

## 丸型 1 インチ画面のスマートウォッチで実行可能な文字入力手法

鈴木 桜史郎<sup>†</sup> 田中 敏光<sup>†</sup> 佐川 雄二<sup>†</sup>名城大学<sup>†</sup>

## 1. はじめに

秋田らが開発した SliT[1]には、画面占有率が低く、初心者でも簡単に使用できる長所がある。ただし、画面の周囲を 45 度刻みに分けてキーを配置するため、普通サイズの腕時計の画面ではキーが指先より小さくなり、誤入力が増える。本研究ではキー配置と文字選択ジェスチャーを改良することで、丸型 1 インチ画面のスマートウォッチで実行可能な文字入力手法を開発する。

## 2. 先行研究

SliT では、平仮名 1 文字を行と段を順に指定することで入力する。初期画面 (図 1) では、画面の周囲 2 mm の範囲を分割してキーとし、行を 2 つずつ割り当てる。キーをスライドイン (画面の外にタッチした指をそのまま画面内に滑り入れる動作) で横切ることで行を 2 つに絞り、指を離す位置でその 1 つを選択する。段選択では、図 2 のように 8 方向に分割された領域のそれぞれに文字が表示されるので、タップで選択する。

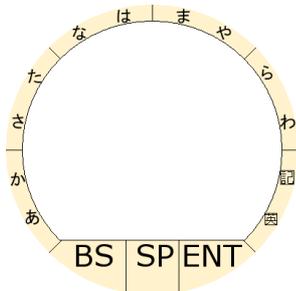


図 1 SliT の初期画面

図 2 た行の段選択画面  
段の選択画面

SliT では、行選択と段選択のどちらでも周囲を 45 度刻みに分けてキーを配置しているため、画面直径が 1 インチの場合、キーの最大幅は 10mm となる。指先より小さいため、隣のキーを誤ってタップするなどのファットフィンガー問題が発生し、誤入力が増える。また、初期画面の画面占有率は低いが、テキストを表示する領域が円形となるため、使い勝手が悪い。

Character input method that works on the 1-inch round screen of the smartwatch

<sup>†</sup>Ojiro Suzuki, Toshimitsu Tanaka, Yuji Sagawa

<sup>†</sup>Meijo University

## 3. 提案手法

テキストを矩形に表示したほうが読みやすい。そこで、初期画面 (図 3) では、画面に内接する正方形をテキスト領域とし、その周囲にキーを配置する。画面の周囲を 90 度ずつ分割することで、1 インチ画面でも指先の幅より長い 17mm のキーを確保する。キーの画面占有率は SliT より増えるが、36%にとどまっている。一方、テキストを矩形表示する場合には、SliT の 1.4 倍の表示ができる。

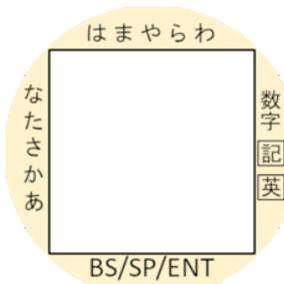


図 3 提案手法の初期画面

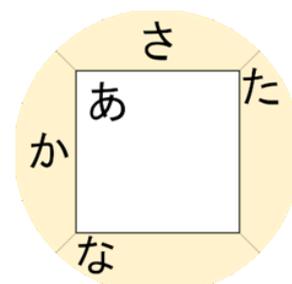


図 4 スライドイン後の画面

本研究でも行と段を順に指定して平仮名 1 文字を入力する。初期画面の左区画に平仮名の“あ～な行”を、上区画に“は～わ行”を割り当てる。区画はスライドインで選択する。指が画面の縁を通過すると、画面が 5 分割される。図 4 は左の区画を横切ったときの画面で、割り当てられた 5 行が各区画に表示されている。この図は右手で操作する場合の表示で、指先で隠されないように表示位置を左上にずらしている。

“あ行”は白色の範囲で指を離すことで選択する。周囲の行はその方向にスライドアウト (指を画面に触れたまま外に滑り出す動作) することで選択する。

幅が狭くても通過を検出できるため、中央の領域を大きくとることができる。

行を選択すると表示が変わる。図 5 は“た行”を選択したときの画面で、中央付近にタップすると

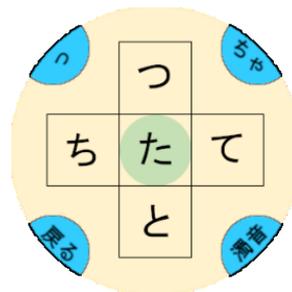


図 5 提案手法の段選択画面

“た”が、4方向へフリックすると、それぞれの方向に書かれた文字が入力できる。画面の四隅にボタン（青色の領域）が配置してあるが、中央付近をタッチした時点でボタンは無効になるため、タッチ位置がずれてフリックがボタンの領域に侵入しても、誤動作にはならない。中央の緑の円や四隅のボタンは、正確にタッチすることを促すために小さく表示されているが、それぞれ直径 12mm の検出範囲が確保されている。四隅のボタンで、濁音／半濁音／小文字に切り替える、文字入力をキャンセルする、“ちゃ”などの拗音付きの文字に変える、“ヴ”や小文字の“つ”を選択する、などができる。

#### 4. 初心者の入力速度の評価

使い始めの入力速度と誤入力率を調べるため、本システムを使用したことがない大学生5名を対象に評価実験を行った。平仮名5単語を入力する課題を、3分間の休憩をはさみ、10回行った。図6に平均の入力速度と誤入力率を示す。

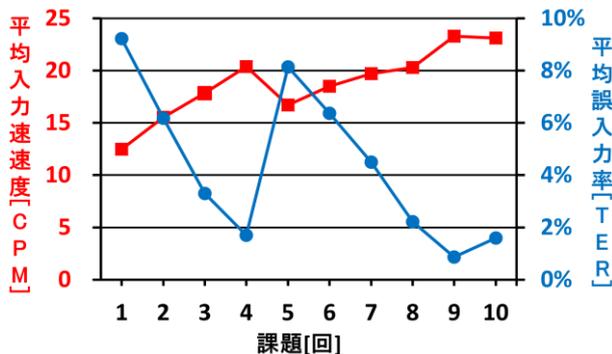


図6 初心者の平均入力速度と平均誤入力率

入力速度は、1分間当たりの入力文字数 (Characters Per Minute, CPM) で評価する。1回目の課題 (使い始め) は 12.5CPM だが、課題が進むにつれて速度が向上し、10回目の課題では 23.1CPM となった。

誤入力は、誤って入力したが修正された文字数と最後まで修正されなかった文字数の合計を課題の文字数で割った値 (Total Error Rate) で評価する。1回目の平均誤入力率は 9.2% である。5回目の課題で誤入力が多くなっているが、これは操作に慣れて指の速度が上がることでミスが重なった一時的なものと考えられる。その後の誤入力率は低下し、10回目には 2% 以下となった。

#### 5. 30日実験

システムを初めて使う (4節の実験に参加していない) 大学生3名に対して、毎日1回10単語ずつ入力する課題を30日間にわたって行い、使

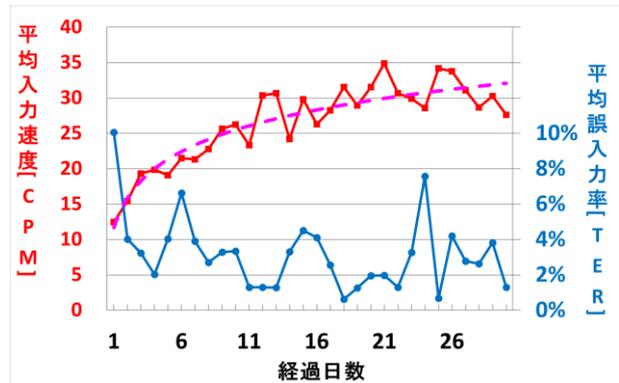


図7 30日実験の平均入力速度と平均誤入力率

い続けることでおこる変化を調べた。図7に平均の入力速度と誤入力率を示す。図7中の点線は入力速度を対数近似した曲線 ( $y = 5.99 \times \ln x + 11.67$ ) である。

1日目の入力速度は 12.5CPM で初心者実験と同じ値になっている。21日目に最高値の 34.9CPM に到達し、その後は 30CPM 付近で推移している。日々の入力速度は変動しているが、近似曲線によると、35CPM 付近で速度が頭打ちとなる。誤入力率は1日目が 10% 以上であったのに対し、20日以降はおおむね 3% 前後で推移している。24日目は 7.6% で、被験者の2人が偶然この日に多くミスをしており、ピークとなっている。入力ログによると、一人の被験者では行選択の誤りに気づかずにフリックしてしまうことで誤入力となっており、もう一人では、単語の読み間違い (課題“にゅうようじ”を“にゅうじょう”と間違えるなど) で誤入力となっていた。これらは被験者が実験に慣れたことで正確さが疎かになったために生じたと考えられるので、文字入力手法の問題ではない。この日以降は再び 3% 前後の誤入力率となっている。

#### 6. まとめ

丸型1インチ画面のスマートウォッチで実行可能な文字入力手法を提案し、評価実験を行った。初心者に対する実験では、約14分の使用で入力速度 23.1CPM、誤入力率 3% 程度で入力できるようになることが示された。30日間の実験では、いずれの被験者も、総使用時間1時間程度で、30CPM 程度の入力速度と 3% 程度の誤入力率を達成できた。

#### 文献

- [1] 秋田 他: 画面占有率低いスマートウォッチ向け文字入力手法 SliT, HI 学会論文誌, .21(1), 131-140 (2019)