

KanaShark: SHARK²に基づく日本語ジェスチャ入力手法の検討

豊田 真吾* 日下部 完* 坂本 大介*

概要. スマートウォッチのような小さな画面での日本語入力の困難さを解決するため、単語単位のジェスチャ入力手法 KanaShark を提案する。KanaShark は、SHARK² や Swype 入力の手法を日本語入力に用いられる 12 キー配列に応用し、単語の子音のみを一筆書きでなぞることで入力する。経路形状から単語を確率的に推定するため、操作の曖昧さを許容しつつ高速な入力を可能とする。本研究では KanaShark の有効性を検証するため、9 名の参加者による 10 日間の評価実験で、フリック入力手法、および子音タップ入力手法と比較した。その結果、KanaShark は学習効果が顕著であり、フリック入力手法より有意に高速かつ少ない誤入力での入力が可能であることが示された。一方で、初期学習コストの高さが課題として示された。これらの結果に基づき、本研究の最後では、スマートウォッチの日本語入力の効率性の観点から KanaShark の将来展望を議論する。

1 はじめに

スマートウォッチは、健康管理や通知の確認など、人々の日常生活において広く活用されている。しかし、スマートウォッチという限られた矩形上でのテキスト入力は操作が煩雑になりやすい。その結果入力ミスが発生しやすく、ユーザビリティの低下を招いている [3, 10, 11, 14]。そのため、代わりに音声入力や定型文の選択による文字入力がよく使用されているが、これらは利用環境やプライバシーの観点から適用範囲が限定されるため [6, 8]、ディスプレイ上のタッチ操作を前提とした文字入力手法の発展が依然として重要である。

ラテン文字圏でよく使われている QWERTY 配列は、アルファベットだけでも 26 個のキーが必要であり、スマートフォンのような小さな画面上にこれらを配置すると、各キーのサイズが小さくなりやすく操作性の課題が生じやすい。一方、日本語圏で広く利用されている 12 キー配列は 12 個のキーの配置からなるレイアウトであり、ラテン文字圏の一般的なキー配置よりも各キーの面積を大きくできる利点がある。2017 年の長澤らの調査では、大学生の 90% が 12 キー配列を基にしたフリック入力やトグル入力をスマートフォンで利用していることが報告された [16]。しかし、スマートウォッチのようなディスプレイが極めて小さいデバイスでは、それでもタップ操作の煩雑さが依然として課題となる。

小さなキーへの入力の問題を解決するため、ラテン文字圏では SHARK² [13] や Swype [12] に代表される QWERTY 配列上で一筆書きのジェスチャから単語を確率的に推定する手法が、スマートフォンやスマートウォッチの効率的な文字入力手法として活用

され、タップによる文字入力が抱える問題を克服してきた。一方で日本語入力においても一筆書きのジェスチャ入力手法の研究が存在する。BubbleGlide [21] は、特定の単語の子音情報（あ、か、さ、た、な、は、ま、や、ら、わ）のみを選択することで文字を入力するジェスチャ入力手法である。しかし、ラテン文字圏で行われたような一筆書きの経路全体の形状から単語全体を確率的に推定するアプローチは十分に検討されていない。

本研究では、日本語 12 キー配列において子音情報のみを一筆書きで入力し、経路全体に基づく確率推定で単語候補を提示する手法 **KanaShark** を提案する。本手法はスマートウォッチ上での高速な日本語入力を実現することを目的とし、評価実験を通じて提案手法の有効性を検証することで、入力体験の向上に寄与することを目指す。

2 関連研究

2.1 ラテン文字圏におけるジェスチャ入力技術の発展

タッチディスプレイにおけるジェスチャ入力手法は、1990 年代の文字単位の認識から始まり、当時の Personal Digital Assistant (PDA) 向けの入力手法として Unistrokes [9] や Graffiti [17] が提案され、一筆書き可能なように簡略化されたアルファベットを手書きで入力する手法について研究された。その後 SHARK² に代表される単語単位のジェスチャ入力が登場し、ユーザの曖昧な単語入力を確率的に解釈するパラダイムが確立され、Swype などの製品として広く普及した。近年では、スマートウォッチのような小型デバイスに最適化されたジェスチャ入力手法として、WatchWriter [11] や C-QWERTY [7]、SwipeRing [18] などが提案されている。本研究は、日本語 12 キー配列においてこのような単語単位の

Copyright is held by the author(s).

* 北海道大学

ジェスチャ入力を提案するものである。

2.2 スマートウォッチの効率的な日本語入力

スマートウォッチでの日本語文字入力においては、12 キー配列をはじめとするような子音のみが配置されたキーボードが主流であり、その上でフリックやジェスチャ操作などによる工夫が検討されてきた。PonDeFlick[2] は、画面端に配置された子音を選択し、その後画面中央でフリック操作を行う方式である。これにより、キーボードの画面占有を画面端に限定しつつ、画面中央の広い領域をフリック領域として有効活用できるようにした。SliT[19] は、画面外からのスライドインジェスチャを用いて文字を入力することで、キーボードが画面を大きく占有しない入力手法である。BubbleGlide[21] や BubbleOne[20] は、入力したい単語の子音のみに注目する入力手法として検討され、BubbleGlide と比較して文字入力速度が約 1.1 倍向上した。

BubbleGlide では、円周上に配置された子音を一筆書きすることで、1 文字ずつ逐次的に入力する手法が提案され、本研究とは子音を一筆書きする点が共通している。しかし、逐次的な認識手法の場合、一般的なグリッド状のキー配列は離れたキー間を結ぶ経路の途中で別のキーを通過し、誤認識を招くため利用できず、これを回避するためにキーを円周上に配置する必要があった。我々は、BubbleGlide のような子音一筆書きによる単語入力のアプローチを基に、さらに一筆書きの経路全体から単語を確率的に推定するアルゴリズムを導入することで、曖昧な入力にも対応可能な 12 キー配列の日本語文字入力手法を実現できるのではないかと考えた。本研究では、日本語 12 キー配列上で単語の子音のみをなぞった経路から、SHARK² に基づく確率的推定により入力単語を決定するシステムを検討する。

3 KanaShark の設計

本章では、KanaShark のジェスチャ入力インタフェースの設計と経路から単語を確率的に推定するアルゴリズムを説明する。なお、KanaShark のソースコードは以下の脚注の URL からアクセスできる。¹

3.1 インタフェース設計

KanaShark のインタフェースは、図 1 に示すようにジェスチャ入力画面と候補語選択画面の二つの画面から構成される。ジェスチャ入力画面では、12 キー配列上に「あ、か、さ、た、な、は、ま、や、ら、わ」の子音情報が表示され、ユーザは入力したい単語の子音情報を直線で結ぶように一筆でなぞる。この際、入力対応表 (表 1) に示すように、濁音・半濁音・小書き仮名への変換や連続する同一子音は省



図 1. KanaShark における「ありがとう」の入力例。

略する。ユーザが指を離すと候補語選択画面へ遷移し、左上を起点とし、左から右、上から下へ尤度順に候補語を表示する。ユーザが候補語をタップして選択すると入力語が確定され、再びジェスチャ入力画面に戻る。

表 1. KanaShark の入力対応表。

単語	入力する子音情報
りんご	らわか
みかん	まかわ
コンピュータ	かわはやわた
インタラクション	あわたらかさやわ
かたたたき	かたか

3.2 辞書の作成

KanaShark の候補語推定に用いる辞書は、日本語 Web コーパス 2010[1] を基に構築した。変換候補として自然な語を収録するため、自立語を先頭に持つ語または付属語のみからなる語を抽出した。例えば、「私は」のような語は日本語の文脈として自然であり変換候補として適切であるが、「は学校」のように付属語が先頭に現れる語は今回は変換辞書から除外した。そのうえで、頻度の高い語から順に 1-gram を 10 万件、2-gram を 10 万件、3-gram および 4-gram を各 2 万件、5-gram を 1 万件収集した (n-gram は形態素単位)。これにより、合計 25 万語を変換辞書として採用した。各語彙については、MeCab (Python ラップとして fugashi[15] を利用) を用いて読み仮名を自動生成し、コーパスにおける語の出現回数を頻度として統合し表 2 に示すような辞書データを作成した。

3.3 候補語推定アルゴリズム

KanaShark では、SHARK² の文字推定アルゴリズムを参考に前処理と入力時推定の 2 フェーズによって文字を推定する。前処理では図 2 に示すように、単語辞書の各語に対して、キーの中央を直線で

¹ <https://github.com/toyoshin5/KanaShark>

表 2. KanaShark で用いる単語辞書データの例.

語	読み仮名	頻度
北海道	ほっかいどう	9395168
北海道札幌	ほっかいどうさっぽろ	287662
北海道大学	ほっかいどうだいがく	109236
北海道で	ほっかいどうで	432722
北海道に	ほっかいどうに	409883
北海道の	ほっかいどうの	1722061

結ぶ理想的なジェスチャ経路を生成し、線形補間を用いて経路を等間隔の 20 点からなる座標データに変換する。その後、アスペクト比を維持したまま水平方向の値域が 0 から 1 の範囲に収まるように正規化する。

入力時には図 3 に示すように、ユーザの入力経路データを同様に整形し、正規化された座標系列に変換する。次に、位置スコア、形状スコア、頻度の三つの観点 [13] から各単語との類似度を計算し、これらのスコアを統合してベイズ推定に基づく総合的な尤度を算出する。最終的に、尤度が一定のしきい値 (0.1%) を超えた単語のみを候補語として画面に表示する。位置スコア、形状スコア、頻度を統合する際の重みや候補語を表示するしきい値は、予備調査で探索的に決定した。処理時間については、Apple Watch Series 10 では前処理は約 5 秒、入力時推定は約 0.1 秒で完了する。

4 評価実験

KanaShark, フリック入力手法, 子音タップ入力手法の 3 種類の日本語入力手法を比較するために、計 10 日間にわたる継続的な評価を行った。本実験は参加者内比較計画で実施し、各参加者は 1 日 1 回のセッション内で 3 種類すべての入力手法を試した。手法の提示順序に起因する影響を排除するため、参加者 3 グループに対しラテン方格法に基づいて手法の提示順序を変更した。ただし、日ごとの学習効果を正しく評価するため、各参加者毎に割り当てられた順序を 10 日間を通して固定した。

4.1 実験条件

本実験では、3 節で説明した KanaShark に加え、フリック入力手法および子音タップ入力手法を含めて 3 種類の日本語入力手法を比較した。具体的なインターフェースは図 4 に示すとおりで、いずれの手法も watchOS の標準キーボードと同じキーの大きさ (7.47 mm × 3.85 mm) と配置を採用した。

フリック入力手法は、日本国内で広く利用されているスマートフォン向け日本語入力手法である。本研究では、watchOS の標準キーボードをそのまま用いるとキー入力イベント自体を取得できないこ

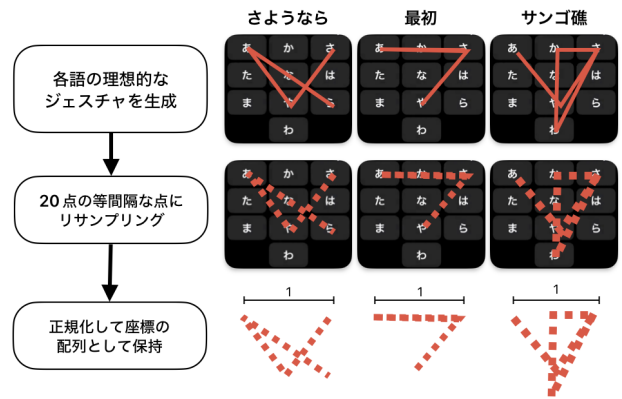
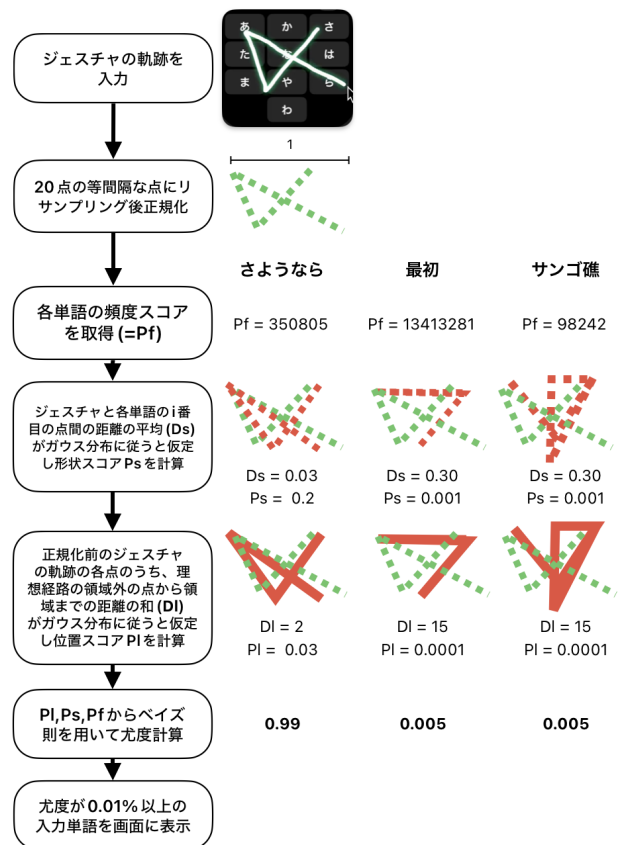


図 2. 候補語推定の前処理.



本図は処理の理解を助けるための模式図であり、数値は仮の値である。

図 3. 候補語推定の入力時処理.

とや、実験に影響を及ぼす可能性のある不要なインターフェース要素が含まれることから、図 4 に示すように、標準キーボードのレイアウトを再現した独自実装を用いた。実験タスクに含めない句読点や英数字・記号入力は実装せず、入力語の変換は「変換」キー押下によって図 1 の候補語選択画面と同形の画面が表示される仕様とした。

子音タップ入力手法は、フリック入力手法に対する



図 4. watchOS の標準キーボード (左) と本実験で比較する入力インタフェース (右)。

対照手法として設計した。入力したい単語の子音をタップし、決定ボタンを押下することで、図1の候補語選択画面と同様の画面が表示される。入力がジェスチャではない点を除き、その他の仕様は KanaShark と同一である。

4.2 評価項目

性能評価指標は文字入力速度 (Characters Per Minute, CPM) と誤入力率 (Errors Per Char, EPC) とした。CPM は、手本となる短文のかな漢字交じり表記の文字数を入力に要した時間で割った値であり、EPC は手本となる短文のかな漢字交じり表記の文字数に対する削除キー押下回数の割合である。また、主観評価として、10 日間の実験終了後には NASA Raw Task Load Index (NASA-RTLX) [5] と System Usability Scale (SUS) [4] に基づくアンケートを実施し、手法の好みとその理由についての記述式アンケートを行った。公平性を保つため、実験全体を通して KanaShark の手法名は「子音ジェスチャ入力手法」として提示した。

4.3 実験環境

実験には、文字入力用のスマートウォッチとして Apple Watch Series 10 46mm (watchOS 11.5) を用いた。実験用アプリケーションは SwiftUI を用いて開発し、Apple Watch 上で動作するように設計した。また、入力指示の提示のために iPhone 15 Pro Max (iOS 18.5) を机上のスマートフォンスタンドに設置して、実験参加者に iPhone 上の専用アプリケーションを通じて手本となる短文を提示した。

4.4 実験参加者

実験には 9 名 (男性 7 名, 女性 2 名, 平均年齢 22.67 歳, 標準偏差 0.67) の大学生・大学院生が参加した。利き手の分布は右利き 6 名, 左利き 3 名であり、いずれの参加者も利き手ではない側の手首に Apple Watch を装着して実験を行った。フリック入力経験年数の分布は、10 年以上が 2 名, 7~9 年間

が 5 名, 3~4 年間で 1 名, 経験なしが 1 名であった。

4.5 実験手順

実験参加者は、椅子に座り、机上に設置された入力指示用のスマートフォンの表示を確認しながら文字入力用のスマートウォッチを操作した。その際に、入力性能への影響を避けるため、参加者には入力中は肘や腕を含め身体を机に接触させないように指示した。各日のセッションは「練習フェーズ」と「本番フェーズ」で構成した。練習フェーズでは、各手法ごとに本番タスクの前に最大 3 分間の練習時間を設け、練習用の短文は本番タスクで用いるものとは異なるものとした。本番フェーズでは、iPhone の画面に提示される手本の短文を Apple Watch を用いて入力するよう指示し、各手法につき 10 個の短文 (平均 8.6 文字) を入力させた。短文セットは、個人差や短文の内容への依存を排除するために全参加者・全日程で共通とし、「あ」から「ん」までのかなが 1 回以上出現するひらがなパングラムを満たしている。また、短文の提示順序は手法・日ごとにランダムとした。入力中に誤りがあった場合は、iPhone 上で該当文字が赤く表示され、必ず削除キーを使用して修正するよう指示した。10 日間の全セッション終了後には、主観評価のためのアンケートを実施した。

5 結果

5.1 文字入力性能

各手法の 10 日間の CPM の推移を図 5 に示す。KanaShark は 1 日目から 10 日目の間に入力速度が約 1.9 倍向上し、1 日目から 4 日目までは他の手法に比べて入力速度が低かったものの、5 日目以降は他の手法を上回る速度を達成した。本研究では、学習効果が安定し各手法の習得後の性能が発揮される区間として、実験期間の後半 1/3 に該当する最終 3 日間を評価区間とし、統計量を表 3 に示す。この区間に対して Friedman 検定を実施した結果、図 6 の (a) に示すように 3 つの入力手法間に統計

的に有意な差が認められた ($\chi^2_{2,N=9} = 13.00, p < .01$). Holm 法で補正した Wilcoxon の符号順位検定による多重比較を行ったところ、提案手法である KanaShark はフリック入力手法と比較して有意に高速であり ($Z = -2.67, p < .05, r = .89$), 子音タップ入力手法もフリック入力手法より有意に高速であった ($Z = -2.67, p < .05, r = .89$). 一方で, KanaShark と子音タップ入力手法の間に有意な差は認められなかった.

EPC については, 実験期間全体を通して大きな変化は見られなかった. 同様に最終 3 日間に対して Friedman 検定を実施した結果, 図 6 の (b) に示すように 3 つの入力手法間に統計的に有意な差が認められた ($\chi^2_{2,N=9} = 15.94, p < .01$). Holm 法による補正を行った Wilcoxon の符号順位検定による多重比較を行ったところ, すべての手法間に有意差が認められ, KanaShark はフリック入力手法 ($Z = -2.67, p < .05, r = .89$), 子音タップ入力手法 ($Z = -2.49, p < .05, r = .83$) と比較して有意に低かった.

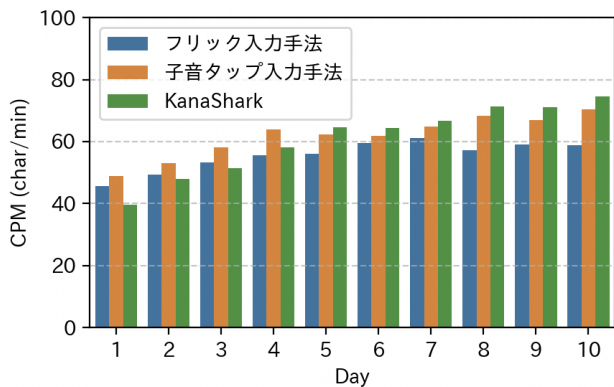


図 5. 各入力手法における CPM の推移.

5.2 メンタルワークロードとユーザビリティ

10 日間の全セッション終了後に実施した NASA-RTLX の Overall スコアを図 7 (a) と表 4 上段に示す. 子音タップ入力手法はフリック入力手法と比較して有意に低いスコアを示し ($p < .05$), メンタルワークロードが低いことを示した. しかし, KanaShark は標準偏差が大きいことから, 中央値が最も低いにもかかわらず他の手法に対して有意差は認められなかった. また, NASA-RTLX の 6 つの個別評価項目ごとにも Friedman 検定を実施したが, いずれの項目も手法間で有意な差は認められなかった.

SUS のスコアを図 7 (b) と表 4 下段に示す. 子音タップ入力手法と KanaShark は標準偏差が大きいことから, Friedman 検定では手法間で総合スコアに有意な差は認められなかった. しかし, 10 の個別評価項目ごとには手法間で有意差が認められ, 項

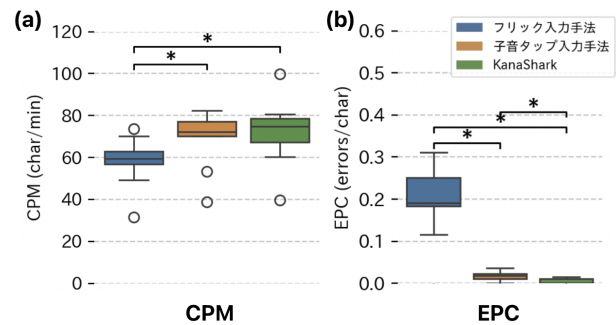


図 6. 最終 3 日間の (a) 平均 CPM. (b) 平均 EPC. 有意差は $*p < .05$.

表 3. 最終 3 日間の CPM と EPC に関する統計量.

CPM	Mean \pm SD	Median
フリック入力手法	58.37 \pm 12.34	59.53
子音タップ入力手法	68.54 \pm 13.94	72.12
KanaShark	72.32 \pm 16.29	74.86
EPC	Mean \pm SD	Median
フリック入力手法	0.205 \pm 0.062	0.191
子音タップ入力手法	0.018 \pm 0.012	0.019
KanaShark	0.005 \pm 0.006	0.000

目 4「この操作手法を利用するのに専門家のサポートが必要だと感じる」と項目 10「この操作手法を利用し始める前に知っておくべきことが多くあると思う」において, KanaShark はフリック入力手法よりもスコアが有意に高く, 学習の負担が大きいと評価された (それぞれ $p < .05$).

5.3 手法の好み

実験参加者 9 名中, 2 名がフリック入力手法, 5 名が KanaShark, 2 名が子音タップ入力手法を最も好んだ. フリック入力手法, 子音タップ入力手法, KanaShark の好みの順位の加重平均は, それぞれ 2.33, 2.11, 1.56 (数値が低い方が良い) であった.

KanaShark を好んだ理由としては, 「ジェスチャの経路が多少ずれてもシステムが意図を汲み取って変換してくれるため, 操作に気を使わなくても正確に入力できる (P2, P4, P7, P9)」, 「指を画面に置いたまま操作できるので楽だった (P3, P7)」, 「慣れれば最も速く入力できそうだと感じた (P2, P7)」といった点が挙げられた. 一方で, 「操作中に指が引っかかる感覚があった (P8)」, 「自分の指で次のキーが見えなくなる (P8)」といった課題も指摘された.

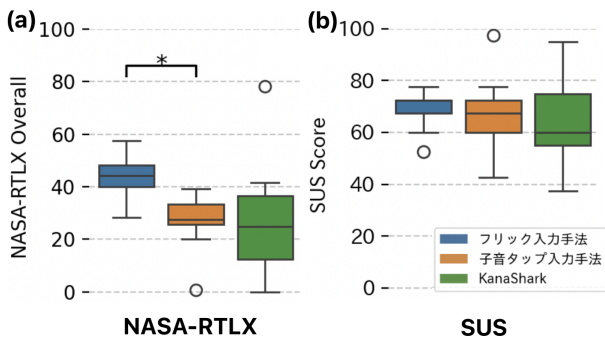


図 7. 実験終了後に実施した (a) NASA-RTLX の Overall. (b) SUS のスコア. 有意差は $*p < .05$.

表 4. NASA-RTLX と SUS スコアに関する統計量.

NASA-RTLX	Mean \pm SD	Median
フリック入力手法	42.78 \pm 9.67	44.17
子音タップ入力手法	26.67 \pm 11.42	27.50
KanaShark	28.33 \pm 23.03	25.00
SUS	Mean \pm SD	Median
フリック入力手法	68.61 \pm 7.92	72.50
子音タップ入力手法	66.94 \pm 15.35	67.50
KanaShark	62.78 \pm 18.13	60.00

6 考察と今後の課題

6.1 学習による入力速度向上

KanaShark は、学習に伴い入力速度が継続的に向上し、10 日間でフリック入力手法を上回る入力速度を達成した。CPM の推移では、実験初期はフリック入力手法に劣後したが、5 日目以降は他手法を上回り、最終 3 日間ではフリック入力手法に対する優位性が統計的に確認された。これは、同じ文字を入力するのに必要な動作が少ないためだと考えられる。KanaShark と子音タップ入力手法の間に有意差は認められなかったが、KanaShark の平均値は子音タップ入力手法以上であり、さらに学習曲線は 10 日間で飽和に達していないため (図 5)、さらなる習熟による向上が期待できる。

一方、EPC では、KanaShark が他の 2 手法より有意に低い値を示した。これは KanaShark が、スマートウォッチのような小さな画面での操作のずれを効果的に許容していることを裏付けるものである。これが結果的に修正操作の減少に繋がり、高速な入力を実現する上で重要な貢献を果たしていることが示唆される。

6.2 習熟過程における学習コストの課題

NASA-RTLX の Overall スコアによるメンタルワークロード評価では、KanaShark のスコアは参加者間でばらつきが大きく、他の手法と比較して統

計的な有意差は認められなかった。一方、SUS のスコアでは、KanaShark はフリック入力手法と比較して学習の負担が大きいと評価された。この学習負担の高さと評価のばらつきは、KanaShark の習熟過程に起因すると考えられる。

完全な習得のためには、子音の配置をある程度記憶する必要があり、参加者 (P8) が「自分の指で次のキーが見えなくなる」と指摘したように、初期段階ではオクルージョン問題が操作感を損なう要因となる。このような習熟上の課題が、一部の参加者にとっては特に高い認知的負荷として現れ、結果として参加者間での評価のばらつきを生んだことが示唆される。

6.3 今後の課題

評価タスクはスマートウォッチの利用文脈を踏まえ、辞書に存在する語による短文入力を中心に設計した。この前提は、日常的な返信が定型化されやすい実態を反映している一方で、辞書に存在しない固有名詞や専門語、文脈依存の語彙選択を含む状況に対する可能性は未検証である。この種の課題に対して Swype では、キーを一つずつタップして単語を入力する機能を併せ持つことで未知語の入力に対応している。これを参考に KanaShark においても、他の操作によって未知語入力を可能にする機能や入力された単語を学習する動的な辞書拡張機能を実装することが今後の課題となる。

また、システム側の実装要件として、語彙規模の拡大とリアルタイム推定の両立は非常に重要である。事前計算の再利用やデータ構造の改善によって、より大規模な辞書に対してもリアルタイムでの候補語推定を可能にすることが求められる。

さらに、環境に対するロバスト性の観点では、歩行中や公共交通機関内など身体への加速度が生じる状況における性能は未評価である。今後はそのような実利用環境での評価を進める。

7 おわりに

本研究では、スマートウォッチの 12 キー配列で、子音だけをなぞった経路から確率的に単語候補を推定し、文字を選択する入力手法 KanaShark を提案し、10 日間の継続評価でその有効性を確認した。実験の結果、10 日間の学習後はフリック入力手法より速く少ない誤りでの入力を実現した。一方で、初期の学習時の負荷についての課題も明らかとなった。本研究は、スマートウォッチにおける日本語入力に新たな選択肢を与えるものである。今後は、実利用環境における有効性の検証や、未知語の入力方法の検討を進めることで、より実用的なシステムの実現を目指す。

謝辞

本研究は、JST 創発的研究支援事業、JPMJFR 226S の支援を受けたものである。

参考文献

- [1] 日本語ウェブコーパス 2010. <http://www.s-yata.jp/corpus/nwc2010/ngrams/>, 2010. Accessed: 2025-08-02.
- [2] K. Akamine, R. Tsuchida, T. Kato, and A. Tamura. PonDeFlick: A Japanese Text Entry on Smartwatch Commonalizing Flick Operation with Smartphone Interface. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24, New York, NY, USA, 2024. Association for Computing Machinery.
- [3] P. Baudisch and G. Chu. Back-of-device interaction allows creating very small touch devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, p. 1923–1932, New York, NY, USA, 2009. Association for Computing Machinery.
- [4] J. Brooke. *SUS – a quick and dirty usability scale*, pp. 189–194. 01 1996.
- [5] J. C. Byers, A. C. Bittner, and S. G. Hill. Traditional and raw task load index (TLX) correlations: Are paired comparisons necessary? In A. 1989.
- [6] B. S. Chaparro, J. He, C. Turner, and K. Turner. Is Touch-Based Text Input Practical for a Smartwatch? In C. Stephanidis ed., *HCI International 2015 - Posters' Extended Abstracts*, pp. 3–8, Cham, 2015. Springer International Publishing.
- [7] G. Costagliola, M. De Rosa, R. D'Arco, S. De Gregorio, V. Fuccella, and D. Lupo. C-QWERTY: a text entry method for circular smartwatches (S). pp. 51–57, 07 2019.
- [8] C. Efthymiou and M. Halvey. Evaluating the Social Acceptability of Voice Based Smartwatch Search. In S. Ma, J.-R. Wen, Y. Liu, Z. Dou, M. Zhang, Y. Chang, and X. Zhao eds., *Information Retrieval Technology*, pp. 267–278, Cham, 2016. Springer International Publishing.
- [9] D. Goldberg and C. Richardson. Touch-typing with a stylus. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, CHI '93, p. 80–87, New York, NY, USA, 1993. Association for Computing Machinery.
- [10] J. Gong, Z. Xu, Q. Guo, T. Seyed, X. A. Chen, X. Bi, and X.-D. Yang. WrisText: One-handed Text Entry on Smartwatch using Wrist Gestures. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, p. 1–14, New York, NY, USA, 2018. Association for Computing Machinery.
- [11] M. Gordon, T. Ouyang, and S. Zhai. WatchWriter: Tap and Gesture Typing on a Smartwatch Miniature Keyboard with Statistical Decoding. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, p. 3817–3821, New York, NY, USA, 2016. Association for Computing Machinery.
- [12] S. Inc. Swype: Text Input for Screens. <https://web.archive.org/web/20080911004604/http://www.swypeinc.com/index.html>, 2008. Accessed: 2025-08-04.
- [13] P.-O. Kristensson and S. Zhai. SHARK2: a large vocabulary shorthand writing system for pen-based computers. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, UIST '04, p. 43–52, New York, NY, USA, 2004. Association for Computing Machinery.
- [14] X. Lin, Y. Chen, X.-W. Chang, X. Liu, and X. Wang. SHOW: Smart Handwriting on Watches. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 1(4), Jan. 2018.
- [15] P. O. McCann. fugashi: A Cython MeCab wrapper. <https://github.com/polm/fugashi>, 2019. Accessed: 2025-08-04.
- [16] N. Nagasawa. How Japanese university students type on smartphone and PC. 43:67–72, 12 2017.
- [17] I. Palm. Graffiti Arrives! – Press Release enclosed. <https://www.grot.com/zoomer/zoomer-list/zoomer-list-3Q1994/0048.html>, 1994. Accessed: 2025-08-24.
- [18] G. Rakhmetulla and A. S. Arif. SwipeRing: Gesture typing on smartwatches using a segmented QWERTY around the bezel. In *Proceedings of Graphics Interface 2021, GI 2021*, pp. 166–177. Canadian Information Processing Society, 2021.
- [19] 秋田 光平, 田中 敏光, 佐川 雄二. 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, 21(1):131–140, 2019.
- [20] 戸羽 遼太郎, 加藤 恒夫, 田村 晃裕. BubbleOne: スマートウォッチ向け日本語曖昧入力インターフェースの試作と評価. 情報処理学会 インタラクシオン 2021, 2021.
- [21] 戸羽 遼太郎, 加藤 恒夫, 山本 誠一. BubbleGlide: N-gram を用いたスマートウォッチ向け日本語曖昧入力ジェスチャーキーボード. 情報処理学会 インタラクシオン 2020, 2020.

未来ビジョン

スマートウォッチの小さな画面は、テキスト入力の大きな障壁とされてきた。画面が小さいためにキーは必然的に小型化し、密集して配置される。この「キーが小さく密集した」環境は、正確な位置をタップする必要がある従来の手法にとっては大きな弱点となり、誤操作を誘発する原因となっていた。

しかし本研究を通して感じたのは、一筆書きでなぞるジェスチャ入力はこのような環境と相性が良いということである。「キーが密集している」ことは、すなわちキー間の距離が短いことを意味し、ジェスチャ入力にとってこの短い移動距離は指の操作量を減らし、素早い入力を可能にする。また、「キーが小さい」ことに対して、ジェスチャ入力は影響を受けにくいと感じた。経路全体の形状で単語を認識するため、

個々のキーを正確に通過する必要がないからである。実際に、キー1つの面積が小さいラテン文字圏のスマートフォンのQWERTY配列で、Swypeなどのジェスチャ入力が広く普及したことも、このような利点を反映していると考えられる。

今後は、本文で述べた今後の課題の対応が進めば、スマートウォッチはより実用的なテキスト入力環境を獲得すると考えている。さらに、大規模自然言語モデルなどの組み合わせにより文脈とジェスチャの形状から最適な語を推定できるような未来も考えられる。

このように、スマートウォッチの小さな画面への入力における障壁に対して、確率的に文字を推定するジェスチャ入力は相性が良く、日本語圏においても新たなインタラクションデザインとして受け入れられるのではないかと感じている。