

筑波大学 情報学群 情報メディア創成学類

卒業研究論文

スマートウォッチのみを用いた  
センシングに基づく片手文字入力手法

角田 陸

指導教員 志築 文太郎, 川口 一画

2023年2月

## 概要

スマートウォッチの画面は超小型であるため、特に文字入力の際にキーボードを画面に表示することによって、スマートウォッチのキーボード以外の表示領域が制限されるという出力の制約がスマートウォッチに生じる。さらに、画面に表示された小さいキーをタップする際には、指に覆われたキーを視認することができない、および誤タップが発生するという入力の制約がスマートウォッチに生じる。また、その際には、スマートウォッチを装着していない手を用いてスマートウォッチを操作する必要があるため、片手がふさがっている状況におけるスマートウォッチの操作は難しい。本手法では、画面におけるタップではなくハンドジェスチャとして入力を行う。ユーザはスマートウォッチを装着した手における親指から他の指へのジェスチャを用いて、アルファベット群の選択および予測された単語の選択を行う。これによって、上述の出力の制約の解決、入力の制約の解決、および片手のみによる文字入力の実現がされた。さらに、本手法は追加のセンサを用いることなく、市販のスマートウォッチのセンサのみを用いて動作する。

# 目次

<b>第1章</b>	<b>序論</b>	<b>1</b>
1.1	背景	1
1.2	目的	2
1.3	アプローチ	2
1.4	貢献	2
1.5	本論文の構成	2
<b>第2章</b>	<b>関連研究</b>	<b>4</b>
2.1	スマートウォッチの加速度センサを用いたジェスチャ識別	4
2.2	スマートウォッチにおける文字入力手法	5
2.2.1	ソフトウェアキーボードを活用した文字入力手法	5
2.2.2	ハンドジェスチャを用いた文字入力手法	6
2.3	単語予測アルゴリズムを用いた文字入力手法	6
2.4	スマートウォッチにおける片手操作	7
<b>第3章</b>	<b>スマートウォッチの加速度センサのデータに基づく片手文字入力手法</b>	<b>8</b>
3.1	概要	8
3.2	実装	10
3.2.1	システム構成	10
3.2.2	リアルタイムジェスチャ識別	10
3.2.3	単語予測アルゴリズム	14
<b>第4章</b>	<b>実験1：ジェスチャ識別精度の調査</b>	<b>15</b>
4.1	実験機器	15
4.2	データ収集	15
4.3	機械学習アルゴリズム比較	15
4.4	セット間交差検証および5分割交差検証	17
4.4.1	結果	17
4.4.2	考察	17
<b>第5章</b>	<b>実験2：文字入力性能の調査</b>	<b>20</b>
5.1	評価指標	20
5.2	実験手順	20

5.3	結果	22
5.4	考察	22
<b>第6章</b>	<b>議論</b>	<b>24</b>
6.1	文字入力性能, ユーザビリティ, および作業負荷の調査	24
6.2	ジェスチャ識別の改良	24
6.2.1	ジェスチャ検知処理の改良	24
6.2.2	データ収集の改良	25
6.3	文字入力方法の改良	25
6.3.1	予測単語の提示	25
6.3.2	コーパスに含まれない単語の入力	25
6.3.3	予測単語の選択	25
6.3.4	予測単語の切り替え	26
6.3.5	文字入力インタフェース	26
6.3.6	ジェスチャセットおよびそれに伴うキーボードレイアウト	27
6.4	本手法の応用	27
6.5	ユビキタス環境における使用に向けて	27
<b>第7章</b>	<b>結論</b>	<b>29</b>
	謝辞	30
	著者論文リスト	31
	参考文献	32

## 目次

3.1	ピンチジェスチャおよびそれに対応するアルファベット群. . . . .	8
3.2	フリックジェスチャおよびそれに対応して選択される予測単語. . . . .	9
3.3	ユーザが本手法を用いている様子. . . . .	9
3.4	文字入力 UI の初期状態. . . . .	11
3.5	初期状態 (図 3.4) から小指のピンチジェスチャ (図 3.1 (a)) を行った後の状態. . . . .	11
3.6	図 3.5 の状態から親指のピンチジェスチャ (図 3.1 (e)) を行った後の状態. . . . .	11
3.7	図 3.6 の状態から薬指のピンチジェスチャ (図 3.1 (b)) を行った後の状態. . . . .	11
3.8	図 3.7 の状態から中指のフリックジェスチャ (図 3.2 (c)) を行った後の状態. . . . .	11
3.9	文字入力の順序. . . . .	12
3.10	実装におけるシステムの構成. スマートウォッチとスマートフォンは, Bluetooth にて加速度データおよび文字入力状態に関するデータの通信を行う. スマートフォンと PC は USB 接続にて加速度データの通信を行い, WebSocket を用いて文字入力状態に関するデータの通信を行う. . . . .	12
3.11	リアルタイムジェスチャ識別処理の流れ. . . . .	12
3.12	リアルタイムジェスチャ識別処理において扱うデータの流れ. . . . .	12
3.13	検知されるジェスチャの Y 軸の波形の例. 赤色の点が負の値のピーク, 青色の点が正の値のピークを示す. 1848 点から 2048 点の間 (末尾 200 点) にピークが存在するときは, ジェスチャは検知されない. . . . .	13
4.1	セット間の交差検証における混同行列. . . . .	18
4.2	すべてのセット間の交差検証の平均の混同行列. . . . .	19
4.3	5 分割交差検証の混同行列. . . . .	19
5.1	文字入力速度実験用のユーザインタフェース. ユーザは sentence の下に表示されているフレーズを入力する. 入力されたフレーズは一番下に表示される. . . . .	21
6.1	図 3.7 における行われたジェスチャの表示を改良したもの. ユーザは「software」という単語を入力しようとしている. . . . .	26
6.2	片手における T9 キーボードの例. . . . .	27

# 第1章 序論

近年、デスクトップコンピュータ、ラップトップコンピュータ、スマートフォン、およびタブレットコンピュータに次ぐパーソナルコンピューティング端末として腕時計に様々な機能を付加したコンピュータ（以降、スマートウォッチ）が普及している。スマートウォッチには、メールの確認、通話、メッセージング、および Social Networking Service (SNS) などの様々なアプリケーションが存在するが、これらのアプリケーションには文字入力が必要不可欠である。既存のスマートウォッチにおける文字入力手法の多くは、QWERTY キーボードを模したソフトウェアキーボードを用いる手法であるが、これらにはいくつかの問題点が存在する。そのため、スマートウォッチにおいて文字入力はあまり行われなない [1]。本研究では、これらの手法に代わる文字入力手法の実現を目的とする。本章においては、まず、本研究の背景を述べ、その後目的、アプローチ、貢献、および本論文の構成を述べる。

## 1.1 背景

スマートウォッチは主にスマートウォッチに搭載されているタッチスクリーンに触れることによって操作される。しかし、スマートウォッチのタッチスクリーンの大きさは、例として Apple Watch Series 8 (41 mm) の場合 1.69 インチ（縦：41 mm，横：35 mm）、Google Pixel Watch の場合 1.25 インチ（直径：41 mm）と小さいため、キーボードを表示すると小さいターゲットが密集した状態になる。したがって、誤タップが発生する問題（Fat Finger 問題 [2]）が生じる。また、文字を入力するためにキーボードが表示されると、タッチスクリーンの大きさの制約によってキーボード以外のコンテンツの表示領域が小さくなるという問題 [3] がある。さらに、ユーザはスマートウォッチを装着していない手を用いてスマートウォッチを操作する必要があるため、片手がふさがっている状況における操作が難しい。上記の3つの理由から、スマートウォッチにおける文字入力は音声入力が主流であるが、音声入力には図書館などの声を発する事ができない場所において用いることが困難であるという問題、他人に入力内容を聞かれるという問題、および騒音の大きい環境では用いることができないという問題がある。よって、現在のスマートウォッチの主な用途は情報提示およびライフログであり、文字入力はあまり行われなない [1]。しかし、スマートウォッチにおける文字入力の利便性を向上させることにより、メールおよびメッセージングアプリケーションにおける返信、およびメモを取ることが可能になるなどのように、用途が広がる。

上記のスマートウォッチにおける文字入力についての問題に対して、スマートウォッチを装着した手のジェスチャを用いる片手文字入力手法 [4, 5, 6, 7] が提案されている。しかし、こ

これらの手法を用いるためには端末に外部センサを追加する必要がある。また、スマートウォッチを装着した手を用いて空中などに描いた文字を、スマートウォッチのセンサのみを用いて識別する片手文字入力手法 [8, 9, 10] も存在する。本研究では、スマートウォッチを装着した手における親指から他の指へのジェスチャを、スマートウォッチのセンサのみを用いて識別することによる文字入力を実現する。

## 1.2 目的

本研究ではスマートウォッチにおいて、誤タップを誘発しない、キーが画面を専有しない、片手のみを用いる、かつ外部センサを用いない文字入力を実現することを目的にする。

## 1.3 アプローチ

スマートウォッチを装着した手によるジェスチャを用いてアルファベット群を選択する。入力されたアルファベット群の組み合わせに対し単語予測アルゴリズムを用いることによって文字入力を行う。なお、スマートウォッチの加速度センサのデータに対し機械学習を用いることによって、スマートウォッチを装着した手によるジェスチャを識別する。

## 1.4 貢献

本研究の貢献を以下に示す。

- スマートウォッチのみを用いたセンシングに基づくジェスチャ検知アルゴリズムを開発した。
- スマートウォッチのみを用いたセンシングに基づくピンチ指およびフリック指識別システムを開発し、精度検証を行った。
- 片手におけるピンチジェスチャおよびフリックジェスチャに基づいた文字入力手法を示した。
- スマートウォッチのみを用いたセンシングに基づく片手文字入力手法の有用性を検証した。

## 1.5 本論文の構成

本論文の構成を示す。第1章では、本研究の背景、目的、アプローチ、貢献、および構成を示した。第2章では、本研究の関連研究を示す。第3章では、本手法の概要および実装を述べる。第4章では、ジェスチャ識別精度を評価するために行った実験を述べる。第5章では、

本手法の入力性能を評価するために行った実験を述べる。第6章では、本手法の課題、および改善点を議論し、今後の課題を示す。第7章では、本研究の結論を述べる。



## 第2章 関連研究

本研究において示す片手文字入力手法では，スマートウォッチの加速度センサのデータに対し機械学習を用いることによって，スマートウォッチを装着した手におけるハンドジェスチャを識別し，さらに単語予測アルゴリズムを用いて片手文字入力を行う．したがって，本章ではスマートウォッチにおける加速度センサを用いたハンドジェスチャ識別に関する研究，スマートウォッチにおける文字入力に関する研究，単語予測アルゴリズムを用いた文字入力に関する研究，およびスマートウォッチにおける片手操作の研究を示し，本手法との位置づけを述べる．

### 2.1 スマートウォッチの加速度センサを用いたジェスチャ識別

スマートウォッチの加速度センサを用いてジェスチャを識別する研究が行われている．Serendipity [11] は，市販のスマートウォッチに搭載された加速度センサおよびジャイロセンサを用いて，指をつまむ，叩く，こするなどの5つの指のジェスチャを識別した．ViBand [12] は，市販のスマートウォッチに搭載されている加速度センサのサンプリングレートを4000 Hzに引き上げることによって，フリック，クラップ，スクラッチ，タップなどのハンドジェスチャ識別を可能にした．Tapprint [13] は，市販のスマートウォッチの加速度センサおよびジャイロセンサのサンプリングレートを引き上げることによって，スマートウォッチを装着した手の甲における被タップ位置の識別および個人認証を実現した．ViFin [14] は，市販のスマートウォッチの加速度センサおよびジャイロセンサを用いて，指による連続的な筆記を識別した．TapSkin [13] は，市販のスマートウォッチの加速度センサ，ジャイロセンサおよびマイクを用いて，スマートウォッチの周囲の皮膚上における最大11種類のタップジェスチャを識別した．iDial [15] は，市販のスマートウォッチの加速度センサ，ジャイロセンサおよびマイクを用いて，スマートウォッチを装着した手の甲における被タップ位置を識別した．また，ユーザからの最小限のデータ収集のみを用いてジェスチャのカスタマイズを実現した研究 [16, 17] もある．

本研究は，サンプリングレートを引き上げた加速度センサのデータをもとに，スマートウォッチを装着した手における親指およびその他の指によるジェスチャの識別を行い，また識別されたジェスチャをもとに文字入力を行う．

## 2.2 スマートウォッチにおける文字入力手法

スマートウォッチにおける文字入力手法が研究されている。本節においては、スマートウォッチにおける代表的な文字入力手法であるソフトウェアキーボードを活用した文字入力手法を述べる。また、本手法ではハンドジェスチャを用いて文字入力を行っている。そのため、ハンドジェスチャを用いた文字入力手法についても述べる。

### 2.2.1 ソフトウェアキーボードを活用した文字入力手法

ソフトウェアキーボードを活用した多くの研究は、2段階のジェスチャを用いている。ZoomBoard [18]では、キーボードの一部を拡大するためのタップ、および目的のキーを選択するためのタップという2段階のタップを用いて、小さなキーのタップを確実にしている。Swipeboard [19]では、9つのグループに分かれたキー群の1つを8方向のフリックおよびタップにより選択し、その後左右のフリックおよびタップによりキーを選択する。SplitBoard [20]では、キーボードを左右にフリックすることによって、3つの領域に分けられたキーボードの表示領域を切り替える。SwipeKey [21]では、複数の領域に分けられたキー群の1つをタップにより選択し、タップ後のスワイプ方向によってキーを選択する。ZShift [22] および Flickey [23]では、画面をタップしている指によって遮蔽されているキー周辺を、キーボード上部に表示することによって Fat Finger 問題 [2] を解決している。ZShift [22]では、画面をタップダウン後、タッチアップによってキーを選択する。Flickey [23]では、画面をタップダウン後、タッチアップもしくは上下方向へのフリックによってキーを選択する。

また、キーの選択およびキーの表示を工夫した文字入力手法が存在する。Driftboard [24]では、固定されたカーソル点に対して、移動可能な QWERTY キーボードを動かし、カーソル点にキーを合わせることによってキー選択を行う。Force9 [25]では、圧力を感知できるタッチスクリーンを用いて、3つのアルファベットが割り当てられた1つのキーの押し分けを可能にした。Invisiboard [26]では、透明な T9 キーボードを画面全体に配置する。そのため、キーのサイズが大きくなることによって入力の制約が解決され、キーボード用の表示領域を確保する必要がなくなることによって出力の制約が解決された。

また、解読器を用いる手法も提案されている。VelocciWatch [27]では、固有名詞のような予測が困難な単語の入力のための仮想キーボードのデザインを行った。WatchWriter [28]では、統計的解読を用いてジェスチャタイピングを可能にした。Yiら [29]は、2 cm から 4 cm のキーボードにおいてベイズ解読器を用いることによって、26.8 WPM から 33.6 WPM を達成することを示した。

しかし、スマートウォッチにおいてソフトウェアキーボードを用いるためには、スマートウォッチを装着していない手を用いてスマートウォッチを操作する必要があるため、両手が必要となる。本手法では、スマートウォッチを装着した手のみを用いることによって、片手のみを用いた文字入力を可能にする。

## 2.2.2 ハンドジェスチャを用いた文字入力手法

スマートウォッチにおけるソフトウェアキーボードを活用した文字入力手法は、多くの場合、画面専有面積と入力精度がトレードオフの関係にある。そのため、タッチを伴わない文字入力手法が研究されている。

両手のハンドジェスチャを用いて文字を入力する研究がある。BiTipText [30] では、両手の人差し指に取り付けた静電容量センサを用いて、両手における親指から人差し指へのタップ位置を検出し、文字入力を行う。InclineType [3] では、手首を傾けることによって文字を選択し、タッチスクリーンをタップすることによって文字を入力する。

また、スマートウォッチを装着した手のみを用いて文字入力を行う研究が存在する。FingerT9 [4] では、スマートウォッチを装着した手における指の 11 個の節に取り付けられた静電容量センサを用いて、親指から指のセグメントへのタッチジェスチャを識別することにより、文字入力を行う。WrisText [6] では、測距センサおよびピエゾ振動センサを用いて、手首のジョイスティックのような動きを識別することにより、文字入力を行う。Back-Hand-Pose [7] では、手首に取り付けたカメラを用いた手形状推定に基づきジェスチャ識別を行うことにより文字入力を行う。しかし、これらの手法にはスマートウォッチ以外の追加のデバイスが必要である。

スマートウォッチを装着していない手のみを用いて文字入力を行う手法も研究されている。TipText [5] では、人差し指の指の腹に取り付けられた静電容量センサを用いて、親指から人差し指の指の腹へのタップ位置を検出することにより、文字入力を行う。ThumbText [31] では、人差し指に取り付けられた静電容量センサを用いて、親指から人差し指へのタップ及びスワイプを検出することにより、文字入力を行う。しかし、これらの手法にも追加のデバイスが必要である。

また、スマートウォッチを装着した手を用いて空中などに描いた文字を、スマートウォッチのセンサのみを用いて識別する片手文字入力手法 [8, 9, 10] も存在する。本研究は、これらの研究と同様にスマートウォッチのセンサのみを用いて片手文字入力を行う。しかし、スマートウォッチを装着した手における親指および他の指によるジェスチャを用いて、文字を入力する。

## 2.3 単語予測アルゴリズムを用いた文字入力手法

PinchType [32] および FingerText [33] では、親指から他の指へのジェスチャを用いてアルファベット群を選択し、選択された連続のアルファベット群の組み合わせに対し、単語を予測して提示する。単語 n-gram モデルを用いて、単語、フレーズ、およびセンテンスベースの文字入力解読をしている研究 [34, 35, 36] もある。TapType [37] では、ベイズニューラルネットワーク分類器から得られるジェスチャの確率と、n-gram 言語モデルから得られる文字の事前確率を組み合わせ、最も可能性の高い文字列を予測している。Optimal-T9 [38] は、曖昧入力における単語衝突（同じタップ順序の単語の衝突）を考慮してキー配置のデザインを行った。

本研究では、片手の親指から小指を用いるアルファベット群の選択、およびアルファベット群の組み合わせから単語を予測するという手法を用いて文字入力を行う。上記の研究とは異なり、入力された文字数以上の単語も予測単語として提示する。

## 2.4 スマートウォッチにおける片手操作

スマートウォッチを装着した手のみを用いて、片手にてスマートウォッチを操作するという研究がある。本節では文字入力以外のスマートウォッチにおける片手操作の研究について述べる。Kerberら [39] は、外出時などの、スマートウォッチを装着していない方の手によるインタラクションを妨げる状況において、スマートウォッチを装着した手におけるインタラクションが、スマートウォッチのインタラクションの真の代替手段となることを示した。GestureWrist [40] は、腕に装着したデバイスを用いて、デバイスを装着した手のジェスチャを識別する最初の研究の1つであり、静電容量センサおよび加速度センサを用いてセンシングを行っている。WristWhirl [41] は、手首の周りに取り付けられた測距センサを用いて手首によるジョイスティック操作を識別することにより、スマートウォッチにおける片手操作を可能にした。Float [42] は、photoplethysmogram 信号、加速度センサ、およびジャイロセンサによって検出される手首の傾きおよび空中における指のタップを組み合わせて、片手対象選択を可能にした。Guoら [43] は、スマートウォッチの加速度センサおよびジャイロセンサを用いて、スマートウォッチにおけるタッチをしない手首のみによるインタラクションを提案した。DigitSpace [44] は、手の解剖学的構造およびタッチ精度という2つの人間工学的要因を考慮した、親指から他の指への操作を行う磁気センサを用いたインタフェースであり、快適なインタラクションが可能な指の領域を探索した。SynchroWatch [45] は、パッシブ磁気リングが装着された親指、および画面上の点滅操作のリズミカルな相関を利用して片手操作を行った。SeeSaw [46] は、触覚と視覚のプロンプトに同期してスマートウォッチを装着した手首を回転させているかどうかに応じて入力を行う手法である。

これらの研究では、スマートウォッチにおいて片手のみを用いた操作は行っているが、スマートウォッチを装着した手における親指とピンチジェスチャを行う指およびフリックジェスチャを行う指の識別は行っていない。また、文字入力は行っていない。本研究では、スマートウォッチを装着した手における、親指とその他の指のピンチジェスチャおよびフリックジェスチャの識別を行い、それに基づいて片手のみを用いた文字入力を行う。

## 第3章 スマートウォッチの加速度センサのデータに基づく片手文字入力手法

本手法はスマートウォッチを装着した手におけるジェスチャを用いて文字入力を行う手法であり、スマートウォッチの加速度センサのデータに対する機械学習を用いてハンドジェスチャの識別を行う。本章では文字入力システムとしての概要、および実装を述べる。

### 3.1 概要

ユーザは単語を入力するために、図 3.1 に示す 5 種類のピンチジェスチャを用いて、目的の単語が表示されるまでアルファベット群の選択を繰り返し行う。ピンチジェスチャとアルファベット群の対応を図 3.1 に示す。アルファベット群は QWERTY キーボード配列を 2 列ずつ分割したものである。システムは、入力された連続のジェスチャに基づき、単語予測アルゴリズムを用いて予測単語を 5 つ表示する。次にユーザは、図 3.2 に示す 5 種類のフリックジェスチャを用いて提示された予測単語の中から入力したい単語を選択する。フリックジェスチャと予測単語の選択の対応を図 3.2 に示す。なお、手を握るジェスチャを用いて undo 操作を行うことができる。ユーザが本手法を用いている様子を図 3.3 に示す。

スマートウォッチにおいて文字入力を行うためのユーザインタフェース（以下、UI）（図 3.4）を作成した。指に割り当てられているキーを確認することができるように、キー割り当てを一番上に表示した。gestures 欄には、1 つの単語を入力するために行ったジェスチャが表示される。suggestion 欄には、入力されたジェスチャに基づく予測単語が表示される。sentence 欄

ジェスチャに対応する文字群	Q W A S Z X	E R D F C V	T Y G H B N	U I J K M	O P L
ジェスチャ	(a)	(b)	(c)	(d)	(e)

図 3.1: ピンチジェスチャおよびそれに対応するアルファベット群。





予測単語の 選択	1番目	2番目	3番目	4番目	5番目
ジェスチャ					

図 3.2: フリックジェスチャおよびそれに対応して選択される予測単語.



図 3.3: ユーザが本手法を用いている様子.

には、入力した複数の単語である文章が表示される。

例えば「software」という単語を入力したい場合、図3.4の初期状態から、小指のピンチジェスチャ（図3.1 (a)）を用いて「s」という文字が含まれる「Q, A, Z, W, S, X」アルファベット群を選択する。すると「and」、「a」、「with」、「are」、「at」という予測単語が提示される（図3.5）。次に、親指のピンチジェスチャ（図3.1 (e)）を用いて「o」という文字が含まれる「O, L, P」アルファベット群を選択する。すると「all」、「so」、「also」、「would」、「some」という予測単語が提示される（図3.6）。次に、薬指のピンチジェスチャ（図3.1 (b)）を用いて「f」という文字が含まれる「E, D, C, R, F, V」アルファベット群を選択する。すると「would」、「work」、「software」、「special」、「april」という予測単語が提示されるので（図3.7）、中指のフリックジェスチャ（図3.2 (c)）を用いて3番目の予測単語である「software」を選択する（図3.8）。なお、単語間には自動的にスペースが挿入される。ユーザは上記の操作を繰り返すことにより文章を入力する。図3.9に文字入力の順序を示す。

以上のように、画面のタップではなくスマートウォッチを装着した手のジェスチャにより文字入力を行うことによって、入力の制約を解決し、片手のみを用いた入力を可能にする。さらに本手法は、キーとジェスチャの割り当てを記憶することによりキーボードを表示する必要がなくなる。これによって、キーボードの画面専有という出力の制約を解決する。

## 3.2 実装

本節では、スマートウォッチの加速度センサのデータに基づく片手文字入力手法を実現するための実装を述べる。

### 3.2.1 システム構成

システム構成を図3.10に示す。PCにおいてスマートウォッチの加速度センサのデータを取得するために、スマートウォッチおよびAndroidスマートフォンをBluetooth接続にてペアリングし、AndroidスマートフォンおよびPCをUSBにて接続する。ジェスチャ識別処理および単語予測処理をPCにおいて行い、文字入力状態に関するデータをWebSocket通信を介してスマートフォンに送信し、更にそのデータをスマートウォッチに送信する。

### 3.2.2 リアルタイムジェスチャ識別

リアルタイムに文字入力を行うためにジェスチャ識別処理（図3.11）を連続して行う。1回のジェスチャ識別処理には約0.03秒要するので、結果として約0.03秒に1度ジェスチャ識別処理を行う。図3.12にリアルタイムジェスチャ識別処理において扱うデータの流れを示す。ジェスチャ識別処理が行われたら、データを約0.03秒分ずらし再度ジェスチャ識別処理を行う。

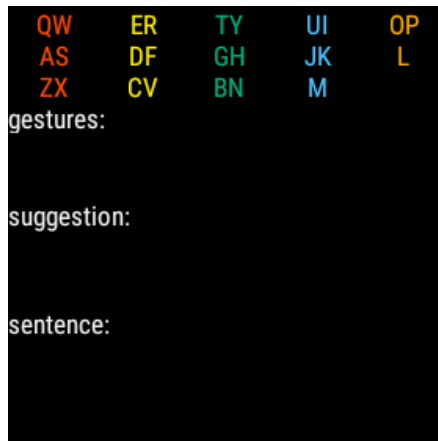


図 3.4: 文字入力 UI の初期状態.

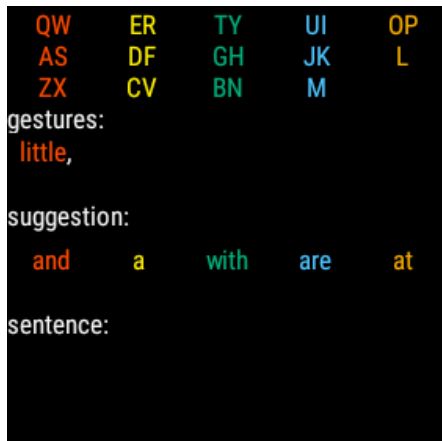


図 3.5: 初期状態 (図 3.4) から小指のピンチジェスチャ (図 3.1 (a)) を行った後の状態.

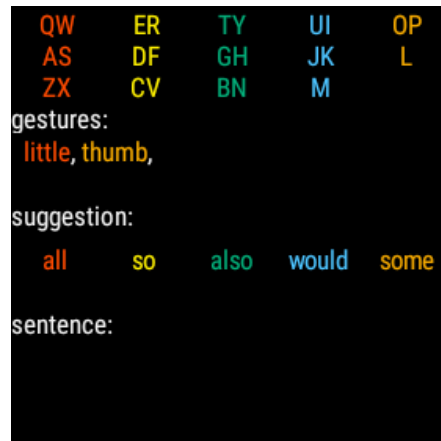


図 3.6: 図 3.5 の状態から親指のピンチジェスチャ (図 3.1 (e)) を行った後の状態.

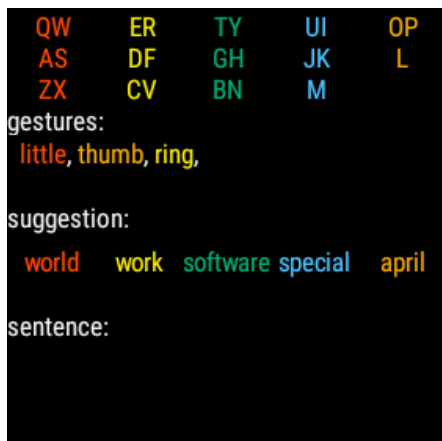


図 3.7: 図 3.6 の状態から薬指のピンチジェスチャ (図 3.1 (b)) を行った後の状態.

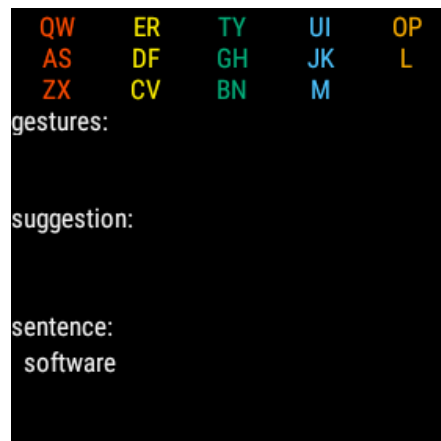


図 3.8: 図 3.7 の状態から中指のフリックジェスチャ (図 3.2 (c)) を行った後の状態.



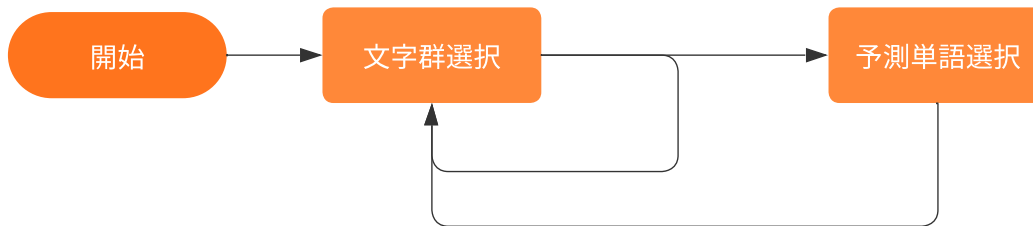


図 3.9: 文字入力の順序.

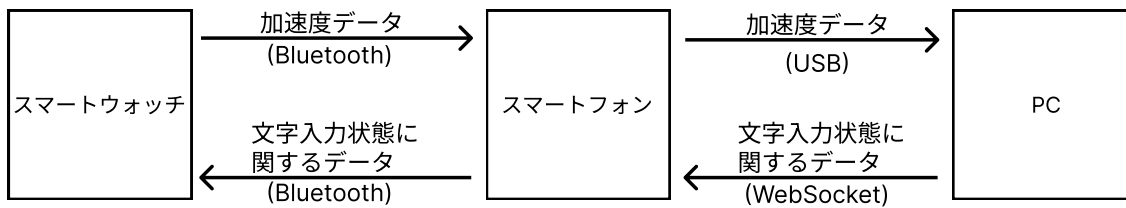


図 3.10: 実装におけるシステムの構成. スマートウォッチとスマートフォンは, Bluetooth にて加速度データおよび文字入力状態に関するデータの通信を行う. スマートフォンと PC は USB 接続にて加速度データの通信を行い, WebSocket を用いて文字入力状態に関するデータの通信を行う.

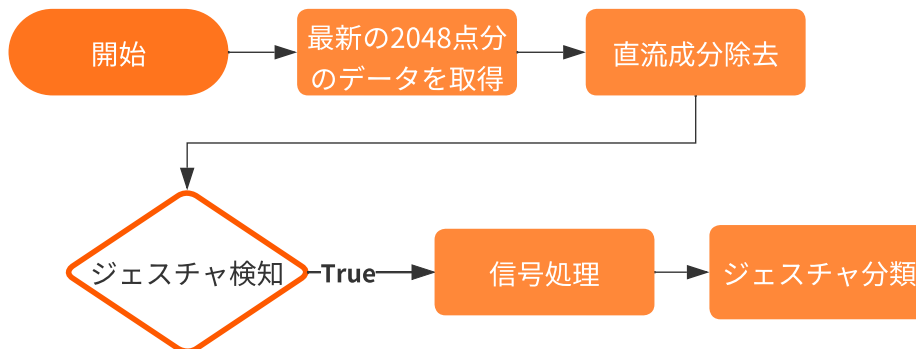


図 3.11: リアルタイムジェスチャ識別処理の流れ.

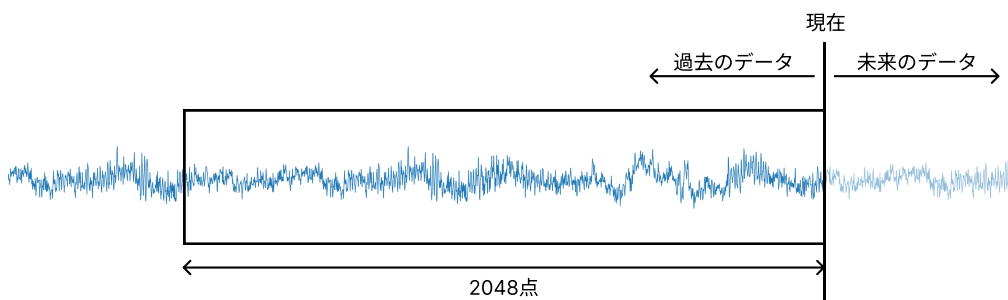


図 3.12: リアルタイムジェスチャ識別処理において扱うデータの流れ.

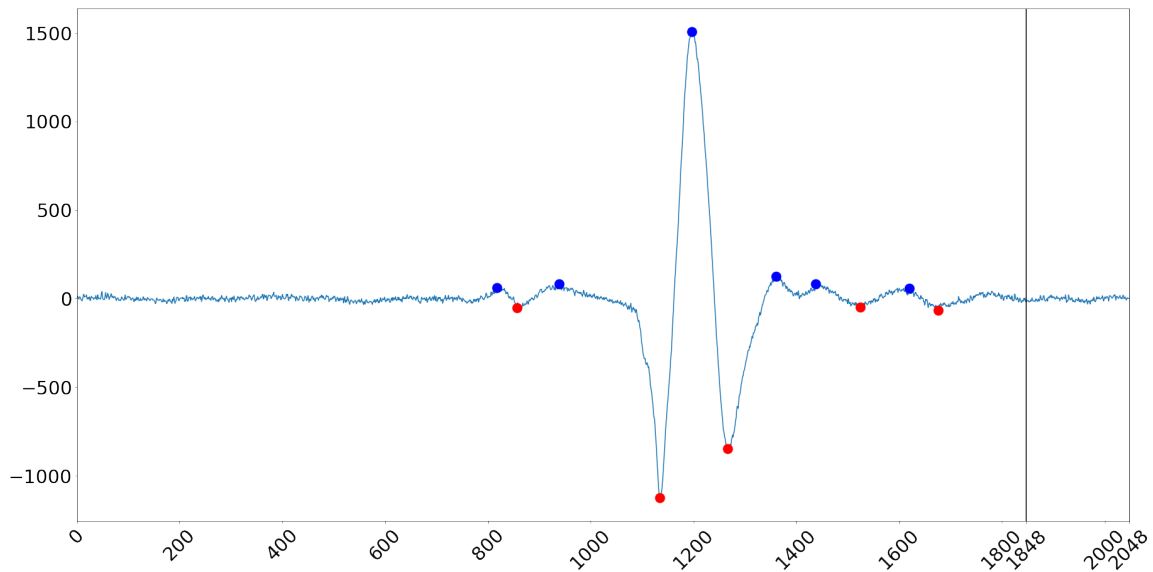


図 3.13: 検知されるジェスチャの Y 軸の波形の例. 赤色の点が負の値のピーク, 青色の点が正の値のピークを示す. 1848 点から 2048 点の間 (末尾 200 点) にピークが存在するときは, ジェスチャは検知されない.

**最新の 2048 点分のデータ取得** サンプリングレートが 4000 Hz の 3 軸 (X, Y および Z 方向の 3 つ) 分の加速度センサのデータそれぞれにおいて, 処理が行われる時点において最も新しい 2048 点 (約 0.512 秒) 分のデータを取得する. 本手法にて行うジェスチャは約 0.512 秒以下であるため, 2048 点のデータはジェスチャ 1 回分のデータを含む.

**直流成分除去** 加速度センサのデータにおける変化を計測するために, 取得した 3 軸ごとの 2048 点のデータに対し, それぞれバターワースフィルタを用いて直流成分 (重力加速度成分) を除去する.

**ジェスチャ検知** 直流成分を除去した Y 軸の 2048 点のデータに対し, ピーク検知アルゴリズムを用いてピークを検知する. ピーク検知アルゴリズムに用いた, ピークの絶対値, プロミネンス (地形突出度), およびピーク間の距離は, それぞれ経験的に 45, 78, および 50 に決定した. 得られたピークをもとにジェスチャ検知を行う [47]. 正の値のピーク, および負の値のピークがそれぞれ 2 つ以上検知されたときにジェスチャを検知する.

また, 意図しないジェスチャを検知しないため, ジェスチャの途中で検知を行わないため, およびスマートウォッチを装着した手の動きが落ち着いている場合に行われたジェスチャのみを検知するために, 2048 点の末尾 200 点 (約 0.05 秒) にピークが含まれる場合はジェスチャを検知しないものとする. 検知されたジェスチャの Y 軸の波形の例を図 3.13 に示す.

なお, 本処理は約 0.03 秒に 1 回行われるため, 同一のジェスチャから生じる波形が複数回処理にかけられ, 意図するジェスチャの後の手の微細な動きも処理にかけられる. この結果,

同一のジェスチャを複数回検知する可能性があるため、ジェスチャの検知後約 0.5 秒間はジェスチャ検知を行わない。

**信号処理** ジェスチャが検知された場合に、検知されたジェスチャのデータに対し信号処理を行う。抽出された 3 軸それぞれのジェスチャの波形データに対し、窓関数としてハミング窓をかけ、その後高速フーリエ変換を用いて周波数パワースペクトルに変換し、0 Hz からサンプリングレートのナイキスト周波数である 2000 Hz までの 1025 点のデータを得る。

**ジェスチャ分類** 信号処理をしたデータから、ジェスチャの特性を表す 0 Hz から 128 Hz までの 67 点を抜き出し、3 軸合わせて 201 点の特徴量に対して、機械学習を用いて分類を行う。分類するジェスチャは図 3.1 (a, b, c, d, e), 図 3.2 (a, b, c, d, e), および undo 操作のための手を握る動作の 11 種類である。

### 3.2.3 単語予測アルゴリズム

識別された連続のジェスチャをもとに単語予測を行う。1 つのジェスチャには 3-6 個のアルファベット（アルファベット群）が割り当てられており、入力された連続のジェスチャに対応する連続のアルファベット群に対するすべての組み合わせを計算し、複数の文字列を得る。それらの文字列から始まるすべての単語の使用頻度をコーパス [48] を用いて検索し、上位 5 つを降順に予測単語として表示する。

## 第4章 実験1：ジェスチャ識別精度の調査

5種類のピンチジェスチャ(図3.1), 5種類のフリックジェスチャ(図3.2), およびundo操作のための手を握る動作の11種類のジェスチャの識別精度を調査するための実験を行った.

### 4.1 実験機器

実験に用いたスマートウォッチはLG G Watch W100(端末サイズ:高さ37.9mm×幅46.5mm×厚さ9.95mm, 画面サイズ:1.65インチ)であり, InvenSense MPU6515という慣性計測ユニット(以下, IMU)を搭載している. これは, 多くのスマートウォッチに搭載されているIMUと同一である. なお, ViBand [12]と同じくデバイスのLinuxカーネルを改造し, 加速度センサのサンプリングレートを4000Hzに引き上げている. ジェスチャ識別処理および単語予測処理を行うPCはMacBook Air (M1, 2020), PCとスマートウォッチの仲介を行うスマートフォンにはGoogle Pixel 4を用いた.

### 4.2 データ収集

著者1人に対し各ジェスチャ50個, 計550個のデータ収集を1セットとし, これを4セット繰り返した. 各セットにおいては約0.5秒に1回ジェスチャを行うことを50回繰り返し, 次のジェスチャへ移行した. なお, 再装着時の識別精度を測るために, セット間においてスマートウォッチの再装着を行った. また, 腕に装着したスマートウォッチの画面を見ながら文字入力を行うことを想定し, データ収集時のスマートウォッチは画面が上向きになるようにした. スマートウォッチは非利き手である左手に装着した.

### 4.3 機械学習アルゴリズム比較

収集したすべてのデータに対し, 8割を訓練データ, 残り2割をテストデータとして用いて, 機械学習アルゴリズムの比較を行った. 機械学習アルゴリズムの比較にはlazypredict<sup>1</sup>を用いた. 結果を表4.1に示す. なお, 適合率とは正と予測したデータのうち実際に正であるものの割合, 再現率とは実際に正であるもののうち正であると予測されたものの割合, 正解率とは全予測に対する正答の割合であり, F1値とは適合率と再現率の調和平均である. 表4.1より, SGDClassifierの正解率, およびF1値が最も高かった. また所要時間は0.21秒であった.

<sup>1</sup><https://github.com/shankarpandala/lazypredict>

Model	Accuracy	Balanced Accuracy	F1 Score	Time Taken
SGDClassifier	0.97	0.98	0.98	0.21
LogisticRegression	0.97	0.97	0.97	0.29
LGBMClassifier	0.97	0.97	0.97	5.71
CalibratedClassifierCV	0.97	0.97	0.97	2.97
ExtraTreesClassifier	0.97	0.97	0.97	0.21
RandomForestClassifier	0.96	0.96	0.96	0.92
SVC	0.96	0.96	0.96	0.28
PassiveAggressiveClassifier	0.96	0.96	0.96	0.19
LinearSVC	0.96	0.96	0.96	0.68
KNeighborsClassifier	0.95	0.95	0.95	0.02
RidgeClassifierCV	0.95	0.95	0.95	0.38
BaggingClassifier	0.93	0.93	0.93	1.05
NuSVC	0.93	0.93	0.93	0.59
LinearDiscriminantAnalysis	0.91	0.92	0.91	0.35
RidgeClassifier	0.91	0.91	0.91	0.09
DecisionTreeClassifier	0.90	0.90	0.90	0.29
Perceptron	0.89	0.89	0.89	0.13
GaussianNB	0.87	0.88	0.87	0.02
NearestCentroid	0.85	0.86	0.86	0.04
ExtraTreeClassifier	0.79	0.80	0.79	0.02
BernoulliNB	0.74	0.74	0.75	0.04
LabelPropagation	0.62	0.61	0.62	0.07
LabelSpreading	0.62	0.61	0.62	0.17
AdaBoostClassifier	0.23	0.22	0.16	1.49
QuadraticDiscriminantAnalysis	0.12	0.13	0.10	0.15
DummyClassifier	0.07	0.09	0.01	0.01

表 4.1: 機械学習アルゴリズムの比較.

	1回目のセットのデータ	2回目のセットのデータ	3回目のセットのデータ	4回目のセットのデータ
検証1	訓練データ	訓練データ	訓練データ	テストデータ
検証2	訓練データ	訓練データ	テストデータ	訓練データ
検証3	テストデータ	テストデータ	訓練データ	訓練データ
検証4	テストデータ	訓練データ	訓練データ	訓練データ

表 4.2: セット間の交差検証の詳細.

## 4.4 セット間交差検証および5分割交差検証

収集したデータに対し、表 4.2 に示すセット間の交差検証、および5分割交差検証を行った。前節の結果より機械学習アルゴリズムには SGDClassifier を用いた。

### 4.4.1 結果

評価指標には、混同行列、正解率、適合率、再現率、および F1 値を用いた。混同行列とは実際のラベルを Y 軸に、予測したラベルを X 軸にして、分類結果をまとめたものである。

セット間の交差検証の混同行列を図 4.1a, 図 4.1b, 図 4.1c, 図 4.1d, および図 4.2 に示す。セット間の交差検証の平均の正解率は 0.96, 適合率は 0.96, 再現率は 0.96, F1 値は 0.96 であった。5分割交差検証の混同行列を図 4.3 に示す。5分割交差検証平均の正解率は 0.98, 適合率は 0.98, 再現率は 0.98, F1 値は 0.98 であった。なお、混同行列において手を握る動作を grasp, その他は「指の名前\_ジェスチャの種類名」として示している。

### 4.4.2 考察

図 4.1a より、小指のフリックジェスチャを薬指のフリックジェスチャと間違える可能性が高いということが分かる。しかし、図 4.1b, 図 4.1c, および図 4.1d では、その確率は高くない。そのため、4回目のセットにおいて行った小指のフリックジェスチャの手の動きが、他のセットにおいて行った薬指のフリックジェスチャの手の動きと類似していたと考えられる。図 4.2 および図 4.3 では、上述した間違いの確率は高くないことから、セット間においてジェスチャの動作が変化することがあっても、異なる動作を含めて学習をすればどちらの動作も正しく分類することができると考えられる。



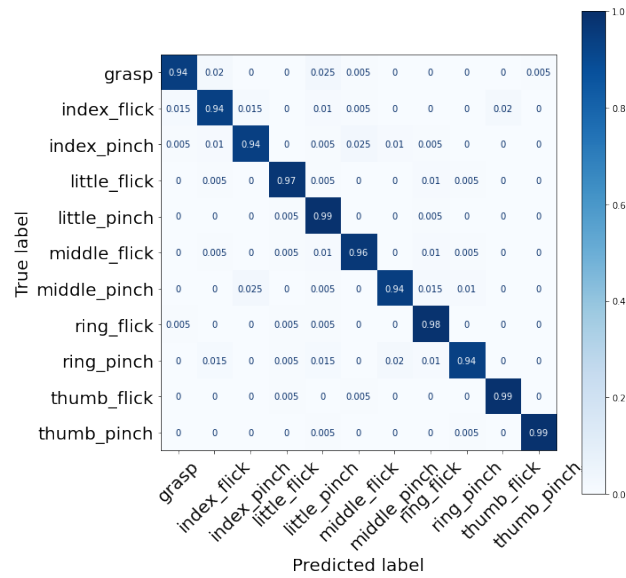


図 4.2: すべてのセット間の交差検証の平均の混同行列.

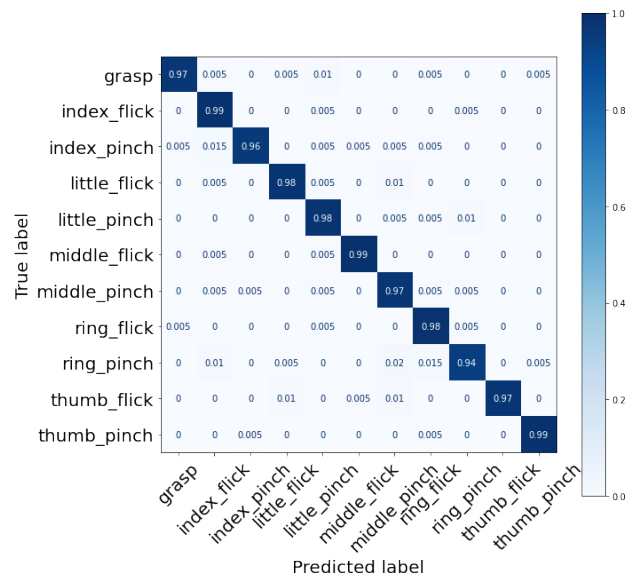


図 4.3: 5 分割交差検証の混同行列.



## 第5章 実験2：文字入力性能の調査

本研究にて提案する手法の文字入力手法としての性能を評価するため、著者1人に対し、文字入力速度および誤入力率を計測する実験を行った。実験機器は実験1と同一である。ジェスチャ識別に使用した機械学習モデルとしては、実験1にて収集したデータすべてを訓練データとして、SGDClassifierを用いて学習したものを用いた。

### 5.1 評価指標

評価に用いる指標には、文字入力速度を測定するための指標である Words Per Minute（以下、WPM）[49]、および文字入力の正確性を測定するための指標である Character Error Rate（以下、CER）を用いた。

WPMは、式5.1のとおりに定義される。

$$WPM = \frac{|T| - 1}{S} \times 60 \times \frac{1}{5} \quad (5.1)$$

式中における  $|T|$  は最終的に入力された合計の文字数（スペースを含む）、 $S$  は1文字目が入力されてから最後の単語が入力されるまでの時間（秒）である。なお、英単語において、1単語はスペース、数字、および他の印字可能文字も含めて平均5文字であるため、5分の1をかける。これを1フレーズずつ計測し、全フレーズのWPMの平均を算出する。

CERは、最終的に入力されたフレーズと入力するべきフレーズ間のレーベンシュタイン距離を、入力するべきフレーズの文字数で割ることによって正規化したものである。

### 5.2 実験手順

実験においては5個のフレーズを順番に提示し、文字入力を行った。実験におけるフレーズセットには、MacKenzieら[50]によって提供されている英語のフレーズセット（表5.1）を用いた。このフレーズセットは500種類の英語のフレーズであり、フレーズ内の単語と単語の間は空白文字にて区切られており、句読点および数字を含まない。また、本手法では小文字しか入力できないため、大文字を小文字に変換した。このフレーズセットから無作為に選んだ5個のフレーズを、被験者である著者が1つずつ入力した。学習効果を抑えるために、入力するフレーズは重複しないようにした。なお、入力するフレーズの記憶強度が結果に影響を与えないように、図5.1のように入力するべきフレーズを常に被験者に提示した。

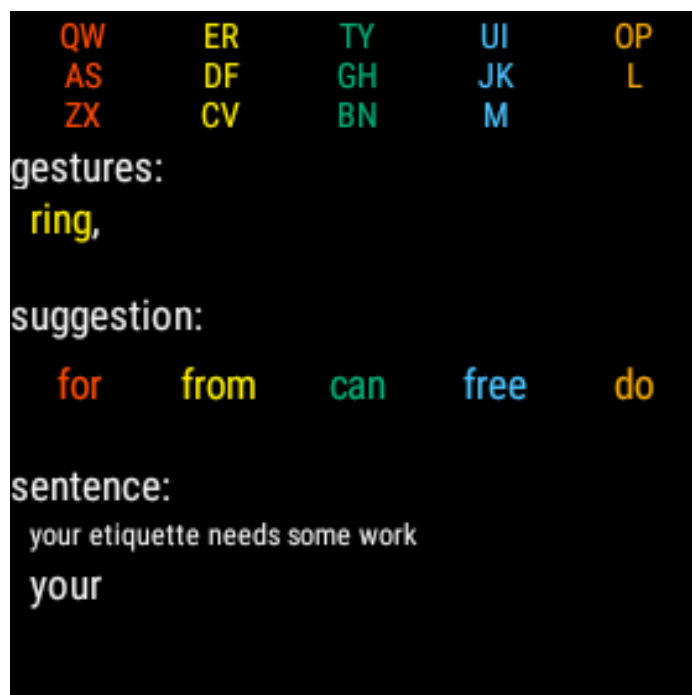


図 5.1: 文字入力速度実験用のユーザインタフェース。ユーザは sentence の下に表示されているフレーズを入力する。入力されたフレーズが一番下に表示される。

ID	フレーズ	文字数
1	my watch fell in the water	26
2	prevailing wind from the east	29
3	never too rich and never too thin	33
4	breathing is difficult	22
5	I can see the rings on Saturn	29
6	physics and chemistry are hard	30
7	my bank account is overdrawn	28
8	elections bring out the best	28
9	we are having spaghetti	23
10	time to go shopping	19

表 5.1: 実験に用いたジェスチャセットの一部。本手法では小文字しか入力できないため、大文字は小文字にて代替した。

1つのフレーズの入力が終了したら、スマートウォッチにペアリングされているスマートフォンの「入力完了」ボタンを押した。なお本手法は片手文字入力手法であるため、本手法においてメッセージアプリケーションなどにおける送信に対応するジェスチャを行う事が可能、かつフレーズの入力完了操作もそれと同一のジェスチャを用いて行えることが望ましい。しかし、その操作が誤って識別される可能性がある。そこで本実験においては正しく文字入力速度を計測するために、スマートフォンのボタンを用いてフレーズの入力完了操作を行うことにした。「入力完了」ボタンが押されると、スマートウォッチにおける現在のフレーズおよび入力した文字が消え、次に入力すべきフレーズが提示される。入力を間違えた場合およびジェスチャが誤識別された場合は、手を握る動作に割り当てられたundo機能を用いて訂正をした。また、なるべく高速かつ正確に文字入力を行った。単語と単語の間は自動的に空白を挿入した。

本番セッションの前に、本番に用いたフレーズ以外で練習をした。また、スマートウォッチは非利き手に装着した。

### 5.3 結果

WPMは4.19、CERは0.0280であった。

### 5.4 考察

WPMを既存のスマートウォッチにおける片手文字入力手法と比較する。WrisText [6]では9.9 WPM、FingerT9 [4]では3.43 WPMであると報告されている。本手法は追加のセンサを用いていないという点でこれらの手法と異なるが、2.36倍および0.82倍の違いがある。本手法では、機械学習を用いてジェスチャの識別を行っているため、ジェスチャが誤って識別される可能性がある。そのため、ジェスチャを行った際にそのジェスチャが正しく識別されたかを確認する必要がある。したがって、ジェスチャの識別結果を確認せずに、複数のジェスチャを素早く連続して行うことができない。さらに、ジェスチャが誤識別された場合は訂正および再入力をする必要がある。また、ジェスチャを行ってからそのジェスチャが識別されスマートウォッチ上に反映されるまでに約0.4秒の遅延（ジェスチャを行う様子を録画することによって算出）がある。加えて、本手法では入力文字数以上の単語を予測単語として提示している。そのため、アルファベット群選択のジェスチャを行うたびに、目的の単語が予測単語として提示されているかを確認する必要がある。以上の要因により、文字入力速度が低下したと考えられる。また、本手法はQWERTY配列を用いているが、物理キーボードを用いたタイピングのようにキーを両手の人差し指から小指に割り当ててのではなく、キーを片手の親指から小指に割り当てているため、直感的にどのキーがどの指に対応するかがわからない。本実験は本手法に慣れている著者1人が行ったが、本手法に慣れていないユーザが用いた場合、目的のアルファベットがどの指に割り当てられているかを確認する動作にも時間がかかると思われる。

次に CER について考察する。CER は入力したフレーズと入力すべきフレーズが同一であれば 0 になる。得られた CER が 0 でないのは、入力すべきフレーズに含まれる単語に入力できなかったものが存在したからである。本手法は予測単語を 5 つしか提示できない。そのため、コーパス [48] における使用頻度が低いものは予測単語として提示されず、入力することができなかった。

また、腕を水平な状態に保ったまま文字入力を行うことは、ゴリラアーム効果 [51]（空中のインタラクションにおける腕の疲労）につながるため、作業負荷が高かった。さらに、行ったジェスチャが正しく識別されず文字入力を円滑に行えないことはストレスに感じられた。

## 第6章 議論

本章では、実験の結果に基づいて本手法の問題点、改善案、応用に関する議論を行い、今後の課題を述べる。

### 6.1 文字入力性能、ユーザビリティ、および作業負荷の調査

本研究にて提案した手法に対し、著者1人に対する文字入力性能を調査する実験は行ったが、複数人を対象とした文字入力性能、ユーザビリティ、および作業負荷に関する調査は行っていない。本手法が実際に有用であるかを検証するため、今後複数人を対象とした実験を行う。

また、文字入力性能を測る指標としてCERを用いた。しかし、CERは誤って入力され修正されなかったエラーを判別することができるが、誤って入力され修正されたエラーを判別することはできない。そのため、今後 Total Error Rate [52]、または Corrected Error Rate [52] という指標を採用し、文字入力性能を測る実験を行う。

### 6.2 ジェスチャ識別の改良

本手法のジェスチャ識別には様々な課題が存在する。それらの課題および改善案を述べる。

#### 6.2.1 ジェスチャ検知処理の改良

図3.13に示す様に、学習に用いるジェスチャの波形に、ジェスチャを行った後に生じる微細な動きを含めている。そのため、ジェスチャを行ってから一定時間が経過しないとジェスチャ検知処理が完了せず、ジェスチャが識別されない。それに伴い、5.4節で述べたジェスチャを行ってからそのジェスチャが識別されスマートウォッチ上に反映されるまでの遅延が生じる。したがって、ジェスチャが終了した瞬間にジェスチャ検知を行えば、遅延が軽減される。例えば、3.2.2節にて述べた意図しないジェスチャを検知しないための処理を削除することが考えられる。多くの場合、学習に用いるデータを減らすことは分類精度とトレードオフの関係にあるが、分類精度を保ったままジェスチャ検知の応答性を向上させることが課題である。

## 6.2.2 データ収集の改良

本研究におけるデータ収集は、ジェスチャ前後の手の動きが落ち着いていることを前提として行った。しかし、実際の文字入力場面においては、複数のジェスチャを連続して行いたいことがある。その際は、ジェスチャ前後の手の動きが落ち着いているとは限らない。そのため、実際の文字入力場面を想定してデータ収集を行い、それに合わせてジェスチャ識別処理を工夫する必要がある。

また、本手法では、ユーザごとに多量のデータ収集を行う必要がある。今後、few-shot learning [16, 17] 等の手法を用いることによって、ユーザごとのデータ収集を最小限にする。

## 6.3 文字入力方法の改良

提案した文字入力方法の課題、改善案を述べる。

### 6.3.1 予測単語の提示

本手法では、文字数が多い単語の入力を素早くするために、入力文字数以上の単語を含めて予測単語として提示している。しかし、それによって文字を入力する度に入力したい単語が予測単語に提示されているかを確認する必要がある。そのため、入力に要する時間が増加する。ジェスチャの応答性および正確性が向上すれば、多くのジェスチャを素早く行えるようになる。したがって、入力文字数以上の単語を予測単語として提示せずとも、文字数が多い単語の入力が素早く行えるようになると考えられる。また、そうした場合、予測単語として提示される単語候補が少なくなる。よって、入力したい単語が予測単語に表示される確率が高くなる。今後、ジェスチャの応答性および正確性に応じた最適な予測単語の提示方法、および予測単語の提示方法が入力速度に与える影響を調査する。

### 6.3.2 コーパスに含まれない単語の入力

本手法ではコーパスに含まれる単語が予測単語として提示され入力される。したがって、コーパスに含まれない単語を入力することができない。しかし、例えば、同一の指のピンチジェスチャを連続して行った回数を用いてその指に割り当てられた特定のアルファベットの選択および入力を行えば、コーパスに含まれない単語の入力が可能になる。今後、上記の文字入力方法の実装および上記の文字入力方法を用いたユーザ実験を行う。

### 6.3.3 予測単語の選択

本手法では、予測単語の選択に5つの指のフリックジェスチャを用いている。しかし、新たにアルファベット群選択の終了操作を行う1つのジェスチャを採用すれば、予測単語の選

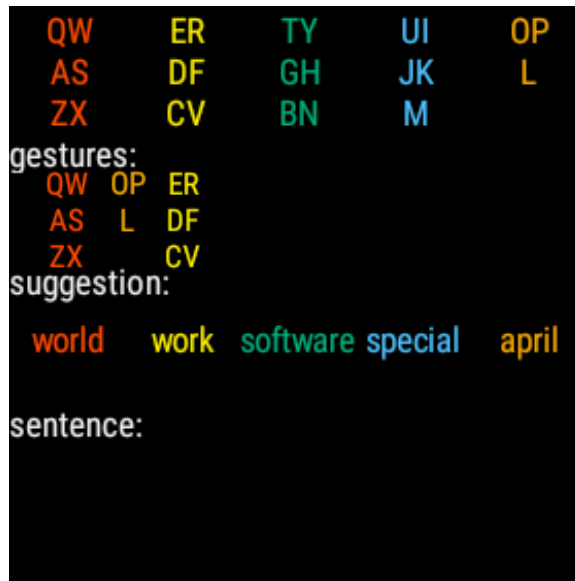


図 6.1: 図 3.7 における行われたジェスチャの表示を改良したもの。ユーザは「software」という単語を入力しようとしている。

択を指のピンチジェスチャにて行うことが可能になる。したがって、指のフリックジェスチャを使用する必要がなくなり、使用ジェスチャ数が減少する。それに伴い、誤識別の減少、およびデータ収集の所要時間の減少が見込まれる。今後、最適な予測単語の選択方法を調査する。

### 6.3.4 予測単語の切り替え

本手法は予測単語を5つしか提示できず、コーパス [48] における使用頻度が低いものは予測単語として提示されず、入力することができない。したがって、提示される予測単語を切り替えるジェスチャを新たに採用する必要がある。

### 6.3.5 文字入力インタフェース

図 3.5, 図 3.6, および図 3.7 に示すように、行ったジェスチャの識別結果のフィードバックとして、文字入力インタフェースに行ったジェスチャ名を表示した。しかし、ジェスチャ名とアルファベット群の対応を直感的に把握することが難しいため、ジェスチャ名の羅列から、現在入力している文字列を把握することが難しい。そのため、行われたジェスチャ名を羅列するのではなく、アルファベット群の羅列を示すインタフェース (図 6.1) に変更する。

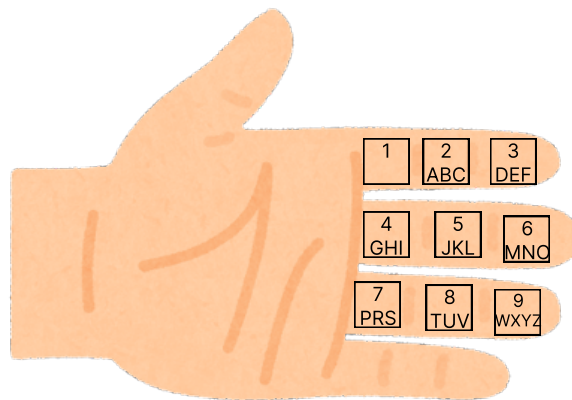


図 6.2: 片手における T9 キーボードの例.

### 6.3.6 ジェスチャセットおよびそれに伴うキーボードレイアウト

本手法の入力に用いるジェスチャ、およびそれに対応するキーボードのレイアウトは著者の主観によって決定されたものである。そのため、現在のジェスチャセットおよびキーボードレイアウトが最適とは限らない。例えば親指による他の指のセグメントの押下ジェスチャを識別することが可能であれば、片手における T9 キーボード式の文字入力 (図 6.2) [4] が可能になる。今後、識別可能なジェスチャの調査、および速度、正確性、快適性、および単語予測精度を考慮したキー割り当て [5] の調査を行う。

## 6.4 本手法の応用

スマートウォッチのみを用いてセンシングを行う文字入力手法である本手法は、スマートウォッチと Head Mounted Display (以下, HMD) を接続することにより、VR 環境および AR 環境 (以下, MR 環境) においても用いることができる。MR 環境においては、ハンドトラッキング技術を用いることにより、手の動作のみを用いて操作をする場面が増えてきた。しかし MR 環境において手の動作のみを用いて文字入力を行うことには、触覚フィードバックがない、HMD のハンドトラッキングの範囲外にてジェスチャを行うことができない、およびゴリラアーム効果 [51] が発生するという問題が伴う。腕を下げた状態にて本研究にて提案した手法を用いることによって、それらの問題を解決することができる。今後、スマートウォッチにおける文字入力手法である本手法を、MR 環境における文字入力に応用する。

## 6.5 ユビキタス環境における使用に向けて

本手法では、ジェスチャ識別システムおよび単語予測システムを PC にて動作させた。しかし、スマートウォッチはウェアラブルデバイスであり、実際にスマートウォッチを用いる際は、PC を用いることができない場合が多い。そのため、ジェスチャ識別システムおよび単語



予測システムをスマートウォッチとペアリングされているスマートフォンにて動作させる必要がある。また、本研究では、スマートウォッチの Linux カーネルを改造することによって、サンプリングレートが 4000 Hz の加速度センシングを可能にしている。しかし、4000 Hz にて加速度センシングを行うことはバッテリー消費が大きい。本手法では高速フーリエ変換後に得られたデータの 128 Hz までのデータのみを機械学習に用いているため、256 Hz の加速度センシングにて本手法は実現できると考えられる。今後、バッテリー消費を抑えるために、低いサンプリングレートにおけるジェスチャ識別を行う。

## 第7章 結論

本研究において、スマートウォッチのみを用いたセンシングに基づく片手文字入力手法を提案し、ジェスチャ識別精度および文字入力性能の調査を行った。本手法では、スマートウォッチを装着した手におけるジェスチャを、スマートウォッチの加速度センサに対する機械学習を用いて識別し、さらに単語予測アルゴリズムを組み合わせることで文字入力を行った。これによって、キーボードを画面に表示することによる出力の制約、Fat Finger 問題、および両手使用が不可能な場合において文字入力不可能という問題を解決した。また、実験を通して、スマートウォッチを再装着してもジェスチャ識別が可能であること、および4.19 WPMの文字入力速度を達成することが分かった。今後は、複数人を対象とした本手法の文字入力性能、ユーザビリティ、および作業負荷の調査、ジェスチャ識別の改良、最適な文字入力方法の調査、およびユビキタス環境において使用可能にするを行う。

## 謝辞

本研究を進めるにあたって、志築文太郎先生、高橋伸先生、川口一画先生には多大なご意見とご指導を頂きました。心から感謝申し上げます。特に志築文太郎先生には、研究の進め方、および論文執筆の仕方をはじめとして、研究に関することを一からご指導いただきました。さらに、研究の相談や研究生活において、多くの助言および指南をいただきました。ここに深く感謝の意を表します。

インタラクティブプログラミング研究室の同輩、先輩方には研究生活において大変お世話になりました。WAVE チームの皆様には、チームゼミにおけるご意見や、論文の添削だけではなく、研究室において多くの助言を頂きました。深く感謝いたします。特に平井李音氏には研究方針および論文執筆など、本研究における重要な部分において多くのアドバイスを頂き大変お世話になりました。重ねて感謝申し上げます。

最後に、学生生活においてお世話になった皆様、そして金銭的および精神的に私の学生生活を支えてくださった家族に深く感謝申し上げます。

# 著者論文リスト

## 本論文に関する論文

- 査読なし国内会議論文
  1. 角田陸, 平井李音, 志築文太郎. スマートウォッチのセンサのみを用いたハンドジェスチャ認識に基づく片手文字入力手法の提案. 第30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, 日本ソフトウェア科学会, 2022年12月, 3 pages.

## その他の論文

- 査読なし国内会議論文
  1. 小西智樹, 角田陸, 國分晴利, 久保勇貴, 志築文太郎. アクティブ音響センシングを用いた手指の関節角度推定. 情報処理学会研究報告 (HCI-201), 情報処理学会, 2023年1月, 5 pages.

## 参考文献

- [1] Ahmed Sabbir Arif and Ali Mazalek. A survey of text entry techniques for smartwatches. In Masaaki Kurosu, editor, *Human-Computer Interaction. Interaction Platforms and Techniques*, pp. 255–267, Cham, 2016. Springer International Publishing.
- [2] Katie A. Siek, Yvonne Rogers, and Kay H. Connelly. Fat Finger Worries: How older and younger users physically interact with PDAs. In *Proceedings of the 2005 IFIP TC13 International Conference on Human-Computer Interaction*, INTERACT '05, pp. 267–280, Berlin, Heidelberg, 2005. Springer-Verlag.
- [3] Timo Götzelmann and Pere-Pau Vázquez. InclineType: An accelerometer-based typing approach for smartwatches. In *Proceedings of the XVI International Conference on Human Computer Interaction*, Interacción '15, pp. 1–4, New York, NY, USA, 2015. Association for Computing Machinery.
- [4] Pui Chung Wong, Kening Zhu, and Hongbo Fu. FingerT9: Leveraging thumb-to-finger interaction for same-side-hand text entry on smartwatches. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pp. 1–10, New York, NY, USA, 2018. Association for Computing Machinery.
- [5] Zheer Xu, Pui Chung Wong, Jun Gong, Te-Yen Wu, Aditya Shekhar Nittala, Xiaojun Bi, Jürgen Steimle, Hongbo Fu, Kening Zhu, and Xing-Dong Yang. TipText: Eyes-free text entry on a fingertip keyboard. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, UIST '19, pp. 883–899, New York, NY, USA, 2019. Association for Computing Machinery.
- [6] Jun Gong, Zheer Xu, Qifan Guo, Teddy Seyed, Xiang ‘Anthony’ Chen, Xiaojun Bi, and Xing-Dong 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, pp. 1–14, New York, NY, USA, 2018. Association for Computing Machinery.
- [7] Erwin Wu, Ye Yuan, Hui-Shyong Yeo, Aaron Quigley, Hideki Koike, and Kris M. Kitani. Back-Hand-Pose: 3D hand pose estimation for a wrist-worn camera via dorsum deformation network. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, UIST '20, pp. 1147–1160, New York, NY, USA, 2020. Association for Computing Machinery.

- [8] Chao Xu, Parth H. Pathak, and Prasant Mohapatra. Finger-Writing with Smartwatch: A case for finger and hand gesture recognition using smartwatch. In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, HotMobile '15, pp. 9–14, New York, NY, USA, 2015. Association for Computing Machinery.
- [9] Yi Gao, Siyu Zeng, Ji Zhao, Wenxin Liu, and Wei Dong. AirText: One-handed text entry in the air for cots smartwatches. *IEEE Transactions on Mobile Computing*, 2021.
- [10] Xinye Lin, Yixin Chen, Xiao-Wen Chang, Xue Liu, and Xiaodong Wang. SHOW: Smart handwriting on watches. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, Vol. 1, No. 4, 2018.
- [11] Cong Shi, Vasileios Lakafosis, Mostafa H. Ammar, and Ellen W. Zegura. Serendipity: Enabling remote computing among intermittently connected mobile devices. In *Proceedings of the Thirteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '12, pp. 145–154, New York, NY, USA, 2012. Association for Computing Machinery.
- [12] Gierad Laput, Robert Xiao, and Chris Harrison. ViBand: High-fidelity bio-acoustic sensing using commodity smartwatch accelerometers. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, pp. 321–333, New York, NY, USA, 2016. Association for Computing Machinery.
- [13] Wenqiang Chen, Lin Chen, Yandao Huang, Xinyu Zhang, Lu Wang, Rukhsana Ruby, and Kaishun Wu. Taprint: Secure text input for commodity smart wristbands. In *The 25th Annual International Conference on Mobile Computing and Networking*, MobiCom '19, pp. 1–16, New York, NY, USA, 2019. Association for Computing Machinery.
- [14] Wenqiang Chen, Lin Chen, Meiyi Ma, Farshid Salemi Parizi, Shwetak Patel, and John Stankovic. ViFin: Harness passive vibration to continuous micro finger writing with a commodity smartwatch. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, Vol. 5, No. 1, pp. 1–25, 2021.
- [15] Maotian Zhang, Qian Dai, Panlong Yang, Jie Xiong, Chang Tian, and Chaocan Xiang. IDial: Enabling a virtual dial plate on the hand back for around-device interaction. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, Vol. 2, No. 1, 2018.
- [16] Xuhai Xu, Jun Gong, Carolina Brum, Lilian Liang, Bongsoo Suh, Shivam Kumar Gupta, Yash Agarwal, Laurence Lindsey, Runchang Kang, Behrooz Shahsavari, Tu Nguyen, Heriberto Nieto, Scott E Hudson, Charlie Maalouf, Jax Seyed Mousavi, and Gierad Laput. Enabling hand gesture customization on wrist-worn devices. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA, 2022. Association for Computing Machinery.

- [17] Naoki Kimura. Self-supervised approach for few-shot hand gesture recognition. In *Adjunct Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, UIST '22 Adjunct, pp. 1–25, New York, NY, USA, 2022. Association for Computing Machinery.
- [18] Stephen Oney, Chris Harrison, Amy Ogan, and Jason Wiese. ZoomBoard: A diminutive QWERTY soft keyboard using iterative zooming for ultra-small devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pp. 2799–2802, New York, NY, USA, 2013. Association for Computing Machinery.
- [19] Xiang ‘Anthony’ Chen, Tovi Grossman, and George Fitzmaurice. Swipeboard: A text entry technique for ultra-small interfaces that supports novice to expert transitions. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, pp. 615–620, New York, NY, USA, 2014. Association for Computing Machinery.
- [20] Jonggi Hong, Seongkook Heo, Poika Isokoski, and Geehyuk Lee. SplitBoard: A simple split soft keyboard for wristwatch-sized touch screens. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pp. 1233–1236, New York, NY, USA, 2015. Association for Computing Machinery.
- [21] Yuan-Fu Shao, Masatoshi Chang-Ogimoto, Reinhard Pointner, Yu-Chih Lin, Chen-Ting Wu, and Mike Chen. SwipeKey: A swipe-based keyboard design for smartwatches. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '16, pp. 60–71, New York, NY, USA, 2016. Association for Computing Machinery.
- [22] Luis A. Leiva, Alireza Sahami, Alejandro Catala, Niels Henze, and Albrecht Schmidt. Text entry on tiny QWERTY soft keyboards. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pp. 669–678, New York, NY, USA, 2015. Association for Computing Machinery.
- [23] Akira Ishii, Hiroyuki Hakoda, and Buntarou Shizuki. Flickey: Flick-based qwerty software keyboard for ultra-small touch screen devices. In Masaaki Kurosu, editor, *Human-Computer Interaction. Interaction Technologies*, pp. 31–42, Cham, 2018. Springer International Publishing.
- [24] Tomoki Shibata, Daniel Afergan, Danielle Kong, Beste F. Yuksel, I. Scott MacKenzie, and Robert J.K. Jacob. DriftBoard: A panning-based text entry technique for ultra-small touchscreens. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, pp. 575–582, New York, NY, USA, 2016. Association for Computing Machinery.

- [25] Lik Hang Lee, Ngo Yan Yeung, Tristan Braud, Tong Li, Xiang Su, and Pan Hui. Force9: Force-assisted miniature keyboard on smart wearables. In *Proceedings of the 2020 International Conference on Multimodal Interaction, ICMI '20*, pp. 232–241, New York, NY, USA, 2020. Association for Computing Machinery.
- [26] Aske Mottelson, Christoffer Larsen, Mikkel Lyderik, Paul Strohmeier, and Jarrod Knibbe. Invisiboard: Maximizing display and input space with a full screen text entry method for smartwatches. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services, MobileHCI '16*, pp. 53–59, New York, NY, USA, 2016. Association for Computing Machinery.
- [27] Keith Vertanen, Dylan Gaines, Crystal Fletcher, Alex M. Stanage, Robbie Watling, and Per Ola Kristensson. VelociWatch: Designing and evaluating a virtual keyboard for the input of challenging text. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, CHI '19*, pp. 1–14, New York, NY, USA, 2019. Association for Computing Machinery.
- [28] Mitchell Gordon, Tom Ouyang, and Shumin 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*, pp. 3817–3821, New York, NY, USA, 2016. Association for Computing Machinery.
- [29] Xin Yi, Chun Yu, Weinan Shi, and Yuanchun Shi. Is it too small?: Investigating the performances and preferences of users when typing on tiny qwerty keyboards. *International Journal of Human-Computer Studies*, Vol. 106, pp. 44–62, 2017.
- [30] Zheer Xu, Weihao Chen, Dongyang Zhao, Jiehui Luo, Te-Yen Wu, Jun Gong, Sicheng Yin, Jialun Zhai, and Xing-Dong Yang. BiTipText: Bimanual eyes-free text entry on a fingertip keyboard. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, CHI '20*, pp. 1–13, New York, NY, USA, 2020. Association for Computing Machinery.
- [31] Junhyeok Kim, William Delamare, and Pourang Irani. ThumbText: Text entry for wearable devices using a miniature ring. In *Proceedings of the 44th Graphics Interface Conference, GI '18*, pp. 18–25, Waterloo, CAN, 2018. Canadian Human-Computer Communications Society.
- [32] Jacqui Fashimpaur, Kenrick Kin, and Matt Longest. PinchType: Text entry for virtual and augmented reality using comfortable thumb to fingertip pinches. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems, CHI EA '20*, pp. 1–7, New York, NY, USA, 2020. Association for Computing Machinery.
- [33] DoYoung Lee, Jiwan Kim, and Ian Oakley. FingerText: Exploring and optimizing performance for wearable, mobile and one-handed typing. In *Proceedings of the 2021 CHI Conference on*



- Human Factors in Computing Systems*, CHI '21, New York, NY, USA, 2021. Association for Computing Machinery.
- [34] Keith Vertanen, Crystal Fletcher, Dylan Gaines, Jacob Gould, and Per Ola Kristensson. The impact of word, multiple word, and sentence input on virtual keyboard decoding performance. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pp. 1–12, New York, NY, USA, 2018. Association for Computing Machinery.
- [35] Keith Vertanen, Haythem Memmi, Justin Emge, Shyam Reyal, and Per Ola Kristensson. VelociTap: Investigating fast mobile text entry using sentence-based decoding of touchscreen keyboard input. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pp. 659–668, New York, NY, USA, 2015. Association for Computing Machinery.
- [36] Mingrui Ray Zhang and Shumin Zhai. PhraseFlow: Designs and empirical studies of phrase-level input. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, pp. 1–13, New York, NY, USA, 2021. Association for Computing Machinery.
- [37] Paul Strelti, Jiayi Jiang, Andreas Rene Fender, Manuel Meier, Hugo Romat, and Christian Holz. TapType: Ten-finger text entry on everyday surfaces via bayesian inference. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, pp. 1–16, New York, NY, USA, 2022. Association for Computing Machinery.
- [38] Ryan Qin, Suwen Zhu, Yu-Hao Lin, Yu-Jung Ko, and Xiaojun Bi. Optimal-T9: An optimized t9-like keyboard for small touchscreen devices. In *Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces*, ISS '18, pp. 137–146, New York, NY, USA, 2018. Association for Computing Machinery.
- [39] Frederic Kerber, Markus Löchtefeld, Antonio Krüger, Jess McIntosh, Charlie McNeill, and Mike Fraser. Understanding same-side interactions with wrist-worn devices. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction*, NordiCHI '16, New York, NY, USA, 2016. Association for Computing Machinery.
- [40] Jun Rekimoto. GestureWrist and GesturePad: unobtrusive wearable interaction devices. In *Proceedings Fifth International Symposium on Wearable Computers*, pp. 21–27. IEEE, 2001.
- [41] Jun Gong, Xing-Dong Yang, and Pourang Irani. WristWhirl: One-handed continuous smart-watch input using wrist gestures. *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, 2016.
- [42] Ke Sun, Yuntao Wang, Chun Yu, Yukang Yan, Hongyi Wen, and Yuanchun Shi. Float: One-handed and touch-free target selection on smartwatches. In *Proceedings of the 2017 CHI*

- Conference on Human Factors in Computing Systems*, CHI '17, p. 692–704, New York, NY, USA, 2017. Association for Computing Machinery.
- [43] Anhong Guo and Tim Paek. Exploring tilt for no-touch, wrist-only interactions on smartwatches. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '16, p. 17–28, New York, NY, USA, 2016. Association for Computing Machinery.
- [44] Da-Yuan Huang, Liwei Chan, Shuo Yang, Fan Wang, Rong-Hao Liang, De-Nian Yang, Yi-Ping Hung, and Bing-Yu Chen. DigitSpace: Designing thumb-to-fingers touch interfaces for one-handed and eyes-free interactions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pp. 1526–1537, New York, NY, USA, 2016. Association for Computing Machinery.
- [45] Gabriel Reyes, Jason Wu, Nikita Juneja, Maxim Goldshtein, W. Keith Edwards, Gregory D. Abowd, and Thad Starner. SynchroWatch: One-handed synchronous smartwatch gestures using correlation and magnetic sensing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, Vol. 1, No. 4, 2018.
- [46] Jason Wu, Cooper Colglazier, Adhithya Ravishankar, Yuyan Duan, Yuanbo Wang, Thomas Ploetz, and Thad Starner. SeeSaw: Rapid one-handed synchronous gesture interface for smartwatches. In *Proceedings of the 2018 ACM International Symposium on Wearable Computers*, ISWC '18, pp. 17–20, New York, NY, USA, 2018. Association for Computing Machinery.
- [47] Michael Xuelin Huang, Yang Li, Nazneen Nazneen, Alexander Chao, and Shumin Zhai. TapNet: The design, training, implementation, and applications of a multi-task learning cnn for off-screen mobile input. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, 2021. Association for Computing Machinery.
- [48] English word frequency. <https://www.kaggle.com/datasets/ratatman/english-word-frequency>. 最終参照日：2023年1月30日。
- [49] Ahmed Sabbir Arif and Wolfgang Stuerzlinger. Analysis of text entry performance metrics. In *2009 IEEE Toronto International Conference Science and Technology for Humanity (TIC-STH)*, pp. 100–105, 2009.
- [50] I. Scott MacKenzie and R. William Soukoreff. Phrase sets for evaluating text entry techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '03, pp. 754–755, New York, NY, USA, 2003. Association for Computing Machinery.
- [51] Sebastian Boring, Marko Jurmu, and Andreas Butz. Scroll, Tilt or Move It: Using mobile phones to continuously control pointers on large public displays. In *Proceedings of the 21st*

*Annual Conference of the Australian Computer-Human Interaction Special Interest Group: Design: Open 24/7, OZCHI '09*, pp. 161–168, New York, NY, USA, 2009. Association for Computing Machinery.

- [52] R. William Soukoreff and I. Scott MacKenzie. Metrics for text entry research: An evaluation of msd and kspc, and a new unified error metric. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '03*, pp. 113–120, New York, NY, USA, 2003. Association for Computing Machinery.