This will slow down the user. If A and B have different OH, they will not be confused by the system as long as the user types them in the correct order of handedness, even if the keys are not selected correctly. With Eq. (1), we can define the *Ambiguity Score* of a word W from the lexicon L as the highest pair-wise confusion score W can get:

$$Ambiguity(W) = \max_{X \in L} Confusion(W, X) \quad (2)$$

Note that the challenge to calculate the ambiguity score is in the lack of the spatial model needed for Eq. (1). Although the model can be acquired through a user study, conducting a study for every candidate layout is not feasible for an initial investigation.

We took a simplified approach similar to the one used in previous work for word clarity on a soft keyboard [39], where x and y coordinates were assumed to be independent to each other, s_i and the center of the touch point distribution were also assumed to be in the center of the target keys, aligned in a grid layout. Furthermore, the standard deviation of the touch point distribution for keys in different sizes and densities are treated as a constant σ based on the result of Study 2 (details later). This way, $P(S|W)$ can be simplified as:

$$P(S|W) = \left(\frac{1}{2\pi\sigma^2}\right)^n \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\|w_i - s_i\|_2^2\right) \quad (3)$$

Therefore, Eq. (1) can be described as the following:

$$Confusion(A,B) = \begin{cases} \min\left\{\frac{\exp(-dis(A,B)) \times P(B)}{P(A)}, 1\right\}, & same\ HS \\ 0, & different\ HS \end{cases} \quad (4)$$

Following Yi et al.'s work [39], $dis(A,B)$ is defined as:

$$dis(A,B) = \frac{1}{S_{key}^2} \times \sum_{i=1}^{n} \|a_i - b_i\|_2^2 \qquad (5)$$

where, a_i and b_i are the 2D coordinates of the key center of the ith character in word A and B respectively, S_{key}^2 refers to the area of the key, and $\|\cdot\|_2^2$ denotes the Euclidean norm. Note that S_{key}^2 was used to normalize the result based on the size of the first segment of a participant's index finger (measured manually for each participant).

With all the ambiguity scores available for the top 3500 words in American National Corpus, the ambiguity score of a layout is the sum of the scores weighted by their frequency. The lower the score, the better the layout is.

Calculating Movement Time Score

Average movement time needed to type on a layout can be measured using a user study but conducting studies for 1295 candidates is not feasible. Instead, we used an alternative approach similar to Oulasvirta et al.'s work [27], where we estimated the average movement time of a layout using a predictive model developed for bimanual thumb-tip input. Our model was developed through a user study by considering two types of tapping tasks unique in the bimanual condition: *same-side taps* and *alternate-side taps*.

Same-side taps are sequential key entries on one side. We assumed the movement time for same-side taps followed the polynomial model of the same type of task developed on a bimanual tablet keyboard [27]. Thus, we model the movement time using a polynomial with *ID* as the predictive variable, where *ID* is $\left(\frac{D}{W} + 1\right)$, D is the key distance, and W is key width.

Alternate-side taps are consecutive taps on different sides. With alternate-side taps, the thumb that is not typing (i.e. *idle thumb*) is assumed to be approaching its next target immediately after the other thumb (*active thumb*) begins typing. This concurrent action is to increase typing speed. The movement time for alternate-side taps is affected by two variables, *ID* and the time elapsed ($t_{elapsed}$) for the active thumb to finish typing. For example, the movement time can be as short as $t_{elapsed}$ if $t_{elapsed}$ is long enough for the idle thumb to complete its journey to the target. In this situation, what remains is the short time needed for the idle thumb to tap the target when its turn comes. The movement time can be longer than $t_{elapsed}$ if $t_{elapsed}$ is shorter than the time needed for the idle thumb to reach its target. We assumed the movement time for alternate-side taps followed the bivariate quadratic model of the same type of task developed on a bimanual tablet keyboard [27]. Thus, we modeled time using a bivariate quadratic function with $t_{elapsed}$ and ID as the predictive variables.

With the predictive models, we were able to estimate average movement time needed to type all the words in the top 3500 words in American National Corpus for a given layout design. The movement time score for a layout is the sum of the movement time of all the words weighted by their frequency.

Metric Normalization

Scores for word ambiguity and movement time need to be normalized in order for them to be appropriately weighted. We normalized them in a linear fashion, such that the lowest and highest scores are mapped to 0 and 1 respectively.

Weight Iteration

Users may weigh word ambiguity and movement time (MT) differently, depending on usage scenarios. This may impact the layout of the keyboard. The last step is thus to iterate all the possible weight combinations (e.g., using a step size of 0.01) and identify an optimized layout for each combination by maximizing the following objective function:

$$\mathrm{F}(Amiguity, MT) = \alpha \times Ambiguity + \beta \times MT \qquad (6)$$

where α and β are the weight of ambiguity and MT, and $\alpha + \beta = 1$.

STUDY 2: MODELING TAPPING SPEED

The goal of this study was to acquire the data needed for creating a predictive model to estimate the tapping speed for the eyes-free thumb-tip tapping task.

Participants and Apparatus

Twelve right-handed participants (4 females) aged from 20 to 25 participated in the study. We used a Vicon Motion Tracking System for finger tracking and the Unity game engine for the estimation of touch location (same as in TipText [38]). Participants wore markers on the nail of the thumbs and index fingers for the Vicon to track finger movement (Figure 7). Vicon data was used to control the movement of the fingers' virtual counterparts in Unity. The virtual fingers were 3D meshes, obtained by scanning clay models of each participant's fingers. Touch location was estimated based on the collision region of the virtual fingers and was updated at 200 fps. A monitor was placed in front of participants to provide instructions.

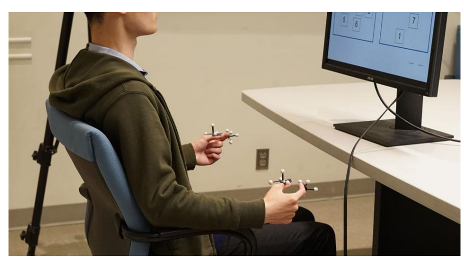

Figure 7. Setup of Study 2.

Task and Procedure

We used the N-return task from Oulasvirta et al.'s work [27]. The task began with a target on an index finger, followed by N targets on the opposite-side index finger before returning to the initial finger for the last target. A rectangular region was shown to the participants to represent the input area of each index finger (Figure 8). The ratio of the rectangle was adjusted for each participant and used for both index fingers (neglecting the small difference between them). All square targets were presented to a participant before a trial started. Knowing the location and order of the targets ahead of time minimized the time they spent to search for the next target. Participants were encouraged to start approaching the last target as soon as the opposite-side thumb started tapping the N targets. They selected each target based on their natural spatial awareness of the target location on the index finger. They were not informed about whether a touch was inside the target. A trial was considered successful if participants believed so.

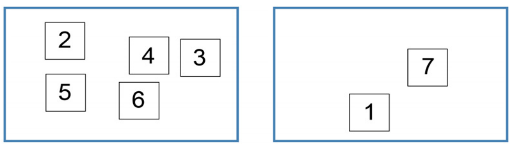

Figure 8. User Interface of Study 2.

Experimental Design

Our study included three target widths, ranging from one-tenth of width of the input area (key size of a full QWERTY layout) to half of the height of it (key size of TipText [38]). The level in the middle is the average of them. We ran two sessions of N-return tasks for each target width. One session started from the left hand and another one from the right. The sessions were counterbalanced among participants. N was chosen to be between 1 to 5 with a step of 2, resulting in 3, 5, or 7 targets across the two index fingers. For each N, we picked 5 sequences of different targets with locations generated randomly without the target overlapping with each other. The experimental design can be summarized as: 3 *Target Width* × 2 *Starting Hand* × 15 *Targets* × 5 *Sequences* × 10 *Repetitions* × 12 *Participants* = 54000 data points in total.

Inferring Target Location and Size

Eyes-free typing is unique in that target location and width is determined by the users' natural spatial awareness of the keys, not those of the target shown visually to participants. Therefore, the predictive model for the movement time should be derived using the distance and width of the imaginary targets, determined by the distribution of the touch points of the targets. Similar to TipText [38], we assumed that participants' touch locations followed a bivariate Gaussian Distribution [3]. Therefore, the center of the imaginary target is the center of the distribution of the touch points, which was used to calculate the distance between the imaginary targets (*D*). The width and height of the imaginary target were calculated using the two standard deviations of the touch points along the x- and y-axis. We chose shorter sides as *W* to calculate the *ID* [23].

Movement Time

The movement time was calculated separately for the same- and alternate-side taps [27]. For the same-side taps, the movement time from a target to the next one was defined as the time from the moment when the thumb left the current target to the moment when the thumb touched the next target (like how soft keyboards work). For the alternate-side taps, we used a trial starting on the left hand as an example. We use t_n and t_{n-1} to denote the last and second last target in a trial. The movement time was defined as the moment when the right thumb touched target t_{n-1} to the moment when the left thumb touched the last target (t_n). The $t_{elapsed}$ from the first target t_1 to the last one t_n was defined as the period between the moment when the left thumb left t_1 (it becomes idle) to the moment when the right thumb touches t_{n-1} (the left thumb can tap again).

Modeling

For modeling, we used 15 *ID* conditions and three $t_{elapsed}$ conditions and only adopted those with samples from more than two participants. For the same side taps, movement time was modeled for the left (R^2=0.82) and right hand (R^2=0.99) respectively. The models are shown below:

$$MT_{left} = 28.7ID^2 + 63.9ID + 110.7 \quad (7)$$

$$MT_{right} = 66.8ID^2 + 120.8ID + 60.8 \quad (8)$$

We also obtained the models for the alternate-side taps for the operating hand switching from right to left ($MT_{right \to left}$; $R^2 = 0.77$) and left to right ($MT_{left \to right}$; $R^2 = 0.92$):

$$MT_{right \to left} = 41.2ID^2 + 101.2ID - 30.98t_{elapsed}^2 + 57.8t_{elapsed} + 47.1ID \times t_{elapsed} + 182.5 \quad (9)$$

$$MT_{left \to right} = 31.8ID^2 + 45.5ID + 126.8t_{elapsed}^2 - 187.6t_{elapsed} + 9.6ID \times t_{elapsed} + 301.5 \quad (10)$$

Discussion

The predictive models fit the data reasonably well, considering that the data for eyes-free target acquisition can be noisy. For the same-side taps, the model fits better for the right (dominant) hand than the left hand. Similarly, the model fits better for $MT_{left \to right}$ than $MT_{right \to left}$. This is an indication of a lack of control on the left hand or perhaps a better fit can be achieved using a different model. A surprising finding is that same-side taps were faster than alternate-side taps. This is different from the results from Oulasvirta et al.'s work conducted on a tablet [27]. We believe that it is because the movement time on a small fingertip is not long enough for the opposite-side finger to take advantage of simultaneous movement. Switching hands introduced cognitive overhead which affected task performance. As such, layouts that are optimized for movement time should avoid alternate-side taps, at least between the letters with high frequency. This is reflected in our optimization process. Finally, our finding suggested that the standard deviation of the touch points distribution in the x and y coordinate (i.e., σ_x, σ_y) were not significantly affected by the size of the targets (σ_x: $F_{2, 22} = 0.622$, $p = 0.546$; σ_y: $F_{2, 22} = 0.314$, $p = 0.733$). We believe this is because none of the targets are big enough for the user to acquire precisely in the eyes-free condition.

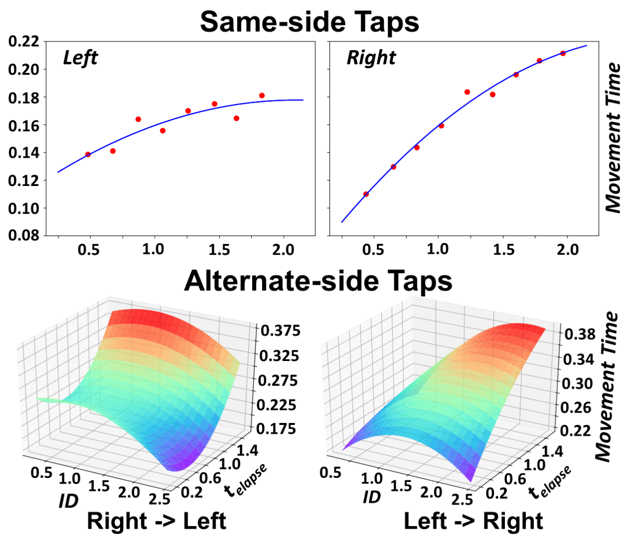

Figure 9. Predictive models of same-side taps and alternate-side taps.

BITIPTEXT KEYBOARD LAYOUT

We used the aforementioned approach to optimize our keyboard layout, where we calculated the word ambiguity score and movement time score for each candidate layout. For each of the 101 different weight combinations, we iterated all 1295 layout candidates, and identified the one with the highest score from Function (6) as the optimal layout. Amongst all the 101 optimal layouts, three are different, suggesting that only a small set of the layouts performed well on our metrics. We plot all the candidate layouts in Figure 10, in which the red dot represents the one that balanced both metrics (Figure 11c), while the orange dots represent the layouts that are optimal for a single metric (e.g., scored highest for word ambiguity or movement time) (Figure 11a-b).

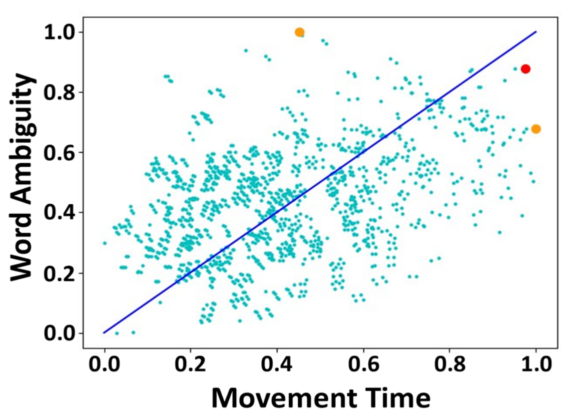

Figure 10. Candidate layouts shown by movement time score and word ambiguity score (orange dots: optimal layouts for single metric; red dot: final layout of BiTipText).

Since each layout represents a compromise between the two metrics, the choice of layout depends on the importance of each metric for a user. For example, for a fast but less accurate user, a layout scored higher on word ambiguity may be preferred, while for a slow but more accurate user, a layout scored higher on movement time may be preferred. We chose the design that balances both metrics. It has a word ambiguity score of 0.88 and movement time score of 0.98. It is similar to the top-ranked layout in handedness expectation (Figure 6a) but differs only in the handedness of two letters. "U" and "B" are now placed on the left index finger. In comparison to that layout, our final design exceeds in the word ambiguity score for more than 35%.

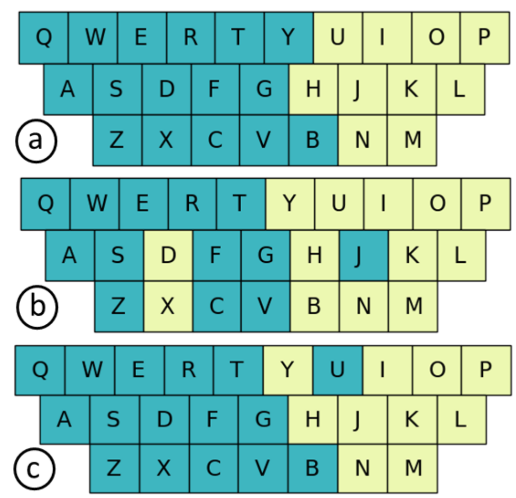

Figure 11. (a) The layout optimized for movement time; (b) the layout optimized for word ambiguity; (c) The layout of BiTipText, which balances both movement time and word ambiguity.

USER STUDY 3: PERFORMANCE EVALUATION

We conducted a user study to evaluate the performance of BiTipText. The evaluation was divided into two sessions. In the first session, we developed a statistical decoder based on the users' eyes-free typing data using our final keyboard layout. The second session took place the next day, where we measured how well users can perform the text entry using our technique.

Participants

Ten participants (4 females) aged between 22 to 25 participated in our study. All the participants were right-handed and were familiar with the QWERTY layout.

Apparatus

The apparatus was the same as in Study 1. The sensors were calibrated for each participant prior to each step.

Session 1: Spatial Model

Task and Procedure

The task and procedure were similar to those in Study 1. Participants were asked to transcribe blocks of 80 different test phrases. The pangrams were used to ensure that every letter had a minimum presence of 30 times each. The same set of 80 phrases was used for all participants. Upon a touch occurring on the correct hand, the system recorded the touch location and displayed the correct letters no matter where they touched on the index finger. A static image of our keyboard layout was shown on the monitor to remind participants about the positions of keys. After entering a phrase, the experimenter pressed a "Done" button to allow the participant to proceed to the next phrase. This process was repeated until they completed all phrases. Before the study, participants were given 10 minutes to familiarize with the system without practicing locations of keys. This session lasted around 30 minutes. We collected 19930 sample points to build the spatial model.

Result
Touch points recorded in Session 1 were normalized to obtain a general distribution. Figure 12 shows all touch points collected from 10 participants. The touch points for each key are shown in different letters. As in the unimanual condition [38], we assumed that touch points for different keys followed the bivariate Gaussian Distribution. Thus, the corresponding letters are shown at the centroids of the touch points along with a 95% confidence ellipse. Touch locations are noisy with considerable overlaps among different ellipses, but it is still observable that the centroids of each letter follow the spatial relationship of a QWERTY layout. While some letters can be hardly separated from each other, this noisy input can likely be tolerated by the statistical decoder. Using the collected data, we derived a general spatial model for our keyboard layout. It was used in the second session of the study.

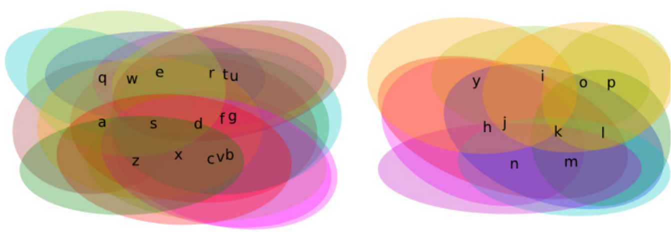

Figure 12. Touchpoints distribution within 95% confidence ellipses of all 26 keys across the left and right hands.

Session 2

Task and Procedure
Participants came back in the next day for the second session of the study. The task and procedure were similar except that: (1) we used a new set of 40 phrases, randomly chosen from MacKenzie's test set, and divided into 4 blocks; (2) the static image of our keyboard was removed from the display; (3) the actual letters entered by the participants were shown instead of the correct ones; and (4) the top three candidate words provided by the statistical decoder were shown for the participants to choose from. Participants can use a right swipe gesture to navigate through candidate list and left swipe to delete the last inputted letter (when typing) or navigate back in the list (when selecting word). Prior to the experiment, participants could practice with the layout visible for as long as they wanted. The second session lasted around 60 minutes. In total, we collected 400 phrases (10 participants × 4 blocks × 10 phrases) in the study.

Result
All data obtained in Session 2 was analyzed by using a one-way repeated measures ANOVA and Bonferroni corrections for pair-wise comparisons. We used a Greenhouse-Geisser adjustment for degrees of freedom.

Text-Entry Speed. The average text entry speed was 23.4 WPM (s.e. = 2.0), indicating the effectiveness of bimanual input on a miniature fingertip keyboard. ANOVA yielded a significant effect of Block ($F_{1.58,\ 14.11} = 10.25$, $p < 0.05$, $\eta_p^2=0.53$). Post-hoc pair-wise comparisons showed a significant difference between the first and second block ($p < 0.0083 = 0.05 / 6$; 6 is the number of pairs). Participants achieved 20 WPM (s.e. = 2) in the first block and the speed increased to 25 WPM (s.e. = 5.5) in the last block (Figure 13). This is exciting as the result suggested that with practices, participants were able to achieve a typing speed comparable to some of the existing techniques on larger devices, such as a smartwatch (e.g., 24 WPM for WatchWriter [13]).

Error Rate. We report uncorrected error rate (UER) and total error rate (TER). Uncorrected errors were the errors detected in the final transcribed sentence, while the total errors included both corrected and uncorrected error. Overall, the average TER was 2.12% (s.e. = 1.02%). The average UER was 0.03% (s.e. = 0.05%). Both types of errors appeared to be low, suggesting that bimanual text input can effectively reduce the occurrence of typing errors.

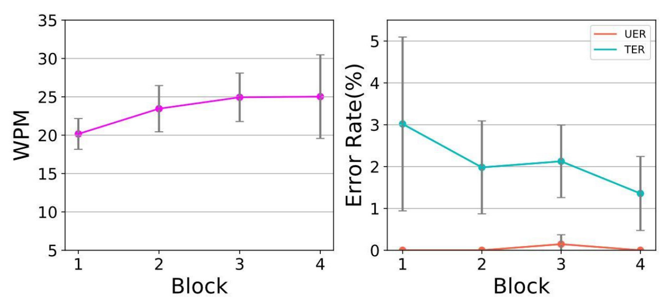

Figure 13. Left: text entry speed; Right: average UER and TER over four blocks.

Auto-Complete Rate. We calculated auto-complete rates of a word using the ratio between the number of characters automatically completed by the system and the total number of characters in the word. The overall auto-complete rate was 5.8% (s.e. = 2.57%) for all the tested words. The average text entry speed with no auto-complete is 23.4 × (1 – 5.8%) = 22.0 WPM. A marginally significant difference ($F_{1.93,\ 17.4} = 3.8$, $p = 0.04$) between the auto-complete rate over blocks suggests that the amount of use of auto-complete did not change significantly with the increased familiarity with the keyboard layout.

Word Ambiguity. We also report the percentage of time the target words appearing at the top of the candidate list. We only used the trials completed without using auto-completion. Amongst all the 1864 words typed in full length, 1477 of them were ranked at Top 1 (79.24%) by the statistical decoder. This is exciting, which explains why the typing speed was fast and again shows the effectiveness of our optimization on word ambiguity.

DISCUSSION, LIMITATIONS AND FUTURE WORK

In this section, we discuss our observations and thoughts collected during the course of our work. We also present the limitations and plan for future research.

Typing Speed. Optimizing the movement time and word clarity improves the typing speed of the bimanual text input. As found in Study 2, the alternative-side taps had made it slower for the users to acquire the targets. Our optimizing approach took this into account by favoring the layouts that permit less alternative-side taps. Additionally, the improvement of word ambiguity had a significant impact on typing speed. This is primarily due to the time saved for the participants to navigate the candidate list, as the system is relatively precise in predicting a users' target word. Unlike TipText [38], BiTipText does not need the letters to be grouped into larger keys. This effectively mitigates ambiguity issue as letters located on different hands can no longer be confused by the statistical decoder. These facts are unique to the fingertip keyboard, as they may not have as much impact on a larger touchscreen device since the overlap of the touch point distributions is far less severe on bigger devices. For example, on a tablet, "S" can hardly be confused with "H" by a statistical decoder due to the long distance between them, but they can be easily confused by a system on a fingertip keyboard. With BiTipText, there are two spatial models, one on each index finger, with no overlap with each other. This has significantly improved the input clarity and thus typing speed. As for future work, we will compare the performance of BiTipText with TipText through a controlled experiment to verify the speed improvement introduced by these advantages.

Predictive Model of Movement Time in the Eyes-free Condition. Our predictive model was developed with an assumption that the index of difficulty follows the same expression as in the Fitts' Law [8]. However, our results showed that aside from the one for the right hand, the models do not show a strong fit into the users' behavioral data. This indicates that parameters can be different for the *ID* for the thumb-tip tapping in the eyes-free condition. We foresee that a better fit can be achieved with a different model or a more precise description of the index of difficulty. Additionally, the ordered target acquisition task may not perfectly represent typing in English, which also leaves spaces for improvements in prediction accuracy. These open fruitful directions for future research.

Statistical Decoding. Our current statistical decoder provided three candidate suggestions based on a user's touch input. Since 79.24% of a target word appeared as the Top 1 in the candidate list, we expect that showing fewer candidates may also work in our technique. This allows screen real estate to be further saved. We plan to investigate the optimal number of suggestions that should be shown on the display. Additionally, a more powerful statistical decoder can be developed by considering auto-correction for error types that are not considered in the current implementation, such as insertion or omission errors. Further, efforts will be made to develop an adaptive spatial model that can achieve better performance for individual user. Note that a statistical decoder cannot handle words outside the dictionary. Therefore, users need to be allowed to add new words to the dictionary. In such a situation, a keyboard can be shown to the user for improved precision on tapping.

Bimanual Input. While our new technique is able to achieve a relatively fast typing speed on a miniature fingertip keyboard, using both hands may not be preferable in some situations, where the user's hand is busy at other tasks (e.g., holding objects). Additionally, typing on a split keyboard may lead to an increase in cognitive load due to the additional requirements for the users to type each letter on the correct hand. Future research will investigate the cognitive overhead and identify ways to mitigate it.

User Study. Our study was conducted in a controlled lab environment with the tasks performed by relatively a small number of participants in a sitting position. We plan for future work to investigate the performance and usability of BiTipText with more participants in more realistic scenarios, such as standing or walking with the hands hanging alongside the body. Other scenarios, such as typing while the hands are holding an object are also interesting and warrants a careful investigation.

CONCLUSION

We proposed a thumb-tip text input method designed and optimized for bimanual input. With our technique, the text input can be performed using both hands by tapping the thumbs on a split fingertip keyboard, residing invisibly on the first segment of a user's two index fingers. Using an iterative and user-centered design process, we sought an understanding of users' natural expectation of the handedness of the keys in a QWERTY layout. Using this knowledge, we identified 1295 layouts that can offer good satisfaction for user expectation from a total number of 67,108,864 candidates. Based on the results of these studies, we computed an optimized bimanual keyboard layout that improves both movement time and word ambiguity. We evaluated our text entry method in a controlled lab study, where we found that our method allowed participant to perform text input at an average speed of 23.4 WPM. With the findings of this work, we see a great potential of the proposed text input method in mobile, wearable, and VR/AR applications.

REFERENCES

[1] 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

[2] Xiaojun Bi, Ciprian Chelba, Tom Ouyang, Kurt Partridge and Shumin Zhai. 2012. Bimanual gesture keyboard. In *Proceedings of the 25th annual ACM symposium on User interface software and technology* (UIST '12), 137-146. DOI=https://doi.org/10.1145/2380116.2380136

[3] 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

[4] Xiaojun Bi and Shumin Zhai. 2016. IJQwerty: What Difference Does One Key Change Make? Gesture Typing Keyboard Optimization Bounded by One Key Position Change from Qwerty. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (CHI '16), 49-58. DOI=https://doi.org/10.1145/2858036.2858421

[5] 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

[6] 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

[7] 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

[8] Paul M. Fitts. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology Fitts*. 47 (6), 381. DOI=https://doi.org/10.1037/h0055392

[9] Mayank Goel, Leah Findlater and Jacob Wobbrock. 2012. WalkType: using accelerometer data to accomodate situational impairments in mobile touch screen text entry. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '12), 2687-2696. DOI=https://doi.org/10.1145/2207676.2208662

[10] Mayank Goel, Alex Jansen, Travis Mandel, Shwetak N Patel and Jacob O Wobbrock. 2013. ContextType: using hand posture information to improve mobile touch screen text entry. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '13), 2795-2798. DOI=https://doi.org/10.1145/2470654.2481386

[11] 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

[12] Jun Gong, Yang Zhwoang, 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

[13] 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

[14] 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

[15] 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

[16] 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

[17] 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

[18] 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

[19] Junhyeok Kim, William Delamare and Pourang Irani. 2018. ThumbText: Text Entry for Wearable Devices Using a Miniature Ring. In *Proceedings of the 44th Graphics Interface Conference* (GI '18), 18-25. DOI= https://doi.org/10.20380/GI2018.04

[20] 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

[21] 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

[22] 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

[23] I Scott MacKenzie and William Buxton. 1992. Extending Fitts' law to two-dimensional tasks. in *Proceedings of the SIGCHI conference on Human factors in computing systems* (CHI '92), 219-226. DOI=https://doi.org/10.1145/142750.142794

[24] 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

[25] Aditya Shekhar Nittala, Anusha Withana, Narjes Pourjafarian and Jürgen Steimle. 2018. Multi-Touch Skin: A Thin and Flexible Multi-Touch Sensor for On-Skin Input. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (CHI '18), Paper 33, 12 pages. DOI=https://doi.org/10.1145/3173574.3173607

[26] 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

[27] Antti Oulasvirta, Anna Reichel, Wenbin Li, Yan Zhang, Myroslav Bachynskyi, Keith Vertanen and Per Ola Kristensson. 2013. Improving two-thumb text entry on touchscreen devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '13), 2765-2774. DOI=https://doi.org/10.1145/2470654.2481383

[28] 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

[29] 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

[30] Brian A Smith, Xiaojun Bi and Shumin Zhai. 2015. Optimizing touchscreen keyboards for gesture typing. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (CHI '15), 3365-3374. DOI=https://doi.org/10.1145/2702123.2702357

[31] 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

[32] Saiwen Wang, Jie Song, Jaime Lien, Ivan Poupyrev and Otmar Hilliges. 2016. Interacting with Soli: Exploring Fine-Grained Dynamic Gesture Recognition in the Radio-Frequency Spectrum. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (UIST '16), 851-860. DOI=https://doi.org/10.1145/2984511.2984565

[33] 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

[34] 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
[35] 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
[36] 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
[37] 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
[38] Zheer Xu, Pui Chung Wong, Jun Gong, Te-Yen Wu, Aditya Shekhar Nittala, Xiaojun Bi, Jürgen Steimle, Hongbo Fu, Kening Zhu and Xing-Dong Yang. 2019. TipText: Eyes-Free Text Entry on a Fingertip Keyboard. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (UIST '19), 883-899. DOI=https://doi.org/10.1145/3332165.3347865
[39] Xin Yi, Chun Yu, Weinan Shi, Xiaojun Bi and Yuanchun Shi. 2017. Word clarity as a metric in sampling keyboard test sets. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (CHI '17), 4216-4228. DOI=https://doi.org/10.1145/3025453.3025701
[40] 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
[41] 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
[42] Mingyuan Zhong, Chun Yu, Qian Wang, Xuhai Xu and Yuanchun Shi. 2018. ForceBoard: Subtle Text Entry Leveraging Pressure. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (CHI '18), Paper 528, 10 pages. DOI=https://doi.org/10.1145/3173574.3174102
[43] 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
[44] 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.