

# Flickey: 超小型タッチパネル端末におけるフリック操作を活用した QWERTY キーボード

石井 晃<sup>1,a)</sup> 箱田 博之<sup>2</sup> 志築 文太郎<sup>3</sup> 田中 二郎<sup>3</sup>

概要: ウェアラブル端末は、体に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パネルは超小型である。これに伴い、タッチ操作に基づくソフトウェア QWERTY キーボードによる文字入力は、キーの押し分けが困難であるため、難しい。この問題に対して本研究は、超小型タッチパネル向けソフトウェア QWERTY キーボードの開発を目的とする。我々は本稿において、フリック操作を活用したソフトウェア QWERTY キーボード「Flickey」(フリッキー)を示す。フリック操作を活用することによりキーを押し分けずに済むので、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。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、有意差はなかったものの Flickey の入力速度が最も速かった。また、実験から多くの改善点が見出された。

## 1. はじめに

スマートウォッチのような超小型タッチパネル端末が注目を集めている。携帯性に優れた端末であり、主に情報提示を行う端末として用いられている。これらの端末への入力操作は、搭載されている超小型タッチパネルへのタッチ入力によって行われる。

しかし、タッチパネルのサイズが小さいため、表示可能な情報量が制限されている。これに伴い、このような超小型タッチパネル端末におけるキーボードを用いた文字入力においては、キーサイズがユーザの指より小さくなるため、キーを押し分けることが困難となる [1]。

そこで我々は、フリック操作を活用したソフトウェア QWERTY キーボード「Flickey」(フリッキー)を考案した(図 1d)。フリック操作を活用することによりキーを押し分けずに済むので、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。Flickey の性能を評価するため、先行研究である ZoomBoard、およ

び ZShift との比較実験を行い、文字入力性能、エラー率、および主観的評価の検証を行った。本稿では、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、超小型タッチパネル端末におけるフリック操作を活用したソフトウェア 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を述べる。

### 2.1 超小型タッチパネル端末における文字入力手法

ZoomBoard [2] は、キーボード上の大まかな位置の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とにより入力するキーボードである。これにより、小さなキーでも確実にタップすることが可能となる。Swipeboard [3] 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して文字入力を行うキーボードである。同様に、SplitBoard [4] 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ことにより、超小型タッチパネルにおいても文字入力可能な手法である。原ら [5] 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。これらは、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、

<sup>1</sup> 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

<sup>2</sup> 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

<sup>3</sup> 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

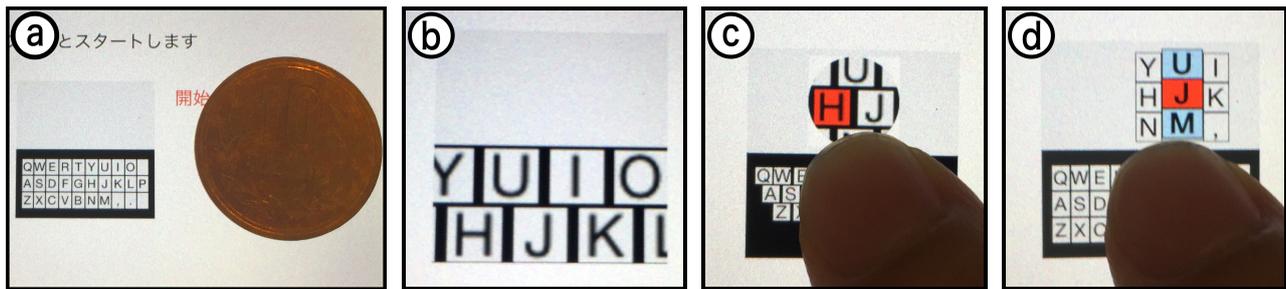


図 1: 作成したプロトタイプ . a) 10 円玉との比較 , b) ZoomBoard , c) ZShift , d) Flickey

Fig. 1 Our prototypes. a) Compared with a 10 YEN coin. b) ZoomBoard, c) ZShift, d) Flickey

レイアウトの変化が頻繁に発生する .

Leiva ら [6] 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フキダシ表示することにより , 超小型タッチパネルにおいても文字入力可能な手法を提案している . 我々も同様に , 押下し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行うことが可能である .

## 2.2 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 [7] は , アルファベットに形状に近いジェスチャ入力を 1 ストロークで入力する手法である . 金井と宮下ら [8] は , 一筆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 . No-look Flick [9] は , 段階的な 4 方向へのフリック入力によって , アイズフリーにひらがな入力を行う文字入力システムである . 同様に , Drag&Flick [10] も , 段階的な 8 方向のドラッグ操作とフリック操作によって , 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であるため , 超小型タッチパネル端末における文字入力に応用できる可能性がある . 我々も同様に , 文字の決定にフリック操作を用いることによりキーを押し分ける必要が無いため , 超小型タッチパネル端末向けキーボードとなっている .

## 3. 超小型タッチパネル端末におけるフリック操作を活用した QWERTY キーボード

本稿において示す Flickey は ZShift [6] に基づき , キーボード上での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 . フリック操作を活用することによりキーを押し分けずに済むので , 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 に

ついて述べる . その後 , 我々の提案手法である Flickey について述べる . 最後に , ZoomBoard , ZShift および Flickey に実装されている共通の機能について述べる .

### 3.1 ZoomBoard キーボード

超小型タッチパネル端末上の QWERTY キーボードは非常に小さいため , そのままではキーの押し分けは困難を極める . ZoomBoard ではキーボードの特定箇所にズームすることでキーの押し分けを容易にする . ユーザがキーボードの適当な位置をタップすると , キーボードは目的の倍率になるまで反復的に拡大する ( 図 1b) . キーボードが十分に拡大されたら , ユーザがキーをタップすることで文字が入力される . その後 , キーボードは初期状態 ( 拡大率 : 1.0 ) に戻る . キーボードが拡大されることで自身の指によって画面が隠れてしまう問題を低減することができる . 結果として , キーの押し分けが容易になり , 非ズーム状態の QWERTY ソフトウェアキーボードと比べ誤入力が減少する .

### 3.2 ZShift キーボード

iOS のソフトウェア QWERTY キーボードなど , 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広く一般的に使用されている . しかしながら , 超小型タッチパネル上においては , 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隠れてしまう . 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 , 従来の手法ではユーザの QWERTY キーボードにおける空間的記憶に強く依存する . そこで , ZShift は Shift [11] を適用し , 押下している位置を画面上部にフキダシ状にて表示することでこの問題を解決している . しかしながら , 特に小さいキーボードにおいては , 単に押下している位置を画面上部にフキダシ表示するだけではキーが小さく表示されてしまい見づらさを伴う . そこで ZShift では , Shift にズームを加え ( Zoomed Shift technique, ZShift ) , かつ押下しているキーをハイラ

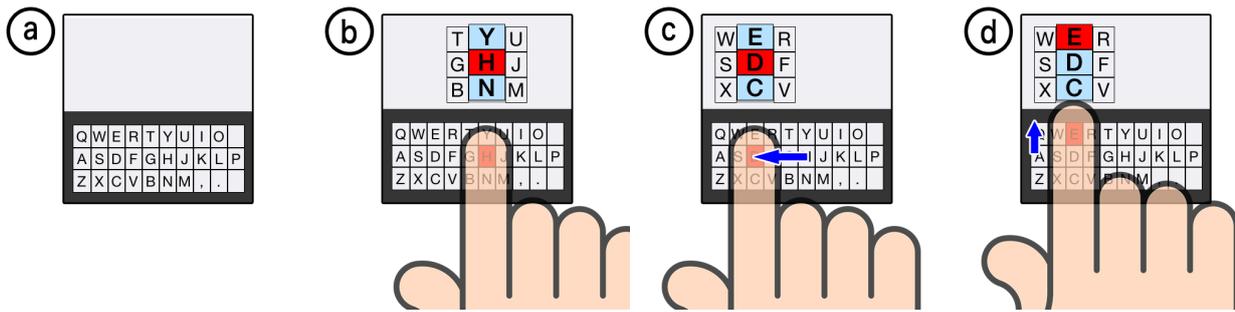


図 2: Flickey における文字入力フロー . a) 初期状態 , b) タッチダウン , c) 横移動によりキー列を選択 , d) そのままタッチアップ , または上下へフリックして文字を入力

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

イト表示することによってフキダシ表示を改善している (図 1c).

### 3.3 Flickey キーボード

Flickey は ZShift [6] に基づき、キーボード上での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。フリック操作を活用することによりキーを押し分けずに済むので、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。Flickey における文字入力のフローを図 2 に示す。Flickey は 2 段階の入力操作によって文字入力を行う手法である。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、フリックアップまたはフリックダウン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 Y 軸方向の動きを気にする必要が無い。その後キーの上中下段の選択もフリック操作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、キーを押し分ける負担を軽減させることができる。また、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 1d)。

### 3.4 共通の機能

ZoomBoard および ZShift はボタンの代わりにスワイプジェスチャによる削除および空白入力が実装されているため、Flickey にも実装した。キーボード上で右スワイプを行うことで空白を入力することができる。キーボード上で左スワイプを行うことで直前の入力文字を削除することが

できる。

## 4. 評価実験

Flickey, ZoomBoard および ZShift の性能を比較するため、文字入力を行う実験を実施した。プレスレット型の端末から、時計型の端末まで様々なウェアラブル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)。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 [6] 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 ずつ拡大 (1.33... 倍)) を用いて決定した。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。したがって、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。

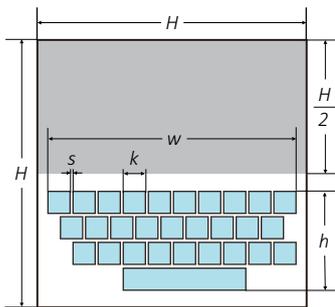
### 4.1 実験機器

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, 4inch, 326ppi) 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り、被験者の腕の太さに柔軟に対応できる。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おり、スマートフォンを腕に固定する用途に使用できる。

キーボードを iOS 端末上にてネイティブ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 (単位は全て mm)。



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	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 our experiment.

medium, large のキーボードはそれぞれ small を基準に 3 分の 4 倍 (1.33... 倍) ずつ拡大したサイズに設定した。すべてのキーボードは、一般的なスマートフォン向けソフトウェアキーボードと比べ小さく, small サイズのキーボードは iPhone 6 (4.7inch) で用いられている QWERTY キーボードと比較して約 20 分の 1 の面積である (図 1a)。

実験の様子を, iPhone のログ出力, iPhone のスクリーンキャプチャおよびボイスレコーダを用いて記録した。

#### 4.2 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), 利き腕は右手だった。スマートフォンにおける文字入力は 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。タッチパネル端末における日常的な英語文字入力は, 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



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

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 [12] 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無かった。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 1,640 円の謝礼を支払った。

#### 4.3 手順

実験は静かな室内で行われた。被験者は実験中, 常に着席姿勢で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 (図 5) を測定した。平均サイズは 14.3 mm (SD = 0.8 mm) であった。測定完了後, スマートフォンをバンド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。このウォームアップセッションは平均 6 分間を要した。

実験者は被験者に, 各手法および各サイズに対して 5 フレーズの文字入力を求め, これを 1 セッションとした。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, 各手法間における文字入力の入力順が実験結果に影響を及ぼす恐れがあるため, カウンターバランスをとるために, ランダムにした。また,

実験用の文字セットとして, MacKenzie ら [13] によって提供されている 500 フレーズの英文を用い, 入力するフレーズはそこからランダムに提示した. これらのフレーズには句読点および数字は含まれておらず, すべて小文字である. フレーズ内の単語と単語の間は空白文字で区切られている. なお, 学習効果を抑えるため各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が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」ボタンを押すことで, 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するように指示した. 全てのキーボードにおいて, 右スワイプをすることによって空白文字を入力することができる. 入力文字を間違えた場合は, 被験者はすべてのキーボードに対して, キーボード上に左スワイプをすることによって入力された最後の文字を削除することができる. 実験中の文字入力操作はすべて利き腕の人差し指で行うように指示した. また, 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. このタスク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 Service Usability Scale (SUS) [14,15] および NASA Task Load Index (TLX) [16] に回答するように求めた. SUS はシステムの使いやすさを評価するためのアンケートである. 被験者は 10 項目の質問に対し, 5 段階のリッカート尺度によって回答する. SUS の質問数は 10 項目と少なく被験者の負担が小さい. また, 10 名前後の少ない回答者からでも良好な結果が得られる等の統計的性質も明らかにされている. 一方で, SUS は英語で記述されているため, 今回は古井ら [17] によって使用された日本語版のものを使用した. なお, 一部の質問文に原文の意図が反映しき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)). 表 1 に今回用いた SUS の質問文を示す. 今回用いた NASA-TLX は, 三宅ら [18] によって日本語に翻訳されたものを使用した. また, 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回答が終了するまでに 120 分間程度の時間を要した.

表 1: SUS の原文と日本語訳  
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ると感じた .
(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 .

## 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一分あたりの文字入力速度 (Words Per Minute, WPM) を用いた. WPM は入力速度を測定する用途で広く一般的に使用されている指標である. 標準化のため WPM では, 5 文字を 1 ワードとして扱う (空白を含む).

表 2 および図 6 に各条件ごとの文字入力速度結果を示す. WPM の平均値はキーボード列に示しており, 括弧内は標準偏差を示す. WPM 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, 全てのサイズ条件に対して有意な差があった. 全てのサイズ条件に対して多重比較検定 (HSD 検定) を行った. small 条件においては, ZoomBoard - ZShift 間 ( $p < .05$ ) および ZoomBoard - Flickey 間 ( $p < .01$ ) に有意な差があった. medium 条件においては, ZoomBoard - ZShift 間に有意な差があった ( $p < .001$ ). 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

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

サイズ	分散分析			キーボード		
	$F_{2,12}$	$p$	$\eta^2$	ZoomBoard	ZShift	Flickey
small	8.2	.006	0.58	7.5 (0.3)	8.5 (0.7)	8.7 (0.3)
medium	13.1	.001	0.69	8.7 (0.5)	10.1 (0.4)	9.4 (0.5)
large	58.1	.000	0.91	8.9 (0.2)	12.4 (0.5)	8.8 (0.9)

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

サイズ	分散分析			キーボード		
	$F_{2,12}$	$p$	$\eta^2$	ZoomBoard	ZShift	Flickey
small	15.82	.0004	0.73	0.00 (0.00)	0.55 (0.31)	0.00 (0.00)
medium	0.64	.543	0.10	0.64 (1.42)	0.41 (0.64)	0.00 (0.00)
large	0.69	.519	0.10	0.00 (0.00)	0.29 (0.64)	0.26 (0.36)

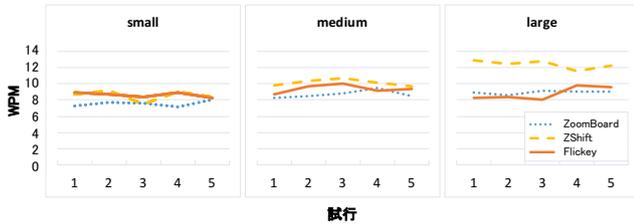


図 6: 文字入力速度 (WPM)  
Fig. 6 Text entry speed (WPM).

た．結果として，small 条件では ZoomBoard と比べ ZShift および Flickey が有意に速く，medium 条件では ZoomBoard と比べ ZShift が有意に速く，large 条件では 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ことで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．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．その理由として，他の手法はサイズが変化しても使用感はあまり変化しないのに対し，Flickey は前述のようなフリック判定しきい値が変化するなどサイズ変化による使用感の変化が生じるため，このような結果になったと考えられる．

#### エラー率

今回我々は文字入力速度の指標として 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 [6]．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で正規化される．

表 3 に CER を示す．CER の平均値はキーボード列に示しており，括弧内は標準偏差を示す．なお，単位は百分率である．CER の値は低い方がよい結果と言える．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．small 条件において

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

サイズ	分散分析			キーボード		
	$F_{2,12}$	$p$	$\eta^2$	ZoomBoard	ZShift	Flickey
small	2.45	.128	0.29	4.59 (3.28)	5.53 (1.94)	9.47 (5.15)
medium	2.91	.093	0.33	4.70 (3.68)	8.95 (3.66)	10.01 (3.71)
large	5.26	.022	0.47	2.01 (1.89)	6.52 (4.99)	9.48 (3.44)

多重比較検定 (HSD 検定) を行ったところ，ZShift - Flickey 間 ( $p < .001$ ) および ZShift - ZoomBoard 間 ( $p < .001$ ) に有意な差があり，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 が有意に悪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した割合も求めた．表 4 に Cerr を示す．Cerr の平均値はキーボード列に示しており，括弧内は標準偏差を示す．なお，単位は百分率である．Cerr の値は低い方がよい結果と言える．分散分析を行ったところ，large 条件においてのみ有意な差があった．large 条件において多重比較検定 (HSD 検定) を行ったところ，ZoomBoard - Flickey 間に有意な差があり ( $p < .05$ )，結果として Flickey は ZoomBoard に比べ large 条件においては Cerr が有意に悪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．UCerr の平均値はキーボード列に示しており，括弧内は標準偏差を示す．なお，単位は百分率である．UCerr の値は低い方がよい結果と言える．分散分析を行ったところ，small 条件においてのみ有意な差があった．small 条件において多重比較検定 (HSD 検定) を行ったところ，ZShift - Flickey 間 ( $p < .001$ ) および ZShift - ZoomBoard 間 ( $p < .01$ ) に有意な差があり，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 が有意に悪いという結果が得られた．

Flickey において，どのようなエラーをしたかを分析したところ，目的のキー列とは 1 つずれたキー列を選択している事象が多く見られた．横方向ドラッグによりキー列を選択している時に，誤って移動しすぎて

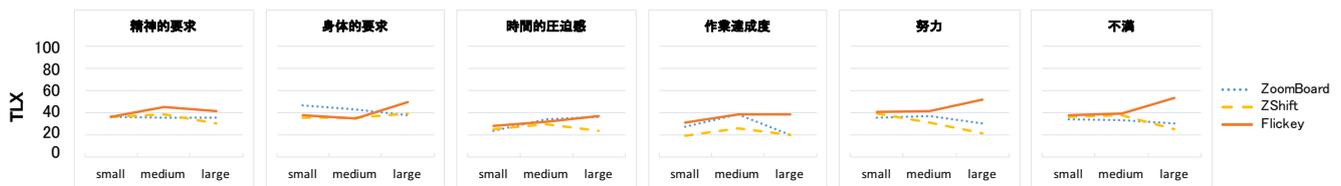


図 7: 6 つの TLX 評価尺度項目の結果

Fig. 7 Results of each of the 6 TLX dimensions.

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

サイズ	分散分析			キーボード		
	$F_{2,12}$	$p$	$\eta^2$	ZoomBoard	ZShift	Flickey
small	15.61	.0004	0.72	0.00 (0.00)	0.47 (0.27)	0.00 (0.00)
medium	0.64	.543	0.10	0.64 (1.44)	0.39 (0.62)	0.00 (0.00)
large	0.69	.519	0.10	0.00 (0.00)	0.26 (0.59)	0.23 (0.32)

表 6: 各条件のユーザビリティ (SUS)  
Table 6 Usability of each conditions (SUS).

サイズ	分散分析			キーボード		
	$F_{2,12}$	$p$	$\eta^2$	ZoomBoard	ZShift	Flickey
small	0.37	.696	0.06	66.4 (13.3)	60.0 (15.2)	59.0 (12.9)
medium	0.91	.427	0.13	72.0 (11.4)	69.5 (14.1)	62.0 (10.8)
large	5.44	.021	0.48	69.5 (17.0)	81.5 (11.8)	54.0 (9.8)

しまい目的のキー列とは別のキー列を選択していた。この原因としてフキダシ表示内の視覚的フィードバックが離散的だったからだと考えられる。ZShift では指のドラッグに応じてフキダシの位置やフキダシ内の表示が連続的に変化するため、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 のフキダシ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。結果として、誤入力につながったと考えられる。この問題の解決策として、Flickey のフキダシ表示をユーザのドラッグ操作に応じて連続的に変化させることが挙げられる。

表 7: 各条件のメンタルワークロード (TLX)  
Table 7 Workload of each conditions (TLX).

サイズ	分散分析			キーボード		
	$F_{2,12}$	$p$	$\eta^2$	ZoomBoard	ZShift	Flickey
small	0.07	.937	0.01	36.4 (24.9)	32.5 (19.7)	37.3 (21.3)
medium	0.14	.872	0.02	40.2 (22.1)	34.7 (25.2)	41.9 (19.8)
large	1.46	.271	0.20	33.7 (13.7)	27.6 (24.1)	49.0 (21.9)

## 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。SUS の平均値はキーボード列に示しており、括弧内は標準偏差を示す。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 条件では有意な差が無かったが、large 条件では有意な差があった。large 条件において多重比較検定 (HSD 検定) を行ったところ、ZShift - Flickey 間に有意な差 ( $p < .05$ ) があり、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表 7 および図 7 に示す。TLX の平均値はキーボード列に示しており、括弧内は標準偏差を示す。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、全てのサイズ条件で有意な差は無かった。

ZShift に関しては、サイズが大きくなるにつれて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ソフトウェア QWERTY キーボードに操作感が近くなっていくためであると考えられる。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、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は「毎回 2 度タップするため操作が少し煩わしい」等の意見が得られた。Flickey は large 条件において small 条件と比べ悪い評価がでている。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

我々はフリック操作を活用することにより、キーの押し分けの負担が少ないソフトウェア 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 条件において、有意差はなかったが、Flickey の入力速度が最も速かった (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 はフリック操作を用いるため、キーボードサイズが変化した場合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してしまう。各サイズにおけるしきい値の設定を再考し、違和感の少ないしきい値に調整する必要がある。さら

に、Flicky はフキダシ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。この問題の解決策として、例えばフキダシをドラムロール状に表示し、ユーザのドラッグ操作に応じて連続的に変化させることが挙げられる。以上の改善点を実装することで、入力速度、エラー率、ユーザビリティおよびメンタルワークロードの値が改善することが予測される。今後はこれらの改善点を実装し、さらなる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 SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pp. 2799–2802, New York, NY, USA (2013), ACM.
- [3] 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.
- [4] 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.
- [5] 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, *インタラクシオン* 2014, pp. 317–320, 一般社団法人情報処理学会 (2014).
- [6] 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.
- [7] Graffiti, <http://www.palm.com/products/input/>.
- [8] 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).
- [9] 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.
- [10] 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, *インタラクシオン* 2013, pp. 72–79, 一般社団法人情報処理学会 (2013).
- [11] Vogel, D. and Baudisch, P.: Shift: A Technique for Operating Pen-based Interfaces using Touch, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pp. 657–666, New York, NY, USA (2007), ACM.
- [12] 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/](http://www.justsystems.com/jp/products/atok_android/) (2015).
- [13] 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.
- [14] Brooke, J.: SUS: A Quick and Dirty Usability Scale (1996).
- [15] Brooke, J.: SUS: A Retrospective, *Journal of Usability Studies*, Vol. 8, No. 2 (2013).
- [16] 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).
- [17] 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).
- [18] 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).