

Flickey : 超小型タッチスクリーン端末におけるフリックに基づく QWERTY キーボード

石井 晃^{1,a)} 箱田 博之^{2,b)} 志築 文太郎^{3,c)} 田中 二郎^{3,d)}

概要: スマートウォッチのような超小型タッチスクリーン端末(以降, 超小型端末)は, 身に付けるという特性上, 小型で軽量であることが要求されるため, それらに搭載されるタッチスクリーンは超小型である. これに伴い, タッチに基づくソフトウェア QWERTY キーボードによる文字入力は, キーの押し分けが難しいため, ユーザに負担がかかる. この問題を解決するために, 我々は本稿において, フリックに基づく超小型端末向けソフトウェア QWERTY キーボード「Flickey」(フリッキー)を示す. Flickey では, キーを選択する際にフリックを用いる. このため, ユーザはキーを押し分けずに済むので, 小さいサイズのキーボードにおいてもキーの選択が容易である. Flickey の有用性を検証するため, 我々は Flickey のプロトタイプを作成し, 既存の手法との比較実験を行った. 結果, キーボードサイズが 16.5 mm の場合において, 有意な差はなかったものの Flickey の入力速度が最も速く, Flickey は極めて小さいサイズの超小型端末において有効な手法であることが示唆された. また, 実験から多くの改善点が見出された.

Flickey: A Flick-based QWERTY Software Keyboard for Ultra-small Touch Screen Devices

AKIRA ISHII^{1,a)} HIROYUKI HAKODA^{2,b)} BUNTAROU SHIZUKI^{3,c)} JIRO TANAKA^{3,d)}

Abstract: Ultra-small touch screen devices (e.g., smartwatch) are required to be small and light weight so that it could be worn on body with no frustration. For this reason, users often have difficulties selecting accurate keys and entering texts. To address this problem, we present Flickey, a flick-based QWERTY software keyboard for ultra-small touch screen devices. Users who use Flickey can easily select keys in small-size keyboard because users can select keys by flick. To investigate usability of Flickey, we developed a prototype of Flickey and conducted a comparative experiment between several existing text entry methods. As a result, with the keyboard size of 16.5 mm, text entry speed of Flickey was the fastest (non-significant), and it suggested that Flickey was useful in ultra-small touch screen devices. Furthermore, we found many potential improvements from the experiment.

¹ 筑波大学情報学群情報メディア創成学類
College of Media Arts Science and Technology, School of Informatics, University of Tsukuba

² 筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻
Department of Computer Science, Graduate School of Systems and Information Engineering, University of Tsukuba

³ 筑波大学システム情報系
Faculty of Engineering, Information and Systems, University of Tsukuba

a) ishii@iplab.cs.tsukuba.ac.jp

b) hakoda@iplab.cs.tsukuba.ac.jp

c) shizuki@cs.tsukuba.ac.jp

d) jiro@cs.tsukuba.ac.jp

1. はじめに

スマートウォッチのような超小型タッチスクリーン端末(以降, 超小型端末)への文字入力操作は, 搭載されているタッチスクリーンへのタッチ入力および音声入力によって行われる. しかしながら, 超小型端末のタッチスクリーンのサイズは小さく, 細かな操作をするにはユーザに負担がかかる. また, 音声入力は誤認識やプライバシーの面において問題がある. このため, 現状では, 超小型端末の用途は主として情報提示となっているため, 超小型端末における

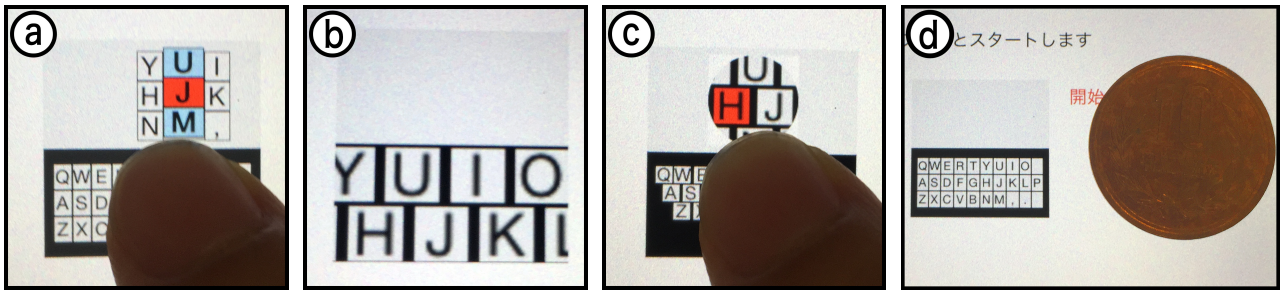


図 1: Flickey のプロトタイプと比較対象 . a) Flickey , b) ZoomBoard , c) ZShift , d) Flickey および 10 円玉 (直径 20 mm) との比較

Fig. 1 The prototype of Flickey and other keyboards used in our experiment. a) Flickey, b) ZoomBoard, c) ZShift, and d) Flickey compared with a 10 JPY coin (diameter: 20 mm).

文字入力性能を向上させることは大きな課題である。

この課題に対して、特に、多くのユーザが慣れ親しんでいる QWERTY 配列のソフトウェアキーボード (以降、QWERTY キーボード) を超小型端末においても使えれば好都合であるが、多くのキーから成る QWERTY キーボード全体をタッチスクリーン内に収まるように表示すると、キーサイズがユーザの指より小さくなるため、ユーザはキーを押し分けることが困難となる [1]。

そこで我々は、フリックを活用した QWERTY キーボード「Flickey」(フリッキー) を考案した (図 1a)。Flickey においてはキーを選択する操作としてフリックを採用した。これにより、ユーザはキーを押し分けずに済むので、小さいサイズのキーボードにおいてもキーの選択が容易である。今回我々は Flickey の性能を評価するため、先行研究である ZoomBoard [2] および ZShift [3] との比較実験を行い、文字入力性能、エラー率、および主観の評価の検証を行った。本稿においては、これらについて報告する。

2. 関連研究

本稿において示す Flickey は、超小型端末におけるフリックを活用した QWERTY キーボードである。したがって、超小型タッチパネル端末における文字入力手法に関する研究、およびフリックなどのタッチジェスチャを活用したソフトウェアキーボードに関する研究と Flickey との位置づけを述べる。

2.1 超小型端末における QWERTY キーボード

超小型端末用の QWERTY キーボードが幾つか提案されている。ZoomBoard [2] を用いるユーザは、キーボード上の大まかな位置をタップにより拡大し、その後もう一度目的のキーをタップする。これにより、ユーザは小さなキーを確実にタップすることが可能となる。Swipeboard [4] を用いるユーザは、9つのグループに分かれたキーを9方向

フリックにより選択し、その後もう一度フリックすることによりキーを選択する。同様に、SplitBoard [5] を用いるユーザは、複数の領域に分けられたキーボードを、スワイプ動作によって切り替える。Komminos ら [6] はキーボードレイアウトを変更し、キーを6つの領域に分けることにより超小型端末においても操作可能なキーボードを提案している。原ら [7] は、スクロールの変位量に応じてキーボードが自動的にズームするキーボードを提案している。ただし、これらのキーボードにおいては、ユーザがタッチする度にキーボードが拡大または移動するため、レイアウトの変化が頻繁に発生する。

Leiva らの QWERTY キーボード [3] は、押下した指によって遮蔽されているキーを画面上部にふきだし表示することにより、超小型端末における文字入力を可能としている。我々の Flickey においても、押下されているキーとその周辺のキーをキーボード上部に表示することにより、指によってキーボード本体が遮蔽されていても文字入力を可能とする。ふきだし表示を用いることにより、キーボードのレイアウトを変更することなく超小型端末においてもキー入力が可能となる。

2.2 タッチジェスチャを活用したソフトウェアキーボード

ソフトウェアキーボードにおけるタッチジェスチャによる文字入力手法は数多く提案されている。Graffiti [8]、Unistroke [9]、および EdgeWrite [10] は、1ストロークで入力可能なアルファベットを模したジェスチャ入力手法である。金井ら [11] は、1ストロークで書くことができるストロークを連続入力することにより、文字入力を行う手法を提案している。Quikwriting [12] はキーを9つの領域に分割表示し、それらを特定の規則にそって、ドラッグすることによって文字入力を実現している。No-look Flick [13] や下岡ら [14] は、段階的な4方向へのフリックによって、アイズフリーなひらがな入力を実現している。同様に、

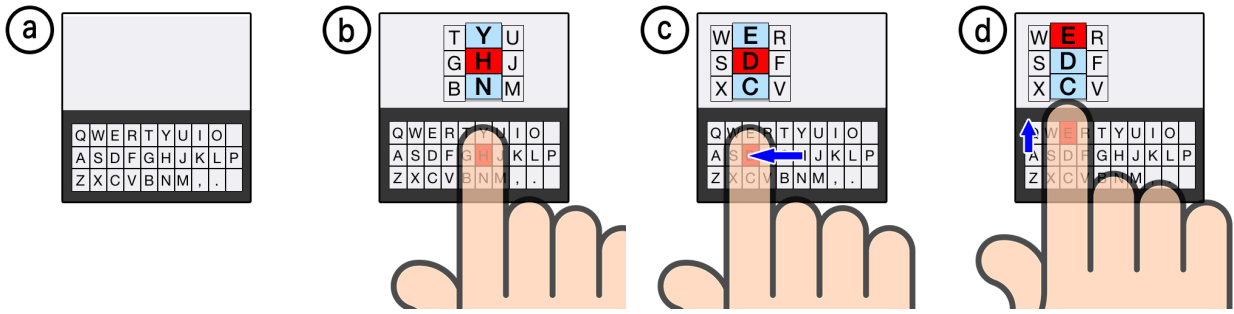


図 2: Flickey における文字入力フロー．a) 初期状態，b) タッチダウン，c) 横移動によりキー列を選択，d) そのままタッチアップ，または上下へフリックして文字を入力

Fig. 2 Text entry flow on Flickey. a) Initial state, b) Touch-down, c) Drag to select a key row, and d) Touch-up or flick-up/down to enter a character.

Drag&Flick [15] も、段階的な 8 方向のドラッグとフリックによって、視覚障害者向けのかな文字入力手法を実現している。これらは、ドラッグやフリックの方向のみの入力を用いているため、超小型端末における文字入力に応用できる可能性がある。我々の Flickey も同様に、キーの選択にフリックを用いることによりキーを押し分ける必要がないため、超小型端末向けのキーボードとなっている。

3. 超小型端末におけるフリックを活用したソフトウェア QWERTY キーボード

本稿において示す Flickey は ZShift [3] と同様にふきだし表示を用い、かつキーボード上における操作をドラッグとフリックに限定するアプローチをとる QWERTY キーボードである。特にフリックを活用することによりキーを押し分けずに済むため、Flickey は小さいサイズのキーボードにおいてもキーの選択が容易であるという特徴を有す。

本節では、比較対象である ZoomBoard および ZShift を述べた後、我々の提案手法である Flickey を述べる。最後に、ZoomBoard, ZShift, および Flickey に実装されている共通の機能を述べる。

3.1 ZoomBoard

ZoomBoard はキーボードの特定箇所を TapTap [16] のように拡大表示することにより、キーの押し分けを容易にする。まず、初期状態（キーボード全体がタッチスクリーン内に収まるように表示されている）においてユーザがキーボードをタップすると、キーボードはタップ位置を中心として一定の倍率にて拡大する（図 1b）。ユーザはこれを決められた回数繰り返した後、キーをタップすることにより文字が入力される。その後、キーボードは初期状態に戻る。このように、ZoomBoard はキーボードを拡大表示することによって、Fat Finger 問題 [1] を低減する。結果として、キーの押し分けが容易になるため、非拡大状態の

QWERTY キーボードと比べ誤入力が減少する。

3.2 ZShift

ユーザが押下しているキーをキーボード上部に表示する手法は、iOS の QWERTY キーボードにおいても採用されており、広く使用されている。しかしながら、超小型端末においては、指の大きさに対してキーボードが小さいため、指を画面上に押下するとキーボードの大部分が指によって遮蔽される。したがって、入力したいキーを、キーボードをなぞりながら探す場合、従来の手法においてはユーザの QWERTY キーボードに対する空間的記憶に強く依存する。そこで、ZShift は Shift [17] という、押下されている位置とその周囲を画面上部にふきだし状にて表示する手法を用いることによりこの問題を解決している。しかしながら、特に小型のキーボードにおいては、単にふきだし表示するだけではキーが小さく表示されるため見づらさを伴う。そこで ZShift では、Shift に拡大表示を加え（Zoomed Shift technique, ZShift）、かつ押下されているキーを強調表示することによってふきだし表示を改善している（図 1c）。

3.3 Flickey

本稿において示す Flickey は ZShift [3] と同様にふきだし表示を用い、かつキーボード上における操作をドラッグとフリックに限定するアプローチをとる QWERTY キーボードである。特にフリックを活用することによりキーを押し分けずに済むため、小さいサイズのキーボードにおいてもキーの選択が容易である。Flickey における文字入力のフローを図 2 に示す。Flickey においては 2 段階の入力操作によって文字入力を行う。はじめに、ユーザはタッチダウンによってキー列の選択を行う（図 2b）。その後、タッチアップもしくはフリックによってキーの選択を行う（図 2d）。タッチアップにより選択されたキー列の中段のキー

が、上フリックまたは下フリックにより上下段のキーが入力される。最初のタッチダウン時に入力したいキー列とは異なるキー列を選択していた場合は、フリックせずに左右に指をドラッグすることによって、選択しているキー列を変更することができる(図 2c)。ユーザは最初のキー列選択時には横方向ドラッグ(X軸方向)のみを行うため縦方向(Y軸方向)の動きを気にする必要がない。その後キーの上中下段の選択もフリックを活用しているため、正確な座標入力を必要としない。結果として、Flickey においてはキーを押し分ける負担が軽減されている。なお、タッチダウン時には現在選択されているキーが画面上部に表示される(図 1a)。

3.4 共通の機能

これまでにタッチスクリーン上のキーを減らすためにスワイプジェスチャが使われてきた [2,3,18,19]。ZoomBoard および ZShift にも実装されており、我々も Flickey に以下の機能を実装した。ユーザはキーボード上にて左スワイプを行うことにより直前の文字を削除することができる。またキーボード上にて右スワイプを行うことにより空白文字を入力することができる。

4. 評価実験

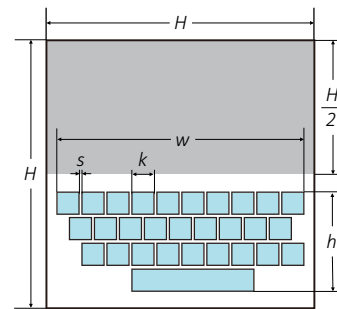
Flickey の性能を評価するために、ZoomBoard および ZShift を比較対象とし、文字入力をタスクとする被験者実験を実施した。プレスレット型の端末から、時計型の端末まで様々なサイズ*1の端末における性能を評価するため、3つのキーボードを3つの異なるサイズ (small : 16.5 mm, medium : 22 mm, large : 29.3 mm) にて提示した(計9条件)。また、本実験においてはスマートフォン上に、スマートウォッチを想定した小さなキーボードを実装し、これを実験に用いた。この理由は、3つのサイズごとに異なるスマートウォッチを用いて実験をした場合、スクリーンの解像度やタッチ感度など端末の性能差による実験結果への影響が懸念されるためである。

4.1 実験機器

キーボードを提示するスマートフォンとして、iPhone 5 (iOS 8.3, 画面サイズ: 4 inch, 解像度: 326 ppi) を用いた。図 3 に示すように、スマートフォンを2つのひざ用サポータ(株式会社 D&M, ひざ下ベルト ラップタイプ, 842XUD2786 BLK M) を用いて、被験者の非利き腕に横向きに取り付けた。このサポータはマジックテープによって固定力の調節が可能であるため、被験者の腕の太さに柔



図 3: 非利き腕に横向きで装着されたスマートフォン
Fig. 3 The smartphone attached in landscape orientation to the non-dominant hand.



サイズ [mm]	H	w	h	s	k
small	18.0	16.5	6.5	0.2	1.5
medium	24.0	22.0	8.7	0.2	2.0
large	32.0	29.3	11.6	0.3	2.7

図 4: 実験で用いたキーボードのレイアウト

Fig. 4 The layout of the keyboards used in the experiment.

軟に対応できる。また、スポーツ用途であるため、伸縮性があり、かつ滑らないように設計されているため、スマートフォンを腕に確実に固定することが可能である。

4.2 キーボード

各キーボードを iOS 端末上にて動作するアプリケーションとして実装した(図 1)。本実験において用いたキーボードのレイアウトを図 4 に示す。サイズに関しては、ZoomBoard [2] において使用されていた 16.5 mm を基準に、Leiva ら [3] の論文において使用されていた3つのサイズの決め方(基準値から3分の4倍ずつ拡大 (small : 1.00倍, medium : 1.33倍, large : 1.77倍))を用いて決定した。なお、すべてのキーボードは、一般的なスマートフォン向け QWERTY キーボードより小さく、small サイズのキーボードの面積は iPhone 6 (4.7 inch) において用いられている QWERTY キーボードの約 20 分の 1 (0.054 倍) である(図 1d)。

実験中、被験者が行った全ての操作を時刻情報およびスクリーンキャプチャとともに記録した。また被験者の発言をボイスレコーダを用いて記録した。

*1 ウェアラブル端末における画面サイズの例 (width × height):

- Samsung Gear Fit 13 × 45 mm
- Apple Watch (38 mm) 21 × 26 mm
- Apple Watch (42 mm) 24 × 30 mm

4.3 被験者

大学生および大学院生 5 名 (男性 4 名, 女性 1 名, 年齢: 21–22 歳) を被験者として雇用した。すべての被験者が日常的にスマートフォンを使用しており (利用歴: 29–67 ヶ月, 平均: 53 ヶ月), 利き腕は右腕であった。スマートフォンにおける文字入力には 3 名が右手のみを使用, 1 名が右手および左手を使用, 1 名が右および両手を使用していた。スマートフォンにおける日常的な英語文字入力には, 2 名は QWERTY キーボードを使用し, 1 名はフリック入力を使用し, 1 名はその両方を併用していた。日本語文字入力には, 3 名はフリック入力を使用し, 1 名はフリック入力および QWERTY キーボードを併用していた。また, 1 名は英語文字入力および日本語文字入力において, ATOK 式ジェスチャ入力 [20] を使用していた。コンピュータにおける QWERTY キーボードの利用歴は 125–213 ヶ月 (平均: 171 ヶ月) であった。全ての被験者は, スマートウォッチの使用経験がなかった。また, 腕時計を着用する時は全ての被験者が非利き腕に着用すると回答した。実験終了後, 各被験者には実験への参加に対する謝礼として 1,640 円を支払った。

4.4 手順

実験は静かな室内で行われた。被験者は実験中, 常に着席しているように求められた。

まず, 実験者は実験内容の説明を行った。その後, スマートフォンの使用歴等を問うアンケートに回答するように指示した。アンケート回答後, デジタルノギスを用いて被験者の人差し指 (利き腕) の遠位指節間関節の幅 (図 5) を測定した。平均サイズは 14.3 mm (SD = 0.8 mm) であり, 日本人の標準的なサイズであった [21]。測定完了後, スマートフォンをサポータを用いて被験者の非利き腕に装着した。

実験者は被験者に対して 3 つのキーボードを提示し, それぞれの手法の説明を行った。その後, medium サイズの各キーボードを用いて練習タスクを行うウォームアップセッションを実施した。練習タスクとして, まず「taro」と入力してもらった。その後, 文字の削除ジェスチャ (キーボード上での左スワイプ) に慣れるため, 先ほど入力した文字を全て削除するように求めた。文字の削除が終了したら, 次に「tsukuba」と入力してもらった。その後, 空白文字の入力ジェスチャ (キーボード上での右スワイプ) に慣れるため, 先ほど入力した文字の直後に空白を入力するように求めた。空白の入力が終了したら, 次に「taro」と入力してもらった。以上の練習タスクを各キーボードに対して行った。なお, 練習を行うキーボードの順序は本番タスクにおいて使用するキーボードの出現順と同様にし, カウンターバランスをとった。このウォームアップセッションには平均 6 分間を要した。

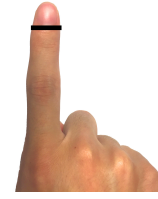


図 5: 人差し指のサイズの測定位置
Fig. 5 Measurement position of index finger.

実験者は被験者に, 各手法および各サイズごとに 5 フレーズの入力 (5 試行) を求め, これを 1 セッションとし条件を変えて 9 セッション行った。したがって, 被験者は計 45 試行 (3 キーボード × 3 サイズ × 5 フレーズ) を行った。なお, 各手法間におけるフレーズの提示順が実験結果に影響を及ぼす恐れがあるため, 被験者に提示するフレーズをフレーズセットからランダムに選択した。このフレーズセットは MacKenzie ら [22] によって提供されている 500 種類の英語のフレーズである。個々のフレーズはすべて小文字にて記述された単語からなり, 句読点および数字を含まない。またフレーズ内の単語と単語の間は空白文字にて区切られている。なお, 学習効果を抑えるため全試行において入力するフレーズが重複しないようにした。入力するフレーズの記憶強度に結果に影響されないように, 入力が完了するまでの間は常に入力すべきフレーズを被験者に提示した。フレーズの入力が終了したら, キーボードの隣に配置された「次へ」ボタンを押すように指示した。「次へ」ボタンが押されると, 自動的に現在のフレーズおよび入力した文字が消え, 次に入力すべきフレーズが表示される。

実験者は被験者に, 各試行においてなるべく素速く, なるべく正確に文字入力を行うように指示した。また, 提示されたフレーズに含まれる空白も入力し, 入力文字を間違えた場合は訂正するように指示した。そして, 実験中の操作をすべて利き腕の人差し指を用いて行うように指示した。なお, 疲労による入力速度への影響を抑えるため, 各セッションの間に 3 分間程度の休息を挟んだ。

被験者は各セッションの計測開始前に毎回該当するキーボードにて練習が可能であった。実験者は被験者に, これ以上練習しなくても迷わず入力を行えると判断した場合, 練習を終了し実験者にその旨を伝えるように指示した。この練習は各被験者ごとに合計で 2 分から 7 分間程度要した。

各セッションの終了時に, 実験者は被験者に System Usability Scale (SUS) [23, 24] および NASA Task Load Index (NASA-TLX) [25] に回答するように求めた。しかしながら, SUS および NASA-TLX は英語で記述されている。そのため, 実験において用いた SUS は古井ら [26] によって使用された日本語版を使用した。表 1 に今回用いた SUS の質問文を示す。なお, 一部の質問文に原文の意図が反映しき

表 1: SUS の原文と今回の実験において用いた日本語訳
Table 1 The Japanese version of SUS.

(1) I think that I would like to use this system frequently. わたしはこのシステムを頻繁に使いたいと思う。
(2) I found the system unnecessarily complex. このシステムは無駄に複雑であると思った。
(3) I thought the system was easy to use. このシステムは簡単に使えると思った。
(4) I think that I would need the support of a technical person to be able to use this system. このシステムを使えるようになるには、わたしは技術者の支援を必要とするだろうと思う。
(5) I found the various functions in this system were well integrated. このシステムでは様々な機能がよくまとまっていると思った。
(6) I thought there was too much inconsistency in this system. このシステムにはあまりにも多くの矛盾があると思った。
(7) I would imagine that most people would learn to use this system very quickly. ほとんどの人々はこのシステムの使い方をすぐに覚えるだろうと思う。
(8) I found the system very cumbersome to use. このシステムはとても扱いにくいと思った。
(9) I felt very confident using the system. このシステムを使用できる自信があったと感じた。
(10) I needed to learn a lot of things before I could get going with this system. わたしはこのシステムを使い始める前に、多くのことを学ぶ必要があった。

れていない恐れのある表現が見受けられたため、日本語訳の修正を行った(表 1(9))。また、今回用いた NASA-TLX は、三宅ら [27] によって日本語に翻訳されたものである。全てのセッション終了時に各手法に対するアンケートに回答するように求めた。

本実験は、実験説明からすべてのセッションおよびアンケートへの回答が終了するまでに 120 分間程度の時間を要した。

5. 実験結果と考察

5.1 文字入力性能

文字入力速度

今回我々は文字入力速度の指標として 1 分あたりの文字入力速度 (Words Per Minute, WPM) [28] を用いた。

表 2 および図 6 に各条件ごとの文字入力速度および分散分析の結果を示す。なお括弧内は標準偏差である。分散分析を行ったところ、全てのサイズ条件において手法間に有意な差があった。多重比較検定 (Tukey's HSD) を行った結果、small 条件においては、ZoomBoard - ZShift 間 ($p < .05$) および ZoomBoard - Flickey 間 ($p < .01$) に有意

表 2: 文字入力速度 (WPM)
Table 2 Text entry speed (WPM).

サイズ	キーボード			分散分析	
	ZoomBoard	ZShift	Flickey	$F_{2,12}$	p
small	7.5 (0.3)	8.5 (0.7)	8.7 (0.3)	8.2	.006
medium	8.7 (0.5)	10.1 (0.4)	9.4 (0.5)	13.1	.001
large	8.9 (0.2)	12.4 (0.5)	8.8 (0.9)	58.1	.000

な差があった。また medium 条件においては、ZoomBoard - ZShift 間に有意な差があった ($p < .001$)。さらに large 条件においては、ZoomBoard - ZShift 間 ($p < .001$) および ZShift - Flickey 間 ($p < .001$) 間に有意な差があった。結果として、small 条件においては ZoomBoard と比べ ZShift および Flickey が有意に速く、medium 条件においては ZoomBoard と比べ ZShift が有意に速く、large 条件においては ZShift が他のキーボードに比べ有意に速いことが示された。

アンケート調査により Flickey は「小さいサイズでは文字が正確に打てた」との回答が得られた一方、キーボードのサイズが変化するに伴って上下フリックの移動量 (フリック判定しきい値) も増減するため「small の後に large を操作すると少し違和感があった」、「large だと自分が思っているよりも大きめに動かさないと上下段が打てないことがあった」等の回答も得られた。これに関連して、他の手法においてはサイズが大きくなるにつれ文字入力速度が速くなっているのに対し、Flickey は横ばいである。この理由として、前述のように、Flickey にはサイズ変化に伴ってフリック判定しきい値が増減することにより使用感に変化が生じるという特性があるため、それが実験結果に影響を与えたと考えられる。

エラー率

今回我々はエラー率の指標として Character Error Rate (CER) を用いた。CER は、転写タスクにおいて入力されたフレーズの正確性を測定するための最も広く使用されている指標である [3]。CER は最終的に入力されたフレーズと入力すべきフレーズ間のレーベンシュタイン距離から計算され、入力すべきフレーズの文字数で割ることによって正規化される。

表 3 に CER を示す。なお括弧内は標準偏差である。分散分析を行ったところ、small サイズ条件においてのみ有意な差があった。この small 条件において多重比較検定 (Tukey's HSD) を行ったところ、ZShift - Flickey 間 ($p < .001$) および ZShift - ZoomBoard 間 ($p < .001$) に有意な差があった。結果として ZShift は他のキーボードに比べ small 条件においては CER が有意に高いという結果が得られた。

CER に加えて今回我々は修正済みエラー率 (Corrected error rate, Cerr), すなわち全入力に対する誤入力を修正

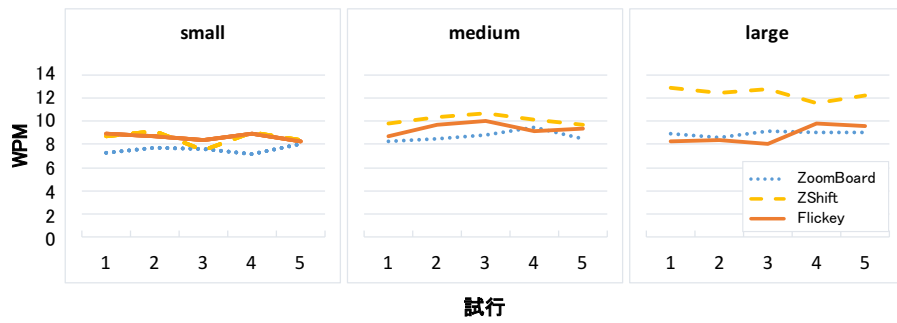


図 6: 文字入力速度 (WPM)
Fig. 6 Text entry speed (WPM).

表 3: エラー率 (CER)
Table 3 Error rate (CER).

サイズ	キーボード			分散分析	
	ZoomBoard	ZShift	Flickey	$F_{2,12}$	p
small	0.00% (0.00)	0.55% (0.31)	0.00% (0.00)	15.82	.0004
medium	0.64% (1.42)	0.41% (0.64)	0.00% (0.00)	0.64	.543
large	0.00% (0.00)	0.29% (0.64)	0.26% (0.36)	0.69	.519

表 4: 修正済みエラー率 (Cerr)
Table 4 Corrected error rate (Cerr).

サイズ	キーボード			分散分析	
	ZoomBoard	ZShift	Flickey	$F_{2,12}$	p
small	4.59% (3.28)	5.53% (1.94)	9.47% (5.15)	2.45	.128
medium	4.70% (3.68)	8.95% (3.66)	10.01% (3.71)	2.91	.093
large	2.01% (1.89)	6.52% (4.99)	9.48% (3.44)	5.26	.022

表 5: 未修正エラー率 (UCerr)
Table 5 Uncorrected error rate (UCerr).

サイズ	キーボード			分散分析	
	ZoomBoard	ZShift	Flickey	$F_{2,12}$	p
small	0.00% (0.00)	0.47% (0.27)	0.00% (0.00)	15.61	.0004
medium	0.64% (1.44)	0.39% (0.62)	0.00% (0.00)	0.64	.543
large	0.00% (0.00)	0.26% (0.59)	0.23% (0.32)	0.69	.519

した割合も求めた [29]。表 4 に Cerr を示す。なお括弧内は標準偏差である。分散分析を行ったところ、large 条件においてのみ手法間に有意な差があった。この large 条件において多重比較検定 (Tukey's HSD) を行ったところ、ZoomBoard - Flickey 間に有意な差があった ($p < .05$)。結果として Flickey は ZoomBoard に比べ large 条件においては Cerr が有意に高いという結果が得られた。

また、未修正エラー率 (Uncorrected error rate, UCerr)、すなわち全入力に対する誤入力の割合も求めた。表 5 に UCerr を示す。なお括弧内は標準偏差である。分散分析を行ったところ、small 条件においてのみ手法間に有意な差があった。この small 条件において多重比較検定 (Tukey's HSD) を行ったところ、ZShift - Flickey 間 ($p < .001$) および ZShift - ZoomBoard 間 ($p < .01$) に有意な差があった。結果として ZShift は他のキーボードに比べ small 条件においては UCerr が有意に高いという結果が得られた。

Flickey において、どのようなエラーがあったかを分析したところ、目的のキー列とは 1 つずれたキー列が選択されている事象が多く見られた。すなわち、横方向ドラッグによりキー列を選択している時に誤って移動しすぎた結果、目的のキー列とは別のキー列を選択していた。この原因としてふきだし表示内の視覚的フィードバックが離散的であったことが考えられる。ZShift ではドラッグに応じて、ふきだしの位置やふきだし内の表示が連続的に変化するため、ユーザはあとどれくらい指を動かせば目的のキーに辿り着けるのかを逐次確認することができた。一方で、Flickey のふきだし表示はその表示が離散的なために、あとどれくらい指を動かせば隣のキー列に移動できるのか、という情報をユーザにフィードバックできていなかった。

この結果として、誤入力が生じたと考えられる。この問題の解決策として、Flickey のふきだし表示をドラッグに応じて連続的に変化させることが挙げられる。

人差し指の太さと速度およびエラー率の相関

4.4 節において述べたように、今回我々は実験前に被験者の人差し指 (利き腕) のサイズを測定した。この人差し指の太さと文字入力速度およびエラー率 (Cerr) との相関分析を行ったところ、全てのキーボード・サイズ条件における被験者ごとに文字入力速度を足し合わせた結果との間に強い正の相関があった ($\rho = .906, p < .05$)。また、ZoomBoard (large サイズ) の Cerr との間に強い負の相関があった ($\rho = -.952, p < .05$)。つまり、指の太さが小さければ小さいほど Fat Finger 問題 [1] が解決され、文字入力速度の向上とエラー率の低下に繋がったと考えられる。

5.2 ユーザビリティとメンタルワークロード

SUS を用いて測定した各条件のユーザビリティを表 6 に示す。なお括弧内は標準偏差である。SUS の値は高い方が良い結果と言える。分散分析を行ったところ、small, medium 条件においては手法間に有意な差はなかったが、large 条件においては有意な差がみられた。この large 条件において多重比較検定 (Tukey's HSD) を行ったところ、

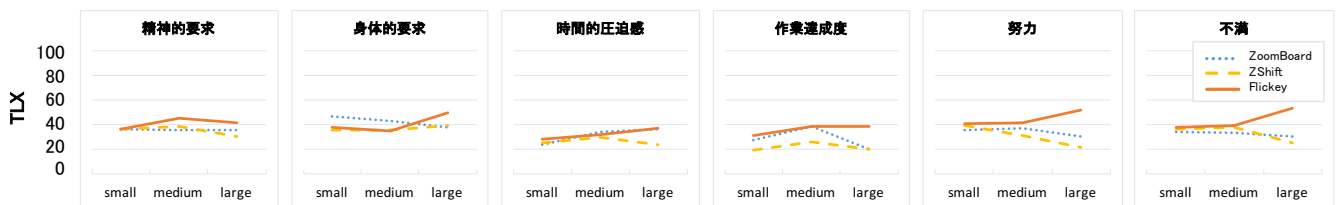


図 7: 6 つの NASA-TLX 評価尺度項目の結果

Fig. 7 Results of each of the six NASA-TLX dimensions.

表 6: 各条件のユーザビリティ (SUS)

Table 6 Usability of each condition (SUS).

サイズ	キーボード			分散分析	
	ZoomBoard	ZShift	Flickey	$F_{2,12}$	p
small	66.0 (13.3)	60.0 (15.2)	59.0 (12.9)	0.37	.696
medium	72.0 (11.4)	69.5 (14.1)	62.0 (10.8)	0.91	.427
large	69.5 (17.0)	81.5 (11.8)	54.0 (9.8)	5.44	.021

ZShift – Flickey 間に有意な差 ($p < .05$) があつた。結果として large 条件においては Flickey よりも ZShift の方が有意に使いやすいことが示された。

NASA-TLX を用いて測定した各条件のメンタルワークロードを図 7 および表 7 に示す。括弧内は標準偏差である。NASA-TLX の値は低い方が良い結果と言える。分散分析を行ったところ、全てのサイズ条件において手法間に有意な差はなかつた。

ZShift に関しては、サイズが大きくなるにつれて SUS の評価が高くなっていった。これはサイズが大きくなればなるほど通常の QWERTY キーボードに操作感が近くなっていくためだと考えられる。アンケート調査においても「large 条件のときは、もはや普通の入力として使った」という意見が得られた。ZoomBoard に関しては、サイズが変化しても 2 段階入力という操作感は変わらないため、SUS および NASA-TLX の両者において評価が横ばいになったと考えられる。アンケート調査においても「毎回 2 度タップするため操作が少し煩わしい」という意見が得られた。また、「ズームする位置が毎回変わりレイアウトが変化するので、常にキーボードを注視する必要があつた」という意見も得られた。Flickey は large 条件において small 条件と比較し低い評価が SUS および NASA-TLX の両者において得られた。これは large 条件程度のキーボードサイズの場合、ピンポイントにキーを選択可能であり、わざわざキーの選択にフリックを用いることはかえって操作を複雑にしていると考えられる。この結果から Flickey は small 条件のような極めて小さいサイズにおいて有効な手法であることが示唆された。

6. まとめと今後の課題

我々はフリックを活用したキーの押し分けの負担が少ない QWERTY キーボード「Flickey」を考案した。その後プロトタイプを作成し、既存手法と比較実験を行った。small

表 7: 各条件のメンタルワークロード (NASA-TLX)

Table 7 Workload of each condition (NASA-TLX).

サイズ	キーボード			分散分析	
	ZoomBoard	ZShift	Flickey	$F_{2,12}$	p
small	36.4 (24.9)	32.5 (19.7)	37.3 (21.3)	0.07	.937
medium	40.2 (22.1)	34.7 (25.2)	41.9 (19.8)	0.14	.872
large	33.7 (13.7)	27.6 (24.1)	49.0 (21.9)	1.46	.271

条件において、有意な差はなかつたものの、Flickey の入力速度が最も速かつた (ZoomBoard : 7.5 WPM, ZShift : 8.5 WPM, Flickey : 8.7 WPM)。また、本実験において多くの改善点が見出された。Flickey はフリックを用いるため、キーボードサイズが変化した際にフリック判定のしきい値が変化し、同時に使用感も変化する。そのため、違和感を調査する追実験を行い、各サイズにおけるしきい値を違和感の少ないものに調整することを考えている。さらに、Flickey はふきだし表示内の視覚的フィードバックが離散的なために、他の手法と比べ誤入力が多いことが明らかとなった。この問題を解決するため、ふきだしの表示内容をユーザのドラッグに応じて連続的に変化させることを考えている。今後はこれらの改善点を実装し、さらなる性能評価実験を行う予定である。

参考文献

- [1] Siek, K. A., Rogers, Y. and Connelly, K. H.: Fat Finger Worries: How Older and Younger Users Physically Interact with PDAs, in *Proceedings of the 2005 IFIP TC13 International Conference on Human-Computer Interaction*, INTERACT'05, pp. 267–280, Berlin, Heidelberg (2005), Springer-Verlag.
- [2] Oney, S., Harrison, C., Ogan, A. and Wiese, J.: ZoomBoard: A Diminutive Qwerty Soft Keyboard using Iterative Zooming for Ultra-small Devices, in *Proceedings of the 31st Annual ACM Conference on Human Factors in Computing Systems*, CHI '13, pp. 2799–2802, New York, NY, USA (2013), ACM.
- [3] Leiva, L. A., Sahami, A., Catala, A., Henze, N. and Schmidt, A.: Text Entry on Tiny QWERTY Soft Keyboards, in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pp. 669–678, New York, NY, USA (2015), ACM.
- [4] Chen, X. A., Grossman, T. and Fitzmaurice, G.: 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, pp. 615–620, New York, NY, USA (2014), ACM.

- [5] Hong, J., Heo, S., Isokoski, P. and Lee, G.: SplitBoard: A Simple Split Soft Keyboard for Wristwatch-sized Touch Screens, in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pp. 1233–1236, New York, NY, USA (2015), ACM.
- [6] Komminos, A. and Dunlop, M.: Text Input on a Smart Watch, *Pervasive Computing, IEEE*, Vol. 13, No. 4, pp. 50–58 (2014).
- [7] 原清貴, 梅澤猛, 大澤範高: 腕時計型端末におけるズーム・スクロールを用いたタッチ入力, *インタラクティブシステム 2014*, pp. 317–320, 一般社団法人情報処理学会 (2014).
- [8] MacKenzie, I. S. and Zhang, S. X.: The Immediate Usability of Graffiti, in *Proceedings of the Conference on Graphics Interface '97*, pp. 129–137, Canadian Information Processing Society (1997).
- [9] Goldberg, D. and Richardson, C.: Touch-typing with a Stylus, in *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, INTERACT '93 and CHI '93, pp. 80–87, New York, NY, USA (1993), ACM.
- [10] Wobbrock, J. O., Myers, B. A. and Kembel, J. A.: EdgeWrite: A Stylus-based Text Entry Method Designed for High Accuracy and Stability of Motion, in *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*, UIST '03, pp. 61–70, New York, NY, USA (2003), ACM.
- [11] 金井達巳, 宮下芳明: 極小タッチデバイスにおける文字の分割入力手法, 第 22 回インタラクティブシステムとソフトウェアに関するワークショップ, WISS2014, pp. 145–146, 日本ソフトウェア科学会 (2014).
- [12] Perlin, K.: Quikwriting: Continuous Stylus-based Text Entry, in *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology*, UIST '98, pp. 215–216, New York, NY, USA (1998), ACM.
- [13] Fukatsu, Y., Shizuki, B. and Tanaka, J.: No-look Flick: Single-handed and Eyes-free Japanese Text Input System on Touch Screens of Mobile Devices, in *Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services*, Mobile-HCI '13, pp. 161–170, New York, NY, USA (2013), ACM.
- [14] 下岡純也, 浅井洋樹, 山名早人: スマートウォッチにおけるアイズフリー日本語入力手法, 第 23 回インタラクティブシステムとソフトウェアに関するワークショップ, WISS2015, pp. 145–146, 日本ソフトウェア科学会 (2015).
- [15] 青木良輔, 橋本遼, 瀬古俊一, 片岡泰之, 井原雅行, 渡辺昌洋, 小林透: Drag&Flick: タッチスクリーンを用いた視覚障がい者向け文字入力方式, *インタラクティブシステム 2013*, pp. 72–79, 一般社団法人情報処理学会 (2013).
- [16] Roudaut, A., Huot, S. and Lecolinet, E.: TapTap and MagStick: Improving One-handed Target Acquisition on Small Touch-screens, in *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '08, pp. 146–153, New York, NY, USA (2008), ACM.
- [17] Vogel, D. and Baudisch, P.: Shift: A Technique for Operating Pen-based Interfaces using Touch, in *Proceedings of the 25th Annual ACM Conference on Human Factors in Computing Systems*, CHI '07, pp. 657–666, New York, NY, USA (2007), ACM.
- [18] Findlater, L., Wobbrock, J. O. and Wigdor, D.: Typing on Flat Glass: Examining Ten-finger Expert Typing Patterns on Touch Surfaces, in *Proceedings of the 29th Annual ACM Conference on Human Factors in Computing Systems*, CHI '11, pp. 2453–2462, New York, NY, USA (2011), ACM.
- [19] Li, F. C. Y., Guy, R. T., Yatani, K. and Truong, K. N.: The ILine Keyboard: A QWERTY Layout in a Single Line, in *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pp. 461–470, New York, NY, USA (2011), ACM.
- [20] ジャストシステム: スマートフォン向け日本語入力システム ATOK for Android, http://www.justsystems.com/jp/products/atok_android/ 2016 年 1 月 10 日閲覧.
- [21] 河内まき子: AIST 日本人の手の寸法データ, <https://www.dh.aist.go.jp/database/hand/index.html> 2016 年 1 月 10 日閲覧.
- [22] MacKenzie, I. S. and Soukoreff, R. W.: Phrase Sets for Evaluating Text Entry Techniques, in *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '03, pp. 754–755, New York, NY, USA (2003), ACM.
- [23] Brooke, J.: SUS: A Quick and Dirty Usability Scale, *Usability Evaluation in Industry* (1996).
- [24] Brooke, J.: SUS: A Retrospective, *J. Usability Studies*, Vol. 8, No. 2, pp. 29–40 (2013).
- [25] Hart, S. G. and Staveland, L. E.: Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research, *Human mental workload*, Vol. 1, No. 3, pp. 139–183 (1988).
- [26] 古井陽之助, 前田環貴, 松本誠義: 講義画面の自動連続キャプチャを行う講義・学習支援システムとその評価実験, 火の国情報シンポジウム 2014, 情報処理学会, 8 pages (2014).
- [27] 三宅晋司, 神代雅晴: メンタルワークロードの主観的評価法, *人間工学*, Vol. 29, No. 6, pp. 399–408 (1993).
- [28] Arif, A. S. and Stuerzlinger, W.: Analysis of Text Entry Performance Metrics, in *Proceedings of the IEEE Toronto International Conference on Science and Technology for Humanity*, pp. 100–105, IEEE (2009).
- [29] Soukoreff, R. W. and MacKenzie, I. S.: Metrics for Text Entry Research: An Evaluation of MSD and KSPC, and a New Unified Error Metric, in *Proceedings of the 21st Annual ACM Conference on Human Factors in Computing Systems*, CHI '03, pp. 113–120, New York, NY, USA (2003), ACM.