

筑波大学大学院博士課程

システム情報工学研究科修士論文

手首装着型端末における  
片手操作のための指動作認識手法

青山 周平

修士（工学）

（コンピュータサイエンス専攻）

指導教員 志築 文太郎

2018年3月

## 概要

本論文は、スマートウォッチと呼ばれる手首装着型端末において、片手のみを用いた操作を可能とするための指動作認識手法を示す。スマートウォッチの多くは、腕時計の文字盤にあたる部分にタッチディスプレイが組み込まれており、これが時刻の表示のみならず、様々な機能を提供することを可能としている。しかし、スマートウォッチへの入力操作においてタッチディスプレイを用いることは「指が画面を隠してしまう」、「スマートウォッチ装着側とは逆側の手を用いる必要がある（片手操作できない）」という問題を引き起こす。そこで、本研究はタッチディスプレイを用いないスマートウォッチへの入力として親指動作を用いることに着目する。本研究では、人差し指側面上にて動かされた親指の移動量を用いてスマートウォッチ入力を行うことを実現する。これに際し、本研究は手首装着型のセンサデバイスを製作し、親指動作に伴う手首の形状変化を測定する実験を行った。また、実験をもとに親指動作認識システムを実装した。本論文においては、実装した親指動作認識システムに関して行った評価実験について述べるとともにアプリケーション例を示し、親指動作を用いたスマートウォッチ操作の可能性を示す。

# 目次

<b>第1章</b>	<b>序論</b>	<b>1</b>
1.1	スマートウォッチとその操作における問題点	1
1.2	本研究の目的とアプローチ	2
1.3	貢献	3
1.4	本論文の構成	3
<b>第2章</b>	<b>関連研究</b>	<b>5</b>
2.1	ウェアラブルデバイスにおける手／指動作認識手法に関する研究	5
2.1.1	腕に取り付けたセンサを用いた認識手法	5
2.1.2	手首に取り付けたセンサを用いた認識手法	5
2.1.3	手（指，甲）に取り付けたセンサを用いた認識手法	7
2.2	スマートウォッチの操作手法に関する研究	7
2.2.1	スマートウォッチの縁や周辺の皮膚への入力を用いた操作手法	7
2.2.2	スマートウォッチ周辺の空中における手の動作を用いた操作手法	8
2.2.3	スマートウォッチ装着側の手による動作を用いた操作手法	8
2.3	本研究の位置付け	9
<b>第3章</b>	<b>スマートウォッチ操作に用いる入力動作</b>	<b>10</b>
3.1	親指スライドジェスチャを用いた次元入力	10
3.2	決定操作に用いる入力動作	13
3.3	認識原理	14
<b>第4章</b>	<b>予備実験：親指スライドジェスチャに伴う手首形状変化の測定</b>	<b>16</b>
4.1	実験協力者	16
4.2	実験機器	16
4.2.1	センサデバイス	16
4.2.2	グローブ	18
4.3	実験設計	18
4.3.1	準備	18
4.3.2	手順	18
4.3.3	記録方法	20
4.4	結果と考察	20

4.4.1	手首の形状変化 . . . . .	20
4.4.2	親指移動量推定の精度検証 . . . . .	20
<b>第 5 章</b>	<b>親指動作認識システム</b>	<b>24</b>
5.1	認識ソフトウェア . . . . .	24
5.1.1	キャリブレーション . . . . .	24
5.1.2	データ加工 . . . . .	25
5.1.3	推定 . . . . .	26
	線形選択アルゴリズム . . . . .	26
	位置重み付けアルゴリズム . . . . .	26
5.2	認識ハードウェア . . . . .	28
<b>第 6 章</b>	<b>評価実験</b>	<b>30</b>
6.1	実験協力者 . . . . .	30
6.2	実験機器 . . . . .	30
6.3	実験設計 . . . . .	30
6.3.1	準備 . . . . .	30
6.3.2	手順 . . . . .	32
6.3.3	記録方法 . . . . .	32
6.4	実験結果 . . . . .	32
6.4.1	各条件におけるターゲット選択精度 . . . . .	32
6.4.2	各アルゴリズムにおけるポインタ移動の成功率 . . . . .	35
6.4.3	アンケート . . . . .	35
	決定動作について . . . . .	35
	親指動作認識アルゴリズムについて . . . . .	35
	本システムの使いやすさについて . . . . .	37
6.5	考察 . . . . .	38
6.5.1	親指動作認識システムに関する考察 . . . . .	38
6.5.2	本システムを用いたスマートウォッチ操作に関する考察 . . . . .	38
<b>第 7 章</b>	<b>アプリケーション例</b>	<b>40</b>
7.1	選択アプリケーション . . . . .	40
7.2	通知アプリケーション . . . . .	41
7.3	ゲームアプリケーション . . . . .	42
<b>第 8 章</b>	<b>親指動作の二次元方向認識への拡張</b>	<b>43</b>
8.1	親指二次元位置の識別精度評価 . . . . .	43
8.1.1	実験アプリケーション . . . . .	44
8.1.2	実験環境 . . . . .	44
8.1.3	実験手順 . . . . .	44

8.1.4	結果及び考察	46
8.2	二次元方向親指動作認識システム	46
8.2.1	システム構成	48
8.2.2	親指位置の学習	48
8.2.3	スマートウォッチ画面への表示	48
<b>第9章</b>	<b>議論</b>	<b>50</b>
9.1	決定操作の改良	50
9.2	適切なアルゴリズムの選択	50
9.3	入力動作の併用	50
9.4	入力状態と非入力状態の区別	50
9.5	日常生活における利用	51
9.6	手法の適用範囲について	51
<b>第10章</b>	<b>結論</b>	<b>52</b>
	謝辞	53
	参考文献	54
	著者論文リスト	60
付録A	アンケート用紙	62

# 目次

1.1	ターゲット選択操作をスマートウォッチのタッチディスプレイを用いて行う様子.	2
1.2	親指スライドジェスチャ.	3
3.1	親指スライドジェスチャを用いた絶対ポインティング. 人差し指上の親指の位置をスマートウォッチ画面におけるポインタ位置に当てはめることにより, ポインティング操作を行う.	11
3.2	親指スライドジェスチャを用いたスクロール操作. 親指の絶対ポインティングにおける中心座標からの距離をスクロールの速度として利用する.	11
3.3	親指スライドジェスチャを用いたスワイプ操作. 親指をすばやく動かすことによりスワイプを行う.	11
3.4	親指タップジェスチャ.	13
3.5	握りジェスチャ.	13
3.6	赤外線近接センサアレイを用いた手首の形状変化計測. それぞれのセンサは, センサと皮膚間の距離を測定する.	14
3.7	親指動作に関連する筋肉および腱.	15
4.1	実験に使用したベルト型センサデバイス.	17
4.2	実験に使用したベルト型センサデバイスの内部.	17
4.3	グローブに描かれた基準点とスライド点.	18
4.4	実験環境.	19
4.5	ベルト型センサデバイスと手の位置関係およびセンサ番号.	19
4.6	親指スライドジェスチャ時における各センサ値の変化量.	21
4.7	親指の移動量とセンサ値の変化量の相関係数.	22
4.8	親指移動実測値と推定値の誤差距離の平均.	23
5.1	収集したデータから各センサごとに 16 点のサンプル点から構成されるサンプル波形 $W_i$ を生成.	24
5.2	キャリブレーションにおけるセンサ値変化区間の移動. A) オフセット値 ( $offset_i$ ) を用いて, センサ値の生値 ( $R_i$ ) がサンプル波形の帯域内に収まるようにセンサ値を調整する. B) 調整後の推定用センサ値 $C_i$ がサンプル波形の帯域外に出る場合オフセット値 ( $offset_i$ ) を更新する.	25

5.3	各アルゴリズムにおける親指移動量の認識原理. グラフの横軸は親指の移動量, 縦軸はセンサ値を表している. A) 線形選択アルゴリズム. B) 位置重み付けアルゴリズム . . . . .	27
5.4	位置重み付けアルゴリズム. A) あるセンサに関してサンプル波形 $W_i$ と推定用センサ値 $C_i$ の交点を見つけ, そのサンプル点に対して重みをつける. この時, 交点における重みが山の頂点となるようにまわりのサンプル点についても重みをつける. B) あるセンサに関して交点が複数見つかった場合には, 交点の数だけ重みの山を小さくし, A と同様のことを行う. C) すべてのセンサに関する重みを合算し, 重みが最大となるサンプル点を親指の推定位置とする.	28
5.5	スマートウォッチベルト一体型センサデバイス. . . . .	29
5.6	スマートウォッチベルト一体型センサデバイスの内部. . . . .	29
5.7	スマートウォッチベルト一体型センサデバイスのセンサアレイ面. . . . .	29
6.1	実験環境: A) 実験時の姿勢, B) スマートウォッチに表示される画面. . .	31
6.2	決定動作時の加速度変化と親指スライドジェスチャ時の加速度変化の違い. .	31
6.3	決定動作によるポイントのずれ . . . . .	34
6.4	決定動作によるずれを考慮した際の各アルゴリズムにおけるターゲット選択精度 (エラーバーは標準誤差を表す). . . . .	34
6.5	各アルゴリズムにおけるポイント移動の成功率. . . . .	36
6.6	各アルゴリズムにおけるポイント移動の成功率. . . . .	37
6.7	実験結果をもとに設計した UI. A) 親指動作認識システムから推定した親指位置を 3 領域に分割. B) 親指の位置に応じてターゲットをスクロール. . . .	39
7.1	選択アプリケーション. . . . .	40
7.2	スクロールアプリケーション. . . . .	41
7.3	ゲームアプリケーション. . . . .	42
8.1	円形学習による二次元認識への応用イメージ. . . . .	43
8.2	識別対象とする親指の位置. . . . .	44
8.3	実験に使用したアプリケーション. . . . .	45
8.4	実験環境: A) 実験時の姿勢, B) センサデバイスにおける各センサの位置.	45
8.5	スマートフォンにおけるアプリケーション画面: A) 学習前, B) 学習後. .	49
8.6	スマートウォッチにおけるアプリケーション画面. . . . .	49

# 第1章 序論

本論文は、スマートウォッチと呼ばれる手首装着型端末において、片手のみを用いた操作を可能とするための指動作認識手法を示す。本章では、まず背景として現在のスマートウォッチの概要を示し、その操作方法の問題点を述べる。次に本研究の目的として、その問題を解決するためのアプローチを示す。その後、本研究の貢献を述べ、最後に本論文の構成を述べる。

## 1.1 スマートウォッチとその操作における問題点

スマートウォッチとは、腕時計を模して作られた手首装着型の情報端末である。スマートウォッチの多くは、腕時計の文字盤にあたる部分にタッチディスプレイが組み込まれており、これが時刻の表示のみならず、様々な機能を提供することを可能としている。例えば、Google の Android Wear [Incb] や Apple の Apple Watch [Inca] では、スマートフォンと連携することにより、メールおよび SNS<sup>1</sup> メッセージの着信通知の表示や、音楽プレーヤの遠隔操作などを可能にしている。このような日常生活において常に身に着けられることを想定した情報端末（ウェアラブルデバイス）は、端末を手で持つ必要がないため使用時に手がふさがらない点や、かばんやポケットから端末を出す必要がないためすぐに使用できる点において優れている。

しかし、タッチディスプレイを用いた入力は、スマートウォッチ操作において不都合となる場合がある。例えば、図 1.1 のようにタッチ対象が指に隠れて見えなくなることにより、正確な位置が指定できなくなるという問題が発生し得る。また、スマートウォッチに搭載されるタッチディスプレイが小さいがために、表示されるタッチ対象が指よりも小さくなり押し分けが困難になるという問題も発生し得る [SRC05]。さらに、タッチディスプレイを用いた操作は、図 1.1 のようにスマートウォッチ装着側とは逆側の手を用いてディスプレイを触れる必要がある。すなわち、スマートウォッチの片手操作が行えない。そのため、ユーザがかばんを持っている状態などにおいて、スマートウォッチを操作することを困難にする。

既存のスマートウォッチにおいても、タッチディスプレイを用いない入力操作方法がある。例えば、Android Wear [Incb]、および Apple Watch [Inca] においては、音声認識による入力操作が可能である。また、Android Wear [Incb] においては、前腕（スマートウォッチ）を回転させることによりスクロール操作を行うことが可能である。しかし、音声入力による操作には、「周りの音の影響を受ける」、「図書館などの声が出せない状況での使用が困難である」という問題がある。また、前腕の回転動作による操作には「操作時にディスプレイが視野から外れてしまう」、「比較的動作が大きくなる」という問題がある。

---

<sup>1</sup>ソーシャル・ネットワーキング・サービス (social networking service) の略称。





図 1.1: ターゲット選択操作をスマートウォッチのタッチディスプレイを用いて行う様子。

## 1.2 本研究の目的とアプローチ

本研究の目的は、スマートウォッチのタッチディスプレイ外に操作領域を拡張し、かつ片手操作を可能とする操作方法を実現することである。本研究は、そのためのアプローチとして、スマートウォッチ装着側の手による小さな動作（マイクロジェスチャ）に着目する。

本論文においてマイクロジェスチャとは、手首や腕を動かす必要がなく、指を小さく動かすことのみで行える動作のことを指す。このようなマイクロジェスチャをスマートウォッチ操作に用いることは、タッチディスプレイを用いない片手操作を実現することのみならず、様々な点において優れている。まず、ユーザが物を掴んだ状態において操作を行えるという利点がある。Wolfらは、ペンや車のハンドルを掴んだ状態で行えるマイクロジェスチャを定義、評価した [WNRM11]。また、Tanらは、自転車のハンドルを掴んだ状態で行えるマイクロジェスチャを評価した [TYR17]。これらの研究と同様に、適切なマイクロジェスチャを設計することにより、かばんなどを掴んだ状態においても入力操作を行うことが可能となる。また、操作に要する予備動作が小さいため、素早く操作に移行できるという利点もある。これら利点により、マイクロジェスチャは「使用時に手がふさがらない」、「すぐに使用できる」というウェアラブルデバイスの利点を損なうことのない入力操作を実現できる。さらに、操作の際に必要な動作が小さいため手が疲れにくい、人目をはばからずに操作が行えるという利点もある。

本研究においては、スマートウォッチ入力用のマイクロジェスチャとして親指を用いることに着目し、親指の動きを認識するシステムの開発および評価を行う。図 1.2 に本研究が提案、実現する親指によるマイクロジェスチャ「親指スライドジェスチャ」を示す。親指スライドジェスチャとは、スマートウォッチを装着した側の手の親指を人差し指の側面をなぞるように動かす動作のことを指す。ユーザはこのような動作を用いることにより、スマートウォッチへの一次元の連続的な入力操作が可能となる。

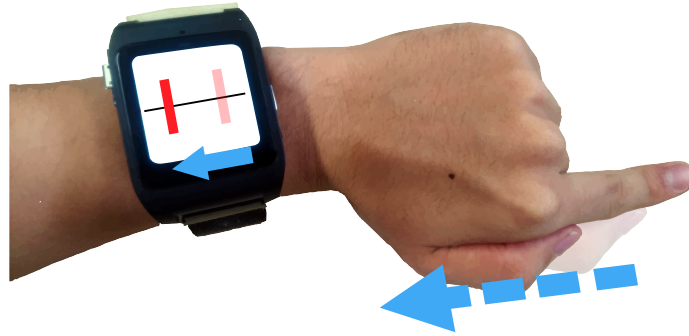


図 1.2: 親指スライドジェスチャ.

### 1.3 貢献

本研究の貢献を以下に列挙する.

- スマートウォッチの片手操作を可能とするための入力動作「親指スライドジェスチャ」および、これと併用して用いることができる決定操作の入力動作「親指タップジェスチャ」、「握りジェスチャ」を示した.
- 予備実験から得たデータをもとに、2種類の一次元方向の親指動作が認識可能なアルゴリズムを設計し、これを用いて親指動作認識システムを実装した.
- 簡易的に実装した決定動作認識システムを用いて、親指動作認識システムおよび親指スライドジェスチャによるスマートウォッチ操作の評価を行った.
- 親指スライドジェスチャを用いたスマートウォッチアプリケーション例を示した.
- 一次元方向の親指動作認識システムの発展として、二次元方向の親指動作を認識する方法を示した.

### 1.4 本論文の構成

本論文の構成は以下の通りである. 第2章では、本研究と関連している研究として「ウェアラブルデバイスにおける手/指動作認識手法に関する研究」および「スマートウォッチの操作手法に関する研究」を述べ、本研究の位置付けを示す. 第3章では、スマートウォッチの片手操作を実現するための入力動作を示し、その認識原理を述べる. 第4章では、親指スライドジェスチャの認識アルゴリズムを設計するために行った、親指動作に伴う手首形状変化の観察実験に関して述べる. 第5章では、実験をもとに開発した親指動作認識システムに関して述べる. 第6章では、開発したシステムに関する評価実験の実験設計を述べ、その結果を考察する. 第7章では、本研究で開発したシステムを用いたアプリケーション例に関して述

べる。第 8 章では，開発した親指動作認識システムの発展として，認識領域を二次元に拡張する方法を述べる。第 9 章では，本研究における議論点を述べる。最後に第 10 章では，結論を述べる。

## 第2章 関連研究

本研究は、スマートウォッチを装着した側の親指動作を認識することにより、スマートウォッチの片手操作を可能とすることを目的とした研究である。本研究と関連している研究として「ウェアラブルデバイスにおける手／指動作認識手法に関する研究」および「スマートウォッチの操作手法に関する研究」が挙げられる。本章では、それぞれの研究の概要を述べた後、本研究の位置付けを示す。

### 2.1 ウェアラブルデバイスにおける手／指動作認識手法に関する研究

ウェアラブルデバイスにおける手／指動作認識手法に関する研究は、様々な方法により行われている。それらの研究は、いずれも体のどこかにセンサを取り付ける必要があるため、センサを取り付ける位置により分類される。

#### 2.1.1 腕に取り付けたセンサを用いた認識手法

カメラを用いて手／指動作を認識する手法は数多く研究されている。例えば Han らは、腕に深度カメラを取り付けることによりスマートウォッチ上の空中における手の動きを認識することを試みた [HAL15]。

手／指動作の認識方法として腕に取り付けた筋電位センサを用いた研究も存在する。例えば, Hasan らは、前腕に取り付けた筋電位センサから得られるデータを解析し、手の姿勢の認識を行った [HRS<sup>+</sup>14]。また, Huang らは腕に取り付けた筋電位センサを用いて、親指の動きを認識するシステムを構築した [HZS<sup>+</sup>15]。さらに, Kerber らは腕に筋電位センサを搭載した腕装着型バンドである Myo [Incc] を取り付けることにより手の動きを認識した [KLK15]。

#### 2.1.2 手首に取り付けたセンサを用いた認識手法

手首に取り付けたカメラを用いて手／指動作を認識する試みは数多くなされている。例えば, Kim らは手の姿勢を認識するために、赤外線カメラおよび赤外線プロジェクタからなる手首装着型端末である Digits を開発した [KHI<sup>+</sup>12]。また, Prätorius らは親指とその他の指先とを叩き合わせる動作を認識するため、魚眼レンズカメラを備えた手首装着型端末である DigiTap を開発した [PVBH14]。

手首に取り付けた加速度センサを用いて手／指動作を認識する研究も数多く存在する．例えば、福本らは手首に装着した加速度センサを用いて、センサを装着した手の任意の指先で行った机や腕を指先で叩く動作を検出した [福本 99]．また、Xu らはスマートウォッチに内蔵された加速度センサおよびジャイロセンサにより腕および指の動きを認識した [XPM15]．同様に Laput らは、スマートウォッチに内蔵された加速度センサを用いて生体音響データを得ることにより、振動を伴う手の動きを認識するシステム ViBand を開発した [LXH16]．さらに Sun らは、スマートウォッチに内蔵された加速度およびジャイロセンサ、心拍センサ（PPG センサ）を用いることにて指先同士を叩き合わせる動作を認識した [SWY+17]．

手首のみに取り付けた筋電位センサを用いて手／指動作を認識する研究も存在する．McIntosh らは、筋電位センサと圧力センサを組み合わせることにより、手首のみに取り付けたセンサデバイスにおいて手の姿勢の認識を行った [MMF+16]．

手首に取り付けた焦電型赤外線センサ（PIR センサ）を用いて指動作を認識する研究も存在する．Gong らはスマートウォッチに取り付けた PIR センサを用いて指先から放熱される熱量を検出し、親指の細かな動きを認識する手法 Pyro を示した [GZZY17]．

手首に取り付けたミリ波レーダを用いて指動作を認識する研究も存在する．Lien らは、スマートウォッチに搭載可能な小型のミリ波レーダに基づいた指動作認識システムである Soli を開発した [LGK+16]．Soli は小型のチップをスマートウォッチに組み込むことにて動作し、スマートウォッチ周辺におけるマイクロジェスチャを認識する．手首に取り付けた音響センサを用いて手／指動作を認識する研究も存在する．Han らは、多数の音響センサを備えたスマートウォッチ SoundCraft を開発した [HHN+17]．SoundCraft はスマートウォッチ周辺における音の発生位置と種類を識別することにより、音の発生を伴う手の動きの認識を行う．また、Amento らは、リストバンドに取り付けた音響センサを用いることにより、親指と他の指を叩き合わせたり、こすり合わせる動作を認識した [AHT02]．

また、本研究と同様に、列状に多数並べたセンサ（センサアレイ）を手首の周囲を囲むように取り付けることにて手／指動作を認識する研究も存在する．Zhang らは電気インピーダンス・トモグラフィを用いた Tomo というシステムを開発し、手の姿勢を認識した [ZH15]．Rekimoto らは、手首に巻き付けた静電容量式のセンサアレイを用いて、指の動作に伴う腱や筋肉の動きから生じる手首の形状変化を計測することにより、手の姿勢を認識した [Rek01]．また、Dementyev らは、手首に感圧（圧力）センサアレイを巻きつけることにより同様のことを行った [DP14]．同様に、Fukui ら [FWSS15] および Ortega-Avila ら [OARSM15] は、手首に赤外線近接センサアレイを巻き、センサと皮膚間の距離から手首の形状変化を計測することにより、手の姿勢を認識した．同様に Gong らは、手首に赤外線近接センサアレイを巻くことにより、手首の動きを認識した [GYI16]．McIntosh らも同様に、赤外線近接センサアレイを備えたブレスレット型デバイス SensIR を開発し、手の姿勢を認識した [MMF17]．ただし、彼らはセンサと皮膚間の距離を測るのではなく、赤外線が手首を透過することに着目し、手首に巻き付けられた 14 個の赤外線エミッタからの発光を、同じく手首に巻き付けられた 14 個の赤外線レシーバを用いて受信することにて手の姿勢を認識した．

### 2.1.3 手（指，甲）に取り付けたセンサを用いた認識手法

指に取り付けたカメラを用いて手／指動作を認識する研究も存在する．例えば，Chanらは手の姿勢を認識する指装着型デバイス CyclopsRing を開発した [CCH<sup>+</sup>15]．CyclopsRing は，視野角が 185 度の魚眼レンズカメラを備えており，指の間から見える手の状態をカメラに収めることにより手の姿勢を認識する．

指に取り付けた加速度センサを用いて手／指動作を認識する研究も存在する．Jingらは，人差し指装着型端末 Magic Ring を開発し，この端末から取得した指の加速度に基づいて指の動きを認識した [JCZ<sup>+</sup>13]．また，Perngらは，各指先および手の甲に加速度センサを備えたグローブを用いて手の動きを認識した [PFHP99]．さらに，塚田らは赤外線センサ，加速度センサ，曲げセンサおよびスイッチを持つ指装着型デバイス Ubi-Finger を開発した [塚田 02]．この Ubi-Finger は，ライトや TV などの機器を「指差す」動作および指曲げや手首の回転などの動きを認識する．

電界や磁界の変化を利用して指の動きを認識する研究も存在する．Wilhelmらは，電界を発生させる指装着型デバイスである eRing を開発し，指の動作に伴う電界の変化を検知することにより指動作の認識を行った [WKTA15]．また，Harrisonらは，指に装着した磁石および手首装着型端末の磁力センサを用いて指の動きを認識する Abracadabra という手法を示した [HH09]．

その他のセンサを用いた研究として，赤外線近接センサやひずみセンサを利用した研究も存在する．Ogataらは，4つの赤外線近接センサを備えた指輪型端末 iRing を開発し，指の曲げや回転から起こる皮膚への指輪の食い込みを測ることにより指の動きを認識した [OSOI12]．また，Linらはひずみセンサアレイから成るセンサデバイスである BackHand を開発した [LWH<sup>+</sup>15]．BackHand は手の甲に貼り付けられることにより，手の動きを認識する．

## 2.2 スマートウォッチの操作手法に関する研究

スマートウォッチの操作手法に関する研究も数多くなされている．これらの手法は「スマートウォッチの縁や周辺の皮膚への入力を用いた操作手法」，「スマートウォッチ周辺の空中における手の動作を用いた操作手法」，「スマートウォッチ装着側の手による動作を用いた操作手法」に分類される．

### 2.2.1 スマートウォッチの縁や周辺の皮膚への入力を用いた操作手法

スマートウォッチのタッチディスプレイの外に入力領域を拡張するために，スマートウォッチの縁に対する操作を認識する研究が存在する．Xiaoらはスマートウォッチの画面枠（腕時計における文字盤）にジョイスティックセンサを組み込むことにより，画面枠に対する押し込みや回転動作を認識し，これをスマートウォッチへの入力に利用した [XLH14]．Ogataらはスマートウォッチの裏面に赤外線近接センサを複数設置することにより，スマートウォッチの縁

に対する押し込みやスマートウォッチを回転させる動きを認識し、これをスマートウォッチへの入力に利用した [OI15].

また、スマートウォッチ周辺の皮膚を入力領域とすることによりスマートウォッチを操作することを試みた研究もある。Weigelらは、皮膚に装着可能なタッチセンサ iSkinを開発し、これをスマートウォッチへの入力に利用した [WLB<sup>+</sup>15]. Knibbeらは、スマートウォッチの側面に取り付けた赤外線センサを用いて手の甲への押し込みおよびつまみ動作を認識し、スマートウォッチへの入力に利用した [KMPB<sup>+</sup>14]. また、Laputらは、スマートウォッチに取り付けられる小型のレーザプロジェクタである Skin Buttonsを開発した [LXC<sup>+</sup>14]. Skin Buttonsは腕にアイコンを表示するためのプロジェクタであり、投影されたアイコンへの指の接触を赤外線近接センサを用いて検知することにより、スマートウォッチへの入力を可能にする。Zhangらは、高周波の交流信号を用いて皮膚への入力を認識する SkinTrackシステムを開発した [ZZLH16]. SkinTrackでは、操作側の指に取り付けた指輪型デバイスから発生される信号を、スマートウォッチに取り付けたセンサにて受信することにより、皮膚に対するタッチの座標を認識する。SkinTrackはこのタッチ座標を用いて、スマートウォッチへのタッチ入力やボタン入力を可能にする。また、Laputらの ViBandは、腕や手の甲など叩かれた位置を認識し、これをスマートウォッチへの入力に利用する [LXH16].

### 2.2.2 スマートウォッチ周辺の空中における手の動作を用いた操作手法

スマートウォッチのタッチディスプレイの外に入力領域を拡張するために、スマートウォッチ周辺の空中における手の動作を用いた操作手法に関する研究が存在する。Withanaらは、スマートウォッチなどのウェアラブルデバイスに赤外線近接センサを取り付けることにより、ウェアラブルデバイス周辺の空中における手の動作を認識する技術である zSense [WPSN15]を示した。zSenseはスマートウォッチ上の空中で動かされた指の動きや手の形を認識し、これをスマートウォッチ操作に利用する。Lienらの Soliは、スマートウォッチを装着していない側の手により行われたマイクロジェスチャ（親指と人差し指をこすり合わせるなど）を認識し、スライダ操作などを実現する [LGK<sup>+</sup>16]. また Gongらの Pyroも同様に、スマートウォッチを装着していない側の手によるマイクロジェスチャを認識し、これをスマートウォッチへの入力に利用する [GZZY17]. さらに Hanらの SoundCraftは、スマートウォッチ周辺において発生させた音の方向と種類を認識し、スマートウォッチへの入力に利用する [HHN<sup>+</sup>17].

### 2.2.3 スマートウォッチ装着側の手による動作を用いた操作手法

本研究と同様にスマートウォッチ装着側の手による動作を用いた操作手法（スマートウォッチの片手操作手法）に着目した研究は存在する。Huangらは、スマートウォッチを片手で操作する方法として、親指で他の指の腹を触れる操作手法 DigitSpaceを示した [HCY<sup>+</sup>16]. DigitSpaceは親指で他の指の腹をなぞったり、タッチすることによりスマートウォッチ上でのボタン操作やスライダ操作を実現する。同様に Loclairらは、ロッククライミング中において、片手でス

スマートウォッチを操作するための入力動作として、親指で他の指の腹をなぞったり、人差し指で親指の腹をなぞる操作を示した [HSP10]. Gong らは、手首の動きを用いてスマートウォッチを片手で操作する手法 WristWhirl を示した [GYI16]. WristWhirl は、ユーザが手首をジョイスティックのように使うことにより、スマートウォッチにおける2次元操作を実現する. Sun らは、スマートウォッチを傾けることにより片手操作を行う操作手法 Float を示した [SWY+17]. Float は、スマートウォッチを傾けることにより座標を決め、指を曲げる動作を用いて決定を行う. 福本らは手首装着型端末を装着した手の任意の指先で行った机や腕を指先で叩く動作(打指)を検出し、片手操作を実現する方法を示した [福本 99]. この研究では、打指のリズムパターンにより 10-30 種類のコマンド入力を行う. この打指による入力は、親指と他の指の指先同士を叩き合わせることも行えるため手首装着型端末において片手操作が実現される. Kerber らは Myo を用いることにより、スマートウォッチにおける片手操作及びアイズフリー入力が可能であるかを調査した [KLK15]. この研究では、Myo にて認識が可能な5つの動作(指先同士を叩き合わせる、手を開くなど)を用いてスマートウォッチを操作する方法を示した. Guo らは、スマートウォッチの片手操作を実現するために前腕の回転を用いた入力方法について調査を行った [GP16]. また、Laput らの ViBand は、指を鳴らす動作や指先をこすり合わせる動きを認識し、これを利用してスマートウォッチの片手操作を実現する [LXH16]. さらに Han らの SoundCraft は、指を鳴らす動作と手首を回す動作を組み合わせることにてスマートウォッチのスクロール操作を実現する [HHN+17].

## 2.3 本研究の位置付け

本研究では、2.1.2 節にて示した研究のように、手首に取り付けたセンサを用いて指動作を認識する. 特に本研究は Fukui ら [FWSS15] および Ortega-Avila ら [OARSM15] の研究を参考に、手首に装着した赤外線近接センサアレイを用いて認識を行う. ただし、これらの研究が手の姿勢認識を目的としているのに対し、本研究は独自のアルゴリズムを用いて親指の動きを認識する.

また、本研究は 2.2.3 節にて示した研究と同様に、スマートウォッチ装着側の手による動作を用いた操作を提案、実現する. 本研究は、その中でも特に指先の動きを操作に利用した先行研究 [HCY+16, HSP10, 福本 99, LXH16] と同様に、親指の動きを操作に利用することに着目する. ただし、Huang らの DigitSpace や Loclair の示した入力動作 [HCY+16, HSP10] は各指の腹を入力面とする操作を提案しているのに対し、本研究は人差し指の側面を入力面とする操作を実現する. また、福本ら [福本 99] は打指のリズムパターンによるスマートウォッチへのコマンド入力を実現するのに対し、本研究は親指動作を用いた連続的な入力を実現する. また、Laput らの ViBand [LXH16] は、指先をこすり合わせるなどの単一の動作を入力に用いるのに対し、本研究は推定した親指の移動量を入力に用いる.



## 第3章 スマートウォッチ操作に用いる入力動作

本研究では，スマートウォッチの片手操作を実現する方法として，マイクロジェスチャを用いた入力操作を提案および実装する．本章においては，スマートウォッチ操作に用いる入力動作を示し，その認識原理を述べる．

### 3.1 親指スライドジェスチャを用いた一次元入力

本研究は，スマートウォッチの操作方法として親指スライドジェスチャを用いる．ユーザはこの入力動作を用いることにより，スマートウォッチへの一次元的な入力が可能となる．例えば，図3.1のように人差し指上の親指の位置をスマートウォッチ画面におけるポインタ位置に当てはめることにより，ポインティング操作を行うことが可能である（絶対ポインティング）．ユーザはこの親指スライドジェスチャを用いたポインティング操作と他の入力動作による決定操作を組み合わせることにより，ユーザが片手にてスマートウォッチを操作し，画面をタッチすることなくターゲットを選択を行うことができる．また，図3.2のようなスクロール操作を行うこともできる．スクロール操作では，親指の絶対ポインティングにおける中心座標からの距離をスクロールの速度として利用する．なお，中心座標からの距離をポインタの速度として利用することにより，選択操作を行うことも可能である．また，図3.3のように，親指移動の速度を利用しスワイプ操作を行うことも可能である．ユーザは左右どちらかの方向に親指をすばやく動かすことにより，スワイプ操作を行う．

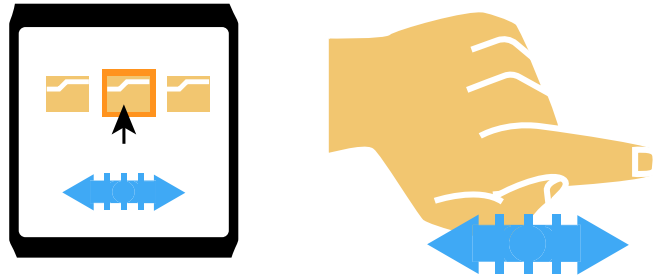


図 3.1: 親指スライドジェスチャを用いた絶対ポインティング. 人差し指上の親指の位置をスマートウォッチ画面におけるポインタ位置に当てはめることにより, ポインティング操作を行う.

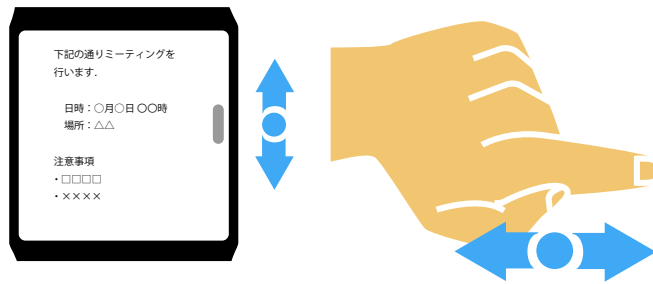


図 3.2: 親指スライドジェスチャを用いたスクロール操作. 親指の絶対ポインティングにおける中心座標からの距離をスクロールの速度として利用する.

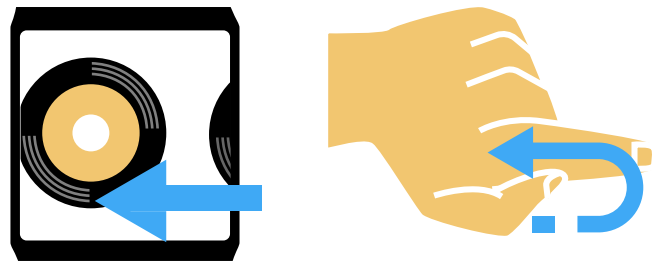


図 3.3: 親指スライドジェスチャを用いたスワイプ操作. 親指をすばやく動かすことによりスワイプを行う.

親指スライドジェスチャをスマートウォッチ操作に用いることには以下のような利点がある。

### タッチディスプレイを用いない

スマートウォッチにおけるタッチディスプレイを用いた入力操作には、画面が小さく操作しづらいという問題がある。一方、親指スライドジェスチャを用いた入力操作には、タッチディスプレイを用いないため、これらの問題が発生しない。

### 操作がスマートウォッチ装着側の手（片手）のみで完結する

親指スライドジェスチャは後述する決定操作の入力動作と組み合わせることにより、選択操作を可能にする。これにより、幅広いアプリケーションにおける片手操作が実現できる。

### 人差し指が支えとなる

親指スライドジェスチャは人差し指が支えとなるので、操作が安定するとともに触覚フィードバックが生まれる。

### 動作が小さい

親指スライドジェスチャは親指の動きのみで行えるため、手全体を用いる入力動作や腕を用いる入力動作と比べて動きが小さい。そのため、動きの大きい入力動作と比べて周りの目を気にしなくて良い。また、疲れにくい。

### 他の入力動作とも併用できる

親指スライドジェスチャは親指のみを用いる入力動作であるため、他の指を用いた入力動作と併用できる。例えば、親指スライド動作をポインティング操作として使い、他の指を握りこむ動作を決定操作として組み合わせることが可能である。これにより、片手のみにより選択操作が可能になる。

### かばんを持ちながらでも行える

親指スライドジェスチャは親指の動きのみを用いる入力動作であるため、手が閉じているか開いているかにかかわらず行うことが可能である。そのため、かばんを持った状態においてもスマートウォッチを操作することができる。

### 3.2 決定操作に用いる入力動作

本研究では，親指スライドジェスチャと併用して用いる決定操作の入力動作として「親指タップジェスチャ」，「握りジェスチャ」を用いる．親指タップジェスチャは図3.4のように，親指で人差し指の側面を叩く動作を指す．また，握りジェスチャは図3.5のように，親指および人差し指を除いた指を握りこむ動作を指す．これらの入力動作は，親指スライドジェスチャと同様に「タッチディスプレイを用いない」，「片手のみで完結する」，「動作が小さい」，「かばんを持ちながらでも行える」という特徴がある．



図 3.4: 親指タップジェスチャ.



図 3.5: 握りジェスチャ.

### 3.3 認識原理

スマートウォッチは常に手首に装着されているため、その見た目に関しても気を付けなければならない。そこで、本研究は上記入力動作を認識する方法として、ベルトにセンサアレイを組み込む。

先行研究である Fukui ら [FWSS15] および Ortega-Avila ら [OARSM15] は、手首に赤外線近接センサアレイを巻き、センサと皮膚間の距離から手首の形状変化を計測することにより、手の姿勢を認識した (図 3.6)。このような手首に巻きつけた赤外線近接センサアレイを用いる認識手法の利点は、センサがベルトの内側に収まる点である。これにより、スマートウォッチの見た目を損なうことなく認識が可能となる。

このような手法は、親指スライドジェスチャの認識においても有効であると考えられる。なぜなら、親指の動作に関連する筋肉が手首において皮膚近くに位置するためである。図 3.7 は親指の曲げ伸ばし動作に関連する筋肉 (長母指外転筋, 短母指伸筋) および腱を示している。親指スライドジェスチャは親指を曲げ伸ばす動作を含んでいるため、動作に応じて筋肉が伸縮し手首の形状に変化を与える。そのため、この親指スライドジェスチャに応じた手首の形状から親指の動きを推定できると考えられる。

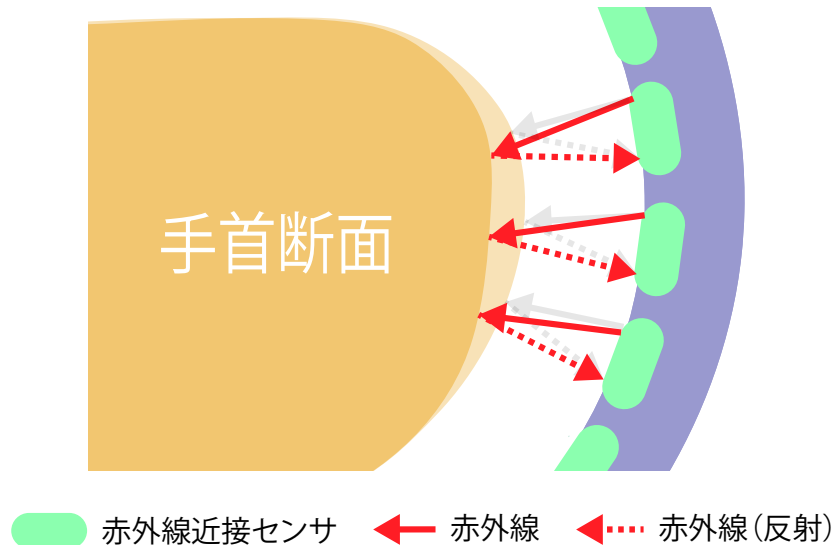


図 3.6: 赤外線近接センサアレイを用いた手首の形状変化計測。それぞれのセンサは、センサと皮膚間の距離を測定する。

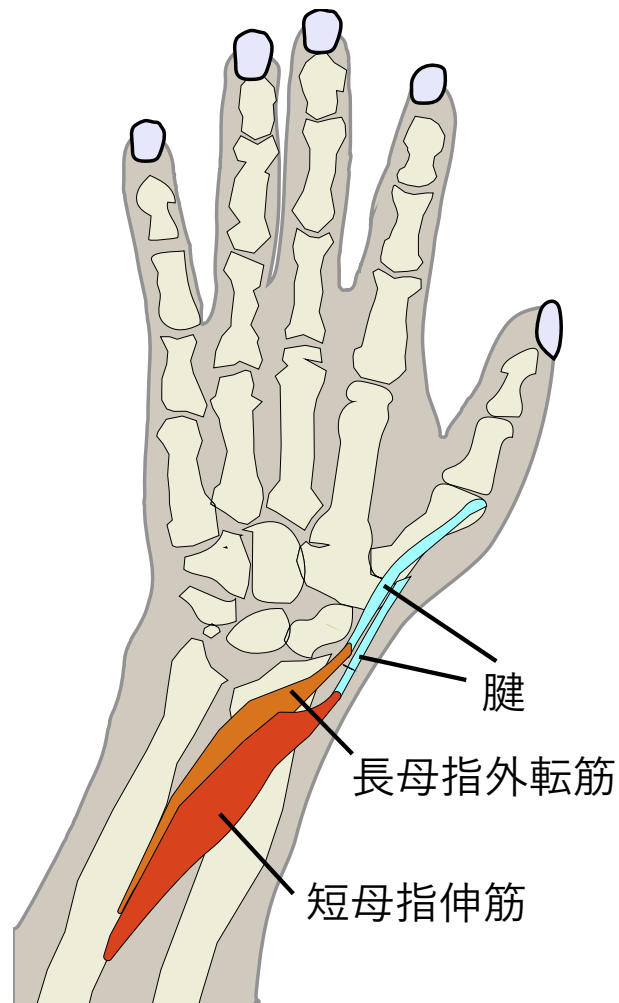


図 3.7: 親指動作に関連する筋肉および腱.

## 第4章 予備実験：親指スライドジェスチャに伴う手首形状変化の測定

親指スライドジェスチャの認識アルゴリズムを設計するにあたり、親指動作に伴う手首形状変化を観察するための予備実験を行った。本章においては、この実験の設計を述べ、その結果を示す。

### 4.1 実験協力者

本実験の実験協力者は、筑波大学に所属する大学生及び大学院生 10 名である。なお、このうち 1 名の実験協力者のデータに関しては実験終了後、実験プログラムの不備が発覚したため、解析には含めなかった。解析にデータを用いた 9 名（男性 9 名、年齢 18–23）のうち、右利きは 8 名、左利きは 1 名であった。

### 4.2 実験機器

実験には手首の形状変化を測定するためのセンサデバイスおよび、実験条件を統一するためのグローブを用いた。本節では、これらの詳細を示す。

#### 4.2.1 センサデバイス

親指スライドジェスチャに伴う手首の形状変化を調べるために、実験用ベルト型センサデバイスを作製した（図 4.1）。本デバイスは 16 個の赤外線近接センサ（GENIXTEK 社製，TPR-105F<sup>1</sup>）が取り付けられたベルト部に、マイクロコントローラ（SparkFun Electronics 社製，Arduino Pro Mini）および BLE モジュール（有限会社浅草ギ研製，BLESerial 2）が接続された構造となっている。本デバイスは、ベルト部に取り付けられた面ファスナを用いて手首に装着される。本デバイスには、手首表面とベルトに取り付けた赤外線近接センサの間に隙間が設けられており、センサがこの隙間の大きさを測ることにより手首の形状変化を取得する。この隙間は、高さ 3 mm のウレタンスポンジを赤外線近接センサを囲むように取り付けることにより設けられた。また、このスポンジは隙間を設けるためだけではなく、太陽光を遮るための役割も持つ。このスポンジは、壁となって太陽光を遮ることにより、太陽光に含まれる

---

<sup>1</sup>検知距離：1 mm–10 mm 程度

赤外線がセンサに影響することを防ぐ。なお、この枠の太さは 10mm であり、枠内の大きさが 140mm × 10mm となるように取り付けられている。

次に、デバイスの内部構造を示す (図 4.2)。本デバイスは、土台としてゴム布を用いており、このゴム布にフレキシブル基板を分割して取り付けることにて作製されている。このように基板を分割することにより基板間に隙間が生じ、デバイスは曲げられるようになっている。この分割された基盤にはそれぞれ、1つの赤外線近接センサと2つのチップ抵抗 (220kΩ, 330Ω) が取り付けられている。また、すべての赤外線近接センサはマルチプレクサ (Texas Instruments 社製, CD74HC4067) に接続されており、これを用いてそれぞれのセンサが取得したセンサ値をマイコンに送信する。

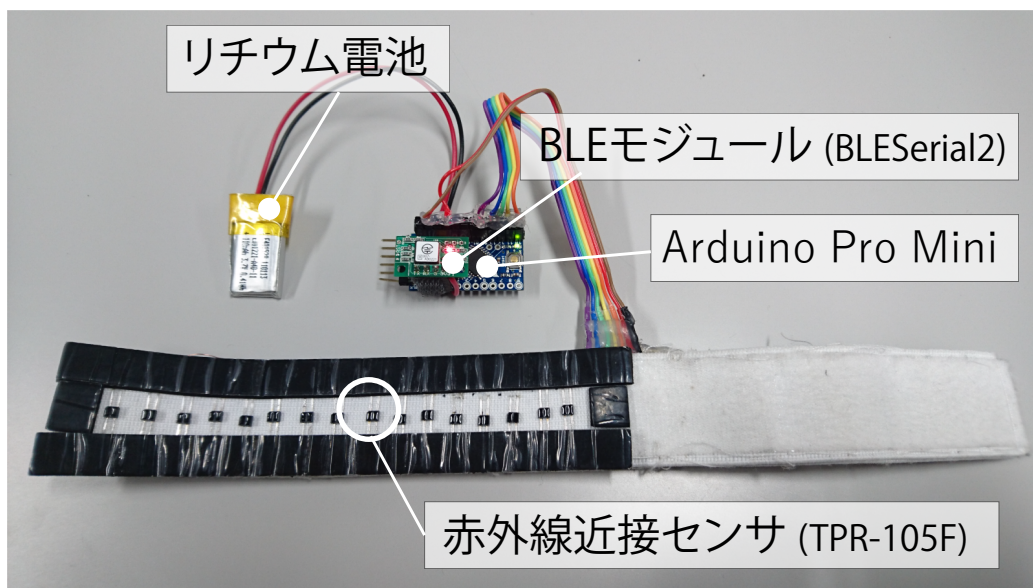


図 4.1: 実験に使用したベルト型センサデバイス。

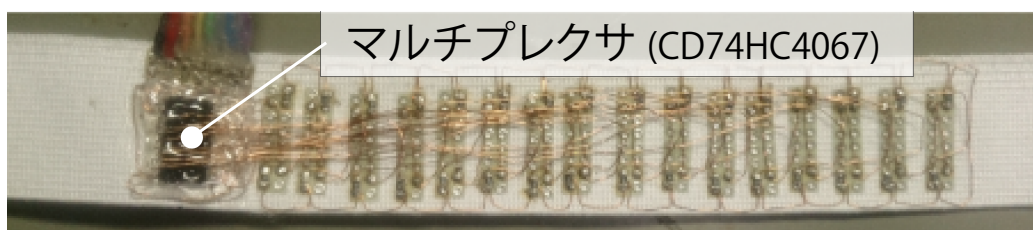


図 4.2: 実験に使用したベルト型センサデバイスの内部。



## 4.2.2 グローブ

実験協力者間において親指の移動量を一致させるため、図 4.3 に示すグローブ（ショーワグローブ株式会社製、ピッタリ背抜き S サイズ<sup>2</sup>）を使用した。このグローブは表面がポリウレタンにてコーティングされたナイロン製のものである。実験における目印のため、実験者はグローブの人差し指側面に 1 つの点（基準点）と親指側面に 5 mm 間隔に 6 つの点（親指の付け根側からスライド点 1-6）を描いた（図 4.3）。

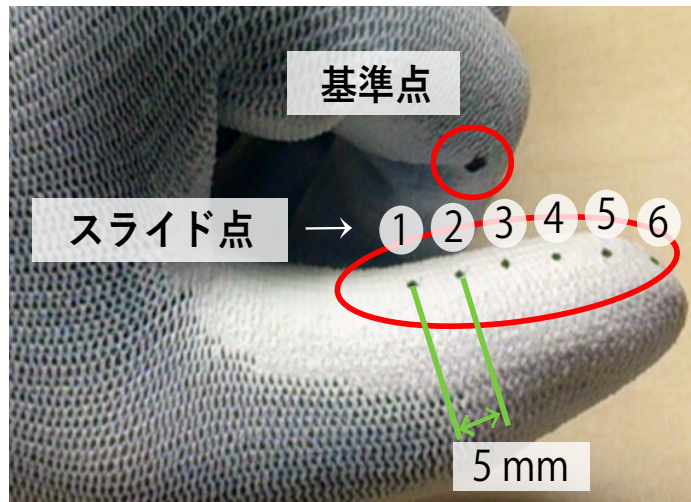


図 4.3: グローブに描かれた基準点とスライド点.

## 4.3 実験設計

### 4.3.1 準備

実験協力者には、グローブを装着した左手および腕を図 4.4 のように机の上に置いた状態にて実験を行ってもらった。またその際、手を軽く握るように指示した。これらは、腕の位置や手首の角度、他の指の曲げ状態が実験結果に影響しないようにするためである。ベルトと手の位置関係および赤外線近接センサ番号を図 4.5 に示す。

### 4.3.2 手順

実験手順を説明する。実験協力者はまず親指側面のスライド点 1 と人差し指側面の基準点を重ねた状態にて待機する。次に実験協力者は指示に従い、次のスライド点と基準点が重なるように親指をスライドさせる。これをスライド点 6 に到達するまで繰り返す。この過程を 1 試行とし、実験協力者はこれを 5 試行行った。

<sup>2</sup>JAN コード: 4901792026109



図 4.4: 実験環境.

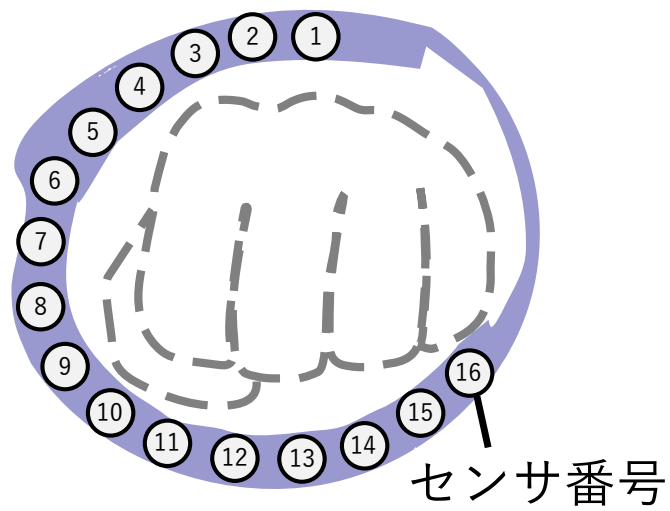


図 4.5: ベルト型センサデバイスと手の位置関係およびセンサ番号.

表 4.1: 実験協力者ごとの手首の周囲の長さ

	A	B	C	D	E	F	G	H	I
長さ (cm)	15.9	14.9	15.0	18.0	14.8	14.3	15.5	16.4	17.7

### 4.3.3 記録方法

センサ値の記録においては、各スライド点と人差し指側面の点が重なり、静止した状態のものを記録した。また、実験結果の解析には、5 試行分の平均を取った値を用いた。さらに、実験終了後にセンサデバイスを装着した箇所における手首の周囲の長さを測定した。表 4.1 に、各実験協力者の手首の周囲の長さを示す。

## 4.4 結果と考察

図 4.6 に各センサにおけるセンサ値の変化量を示す。この図におけるセンサ値の変化量とは、スライド点 6 におけるセンサ値とスライド点 1 におけるセンサ値の差分を表している。すなわち、変化量が大きいセンサがある位置ほど手首の形状が大きく変化することを示している。また、図 4.7 に、親指の移動量とセンサ値の変化量の相関係数を示す。この相関係数は 1.00 に近いほど親指の移動に伴って線形にセンサ値が上昇していることを示している。反対に、-1.00 に近いほど線形にセンサ値が下降していることを示している。

### 4.4.1 手首の形状変化

図 4.6 および図 4.7 から、いずれの実験協力者においても、センサ値が大きく変化するセンサ位置と相関係数が高い位置が一致していることが確認される。このことから、親指スライドジェスチャは手首形状に影響を与え、かつその形状変化は親指の移動量に応じて線形に変化することがわかる。したがって、これら相関係数の高い位置のセンサ値を用いることにより、センサ値の大小から親指の移動量推定が可能であることが伺える。なお、手首の周囲の長さや皮膚の膨らむ位置や膨らみの大きさとの関係は特に見られなかった。

### 4.4.2 親指移動量推定の精度検証

実験にて得られたセンサ値をもとに、センサ値の大小から親指の移動量推定が可能であるかを検証した。図 4.8 に親指移動量の実測値と推定値の誤差距離の平均を示す。図 4.8 の各折れ線グラフは推定に使用したセンサの相関係数の高さごとの誤差距離を表している。この推定に使用するセンサは、設定した閾値以上の相関係数であるかを判断基準とした。また、本検証においては、スライド点  $i$  における親指の推定移動量  $T_i$  を下記の式にて計算した。

$$T_i = \frac{S_i - S_1}{S_6 - S_1} \times 25 \text{ mm}$$



図 4.6: 親指スライドジェスチャ時における各センサ値の変化量.

センサ番号

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
A	0.94	0.98	0.98	0.98	0.98	0.98	0.23	0.89	0.90	-0.28	-0.43	0.83	0.93	0.93	0.93	0.93
B	-0.99	-0.96	-0.31	0.88	0.93	0.63	0.97	0.94	0.41	-0.70	-0.99	-0.97	-0.90	-0.88	0.67	0.64
C	0.47	0.19	0.09	0.92	0.63	0.17	0.92	0.87	0.88	0.93	-0.42	0.58	0.28	0.38	0.05	0.11
D	0.36	0.38	0.69	0.97	0.53	-0.37	-0.89	-0.94	-0.99	0.28	0.23	0.33	0.39	0.53	0.34	0.34
E	-0.90	0.60	0.87	0.97	0.97	0.99	-0.89	0.80	0.79	0.32	0.90	0.91	0.88	0.82	0.84	0.88
F	-0.77	-0.81	-0.75	-0.28	-0.36	-0.92	1.00	0.79	0.83	0.87	0.29	0.81	0.92	0.97	0.87	0.98
G	-0.90	0.04	0.92	0.97	0.86	0.79	0.73	0.81	0.66	0.67	-0.95	0.22	0.02	0.26	0.42	-0.80
H	-0.83	-0.87	-0.82	-0.73	-0.74	-0.83	0.86	0.86	0.93	-0.87	0.77	0.28	0.28	0.68	0.48	0.68
I	0.68	0.98	0.98	0.97	0.99	0.92	-0.99	-0.97	-0.32	0.87	0.80	0.81	0.95	0.84	0.89	0.89

被験者

図 4.7: 親指の移動量とセンサ値の変化量の相関係数.

この式における  $S_i$  は、スライド点  $i$  における相関係数の絶対値が閾値  $n$  以上のセンサの値の合計を表している。なお、相関係数の絶対値を用いる理由は負の相関が大きいセンサも推定に含めるためである。

図 4.8 より、相関係数の高いセンサのみを用いるほど、小さな誤差で親指の移動量が推定可能であることが確認された。ただし、閾値を高くしすぎるとすべてのセンサの相関係数が閾値を下回り、推定が行えなくなる可能性がある。これは、実験協力者によって相関係数の高さに差があるためである。例えば、実験協力者 A においては相関係数の絶対値における最大値が 0.98 であるのに対し、実験協力者 H においては 0.93 である。このことより、特定の閾値を用いて使用するセンサを選定する場合においては、人によっては推定が行えない可能性が示唆される。

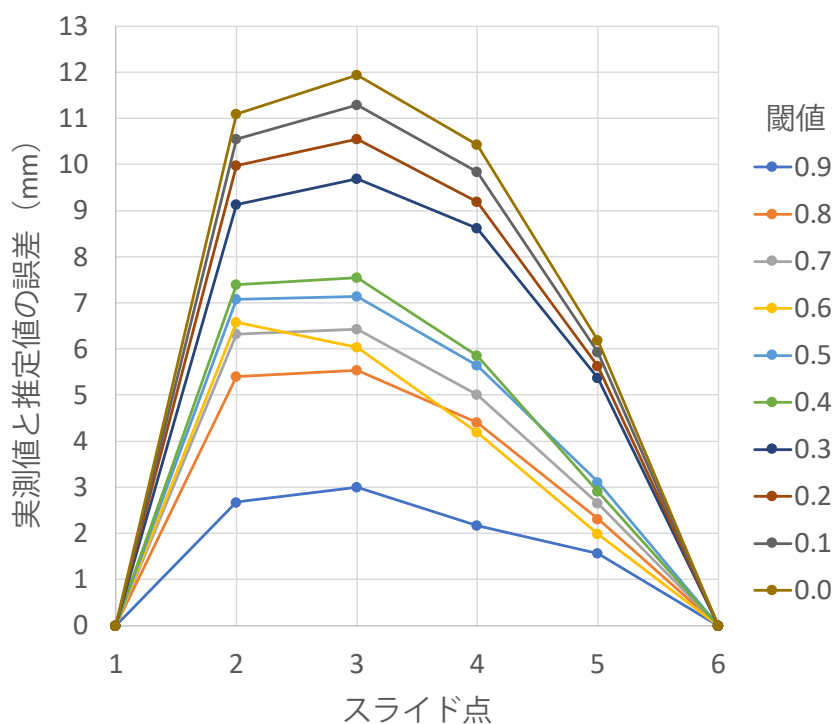


図 4.8: 親指移動実測値と推定値の誤差距離の平均。

## 第5章 親指動作認識システム

予備実験における考察を踏まえ、親指動作を認識するシステムを実装した。本章においてはこのシステムにおけるソフトウェア側の処理を述べるとともに、新たに製作したハードウェアについて述べる。

### 5.1 認識ソフトウェア

ソフトウェアはセンサデバイスから得たセンサ値を用いて、親指動作を認識する。この親指動作認識のための処理は「キャリブレーション」、「データ加工」、「推定」の3つの段階から構成される。

#### 5.1.1 キャリブレーション

親指動作の認識にあたり、システムは事前に親指動作に伴う各センサ値の変化波形（サンプル波形）を得る必要がある。そのため、システムはキャリブレーションとして、ユーザに一度親指で人差し指の側面上をなぞらせ、センサ値データを収集する。このキャリブレーションは親指動作認識の開始前に行われる。キャリブレーションにおいて、データ収集の開始と終了のタイミングはユーザの任意のタイミングで行われる。システムは収集したデータから各センサごとに16点のサンプル点から構成されるサンプル波形  $W_i$  を生成する（図5.1）。 $W_i$  において  $i$  はセンサ番号を表す。また、サンプル波形における各サンプル点は  $W_{i,t}$  と表す。 $W_{i,t}$  において  $i$  はセンサ番号、 $t$  はサンプル点の番号を表している。

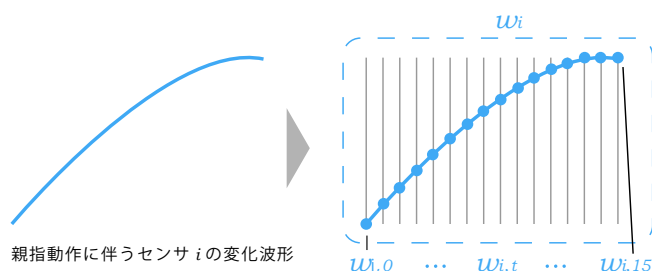


図 5.1: 収集したデータから各センサごとに16点のサンプル点から構成されるサンプル波形  $W_i$  を生成。

### 5.1.2 データ加工

システムは、サンプル波形と推定用のセンサ値を比較することにて親指の移動量を推定する。しかし、センサ値の変化は腕の角度にも影響をうけるため、同じ指の移動量においても、腕の角度に応じてセンサ値は大きく変化する [FWSS15]。そのため、センサ値の生値 ( $R_i$ ) がサンプル波形の最大値  $Max(W_i)$  よりも大きくなる場合や、最小値  $Min(W_i)$  よりも小さくなる場合がある。そこで、本システムはオフセット値 ( $offset_i$ ) を用いることにより、センサ値がサンプル波形の帯域内に収まるようにセンサ値を調整する (図 5.2)。すなわち、調整された推定用センサ値  $C_i$  は次の式で表される。

$$C_i = R_i + offset_i.$$

さらに、オフセット値は次式にて随時更新される。

$$offset_{i,f} = \begin{cases} Max(W_i) - R_i & (R_i + offset_{i,f-1} > Max(W_i)) \\ offset_{i,f-1} & (Min(W_i) \leq R_i + offset_{i,f-1} \leq Max(W_i)) \\ Min(W_i) - R_i & (R_i + offset_{i,f-1} < Min(W_i)), \end{cases}$$

この式において、 $f$  はフレーム番号を表す (すなわち、 $f - 1$  は直前のフレームを表す)。

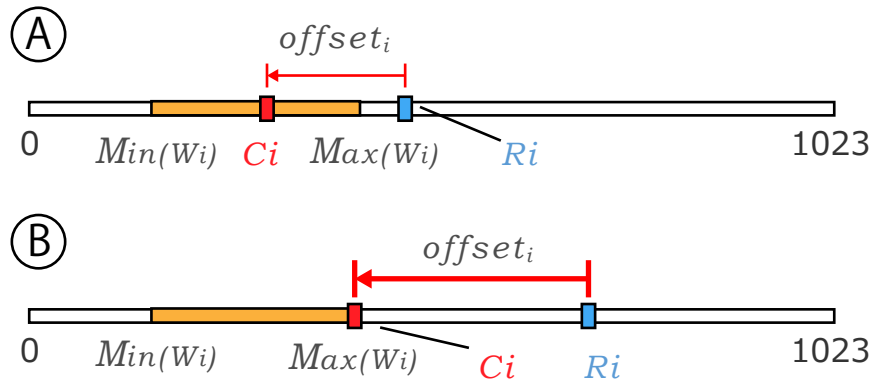


図 5.2: キャリブレーションにおけるセンサ値変化区間の移動. A) オフセット値 ( $offset_i$ ) を用いて、センサ値の生値 ( $R_i$ ) がサンプル波形の帯域内に収まるようにセンサ値を調整する. B) 調整後の推定用センサ値  $C_i$  がサンプル波形の帯域外に出る場合オフセット値 ( $offset_i$ ) を更新する。



### 5.1.3 推定

第4章の実験における考察を踏まえ、親指の移動量を推定するための2つのアルゴリズムを設計した。1つ目のアルゴリズムは親指動作に伴うセンサ値変化の線形性に着目した「線形選択アルゴリズム」である。また、2つ目のアルゴリズムはセンサ値の変化波形に着目した「位置重み付けアルゴリズム」である。

#### 線形選択アルゴリズム

第4章の実験より、センサデバイスにおける特定のセンサの値が、親指動作に伴い線形に変化することが確認された。線形選択アルゴリズムでは、この線形性を利用し、センサ値の大小から親指の移動量を推定する(図5.3)。図5.3における、横軸は親指の移動量、縦軸はセンサ値を表している。図5.3のように、親指の移動に伴うセンサ値の変化が直線であると、センサ値と親指の移動量の対応は一對一となる。線形選択アルゴリズムでは、この関係を用いることにて親指の移動量を推定する。

線形選択アルゴリズムにおける、親指移動量推定の手順を述べる。システムはまず親指の移動量と各センサのモデル波形の値との相関係数を算出する。次に、システムは相関係数が0.90より大きいセンサを選択する。推定では、これら選択されたセンサに対して  $C_i - \text{Min}(W_i)$  の合計値と  $\text{Max}(W_i) - \text{Min}(W_i)$  の合計値を比較する。具体的には、親指の移動量  $T$  は次式に示す式にて計算される ( $T$  は0-1の値にて表される)。

$$T = \frac{\sum_{i=\text{selected\_sensors}} C_i - \text{Min}(W_i)}{\sum_{i=\text{selected\_sensors}} \text{Max}(W_i) - \text{Min}(W_i)}$$

#### 位置重み付けアルゴリズム

第4章の実験より、センサ値の大小から親指の移動量を推定する方法(線形選択アルゴリズム)は、人によって推定が行えない可能性があることが示唆された。そのため、線形に動かないセンサも推定に用いる位置重み付けアルゴリズムを設計した。このアルゴリズムでは、図5.3のような親指動作に伴うセンサ値の変化波形が線形にならない場合での利用を想定している。このアルゴリズムにおいては、それぞれのセンサ単一では一意に親指の移動量が定まらずとも、それぞれのセンサが尤もらしい親指の移動量を求めそれを組み合わせることにより、一意に親指の移動量を推定する。

位置重み付けアルゴリズムにおける、親指移動量推定の手順を述べる。まず初めに、システムはサンプル波形  $W_i$  と推定用センサ値  $C_i$  の交点を見つけ、そのサンプル点に対して重みをつける(図5.4A)。なお、交点は  $W_{i,t} \leq C_i < W_{i,t+1}$  もしくは  $W_{i,t} \geq C_i > W_{i,t+1}$  となった際の、 $C_i$  に近い値を持つサンプル点 ( $W_{i,t}$  もしくは  $W_{i,t+1}$ ) とする。もし、複数の交点が見つかった場合には、重み点数をその交点の数で割った値を重みとしてつける(図5.4B)。さらに、このアルゴリズムは交点が重みの頂点となるように両隣のサンプル点に対しても順々

に重みをつけていく (図 5.4B)。各センサのそれぞれのサンプル点に関して重みが計算されると、最も重みの総計が大きいサンプル点が親指の推定移動量として決定される (図 5.4C)。

具体的には、それぞれのサンプル点の重み  $w_t$  ( $t = 0-15$ ) は式 5.1 および式 5.2 にて計算される。まず、交点座標に対しては以下の式を用いて重みを加算していく。

$$(if W_{i,t} = intersections_i) w_t = w_t + \frac{weight}{Num(intersections_i)}. \quad (5.1)$$

ここで、 $intersections_i$  はそれぞれのセンサにおける交点を表す。また、 $Num(intersections_i)$  はそれぞれのセンサにおける交点数を表す。なお、現在の実装における  $weight$  は 8 である。次に、それぞれの交点の両隣のサンプル点について重みを加算していく。

$$(if \frac{weight}{Num(intersections_i)} - k > 0) w_{t \pm k} = w_{t \pm k} + \frac{weight}{Num(intersections_i)} - k. \quad (5.2)$$

この式において  $k$  は交点から見ていく隣のサンプル点であるかを表している。最後に、システムは  $w_t$  および 1 つ前のフレームの親指移動量を用いることで、推定の解像度を上げる。

$$T_f = \frac{Index(Max(w_t))}{Num(sampling\_point)} \times 0.1 + T_{f-1} \times 0.9.$$

ここで  $Index(Max(w_t))$  は最大の重みを持つサンプル点のインデックスを表し、 $Num(sampling\_point)$  はサンプル点の数を表す (すなわち、16)。

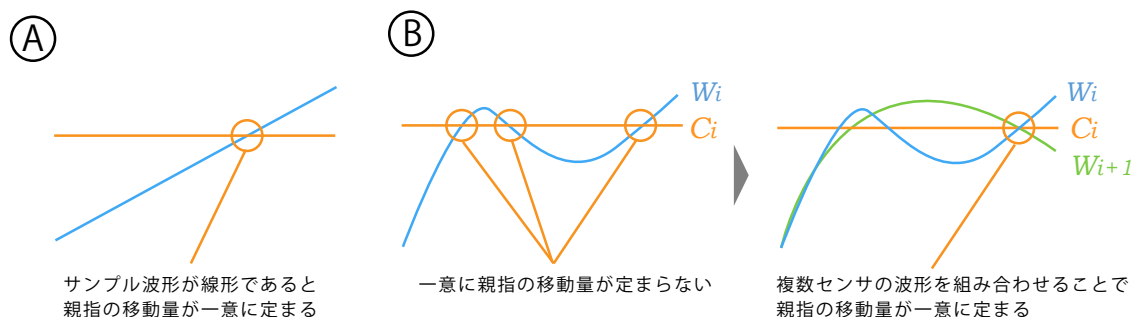


図 5.3: 各アルゴリズムにおける親指移動量の認識原理。グラフの横軸は親指の移動量、縦軸はセンサ値を表している。A) 線形選択アルゴリズム。B) 位置重み付けアルゴリズム

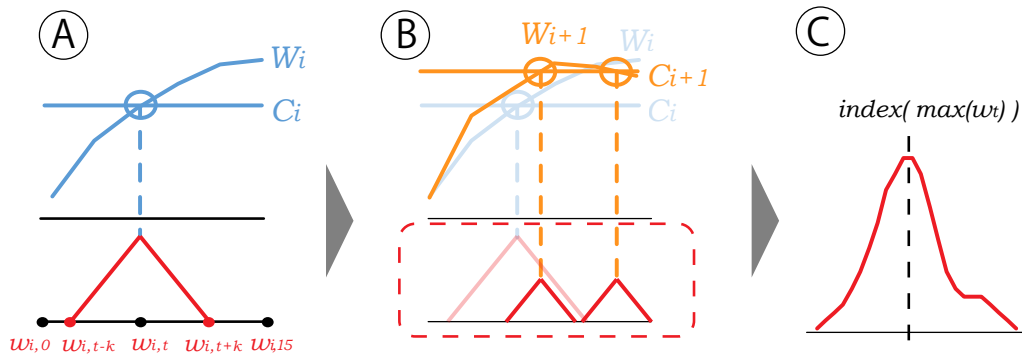


図 5.4: 位置重み付けアルゴリズム. A) あるセンサに関してサンプル波形  $W_i$  と推定用センサ値  $C_i$  の交点を見つけ, そのサンプル点に対して重みをつける. この時, 交点における重みが山の頂点となるようにまわりのサンプル点についても重みをつける. B) あるセンサに関して交点が複数見つかった場合には, 交点の数だけ重みの山を小さくし, A と同様のことを行う. C) すべてのセンサに関する重みを合算し, 重みが最大となるサンプル点を親指の推定位置とする.

## 5.2 認識ハードウェア

4.2 節にて使用したセンサデバイスを改良し, 新たなセンサデバイスを作製した (図 5.5). このデバイスは柔軟性のある土台上に構築され, マグネットバックル<sup>1</sup>タイプのスマートウォッチベルトと一体化するように設計されている. 土台はフレキシブルフィラメント (PVA Filament 1.75 mm natural, Shenzhen Esun Industrial) を用いて 3D プリンタから出力され, その厚さは 1 mm である. 実験用のセンサデバイスと同様に, 土台上には 16 個の赤外線近接センサとマルチプレクサが組み込まれている (図 5.6). また, 赤外線近接センサから得たセンサ値を送信するため, BLE (nRF51822 Rev.3 SoC, Nordic) が組み込まれた mbed (mbed TY51822r3, Switch Science) も組み込まれている. なお, 赤外線近接センサアレイを囲む枠は高さ 2 mm である (図 5.7).

<sup>1</sup>磁石式のベルトの留め金のこと. 穴に留め金を通すのではなく, 磁石にてベルトを固定する.



図 5.5: スマートウォッチベルト一体型センサデバイス.

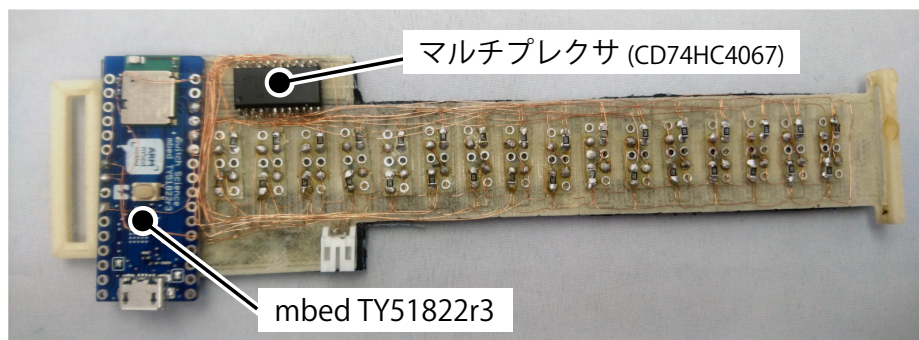


図 5.6: スマートウォッチベルト一体型センサデバイスの内部.

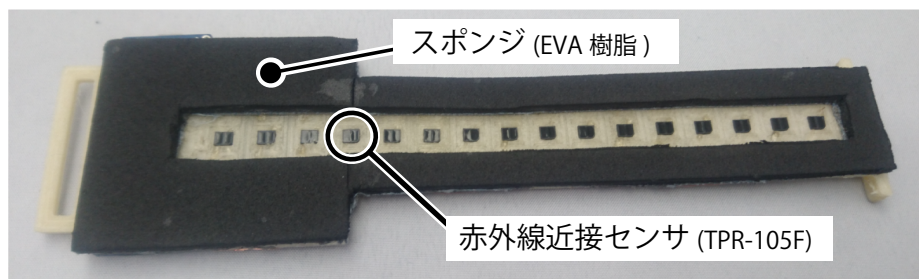


図 5.7: スマートウォッチベルト一体型センサデバイスのセンサアレイ面.

## 第6章 評価実験

実装した親指動作認識システムおよびこれを用いたスマートウォッチ操作に関する評価実験を行った。本章においては、この実験の設計を述べ、その結果および考察を示す。

### 6.1 実験協力者

実験には12人の参加者（P1–P12）を雇用して、評価実験を行った。実験協力者は19歳から24歳までの11人の男性と1人の女性である。なお、参加者は全員右利きであった。

### 6.2 実験機器

実験においては、5.2節にて示したセンサデバイスおよびスマートウォッチ（SONY SmartWatch 3 SWR50; Android Wear 1.5.0, 1.6 インチスクリーン, 解像度 320 px × 320 px), スマートフォン（SONY Xperia Z5 E6653; Android 6.0.1）を使用した。

### 6.3 実験設計

#### 6.3.1 準備

本実験は決定動作を含めたスマートウォッチ操作に関して評価を行う。そのため、簡易的な決定動作認識システムを実装し、これを実験に用いた。本システムは、スマートウォッチに組み込まれた加速度センサを用いることにて「親指タップジェスチャ」および「握りジェスチャ」を認識する。これら決定動作時に引き起こされる加速度の変化は親指スライドジェスチャ時の加速度変化とは異なる（図6.2）。そこで、親指スライドジェスチャに伴う加速度変化と、決定動作に伴う加速度変化を閾値で区別することにより決定動作を認識する。実験においては、実験協力者が最も決定動作が行いやすくなるように、実験者が手動にて閾値を調整した。

実験時の姿勢を図6.1Aに示す。本実験は、実験協力者が椅子に深く腰掛けた状態にて行われた。また、実験協力者の左手首にはセンサデバイスが取り付けられたスマートウォッチが装着された。この際、手は人差し指を伸ばした状態にてその他の指を軽く握りこむよう指示した。また、実験中は腕を空中に留めておくよう指示した。

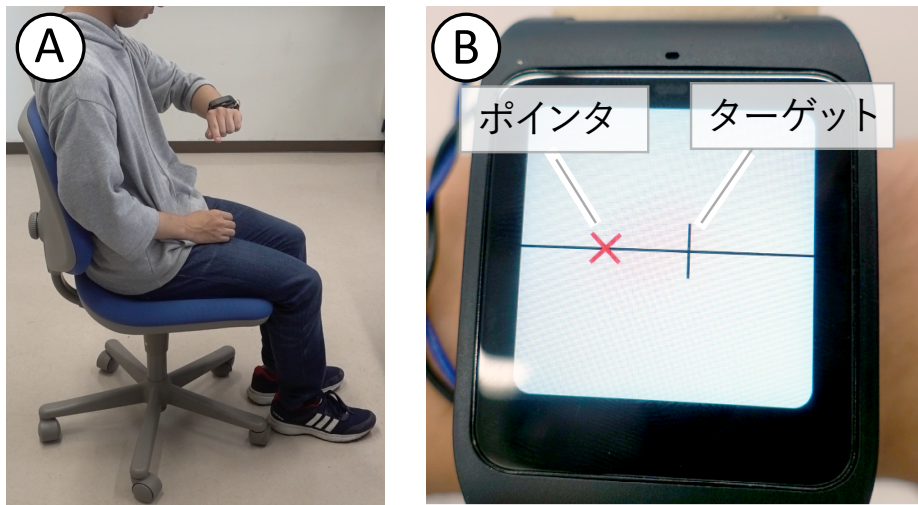


図 6.1: 実験環境：A) 実験時の姿勢，B) スマートウォッチに表示される画面．

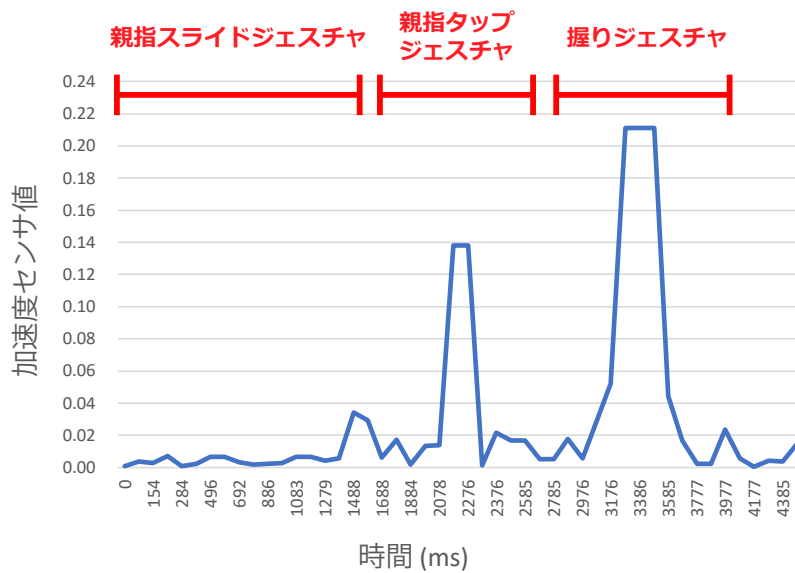


図 6.2: 決定動作時の加速度変化と親指スライドジェスチャ時の加速度変化の違い．

### 6.3.2 手順

実験手順を説明する。まず、実験協力者はセンサデバイスを装着した左手及び腕を図 6.1A のように胸の前に構えた姿勢をとる。実験が開始されると、スマートウォッチ画面上に図 6.1B のようにターゲットおよびポインタが表示される。実験協力者はスマートウォッチの画面に表示されたターゲットにポインタを移動させ、決定動作にてターゲットを選択する。この時、実験協力者はポインタの中心とターゲットの中心ができるだけ重なるように指示された。システムが決定動作を認識すると、1 秒間の待機時間の後、ターゲットはランダムな位置に移動する (0–320 px)。本実験においては、1 回のターゲット選択を 1 試行とし、1 セッションあたり 10 試行のターゲット選択を行ってもらった。また、本実験は各親指動作認識アルゴリズムおよび各決定動作を評価するために、4 つの条件 (2 アルゴリズム × 2 決定動作) にて行われた。実験協力者は各条件ごとに 1 回のトレーニングセッションと 3 回のメインセッション (合計 160 試行) を行った。なお、各実験条件は実験協力者ごとにランダムな順番にて行われた。実験終了後、実験協力者はアンケートに回答するように指示された。

### 6.3.3 記録方法

実験においては、各時間ごとに推定された親指位置および加速度値を記録した。また各試行ごとにターゲットの位置および決定動作が認識された時間を記録した。なお、親指位置の推定および記録はすべてスマートフォン上で行われた。また、加速度値においてもスマートウォッチからスマートフォンに送られ、スマートフォン上で決定動作の認識および記録を行った。

## 6.4 実験結果

実験から得られたデータを解析し、各条件におけるターゲット選択精度および各アルゴリズムにおけるポインタ移動の成功率を求めた。本節においてはこれら結果を示すとともに、本システムを用いたスマートウォッチ操作に関するアンケートの結果を示す。

### 6.4.1 各条件におけるターゲット選択精度

表 6.1 はターゲット位置と選択位置のずれ (誤差距離) の平均を示している。この表において、LT, LC, WT, WC はそれぞれの以下に示す実験条件を表している。

**LT 条件** 線形選択アルゴリズム – 親指タップジェスチャ。

**LC 条件** 線形選択アルゴリズム – 握りジェスチャ

**WT 条件** 位置重み付けアルゴリズム – 親指タップジェスチャ

**LC 条件** 位置重み付けアルゴリズム – 握りジェスチャ

表 6.1 より、実験協力者ごとに精度の良い条件と悪い条件が異なっていることが確認された。LT 条件が最も良い成績であったのが 3 人、LC 条件が 3 人、WT 条件が 2 人、WC 条件が 4 人であり、それぞれの誤差距離の平均は 60.4 px, 72.0 px, 63.3 px, 57.2 px となる。これら結果より、やや WC 条件が良い結果とも言えそうだが、条件間で大きな違いは見られない。

表 6.1: 各実験条件におけるターゲット位置と選択位置のずれ（誤差距離）の平均（かっこの中は標準偏差を表す）。

	<i>LT</i>	<i>LC</i>	<i>WT</i>	<i>WC</i>
P1	46.3 (51.7) px	59.6 (49.4) px	40.0 (30.4) px	32.9 (29.1) px
P2	27.4 (23.2) px	67.6 (45.3) px	54.1 (40.7) px	31.3 (37.0) px
P3	40.3 (43.5) px	76.7 (68.0) px	78.7 (59.5) px	58.4 (48.4) px
P4	58.9 (49.3) px	77.6 (74.4) px	62.3 (35.3) px	47.1 (34.2) px
P5	42.6 (45.4) px	36.5 (30.5) px	56.4 (54.2) px	53.7 (59.9) px
P6	53.3 (41.5) px	68.0 (54.8) px	87.1 (66.4) px	32.6 (30.9) px
P7	70.4 (52.7) px	32.9 (40.5) px	61.3 (51.5) px	45.7 (42.4) px
P8	57.0 (66.7) px	98.4 (68.5) px	58.8 (58.8) px	62.8 (50.6) px
P9	67.3 (66.1) px	112.8 (73.7) px	89.6 (71.5) px	62.2 (39.9) px
P10	128.6 (120.3) px	82.0 (71.0) px	35.9 (39.6) px	75.6 (50.3) px
P11	46.5 (33.1) px	36.4 (30.6) px	65.8 (51.0) px	106.9 (60.9) px
P12	85.5 (74.4) px	115.5 (90.0) px	69.0 (56.5) px	77.7 (61.2) px

解析により、決定動作がポインタ操作に影響を与えていることが確認された。図 6.3 は、LC 条件のある試行におけるターゲット位置と選択位置の誤差の絶対値（オレンジ色の折れ線グラフ）および加速度センサ値（青色の棒グラフ）を示している。図 6.3 に示す試行においては、ポインタがターゲット上に重なってから、決定動作が認識されるまでに 182.5 px のずれが発生していた。このようなポインタのずれは他の実験協力者および他の実験条件においても発生していた。さらに、実験協力者の中には決定動作時のポインタのずれを考慮して、決定動作検出後にターゲットまでの距離が最小になるように調整する実験協力者もみられた。そこで、決定動作がどの程度ポインタ操作に影響を与えているかを調べるため解析を行った。この解析では、各条件において決定動作が認識されたタイミングの前後 500 ms 間のデータの最小誤差距離と実際に決定動作が認識されたタイミングでの誤差距離を比較した。この結果、各条件における決定動作によるずれの平均はそれぞれ、LT 条件が 61.5 px (SD=58.7) , LC 条件が 80.0 px (SD=71.5) , WT 条件が 69.9 px (SD=55.0) , WC 条件が 64.5 px (SD=59.4) であった。

図 6.4 に決定動作によるポインタのずれを考慮した際の、実験協力者ごとの各アルゴリズムの精度を示す。図 6.4 の各バーの高さは、決定動作認識時の前後 500 ms の間のデータにおけ



る最小誤差距離の平均を表している。この図より、参加者間で有効なアルゴリズムが異なっていることがわかる。なお、各参加者においてより良い精度のアルゴリズムを採用する場合、誤差距離の平均は 15.4 px となった。

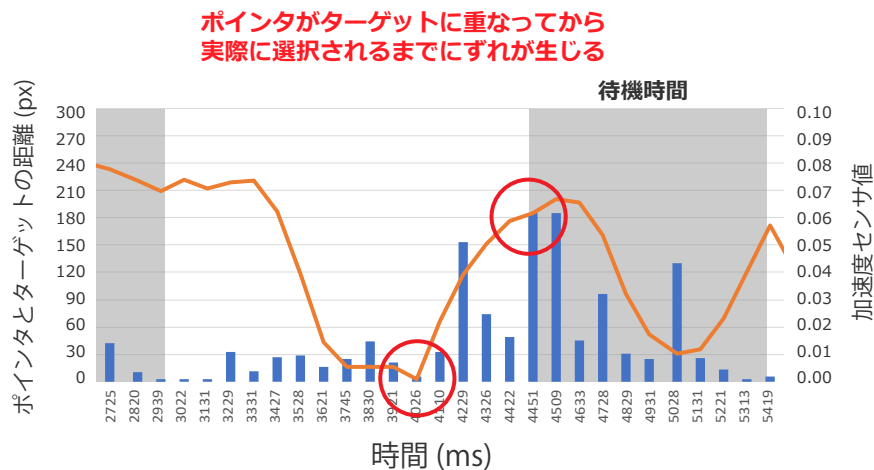


図 6.3: 決定動作によるポインタのずれ

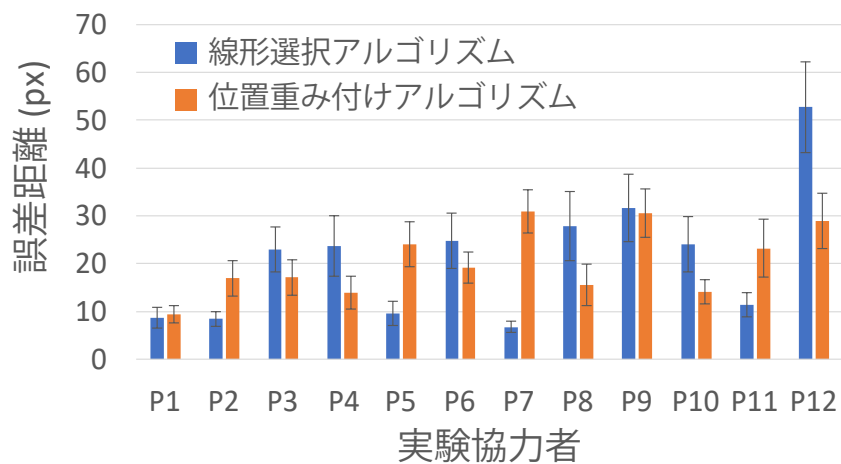


図 6.4: 決定動作によるずれを考慮した際の各アルゴリズムにおけるターゲット選択精度（エラーバーは標準誤差を表す）。

## 6.4.2 各アルゴリズムにおけるポインタ移動の成功率

より純粋な親指動作認識アルゴリズムの精度を評価するため、1 試行内に任意の位置に親指が移動できていたかを解析した。この解析においては、画面を 8 つの領域に分割し、試行開始時の領域からターゲットが表示された領域にポインタを移動できたかの成功率を求める。例えば、ポインタが 30 px の位置にあり、ターゲットが 130 px の位置に現れた場合、ポインタが領域 1 から領域 4 に移動できた場合を成功とする。図 6.5 に各アルゴリズムにおけるこのポインタ移動の成功率を示す。また、図 6.6 に、各実験協力者におけるより良い平均精度のアルゴリズムを採用した場合の成功率を示す。なお、位置重み付けアルゴリズムにおいては領域 8 から領域 3 へ移動させるような試行が発生しなかったため、表の 8 行 3 列目には「nun」と表示している。図 6.5 より、ポインタ移動の成功率はスマートウォッチ画面の端に向かうほど低くなっていることがわかる。なお、線形選択選択と位置重み付けアルゴリズムの平均精度はそれぞれ 83.2 % (SD=15.8), 81.9 % (SD=17.0) であった。また、各実験協力者のより良い平均精度のアルゴリズムを採用した場合の平均精度は 87.0 % (SD=11.4) に改善された。

## 6.4.3 アンケート

実験終了後に、実験協力者に回答してもらったアンケートに関しての内容およびその結果を示す。

### 決定動作について

親指タップジェスチャと握りジェスチャのどちらの決定動作が使いやすかったかを尋ねたところ、7 人の実験協力者が握りジェスチャのほうが使いやすかったと答えた。このうち、握りジェスチャのほうが使いやすかった理由として、反応の良さを理由にした実験協力者は 5 人であり、ずれの少なさを理由にした実験協力者は 2 人であった。一方、親指タップジェスチャのほうが使いやすいと答えた実験協力者は 4 人であり、そのうちの 2 人がずれの少なさを理由にした。また、反応の良さを理由にした実験協力者は 0 人であり、他 2 人の理由としては「握りは疲れた」、「人差し指を伸ばした場合親指タップジェスチャのほうがしやすかった」というものだった。なお、1 人の実験協力者はアルゴリズム条件ごとに使いやすい決定動作が異なると答えた。

### 親指動作認識アルゴリズムについて

線形選択アルゴリズムと位置重み付けアルゴリズムのどちらが使いやすかったかを尋ねたところ、7 人の実験協力者が線形選択アルゴリズムのほうが使いやすかったと答えた。一方、位置重み付けアルゴリズムのほうが使いやすいと答えた実験協力者は 5 人であった。すべての実験協力者について、それぞれのアルゴリズムのほうが使いやすいとした理由は認識精度

### 線形選択アルゴリズム

開始領域	1	100.0	100.0	100.0	92.9	73.3	80.0	92.3	75.0
	2	83.3	100.0	100.0	100.0	100.0	50.0	80.0	100.0
	3	75.0	90.9	100.0	100.0	80.0	87.5	73.3	84.6
	4	75.0	100.0	66.7	100.0	88.9	83.3	92.3	88.9
	5	58.3	83.3	76.9	94.4	100.0	100.0	100.0	91.7
	6	75.0	53.8	54.5	82.4	88.2	100.0	57.1	90.0
	7	63.6	90.0	80.0	57.1	71.4	85.7	100.0	75.0
	8	36.4	46.7	69.2	88.9	90.0	84.6	66.7	100.0
		1	2	3	4	5	6	7	8
		ターゲット領域							

### 位置重み付けアルゴリズム

開始領域	1	100.0	92.3	75.0	100.0	88.2	62.5	50.0	60.0
	2	90.9	100.0	92.9	94.1	76.9	87.5	78.6	46.7
	3	65.0	75.0	100.0	100.0	80.0	77.8	87.5	43.8
	4	66.7	71.4	81.3	100.0	87.5	94.1	93.3	41.2
	5	50.0	70.0	75.0	88.9	100.0	88.9	53.8	46.2
	6	77.8	84.6	83.3	90.0	75.0	100.0	100.0	75.0
	7	75.0	80.0	88.9	83.3	80.0	100.0	100.0	100.0
	8	50.0	100.0	nun	100.0	85.7	100.0	100.0	100.0
		1	2	3	4	5	6	7	8
		ターゲット領域							

図 6.5: 各アルゴリズムにおけるポインタ移動の成功率。

1	100.0	93.3	75.0	93.3	82.4	81.8	77.8	100.0
2	92.9	100.0	93.8	100.0	88.2	90.9	77.8	60.0
3	72.2	85.7	100.0	100.0	81.8	81.8	92.9	76.5
4	77.8	87.5	90.0	100.0	83.3	93.8	86.7	73.7
5	75.0	91.7	73.7	92.9	100.0	91.7	85.7	66.7
6	85.7	83.3	66.7	80.0	81.8	100.0	100.0	75.0
7	90.0	85.7	85.7	75.0	77.8	100.0	100.0	100.0
8	50.0	100.0	100.0	100.0	81.8	75.0	100.0	100.0
	1	2	3	4	5	6	7	8

図 6.6: 各アルゴリズムにおけるポインタ移動の成功率.

に関係していた. なお, 12 中 8 名についてアンケートに使いやすいと答えたアルゴリズムと図 6.4 の誤差距離が小さなアルゴリズムが一致していた.

#### 本システムの使いやすさについて

本システムの使いやすさを 5 段階にて評価してもらった (5 が使いやすい). その結果, 5 人の実験協力者の評価は「4」, 3 人の実験協力者の評価は「3」, 4 人の実験協力者の評価は「2」となった. また, 本システムについて意見も述べてもらった. 得られた肯定的な意見としては「操作が直観的で良い」, 「片手でスマートウォッチができるのが良い」というものがあった. また, 否定的な意見としては「長時間の使用は辛い」, 「決定動作の誤判定があった」, 「画面端にポインタを動かすことが難しい」, 「決定動作時にポインタが大きくなる」というものであった.

## 6.5 考察

親指動作認識システムおよびこれを用いたスマートウォッチ操作に関して、実験結果を考察する。

### 6.5.1 親指動作認識システムに関する考察

実験データの解析およびアンケート結果から、決定動作が親指動作認識に大きな影響を与えることが確認された。そのため、今後は決定動作時の影響を考慮し、選択位置を決定すべきであることが示唆された。具体的には、決定動作から起こる手首形状変化への影響を調べ、これをもとに加速度変化が観測されてから数フレーム前の親指位置を選択位置として決定する。

図 6.4 より、各実験協力者ごとに精度の良い認識アルゴリズムは違うことが確認された。そのため、今後はユーザごとに適切な認識アルゴリズムを選択する必要性が示唆された。

解析により、実験協力者ごとに適切な親指移動認識アルゴリズムを選択した場合、実験協力者は平均 87.0 % (SD=11.4) の精度にてポインタを任意の領域に移動させることができることがわかった。しかし、図 6.5 より、いずれのアルゴリズムにおいてもスマートウォッチ画面の端に向かうほどポインタ移動の成功率が低くなることが確認された。この原因の 1 つとして、親指移動時における決定動作の誤判定により、ポインタが画面端に到達する前に選択が行われた可能性ある。この誤判定は画面端にターゲットが現れた際、勢いをつけて親指を動かすことにより、手首形状が急激に変化し、加速度変化が決定動作判定の閾値を超えてしまうことにより発生すると考えられる。この問題については、親指移動と決定動作それぞれの手首形状変化を区別することにて対応する必要がある。また、2 つ目の原因として、システムの特長として画面端にポインタが動かしにくいことが考えられる。本システムは、キャリブレーションより得たサンプル波形と推定に用いるセンサ値を比較することにより、親指位置の推定を行う。しかし、腕の角度や力み具合が変化することにより、センサ値の変化量がキャリブレーション時よりも小さくなると、十分なセンサ値変化が得られず画面端にポインタが行かなくなる可能性がある。特に実験においては、疲労により、力み具合が変化する可能性が大きい。この問題については、定期的にサンプル波形を更新するなどの工夫をすることにて対応する必要がある。

### 6.5.2 本システムを用いたスマートウォッチ操作に関する考察

アンケート結果から、親指スライドジェスチャを用いたスマートウォッチ操作自体に関しては「直観的で良い」という評価が得られた。ただし、アンケートからは疲労に関しての否定的な評価も得られた。この疲労の原因は実験環境によるものが考えられる。実験環境による疲労の原因の 1 つとして腕を上げ続けることに起因するものがある。本実験においては、実験協力者には 1 セッション中、腕を常に胸の前に構えた状態にてターゲット選択を行ってもらった。そのため、実験協力者はこの姿勢を続けることにより疲労を感じたと考えられる。また、疲労の原因として認識精度に起因するものがある。実験の様子を観察した結果、うまく

画面端にポインタが動かない場合、実験協力者は繰り返し親指を端に動かしていることが確認された。また、決定動作が認識されない場合も同様に、実験協力者は繰り返し決定動作を行っていた。これら実験環境による疲労は、今後実験条件や認識システムの改良を行うことにより、解決できると考えられる。

またアンケートからは認識精度が原因として使いづらいという否定的な評価も得られた。このような評価を得た原因として決定動作によるポインタのずれや画面端にポインタが行かないことが影響したと考えられる。今回の実験においては、画面全体にターゲットを表示し、実験協力者にこれを選択させた。しかし、このような細かな選択を要する操作は現状のシステムにおいては適していなかったと考える。そこで、本システムを用いるのに適したUIを設計することにより、ポインタのずれや画面端にポインタが行かない問題を解決する。具体的なUI設計を図6.7に示す。本UI設計においては親指動作認識システムから推定した親指位置を3領域に分割する(図6.7A)。本UI設計を用いてユーザがターゲットを選択する際、ユーザはまず親指を右(左)の領域に親指を動かすことにてターゲットを右(左)にスクロールさせ(図6.7B)、決定動作にて画面中央のターゲットを選択する。本UI設計はターゲットサイズを大きくすることにより、ポインタのずれを許容する。また、ターゲットをスクロールさせることにより、画面端にポインタが行かないことを許容する。

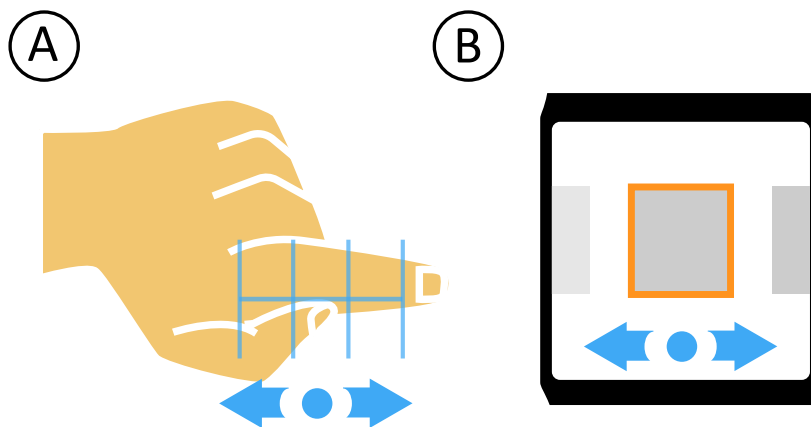


図 6.7: 実験結果をもとに設計した UI. A) 親指動作認識システムから推定した親指位置を 3 領域に分割. B) 親指の位置に応じてターゲットをスクロール.

## 第7章 アプリケーション例

本認識手法の応用可能性を示すため、第6章の評価実験の結果に基づいたアプリケーションを作製した。

### 7.1 選択アプリケーション

6.5節にて述べたUI設計を持つ選択アプリケーションを作製した(図7.1)。このアプリケーションにおいて、ユーザは親指を左右に動かしてメニューをスクロールし、中心にあるターゲットを選択する。スクロール操作を使用することにより、ターゲットサイズを大きくすることができる。これにより、親指スライドジェスチャにより推定位置がずれてしまっても、そのずれを許容することができる。



図 7.1: 選択アプリケーション。

## 7.2 通知アプリケーション

親指スライドジェスチャを用いたスクロール操作によりメールの通知表示を行うアプリケーションを製作した。親指スライドジェスチャを用いたスマートウォッチ入力は、タッチディスプレイを用いた入力と比べて予備動作が小さいため、素早く操作に移行できる。そのため、ユーザは親指スライドジェスチャによりスクロール操作を行うことにより、メールやSNSメッセージなどの通知内容を素早く確認することが可能となる。

図7.2に製作したメールアプリケーションを示す。本アプリケーションにおいては、親指の左右への移動量を上下へのスクロールの速度として利用している。本認識手法は、このようなスマートウォッチに文章を通知するアプリケーションとの相性が良いと考える。



図 7.2: スクロールアプリケーション。



### 7.3 ゲームアプリケーション

親指スライドジェスチャによる絶対ポインティングを用いたゲームアプリケーションを製作した。親指スライドジェスチャを用いたスマートウォッチ入力では、指がディスプレイを遮る問題を解決する。そのため、本手法はシューティングゲームのような連続的な入力を必要とし、かつ、常に画面の状況を把握する必要があるアプリケーションに対して有用である。図7.3に作成したシューティングゲームアプリケーションを示す。本アプリケーションにおいては、ユーザは親指スライドジェスチャを用いて自機の左右移動を行う（弾は自動で射出される）。本認識手法は、このような左右移動を伴うゲーム（ブロック崩しなど）などのコントローラとして使用されることが期待される。



図 7.3: ゲームアプリケーション。

## 第8章 親指動作の二次元方向認識への拡張

第5章にて述べたシステムの認識方向を二次元方向に拡張するため、手を握りこんだ際にできる人差し指側面平面上における親指の位置を識別することを試みた。この識別が実現されることにより、ユーザは親指の動きをジョイスティックの動きに見立てて、二次元方向への入力を行うことが可能となる（図8.1）。本章においては、親指位置を識別できるかを評価するために行った著者実験について述べるとともに、実験結果をもとに実装を行った親指動作の二次元方向認識システムについて述べる。

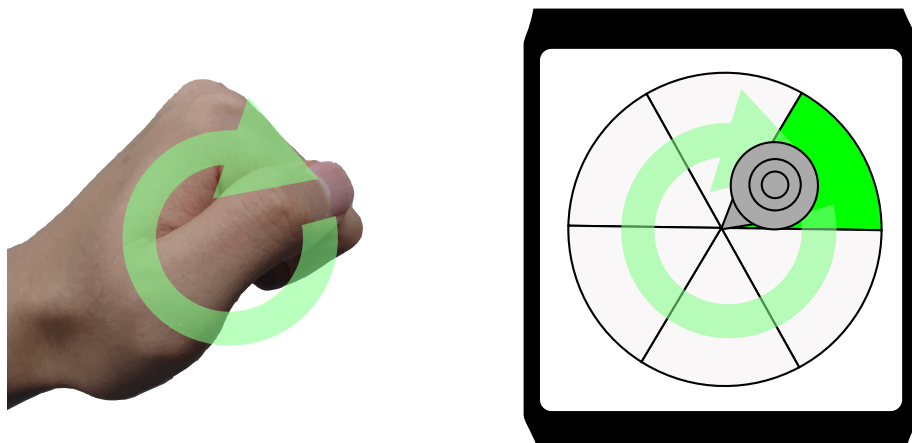


図 8.1: 円形学習による二次元認識への応用イメージ。

### 8.1 親指二次元位置の識別精度評価

製作したセンサデバイスから得たセンサ値をもとに親指位置の識別が行えるかを評価するため、著者1名（男性、年齢24歳、右利き）を実験協力者とした実験を行った。本実験が識別の対象とするのは、手を握った際にできる人差し指側面の平面上における親指の位置である。ここで述べる親指の位置とは、図8.2のように親指を人差し指の側面で回転させるようになぞった際の円周上の各点（P0-P29）および円の中心（P30）のことを指す。以降、P0-P29を円周点、P30を中心点と呼称する。

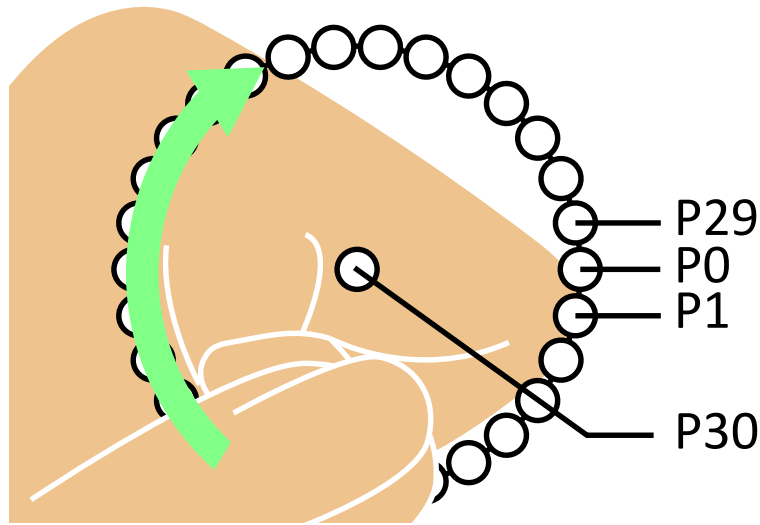


図 8.2: 識別対象とする親指の位置.

### 8.1.1 実験アプリケーション

実験アプリケーションは Processing (Version 2.2.1) にて実装された。図 8.3 に本アプリケーションのスクリーンショットを示す。本アプリケーション画面には、円形状に「○」印が並べて表示されている。実験が開始されると、右端の印（時計における 3 時の位置の印）が赤くハイライトされる。この位置におけるセンサ値の記録が終わると、時計回りにハイライトされる印が切り替わる。実験協力者がこのハイライトされた印の位置に合わせて親指を動かすことにより、アプリケーションは円周点におけるセンサ値を記録する。また、本アプリケーションは円周点におけるセンサ値の記録後、実験協力者が親指を中心点に動かし、ボタンを押すことにより、中心点におけるセンサ値を記録する。

### 8.1.2 実験環境

実験時の姿勢を図 8.4A に示す。本実験は、実験協力者が椅子に深く腰掛けた状態にて行われた。また、実験協力者の左手首にはセンサデバイスが取り付けられたスマートウォッチが装着された。このセンサデバイスにおける各センサの位置を図 8.4B に示す。なお、本実験においては BLE による通信ではなく、通信速度がより速い USB ケーブルによる通信にてセンサ値を PC に転送した。

### 8.1.3 実験手順

実験手順を説明する。まず、実験協力者はセンサデバイスを装着した左手及び腕を図 8.4A のように胸の前に構えた姿勢をとる。またこの際、左手は軽く握る。次に、実験協力者は実

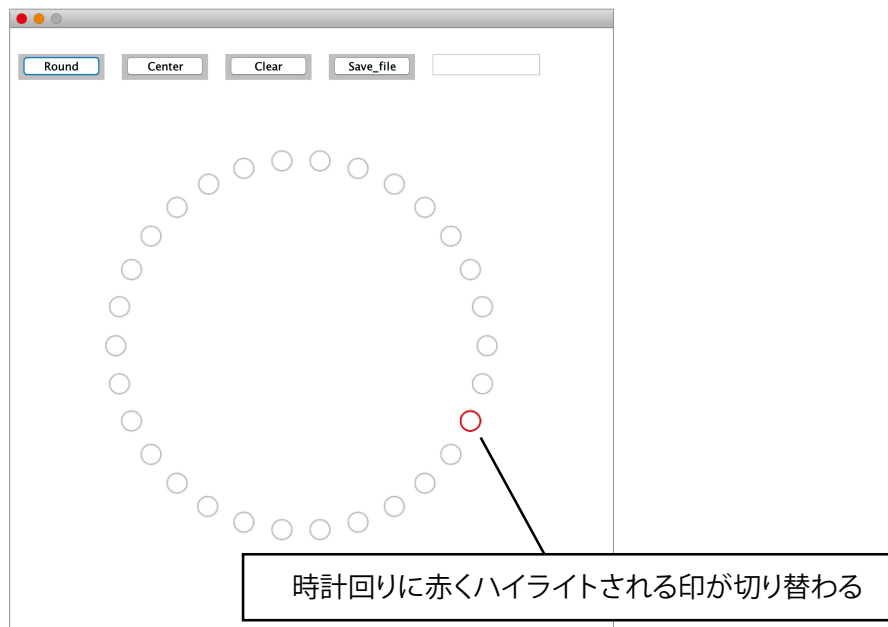


図 8.3: 実験に使用したアプリケーション.

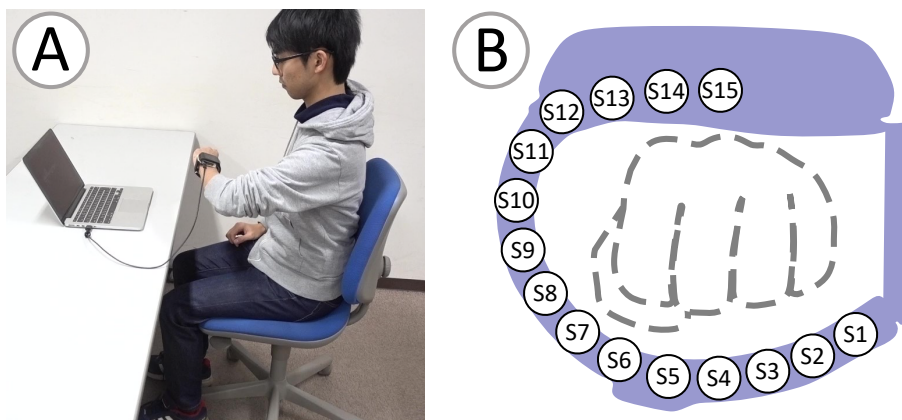


図 8.4: 実験環境 : A) 実験時の姿勢, B) センサデバイスにおける各センサの位置.

験アプリケーション上のボタンをマウスにてクリックし、円周点におけるセンサ値の記録を開始する。記録開始後、実験協力者は赤くハイライトされた「○」印の位置に合わせて、親指を移動させる。本実験においては、円周点におけるセンサ値の記録を12周分連続して行った（P29の測定後すぐにP0から記録を再開する）。なお、円周点を1周記録するのにかかる時間はおよそ2.0秒である。

円周点におけるセンサ値を記録後、実験協力者は中心点におけるセンサ値を記録する。実験協力者はまず、親指を中心点（P30）に動かし、中心点記録用のボタンをクリックする。ボタンがクリックされた際、1回分のセンサ値が記録される。その後、実験協力者は一度親指をP0の位置に動かし、再度中心点に親指を動かしてボタンをクリックする。実験協力者はこの中心点におけるセンサ値の記録を、10回繰り返す。

#### 8.1.4 結果及び考察

実験から得たデータから親指位置の識別精度を評価するため、機械学習ソフトウェアである Weka Machine Learning Toolkit [HFH<sup>+</sup>09] を用いて、10-fold 交差検定を行った。交差検定に用いたデータは、円周点の記録における最初と最後の周を除いた10周分のデータ（10周 × 30点 × 16次元のセンサ値データ）および中心点における10回分のデータ（10回 × 1点 × 16次元のセンサ値データ）である。また、分類器には LIBSVM [CL11] を使用した。なお、円周点の記録において最初と最後の周のデータを除いた理由は、動作の開始および終了時のデータを除くためである。

交差検定の結果を表 8.1 に示す。検定の結果、親指位置の識別率は16.1%と低い識別率であった。しかし、各円周点の識別において大きな識別位置のずれはなく、いずれも付近の円周点との誤識別が起こっていた。これは、円周点におけるセンサ値の記録の際、周回ごとに親指を動かす速度がずれることにより発生した可能性がある。なお、各円周点の識別において、ずれが24度以内に収まる確率（すなわち、両隣を含めた3点のいずれかに識別される確率）は76.8%であった。また、識別のずれが48度以内に収まる確率（すなわち、両側2つ隣まで含めた5点のいずれかに識別される確率）は91.3%であった。

以上の結果より、現状のセンサ値の記録（学習）方法においても、およそその親指位置の識別は可能であると考えられる。特に、中心点の識別率は90%であり、中心点からどの方向に親指が動いたかを認識することが可能であることが示唆される。

## 8.2 二次元方向親指動作認識システム

実験の結果をもとに、機械学習による親指位置推定に基づく二次元方向親指動作認識システムを実装した。本システムは、親指の動きをジョイスティックの動きに見立てることにて、二次元方向への入力を可能にするシステムである。本節においては、システム構成および使用手順を述べる。

表 8.1: 親指位置の識別結果の混同行列.

識別結果 (%)

	P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20	P21	P22	P23	P24	P25	P26	P27	P28	P29	P30	
P0	20	60			10																										10	
P1	50	20	20			10																										
P2		20	30	40	10																											
P3			50	10	30	10																										
P4			10	20	10	30	20	10																								
P5			10	10	30	20	30																									
P6				10	20	20	20	20		10																						
P7					10	10	20	20	10	20														10								
P8						10	10	10	20	30	20																					
P9							30	10	20	10	30																					
P10								30	20		30	20																				
P11									30	30		40																				
P12									10	10	30		50																			
P13											10	50		40																		
P14														40	60																	
P15														10	50	20	10	10														
P16																30	50	10	10													
P17																20	30	10	20	10		10										
P18																	10	20	30	20			10	10								
P19																		40	20	10	20	10										
P20																		10		10	30	40			10							
P21																						20	40	30	10							
P22																							40	20	40							
P23																	10	10					20	20		40						
P24																							10		20	20	40	10				
P25																							10		30	20	40					
P26																										20		70	10			
P27																											60		40			
P28	10	10																										30	20	30		
P29	20	10																											60	10		
P30																								10								90

実際の親指位置

### 8.2.1 システム構成

実装したシステムは、手首形状測定用のセンサデバイスおよび、親指動作認識処理用のスマートフォン (SONY Xperia Z5 E6653; Android 6.0.1), 画面表示用のスマートウォッチ (SONY SmartWatch 3 SWR50; Android Wear 1.5.0) から成る. センサデバイスから得られたセンサ値は BLE 通信を用いてスマートフォンに送られ, そこで親指動作認識にかかわる処理が行われる. また, スマートフォンにて処理された結果は, スマートウォッチに表示される.

### 8.2.2 親指位置の学習

二次元方向親指動作認識システムは, 実験時と同様に各円周点におけるセンサ値と中心点におけるセンサ値を学習させることにより, 親指の位置を識別する. なお, 本実装においては, 機械学習ライブラリとして, LIBSVM [CL11] を用いた.

図 8.5A にスマートフォンに表示されるアプリケーション画面を示す. ユーザは実験と同様の手順により, 親指位置の学習を行う (すなわち, 円周点の学習後, 中心点の学習を行う). 学習終了後, アプリケーションは親指位置の識別を行う. 本アプリケーションにおいて, 識別された円周点は緑色の「○」にて表示されている (図 8.5B). またこの際, アプリケーションは識別時における尤度 (0-1 の値にて表される) を重みとして, 親指位置を推定する. すなわち, 親指の位置 ( $X$  座標および  $Y$  座標) は以下のような式にて表される.

$$X = \sum_{i=P_0}^{P_{29}} \{Prob_i \times \cos(Rad_i)\} \times r \quad (-r < X < r)$$
$$Y = \sum_{i=P_0}^{P_{29}} \{Prob_i \times \sin(Rad_i)\} \times r \quad (-r < Y < r)$$

この式において,  $Prob_i$  は円周点  $i$  における尤度を表し,  $Rad_i$  は円周点  $i$  における弧度を表す. また,  $r$  は円の半径 (中心点から各円周点までの距離) を表す. なお, この処理は, 実験時における識別位置のずれを緩和する目的, および中心点と円周点間の親指位置を補完することを目的として行われる. 本アプリケーションにおいて, 推定された親指位置は赤色の「●」にて表示されている (図 8.5B).

### 8.2.3 スマートウォッチ画面への表示

スマートフォンにて推定した親指位置をスマートウォッチ画面に表示する. 図 8.6 にスマートウォッチに表示されるアプリケーション画面を示す. スマートウォッチ画面には, 親指位置をジョイスティックの位置に見立てたコントローラが表示されている. ユーザはこの仮想ジョイスティックを親指を動かして操作することにより, スマートウォッチへの二次元入力を行うことができる.





## 第9章 議論

### 9.1 決定操作の改良

第6章より、加速度変化のみを用いた決定動作認識システムは、親指の移動量推定に影響を与えることが確認された。この問題を解決する方法として、親指の移動量推定時にスマートウォッチの加速度変化も考慮することにて、決定動作の影響を除くことが考えられる。また、閾値ではなく LIBSVM [CL11] などの機械学習ライブラリを用いて、決定動作認識の精度を向上させることにより、認識に必要な決定動作の大きさを小さくすることも考えられる。

### 9.2 適切なアルゴリズムの選択

第6章より、ユーザごとに親指スライドジェスチャの認識に適切なアルゴリズムが異なることが確認された。このアルゴリズムの選択に関してはユーザが各々使いやすいアルゴリズムを選択できるようにすることにより、この問題を解決できると考える。しかし、この選択にはユーザに手間を強いる。そのため、今後はユーザごとの特徴を調べ、アルゴリズム選択を自動化する方法を探る必要がある。

### 9.3 入力動作の併用

本論文においては、親指スライドジェスチャと併用して使用する決定動作として「親指タップジェスチャ」、「握りジェスチャ」を示した。これら入力動作は同時に使用することが可能である。そのため、親指タップジェスチャを決定操作、握りジェスチャを取り消し操作に割り当てるといったことが可能である。また、手を開く動作など、その他の入力動作とも併用して使用できると考える。そのため、今後は親指スライドジェスチャと併用できる入力動作を複数組み合わせることにて、スマートウォッチへの入力語彙を拡張することを検討している。

### 9.4 入力状態と非入力状態の区別

スマートウォッチは常に身に着けているデバイスであるため、本手法を用いる状態（入力状態）とそうでない状態（非入力状態）を区別する必要がある。特に、本手法が親指の移動量推定に用いる手首形状の変化は、腕の回転や手の姿勢変化と比べて小さいため、日常の多くの動作に影響を受けると考えられる。しかし、スマートウォッチを利用する際にユーザは腕を

構えるため、本手法が利用される姿勢は一意に定まる。このため、腕の姿勢とその他の指の姿勢を固定した状態、すなわち親指動作以外の手首形状に影響を与える要素を固定した状態をシステムの起動姿勢とすることにより、シームレスに入力状態と非入力状態を区別できると考える。具体的には、腕を胸の前に構え、人差し指のみを伸ばした姿勢を起動姿勢とすることを考えている。なお、Fukui ら [FWSS15] が示すように手首の形状は腕の回転角度と手の姿勢情報を持っているため、手首の形状から特定の腕の姿勢及び手の姿勢を識別可能である。また、親指を動かした際の手首の形状変化は腕の回転や手の握り動作から起こる手首の形状変化と比べて小さいため、これらの識別に影響を与えないと考えられる。ただしこのような識別方法が日常生活において実際に使用できるのかを今後調査する必要がある。

## 9.5 日常生活における利用

親指スライドジェスチャを用いたスマートウォッチ入力は電車やバスなどの振動を伴う環境において利用されることが予想される。そのため、今後このような環境においても、本認識手法が利用可能であるかを調査する必要がある。また、かばんを持った状態においても利用できるかを調査する必要がある。

## 9.6 手法の適用範囲について

本認識手法が利用する手首の形状変化は前腕部の筋肉の動きから起こる。そのため、筋肉量の少ない女性や子どもにおいては、十分にセンサ値の変化が得られない可能性がある。この問題に関しては、今後より多くの女性や子どもなどに対しても実験を行い、親指を動かした際の筋肉の動きに関して調査を行う必要がある。

## 第10章 結論

本論文では、スマートウォッチと呼ばれる手首装着型端末を操作するための入力動作「親指スライドジェスチャ」を示し、その認識手法を示した。親指スライドジェスチャは、スマートウォッチを装着した側の手の親指を人差し指の側面をなぞるように動かす動作のことを指す。ユーザはこのような動作を用いることにより、スマートウォッチへの一次元の連続的な入力操作が可能となる。この親指スライドジェスチャは決定操作用の動作（親指タップジェスチャ、握りジェスチャ）と組み合わせることにより、ユーザが片手のみにてスマートウォッチを操作することを可能とする。

本研究では、この親指スライドジェスチャを認識するため、赤外線近接センサを用いたベルト型センサデバイスを製作した。また、このデバイスを用いて、親指動作に伴う手首の形状変化を測定する実験を行った。さらに、この実験から得たデータをもとに、親指動作認識システムを開発し、その評価実験を行った。評価実験から、本システムは平均 87.0 % (SD=11.4) の精度にて、任意の領域にポインタを移動させられることを示した。また、評価実験から得た知見をもとに、本システムに適する UI を設計した。さらに、本認識手法を用いたアプリケーション例、および認識方向を二次元方向に拡張する方法を示した。

## 謝辞

本研究を進めるにあたり，志築文太郎先生，高橋伸先生および田中二郎先生には多くのご意見とご指導を頂きました。特に，志築文太郎先生には，日頃から研究の相談や論文執筆に関する事等，数えきれないほど多くのご指導を頂きました。無事本論文を執筆することができたのは，志築文太郎先生のご助力があつてのことです。深く感謝致します。

また，IPLABの皆様には，研究活動において様々なご助言を頂きました。特に WAVE チームの皆様には，チームゼミをはじめ，研究の進め方や文章の書き方，アプリケーションの実装まで様々なアドバイスを頂きました。論文執筆の際には，忙しい中，多くのご助言や添削をしていただきました。深く感謝致します。また，実験協力者としてご協力頂いた皆様に，感謝申し上げます。

最後に，筆者の生活面，精神面を支えてくださった家族と友人，大学院生活においてお世話になったすべての方々に心より感謝致します。

## 参考文献

- [AHT02] Brian Amento, Will Hill, and Loren Terveen. The Sound of One Hand: A wrist-mounted bio-acoustic fingertip gesture interface. In *CHI '02 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '02, pp. 724–725, New York, NY, USA, 2002. ACM.
- [CCH<sup>+</sup>15] Liwei Chan, Yi-Ling Chen, Chi-Hao Hsieh, Rong-Hao Liang, and Bing-Yu Chen. CyclopsRing: Enabling whole-hand and context-aware interactions through a fisheye ring. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology*, UIST '15, pp. 549–556, New York, NY, USA, 2015. ACM.
- [CL11] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, Vol. 2, No. 3, pp. 27:1–27:27, May 2011. ACM.
- [DP14] Artem Dementyev and Joseph A. Paradiso. WristFlex: Low-power gesture input with wrist-worn pressure sensors. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, pp. 161–166, New York, NY, USA, 2014. ACM.
- [FWSS15] Rui Fukui, Masahiko Watanabe, Masamichi Shimosaka, and Tomomasa Sato. Hand shape classification in various pronation angles using a wearable wrist contour sensor. *Advanced Robotics*, Vol. 29, No. 1, pp. 3–11, 2015.
- [GP16] 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, pp. 17–28, New York, NY, USA, 2016. ACM.
- [GYI16] Jun Gong, Xing-Dong Yang, and Pourang Irani. WristWhirl: One-handed continuous smartwatch input using wrist gestures. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, pp. 861–872, New York, NY, USA, 2016. ACM.
- [GZZY17] Jun Gong, Yang Zhang, Xia Zhou, and Xing-Dong Yang. Pyro: Thumb-tip gesture recognition using pyroelectric infrared sensing. In *Proceedings of the 30th Annual*

*ACM Symposium on User Interface Software and Technology*, UIST '17, pp. 553–563, New York, NY, USA, 2017. ACM.

- [HAL15] Jaehyun Han, Sunggeun Ahn, and Geehyuk Lee. Transture: Continuing a touch gesture on a small screen into the air. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '15, pp. 1295–1300, New York, NY, USA, 2015. ACM.
- [HCY<sup>+</sup>16] 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. ACM.
- [HFH<sup>+</sup>09] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA data mining software: An update. Vol. 11, pp. 10–18, New York, NY, USA, November 2009. ACM.
- [HH09] Chris Harrison and Scott E. Hudson. Abracadabra: Wireless, high-precision, and unpowered finger input for very small mobile devices. In *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology*, UIST '09, pp. 121–124, New York, NY, USA, 2009. ACM.
- [HHN<sup>+</sup>17] Teng Han, Khalad Hasan, Keisuke Nakamura, Randy Gomez, and Pourang Irani. SoundCraft: Enabling spatial interactions on smartwatches using hand generated acoustics. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, UIST '17, pp. 579–591, New York, NY, USA, 2017. ACM.
- [HRS<sup>+</sup>14] Md. Mehedi Hasan, Arifur. Rahaman, Md. Faisal Shuvo, Md. Abu Saleh Ovi, and Md. Mostafizur Rahman. Human hand gesture detection based on EMG signal using ANN. In *Informatics, Electronics Vision (ICIEV), 2014 International Conference on*, ICIEV'14, pp. 1–5, 2014.
- [HSP10] Loclair Hristian, Gustafson Sean, and Baudisch Patrick. PinchWatch: A wearable device for one-handed microinteractions. In *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services workshop on Ensembles of on-body devices*, MobileHCI '10 workshop on Ensembles of on-body devices, pp. 1–4, New York, NY, USA, 2010. ACM.
- [HZS<sup>+</sup>15] Donny Huang, Xiaoyi Zhang, T. Scott Saponas, James Fogarty, and Shyamnath Gollakota. Leveraging dual-observable input for fine-grained thumb interaction using

- forearm EMG. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology*, UIST '15, pp. 523–528, New York, NY, USA, 2015. ACM.
- [Inca] Apple Inc. Watch - apple. <https://www.apple.com/watch/> (2017/12/13 accessed).
- [Incb] Google Inc. Android wear. <https://www.android.com/wear/> (2017/12/13 accessed).
- [Incc] Thalmic Labs Inc. Myo gesture control armband — wearable technology by thalmic labs. <https://www.myo.com> (2017/12/19 accessed).
- [JCZ<sup>+</sup>13] Lei Jing, Zixue Cheng, Yinghui Zhou, Junbo Wang, and Tongjun Huang. Magic Ring: A self-contained gesture input device on finger. In *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia*, MUM '13, pp. 39:1–39:4, New York, NY, USA, 2013. ACM.
- [KHI<sup>+</sup>12] David Kim, Otmar Hilliges, Shahram Izadi, Alex D. Butler, Jiawen Chen, Iason Oikonomidis, and Patrick Olivier. Digits: Freehand 3D interactions anywhere using a wrist-worn gloveless sensor. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, pp. 167–176, New York, NY, USA, 2012. ACM.
- [KLK15] Frederic Kerber, Pascal Lessel, and Antonio Krüger. Same-side hand interactions with arm-placed devices using EMG. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '15, pp. 1367–1372, New York, NY, USA, 2015. ACM.
- [KMPB<sup>+</sup>14] Jarrod Knibbe, Diego Martinez Plasencia, Christopher Bainbridge, Chee-Kin Chan, Jiawei Wu, Thomas Cable, Hassan Munir, and David Coyle. Extending Interaction for Smart Watches: Enabling Bimanual Around Device Control. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '14, pp. 1891–1896, New York, NY, USA, 2014. ACM.
- [LGK<sup>+</sup>16] Jaime Lien, Nicholas Gillian, M. Emre Karagozler, Patrick Amihood, Carsten Schweisig, Erik Olson, Hakim Raja, and Ivan Poupyrev. Soli: Ubiquitous gesture sensing with millimeter wave radar. *ACM Transactions on Graphics*, Vol. 35, No. 4, pp. 142:1–142:19, July 2016. ACM.
- [LWH<sup>+</sup>15] Jhe-Wei Lin, Chiuan Wang, Yi Yao Huang, Kuan-Ting Chou, Hsuan-Yu Chen, Wei-Luan Tseng, and Mike Y. Chen. BackHand: Sensing hand gestures via back of the

- hand. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology*, UIST '15, pp. 557–564, New York, NY, USA, 2015. ACM.
- [LXC<sup>+</sup>14] Gierad Laput, Robert Xiao, Xiang ‘Anthony’ Chen, Scott E. Hudson, and Chris Harrison. Skin Buttons: Cheap, small, low-powered and clickable fixed-icon laser projectors. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, pp. 389–394, New York, NY, USA, 2014. ACM.
- [LXH16] 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. ACM.
- [MMF<sup>+</sup>16] Jess McIntosh, Charlie McNeill, Mike Fraser, Frederic Kerber, Markus Löchtefeld, and Antonio Krüger. EMPress: Practical hand gesture classification with wrist-mounted EMG and pressure sensing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pp. 2332–2342, New York, NY, USA, 2016. ACM.
- [MMF17] Jess McIntosh, Asier Marzo, and Mike Fraser. SensIR: Detecting hand gestures with a wearable bracelet using infrared transmission and reflection. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, UIST '17, pp. 593–597. ACM, 2017.
- [OARSM15] Santiago Ortega-Avila, Bogdana Rakova, Sajid Sadi, and Pranav Mistry. Non-invasive optical detection of hand gestures. In *Proceedings of the 6th Augmented Human International Conference*, AH '15, pp. 179–180, New York, NY, USA, 2015. ACM.
- [OI15] Masa Ogata and Michita Imai. SkinWatch: Skin gesture interaction for smart watch. In *Proceedings of the 6th Augmented Human International Conference*, AH '15, pp. 21–24, New York, NY, USA, 2015. ACM.
- [OSOI12] Masa Ogata, Yuta Sugiura, Hirotaka Osawa, and Michita Imai. iRing: Intelligent Ring Using Infrared Reflection. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, pp. 131–136, New York, NY, USA, 2012. ACM.
- [PFHP99] John Kangchun Perng, Brian Fisher, Seth Hollar, and Kristofer S. J. Pister. Acceleration sensing glove (ASG). In *The International Symposium on Wearable Computers*, ISWC '99, pp. 178–180, 1999. IEEE.



- [PVBH14] Manuel Prätorius, Dimitar Valkov, Ulrich Burgbacher, and Klaus Hinrichs. DigiTap: An eyes-free VR/AR symbolic input device. In *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology, VRST '14*, pp. 9–18, New York, NY, USA, 2014. ACM.
- [Rek01] Jun Rekimoto. GestureWrist and GesturePad: Unobtrusive Wearable Interaction Devices. In *Proceedings of the 5th IEEE International Symposium on Wearable Computers, ISWC '01*, pp. 21–27, Washington, DC, USA, 2001. IEEE Computer Society.
- [SRC05] 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.
- [SWY<sup>+</sup>17] 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*, pp. 692–704, New York, NY, USA, 2017. ACM.
- [TYR17] Yanke Tan, Sang Ho Yoon, and Karthik Ramani. Bikegesture: User elicitation and performance of micro hand gesture as input for cycling. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems, CHI EA '17*, pp. 2147–2154, New York, NY, USA, 2017. ACM.
- [WKTA15] Mathias Wilhelm, Daniel Krakowczyk, Frank Trollmann, and Sahin Albayrak. eRing: Multiple finger gesture recognition with one ring using an electric field. In *Proceedings of the 2nd International Workshop on Sensor-based Activity Recognition and Interaction, WOAR '15*, pp. 7:1–7:6, New York, NY, USA, 2015. ACM.
- [WLB<sup>+</sup>15] Martin Weigel, Tong Lu, Gilles Bailly, Antti Oulasvirta, Carmel Majidi, and Jürgen Steimle. iSkin: Flexible, stretchable and visually customizable on-body touch sensors for mobile computing. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI '15*, pp. 2991–3000, New York, NY, USA, 2015. ACM.
- [WNRM11] Katrin Wolf, Anja Naumann, Michael Rohs, and Jörg Müller. A Taxonomy of Microinteractions: Defining microgestures based on ergonomic and scenario-dependent requirements. In *Proceedings of the 13th IFIP TC 13 International Conference on Human-computer Interaction - Volume Part I, INTERACT'11*, pp. 559–575, Berlin, Heidelberg, 2011. Springer-Verlag.
- [WPSN15] Anusha Withana, Roshan Peiris, Nipuna Samarasekara, and Suranga Nanayakkara. zSense: Enabling shallow depth gesture recognition for greater input expressivity on

smart wearables. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pp. 3661–3670, New York, NY, USA, 2015. ACM.

- [XLH14] Robert Xiao, Gierad Laput, and Chris Harrison. Expanding the input expressivity of smartwatches with mechanical pan, twist, tilt and click. In *Proceedings of the 32rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '14, pp. 193–196, New York, NY, USA, 2014. ACM.
- [XPM15] 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. ACM.
- [ZH15] Yang Zhang and Chris Harrison. Tomo: Wearable, low-cost electrical impedance tomography for hand gesture recognition. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology*, UIST '15, pp. 167–173, New York, NY, USA, 2015. ACM.
- [ZZLH16] Yang Zhang, Junhan Zhou, Gierad Laput, and Chris Harrison. SkinTrack: Using the body as an electrical waveguide for continuous finger tracking on the skin. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pp. 1491–1503, New York, NY, USA, 2016. ACM.
- [塚田 02] 塚田浩二, 安村通晃. Ubi-finger : モバイル指向ジェスチャ入力デバイスの研究. 情報処理学会論文誌, Vol. 43, No. 12, pp. 3675–3684, 2002. 一般社団法人情報処理学会.
- [福本 99] 福本雅朗, 外村佳伸. 指釦 : 手首装着型コマンド入力機構. 情報処理学会論文誌, Vol. 40, No. 2, pp. 389–398, 1999. 一般社団法人情報処理学会.

# 著者論文リスト

## 参考論文

本論文の主内容は，下記(1)(2)(3)として公表済みである。

- 査読付き国内論文誌論文（公表済み論文）
  - (1) 青山周平, 阿部哲也, 志築文太郎, 田中二郎. 親指の移動量推定に基づくスマートウォッチの片手操作手法. コンピュータソフトウェア, Vol. 34, No. 2, pp. 102–113, 2017年5月. 日本ソフトウェア科学会.
- 査読付き国際会議論文（公表済み論文）
  - (2) Shuhei Aoyama, Buntarou Shizuki, Jiro Tanaka. ThumbSlide: An Interaction Technique for Smartwatches using a Thumb Slide Movement. In Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '16), pp. 2403–2409, May 2016. ACM.
- 査読付き国内会議論文（公表済み論文）
  - (3) 青山周平, 阿部哲也, 志築文太郎, 田中二郎. ThumbSlide: 親指スライドジェスチャによるスマートウォッチ操作手法. 第23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論文集, pp. 79-84, 2015年12月. 日本ソフトウェア科学会.

## その他の論文

- 査読付き国際会議論文（公表済み論文）
  - (1) Masaya Tsuruta, Shuhei Aoyama, Arika Yoshida, Buntarou Shizuki, and Jiro Tanaka. Hover Detection using Active Acoustic Sensing. In Proceedings of 18th International Conference on Human-Computer Interaction (HCI International 2016), pp. 104-114, July 2016. Springer International Publishing.
- 査読付き国際会議論文（採録決定論文）
  - (2) Koshi Ikegawa, Shuhei Aoyama, Shogo Tsuchikiri, Takuto Nakamura, Yuki Hashimoto, and Buntarou Shizuki. Investigation of Touch Interfaces Using Layered Urushi Circuit. In Proceedings of the Twelfth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '18), 8 pages, March 2018. ACM.（採録決定済）
- 査読なし国内会議論文
  - (3) 青山周平, 川畑裕也, 駒田悠, 明神優, Zhang Yunze, 橋本悠希, 志築文太郎. 積層漆回路を用いたタッチインタフェースの検討. 第79回全国大会講演論文集, No. 1, pp. 335-336, 2017年3月. 情報処理学会.
  - (4) 青山周平, 志築文太郎. ThumbStick: 片手スマートウォッチ操作のための親指の二次元動作認識システム. インタラクション2018論文集, 6 pages, 2018年3月. 情報処理学会.

## 付録A アンケート用紙

第6章にて示した実験において使用した，アンケート用紙を示す。

### 実験に関するアンケート

文責：青山 周平

1. 年齢, 性別, 利き手, 腕時計をつける手についてお答えください.

年齢：        歳                      性別： ( 男・女 )

利き手：( 左・右 )      腕時計をつける手：( 左・右・両方 )

2. 腕時計を普段つけますか (スマートウォッチを除く)

いつもしている・ときどきしている・つけない

3. 普段使用しているスマートフォンの機種は何ですか?

(例：iPhone4S, docomo Galaxy S II, docomo AQUOS PHONE ZETA SH-01F)

\_\_\_\_\_

4. スマートフォンの利用歴はどのくらいですか?

       年                             カ月

5. スマートウォッチはどちらの腕に取り付けましたか?

左・右

6. スマートウォッチを使用したことはありますか?

普段から使用している・使用したことがある・ない

7. 質問 6. において、普段使用している・使用したことがあると答えた方に質問です。スマートウォッチの利用歴はどのくらいで、使用機種は何ですか?

利用歴：       年                             カ月      使用機種：\_\_\_\_\_

8. 2つの決定方法（握り、タップ）のどちらが使いやすかったですか？

(決定方法)  
(理由)

9. 2つのアルゴリズム（A, B）のどちらが使いやすかったですか？

(認識アルゴリズム)  
(理由)

10. 本システムは使いやすかったですか？

(そう思う) 5・4・3・2・1 (まったく思わない)

理由

11. 本システムについて、改善すべき点、感想をご記入ください。

改善すべき点

その他・感想

アンケートは以上です。ご協力ありがとうございました。

実験者使用欄

腕の太さ ( ) cm、被験者 No. ( )、順序 ( )