

## 原著論文

# スマートウォッチ向け文字入力手法 SLiT 専用の マイクロタッチボードの開発

坂 香太朗<sup>1)</sup>, 田中 敏光<sup>2)</sup>, 佐川 雄二<sup>2)</sup><sup>1)</sup> 名城大学理工学研究科, <sup>2)</sup> 名城大学理工学部

Development of a micro touch board specialized to SLiT  
that is the character input method for smartwatches.

Kotaro SAKA<sup>1)</sup>, Toshimitsu TANAKA<sup>2)</sup>, Yuji SAGAWA<sup>2)</sup><sup>1)</sup> Faculty of Science and Technology, Meijo University<sup>2)</sup> Graduate School of Science and Technology, Meijo University

**Abstract:** One of the problem in character input on smartwatches, the screen of a smartwatch is hidden by the finger during touch operations. We solve this problem by separating input using the micro touch board. The board is optimized the character input method SLiT. Advantage of SLiT is that the entry speed of the novice is fast and the screen occupancy is low. The speed is 28.7[CPM (Characters Per Minute)] and the occupancy rate is 26.4%. In SLiT, since Japanese hiragana characters are input by combining 12 kinds of strokes and 8 tapping points, the board is designed to detect these strokes and taps only. A circular area is placed at the center of the touch board, and eight touch sensors are arranged so as to surround the circle. Between the sensors, numbers from 1 to 8 are allocated clockwise. Only when two adjacent sensors are touched at the same time, the number assigned to their boundary is entered. The stroke is entered by sliding a fingertip till passing through three boundaries. In computer experiments, the recognition rate of the strokes was 98.7% and the recognition rate of the taps was 99.4%. This touch board is sufficient as the input device.

**Keywords:** smartwatch, character input, dedicated hardware, touch sensor, user interface

**キーワード:** スマートウォッチ, 文字入力, 専用ハード, タッチセンサ, ユーザインタフェース

## 1. はじめに

スマートウォッチの画面はきわめて小さいため、スマートフォンで一般的なフリックキーボードを表示すると、画面の大半がキーボードに占有されてしまう。この対策として、5-TILES[26]等の画面占有率が低いことを特徴とするソフトウェアキーボードが開発されている。我々が開発した SLiT[24][25]もその1つで、画面占有率を26.4%まで低減している。しかし、画面をタッチデバイスとして使う以上、操作する途中で指が画面の一部を隠すことは避けられない。画面が小さいと指に隠れる割合が高くなるため、入力に支障をきたすこともある。

遮蔽の問題を根本的に解決するには、表示デバイスと入力デバイスを分ける必要がある。このために、文字入力手法 SLiT 専用のタッチボードを開発した。このボードは画面の下側のバンド部分に設置され、SLiT が要求するストローク操作とタップ操作をタッチセンサで検出する。本報告では、このタッチボードの詳細を述べるとともに、ボードの性能を評価する。

2017年7月13日受理。

## 2. 関連研究

モバイル環境で利用することを前提とした入力デバイスはいくつも提案されている。富士通の指輪型デバイス[1]や Ring ZERO[2]では、空中に指で文字を描き、手書き文字認識技術を使って漢字仮名交じり文を入力できる。ただし、何もないところに文字を描くため、正しい書体で書くことが難しく、認識精度が下がる。また、腕を大きく動かして文字を描くので、入力が遅くなる。カメラでジェスチャーを認識する手法[3]もあるが、画像処理の負荷が大きい。

手や腕をタッチパッドとして使い、仮想キーボードやジェスチャーにより文字を入力する手法もある。ただし、SkinTrack[4]では位置の検出に指先から皮膚の表面を通して流れる高周波電流を使うため、素肌に触れる必要がある。また、リング状のトランスミッターを指にはめる必要がある。ARmKeypad[5]はAR技術でキーボードを表示するため、シースルー型HMDを装着する必要がある。FingerT9[6]や DigiTouch[7]では指をキーボードとして使うが、このために、人差し指～小指の節にタッチセンサを取り付ける必要がある。Finger touching[8]も指を使うが、こちらは手首に付けたデバイスからキーボードを投影

するタイプである。

TAP[9]は 5本の指にはめて使う板状のデバイスで、タップする指の組み合わせで文字を指定する。ブラインド入力ができるが、タップする面が必要のため、移動中は使える場所が限られる。投影するバーチャルキーボード[10][11]や紙に印刷したキーボードをカメラで読み込むアプリ[12]なども、使用時に平らな面が必要となる。逆に、Ash Smartwatch[13]では、内蔵したプロジェクターで手の甲に画面を表示することで、入力と表示を分離できる。ただし、手首の角度により像が歪む。

手首の傾きを使う方法には、スマートウォッチに内蔵された加速度センサで傾きを検出する Float[14]や加速度と筋電を組み合わせた手法[15]などがある。WristWhirl[16]では手首の動きでジェスチャーを入力する。WrisText[17]はバンドに取り付けたセンサで検出した手首の角度で OneLine keyboard[18]のキーを選択する。これらの手法にはスマートウォッチを付けた腕だけで操作できる利点はあるが、手を動かすと画面も一緒に動いてしまう、手首が疲れるなどの問題がある。

Siri[19]や Cortana[20]に代表される音声認識手法を使えば、スマートウォッチに話しかけることで文を入力できる。ただし、文の修正を音声のみで行うと手間と時間がかかるので、キーボード等の別の入力手段と併用することが前提となる。また、周囲の人に内容を知られてしまう欠点がある。

### 3. 先行研究

スマートフォン専用の文字入力デバイスには、(1)手や指にセンサを付けない、(2)使用する環境の制限が少ない、(3)スマートウォッチ本体またはベルト部分に組み込める、ことが必要だと考えた。指に何か付いていると、物を持ったり作業したりする邪魔になる。移動中に使うことが想定されるので、明るい場所では見づらい、平らな場所を必要とする、などの制限があると実用にならない。すぐに使えることやデバイスの紛失を防ぐ上から、スマートウォッチと一体になっていることが望ましい。

以上の点を考慮して、先行研究[21]では図 1 に示すタッチボードを開発した。ディスプレイの下側に続くバンド部分に埋め込むことができる大きさとして、1辺の長さを 20mm に選んでいる。また、薄く作れるように、単純なタッチセンサで構成されている。ただしプロトタイプはバンドに後付けするので、強度や配線のために一定の厚みがある。

このボードは尾崎[22]らの文字入力手法の入力デバイスとして設計されている。尾崎らの手法では、スマートウォッチの画面の縁をなぞるジェスチャーで行を、タップで段を指定することで平仮名 1 文字を入力する。画面の四隅と各辺の midpoint の合計 8 か所のどこに指先が位置しているかでジェスチャーを判別するため、細かい座標を必要としない。タッチボードではこの 8 か所だけを識別すればよいので、小型化が可能である。ただし、20mm を 3 分割するとセンサの間隔が指先より狭くなる

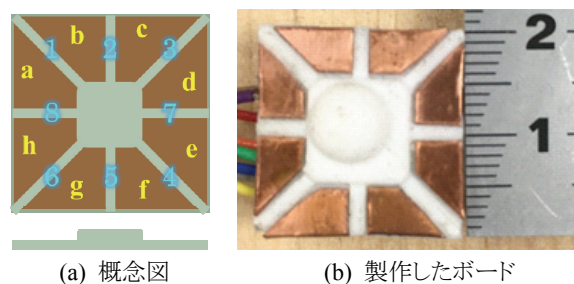


図 1 先行研究のタッチボード

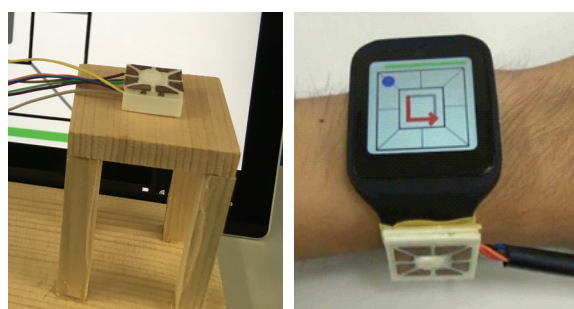


図 2 先行研究のタッチボードの実験状況

ため、タッチミスが増加する。そこで、8 個のタッチセンサ(図 1(a)の a~h)を放射状に配置し、隣り合うセンサに同時に触れたときに、その境界に書かれた数字を出力する。

タッチボード単体を図 2(a)のように台に固定して行った実験では、ストローク認識率 99.4%、タップ認識率 99.7%と高い値が達成されている[21]。しかし、図 2(b)のようにバンドに装着して計測したところ、タップ認識率が 84.2%に低下した[23]。

### 4. 文字入力手法 SliT

専用デバイスから入力する場合でも、表示には画面を使うので、ベースにする文字入力手法には画面占有率が低いことが求められる。また、画面を見ながら操作するので、表示が見やすいことも必要である。長文の入力には画面が大きいスマートフォンやタブレットが適しているので、スマートウォッチでは比較的短い文を少数入力することが予想される。このため、利用機会が少なくても容易に使えることも求められる。尾崎らの手法はこれらの条件を満たしているが、その後、さらに画面占有率が低く、入力が速い文字入力手法 SliT[24][25]が開発されている。

SliT も行・段の順で平仮名を指定する。図 3(a)は行選択の初期画面で、画面の左辺、上辺、右辺がそれぞれ 2 分割され、1つの区画に 2 つずつの行が割り当てられている。その 1 つをスライドインで選択する。スライドインとは、画面の外にタッチした指を内側に向かって滑らせる動作である。スライドインが画面の縁を横切ると、(b)のように画面が 2 分割され、横切った位置に割り当てられた行がそれぞれに表示される。指をどちらか一方の区画に移動すると、(c)のように背景が緑に変わる。この

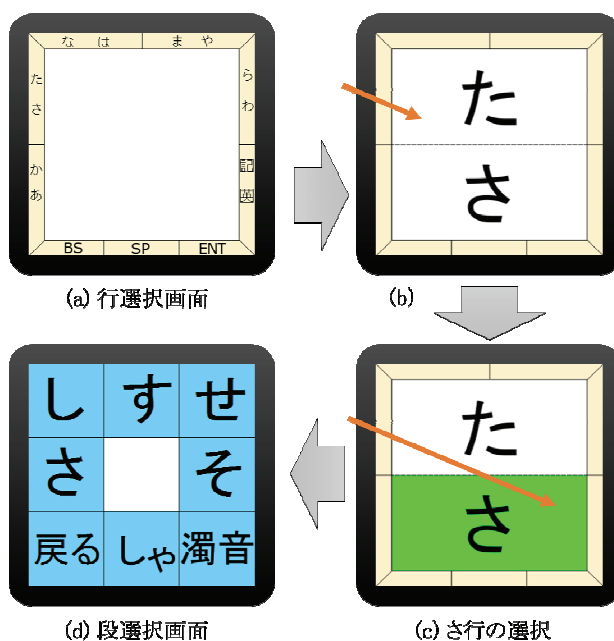


図 3 SliT の表示画面

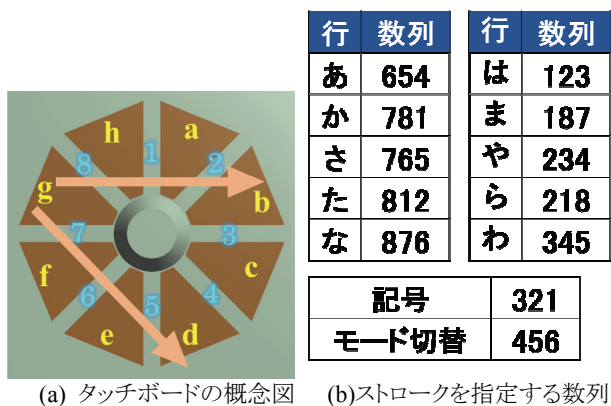


図 4 タッチボードのプロトタイプ



状態で指を画面から離せば、行が選択できる。段選択では、図 3(d)に示すように、9 分割した画面に平仮名が表示されるので、その 1 つをタップで選択する。濁音等は右下のボタンで表示を変えてから選択する。

SliT では直線のストロークで行を選択するため、L 字型のストロークを使う尾崎らの手法より移動距離が短い。また、キーの幅が画面の高さの約 1/2 であるため、ラフな位置決めでも誤入力となりにくい。このため、入力速度が向上している。使い始めから 125 文字入力した時点での速度は 28.7[CPM] (Characters Per Minute, 1分あたりの入力文字数)であり、初心者でも使いこなせている。このときの誤入力率は 4.7%となっ

ている。慣れれば 50[CPM]以上での入力も可能である。

図 3(a)の画面占有率は 26.4%で、5-TILES[26], TouchOne Keyboard[27], ZoomBoard[28]等の既存のソフトキーボードより低い値となっている。

タッチボードを小型化する点でも SliT は優れている。QWERTY キーボードを段階的に拡大表示する手法 [28][29][30]や、行段に対応する 15 個のキーを画面に並べる手法[31]では、キーの数が多いため、タッチボードが小さくなると誤入力が増える。画面の縁にキーを並べる TouchOne[27], HARI[32], BubbleSlide[33]は、キーを配置できる長さは SliT と同じだが、SliT より多くのキーを使い、また、それぞれのキーから 3~4 方向/距離の移動を区別する必要があるため、より高い精度の位置検出が求められる。このため、デバイスの小型化は SliT より難しい。

デバイス専用の文字入力手法を設計することも考えられるが、この場合には、デバイスを持ち歩くことやデバイスを内蔵したスマートウォッチのみを使うことが求められる。このため、タッチ画面と同じ操作で使えるほうが、汎用性が高いと考えた。

以上を考慮して、SliT 専用のタッチボードを開発した。

### 5. プロトタイプとその問題点

尾崎らの手法向けのタッチボードを変形して SliT に対応することは可能である。図 4 にそのデザインを示す。以後、これをプロトタイプと呼ぶ。図 4(a)の a~h がタッチセンサで、放射状に 8 個配置されている。先行研究と同様に、隣り合う 2 つのセンサに同時に触れたときに、その境界に割り当てられた数字が出力される。SliT では図 3(d)に水色で示した 8 区画をタッチで選択するので、この 1~8 の数字に対応付けることができる。

ストロークは 12 種類(開始位置が 6 区画×移動が 2 方向)あるが、これらは指先が通過する位置に割り当てられた数字の列で指定できる。たとえば、た行は左上から右上に移動するストロークなので、指先はセンサ g から h と a を通って b まで移動する。指がセンサの境界を通過する時には両側のセンサに同時に触れた状態となるため、間に割り当てられた数字が出力される。この出力を使って、た行を 812 と定義する。さ行のストロークは、指先がセンサ g, f, e, d と移動するので、765 となる。他のストロークも図 4(b)に示すとおり定義できる。

プロトタイプのボードで行った入力実験(方法は 7 章で解説する手法と同じ)では、10 名の被験者の平均で、ストロークの認識率が 88.9%、タップの認識率が 84.4%と低い値となった [34]。実験結果を解析したところ、ストロークでは左上から始める 781 や 876 で入力誤りが多く発生した。タップでは左側の 8, 7, 6 の入力で誤りが多かった。

被験者はすべて右手の親指でタッチしているの、指を伸ばした状態でタッチする場合に誤りが発生している。図 5 に 2 名の被験者が 8 の位置にタッチしたときの写真を示す。認識



率が高い被験者は親指の第一関節を曲げてタッチしているのに対して、認識率が低い被験者は関節を曲げずにタッチしている。指を曲げることで、タッチボードの鉛直方向から指先でボードに触れるようになるため、接触面積が減り、余分なセンサに触りにくくなるものと推測された。

## 6. 提案手法

### 6.1 タッチボードのデザイン

プロトタイプも先行研究のタッチボードも、センサに触りやすくするために、センサの面積が大きくなるようにデザインされている。このため、ボードの中心付近ではセンサの距離が近くなり、余分なセンサに触ってしてしまう原因となった。そこで、製作するボードのデザインを図6に変更した。センサのレイアウトが変わりはないが、中央の非タッチ領域が拡大されている。拡大によりセンサを狭くし、さらに、センサの間を2mmに広げることで、センサ間の距離を確保している。

また、非タッチ領域全体を一様に高くして周囲に段差を作ることで、ボードに対して指を立てないとセンサに触ることができないデザインにした。図7(a)はボードに指を置いた写真だが、この状態では中央の盛り上がりのために指先がセンサに届いていない。このまま指を左にずらしても、センサの幅が短いため、センサがない部分に触れることになる。しかし、図7(b)のように指を曲げれば、センサに容易に触れることができる。

センサには銅板を使っているため、土台の樹脂とは触感が異なる。また、センサは土台に埋め込まれているが、周囲には多少の段差が生じる。このため、指先の触覚でセンサに触れているか否かを判断できる。先に述べたように、センサにタッチするには指を曲げる必要があるため、センサに触れる動作を繰り返すうちに、意識しなくても指を立てた姿勢となる。この時の指の形は図5(a)の認識率が高い被験者と同じになるため、認識率の向上が期待できる。

指を立てることで、指先の接触面積が減り、余分なセンサに触れにくくなる。タッチボードの表面に薄い紙を張り、指先にインクをつけてタッチすることで、タッチ状態の指先の形と大きさを調べた。結果を図8に示す。指先の跡が重なることを防ぐために、2回に分けて計測している。図8(a)は奇数番号の、(b)は偶数番号の位置をタッチした結果である。それぞれの左側にはタッチ跡の画像を示している。右側は対応するセンサの位置で、丸はタッチした範囲を示している。

この計測では、画像の右側から、右手の親指でタッチしている。右側の2,3,4の位置では、タッチするときに邪魔になるものが無いため、指が傾いた状態で触れており、円形に近い跡が残っている。面積も大きめとなっているが、タッチ位置の右側にはセンサが無いので、誤入力とはならない。1と5の位置では、指を傾けると腹の部分が中央の領域に乗り上げて指先が浮く。このため、指先を立てた状態で触っており、縦長の跡

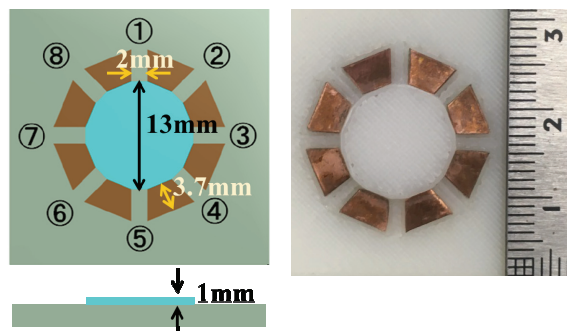
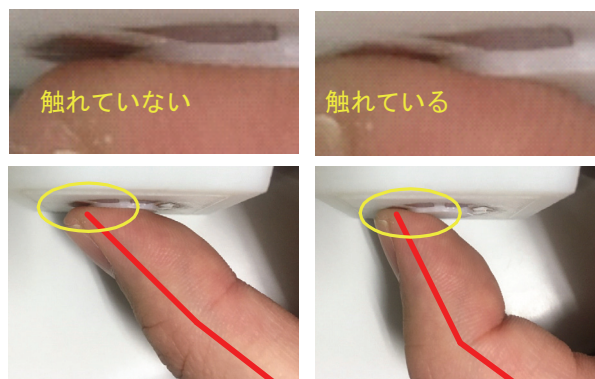
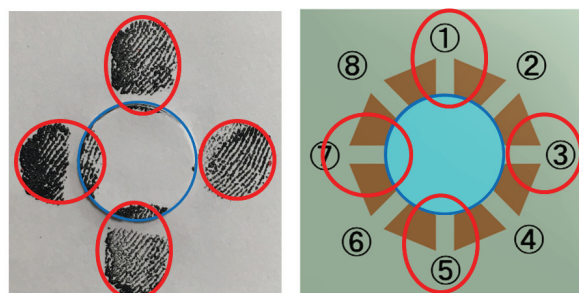


図6 本研究で作成したタッチボード

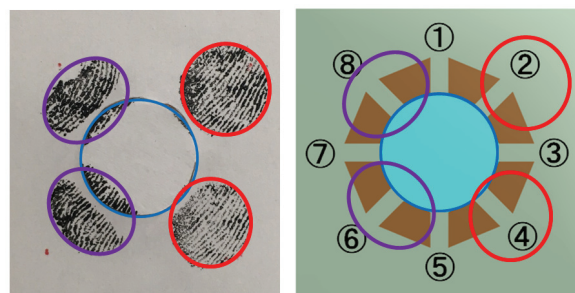


(a) 指を置いた状態 (b) タップした状態

図7 指の角度と指先のタッチ



(a) 奇数番号の位置をタッチした場合



(b) 偶数番号の位置をタッチした場合

図8 タッチしたときの指先の形

が残っている。上下に伸びているが、センサのない方向なので、誤入力とはならない。

左側の6, 7, 8の位置では、指が中央のふくらみに乗り上げることで、センサの内側(中央に近い部分)には触っていない。プロトタイプではこれらの位置で誤入力が多かったが、タッチ

位置が外側にずれることで隣のセンサから離れるため、誤って触れる回数の減少が期待できる。

## 6.2 タッチセンサの信号処理

タッチボードの8つのタッチセンサの信号をArduinoで処理し、1ms毎にタッチ位置を判定する。隣り合う2つのタッチセンサだけに触れている場合には間の位置に割り当てられた値(1~8)を出力する。指が離れているときや、センサ1つだけに触れているときには0を出力する。3つ以上のセンサに同時に触れた場合の出力は、その前の状態に依存する。候補となる値(全ての隣り合うセンサの組み合わせ)の中に直前に出力されていたものがあれば、その値が継続して出力される。存在しなければ、候補の中で最も小さい値を出力する。

ストロークの途中で一時的に1つのセンサだけに触れたり、指がセンサから離れたりすることがあるため、タッチボードの出力には途中で0が挟まる。ただし、上記を原因とする0は短い時間に限られるので、0が100ms未満の区間を削除する。次に、同じ数字が連続する区間を1つにまとめる。この結果、ストロークが正しく入力されれば、01230のような先頭と末尾が0(指がボードから離れていることを示している)で、間が連続する3つの数字(ただし、8の次は1に戻る)となる数字列が入力される。タップの場合は、010のようにタップした位置(1~8のいずれかの数字)が0に挟まれた数字列となる。

## 6.3 入力の補完

ストロークの入力においては、最初にタッチする位置がずれることで先頭の数字が欠けたり、1つ前の数字が追加されたりする。同様に、指を離す位置がずれると、最後の数字が欠けたり、1つ後の数字が追加されたりする。このような場合でもストロークを認識するため、入力を補完する機能を実装する。

補完方法として、数字列の先頭を信頼する前優先補完と、数字列の末尾を信頼する後優先補完が考えられる。前優先補完では、最初の2つの数字が連続していれば、それに続く数字を後に加えてストロークを判定する。後優先補完では、最後の2つの数字が連続していれば、前に連続する数字を加えてストロークを判定する。この両者のどちらが有効であるかは、評価実験で判定する。間の数字が抜けることも考えられる。そこで、0を除いた残りが13のように1つ飛ばした2つの数字となる場合には、123のように間の数字を埋めてストロークを判定する。この補完は前優先でも後優先でも行う。タップの場合には、前優先補完では最初の数字だけで、後優先では最後の数字だけで、タップ位置を判定する。

## 7. 評価実験

### 7.1 実験方法

実際の使用環境に近づけるため、図9に示すバンドを3Dプリンタで作成し、タッチボードとスマートウォッチ(SONY Smartwatch3)を埋め込んだ。被験者はこれを手首に着け、利



図9 入力実験に使う機材と表示される画像

き手の親指で操作する。親指以外の指や手の位置は、ボードに接触しない限り、被験者の自由とした。

実験に先立ち、タッチボードの使い方と実験方法を5分程度で説明する。次に、タッチボードを使って練習を行い、複数のセンサに触れる必要があることや、指の曲げ伸ばしの感覚などを体験してもらう。ここでは、後で述べる入力実験の1セット分の入力を行ない、入力が完全に正しければ、実験に移行する。誤りがある場合には、短い練習の後に上に述べた判定を繰り返す。ただし、練習開始から5分たったら、状況によらず入力実験を開始する。

ストロークの入力実験では、被験者は図4に示した12種類のストローク(連続する3つの位置をなぞる指の動き)を入力する。各ストロークをランダムな順に1回ずつ入力することを1セットとする。これを10秒間の休憩を挟みながら15セット行う。

秋田らの報告[24]によると、熟練者は1文字を約1秒で入力している。1文字はストロークとタップの組で指定されるため、ストロークに要する時間は0.5秒程度と考えられる。そこで、映像を用いて被験者が1回のストロークを0.5秒で行うように誘導した。図9に示すように、被験者はスマートウォッチの画面に表示された画像と同じストロークを入力する。このとき、青丸のアニメーションを表示し、その動きを追従するように指を動かしてもらうことで、入力速度を一定に保つ。ただし、入力速度がアニメーションと異なっても誤りとはしない。

タップ実験では、1セットで8か所あるタップ位置を1回ずつ入力する。これを10秒間の休憩を挟んで15セット行う。タップ順はセットごとにランダムに変える。この実験では、画面に青丸が表示された直後にその場所をタップする。表示されてから0.5秒の間にタッチボードから出力された位置で、タッチの正誤を判定する。

### 7.2 ストローク入力実験の結果と考察

10名の被験者(男性9名、女性1名)で実験を行った。年齢は20~24歳(平均22.2歳)で、6名が大学生、4名は社会人

である。いずれの被験者も右利きである。被験者は実験前に本研究、及び先行研究のタッチボードに一度も触れたことはない。表1にストロークの認識率を示す。無補完の行の数値は補完をしない場合の認識率、つまり、ストロークの3つの数字がすべて正しく入力された割合を示している。表の右端の数値は10名の被験者の平均値である。補完をしなくても平均で94.7%のストロークが正しく認識できている。

前優先補完と後優先補完の比較では、全ての被験者で前優先補完が勝っている。認識率は10名の平均で98.7%であり、高い値となっている。10名の被験者の認識率を標本として、前優先と後優先でt検定を行ったところ、 $t=2.18$ ,  $P=0.013$ となり、有意水準5%で認識率に差があると判定された。Mann-WhitneyのU検定でも同じ判定結果となっている。

以上の結果から、前優先補完を採用する。前優先では、最初に検出した位置で図3(b)のガイド画面を表示し、次に検出した位置で図3(c)のように色を変えることができるので、画面の表示を指でタッチする場合と同じにできる利点もある。

総計1800回の入力のうち、補完で救われる誤りも含めて96回の誤入力が発生している。誤入力が多かったストロークを表2に示す。この表の誤入力数の欄は正しく入力できなかった回数を示している。また、前欠落/後欠落は、それぞれ、最初/最後の数字が入力されなかった回数を示している。後欠落が多いため、前優先補完のほうが認識率の改善が大きくなる。

誤入力の上位2つは、1の位置、つまりボードの頂上から左右へのストロークとなっている。本研究のタッチボードは中央の非タッチ領域が大きいので、直線的な動きでは指先がボードから外れやすい。上から下への動きでは指先の速度が上がるため、最後の位置でセンサから外れたのだと考えられる。345はボードの右下の部分にあたり、指を曲げてストロークを行うため、指先の動きが窮屈になる。このため、指がボードから外れたと考えられる。前欠落と後欠落の差が無いことから、指が動かし辛いことが原因だと推察される。218と812は共にボードの上辺のストロークだが、どちらも8の入力に失敗している(218の最後の数字は8、812の最初の数字も8)。この理由としては、8が指を伸ばして触る位置にあるため、指の角度が足らず、指先がセンサから外れたことが考えられる。

### 7.3 タップ入力実験の結果と考察

表3にタップの認識率を示す。10名の被験者の平均値は、補完しなくても97%であり、ストロークに比べて高い値になっている。タップでも前優先補完が勝っており、10名の平均で99.4%と極めて高い認識率となった。ストローク実験と同様にt検定を行ったところ、 $t=2.11$ ,  $P=0.001$ となり、有意水準5%で前優先補完が優れていると判定された。U検定でも同じ結果となっている。このため、タップでも前優先補完を採用する。

表4は総計1200回のタップのうち、正しい位置とは異なる数字が入力された回数と、そのパターンを示している。表の後

表1 ストロークの認識率

被験者	A	B	C	D	E
無補完(%)	96.7	97.8	91.1	98.3	93.9
前優先補完(%)	98.9	99.9	97.8	100	96.1
後優先補完(%)	97.2	98.3	92.8	98.3	96.1

被験者	F	G	H	I	J	平均
無補完(%)	86.1	95.0	96.1	94.4	97.2	94.7
前優先補完(%)	97.8	100	98.9	97.8	100	98.7
後優先補完(%)	87.8	95.0	97.2	95.6	97.2	95.6

表2 ストローク別の入力誤りの回数と内訳

ストローク	誤入力数	後欠落	前欠落
187	24	15	0
123	20	18	1
345	16	8	8
218	12	11	0
812	11	0	7
その他	13	9	0
計	96	61	16

表3 タップの認識率

被験者	A	B	C	D	E
無補完(%)	97.8	97.8	97.8	98.9	95.6
前優先補完(%)	100	100	98.9	100	98.3
後優先補完(%)	97.8	97.8	98.9	98.9	97.2

被験者	F	G	H	I	J	平均
無補完(%)	93.9	97.8	95.6	98.9	96.1	97.0
前優先補完(%)	97.8	100	100	100	98.9	99.4
後優先補完(%)	96.1	97.8	95.6	98.9	97.2	97.6

表4 タップ位置別の入力誤りの回数とパターン

タップ位置	誤入力数	後過剰	前過剰
8	23	81	23
6	18	65	18
4	9	45	54
1	4		21
計	54	43	11

過剰には、正しい位置の後に余分な数字が追加される誤入力のパターンとその回数を示している。この誤りは前優先補完を使えば解消される。前過剰はその逆となる。表に示した4つの位置の他では誤入力は発生していない。

8の位置と6の位置でのエラーが突出している。これらの位置は指を伸ばしてタッチするため、指が寝た状態になりやすい。また、入力パターンが81、65となっていることから、それぞれ右隣の位置を遅れてタッチしている。これは、正しい位置をタッチした後に指が倒れるか、指を戻すときにセンサから離すのが遅れるかしたためだと推測される。8も6も中央の非タッチ



領域がタッチの邪魔になりづらい場所のため、エラーが多くなったと考えられる。4 の位置では前過剰が多数となっているが、これは、指が曲がり切る前にボードに触れてしまうことで、5,4 の順にタッチしたのだと推測される。

## 8. まとめと今後の課題

スマートウォッチ用文字入力手法 SliT 専用のタッチボードを開発した。このタッチボードでは、指先を立ててボードに触るように誘導するために、中央の非タッチ領域を大きくし、周囲に段差ができるようにデザインしている。さらに、センサの幅を狭くし、センサの間を広げることで、隣のセンサを誤ってタッチすることが起こりづらくしている。

SliT ではストロークとタップで文字を指定する。入力実験では、前優先補完を行うことで、1800 回のストロークの 98.7% を正しく認識することができた。タップでは 1200 回の入力の 99.4% を正しく認識することができた。これらの結果から、SliT 用の入力デバイスとして十分に高い精度が達成できたと考えている。

今後は、タッチボードを SliT に組み込んで動作を確認する。その後、入力実験を行い、画面を使う場合とタッチボードを使う場合の速度や誤入力率を比較する。また、主観評価実験により使いやすさを比較する。これらの結果から、SliT でタッチボードを使うことの有効性を検証する。

## 参考文献

- [1] 空中に手書きで文字入力する指輪型デバイス, <http://www.itmedia.co.jp/news/articles/1501/13/news090.html> (2018/3/3 確認).
- [2] Ring Zero ユーザーガイド, <https://logbar.jp/howtopdf.pdf> (2018/3/3 確認).
- [3] 萩原正人, 益子宗: KooSHO - 空中での手書きジェスチャーに基づく日本語入力環境, 言語処理学会 第 19 回年次大会 発表論文集, pp.414-417 (2013).
- [4] Y. Zhang, et. al.: SkinTrack: Using the Body as an Electrical Waveguide for Continuous Finger Tracking on the Skin, Proc. CHI'16, pp.1491-1503 (2016).
- [5] NEC, 腕を仮想キーボード化し, 非接触操作を実現する「ARmKeypad Air」を開発, [http://jpn.nec.com/press/201607/20160713\\_01.html](http://jpn.nec.com/press/201607/20160713_01.html) (2018/3/3 確認).
- [6] P. Wong, K. Zhu, and H. Fu: FingerT9: Leveraging Thumb- to-finger Interaction for Same-side-hand Text Entry on Smartwatches, Proc. CHI'18, Paper No.178 (2017).
- [7] E. Whitmier, et. al.: DigiTouch: Reconfigurable Thumb-to-Finger Input and Text Entry on Head-mounted Displays, Proc. ACM IMWUT'17, Vol.1, No.3, Article 133 (2017).
- [8] Finger touching, <http://www.yankodesign.com/2007/10/02/wearable-mobile-device-for-enhanced-chatting/> (2018/6/17 確認).
- [9] Tap, <http://www.tapwithus.com>. (2018/3/3 確認).
- [10] 投映型 Bluetooth プロジェクションキーボード TK-PBL042BK, <http://www2.elecom.co.jp/peripheral/full-keyboard/tk-pbl042> (2018/3/3 確認).
- [11] Smart Cast, <http://japanese.engadget.com/2015/06/03/smart-cast> (2018/3/3 確認).
- [12] Paper Keyboard, <https://itunes.apple.com/us/app/paper-keyboard-type-on-real/id715319520> (2018/3/3 確認).
- [13] Asu Smartwatch, <https://www.digitaltrends.com/watch-reviews/haier-asu-review/> (2018/6/17 確認).
- [14] K. Sun, et. al.: Float: One-Handed and Touch-Free Target Selection on Smartwatches, Proc. CHI'17, pp.692-704 (2017).
- [15] 黒澤紘生, 坂本大介, 小野哲雄: スマートウォッチの傾きと筋電情報の組み合わせによるポインティング手法, 情報処理学会インタラクティブ 2018, pp.926-931 (2018).
- [16] J. Gong, X. Yang, and P. Irani: WristWhirl: One-handed Continuous Smartwatch Input using Wrist Gesture, Proc. UIST'16, pp.861-872 (2016).
- [17] J. Gong, et. al.: WrisText: One-handed Text Entry on Smartwatch using Wrist Gestures, Proc. CHI'18, Paper No.181 (2018).
- [18] F. Li, R. Guy, K. Yatani, and K. Truong: The 1line keyboard: a QWERTY layout in a single line. Proc. UIST'11, pp.461-470 (2011).
- [19] iPhone, iPad, iPod touch で Siri を使う, <https://support.apple.com/ja-jp/HT204389> (2018/3/3 確認).
- [20] Cortana とは, <https://support.microsoft.com/ja-jp/help/17214/windows-10-what-is> (2018/3/3 確認).
- [21] 坂香太朗, 田中敏光, 佐川雄二: スマートウォッチのバンドに付けて使う極小文字入力ボードの開発, モバイル学会誌 Vol.7 No.1/2, pp.15-22 (2017).
- [22] 尾崎尚人, 田中敏光, 佐川雄二: 画面の縁を使うスマートウォッチ向けの文字入力方法, シンポジウム「モバイル'17」論文集, pp.75-78 (2017).
- [23] 藤中優, 田中敏光, 佐川雄二: スマートウォッチ専用文字入力ボードの開発 - ボードデザインの改良 -, 電子情報通信学会東海支部卒業研究発表会, D-4-3 (2018).
- [24] 秋田光平, 田中敏光, 佐川雄二: スマートウォッチ向けのスライドインを用いた文字入力手法, ヒューマンインタフェースシンポジウム 2017 予稿集, 5A2-1 (2017).
- [25] 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, 情報処理学会インタラクティブ 2018, pp.276-281 (2018).
- [26] 5-TILES, <http://fivetiles.com> (2017/12/11 確認).
- [27] TouchOne Keyboard, <http://www.touchone.net> (2017/12/11 確認).
- [28] S. Oney, C. Harrison, A. Ogan, and J. Wiese: ZoomBoard : A Diminutive QWERTY Soft Keyboard Using Iterative Zooming for Ultra-Small Devices, ACM CHI '13, pp. 2799-2802 (2013).
- [29] L. Leiva, et. al., Text Entry on Tiny QWERTY Soft Keyboards, Proceedings of CHI'15, pp.669-678 (2015).
- [30] 石井晃, 箱田博之, 志築文太郎, 田中二郎: Flicker: 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, 研究報告 HCI, 2015-HCI-164(6), pp.1-8 (2015).
- [31] 下岡純也, 山名早人: 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, DEIM Forum 2017 I3-2

(2017).

- [32] 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, 情報処理学会インタラクシオン 2016, pp.701-703 (2016).
- [33] 東條貴希, 加藤恒夫, 山本誠一: 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ーフェース, 情報処理学会インタラクシオン 2018, pp.1048-1053 (2018).
- [34] 坂香太郎, 田中敏光, 佐川雄二: スライドインを用いたスマートウォッチ向け文字入力手法専用のタッチボード, ヒューマンインタフェースシンポジウム 2017, 6A1-1 (2017).

**謝辞** 本研究はJSPS 科研費 JP16K00286 の助成を受けたものです.

### 著者紹介



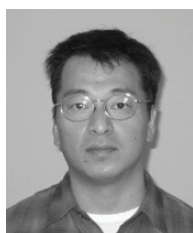
**坂 香太郎**(学生会員)

2016 名城大学工学部情報工学科卒業. 名城大学大学院理工学研究科情報工学専攻入学. 現在に至る. スマートウォッチに特化した文字入力手法専用デバイスの研究に従事. モバイル学会.



**田中 敏光**(正会員)

1984 名古屋大学大学院情報工学専攻博士(前期)修了. 工学博士. 1984 NTT 入社. 1994 名古屋大学大型計算機センター助教授. 2000 名城大学工学部情報科学科(現情報工学科)教授. 現在に至る. CG の研究・教育に従事. 情報処理学会.



**佐川 雄二**(非会員)

1987 名古屋大学大学院情報工学専攻前期課程修了. 1992 名古屋大学大学院情報工学専攻後期課程単位取得退学. 工学博士. 1992 名古屋大学助手, 同講師を経て, 2000 名城大学工学部講師, 2008 同教授. 現在に至る. 自然言語処理の研究・教育に従事. 電気学会.