

筑波大学 情報学群 情報メディア創成学類

卒業研究論文

持ち方に応じた操作を可能とする
携帯端末向け入力インタフェース

池上 秀

指導教員 田中二郎 志築文太郎 三末和男

2013年1月

概要

高機能携帯端末は一般的に入力手法としてタッチパネルが用いられる。一方で、使用頻度の高い機能にはタッチパネルの GUI 操作よりも素早く、操作が簡単なハードキーが用いられる。高機能携帯端末は多くの機能を備えており機能に応じた持ち方が多様に存在するが、ハードキーは入力位置が固定されるため、端末の持ち方を制限する問題が生じる。

本研究では、ハードキーのように素早く簡単な操作を踏襲し、かつ多様な持ち方に対応する入力インタフェースの実現を目的とする。そのために、我々は携帯端末の側面に持ち方に応じて仮想的なハードキーを配置する手法を提案する。本手法は、端末の側面全体に設置した多数の圧力センサによる圧力分布からパターン認識を用いて持ち方を推定し、仮想ハードキーの配置を行う。また、携帯端末に装着するカバー型入力デバイスを作成し、持ち方の認識や入力精度の評価と仮想ハードキーを用いた携帯アプリケーションの開発を行った。

目次

第1章	はじめに	1
1.1	研究背景	1
1.2	既存の入力インタフェースで対応できない問題	2
1.3	本研究の目的とアプローチ	3
1.4	本論文の構成	3
第2章	関連研究	4
2.1	携帯端末周辺を利用した研究	4
2.2	コンテキスト情報を活用した研究	5
2.3	端末の持ち方に着目した研究	5
第3章	多様な持ち方に対応する入力インタフェース	6
3.1	多様な持ち方に対応する入力インタフェースの実現	6
3.2	仮想ハードキーの設計	7
3.3	仮想ハードキーの利用	7
3.3.1	地図アプリケーションへの利用	8
	利用シナリオ例	10
3.3.2	音楽再生アプリケーションへの利用	10
	利用シナリオ例	10
第4章	実装	12
4.1	開発環境	12
4.2	ハードウェア構成	14
4.2.1	圧力検出部	16
4.3	持ち方認識の実装	16
4.3.1	パターン認識を用いた持ち方認識	16
4.4	仮想ハードキー配置手法の実装	17
4.4.1	持ち方の分類を利用した指の大きな存在範囲推定	17
4.4.2	極大点の算出による指位置の推定と対応付け	18
第5章	評価実験	21
5.1	実験1:持ち方認識に関する実験	21
5.1.1	実験目的	21

5.1.2	実験内容	21
5.1.3	実験結果	21
5.1.4	考察	23
5.2	実験 2: 仮想ハードキー入力に関する実験	24
5.2.1	実験目的	24
5.2.2	実験内容	24
5.2.3	実験結果	25
5.2.4	考察	25
第 6 章	結論と今後の課題	28
	謝辞	29
	参考文献	30

図目次

1.1	携帯端末の様々な持ち方	2
3.1	ハードキーを全く利用することができない持ち方の例	6
3.2	仮想ハードキーによる入力までの流れ	7
3.3	持ち方と仮想ハードキーの割り当て例	8
3.4	地図アプリケーション実行画面	9
3.5	地図アプリケーションの使用場面	9
3.6	音楽再生アプリケーション実行画面	11
3.7	音楽再生アプリケーションの使用場面	11
4.1	携帯端末装着型圧力検出デバイス	12
4.2	システム構成図	13
4.3	ハードウェア構成	15
4.4	持ち方のクラス分類	17
4.5	持ち方例	18
4.6	圧力分布データの例	19
5.1	実験を行った持ち方の種類	23
5.2	実験1用アプリケーション画面	24
5.3	入力精度を測定する仮想ハードキー	25
5.4	実験2用アプリケーション画面	27

第1章 はじめに

1.1 研究背景

計算機端末は電子工学技術の発展に伴い、小型化や高性能化が行われてきた。計算機端末の役割は研究用から事務用、家庭用へと幅を広げ、ついには個人用の計算機として携帯端末が登場した。携帯端末は急速に普及し、2013 年現在、日本においては人口の大多数が所有している¹。特に、2011 年以降、携帯端末市場ではスマートフォンと呼ばれる高機能携帯端末が急速に普及した²。また、携帯端末はさらに高性能化と小型化が進められ、日々便利なものへと進化している。

計算機の小型化は、携帯端末を生み出したが、一方で大きな障壁をもたらした。Olsen 氏はこの障壁について、次のように述べた。

“ If I can fit my entire PC in a cubic inch, how will I interact with it? ”³

「もし私が PC を立体の 1 インチのサイズに収めることができれば、どうやってそれとインタラクションすることができるか？」

この障壁の存在は計算機端末が小さくなったとしても、人間の指や手といったヒューマンファクターは不変であることに起因する。この障壁から、端末が小さくなるほど物理的なインタフェースを多く搭載することは困難になると言える。そのため、小型化された携帯端末のスペースは効率的に利用されることが要求される。

小型化された携帯端末では、効率的な入力インタフェースとしてタッチパネルが一番普及している。タッチパネルは出力領域と入力領域が一致しており、スペースの節約が可能という点で他の入力手法より優れている。また、多くの高機能携帯端末には端末側面にハードキーが設置されている。ハードキーは一般的に使用頻度が高い操作のショートカットが割り当てられるため、使用頻度が高い操作においてはタッチパネルの GUI 操作よりも素早く簡単に操作が行える。以上のように、小型化された携帯端末はスペースを効率的に利用することが要求されるが、そこで利用されるタッチパネルとハードキーは携帯端末に必要な入力インタフェースであると言える。

¹ 社会法人電気通信事業者協会. 事業者別契約数 <http://www.tca.or.jp/database/2010/12/>

² 矢野経済研究所. 携帯電話世界市場に関する調査結果 2011 <http://www.yanoict.com/yzreport/210>

³ Olsen, D. 2007. UIST 2.0 Interviews . Dan Olsen. <http://www.acm.org/uist/archive/uist2.0/DanOlsen.html>

1.2 既存の入力インタフェースで対応できない問題

既存の入力インタフェースで対応できない問題として、様々な持ち方への対応が挙げられる。携帯端末には、右手持ちや左手持ちの他、カメラの持ち方、ゲームの持ち方、動画再生時の持ち方などの多くの持ち方が存在する (図 1.1)。

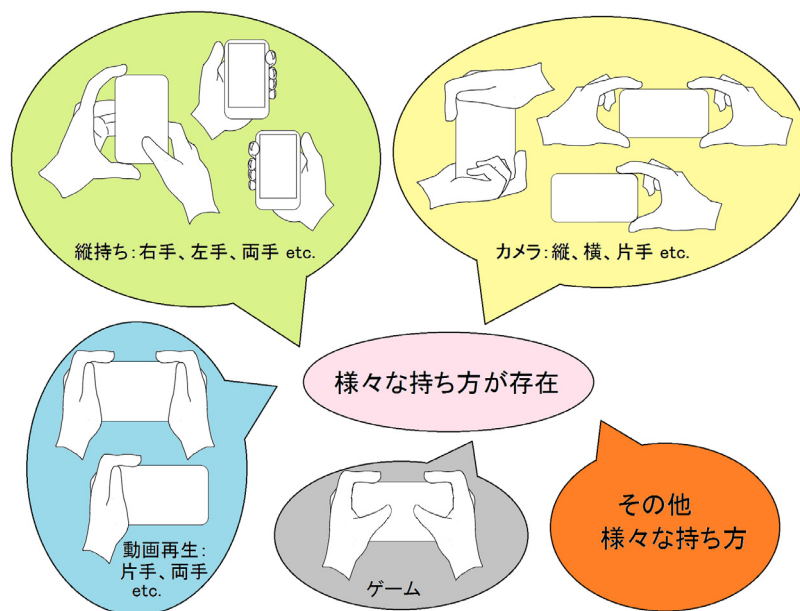


図 1.1: 携帯端末の様々な持ち方

持ち方は操作にかかわるため、携帯端末には持ち方に対応してインタフェースを変化させることが要求される。しかし、ハードキーは入力位置が固定であるため、入力位置の変更が必要となるような使い方をすることができず、持ち方や操作自体の制約になる可能性が高い。一方でタッチパネルは GUI を変化させることで任意の場所に仮想的に入力位置を配置することが可能である。例えば、スマートフォンは内蔵の加速度センサで端末の方向を取得することにより、画面の方向を変化させる機能を持つことが多い。これにより、その方向に合わせて操作の GUI を設定することが可能である。しかし、右手持ちや左手持ち、カメラの持ち方、ゲームの持ち方といった詳細な持ち方を識別することはできないため、指の届きにくい場所に入力位置が配置されてしまうことがある。そのため、どちらのインタフェースにおいても、様々な持ち方に対応することは難しい。

この問題を解決するためには、様々な持ち方に応じて入力位置を変更する入力インタフェースを実現する必要があると言える。また、ハードキーは入力位置を変更できないが、タッチパネルの GUI 操作よりも素早く簡単に操作を行えるという利点があり、それが携帯端末に必要とされていた。そこで、ハードキーの素早く簡単な操作を継承し、かつ多様な持ち方に対応するインタフェースを実現することが必要である。

1.3 本研究の目的とアプローチ

本研究の目的は、ハードキーの素早く簡単な操作を継承し、かつ端末の多様な持ち方に対応する入力インタフェースを実現することである。そこで我々はアプローチとして携帯端末を握り方に着目した。また、それを判別するために握るときの圧力を活用する。携帯端末の側面部分に複数の圧力センサを設置し、携帯端末を握る圧力を取得することで持ち方を認識する。認識された持ち方から指の可動範囲を推定し、仮想的なハードキーを設置することで入力位置を変化させる。また、側面全体に設置した圧力センサを入力操作自体にも活用することで、携帯端末の多様な持ち方に対応して入力を行える入力インタフェースを実現する。

1.4 本論文の構成

本章では、携帯端末における既存の入力インタフェースについての問題点を挙げ、それを解決する手法の提案と研究の目的を述べた。続いて第2章では関連研究について述べる。第3章では多様な持ち方に対応する入力インタフェースについて説明し、第4章ではシステムの実装について詳細に述べる。第5章で評価実験について述べ、最後に第6章で結論と今後の課題を述べる。

第2章 関連研究

本章では携帯端末の入力インタフェースの関連研究を紹介する。本研究では圧力センサを用いて端末の持ち方というコンテキスト情報を得る。またその情報を利用して、携帯端末側面を活用する入力インタフェースを実現する。そのため、携帯端末周辺を活用する研究、コンテキスト情報を活用する研究、端末の持ち方に着目した研究を先行研究とする。

2.1 携帯端末周辺を利用した研究

携帯端末の周辺を利用した入力手法の研究はこれまでに多く行われている。SideSight[1]は側面にIRセンサを設置し、装置の外側の空間を利用してインタラクションを行う。これによってタッチパネルの利用をせずに、効果的にインタラクションを行う方法を示した。この研究では指によるタッチパネルの閉塞問題を解決する入力方法を実現することを目的としている。本研究は、この研究と同じくタッチパネルを使用せず、装置周辺を利用する入力インタフェースを実現するが、様々な持ち方に対応する入力インタフェースであるという点で、この研究と異なる。

携帯端末の周辺を利用した入力手法の研究には入力を拡張する研究が多く存在する。ForceTap[2]は加速度センサの値を解析し、タッチパネルを押す圧力を求めることで、タップ入力に押し込み情報を付加し、タップ入力を拡張する。中島らの研究[3]では携帯端末における入力操作を拡張する入力手法として、タッチパネル外部の四隅をタップする入力手法を提案している。ForceTap[2]と中島らの研究[3]は、端末にあらかじめ備えられているセンサを用いているため、携帯端末のスペースを圧迫せず、入力の拡張を実現している。本研究では端末側面を利用して入力操作を拡張するため、端末周辺を利用した入力拡張という点でこの2つの研究と共通している。しかし、本研究はこの2つの研究と異なり、端末の持ち方というコンテキスト情報を利用した柔軟な入力インタフェースを実現する。

端末を持つ手を利用する入力インタフェース研究も存在する。椎尾ら[4]はタッチパネルの外部にタッチセンサを設置し、タッチパネルの入力操作の補助に持ち手を利用するインタフェースを提案している。このインタフェースでは持ち手を文鎮のように扱うことで、コンテンツのスクロール操作を一時的に編集操作に置き換える。この研究では、持ち手を利用した操作補助を実現しているが、本研究では持ち手から得られる情報をさらに活用し、入力操作を実現している。Pressages[5]は携帯端末の側面に圧力センサを複数取り付け、端末を握る強さを入力として、遠隔地コミュニケーションに利用したものである。この研究ではコミュニケーションの手段として端末を握ったときの圧力情報を利用しているが、本研究では、こ

の圧力情報を端末の入力位置の変更と操作に活用した。

2.2 コンテキスト情報を活用した研究

使用者の動作などのコンテキスト情報を検出し利用することによって入力操作や出力を変更する研究は多く行われている。中村ら [6] や嶋谷ら [7] は加速度センサの値を解析し、携帯端末の使用者の動作を認識することで、入出力を変化させる。本研究では、これらの研究と同じくコンテキスト情報を取得し活用する。しかし本研究では、これらの研究で取得されていない端末の持ち方というコンテキスト情報を取得し、入力インタフェースに活用している。

2.3 端末の持ち方に着目した研究

コンテキスト情報として端末の持ち方を認識し利用する研究も多く存在する。GripSense[8] は携帯端末の加速度センサの値を解析することで、持ち方やタッチパネルへの指の押し込みの強さを認識する。また、この認識結果を入力操作に利用している。HandSense[9] は携帯端末の側面に設置した静電容量センサの値を解析することにより、持ち方を認識する研究である。Graspables[10] は側面に取り付けた複数の静電容量センサから、物体の持ち方を解析して、それをインタフェースとして活用する方法を示している。これらの研究は、いずれも持ち方の認識を行う。GripSense[8] と Graspables[10] は、認識した持ち方を活用する点で本研究と共通している。しかし、これらの研究とは異なり、本研究では持ち方認識によって入力位置を変更できる新たな入力インタフェースを実現している。また、HandSense[9] や Graspables[10] は静電容量センサを用いて持ち方認識を行っているが、本研究では圧力センサを使っているため、強弱情報を扱える。そのため、設置したセンサは認識だけでなく入力に直接活用できる。

また、持ち方を認識することによって GUI を変化させる研究が多く行われている。BiTouch and BiPad[11] は持ち方ごとに扱いやすい UI を適用させるため、タッチパネルに触れる指の位置から端末の持ち方を推定し、タッチパネル上の UI を変化させる。iRotate Grasp[12] は従来の加速度センサを用いた端末の方向認識による間違った画面方向回転の問題を解決するため、端末の側面に設置した複数の光センサの値から携帯端末の持ち方と方向を認識し、体の向きに合わせた画面方向回転を行う。Kee-Eung ら [13] の研究は本体全体にタッチセンサを取り付けて持ち方を認識し、その持ち方に合わせて機能呼び出すなど、タッチパネル上の GUI の変化に利用している。これらの研究はタッチパネル上の GUI を持ち方に合わせて使い易く変化させることが目的であるが、本研究ではタッチパネルではなく端末の側面で様々な持ち方に対応する入力インタフェースを実現する。

第3章 多様な持ち方に対応する入力インタフェース

3.1 多様な持ち方に対応する入力インタフェースの実現

先行研究 [8,9,10,12,13] では、端末側面・背面に取り付けた静電容量センサ・光センサなどからデータを解析することで持ち方の認識を行ってきた。持ち方を認識することによって、タッチパネル上の GUI を変化させることができるため、多様な持ち方に合わせた GUI の提供が可能である。しかし、その一方で携帯端末に設置されているハードキーは入力位置が固定であるため、持ち方を認識したとしてもそれに対応することができない (図 3.1)。使用頻度の高い操作がハードキーに割り当てられているとき、ハードキー操作はタッチパネルの GUI 操作と比べ、素早く簡単であるという利点があるが、この場合はそういった操作ができなくなる。そこでハードキーのように素早く簡単な操作が可能であり、多様な持ち方に対応する新しい入力インタフェースが求められる。



図 3.1: ハードキーを全く利用することができない持ち方の例

そこで、我々はハードキーに似た操作が可能で、かつ多様な持ち方に対応できる新たな入力インタフェースを考案した。この入力インタフェースは入力位置が固定されるハードキーとは異なり、先行研究における GUI のように、持ち方に応じて入力位置を変化させる。また、ハードキーの利点を踏襲するため、ハードキーに似た操作で入力を行う。考案した入力インタフェースは仮想的に入力位置が変化するキーであり、ハードキーに似た操作方法を与えるため、我々はこの入力インタフェースを仮想ハードキーと呼ぶことにする。

3.2 仮想ハードキーの設計

我々は仮想ハードキーを以下のように設計した。

1. 仮想ハードキーは押し込みによって入力を行う。
仮想ハードキーによって、タッチパネルの GUI 操作よりも素早く簡単な操作を実現したため、実際のハードキーと同じように押し込みによる操作ができるように設計した。
2. 仮想ハードキーは指が置かれた位置とその可動範囲に配置する。
様々な持ち方に対応して入力位置を変更することが求められるため、仮想ハードキーは常に指の置かれた位置とその可動範囲に配置されるように設計した。
3. 仮想ハードキーは指の種類に対応して割り当てる。
持ち方を変えたとき、入力位置は指が置かれた位置に設定されるが、そのときに以前の持ち方と違う操作方法になると扱いにくい。そのため、持ち方が変わっても、同じ指を押し込んだときは同じ機能が割り当てられるように、仮想ハードキーは指の種類に対応して割り当てられるように設計した。しかし、片手持ちから両手持ちへと変えるときのように、置かれる指の種類が変わるときには、その持ち方に合わせて仮想ハードキーと指を対応させる。

以上のように、仮想ハードキーは実際のハードキーの素早く簡単な操作を踏襲し、かつ多様な持ち方に対応できる入力インタフェースとして設計した。

3.3 仮想ハードキーの利用

仮想ハードキーによる入力は図 3.2 の流れで行われる。

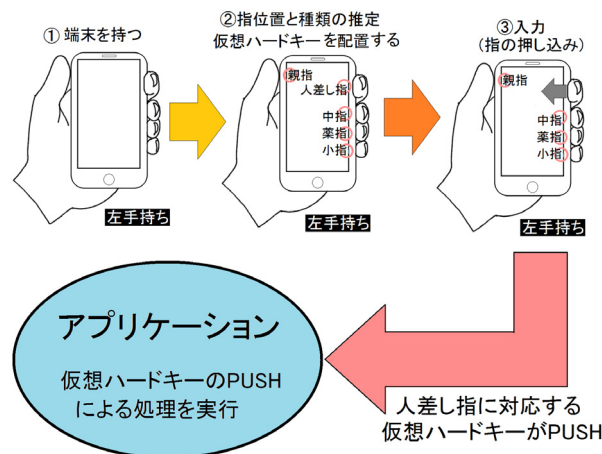


図 3.2: 仮想ハードキーによる入力までの流れ

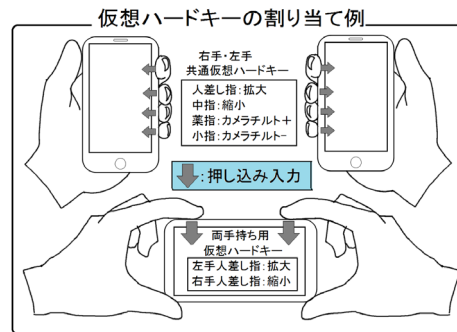


図 3.3: 持ち方と仮想ハードキーの割り当て例

図 3.3 のように仮想ハードキーは各指の種類に対応して割り当てる。携帯端末には様々な持ち方が考えられるが、仮想ハードキーはどのような持ち方であっても使用可能である。また、持ち方が同じでも指位置には差が出るため、仮想ハードキーは指の位置を詳細に推定して配置される。これにより、位置を確認するといった動作をする必要が少なくなるため、フィードバックのない状況での使用が可能である。例えば、右手持ちと左手持ちでは、指の位置が異なるが、どちらの持ち方であっても、同じ種類の指を押し込めば、アプリケーション内で同じ操作が可能である。このため、ハードキーと異なり、持ち方によってできない操作はなくなる。また、通常のハードキーと同じように、長押しや同時押しなどの入力操作も実現可能である。

以下に仮想ハードキーを実装したアプリケーション例を示す。

3.3.1 地図アプリケーションへの利用

入力インタフェースとして仮想ハードキーを実装した地図アプリケーションは、右手持ち、左手持ち、両手持ちの持ち方にそれぞれ対応する。片手操作時には、人差し指に拡大、中指に縮小、薬指・小指にカメラのチルト機能を持つ仮想ハードキーが割り当てられており、両手操作時には、左手人差し指に拡大、右手人差し指に縮小の機能が割り振られている。地図アプリケーションに入力インタフェースとして仮想ハードキーを実装したことにより、拡大・縮小・カメラチルトなどの親指のみでは難しい操作を片手時でも扱えるようになった。そのため、荷物があるときなどの制限された状況下でも、手の持ち代えを行わずに使用することが可能である。また、拡大・縮小・カメラチルトの操作に対して、GUIを呼び出したりジェスチャを用いて操作したりする必要がなくなるため、既存の地図アプリケーションよりも、素早く簡単な操作が可能である。

地図機能の実装には Google Maps Android API v2⁴を利用した。

⁴<https://developers.google.com/maps/documentation/android/?hl=ja>

利用シナリオ例

就職活動中の学生 A 君は、面接予定の会社へ向かっていた。この企業に来るのは今日が初めてだったが、A 君はスマートフォンの地図アプリケーションを駆使して向かおうと思っていた。企業の近くまで来たが、突然の雨に見舞われた。A 君は仕方なく傘を出すことにしたが、傘と鞆で右手は完全に埋まってしまった。A 君は詳細な場所がわからないので地図アプリケーションを使おうとしていたが、片手操作だと拡大・縮小が難しい。鞆を濡れた地面に置くわけにもいかない。そこで、仮想ハードキーを実装した地図アプリケーションを使用した。この入力インタフェースは片手でも拡大・縮小が簡単である。また、A 君は鳥瞰図の地図では位置関係がいまいちわからなかったが、カメラチルト機能を使うことによって建物の形状が良くわかった。カメラチルト機能の利用も GUI やジェスチャを用いて行うより、ずっと簡単である。結果、A 君は濡れることもなく無事企業の住所を確認。面接の時間までになんとか間に合ったのだった。

3.3.2 音楽再生アプリケーションへの利用

入力インタフェースとして仮想ハードキーを実装した音楽再生アプリケーションは、右手持ち、左手持ち、両手持ちの持ち方に対応する。片手操作時には、人差し指に再生・停止機能、中指に早送り機能、薬指に巻き戻し機能を持つ仮想ハードキーが割り当てられ、両手操作時には、両手人差し指同時押しが再生・停止機能、右手人差し指が早送り機能、左手人差し指が巻き戻し機能を持った仮想ハードキーになる。入力インタフェースとして仮想ハードキーを実装したことにより、音楽の再生操作に対して GUI を呼び出す必要がなくなる。そのため、既存の音楽再生アプリケーションよりも素早く簡単な操作が可能である。また、画面を GUI によって占領せず、持ち手の指を押し込むだけで操作が可能であるため、音楽を聴きながら端末で別の作業をすることが可能である。押し込む指によって操作が割り振られるため、入力位置の確認をする必要がないので、ポケット内での操作も簡単に行える。

利用シナリオ例

B 君はスマートフォンを駆使して電車で音楽を聴きながらその日のニュースを調べていた。B 君は、ふと今聞いている音楽が季節にそぐわないと思い、早送りをしたくなった。しかし、今、画面には大量のニュースが開かれている。これを一旦中断して、音楽再生アプリケーションの GUI を呼び出すと今どこを見ていたのかわからなくなる。そこで B 君は仮想ハードキーを実装した音楽アプリケーションで操作をすることにした。これにより、ニュースを開いたまま、音楽を送ることができた。そのあと、B 君の乗っている電車は満員電車となった。この過密な車内で腕を動かしたら人の迷惑になると B 君は感じ、スマートフォンをしぶしぶポケットにしまった。しかし、そんな時、想定外なことに音楽の再生が終わってしまった。電車内は騒音が激しかったため B 君はどうしても音楽が聴きたかった。そこで、B 君は再び、仮想ハードキーを実装した音楽再生アプリケーションで操作を行うことにした。B 君はポケッ

トの中でスマートフォンを握り、人差し指を押し込んだ。そうすると、音楽はまた再生された。結果、B 君は人に迷惑をかけずに楽しむことができたのであった。



図 3.6: 音楽再生アプリケーション実行画面



図 3.7: 音楽再生アプリケーションの使用場面

第4章 実装

我々は仮想ハードキーを実現する入力システムを開発した。本章では、システムの実装について詳細に説明する。

本システムは、ハードウェアとソフトウェアの両方で構成される。仮想ハードキーは持ち方に応じて位置変更を行うため、ユーザによる端末の持ち方を認識する必要がある。そのため、ソフトウェアには“持ち方認識”と“仮想ハードキーの配置”の2つの処理を実装した。また、この2つの処理では携帯端末を握るときの圧力を利用した。理由として、圧力の分布と強弱が持ち方認識の役に立つこと、圧力の強弱が入力に活用できることが挙げられる。また、ハードウェアとして、携帯端末を握るときの圧力を検出する試作デバイスを製作した。

まず、開発環境と試作携帯端末装着型圧力検出デバイス(図4.1)のハードウェア構成について記述し、次に、ソフトウェアにおける“持ち方認識”処理と“仮想ハードキーの配置”処理の実装について詳しく説明する。また、図4.2にシステム全体の構成図を示す。

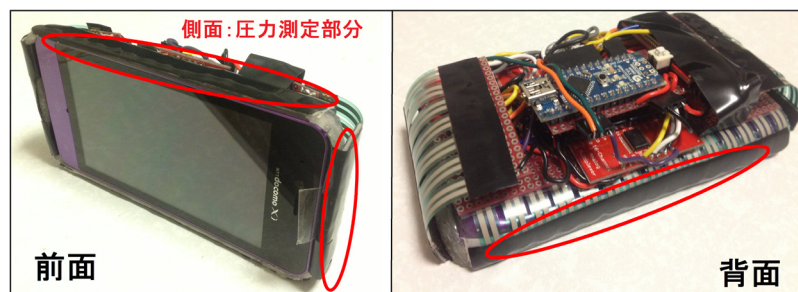


図 4.1: 携帯端末装着型圧力検出デバイス

4.1 開発環境

本研究では、ハードウェアとして試作携帯端末装着型圧力検出デバイス、ソフトウェアとして携帯端末上で持ち方認識と仮想ハードキーの配置の実装を行った。ハードウェアの実装にはマイコンである Arduino nano⁵ を使用し、圧力データの送信は Bluetooth モジュールによ

⁵<http://arduino.cc/en/Main/ArduinoBoardNano>

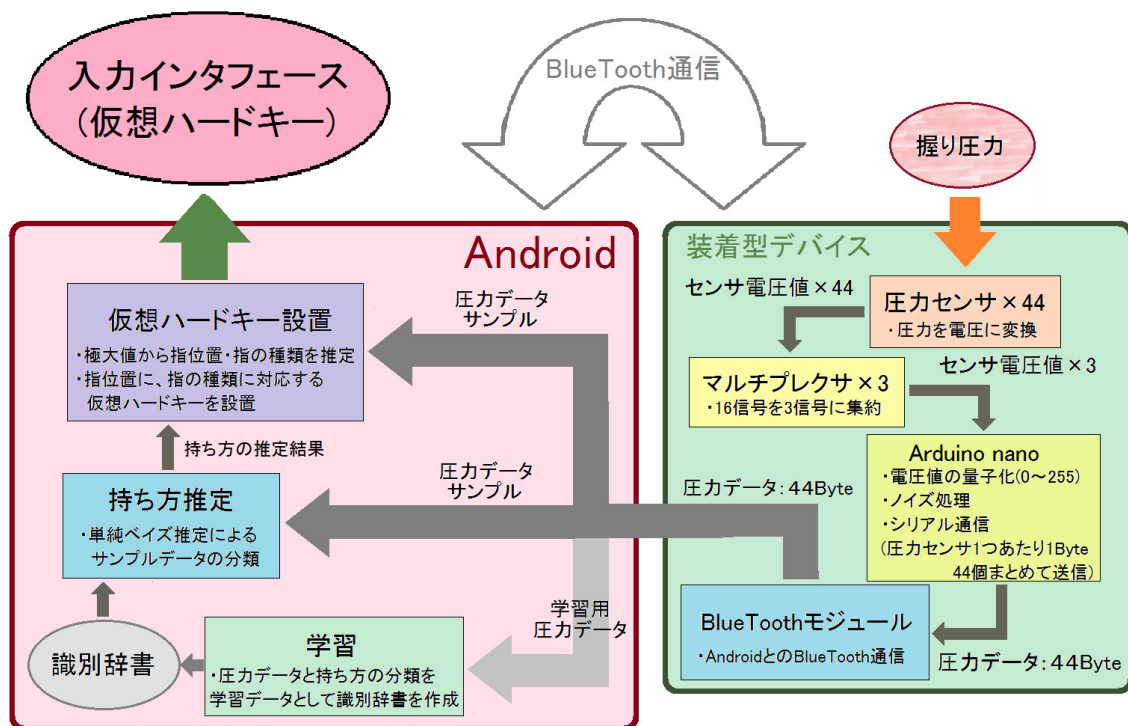


図 4.2: システム構成図

る Bluetooth シリアル通信を利用している。Arduino の制御プログラムは Arduino1.02⁶を開発環境として開発を行った。また携帯端末は Android 端末⁷を使用し、端末上のソフトウェアは Google の AndroidSDK⁸を利用した Java 言語を用いて開発を行った。開発環境として OS は Windows7 Professional32bit 版⁹を利用し、IDE は Eclipse4.2Juno¹⁰を使用している。

4.2 ハードウェア構成

携帯端末は圧力値を取得できないため、本研究では圧力センサを複数取り付けたカバーを携帯端末に装着する。我々は携帯端末に装着する試作携帯端末装着型圧力検出デバイスを制作した。このデバイスは、側面の圧力センサからデータを集め、マイコンを通して Bluetooth 通信で携帯端末へ圧力分布データを送信する。ハードウェアの詳細構成は以下のようになる。

圧力センサ

インターリンク株式会社の薄型圧力センサ FSR400 (Force Sensing Resistors400) ¹¹を 44 個 (左右 15 個ずつ、上下 7 個ずつ)、カバー側面部に間隔を開けずに設置している。圧力センサは、長さ 38.1mm、幅 7.6mm、厚さが 1.25mm、センシング部の直径が 6.4mm であり、圧力感知範囲は 10g~10kg(0.1N~100N) である。圧力の値は抵抗値の変化であらわされるが、抵抗を直列に接続することで電圧の変化を出力する。

抵抗アレイ

8 素子 10k Ω の SIL 型抵抗アレイを使用している。圧力の値を電圧信号としてマイコンで読み取るために使用する。抵抗 1 素子と圧力センサを直列に接続することで、抵抗値の比により電圧比が生まれるため、圧力値の電圧信号が出力される。

マルチプレクサ

マイコンのポート数が少ないため、マルチプレクサによりスイッチを行い 44 個の圧力値データをまとめて入力する。44 個の圧力センサを利用しているため、16 チャンネル入力のマルチプレクサ CD74HC4067¹²を 3 個用いて 3 出力にまとめている。

マイコン

携帯端末の背面に設置するためサイズの小さい Arduino nano を利用した。

Bluetooth シェル

Bluetooth シェルとして Bluetooth Mate Gold¹³を使い、圧力センサから取得した 44 の圧力値データをマイコンから携帯端末へとシリアル通信で送信する。

⁶<http://arduino.cc/en/Main/Software>

⁷<http://www.android.com/>

⁸<http://developer.android.com/sdk/index.html>

⁹<http://windows.microsoft.com/ja-JP/windows7/products/home>

¹⁰<http://www.eclipse.org/>

¹¹<http://www.interlinkelectronics.com/jp/FSR400.php>

¹²<http://www.tij.co.jp/product/jp/CD74HC4067>

¹³<https://www.sparkfun.com/products/9358>

携帯端末

LG エレクトロニクス製の Android 端末である Optimus it L-05D¹⁴を使用した。OS のバージョンは Android4.1 である。

図 4.3 にハードウェアの構成図を示す。

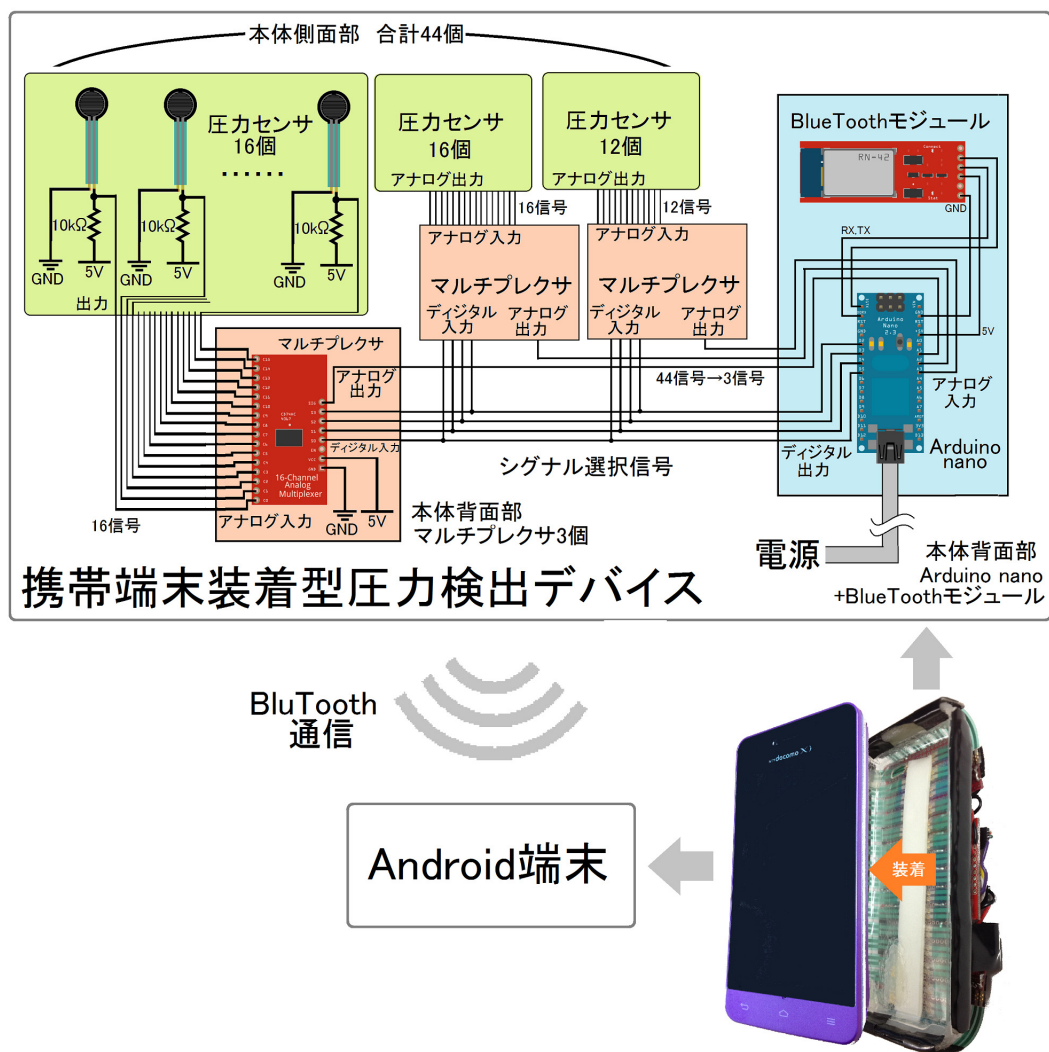


図 4.3: ハードウェア構成

¹⁴<http://www.lg.com/jp/mobile-phone/lg-Optimus-it-L-05D>

4.2.1 圧力検出部

試作携帯端末装着型圧力検出デバイスは、携帯端末を握る圧力分布のデータを携帯端末へ送信している。Arduino の制御プログラムでは、圧力センサと抵抗の抵抗比から生じる電圧値を 0~255 までの値で量子化している。また、44 個の圧力センサの圧力データをまとめて取得するため、16 チャンネルのマルチプレクサにシグナル選択信号を送ることでスイッチを行い、3 つの出力に集約している。スイッチの時間間隔は 1ms であるため、全ての圧力センサから圧力データを得るためには 16ms の時間がかかる。制御プログラム内では圧力データ 1 つあたり 1 Byte のデータに変換しており、1 回のデータ通信で 44Byte のデータを送信している。また、データはセンサの実際の並び順にソートされ送信される。圧力センサは非常に小さい圧力も検出するため、ノイズ対策として 0~30 の値をカットして送信している。

4.3 持ち方認識の実装

ソフトウェアは、“持ち方の認識”と“仮想ハードキーの配置”の処理で構成されている。本セクションでは“仮想ハードキーの配置”の前に行う“持ち方の認識”処理の実装について詳細に説明する。

4.3.1 パターン認識を用いた持ち方認識

本研究では持ち方を認識する手段として、パターン認識を用いた。パターン認識は実世界の情報をパターンとして受け取り、それを複数の概念(クラス)のうちの一つに分類する手法である。通常、パターン認識は入力された信号について、前処理、特徴抽出、識別の 3 つの処理を行い、パターンの分類を行う。また、識別時には識別辞書が必要であるため、事前に多数のパターン信号によって“学習”を行い、識別辞書を作成する必要がある。

持ち方認識は、パターン信号として携帯端末を握るときの圧力分布データを入力し、そのパターンがどの持ち方の圧力分布であるかを分類している。パターン認識は未知データに対して高い精度で分類を行うことができるため、測定の度に値が変わる圧力分布データの分類に適している。

持ち方認識では、まず圧力分布データを試作携帯端末装着型圧力検出デバイスから取得する。この圧力分布データは、BlueTooth 通信によって携帯端末 (Android 端末) へと送信される。そのあと、前処理として圧力分布データを端末上で 0~255 の整数値に変換する。次に行う特徴抽出処理は、識別精度を上げるため、無駄な情報の多いデータから認識に必要な特徴を抽出する処理である。しかし、試作デバイスに設置された圧力センサの数が少なく、圧力分布データには無駄な情報が含まれにくいと考えられるため、特徴抽出処理を行わず圧力分布データそのものを識別と識別辞書作成時の学習に利用した。

前処理が完了した圧力分布データは、単純ベイズ分類器と呼ばれるパターン認識手法を利用して識別する。実装には Android 用ライブラリ OpenCV2.3.1¹⁵を利用した。学習には 18 のクラス (図 4.4) を割り当てた圧力分布データを利用した。また、識別率の低いクラスに対して圧力分布データを追加していく方法で学習を行い、合計で 700 個のデータを学習に利用した。

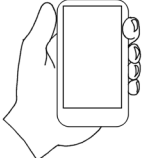
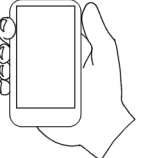
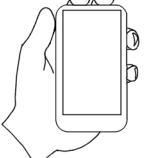
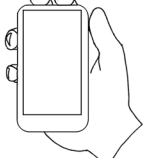
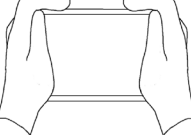
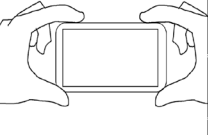
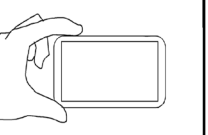
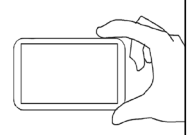
			
クラス:1~4 同じ持ち方でも 指の位置によって クラスを分けた	クラス:5~8 同じ持ち方でも 指の位置によって クラスを分けた	クラス:9	クラス:14
			
クラス:11,13 上下逆の持ち方 に応じてクラス分け	クラス:10,12 上下逆の持ち方 に応じてクラス分け	クラス:15,18 上下逆の持ち方 に応じてクラス分け	クラス:16,17 上下逆の持ち方 に応じてクラス分け

図 4.4: 持ち方のクラス分類

4.4 仮想ハードキー配置手法の実装

“仮想ハードキーの配置”では、“持ち方認識”によって得られた現在の持ち方の分類から指の存在範囲を推定した後に、圧力分布データを解析して指位置を求め、仮想的なハードキーを配置する。圧力センサは側面を覆うように複数取り付けられているため、任意の場所に仮想ハードキーを配置することが可能である。また、仮想ハードキーを設置した位置にある圧力センサへの押し込みを入力として認識する。本セクションでは、仮想ハードキーの配置手法の実装について詳細に説明する。

4.4.1 持ち方の分類を利用した指の大まかな存在範囲推定

本システムでは、仮想ハードキーの配置を行うときに指の位置を推定している。指の位置は、まず持ち方認識から得られた現在の持ち方の分類から大まかに範囲を推定し、その後に圧力分布データから詳細な位置を推定している。

¹⁵<http://opencv.org/>

最初に行う大まかな推定では、指の詳細な位置ではなく、どの範囲に指が存在するかを推定する。例えば、図 4.5 のような持ち方であると識別されれば、人差し指と中指は端末の上辺、親指は端末の左辺、薬指と中指は端末の右辺と推定できる。このように持ち方の分類に対して指の存在する範囲をあらかじめ登録しておくことで、大まかに指の存在範囲を推定する。

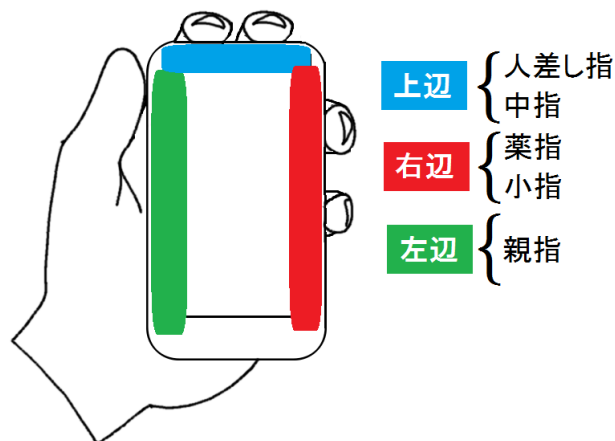


図 4.5: 持ち方例

詳細な指位置の推定は圧力分布データを活用するが、指の大まかな存在範囲を利用することで、指の種類の対応付けと圧力分布データから位置推定に必要なデータのみの取り出しが可能である。例えば、持ち方によっては端末の側面に手のひらが接触することがあるが、そこから検出される圧力値が指によるものであるかそうでないかの判断は難しい。しかし、指の存在する範囲の圧力分布データだけを取り出すことで詳細な位置の推定を行うことが可能である。また、元の圧力分布データからは指がどのような位置関係にあるか直接に判断できないが、指の存在する範囲を利用すれば、指の種類の対応付けも可能である。

そのため我々は、各持ち方に対して、各指の存在する大まかな範囲を登録しておき、その範囲の圧力分布データのみを取り出して次の処理に渡した。また、次の詳細な指位置推定処理では、その位置関係から指の種類を対応させた。

4.4.2 極大点の算出による指位置の推定と対応付け

指の存在する範囲の圧力分布データを得たら、データを解析することで詳細な指位置を推定する。片手持ち時における人差し指、中指、薬指、小指の存在範囲の圧力分布データの一例を図 4.6 に示す。

図 4.6 からわかるように、指位置の圧力値は圧力分布における極大点である。このため、詳細な指の位置は圧力分布の極大点を求めることで推定する。極大点はその場所での圧力値と両隣圧力センサの圧力値を比較することで判断される。このとき、両隣よりも圧力値が大きければ、その場所は極大点である。

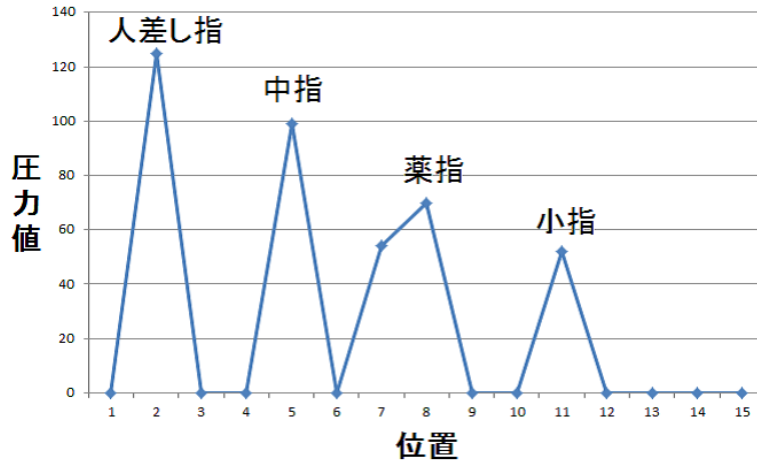


図 4.6: 圧力分布データの例

入力された圧力値のデータが N 個のとき、圧力分布データを $P(0), P(1) \cdots P(N-1)$ と表すと、極大点の集合 X は以下の式 (4.1) によって求めることが可能である。式 (4.1) では、 $g(x)$ は右隣の圧力センサの圧力値と比較を行う。 $g(x)$ の値は圧力値が右隣よりも大きければ真、小さければ偽となる。また、 $h(x)$ は左隣と比較する式であり、その値は圧力値が左隣よりも大きければ真、小さければ偽となる。このとき、 $g(x)$ と $h(x)$ の論理積を $f(x)$ の値とする。極大点は $f(x)$ の値が真となるような x である。また、 x は圧力センサに割り振った番号であるため、データ数が N 個であった場合は $0 \sim N-1$ までの整数値となる。 x が 0 である場合は、左隣が存在しないため、 $f(x)$ は $g(x)$ の値となり、 x が $N-1$ である場合は、右隣が存在しないため、 $f(x)$ は $h(x)$ の値となる。

$$X = \{x | f(x) = true\} \quad (4.1)$$

$$f(x) = \begin{cases} g(x) & (x = 0) \\ g(x) \wedge h(x) & (x \neq 0 \wedge x \neq N - 1) \\ h(x) & (x = N - 1) \end{cases}$$

$$g(x) = \begin{cases} true & : P(x) > P(x+1) \\ false & : P(x) \leq P(x+1) \end{cases} \quad h(x) = \begin{cases} true & : P(x-1) < P(x) \\ false & : P(x-1) \geq P(x) \end{cases}$$

$$x \in \text{整数 } \mathbb{Z}, \quad 0 \leq x \leq N - 1$$

図 4.6 の圧力分布では極大点が 4 点検出されれば、全ての指が置かれていることになる。全ての指が置かれているときには、位置関係から指の種類がわかるため、指の位置と指の種類の対応付けを行う。

仮想ハードキーは指の可動範囲に配置するため、指位置とその隣の圧力センサのある場所に配置する。また、仮想ハードキーには指の種類に対応して番号を振り、同じ番号の仮想ハードキーには同じ機能が割り当てる。

実際のハードキーと同じような操作で扱えるようにするため、仮想ハードキーに対する入力はその位置にある圧力センサを押し込むことによってなされる。このとき、圧力データの値が設定した閾値を超えた時に入力と判断し、その仮想ハードキーが押されたことをシステムに伝える。

第5章 評価実験

我々は作成したシステムに関して評価実験を行った。まず、実装した持ち方認識の精度と認識所要時間を計測する実験を行った。次に仮想ハードキーによる入力精度を計測する実験を行い、各実験について考察を行った。また、認識の際に使用する識別辞書は、筆者が入力を行った合計 700 個の圧力分布データから学習し、作成した。

5.1 実験 1:持ち方認識に関する実験

5.1.1 実験目的

本実験の目的は、持ち方の認識がどの程度の性能であるか調べることである。開発したシステムでは、仮想ハードキーを割り当てる前に持ち方の認識を行うことで、指の大まかな位置を推定する。そのため、持ち方認識の性能はシステム全体の性能にかかわる。そこで、本実験では、持ち方認識の性能を調べるために持ち方認識の精度と認識の所要時間を評価する実験を行った。

5.1.2 実験内容

実験は各 8 種類の持ち方 (図 5.1) をするように被験者に指示を出し、その持ち方で端末を握ってもらい、10 秒間の認識を行う (図 5.2)。被験者は、指示が与えられるごとに持ち方を変え、8 秒の準備時間が経過した後に計測を開始する。指示した 8 種類の持ち方は片手持ちが 6 種類で、両手持ちが 2 種類である。片手持ちに関しては右手・左手それぞれの持ち方について実験を行った。実験では持ち方を正しく認識をするまでに経過した時間と握り方が正しく認識された時間を測定した。持ち方認識の精度はこのときの認識の正答率とし、正答率は、 10 秒中 正しく握り方が認識された時間 $\div$ 測定時間 で求めた。なお、実験は 1 試行で 8 種類の持ち方を各 10 秒ずつ認識する。また、被験者は大学生・大学院生男女 5 人であり、1 人当たり 10 回の試行を行った。

5.1.3 実験結果

持ち方認識の平均正答率を被験者別に示し、本実験全体の結果をまとめた (表 5.1)。また、表 5.2 は本実験全体で各持ち方がどの持ち方に認識されたかを示す。

正答率と同時に測定した認識成功までの平均所要時間を表 5.3 に示す。

表 5.1: 実験 1:被験者別の持ち方認識の正答率

持ち方の種類	1	2	3	4	5	6	7	8	全体
被験者 A	70%	41%	56%	70%	28%	44%	50%	71%	54%
被験者 B	82%	51%	99%	60%	13%	12%	67%	95%	60%
被験者 C	82%	81%	90%	84%	85%	91%	100%	92%	88%
被験者 D	59%	79%	66%	58%	53%	96%	91%	84%	73%
被験者 E	94%	96%	100%	93%	79%	97%	100%	100%	95%
平均	78%	70%	82%	73%	52%	68%	82%	88%	74%

表 5.2: 実験 1:各持ち方の認識結果

		認識された持ち方の種類							
		1	2	3	4	5	6	7	8
指示された 持ち方の種類	1	78%	4%	10%	0%	3%	0%	2%	1%
	2	6%	70%	6%	6%	6%	1%	2%	3%
	3	4%	1%	82%	1%	9%	0%	1%	1%
	4	6%	3%	3%	73%	8%	0%	1%	5%
	5	1%	4%	11%	1%	52%	23%	4%	0%
	6	1%	1%	1%	0%	21%	68%	8%	1%
	7	1%	1%	6%	1%	3%	2%	82%	5%
	8	1%	0%	2%	0%	3%	0%	5%	88%

表 5.3: 実験 1:認識成功までの平均所要時間

持ち方の種類	1	2	3	4	5	6	7	8	全体
被験者 A	1.4 秒	0.8 秒	2.0 秒	0.7 秒	2.0 秒	0.9 秒	2.4 秒	1.5 秒	1.5 秒
被験者 B	1.0 秒	0.5 秒	0.4 秒	2.4 秒	1.5 秒	0.3 秒	2.2 秒	0.5 秒	1.1 秒
被験者 C	0.6 秒	1.0 秒	0.6 秒	0.6 秒	1.1 秒	0.6	0.4 秒	0.5 秒	0.7 秒
被験者 D	2.4 秒	0.3 秒	2.5 秒	1.4 秒	3.6 秒	0.5 秒	0.4 秒	1.1 秒	1.5 秒
被験者 E	0.8 秒	0.6 秒	0.3 秒	1.1 秒	1.6 秒	0.4 秒	0.4 秒	0.3 秒	0.7 秒
平均	1.3 秒	0.6 秒	1.2 秒	1.2 秒	2.0 秒	0.5 秒	1.1 秒	0.8 秒	1.1 秒

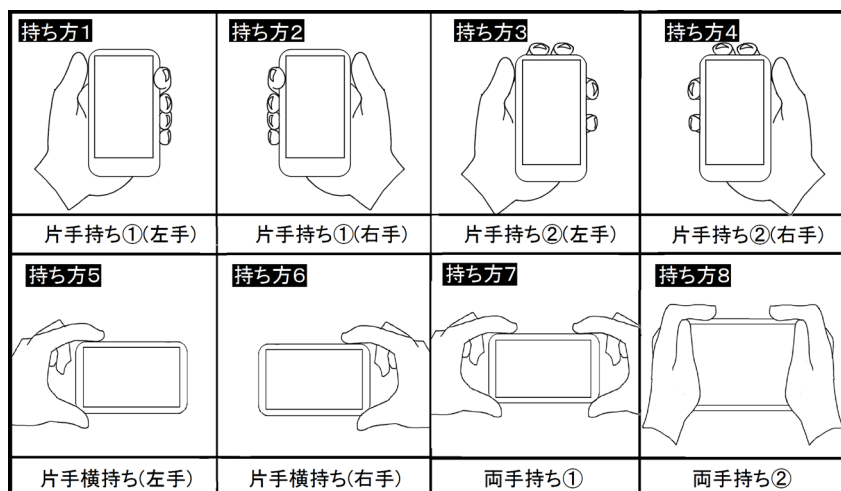


図 5.1: 実験を行った持ち方の種類

5.1.4 考察

本実験全体の認識の平均正答率は 70% を超える結果となった。また、認識率の高い持ち方 8 は約 90% の認識正答率を持つ。しかし一方で、認識正答率が低い持ち方 5 では 50% 程度となっている。この持ち方は片手で横に持った時の持ち方だが、逆の手で同じように持った場合の持ち方 6 も比較的に認識正答率が低い。さらに、表 5.2 をみると、この 2 つの持ち方は互いに誤認識している確率が高いことがわかる。この理由は、片手横持ちが共通した部分を持つことにありとえられる。例えば、端末左端を左手で持ち、右手で右端を持った場合は握る場所が異なるが、端末を上下逆にしたときは、左端を右手で持ち、右端を左手で持つことになる。このようなとき、システムは圧力値の特徴によりどちらの手で握っているかを判断するが、実際には様々な力の入れ方があるため誤認識が多くなったと考えられる。問題を改善するためには、識別辞書を作成する学習データを現在よりも多くすることが必要であると考えられる。

認識正答率は被験者によって大きな差が生じた。被験者 A と被験者 B は平均して 50~60% 程度の低い認識正答率である。特に被験者 B の持ち方 5 と持ち方 6 は 10% 強程度の認識率であった。一方、被験者 C、被験者 D、被験者 E(尚、被験者 E は筆者自身である) は比較的、良い認識正答率を持つと言える。この理由は、手の大きさや持ち方の個人差にあると考えられる。特に、手の大きさによっては握る場所が大きく違うため、筆者の入力した学習データによる識別が難しくなってしまったと考える。学習に使う特徴データは各データが独立していることを前提としているため、今回のように圧力データをそのまま特徴データとして利用した場合、指の位置がずれただけでも認識正答率が大きく下がってしまうようである。また、学習や認識に用いた圧力分布データは特殊な特徴抽出を行わなかったが、そのために握り方の特徴を上手くとらえた識別ができなかったと考える。この問題を改善するためには、新たに特徴抽出の方法を考案することや複数人の学習データを得ることが必要と思われる。



図 5.2: 実験 1 用アプリケーション画面

認識成功までの平均所要時間は本実験全体で 1.1 秒であった。この結果から、握った直後に操作をするようなことがない限りは十分に短い認識所要時間であると言える。また、認識所要時間が短いものであれば、0.3 秒程度で認識される。しかし、所要時間が長くなってしまった例では被験者 D の持ち方 5 がある。この場合、約 4 秒弱異なった持ち方の認識がされているため、誤操作の原因となる。このような例の改善には、認識正答率を上げ、認識のブレが生じないようにする必要があると言える。

5.2 実験 2: 仮想ハードキーに関する実験

5.2.1 実験目的

本実験の目的は、本システムによる入力为正しく行われているかどうかを調べることである。本システムでは、仮想ハードキーの配置された場所を押し込むことによって入力を行うが、仮想ハードキーの配置は持ち方によって変更される。仮想ハードキーは持ち手の指のある場所に配置されるため、ユーザが入力位置に指を合わせる必要がなく、その場で指を押し込むだけで入力が可能である。そのため、我々は本システムによる入力が正しく行われているかどうかを評価するため、仮想ハードキーによる入力精度を測定する実験を行った。

5.2.2 実験内容

この実験では仮想ハードキーによる入力の精度を測定する。仮想ハードキーは指の種類に対応して割り振られているため、その場での指の押し込みによって入力を行う。実験では被験者に持ち方と押し込む指を指示し、押し込まれた時の入力信号が正しいかどうかを判断す

る(図 5.4)。このとき、指示された指による入力であると判断されれば正答とし、異なる指による入力であれば誤答とする。また、各指ごとに 10 秒の制限時間が与えられており、入力なしにその時間が経過してしまった場合は、入力ができなかったとして誤答とする。入力精度は 正答数/(正答数+誤答数) で求める。実験では 6 種類の持ち方に対して配置される全 24 種類の仮想ハードキー(図 5.3)について入力精度を求めた。なお、実験者は大学生・大学院生男女 3 人であり、1 種類の仮想ハードキーにつき 1 人当たり 40 回の入力を行った。

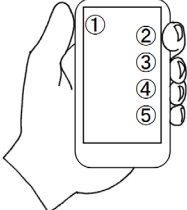
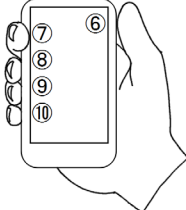
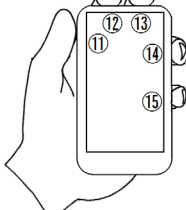
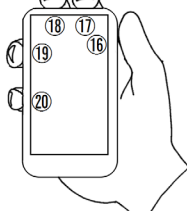
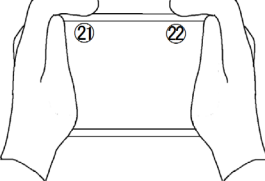
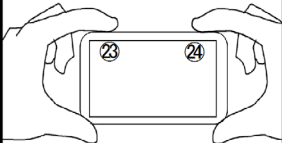
持ち方1 	持ち方2 	持ち方3 
片手持ち①(左手) 仮想ハードキー5種	片手持ち①(右手) 仮想ハードキー5種	片手持ち②(左手) 仮想ハードキー5種
持ち方4 	持ち方5 	持ち方6 
片手持ち②(右手) 仮想ハードキー5種	両手持ち① 仮想ハードキー2種	両手持ち② 仮想ハードキー2種

図 5.3: 入力精度を測定する仮想ハードキー

5.2.3 実験結果

各持ち方における仮想ハードキーによる入力の精度を被験者別に示し、本実験全体の結果をまとめた(表 5.4)。

5.2.4 考察

本実験全体の平均入力精度は 70% を超える結果となった。また、入力精度の高い仮想ハードキーは約 90% の程度の入力精度を持つ。しかし、持ち方や指の違いによって入力精度は大きく左右される結果となった。持ち方別にみると持ち方 3、持ち方 4 が全体的に入力精度が低い。この理由は、上辺と左辺か右辺を使う持ち方にあると考えられる。本研究で作成したデ

表 5.4: 実験 2: 仮想ハードキーの入力精度

持ち方の種類	仮想ハードキー	被験者 F	被験者 G	被験者 H	全体
1	① (親指)	88%	75%	80%	81%
	② (人差し指)	93%	95%	85%	91%
	③ (中指)	80%	93%	65%	79%
	④ (薬指)	80%	93%	60%	78%
	⑤ (小指)	90%	83%	60%	78%
2	⑥ (親指)	85%	78%	90%	84%
	⑦ (人差し指)	88%	93%	98%	93%
	⑧ (中指)	90%	78%	83%	83%
	⑨ (薬指)	80%	90%	73%	81%
	⑩ (小指)	43%	43%	70%	52%
3	⑪ (親指)	75%	68%	63%	68%
	⑫ (人差し指)	73%	55%	63%	63%
	⑬ (中指)	53%	50%	45%	49%
	⑭ (薬指)	68%	68%	48%	61%
	⑮ (小指)	78%	78%	53%	69%
4	⑯ (親指)	73%	78%	75%	75%
	⑰ (人差し指)	68%	75%	85%	76%
	⑱ (中指)	65%	68%	60%	64%
	⑲ (薬指)	63%	58%	60%	60%
	⑳ (小指)	63%	55%	28%	48%
5	㉑ (左手人差し指)	78%	89%	62%	76%
	㉒ (右手人差し指)	87%	93%	73%	84%
6	㉓ (左手人差し指)	87%	99%	60%	82%
	㉔ (右手人差し指)	92%	100%	81%	91%
平均		76%	77%	67%	74%



図 5.4: 実験 2 用アプリケーション画面

バイスは試作機であるため、構造が頑強なものではなく、ノイズが乗りやすい。持ち方によっては意図せず押されてしまう部分が存在するが、持ち方3、持ち方4は比較的指を置いている範囲が広く、ノイズのあるデータが渡されてしまうことがある。それによって、仮想ハードキーの配置場所が狂い、入力ミスが発生すると考えられる。この問題を改善するには、デバイスの改善とノイズカットの徹底が必要である。また、持ち方3、持ち方4の入力精度が低い理由のひとつに指への力の入れにくさが考えられる。このような場合は、圧力の入力閾値を下げる対策をする必要があると考える。

入力精度に差をもたらす理由として、仮想ハードキーの種類の差がある。例えば人差し指と小指の入力精度の差は明らかである。また、入力範囲を広く設定してある親指も入力精度が高いとは言えない。これらの差の理由は、力の入れ易さにあると考えられる。実験を行ったとき、小指や薬指は明らかに力を入れにくかった。また、親指を押し込もうと力を入れると、対面にある指に力が入り押し込まれてしまうという問題もあった。このような問題に対する改善策としては、まず指ごとに入力の閾値を変えることが挙げられる。力の入りにくい指は入力の閾値を下げる。それにより、その指は軽く押し込めば良いため、入力精度は向上するはずである。また、なるべく押しにくい指に単体で機能を割り当てないことも重要である。このような指を使うときは、他の指でサポートをするような形で機能を与えた方が使いやすいと考える。

第6章 結論と今後の課題

本研究では多様な持ち方に対応する携帯端末の新たな入力インタフェースとして、持ち方を認識し、携帯端末の側面に仮想的なハードキーを設置する手法を用いた入力インタフェースを提案し、それを実現する入力システムを開発した。本システムでは、複数の圧力センサを携帯端末の側面に取り付け、携帯端末を握る力の圧力分布データからパターン認識を行い持ち方を認識する。また、持ち方から指の可動範囲を推定し、仮想的なハードキーを設置することで、通常ハードキーを用いることができない持ち方でも入力を制限されない入力インタフェースを実現した。本システムを用いることで、タッチパネル GUI よりも素早く簡単な操作を様々な持ち方で行うことが可能となる。

さらに我々は持ち方の認識精度と認識所要時間を評価する実験と、仮想ハードキーによる入力の精度を評価する実験を行った。最初の実験では平均 74% の持ち方認識正答率と、認識成功までの平均所要時間が 1.1 秒という結果が得られ、次の実験では仮想ハードキーによる平均入力精度が 74% という結果が得られた。

本システムでは高い持ち方認識の精度が要求されるため、今後の課題としては、まず認識正答率を上げることが挙げられる。そのため認識正答率の低い持ち方に対してより多くのパターンを学習をするとともに、様々な人の持ち方を学習することが重要である。また、認識正答率を上げるためには、学習データの特徴抽出を工夫することが必要である。次の課題としては、仮想ハードキーの入力精度の向上と使いやすさの追求が挙げられる。入力精度の向上は、圧力の入力閾値の調整によって実現できると考えている。使いやすさの追求では、様々なアプリケーションを作成し、実際に使ってもらいフィードバックすることが重要である。

今回の研究では仮想ハードキーは押し込み操作のみであったが、今後ソフトウェアに様々な機能を実装することで、スライドやタップといった操作や、握ることではしか実現できないような特殊な機能を持たせ、多様な利用シーンを考えていきたい。また、今回は入力位置を変更させるのみであったが、今後の研究では認識した持ち方の情報を活用して、操作自体を変更させるような機能を追加していきたい。

謝辞

本論文を執筆するにあたり、指導教員である田中二郎教授をはじめ、高橋伸准教授、三末和男准教授、ならびに志築文太郎准教授には、ゼミやミーティングを通して大変貴重なご意見やアドバイスをいただきました。心より感謝申し上げます。また、インタラクティブプログラミング研究室の皆様には多くのご意見やご指摘を頂きました。特に、ユビキタスチームの皆様には研究生生活を通して数多くのご意見やアドバイスをいただきました。この場を借りて御礼申し上げます。最後に経済面で私を支えてくれた両親や、精神面で支えてくれた全ての友人に心より感謝を申し上げます。本当にありがとうございました。

参考文献

- [1] Alex Butler, Shahram Izadi, Steve Hodges: SideSight: Multi-“ touch ” Interaction Around Small Devices, UIST 08, Pages 201-204 , (2008).
- [2] Seongkook Heo, Geehyuk Lee: Forcetap: extending the input vocabulary of mobile touch screens by adding tap gestures, MobileHCI 2011, Pages 113-122, (2011).
- [3] 中島 崇之, 三浦 元喜: フレームタップ操作によるタブレット入力拡張方式の検討, 情報処理学会 インタラクション 2012, Pages 361-366, (2012).
- [4] 椎尾 一郎, 辻田 眸: 文鎮メタファを利用した小型情報機器向けインタフェース, 情報処理学会論文誌, Vol.48, No.3, pp.1221-1228, (2007).
- [5] Eve Hoggan, Craig Stewart, Laura Haverinen, Giulio Jacucci, Vuokko Lantz: Pressages: augmenting phone calls with non-verbal messages, UIST 2012, Pages 555-562, (2012).
- [6] 中村 友宣, 小川 剛史, 清川 清, 竹村 治雄: 連続的なウェアラブル学習システムのための利用者コンテキスト認識機構の実装と評価, 電子情報通信学会技術研究報告. MVE, マルチメディア・仮想環境基礎, vol.106, no.396, Pages.17-22, (2006).
- [7] 嶋谷 健太郎, 間下 以大, 宮本 大樹, 岩田 麻佑, 原 隆浩, 清川 清, 竹村 治雄, 西尾 章治郎: スマートフォンを用いたコンテンツ検索支援のための動作認識, 電子情報通信学会技術研究報告. PRMU, パターン認識・メディア理解, Vol. 111, No. 430., Pages 47-52, (2012).
- [8] Mayank Goel, Jacob Wobbrock, Shwetak Patel: GripSense:using built-in sensors to detect hand posture and pressure on commodity mobile phones, UIST 2012, Pages 545-554, (2012).
- [9] Raphael Wimmer, Sebastian Boring: HandSense: discriminating different ways of grasping and holding a tangible user interface, TEI 2009, Pages 359-362, (2009).
- [10] Brandon T.Taylor, V.Michael Bove, Jr.: Graspables: grasp-recognition as a user interface, CHI 2009, Pages 917-926, (2009).
- [11] Lung-Pan Cheng, Fang-I Hsiao, Yen-Ting Liu, Mike Y. Chen: iRotate Grasp: Automatic Screen Rotation based on Grasp of Mobile Devices, UIST Adjunct Proceedings 2012, Pages 15-16, (2012).

- [12] Julie Wagner, Stephane Huot, Wendy E. Mackay: BiTouch and BiPad: Designing Bimanual Interaction for Hand-held Tablets, CHI 2012, Pages 2317-2326, (2012).
- [13] Kee-Eung Kim, Wook Chang, Sung-Jung Cho, Junghyun Shim, Hyunjeong Lee, Joonah Park, Youngbeom Lee, and Sangryong Kim: Hand Grip Pattern Recognition for Mobile User Interfaces, IAAI2006, Pages 1789-1794, (2006).