

InvisibleFlick: 小型タッチスクリーン端末における キートップが透明な日本語入力キーボード

鳥山 らいか¹ 宮下 芳明¹

概要: スマートウォッチのような小型タッチスクリーン端末（以降、小型端末）で日本語入力を行う際、選択肢の1つとしてソフトウェアキーボードを用いる方法がある。しかし、現状のソフトウェアキーボードでは、キーやテキストエリアを1つの平面上に配置するため、各領域が狭くなってしまい、ミスタップの頻発による文字入力速度の低下を招いている。本研究では、小型端末における、フリック操作に基づく、キートップが透明な日本語入力キーボード「InvisibleFlick」を提案する。日本語入力手法として一般的なフリックキーボードと同様に、InvisibleFlickでは、テンキー配列のキーボード上でのフリック操作により文字を入力する。テキストエリアをテンキー部分の背景とし、テンキー部分のキートップを透明にすることによって、入力中のテキストを明瞭に視認することができる。フリックキーボードとテキストエリアの2層のレイヤ構造の導入により、キーとテキストエリアを同時に拡大することができる。InvisibleFlickの性能を評価するため、プロトタイプを作成し、既存手法との比較実験を行った。結果、InvisibleFlickの使用を開始してから20-30分後の、最終セッションにおける文字入力速度は平均163.7CPM（Character Per Minute）に達し、InvisibleFlickは文字入力速度・エラー率ともに既存手法より優れていることが示された。また、印象に関する5段階評価では、使いやすく、今後も使用したいと感じるキーボードであると示された。

1. はじめに

今日、スマートウォッチのような小型タッチスクリーン端末（以降、小型端末）が多く普及している。小型端末への文字入力には、主に音声入力や搭載されているタッチスクリーンでの入力によって行われる。しかし、音声入力は同音異義語の識別や騒音下での認識率の低下、プライバシーなどの面で問題がある。また、ソフトウェアキーボードは、1つの平面上にテキストエリアや多くのキーを配置するため、それらのサイズが小さくなっている（図1a）。それにより、指よりも小さいボタンを押し分けることが困難であるというFat Finger問題[1]を引き起こしてしまう。したがって、現在の小型端末の用途は情報提示が主であり、文字入力性能を向上させることは大きな課題である。実用的な文字入力手法が確立されれば、メッセージングアプリでの返信や文章によるメモ書きなど、小型端末の利便性はさらに高まると考えられる。

日本語を入力する場合、アルファベットの入力を対象としたキーボードにおけるローマ字入力を用いることができる。しかし、ひらがな1文字を入力するための平均操作回

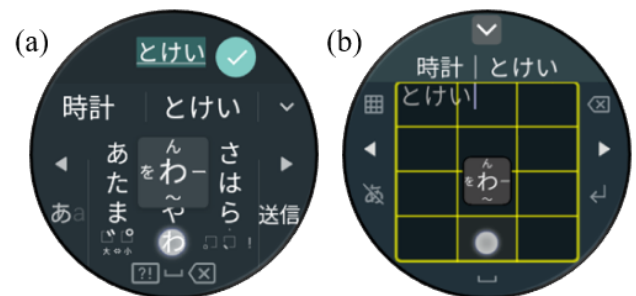


図1 a) Google 日本語入力, b) InvisibleFlick

数が多い上に、キーが小さく、入力ミスが頻発するという問題がある。そのため、一般的に図2のような、テンキー配列のフリックキーボード（以降、フリックキーボード）が用いられる。フリックキーボードでは、タップ位置で子音、フリック方向で母音を指定して文字入力を行う。フリック操作により、キー押下を繰り返すグル入力よりも軽負担で素早い操作を行うことができる。

そこで本稿では、フリック操作に基づく、キートップが透明な日本語入力キーボード「InvisibleFlick」を提案する（図1b）。日本語入力手法として一般的なフリックキーボードと同様に、InvisibleFlickでは、テンキー配列のキーボードにおけるフリック操作を採用した。テキストエリアをテ

¹ 明治大学
Meiji University



図 2 フリックキーボードの例. 「り」と入力する場合, 「ら」行のキーにタッチダウンし, 左方向にフリックする.

ンキー部分の背景とし, テンキー部分のキートップを透明にすることで, フリックキーボードとテキストエリアの 2 層のレイヤ構造を実現する. これにより, キーとテキストエリアを同時に拡大することができるため, 文字入力速度の向上とエラー率の低減が期待できる. InvisibleFlick の性能を評価するため, 一般的なフリックキーボードとの比較実験を行い, 文字入力速度, エラー率, および印象評価の分析を行った.

2. 関連研究

InvisibleFlick は一部が透明な UI^{*1}を用いて, 小型端末における速く正確な文字入力を目指すものである. 本章では, 透明・半透明な UI や小型端末における文字入力手法に関する研究を示す.

2.1 透明・半透明な UI

クラゲ日本語入力 [2] では, テンキー配列のフリックキーボードを半透明にすることで背景を視認可能にした. Invisiboard [3] では画面を 3 × 3 の透明な領域に分割し, タッチジェスチャと組み合わせることによって, アルファベットを入力する. Zhu ら [4] はスマートフォンにおいて, 空間モデルを利用した透明な QWERTY キーボードを提案した. ユーザがキーボードの配列を覚えていれば入力は可能であり, 少ない練習で通常の QWERTY キーボードに近い文字入力速度が出せることを示した. Kim ら [5] はタブレット端末において, 半透明な QWERTY キーボードのデザインがパフォーマンスに与える影響を調査した.

2.2 ソフトウェアキーボード

2.2.1 アルファベットの入力

QWERTY 配列を採用し, 1 文字を複数段階に分けて入力する手法が提案されている. ZoomBoard [6] はタップ操作で QWERTY キーボードを段階的に拡大する. Swipeboard [7] は QWERTY キーボードのキーを 9 つのグループに分け, スワイプによって段階的に決定する.

同様に QWERTY 配列を採用しているが, 1 文字を 1

段階で入力する手法もある. SplitBoard [8] は QWERTY キーボードを左右に分割し, スワイプによって切り替える. FlickKey Mini Keyboard [9] や Flit Keyboard [10] では, 1 つのキーに 9 個のアルファベットを割り当て, 8 方向にフリックすることでキーを押し分ける. Flickey [11] は指が触れた周囲を拡大して, 指の移動やフリックで文字を選択する. DriftBoard [12] はキーをタッチするのではなく, 画面上的小さなキーボード自体を指で動かし, 固定されたカーソルに合わせて入力する. ForceBoard [13] では隣り合う 2 つのキーをタップの圧力で押し分ける. DiaQwerty [14] では様々な形状の画面に対応可能な, アスペクト比が 10:7 のキーボードを提案した. Leiva ら [15] はタッチしている指の下に隠れているキーを拡大し, ポップアップ表示する手法を提案した.

QWERTY 配列を採用しているが, 1 単語を 1 段階で入力する手法もある. WatchWriter [16] は縮小表示された QWERTY キーボードの上でキーをなぞるジェスチャを行い, 統計モデルを利用して指の動きから英単語を推定する. 辞書に登録された単語の中から, ジェスチャによって入力された確率が最も高い単語を推定するため, 指の位置が正確でなくても単語を入力できる.

QWERTY 以外の配列を採用した手法も提案されている. SwipeKey [17] では, アルファベットを順に 8 つの領域に分け, スワイプの方向で押し分ける. ETAO Keyboard [18] はアルファベットを順に 3 つの領域に分け, タップ操作で段階的に選択する. 初期画面にはよく使用される文字が表示されている. 1 行で構成されたキーボードには, 画面の周囲に配置した 8 つのキーを使う TouchOne Keyboard [19] や Minuum Keyboard [20] がある. これらの手法では, 1 つのキーに 3-4 個のアルファベットを割り当てており, 辞書を用いて単語を特定する. 辞書に載っていない単語は, 1 文字ずつ入力する必要がある.

2.2.2 日本語の入力

Wear OS by Google^{*2}を搭載する端末では, テンキー配列のフリックキーボードである Google 日本語入力 [21] を使用できる (図 1a). テンキー上でタップ, もしくは上下左右の 4 方向にフリックを行うことで文字を入力できる. テンキー配列以外の配列を採用した手法も提案されている. 秋田ら [22] はスライドイン (画面外から画面内に指を滑らせる動作) を使用し, 段階的に子音や母音を入力する手法を提案した. BubbleSlide [23] は子音キーから画面中央方向にスワイプし, その長さによって母音を押し分ける. HARI キーボード [24] は子音キーを押して指を画面中央部までスライドし, 母音キー上で離すことで入力を行う. これらはいずれも画面周囲にキーを配置しているが, その位置や大きさには差がある. ShuttleBoard [25] は 10 個の子

^{*1} User Interface

^{*2} <https://wearos.google.com/>

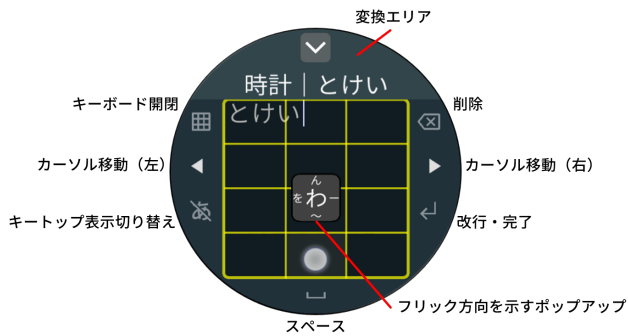


図 3 InvisibleFlick の各部の名称と機能

音キーと 5 個の母音キーで構成され、子音から母音へキーをなぞって入力する。指を離すことなく繰り返すことで連続して文字を入力できる。

2.3 タッチ操作によるアイズフリー入力

画面を見ることなく（アイズフリー）、タッチ操作によって文字入力を行う手法が提案されている。下岡ら [26] は、テンキー配列を採用し、2 本指でフリックする方向によって子音と母音を段階的に押し分けて入力する手法を提案した。井川ら [27] は、テンキー配列を採用し、1 本指でフリックする方向によって子音と母音を段階的に押し分けて入力する手法を提案した。

アイズフリーの手書き入力に関しても提案がされている。植野ら [28] は画面内を縁に沿ってなぞる操作によって文字を入力する手法を提案した。金井ら [29] は文字をストロークごとに分割して手書き入力する手法を提案した。

3. InvisibleFlick

InvisibleFlick は、テンキー部分のキートップが透明な日本語入力キーボードである。日本語入力手法として一般的なフリックキーボードと同様に、テンキー配列のキーボード上でのフリック操作により文字を入力する。各部の名称と機能について、図 3 に示す。テキストエリアをテンキー部分の背景とし、テンキー部分のキートップを透明^{*3}にすることによって、入力中のテキストを明瞭に視認することができる。フリックキーボードとテキストエリアの 2 層のレイヤ構造の導入により、キーとテキストエリアを同時に拡大することができるため、文字入力速度の向上とエラー率の低減が期待できる。

キートップが完全に透明であるため、半透明なキーボード [2, 5] と比較して、より明瞭にテキストを視認することができる。また、既存のフリックキーボード (図??) では、テキストエリアに表示できる文字数は全角で 6 文字 (6 文字 × 1 行) であるのに対し、InvisibleFlick では、文字サイズが同一の場合、全角で 49 文字 (7 文字 × 7 行) 表示することができる。

^{*3} 境界線は半透明な線で描画

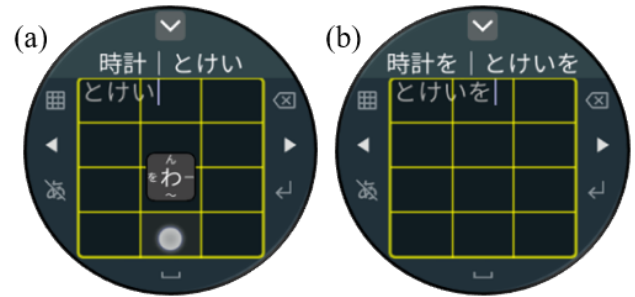


図 4 「を」を入力する例。a) 「わ」行のキーでタッチダウンし、左方向にフリック。b) テキストエリアに「を」が入力される。

3.1 ひらがなの入力とキー操作

ユーザはテンキー部分の上でフリック入力を行う。テンキー部分のキー配置に関しては既存手法 [21] を参考に、一般的なものと同様にした。各キーを押したとき、フリックする方向によって入力される文字をポップアップ表示する。削除キーをタップすることで 1 文字削除、長押しで文字の連続削除を行う。左右のカーソル移動キーでカーソルを移動し、入力位置を変更する。テキストエリアをスクロールする場合は、キーボード開閉ボタンを押してキーボードを閉じ、背景のテキストエリアを操作する。キーボードが閉じた状態で文章をタップすると、タップした場所へカーソルが移動する。改行・完了キーをタップすると、文章に改行を挿入し、長押しで入力を完了する。

3.2 変換機能

日本語にはひらがなの他にカタカナや漢字などの多くの種類の文字が存在する。それらに対応するため、日本語入力キーボードには変換機能が必須である。Google 日本語入力 [21] では、テキストエリアとキーボードの間の領域に 1 行の変換候補選択領域（以降、変換エリア）を設けている（図??）。変換候補が多くある場合に変換エリアの右端にある変換エリア拡大ボタンをタップすることで、変換エリアを展開し、2 行目以降を表示することができる。変換する可能性が高い順に候補が左上から敷き詰められていくため、1 行表示状態の変換エリアを活用することで、より速く文字を変換することができる。

同様に、InvisibleFlick でも 1 行の変換エリアと、拡大ボタンによる 2 行目以降の表示機能を実装した（図 5）。変換候補の提示には、Google 日本語変換 CGI API^{*4}を利用した。変換を行うには、はじめに、ひらがなの入力を行う。入力されるごとに、キーボード上部の 1 行の変換エリアに入力したひらがなの変換候補が表示される（図 5a）。変換する候補が存在していれば、その候補をタップして確定する。変換する候補が存在していなければ、最上部にある変換エリア拡大ボタンをタップすることで、変換エリアを展開し、2 行目以降を表示する（図 5b）。そこで、変換

^{*4} <https://www.google.co.jp/ime/cgiapi.html>

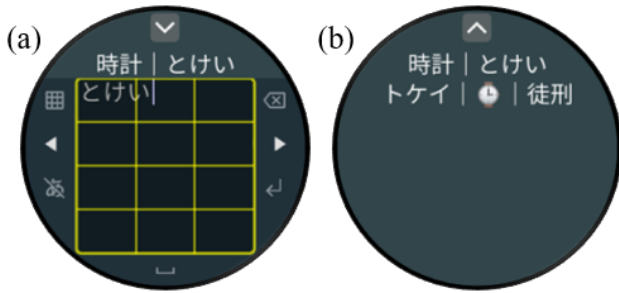


図 5 「とけい」を変換する例. a) 1 行の変換エリア. 最上部の拡大ボタンをタップすると b へ遷移. b) 変換エリアが拡大され, 2 行目以降が表示される.

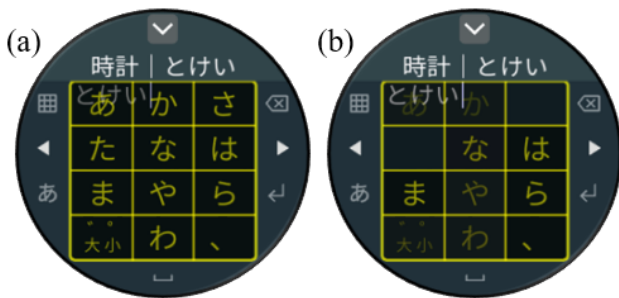


図 6 段階的な透明化. a) 初期はキーがしっかり見える程度の透明度でキートップを表示. b) 入力ミスの頻度が少ないキーは透明度を上げていき, 最終的に完全に透明な状態になる.

したい候補をタップして確定する.

3.3 段階的な透明化

InvisibleFlick では, テンキー部分のキートップが透明であるため, テンキー配列に慣れていないユーザがすぐに使いこなすことは難しいと予想される. そのため, テンキー配列の学習を支援する方法として, 各キーを段階的に透明化させていくことができる (図 6). はじめにキートップを半透明で表示しておき, キーごとの入力ミスの頻度に応じて, 完全な透明に近づけていく. 完全にキートップが透明になった場合でも, 入力ミスの頻度が増えれば, キートップの透明度が下がり, 視認できる状態に戻る. これにより, テンキー配列に慣れていないユーザも, 使い続けていくことで徐々に配列を覚え, キートップが透明でも入力ができるようになることを期待できる.

4. 評価実験

InvisibleFlick の性能と操作性を検証するために, 既存手法であるフリックキーボードとの比較実験を実施した. フリックキーボードとは, スマートウォッチにおける日本語入力キーボードで一般的に用いられる手法である. キートップが透明であることやキーサイズの拡大の影響を調べるため, ひらがなのフレーズを入力するタスクを行った. スマートフォン上にスマートウォッチを模したキーボードを実装し, これを実験に用いた.



図 7 実験機器 (左) と実験風景 (右)

4.1 参加者

本実験では大学生・大学院生 10 名が参加した. 男性 7 名, 女性 3 名で, 年齢は 22-24 歳 (平均年齢 23.4 歳) であった. 右利きが 9 名, 左利きが 1 名で, 全員が左腕に時計をつけると答えた. スマートウォッチの使用経験者は 4 名であり, スマートウォッチ上でのソフトウェアキーボードによる日本語入力経験者はいなかった. また, 全員がスマートフォンを使用しており, 使用歴は 6-12 年で, 日本語入力を行う際に 8 名がテンキー配列のフリックキーボードを使用しており, うち 1 名はトグル入力のみを使用していた. テンキー配列のフリックキーボードを使用していない 2 名は QWERTY 配列のキーボードを使用していた.

4.2 実験機器

スマートフォンは LG 社製 Nexus5X (LG-H791, Android v8.1.0) を使用し, ディスプレイ解像度は 1920×1080 pixels ($115.3 \text{ mm} \times 64.9 \text{ mm}$, 423 ppi) であった. 実験システムとなるアプリケーションは Processing^{*5} v3.3.4 (Processing for Android v3.0.1) によって実装した. スマートフォンは, 2 本のマジックテープを用いて, 参加者の左腕に横向きに取り付けた (図 7 左).

4.3 実験システム

各手法の画面を図 8 に示す. スマートウォッチを模した円形の領域 (以降, 円形領域) 内に既存手法と InvisibleFlick を実装した. 円形領域の直径は 356 pixel (35.6 mm) であり, スマートウォッチの本体サイズ^{*6}として一般的な 42mm の端末に搭載されている, タッチスクリーンと同程度のサイズとした.

既存手法のレイアウトは WearOS 向けに提供されている Google 日本語入力 [21] のフリックキーボードと同様にした. 円形領域のすぐ上にはひらがなの提示フレーズの表示領域があり, 漢字表記もその上に表示した.

4.4 フレーズセット

提示する文字として, 5-6 文字のひらがなで構成されるフレーズを 144 個用意した. これらのフレーズを 6 つまと

^{*5} <https://processing.org/>

^{*6} 画面とベゼル (縁) を含むケース部分の幅

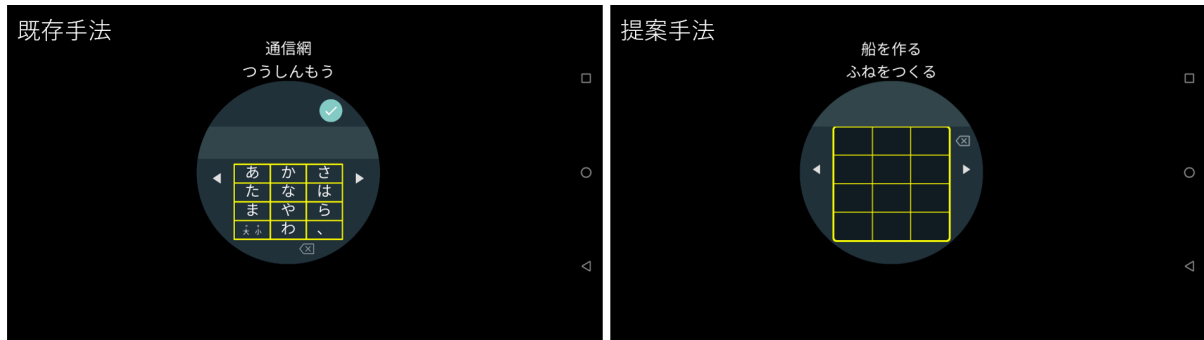


図 8 実験システムの画面

めたものを1つのフレーズセットとして、24個のフレーズセットに分ける。濁音・半濁音・小文字の数、および合計文字数は、各フレーズセットで統一した。1人の実験参加者に対して各フレーズセットは一度しか提示しない。

4.5 タスク

参加者は画面上部の提示フレーズを入力する。このとき、入力された文字が提示フレーズと完全一致するまで修正を行う。一致すると、自動で次のフレーズが表示される。参加者はこれを繰り返す。1フレーズセット分(6フレーズ)の入力を行うことを1セッションと定義する。

4.6 手順

実験は静かな室内で行われた。疲労や慣れによる手法ごとに偏りが出るのを防ぐため、参加者を2つのグループに分け、使用する入力手法の順番を変更した。

まず、参加者は実験の目的および内容、タスクの説明を受けた。その後、ノギスを用いて左手の人差し指の幅^{*7}を測定した。平均サイズは14.9mm ($SD=1.4\text{mm}$)であり、日本人の標準的なサイズの範囲内であった[30]。測定後、マジックテープを用いてスマートフォンを左手に装着した。

次に、参加者はフリック入力と2手法すべての入力方法の説明を約10分間受けた。このとき、テンキー配列およびフリックする方向と入力される文字の対応を記した図を用いた。

次に、文字入力手法の1つ目で5分間練習を行った。入力する文字は自由であり、テンキー配列の図を確認することができた。また、腕の疲労を抑えるため、練習中は腕を机などに乗せることを許可した。1分間休憩した後、練習した手法での入力を12セッション行った。このとき、テンキー配列の図を確認すること、腕を机などに乗せることは禁止した。また、最終セッションをのぞき、3セッションごとに1分間休憩した。

3分間休憩した後、文字入力手法の手法2つ目についても、1つ目と同様に行った。最後にアンケートへの回答を行った。内容はスマートフォンの使用歴等を問う質問、各

手法についての印象に関する5段階評価および、各手法についての感想(自由記述)の3つであった。

参加者はすべての実験を通して144フレーズ入力することになる(6フレーズ×12セッション×2手法)。各手法における操作時間は20-30分程度であり、実験全体でかかった時間は60-90分程度であった。

入力の際には、できる限り正確に、かつ速く入力するように指示した。参加者は両腕を空中に浮かせた状態で入力を行う(図7右)。参加者は自身の疲労度合いにより、休憩時間を短縮することができた。

4.7 評価指標

文字入力速度の評価指標としてCPM(Character Per Minute)を用いる。CPMは1分あたりの文字入力速度を表す。CPMは式1より求める。

$$\text{CPM} = \frac{\text{提示フレーズの文字数}}{\text{入力にかかった時間[分]}} \quad (1)$$

「入力にかかった時間」は、フレーズが提示された後、はじめてタッチしたタイミングから、入力文字が提示フレーズと完全に一致したタイミングまでの時間である。

文字入力速度に加えて、修正済みエラー率(Corrected error rate, Cerr)[31]を測る。Cerrはすべての入力に対する誤入力を修正した割合[%]を表す。Cerrは式2より求める。

$$\text{Cerr} = \frac{\text{修正した文字数}}{\text{入力した文字数}} \times 100 \quad (2)$$

5. 結果・考察

5.1 文字入力速度

各手法の各セッションにおける平均文字入力速度を図9に示す。全セッションにおいて、InvisibleFlickは既存手法より高い文字入力速度を示した。全セッションを通じた平均文字入力速度は、既存手法が111.0CPM、InvisibleFlickが147.6CPMとなった。最終セッションにおける平均文字入力速度は、既存手法が109.3CPM、InvisibleFlickが163.7CPMとなった。全セッションを通じた平均文字入力速度について、対応のあるt検定を行った結果、有意な差

^{*7} 指を伸ばした状態での、指の遠位指関節間関節の最大幅

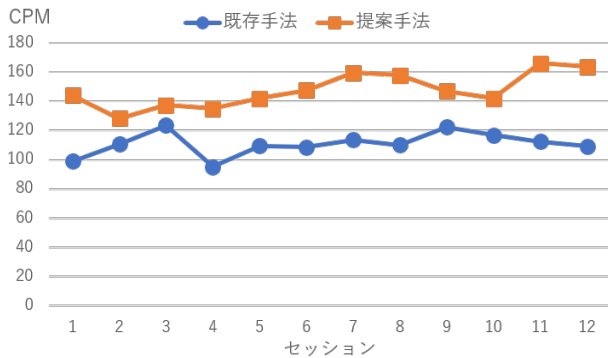


図 9 各手法の各セッションにおける平均文字入力速度 (CPM)

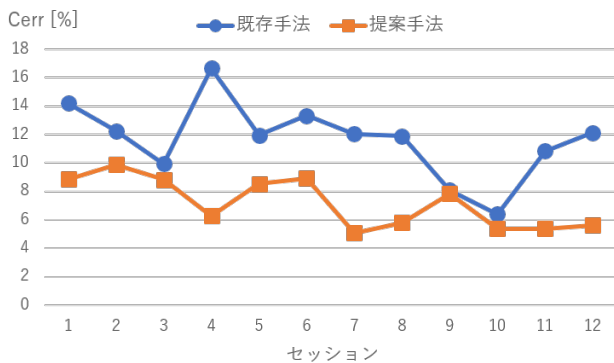


図 10 各手法の各セッションにおける平均修正済みエラー率 (Cerr)

がみられた ($t_9 = -3.41$, $p < .01$).

上記の結果より, InvisibleFlick は既存手法より速く文字を入力することができるといえる. InvisibleFlick は既存手法と比較して, キーサイズが大きく, キーに狙いを定めるまでの時間が短くなり, 操作時間が短くなったためであると考えられる.

5.2 エラー率

各手法の各セッションにおける平均修正済みエラー率 (Cerr) を図 10 に示す. 全セッションにおいて, InvisibleFlick は既存手法より低い Cerr を示した. 全セッションを通した平均 Cerr は, 既存手法が 11.7%, InvisibleFlick が 7.21% となった. 最終セッションにおける平均 Cerr は, 既存手法が 12.2%, InvisibleFlick が 5.64% となった. 最終セッションの Cerr は第 1 セッションと比較して, 既存手法は約 2.1 ポイントの減少, InvisibleFlick は約 3.2 ポイントの減少が見られる. 全セッションを通した平均 Cerr について, 対応のある t 検定を行った結果, 有意な差がみられた ($t_9 = 2.47$, $p < .05$).

上記の結果より, InvisibleFlick は既存手法より入力ミスが少ないといえる. InvisibleFlick は既存手法と比較して, キーサイズが大きく, ミスタップが減ったためであると考えられる.

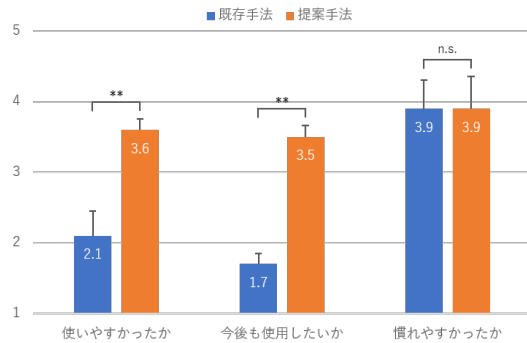


図 11 各手法に対する印象の 5 段階評価の平均値

5.3 印象評価

4.6 節において述べたように, 参加者に対し, 各手法についての印象をリッカート尺度による 5 段階評価を行った. 印象に関する質問項目は以下の 3 つである.

- 使いやすかったか
- 今後も使用したいか
- 慣れやすかったか (慣れるまでの時間が短かったか)

上記の項目に対し, 5: 非常に同意できる / 4: 同意できる / 3: どちらともいえない / 2: 同意できない / 1: 全く同意できない の 5 段階で評価を行った.

各手法に関する 3 つの質問項目に対し, 参加者自身の印象を 5 段階評価した平均値を図 11 に示す (**: $p < .01$, *: $p < .05$, n.s.: 非有意). 数値が高いほど, その質問項目に同意できることを表す. 質問項目の「使いやすかったか」・「今後も使用したいか」について, 対応のある t 検定を行った結果, 有意な差がみられた ($t_9 = -3.50$, $p < .01$, $t_9 = -9.00$, $p < .01$). また, 「慣れやすかったか (慣れるまでの時間が短かったか)」について, 対応のある t 検定を行った結果, 有意な差は認められなかった.

よって, InvisibleFlick は既存手法と比較して, 使いやすく, 今後も使用したいと感じる文字入力手法であるといえる. 一方で, 慣れやすさ (慣れるまでの時間の長さ) については, 各手法間での印象に差はないといえる.

6. 議論

6.1 熟達者が使用した場合の性能

今回の実験では, 熟達者が使用した場合の性能については示していない. そこで, InvisibleFlick を 30 日間使用した著者 1 人を対象として, 4 章の実験における入力タスクを行った. なお, 著者はスマートフォンの使用歴が 7 年 (フリック入力を使用), スマートウォッチの使用歴が 2 年 (フリック入力を使用) である. 結果, 全 12 セッションを通して, 文字入力速度は平均 195.9CPM であり, 修正済みエラー率は平均 0.794% であった. 4 章の実験での最終セッションにおける結果 (平均文字入力速度 163.7CPM,

修正済みエラー率 5.64%) と比較すると、より速く正確に入力ができている。これは、キー配置が正確に記憶できているため、キーの押し間違いによる入力ミスが減ったことによって、修正時間が短縮されたためである。このことから、InvisibleFlick を長期的に使用することによって、文字入力速度は向上し、エラー率は低減すると考えられる。

6.2 テキストエリアの操作方法

現状、背景のテキストエリアを操作するには、テンキー部分を非表示にする必要がある。しかし、キーボードを表示したまま操作できれば、より速い入力が可能となる。Gupta ら [32] は各指を識別することによって、重なった半透明な UI レイヤーを別々に操作するインタフェースを提案している。しかし、この先行研究ではセンサを指に装着する必要がある。テンキー部分を非表示にせずにテキストエリアを操作するには、以下のような方法が考えられる。

- (1) ダイヤル（竜頭）を用いる方法
- (2) Apple Watch^{*8}に搭載されているような圧力検知可能なタッチスクリーンを用いる方法
- (3) テンキー部分の外からスライドインする方法
- (4) 2本の指でマルチタッチジェスチャを行う方法

1 と 2 は現状のタッチ操作をそのまま利用することが可能であるという利点がある。一方、3 と 4 はすべての操作がタッチパネルで完結するという利点がある。

6.3 腕の負担

小型端末におけるソフトウェアキーボードによる文字入力の際、画面サイズ以外の問題として、装着している腕への負担がある。その要因として、以下の3つが挙げられる。

- (1) 腕の外側（手の甲側）に画面が向いているため、タッチ操作するためには手首を画面が見える顔側にひねる必要がある。
- (2) 端末を装着している側の脇を開き、腕を空中に上げたまま維持する必要がある。
- (3) 傾きのある画面でタッチ操作するため、操作する腕を動かし続ける必要がある。

1つ目の問題は、画面が腕の外側ではなく、腕を上げたときに即座に見える位置に移動することで解決できる。図 12 に示す SHIFT^{*9}はランニングやサイクリングの際、瞬時に画面を操作できるよう、画面の位置が手の甲に来るように止めることができるバンドである。これにより、腕を

COMPATIBILITY EXAMPLES



図 12 ランナー向けスマートウォッチ用バンド SHIFT

ひねる必要がなくなり、腕の負担は軽減される。2つ目の問題は、装着している側の肘を身体につけて支えることで、脇を開いて腕を空中に支える必要がなくなるため、腕の負担を軽減できる。3つ目の問題は、Gorilla Arm Syndrome と呼ばれている。タッチする画面が垂直であればあるほど、腕に負担がかかってしまう。これについての根本的な解決策は、現状存在しないが、こちらも2つ目の問題の解決策と同様に、操作している側の肘を身体につけて支えることで、多少は腕の負担を軽減できると考えられる。その他には、予測変換や誤入力修正機能による文字入力手法自体の高速化が解決策であると考えられる。

7. 結論

本稿では、小型端末で日本語入力を行う際に、現状のソフトウェアキーボードの問題点であるテキストエリアやキーの小ささを克服するため、キートップが透明な日本語入力キーボード「InvisibleFlick」を提案した。日本語入力手法として一般的なフリックキーボードと同様に、InvisibleFlick では、テンキー配列のキーボード上でのフリック操作により文字を入力する。テキストエリアをテンキー部分の背景とし、テンキー部分のキートップを透明にすることによって、入力中のテキストを明瞭に視認することができる。フリックキーボードとテキストエリアの2層のレイヤ構造の導入により、キーとテキストエリアを同時に拡大することができる。InvisibleFlick の性能を評価するため、プロトタイプを作成し、既存手法との比較実験を行った。結果、InvisibleFlick の使用を開始してから 20-30 分後の、最終セッションにおける文字入力速度は平均 163.7CPM に達し、InvisibleFlick は文字入力速度・エラー率ともに既存手法より優れていることが示された。また、印象に関する5段階評価では、使いやすく、今後も使用したいと感じるキーボードであると示された。最後に、本稿における議論点について述べた。

^{*8} <https://www.apple.com/apple-watch-series-4/>

^{*9} <https://www.getedgegear.com/>

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, *Proceedings of the 2005 IFIP TC13 International Conference on Human-Computer Interaction*, INTERACT '05, Berlin, Heidelberg, Springer-Verlag, pp. 267–280 (online), DOI: 10.1007/11555261_24 (2005).
- [2] TinyFort: クラゲ日本語入力, <http://kurage.tinyfort.net/>. (Accessed on 01/01/2019).
- [3] Mottelson, A., Larsen, C., Lyderik, M., Strohmeier, P. and Knibbe, J.: InvisiBoard: Maximizing Display and Input Space with a Full Screen Text Entry Method for Smartwatches, *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '16, New York, NY, USA, ACM, pp. 53–59 (online), DOI: 10.1145/2935334.2935360 (2016).
- [4] Zhu, S., Luo, T., Bi, X. and Zhai, S.: Typing on an Invisible Keyboard, *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, New York, NY, USA, ACM, pp. 439:1–439:13 (online), DOI: 10.1145/3173574.3174013 (2018).
- [5] Kim, S. and Lee, G.: Design and Evaluation of Semi-Transparent Keyboards on a Touchscreen Tablet, *Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces*, ISS '18, New York, NY, USA, ACM, pp. 155–166 (online), DOI: 10.1145/3279778.3279801 (2018).
- [6] Oney, S., Harrison, C., Ogan, A. and Wiese, J.: ZoomBoard: A Diminutive Qwerty Soft Keyboard Using Iterative Zooming for Ultra-small Devices, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, New York, NY, USA, ACM, pp. 2799–2802 (online), DOI: 10.1145/2470654.2481387 (2013).
- [7] Chen, X. A., Grossman, T. and Fitzmaurice, G.: Swipeboard: A Text Entry Technique for Ultra-small Interfaces That Supports Novice to Expert Transitions, *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, New York, NY, USA, ACM, pp. 615–620 (online), DOI: 10.1145/2642918.2647354 (2014).
- [8] Hong, J., Heo, S., Isokoski, P. and Lee, G.: 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, New York, NY, USA, ACM, pp. 1233–1236 (online), DOI: 10.1145/2702123.2702273 (2015).
- [9] FLICKKEY LLC: FlickKey Mini, http://www.flickkey.com/FlickKey_Mini.html. (Accessed on 01/01/2019).
- [10] Locnet: Flit Keyboard, <https://sites.google.com/site/flitkeyboard/>. (Accessed on 01/01/2019).
- [11] 石井晃, 箱田博之, 志築文太郎, 田中二郎: 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, インタラクシオン 2016 論文集, インタラクシオン 2016, 情報処理学会, pp. 80–88 (オンライン), 入手先 (<http://www.interaction-ipsj.org/proceedings/2016/>) (2016).
- [12] Shibata, T., Afergan, D., Kong, D., Yuksel, B. F., MacKenzie, I. S. and Jacob, R. J.: 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, New York, NY, USA, ACM, pp. 575–582 (online), DOI: 10.1145/2984511.2984591 (2016).
- [13] Hsiu, M.-C., Huang, D.-Y., Chen, C. A., Lin, Y.-C., Hung, Y.-p., Yang, D.-N. and Chen, M.: ForceBoard: Using Force As Input Technique on Size-limited Soft Keyboard, *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, MobileHCI '16, New York, NY, USA, ACM, pp. 599–604 (online), DOI: 10.1145/2957265.2961827 (2016).
- [14] Kim, S., Ahn, S. and Lee, G.: DiaQwerty: QWERTY Variants to Better Utilize the Screen Area of a Round or Square Smartwatch, *Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces*, ISS '18, New York, NY, USA, ACM, pp. 147–153 (online), DOI: 10.1145/3279778.3279792 (2018).
- [15] Leiva, L. A., Sahami, A., Catala, A., Henze, N. and Schmidt, A.: Text Entry on Tiny QWERTY Soft Keyboards, *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, New York, NY, USA, ACM, pp. 669–678 (online), DOI: 10.1145/2702123.2702388 (2015).
- [16] Gordon, M., Ouyang, T. and Zhai, S.: WatchWriter: Tap and Gesture Typing on a Smartwatch Miniature Keyboard with Statistical Decoding, *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, New York, NY, USA, ACM, pp. 3817–3821 (online), DOI: 10.1145/2858036.2858242 (2016).
- [17] Shao, Y.-F., Chang-Ogimoto, M., Pointner, R., Lin, Y.-C., Wu, C.-T. and Chen, M.: SwipeKey: A Swipe-based Keyboard Design for Smartwatches, *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '16, New York, NY, USA, ACM, pp. 60–71 (online), DOI: 10.1145/2935334.2935336 (2016).
- [18] Rajkumar, Darbara Punyashlok, D. and Debasish, S.: ETAO Keyboard: Text Input Technique on Smartwatches, *Procedia Computer Science*, Vol. 84, pp. 137–141 (online), available from (<https://doi.org/10.1016/j.procs.2016.04.078>) (2016).
- [19] Infiniti Technology: TouchOne Keyboard, <http://www.touchone.net/>. (Accessed on 01/01/2019).
- [20] Whirlscape: Minuum on Smartwatches, <http://minuum.com/minuum-on-smartwatch/>. (Accessed on 01/01/2019).
- [21] Google LLC: Google 日本語入力, <https://www.google.co.jp/ime/>. (Accessed on 01/01/2019).
- [22] 秋田光平, 田中敏光, 佐川雄二: スライドインによるスマートウォッチ向けの文字入力手法, インタラクシオン 2018 論文集, インタラクシオン 2018, 情報処理学会, pp. 276–281 (オンライン), 入手先 (<http://www.interaction-ipsj.org/proceedings/2018/>) (2018).
- [23] 東條貴希, 加藤恒夫, 山本誠: BubbleSlide:フリック操作の規則性を高めたスマートウォッチ向け日本語かな入力インタフェース, インタラクシオン 2018 論文集, インタラクシオン 2018, 情報処理学会, pp. 1048–1053 (オンライン), 入手先 (<http://www.interaction-ipsj.org/proceedings/2018/>) (2018).
- [24] 齋藤航平, 奥寛雅: HARI キーボード: 超小型タッチパネル端末向け日本語入力キーボード, インタラクシオン 2016 論文集, インタラクシオン 2016, 情報処理学会, pp. 701–703 (オンライン), 入手先 (<http://www.interaction-ipsj.org/proceedings/2016/>) (2016).
- [25] 下岡純也, 山名早人: ShuttleBoard: スマートウォッチにおけるタップ動作の少ない仮名文字入力手法, 日本

- データベース学会和文論文誌, Vol. 16-J, No. 5 (オンライン), 入手先 http://dbsj.org/journal/dbsj_journal_j/dbsj_journal_vol_16_05/ (2018).
- [26] 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論文集, WISS2015, 日本ソフトウェア科学会, pp. 145–146 (オンライン), 入手先 <http://www.wiss.org/WISS2015Proceedings/> (2015).
- [27] 井川洋平, 宮下芳明: アイズフリーで速記できる「方向のみ」のフリック入力手法, インタラクション 2013 論文集, インタラクション 2013, 情報処理学会, pp. 651–656 (オンライン), 入手先 <http://www.interaction-ipsj.org/archives/paper2013/> (2013).
- [28] 植野圭一, 郷健太郎, 木下雄一郎: 小型円形タッチスクリーンを対象とした縁なぞり操作による文字入力手法, インタラクション 2016 論文集, インタラクション 2016, 情報処理学会, pp. 473–477 (オンライン), 入手先 <http://www.interaction-ipsj.org/proceedings/2016/> (2016).
- [29] 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論文集, WISS2014, 日本ソフトウェア科学会, pp. 145–146 (オンライン), 入手先 <http://www.wiss.org/WISS2014Proceedings/> (2014).
- [30] 河内まき子: Aist 日本人の手の寸法データ, <https://unit.aist.go.jp/hiri/dhrg/ja/dhdb/hand/>. (Accessed on 01/01/2019).
- [31] Soukoreff, R. W. and MacKenzie, I. S.: Metrics for Text Entry Research: An Evaluation of MSD and KSPC, and a New Unified Error Metric, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, New York, NY, USA, ACM, pp. 113–120 (online), DOI: 10.1145/642611.642632 (2003).
- [32] Gupta, A., Anwar, M. and Balakrishnan, R.: Porous Interfaces for Small Screen Multitasking Using Finger Identification, *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, New York, NY, USA, ACM, pp. 145–156 (online), DOI: 10.1145/2984511.2984557 (2016).