

PonDeFlick: スマートフォンの日本語入力とフリックの向きを 共通化したスマートウォッチ向け円環型かな入力インタフェース

赤峰快[†] 加藤恒夫[†] 田村晃裕[†]

概要: スマートウォッチ向け日本語入力インタフェースは、これまで限られた操作領域の効率的な利用と規則的で分かりやすい操作性に主眼をおいた提案が行われてきた。本研究では、ユーザが使い慣れた既存のインタフェース、具体的にはスマートフォンのフリック日本語入力インタフェース（以降、テンキーフリック方式）との操作の共通性を考慮したかな入力方式 PonDeFlick を提案する。PonDeFlick はタッチスクリーン全体をフリック操作に利用できる円環型レイアウトを採用しながら、テンキーフリック方式とフリックの向きを共通化している。比較対象をスマートフォン上のテンキーフリック方式として大学生 18 名を対象に 10 日間の評価実験を行った結果、普段からスマートフォンでテンキーフリック方式を使用している実験群は、6 日目以降統制群に迫る入力速度に至った。誤入力率は、統制群の約 60% の 14.9% であった。

1. はじめに

現在、スマートウォッチは内蔵センサを用いた運動量計やスマートフォンからの各種通知の受信用として主に用いられているが、テキスト入力が簡便になればメッセージ送信や情報検索などに用途が広がると考えられる。

スマートウォッチにおける日本語入力には、限られた面積のタッチスクリーンを効率よく利用するとともに、操作の規則性が高くわかりやすいかな入力方式が必要になる。これまで、小さなタッチスクリーンでかな文字を効率よく入力する方式として ShuttleBoard[1], HARI キーボード[2], QuadKey[3], SliT[4] などが提案されてきた。当研究室でも円環型に行の先頭文字を配置するレイアウトに基づく一連のインタフェース BubbleFlick[5], BubbleSlide[6], BubbleGlide[7], BubbleOne [8]を提案してきた。

従来の日本語入力では、狭い操作領域の効率的な利用と規則的で分かりやすい操作性に主眼をおいた提案が行われてきた。しかし、使い慣れた既存のインタフェース、具体的にはスマートフォンのフリック日本語入力インタフェース（以降テンキーフリック方式）との操作の共通性は見過ごされてきた。

そこで本研究では、タッチスクリーン全体を入力操作に利用できる円環型レイアウトを採用しながら、テンキーフリック方式とフリックの向きを共通化した日本語かな入力方式 PonDeFlick を提案する。円環型に配置した五十音各行の先頭キーにタッチダウンして行を決定し、ア段であればそのままタッチアップ、イ段からオ段であれば一度画面中央に向けてスライドした後テンキーフリック方式と同じ向きにフリックをすることで母音を決定する。

2. 関連研究

スマートウォッチは画面が小さいため、その限られた領域で効率的に文字入力ができる入力方式が提案されてきた。英語入力用には、多くの人が使い慣れた QWERTY キーボ

ード配列を用いた入力方式が多く提案されている。Swipeboard[9]は QWERTY 配列の文字を 9 つのキーに分けて割り当て、最初のスワイプでキーを選択し、2 回目のスワイプで文字を特定する。Zshift[10]は、選択中の文字とその周辺領域をポップアップ表示する Shift[11]という方式を用いることで、選択している文字が指で隠れてしまう問題を解決している。Flickey[12]では、Zshift のポップアップの機能を採り入れ、タッチした列が目的の列と異なる場合にはそのまま左右にドラッグすることで列を選択し直し、そのまま指を離すか上下にフリックすることで文字を入力する。WatchWriter[13]では、タップ入力用の確率的キーボードと 1 ストロークで単語を入力できるジェスチャーキーボードを組合せることで高精度な入力を実現している。

日本語入力用には、五十音表の行と段、即ち子音と母音の組合せの規則性を活かしたインタフェースが多く提案されてきた。ShuttleBoard[1]は 10 個の子音キーと 5 個の母音キーで構成され、子音キーから母音キーへ指でなぞることで文字を入力する。指を離すことなくこの動作を繰り返すことで連続して文字を入力できる。HARI キーボード[2]は円環状に子音キーを配置したキーレイアウトで、子音キーを指定し画面中央へ指をスライドすると子音キーが母音キーに切り替わり、目的の母音キーへ再度スライドすることでかなを確定する。BubbleFlick[5]は円環状に行の先頭文字のキーを配置し、キーをタップすると放射上に当該行の文字が表示され、目的の文字の方向へフリックすることでかなを確定する。BubbleSlide[6]は各キーから画面中央へスライドした距離に応じて母音を確定する。BubbleGlide[7]と BubbleOne[8]は、行の先頭文字だけを指定する曖昧入力を導入し、N-gram 言語モデルを用いてかな漢字変換を含む入力を可能にした。

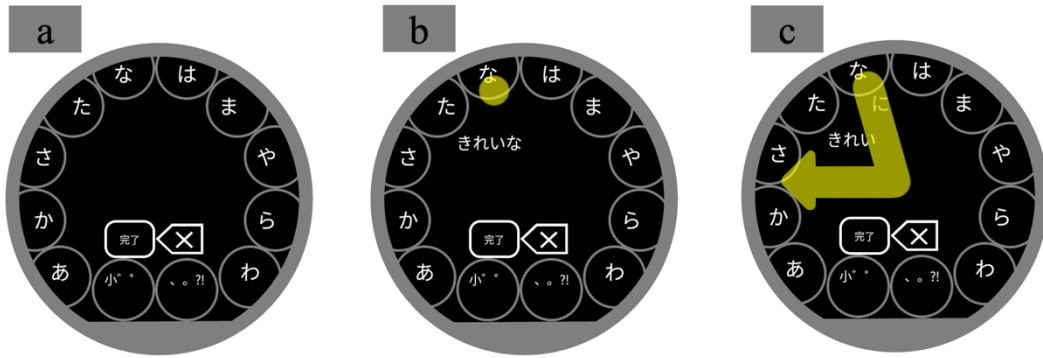


図 1. PonDeFlick a) 初期画面, b) 「な」の入力. 「な」のキーにタッチダウンし, タッチアップする. c) 「に」の入力. 「な」のキーにタッチダウンし中央にスライドしてから左方向にフリックする.

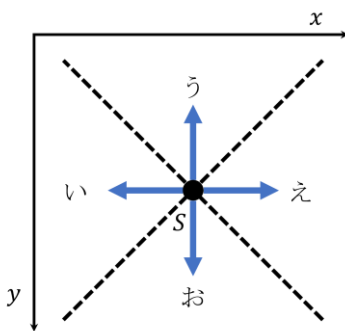


図 2. フリック方向の判定

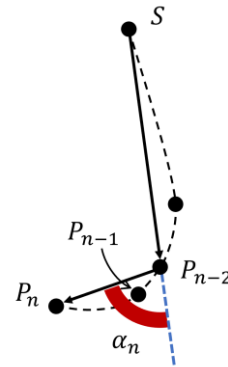


図 3. フリック開始点となる変曲点Sの探索例

3. PonDeFlick

図 1 に PonDeFlick の画面を示す. 図 1 a) は初期画面である. 10 種類の行の先頭文字キーを画面の縁に沿うようにして円環状に配置している. 小文字キーと句読点キーを画面下部に, 完了キーとバックスペースキーをその上に配置した. 入力テキストは画面中央に表示される. フリックによって選択された文字は入力テキスト表示部の上部に表示される. キーのサイズは直径約 8mm, 入力テキスト表示部は横 15mm × 縦 8mm である.

図 1 b) に, 「な」の入力操作を示す. ア段のかなは, 行の先頭文字キーにタッチダウンしたらその位置でタッチアップして確定する. 図 1 c) に「に」の入力操作を示す. イ段からオ段のかなは, 行の先頭文字キーにタッチダウンした後, 画面中央にスライドしてから, 図 2 に示すとおりテンキーフリック方式と同じ向きにフリックして確定する.

提案方式では, タッチダウン後のストローク途中で向きが変わることがある. フリックの向きの判定にはフリック操作の始点とみなす変曲点を確定する必要がある. 以下に変曲点の更新とフリック操作の判定アルゴリズムを示す.

1 ストロークの軌跡はタッチダウン座標を P_0 , タッチアップ座標を P_N として, 次の点列で与えられる.

$$P_0 = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}, P_1 = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}, \dots, P_N = \begin{pmatrix} x_N \\ y_N \end{pmatrix}$$

フリック操作の始点となる変曲点を S , フリック判定に必要な最小の移動距離閾値を D , 変曲点判定に必要な角度閾値を θ とする. S はタッチダウン座標 P_0 で初期化する.

タッチダウン後, タッチ位置の移動を検出すると約 2 ミリ秒間隔でタッチ位置を取得できる. まず, タッチダウン座標からの移動距離が D に満たずにタッチアップが検出された場合には, ア段のかなが入力されたと判定する.

タッチ位置の移動が検出されると, ストローク途中の新しいタッチ位置 P_n が取得されるたびに変曲点を探索する. 図 3 に模式図を示す. P_n に至る移動距離が閾値 D を超える最も近い点を新しい変曲点候補 $P_{n-k} (k \geq 1)$ とする (図 3 の場合は P_{n-2}). 変曲点候補 P_{n-k} までのストロークの変位ベクトルは $\overrightarrow{SP_{n-k}}$, 変曲点候補以降の最新の变位ベクトルは $\overrightarrow{P_{n-k}P_n}$ である. この 2 つの変位ベクトルのなす角 α_n を測る. $\alpha_n \geq \theta$ であれば変曲点を P_{n-k} に更新する ($S = P_{n-k}$). なお, 今回 $\theta = 70^\circ$, $D = 30\text{dp}$ (約 4.8mm) と設定した.

タッチアップが検出されると, フリックの変位ベクトル $\overrightarrow{SP_N}$ が確定する. 図 2 に示すとおり, 45° を境界としてイ段からオ段のかなのいずれかに判定する. 具体的には,

$$\vec{d}_i = (-1, 0)^T, \vec{d}_y = (0, -1)^T, \vec{d}_e = (1, 0)^T, \vec{d}_o = (0, 1)^T$$

との内積をとって最大のかなに確定する.



図 4. スマートウォッチにおけるテンキーフリック入力画面 (実物大) (2017 Google 日本語入力)



図 5. スマートフォンにおけるテンキーフリック入力画面 (2020 Google 日本語入力)

表 1. 実験で使用した短文 (左: 練習用, 右: 計測用)

私は大学の学生です。	夏がくるのを待っていた。
船に乗って向かう。	あの鳥は空を飛ぶ。
底抜けに明るい子供。	高級魚を食べる。
雪の中を歩く、血まみれの女。	内閣総理大臣は、話し上手。
カッパはエサを求めて、穴へやってきた。	補欠の夢は、先発出場。

4. テンキーフリック入力

テンキーフリック入力はスマートフォン上の日本語入力方式として最もポピュラーなインターフェースである。今回は Google 日本語入力を比較用インターフェースとした。スマートウォッチにおける入力画面を図 4 に、スマートフォンにおける入力画面を図 5 に示す。画面上部はテキスト表示部、それより下部は入力部になっている。入力部の中央には 3x4 の並びで行の先頭文字、濁点半濁点、句読点のキーが配置されている。また入力部の両外縁にはカーソル移動、文字削除などのキーが配置される。文字キーをタッチすると上下左右に当該行の文字が表示され、目的の文字の方向にフリックすることで文字を決定する。また予測変換機能により、入力部の上部に候補文字が表示される。入力方式にはフリック入力に加えてトグル入力も備えており、どちらでも入力ができる。トグル入力では文字のキーを連続してタッチすることで、文字を決定する。

5. 評価実験

5.1 実験の概要

PonDeFlick とテンキーフリック入力の性能と操作性を検証するために実験協力者を募り、10 日間の連続評価実験を実施した。実験期間を 10 日間とした理由は、多くの実験協力者にとってすでに慣れているインターフェースであるテンキーフリック入力に対して PonDeFlick がどれだけ迫れるのかを観察するためである。全 18 名の実験協力者を PonDeFlick を使用する実験群とテンキーフリックを使用する統制群に分けて、それぞれのインターフェースを 10 日間使用してもらった。実験の準備として、実験協力者に入力

してもらった短文を用意した。また、スマートウォッチでの日本語かな入力とスマートフォンのフリック日本語入力 (Google 日本語入力[14]) との関係进行调查するため、最終日にスマートフォンを用いて文字入力を行ってもらった。

5.2 実験条件

5.2.1 実験協力者

実験協力者は、大学生、大学院生の男女計 18 名とし内訳は男性 14 名、女性 4 名、年齢が 19 歳～23 歳、右利きが 16 名、左利きが 2 名であった。スマートウォッチを使用することがある協力者が 2 名であったが、スマートウォッチで文字入力をしたことがある協力者はいなかった。スマートフォンの使用歴は 7～11 年であった。

実験群の協力者のうち、普段からスマートフォンでテンキーフリック入力を利用している協力者 7 名を実験群 (テンキーフリック)、普段からスマートフォンの QWERTY ローマ字入力を使用している協力者 2 名を実験群 (QWERTY) とした。統制群の実験協力者は 9 名で、その中には普段から QWERTY ローマ字入力を使用している人はいなかった。

5.2.2 入力文セット

実験協力者には、1 日に練習用 5 文、計測用 5 文の計 10 文の短文を入力してもらった。毎日の文セットは日替わりである。2 日分の文字を足し合わせると、ひらがなパングラムになるように短文を作成した。パングラムとは、指定した言語の文字を文章中に 1 文字以上使用して構成された文章を意味している。1 日の文字数は実験協力者の負担を考慮して、平均 80 文字程度とした。短文中には句読点も含まれており、句読点も 1 文字として数えている。実際に使用した例文を表 1 に示す。実験では漢字部分にルビを振って 1 文字ずつモニターで提示した。

表 2. 3つの実験協力者群の5日毎の平均 CPM と平均誤入力率

入力方式	スマートフォン 入力方式	人数	平均 CPM [char/min]			平均誤入力率 [error/char, %]		
			1日目	5日目	10日目	1日目	5日目	10日目
実験群	テンキーフリック	7	25.4	36.2	45.3	23.2	20.3	14.9
	QWERTY	2	14.3	25.6	26.4	44.8	22.6	21.1
統制群	テンキーフリック	9	41.4	45.4	47.1	22.9	16.2	25.0

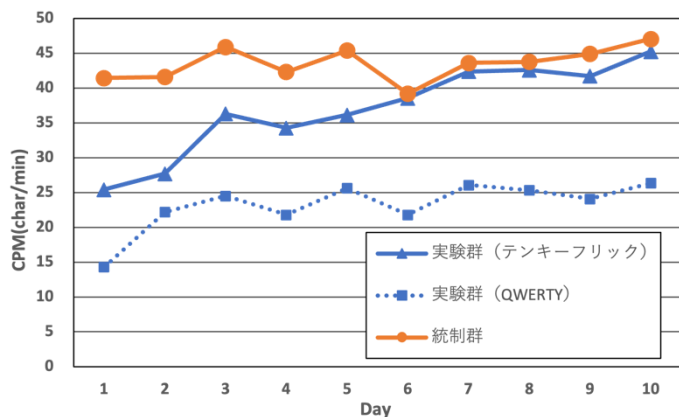


図 6. 3群の10日間の平均文字入力速度の推移

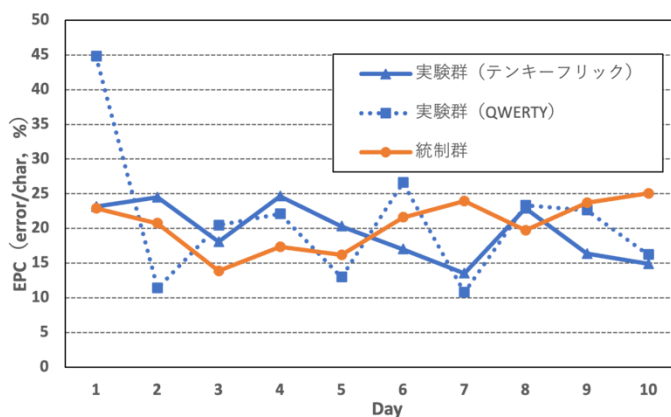


図 7. 3群の10日間の平均誤入力率の推移

5.2.3 実験手順

実験協力者には、研究室に来てもらい、割り当てられたインタフェースで、用意した短文のセットを入力してもらった。練習用の文セットを入力した後に計測用の短文を入力してもらった。入力する姿勢は自身が最適に入力できる姿勢に定め、デバイスを利き腕と反対の腕に装着してもらった。誤入力に対しては両方式ともにバックスペースキーで誤った部分を修正してもらった。

最終日にスマートフォンのフリック日本語入力での別の5文を練習なしで入力してもらい、スマートフォンにおける文字入力速度を計測した。今回の実験ではかな文字入力機能に限定して比較評価を行った。そのため実験協力者には、スマートウォッチとスマートフォンのテンキーフリック入力に備わっている予測変換、かな漢字変換機能は使用しないように伝えた。

5.2.4 評価項目

性能評価指標は文字入力速度 (CPM, char/min) と誤入力率 (EPC, error/char) とした。誤入力率を算出するにあたり、文字入力中に修正されたものも誤入力としてカウントした。誤入力率は (入力中の誤入力数) / (入力文の文字数) である。また、ユーザビリティの主観評価を取得するため、実験終了後には System Usability Scale (SUS) [15] の評価とインタフェースの使用感についてのインタビューを実施した。

5.2.5 使用したデバイス

スマートウォッチは Moto360 2nd Gen を使用した。Moto360 2nd Gen の解像度は 263[ppi]、画面の直径は 1.37

インチ (約 3.48[cm]) である。また、内部ストレージは 4GB で、RAM が 512MB である。

6. 評価結果

表 2 に 3 群の 5 日おきの平均文字入力速度、平均誤入力率を示す。

6.1 文字入力速度

3 群の 10 日間の文字入力速度の推移を図 6 に示す。図 6、表 2 より実験群 (QWERTY) の平均文字入力速度は、1 日目は 25.4[char/min]、10 日目が 45.3[char/min]であった。習熟の観点では 1 日目から 10 日目の間に 1 分間に約 1.8 倍多くの文字を入力できるようになった。実験群 (テンキーフリック) の平均文字入力速度は、1 日目は 14.3 [char/min]、10 日目が 26.4[char/min]であった。統制群の文字入力速度は 1 日目が 41.4[char/min]、10 日目が 47.1[char/min]であった。

PonDeFlick はテンキーフリック入力と比べると遅い結果となったが、普段からテンキーフリックを使用している実験協力者において、10 日間で同程度の入力速度まで上昇した。

6.2 平均誤入力率

3 群の 10 日間の平均誤入力率の推移を図 7 に示す。図 7 と表 2 より 10 日間の実験で実験群の平均誤入力率は統制群よりも低い値となった。10 日間の習熟の観点では、実験群 (テンキーフリック) は 23.2% から 14.9%、実験群 (QWERTY) は 44.8% から 21.1% に低減された。統制群は 1 日目が 22.9%、5 日目が 16.2%、10 日目が 25.0% となり日によってばらつきが見られる。

表 3. 評価項目と各入力方式における平均 SUS スコア

項目	実験群 (テンキーフリック)	実験群 (QWERTY)	統制群
頻繁に使いたいと思う	3.6	3.5	2.2
複雑だと感じる	1.6	1.5	1.4
簡単に使える	4.1	4.0	3.6
技術者のサポートが必要	2.3	3.0	1.7
様々な機能がまとまっている	3.6	4.5	2.7
多くの矛盾があると思う	1.4	1.0	1.3
使い方をすぐに覚えると思う	4.4	4.5	4.7
とても扱いにくいと思う	2.3	1.0	3.9
使用できる自信があると感じた	4.7	3.5	3.4
使用前に多く学ぶ必要がある	1.7	1.5	1.0
平均 SUS スコア	77.9	80	68.1

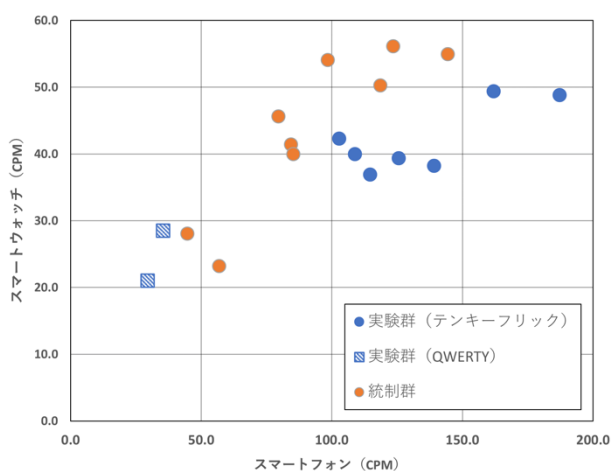


図 8. 各実験協力者のスマートウォッチによる 5 日目以降の平均文字入力速度とスマートフォンにおける文字平均入力速度との関係の散布図

6.3 SUS スコア

3 群の平均 SUS スコアを表 3 に示す. 表 3 から PonDeFlick はテンキーフリック入力よりも平均 SUS スコアが高いことが確認できた. また, 項目ごとのスコアを見ると PonDeFlick は「頻繁に使いたい」, 「簡単に使える」, 「機能がまとまっている」, 「使用できる自信がある」の項目で PonDeFlick がテンキーフリック入力を上回っている.

6.4 スマートフォンのフリック日本語入力との比較

PonDeFlick, テンキーフリック入力の両方式ともに, スマートフォンのテンキーフリック入力とフリックの向きを統一している. そこで, スマートフォンのフリック日本語における文字入力速度との関係を調べた. 各群での 5 日目以降の平均文字入力速度とスマートフォンのフリック日本語入力の文字入力速度を比較した. 各実験協力者のスマートウォッチによる 5 日目以降の平均文字入力速度とスマー

トフォンにおける文字平均入力速度との関係の散布図を図 8 に示す. 相関係数は実験群全体が 0.89, 統制群は 0.93 である. 平均文字入力速度についてそれぞれの入力方式で高い相関が見られる. PonDeFlick はスマートフォンのテンキーフリック入力の文字入力速度が速い人ほど早く入力ができるといえる.

両方式ともに高い相関が見られたため, 今後の研究では, 事前にスマートフォンの文字入力速度によってグループ分けすることで, 個人差の偏りを抑えた比較評価が行える.

6.5 インタビュー調査

10 日間の実験終了後に, PonDeFlick と実験全体についてインタビュー調査を実施した. 以下に PonDeFlick に対しての回答で有効だと考えられるものを示す.

- ・「フリック入力と方向が同じで覚えやすかった」という意見が得られた. このことから PonDeFlick のスマートフォンのフリック入力とフリックの向きを統一するというテーマは効果的であったといえる.

- ・「現在選択されている文字が分かりづらい」という意見があった. PonDeFlick では選択されている文字を画面上部に表示するだけであった. この問題を解決するにはキーをタップした際にそれぞれのフリック方向の入力候補を示すなど, 視覚的に見やすくする対策が必要である.

- ・「ヴァイブレーションのタイミングがつかめない」という意見が得られた. PonDeFlick のヴァイブレーションのタイミングはキーをタップした瞬間のみである. 既存のテンキーフリック入力では, キータッチ時だけでなく, フリックを検出し入力する文字が確定した瞬間にもヴァイブレーションが起こっている. フリック検出時や選択文字が変わるたびにヴァイブレーションを追加するなどの対策をする必要がある.

7. おわりに

本研究では, スマートフォンの日本語入力とフリック入力を統一したスマートウォッチ向けかな入力インタフェース PonDeFlick を提案し, 従来のテンキーフリック入力と比較実験を行い, 結果を比較した. 実験期間は多くの実験協力者にとってすでに慣れているインタフェースであるテンキーフリック入力に対して PonDeFlick がどれだけ迫れるのかを観察するため 10 日間とした.

PonDeFlick とテンキーフリック入力との比較では, 文字入力速度では普段からテンキーフリック入力を行っている協力者で同程度の値を示し, 平均誤入力率では PonDeFlick が上回る結果となった.

アンケート調査やスマートフォンとの比較から, 改善の必要はあるもののテンキーフリック入力とフリックの向きを統一するという試みは効果的であったと考える.

今回の実験では, 実験協力者数が計 18 名と少数であった. 今後は改良を行いさらに実験協力者数とバリエーションや

日数を増やして評価を行いたい。

参考文献

- [1] 齋藤, 奥, 佐川.: 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, インタラクション 2016 論文集, pp701-703 (2016).
- [2] 下岡, 山名.: 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).
- [3] 安福, 中村.: QuadKey: キーの数を4つにしたスマートウォッチ向けかな文字入力方式, 情報処理学会論文誌. Vol.60, no.8, p.1403-1412(2019).
- [4] 秋田, 田中, 佐川.: 画面占有率の低いスマートウォッチ向け文字入力手法 SliT, ヒューマンインタフェース学会論文誌. Vol.21, No.1 pp131-p140(2019).
- [5] T. Tojo, T. Kato, and S. Yamamoto.: BubbleFlick: Investigating effective interface for Japanese text entry on smartwatches, Proc. of MobileHCI' 18, No.44(2018).
- [6] 東條, 加藤, 山本.: BubbleSlide: スマートウォッチ向け円環型日本語かな入力インタフェース, 情報処理学会論文誌 vol. 60, no. 11, pp.2075-2084(2019).
- [7] 戸羽, 加藤, 田村.: グライド操作を用いたスマートウォッチ向け日本語曖昧入力キーボードの試作, ヒューマンインタフェースサイバーコロキウム, 1F4-4 (2020).
- [8] 戸羽, 加藤, 田村.: BubbleOne: スマートウォッチ向け日本語曖昧入力インタフェースの試作と評価, インタラクション 2021.
- [9] X. A. Chen, T. Grossman, and G. Fitzmaurice.: Swipeboard: A Text Entry Technique for Ultra-Small Interfaces That Supports Novice to Expert Transitions, Proc. of UIST'2014, pp.615-620, Honolulu, HI, USA(2014).
- [10] L. A. Leiva, A. Sahami, A. Catala, N. Henze, and A. Schmidt.: Text Entry on Tiny QWERTY Soft Keyboards. Proc. of CHI '15, pp. 669–678, New York, NY, USA (2015), ACM.
- [11] D. Vogel and P. Baudisch.: Shift: A Technique for Operating Pen-based Interfaces using Touch. Proc. of CHI '07, pp. 657–666, New York, NY, USA (2007), ACM.
- [12] 石井, 箱田, 志築, 田中.: 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, 情報処理学会研究報告, Vol.2015-HCI-164, No.6, pp.1-8, 2015.
- [13] M. Gordon, T. Ouyang and S. Zhai.: WatchWriter: Tap and Gesture Typing on a Smartwatch Miniature Keyboard with Statistical Decoding, Proc. of CHI' 16, pp.3817-3821(2016).
- [14] Google LLC (2020) .: Google 日本語入力. Google Play. <https://play.google.com/store/apps/details?id=com.google.android.inputmethod.japanese&pli=1>, (参照 2022 年 12 月 15 日)
- [15] J. Brooke.: SUS: A quick and dirty usability scale, Usability Evaluation in Industry, pp.189-194,(1996).