

## 修 士 論 文 の 和 文 要 旨

研究科・専攻	大学院 情報理工学 研究科 情報・通信工学 専攻 博士前期課程		
氏 名	栗原 諒	学籍番号	1131041
論 文 題 目	移動中のスマートフォンを用いたウェブ閲覧における 手ぶれ補正システムの提案と評価		
<p style="margin: 0;"><b>要 旨</b></p> <p style="margin: 0;">本研究では、小サイズのモバイル端末を、歩行中や電車・バスに乗車している等の移動中に使用し、その状況でブラウジングする際に、体が揺れることにより発生する「手元のぶれ」を補正し、ポインティングを正確に行えるようにするシステムを提案、実装した。また、提案手法を用いた歩行時のポインティング実験を行い、ポインティング成功率による客観的評価によって、その有用性を評価した。</p> <p style="margin: 0;">近年、タッチパネルを搭載し、加速度センサーなどの各種センサーを搭載した小サイズのモバイル端末-スマートフォンが普及している。しかし、スマートフォンは画面が小さく、正確にポインティングができないことがある。特に、使用している場所が電車やバス内などの、意図しない「揺れ」が発生する状況下である場合、その「揺れ」が原因でポインティングがずれることがある。そこで本研究では移動中のブラウジングに発生する「手元のぶれ」を補正し、ポインティングを正確に行えるようにする手法として、加速度を元にポインティング位置を補正するシステムを提案、実装した。</p> <p style="margin: 0;">まず予備実験を行い、ぶれの量と加速度の値の関係性を求めた。その結果、加速度とぶれの量の間に関連は見られなかった。分布を見ると、ぶれの量は加速度の値によらず、一定値以内に収まることが判明した。そこで補正量と補正方向を別に求め、補正量はぶれの量の平均と偏差から、補正方向は機械学習によって求めた。評価実験を行ったところ、補正前のポインティング成功率は 91.2%、補正後は 92.0%となり。有意差は無いがわずかに補正後の成功率が高くなる結果となった。また補助補正を加える事で全体的な成功率が上昇し、その場合でも加速度による補正は、補正前が 94.3%、補正後が 95.5%と、わずかに有効であった。以上から、加速度によるポインティングの補正はわずかに有効であり、より正確な補正を行うことが出来れば、補正なしのポインティングよりも有意に正確にポインティングができると考えられる。</p>			

# 平成24年度修士論文

## 移動中のスマートフォンを用いたウェブ閲覧における 手ぶれ補正システムの提案と評価

電気通信大学大学院 情報理工学研究科  
情報・通信工学専攻 コンピュータサイエンスコース

角田 HI\* 研究室

指導教員：角田 博保 (Hiroyasu Kakuda)  
岩田 茂樹 (Shigeki Iwata)

学籍番号：1131041 / 栗原 諒 (Kurihara Ryo)

提出日：平成25年1月25日(金)

# 概要

本研究では、小サイズのモバイル端末を、歩行中や電車・バスに乗車している等の移動中に使用し、その状況でブラウジングする際に、体が揺れることにより発生する「手元のぶれ」を補正し、ポインティングを正確に行えるようにするシステムを提案、実装した。また、提案手法を用いた歩行時のポインティング実験を行い、ポインティング成功率による客観的評価によって、その有用性を評価した。

近年、タッチパネルを搭載し、加速度センサーなどの各種センサーを搭載した小サイズのモバイル端末-スマートフォンが普及し、日常的に様々な状況下で使用できるようになっている。いつでも気軽にインターネットに接続できるという特徴から、ウェブブラウジングや地図検索、ソーシャルゲーム等に使用することが多い。しかし、iPhone を代表とするスマートフォンは、画面が小さくて正確にポインティングができないことがある。特に、使用している場所が、電車やバス内などの意図しない「揺れ」が発生する状況下であり、その「揺れ」が原因でポインティングがずれてしまうことがある。そこで本研究では移動中のブラウジングに発生する「手元のぶれ」を補正し、ポインティングを正確に行えるようにする手法として、加速度を元にポインティング位置を補正するシステムを提案、実装した。

まず予備実験を行い、ぶれの量と加速度の値の関係性を求めた。その結果、加速度とぶれの量の間に関連は見られなかった。分布を見ると、ぶれの量は加速度の値によらず、一定値以内に収まることが判明した。そこで補正量と補正方向を別に求め、補正量はぶれの量の平均と偏差から、補正方向は機械学習によって求めた。評価実験を行ったところ、補正前のポインティング成功率は91.2%、補正後は92.0%となり、有意差は無いがわずかに補正後のほうがポインティング成功率が高くなる結果となった。また補助補正を加える事で全体的な成功率が上昇し、その場合でも加速度による補正は、補正前が94.3%、補正後が95.5%と、わずかに有効であった。以上から、加速度によるポインティングの補正はわずかに有効であり、より正確な補正を行うことが出来れば、補正なしのポインティングよりも有意に正確にポインティングができると考えられる。

# 目次

概要	i
<b>第1章 はじめに</b>	<b>1</b>
1.1 現状の問題点	1
1.2 タブレット	1
1.3 本研究の目的	2
1.4 本論文の構成	2
<b>第2章 関連研究</b>	<b>3</b>
2.1 加速度に関する研究	3
2.2 人の歩行に関する研究	5
2.3 ポインティング特性	6
2.4 本研究の位置づけ	7
<b>第3章 提案システム</b>	<b>8</b>
3.1 概要	8
3.2 設計方針	8
3.2.1 開発環境	8
3.2.2 内部設計	8
3.2.3 手ぶれを補正する方法	8
3.2.4 ポインティングについて	9
3.2.5 補正方法	9
<b>第4章 予備実験</b>	<b>10</b>
4.1 目的	10
4.2 概要	10
4.3 被験者	10
4.3.1 実験装置	10
4.4 実験内容	11
4.4.1 起動から終了まで	11
4.4.2 実験中	12
4.4.3 ログデータ	12
<b>第5章 予備実験結果と考察</b>	<b>14</b>
5.1 結果	14
5.2 考察	14

<b>第6章 本実験</b>	<b>15</b>
6.1 目的	15
6.2 概要	15
6.3 被験者	15
6.4 実験内容	15
6.4.1 起動から終了まで	15
6.4.2 実験中	17
<b>第7章 本実験結果と考察</b>	<b>19</b>
7.1 補正方向の決定	19
7.2 補正の決定	19
7.3 結果の算出方法	19
7.4 結果	20
7.4.1 ポインティング補正の成否	20
7.5 問題点と改善案	21
<b>第8章 追加実験</b>	<b>22</b>
8.1 目的	22
8.2 補助補正方法概要	22
8.3 追加実験被験者	23
8.4 追加実験内容	23
8.5 追加実験結果	23
8.6 追加実験考察	24
<b>第9章 おわりに</b>	<b>25</b>
9.1 展望	25
9.1.1 より精度の高い補正	25
9.1.2 既存のインターネットブラウザアプリへの組み込み	25
<b>参考文献</b>	<b>26</b>
<b>謝辞</b>	<b>27</b>

# 目次

2.1	WalkType:システム使用時	3
2.2	WalkType:システム未使用時	3
2.3	NoShake:画面の表示の安定化	3
2.4	TiltText:プッシュキーと加速度を組み合わせた入力	4
2.5	GesText:Matrix レイアウト	4
2.6	GesText:tri center レイアウト	4
2.7	WUI:歩行ユーザーインタフェース	5
2.8	計測した加速度の値によるポインティング位置の補正	6
2.9	ボタンサイズによるタッチ位置の違い	6
3.1	ポインティング位置がずれた場合	9
3.2	補正先にボタンがあった場合	9
3.3	補正先にボタンがない場合	9
4.1	Android ウォークマン:NW-Z1000 シリーズ	11
4.2	予備実験:ログイン画面	11
4.3	予備実験:実験中	11
4.4	予備実験:終了画面	11
4.5	予備実験:実験中1	12
4.6	予備実験:実験中2	12
4.7	予備実験:実験中3	12
4.8	追加予備実験ログデータ	13
5.1	加速度の影響:X 軸方向のぶれと X 軸加速度の値の関係	14
5.2	加速度の影響:Y 軸方向のぶれと Y 軸加速度の値の関係	14
6.1	ログイン画面	16
6.2	準備画面	16
6.3	実験画面	16
6.4	終了画面	16
6.5	初期状態	18
6.6	中央ボタンをポインティング	18
6.7	中央ボタンをポインティングした後	18
6.8	目標ボタンをポインティング	18
6.9	目標ボタンの色の変化, 次の目標表示	18
6.10	最後の目標ボタンをポインティング	18
6.11	すべての目標ボタンの色が元に戻る	18

6.12	目標をポインティングできなかつた場合	18
6.13	目標をポインティングできなかつた結果	18
7.1	ポインティング成功率(被験者 A)	20
7.2	ポインティング成功率(被験者 B)	20
7.3	ポインティング成功率(被験者 C)	20
7.4	ポインティング成功率(被験者 D)	20
8.1	ずれた位置をポインティング	22
8.2	加速度によって補正する	22
8.3	一定範囲内の要素を検索	22
8.4	最も範囲の大きい要素へイベントを送る	22
8.5	追加実験結果	23
8.6	本実験と追加実験の比較	24

# 表 目 次

7.1	補正前のポインティング成功率 . . . . .	20
7.2	補正後のポインティング成功率 . . . . .	21
7.3	全体のポインティング成功率 . . . . .	21
8.1	補助補正を行ったポインティング成功率 . . . . .	23
8.2	加速度補正の有無と補助補正の有無の比較 . . . . .	24

# 第1章 はじめに

## 1.1 現状の問題点

昨今、タッチパネル式携帯端末の普及により、様々な状況下でタッチパネルを使用する機会が増加した。タッチパネルは直感的な操作が可能であること、入力と表示を一つの画面にまとめているという点から、小型機器に使用出来る、デザインの自由度が高いなどの特長がある。小型機器の代表格である携帯電話においては Apple の iPhone<sup>1</sup> を筆頭に、Google の Android<sup>2</sup> を搭載したスマートフォンが数多く存在し、多くのタッチパネルを搭載した携帯端末が普及している。このような携帯端末は、日常的にどこでもインターネットに接続できるという特徴から、ウェブブラウジングや地図検索、メールチェックやゲーム等、使用用途は多岐にわたる。しかし、これらの端末は小型であるが故に画面が小さく、文字が小さく読みづらい、ポインティングターゲットが小さく正確にポインティングできない、といったミスタッチを誘発する問題点となっている。また、どこでも使用出来るという特徴から、歩行中や電車、バスなどの利用中など、揺れが生じて手元が安定しない状況下で使用することも多く、上記の問題点であるポインティングターゲットの小ささも相まって、かなりポインティングしにくいものになっている。こういった外的要因による「手元のぶれ」は抑えることが困難で、かつ日常的に発生するものである。この問題は表示の拡大操作を行う、もしくはページレイアウトを変更してポインティングターゲットを大きくするといったことで多少解決できるが、拡大操作はその分操作に余計な時間がかかり、またページレイアウトの変更はページ作者の意図を無視してしまいかねない。

## 1.2 タブレット

元々の問題点である「画面が小さい」点を克服するために、iPad<sup>3</sup>を代表とするタブレット端末を使用することで、上記の問題を軽減することができる。だが、端末自体のサイズがおおよそ 10 インチと大きくなり、また重量が 600g 前後になるため持ち運びがしにくくなる、操作するために両手で把持する、もしくは片手で支持しもう片方の手で入力を行うなど確実に両手を使わなければならない、等の別の問題が発生する。また、両手で把持をすると指が届かない場所が存在してしまう。

---

<sup>1</sup><http://www.apple.com/iphone/>

<sup>2</sup><http://www.android.com/>

<sup>3</sup><https://www.apple.com/ipad/>

### 1.3 本研究の目的

本研究では、常時携帯する小サイズのモバイル端末を対象にし、1.1節で述べたウェブ閲覧時のポインティングに関する問題点に対し、加速度を用いて手ぶれを補正するシステムを提案し、その有用性を評価することでタッチインタラクションの改善を目指す。提案するシステムは、現在市場に流通している一般的な小サイズモバイル端末でセンシング可能な加速度を利用することで、ポインティング位置の補正を行う。また有用性の評価は、被験者によるポインティング実験を行なって得たデータを解析し、ポインティングのエラー率による客観的評価を行う。

### 1.4 本論文の構成

以下、

第2章では本研究に関連する加速度を用いたシステムや、人のポインティング特性についての研究を挙げる。

第3章では提案システムの設計方針について述べる。

第4章では予備実験について述べる。

第5章では予備実験の結果及び考察を述べる。

第6章では提案システムについて述べる。

第7章では評価実験について説明する。

第8章では評価実験の結果及び考察を述べる。

第9章では今後の展望を述べる。

第10章では本研究の結論を述べる。

## 第2章 関連研究

本章では、本研究で提案する加速度を用いたポインティング位置の補正の関連研究として、加速度を用いた研究、および人のポインティング特性について述べる。

### 2.1 加速度に関する研究

加速度を用いた研究は、WalkType[4]、NoShake[1]、TiltText[2]、GesText[3]などが存在する。

WalkTypeは歩行時のタッチスクリーンにおけるタイピングにおいて、加速度センサーの値を使用して手ぶれを補正している。また入力の際にタッチした座標を記憶し、機械学習することで、静止時における文字入力の精度も向上させている。結果として、歩行時の文字入力のエラーを42%抑えることができ、入力速度も12%ほど向上している。

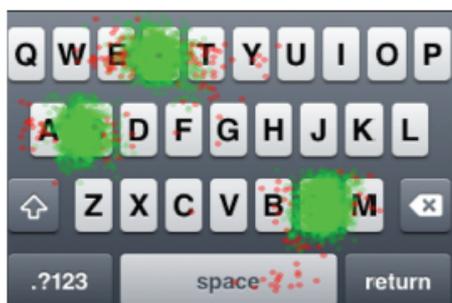


図 2.1: WalkType:システム使用時



図 2.2: WalkType:システム未使用時

NoShakeはスクリーン上に存在するコンテンツの表示安定化のためのシステムである。加速度センサーの値を使用することで、ぶれが発生した方向と逆の方向に画面内容をシフトすることで、画面の揺れを低減させ、ユーザーエクスペリエンスを向上させることができた。



図 2.3: NoShake:画面の表示の安定化

TiltText はプッシュキーを搭載した携帯電話に加速度センサーを取り付けた装置を使う。結果として TiltText はエラー率が既存の手法であるマルチタップ方式より高いものの、それでも入力速度は 23% 高速であった。



図 2.4: TiltText:プッシュキーと加速度を組み合わせた入力

GesText は手に加速度センサーを搭載した装置を持ち、手首を基点とするジェスチャーを行うことで文字入力を行う。結果は、マトリックスレイアウトインターフェースを採用するとおおよそ 3.7wpm(word per minutes) の入力速度となり、エラー率は 0.09% となった。初心者を使いやすく、また少量のテキスト入力に向く、と述べられている。

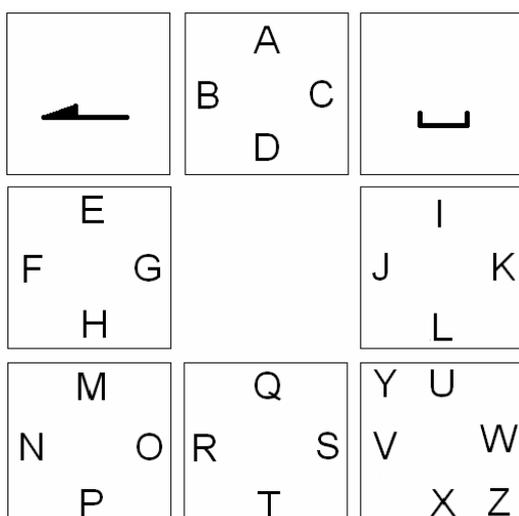


図 2.5: GesText:Matrix レイアウト

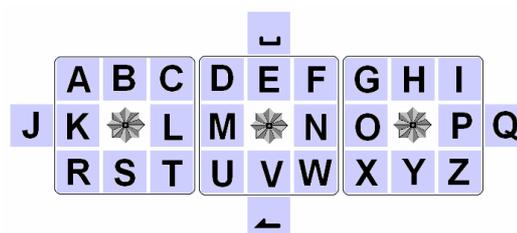


図 2.6: GesText:tri center レイアウト

## 2.2 人の歩行に関する研究

人の歩行に関する研究として Andrew ら [5] と Shaun ら [6] の研究がある。

Andrew らはユーザーがスクリーンをタップした際の、歩いている時、座っている時など、活動状況に関する情報を、加速度を搭載した PocketPC を使って収集した。これらのデータからユーザーの足取りを導出し、歩行に関するタップ動作を調査した。その結果、ユーザーは座っている方が正確にタップを行えること、歩行時に、大幅に増加するタップ尤度を伴う位相領域があり、その領域ではエラー率はかなり低く、またエラーのばらつきが少ない事がわかった。

Shaun らの研究では、歩行を感じるとボタンサイズが変化する WUI(Walking User Interfaces) を作り、その性能を評価した。結果として、WUI の 3 つのインタフェース”static-simple”、”Adapted”、”static-complex”のうち”static-simple”が 1 タスクあたりのページ遷移数も多く、エラー率も低く、ユーザーからのリッカート 7 段階評価でも高評価を得ることができた。



図 2.7: WUI:歩行ユーザーインタフェース

画像左側が静止状態 (加速度なし) の状態のもの、右側が歩行を感じた状態である。

## 2.3 ポインティング特性

人のポインティング特性を調べる研究では、Niel[7]らの研究や、西川ら[8]、西村ら[9]の研究が存在する。

Nielらの研究では、タイピングゲームによって人間のタイピング挙動特性を調べた。人はポインティング目標の若干下をタッチすることがわかり、標準のAndroidキーボードではタッチ位置を10dp補正していることがわかった。また、開発した補正システムを適用することで入力速度を2.2%向上させ、エラー率を9.1%低減させている。

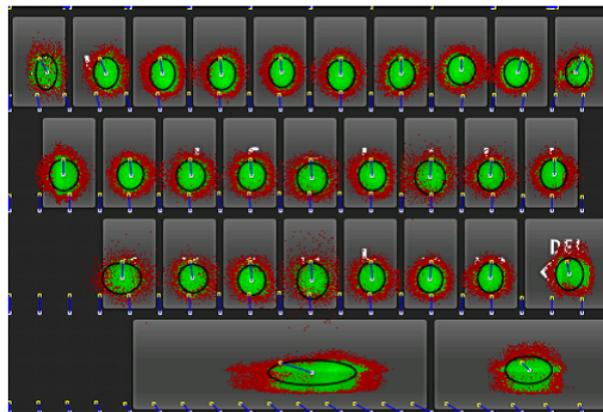


図 2.8: 計測した加速度の値によるポインティング位置の補正

西川らの研究では、人体に振動負荷をかけた時のタッチスクリーンのポインティング特性を調べた。入力加速度の強度と共振周波数の増加によって振動障害の影響が大きくなり、水平方向の振動よりも、鉛直方向の振動の方が、影響が大きいことが分かった。

西村らの研究では、ボタンのサイズの違いが、タッチパネル携帯端末の操作性にどういった影響を及ぼすかを調査した。ここでは”縦長”、”やや縦長”、”正方形”、”やや横長”、”横長”の5パターンについて調べている。

いずれの形状においてもサイズの増加によって、エラー率の低下、ポインティング時間の短縮が見られたが、ある一定以上のサイズになるとほぼ一定の値に収束する。また  $150\text{mm}^2$  以上にする事でエラー率、ポインティング時間が、ともに低くなることが判明した。加えて”縦長”、”横長”といった極端な形のボタンは、スイッチの反応領域が狭くなりやすく、意図した位置をポインティングしにくくなるため、操作性向上にはボタンの形状を考える必要があるとわかった。



図 2.9: ボタンサイズによるタッチ位置の違い

## 2.4 本研究の位置づけ

2.1 節で述べた研究は、加速度を用いた研究であるが、TiltText や GesText は入力対象がモバイル端末ではないという点、NoShake は画面の揺れを抑えるものであるという点、WalkType は装置を両手で持ち、両親指でタイピングすることを指定しているため、片手のみで持っている場合や、片手で装置を持ち、もう片手でタイピングをするといった実験をおこなっていない点が本研究と異なる。加えて、WalkType は文字入力の手法なので、どのボタンを中心に補正を行うかを入力予測を使って決定している。文字入力ではないポインティングは、どの要素をポインティングするかは予測できないため、同様の補正方法では効果が薄いと予想される。

2.3 節で述べた研究は、Niel らの研究では補正システムを開発しているが、この研究の補正量は、タイピング時のタッチ分布から導出されたものであり、フレキシブルに変化するものではない。そのため、突発的に発生する「ぶれ」には対応できない点が本研究とは異なる。

西川らや西村らの研究においては、この研究で得られた結果を本研究でも用いる。

# 第3章 提案システム

## 3.1 概要

本章では、本研究で提案したユーザーが歩行中などの移動中に、タップ操作を行った際に発生する「手ぶれ」を補正するシステムについて述べる。

## 3.2 設計方針

### 3.2.1 開発環境

システムはAndroid端末を対象とし、Javaで作成する。HTMLを使用するため、JavascriptとjQueryを併用する。

### 3.2.2 内部設計

#### Java

ファイル入出力や、加速度の取得など、Javascript・jQueryの処理に必要なデータを保持する。

#### HTML+Javascript

実験システムのレイアウトを記述し、ページの遷移などHTMLに関する処理を行う。

#### jQuery

タッチした時、ボタンをクリックした時の処理、加速度の処理をここで行う。なお、ポインティング位置の補正もここで行う。

### 3.2.3 手ぶれを補正する方法

手ぶれが発生する原因は外的要因による体の揺れである。外的要因自体の排除は困難であるので、手ぶれを補正する方向を目指した。手ぶれの補正を考える際に、スクリーン上のどこをタッチしたか(画面中央と画面端での違い)、個人の指の大きさの違い、ポインティング対象の大きさ、体の揺れの度合いなど、複数の要素が絡む。その中で今回は体の揺れに焦点を当てた。補正システムを考案するにあたって、以下の設計方針を立てた。

- Android を搭載したモバイルデバイス上に実装する
- 体の揺れから発生する手ぶれを、モバイル端末に生じる加速度として捉える
- 加速度の大きさによって、ポインティング位置の補正方向・補正量を変える

加速度の大きさによって補正方向・補正量を変えるので、その関係性を明らかにするために予備実験を実施した。

### 3.2.4 ポインティングについて

ポインティング位置を補正するということは、ユーザーがポインティングした位置は正しい位置ではないことになる。ポインティング位置の補正について以下の設計方針を立てた。

- ボタンやリンクなどが無い位置をタッチした場合
  - － 位置を補正した先にボタンやリンク等の要素があれば実行する
  - － 上記のようなアイテムがなければ何もしない
- ボタンやリンクなどをタッチした場合
  - － 位置を補正した先にボタンやリンク等の要素があれば、タッチ位置の要素は無視して、補正先の要素を実行する
  - － 補正した先がタッチした位置にある要素自身であれば、その要素を実行する

### 3.2.5 補正方法

「手元のぶれ」を装置の加速度とみなして取得し、その値によって補正方向を変化させる。20ms ごとに加速度を取得し、直近 100ms 分の加速度を保持する。タップイベントが発生した時刻から 100ms 分の加速度の変化量を確認し、ポインティング位置を補正する。位置の補正後、3.2.2 説で述べた流れに従いタッチイベントを処理する。

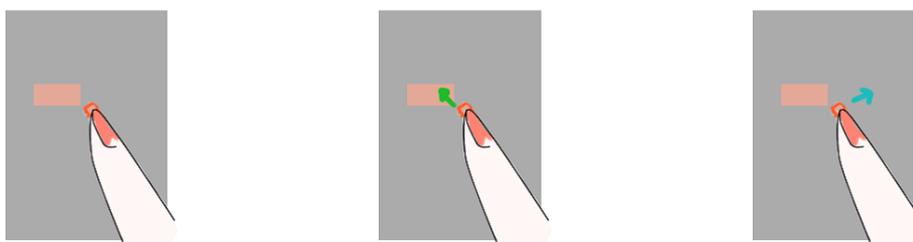


図 3.1: ポインティング位置がずれた場合

図 3.2: 補正先にボタンがあった場合

図 3.3: 補正先にボタンがない場合

## 第4章 予備実験

モバイル端末から取得した加速度センサーの値を使って、ポインティングが体の揺れによって、どの方向へどの程度ずれるのか、そのずれを補正するには加速度センサーの値をどのように使えばよいかを調査するため、予備実験を行った。

### 4.1 目的

歩行による体の揺れが、タッチパネル上でのポインティングにどの程度影響を与えるかを調査する。

以下の点についての調査を行った。

- 歩行中に発生する体の揺れを加速度センサーの値として取得
- タッチ目標の中心点と、実際にタッチした位置のずれ

またこのデータを元に、ポインティングの補正方法を決定する。

### 4.2 概要

片手で装置を持ち、もう片方の手でポインティングする方式でポインティングを160回行う。

最初に椅子に座った状態で、加速度を生じさせない状態でポインティングを行い、その人のポインティングの特性を調査する。

その後歩行しながらポインティング実験を行い、加速度の影響を調べる。

### 4.3 被験者

本大学の学生である男性11人に対して実験を行った。被験者はすべて右利きであるが、2人はスマートフォンを使用するときは左手で使用する と 答え、左手で使用した。

#### 4.3.1 実験装置

実験装置は Sony 製の Android ウォークマン NW-Z1050<sup>1</sup>を使用する。

---

<sup>1</sup>[http://www.sony.jp/walkman/products/NW-Z1000\\_series/index.html](http://www.sony.jp/walkman/products/NW-Z1000_series/index.html)



図 4.1: Android ウォークマン:NW-Z1000 シリーズ

## 4.4 実験内容

### 4.4.1 起動から終了まで

追加予備実験システムの起動から終了までの流れをいかに示す。

1. ログイン画面:これから実験を行う被験者名を選択し、ログインする。
2. 実験画面:合計 160 回のポインティングを行う。
3. 終了画面:ボタンを押すことでシステムを終了する。



図 4.2: 予備実験:ログイン画面



図 4.3: 予備実験:実験中



図 4.4: 予備実験:終了画面

## 4.4.2 実験中

被験者は歩行しつつ、ポインティング実験を行う。左右どちらかの手で装置を持ち、もう片方の手でポインティングを行う。実験中の流れをいかに示す。

1. 中央に「ポインティング目標確認ボタン」が表示される。
2. ボタンを押すと、ポインティング目標が表示される。
3. 再び中央のボタンを押すと、時間などの各種計測が始まる。
4. ポインティング目標のボタンを押す。

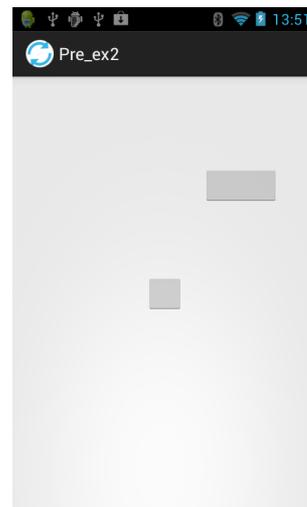


図 4.5: 予備実験:実験中 1 図 4.6: 予備実験:実験中 2 図 4.7: 予備実験:実験中 3

ポインティング対象の大きさは一律 60\*120pixel である。

## 4.4.3 ログデータ

予備実験のログデータを以下に示す。データは以下のとおりである。

- 1: イベントの種類
  - － A: 加速度の値
  - － B: ボタンのタップイベント
  - － T: ボタン外のタップイベント
  - － E: 1 タスク終了イベント
- 2: イベント発生時の時間
- 3: イベント発生時の X 軸加速度の値

	①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩	⑪	⑫	⑬	⑭	⑮	⑯
#S	1.36E+12															
#A	1.36E+12	-0.35304	1.676937	10.30679	-0.47044	1.780862	10.34863	0.117404	-0.10392	-0.04184						
#A	1.36E+12	-0.42169	1.510224	10.21853	-0.46069	1.726734	10.32261	0.039006	-0.21651	-0.10408						
#A	1.36E+12	111.4689	-8.54443	-2.26521	-0.44897	1.683433	10.26845	0.046896	-0.17321	-0.21663						
#A	1.36E+12	-0.35304	1.72597	9.738003	-0.42978	1.68194	10.16236	0.076743	0.03403	-0.42436						
#A	1.36E+12	-0.40207	1.735777	9.757616	-0.42424	1.700708	10.08141	0.022168	0.035069	-0.3239						
#A	1.36E+12	-0.46091	1.765197	9.845877	-0.43157	1.713605	10.03431	-0.02934	0.051592	-0.18843						
#A	1.36E+12	-0.51975	1.794617	9.973363	-0.44821	1.729808	10.02212	-0.07054	0.064609	-0.04875						
#A	1.36E+12	106.4982	-1.05231	-2.92521	-0.46528	1.762383	10.05552	-0.06428	0.130301	0.133594						
#A	1.36E+12	-0.55898	1.941717	10.28718	-0.48402	1.79825	10.10185	-0.07496	0.143467	0.185328						
#A	1.36E+12	-0.50995	2.029977	10.28718	-0.48921	1.844595	10.13891	-0.02074	0.185392	0.148262						
#A	1.36E+12	-0.56879	2.069397	10.37544	-0.50512	1.887555	10.18622	-0.06366	0.171841	0.189218						
#A	1.36E+12	-0.57859	2.02017	10.37544	-0.51952	1.914076	10.22406	-0.05878	0.106092	0.151374						
#A	1.36E+12	106.1893	-11.0241	-3.61131	-0.54726	1.929413	10.22298	-0.10976	0.061337	-0.00443						
#A	1.36E+12	-0.60901	1.93191	10.04201	-0.55941	1.929812	10.18677	-0.0486	0.001998	-0.14476						
#A	1.36E+12	-0.53937	1.882877	10.04201	-0.5554	1.920505	10.15782	0.016037	-0.03763	-0.11581						
#A	1.36E+12	-0.5984	1.84365	10.05182	-0.562	1.905134	10.13662	-0.0264	-0.06148	-0.0648						
#A	1.36E+12	-0.5984	1.833844	9.89491	-0.56728	1.890876	10.09829	-0.02112	-0.05703	-0.19337						
#A	1.36E+12	109.7849	-10.9766	-3.06711	-0.56366	1.903008	10.08295	0.014487	0.048518	-0.02132						
#A	1.36E+12	-0.56879	2.069203	10.14988	-0.5655	1.954992	10.10234	-0.00328	0.114212	0.047546						
#A	1.36E+12	-0.56879	2.069203	10.14988	-0.5655	1.954992	10.10234	-0.00328	0.114212	0.047546						
#B	1.36E+12	-0.56879	2.618376	9.816457	-0.5655	1.954992	10.10234	-0.00328	0.114212	0.047546	240	276	252.2373	287.3204	60*120	
#A	1.36E+12	-0.65705	2.618376	9.816457	-0.58381	2.087668	10.04516	-0.07323	0.530707	-0.2287						
#A	1.36E+12	-0.85318	2.029977	10.14008	-0.63769	2.07613	10.06414	-0.21549	-0.04615	0.075933						
#A	1.36E+12	112.6727	-1.04742	-3.36292	-0.63567	2.049248	10.1519	0.008048	-0.10753	0.351023						
#E	1.36E+12															
#A	1.36E+12	110.1002	-9.40982	-2.12802	-0.45425	1.84612	10.23494	0.06198	-0.12015	0.179723						
#start task2																
#S	1.36E+12															
#A	1.36E+12	-1.42196	4.442412	8.629851	-1.59935	4.070028	8.115141	0.177381	0.372385	0.51471						

図 4.8: 追加予備実験ログデータ

- 4: イベント発生時の Y 軸加速度の値
- 5: イベント発生時の Z 軸加速度の値
- 6: イベント発生時のローパスフィルターを適用した X 軸加速度の値
- 7: イベント発生時のローパスフィルターを適用した Y 軸加速度の値
- 8: イベント発生時のローパスフィルターを適用した Z 軸加速度の値
- 9: イベント発生時のハイパスフィルターを適用した X 軸加速度の値
- 10: イベント発生時のハイパスフィルターを適用した Y 軸加速度の値
- 11: イベント発生時のハイパスフィルターを適用した Z 軸加速度の値
- 12: ポインティング目標の X 座標
- 13: ポインティング目標の Y 座標
- 14: タップ位置の X 座標
- 15: タップ位置の Y 座標
- 16: ポインティング目標のサイズ

# 第5章 予備実験結果と考察

## 5.1 結果

静止状態でポインティング実験を行うことで、被験者のポインティング特性を知ることができた。そのポインティング特性を加味しつつ、加速度の影響を調べたところ、加速度の値とずれの量との相関がほぼ0となった。よって、加速度の値によって大きく変わるということではなく、おおよそ±20pixelの範囲内に収まった。加速度とぶれの散布図をいかに示す。

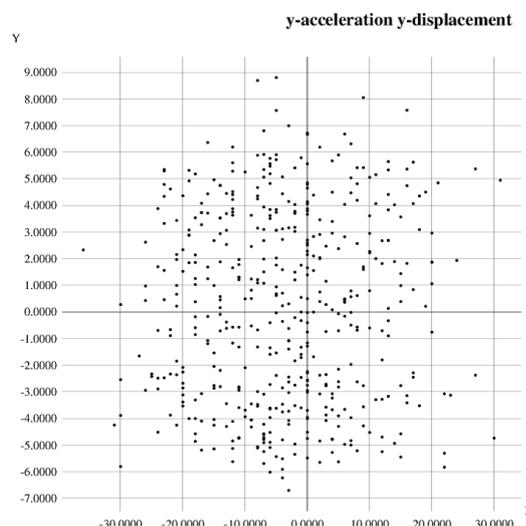
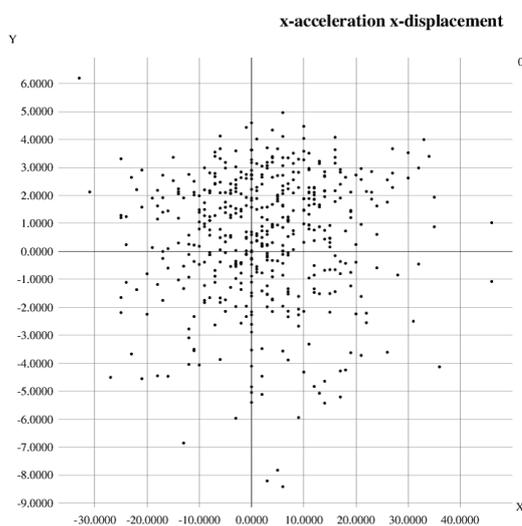


図 5.1: 加速度の影響:X 軸方向のぶれと X 軸  
加速度の値の関係

図 5.2: 加速度の影響:Y 軸方向のぶれと Y 軸  
加速度の値の関係

## 5.2 考察

加速度の値とポインティングのずれの大きさの相関がほぼ0であることから、ユーザーが無意識のうちにポインティング位置を補正していると考えられる。そのため補正量と補正方向をそれぞれ求めることにした。補正方向は、加速度の値を特徴量とした機械学習によって求めることにした。補正量は、ずれの分布の平均と偏差から、ポインティング特性の値とずれの量を決定した。

# 第6章 本実験

## 6.1 目的

提案システムの有用性を評価するため、提案システムを用いたポインティング実験を行う。また、提案システムを使わない、補正前のポインティングとの比較を行う。

## 6.2 概要

実験は横3個、縦3個、合計9個並んだボタンのうち、中央のボタンを始点とし、残りの8箇所のボタンの中から指定されたボタンをポインティングし、合計80回のポインティングを行う。実験中、被験者は常に歩いているものとし、装置を左手で把持し、右手でポインティングを行う。(実験中、被験者はバスに乗車しているものとし、装置を左手で把持し、右手でポインティングを行う)

## 6.3 被験者

本研究室の学生である男性4名に対して実験を行う。全員が普段から小サイズのタッチパネルを搭載したモバイル端末を使用している。

## 6.4 実験内容

### 6.4.1 起動から終了まで

追加予備実験システムの起動から終了までの流れをいかに示す。

1. ログイン画面 (図 6.1):これから実験を行う被験者名を選択し、ログインする。
2. 準備画面 (図 6.2):選択した被験者の名前と、何回目の実験かを確認する。
3. 実験画面 (図 6.3):合計 80 回のポインティングを行う。
4. 終了画面 (図 6.4):ボタンを押すことでシステムを終了する。

これを1セッションとし、機械学習用のデータを収集するために1セッション、評価実験として機械学習結果を組み込んだシステムを2セッション行う。

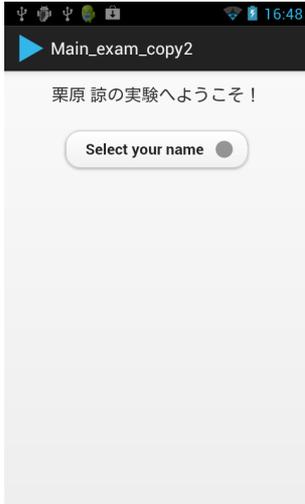


図 6.1: ログイン画面



図 6.4: 終了画面

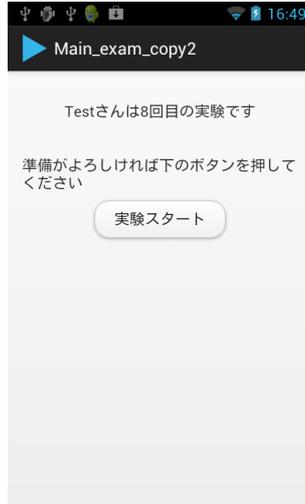


図 6.2: 準備画面

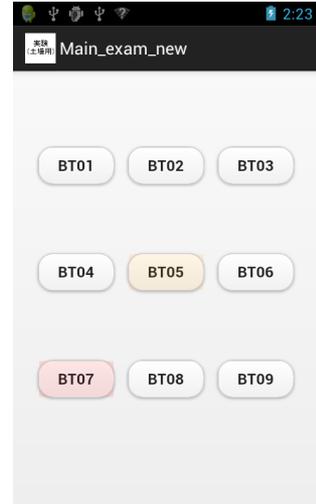


図 6.3: 実験画面

## 6.4.2 実験中

被験者は歩行しつつ、ポインティング実験を行う。左手で装置を持ち、右手でポインティングを行う。実験中の流れをいかに示す。

1. 最初に中央のボタンは黄色く、ポインティング目標のボタンが赤く表示されている (図 6.5)。
2. 黄色く表示されている中央のボタンをポインティングする (図 6.6)。
3. 中央のボタンを押すと、中央のボタンが黒色に変化する (図 6.7)。
4. 中央のボタンをポインティングした後、すぐに赤く表示されているボタンをポインティングする。 (図 6.8)。
5. ポインティング目標のボタンも黒く変換し、中央ボタンは黄色に戻り、次の目標ボタンが赤く表示される (図 6.9)
6. これを 8 箇所ボタン全てに対して行う (図 6.10)。
7. ボタンの色が全て元に戻る (図 6.11)。
8. 上記タスクを 1 タスクとして、これを 10 回繰り返し、合計 80 回のポインティングを行う。
9. なお間違えたボタンを押したり、ボタンをポインティングできないと、ボタンが青く変化する (図 6.12)。

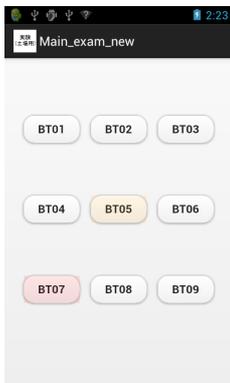


図 6.5: 初期状態



図 6.6: 中央ボタンをポインティング

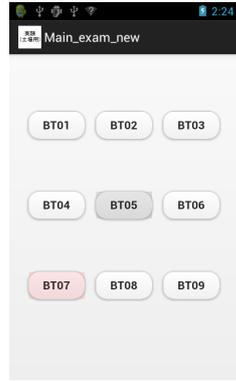


図 6.7: 中央ボタンをポインティングした後

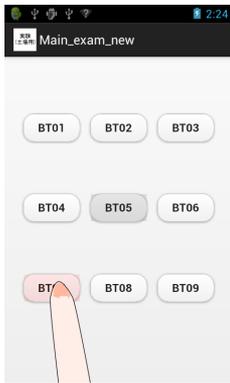


図 6.8: 目標ボタンをポインティング

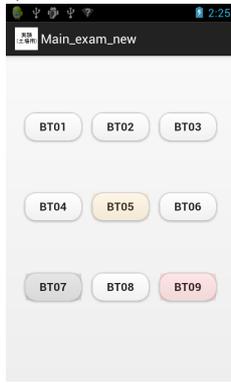


図 6.9: 目標ボタンの色が変化, 次の目標表示

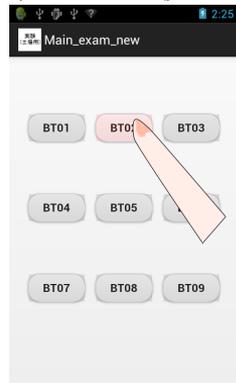


図 6.10: 最後の目標ボタンをポインティング

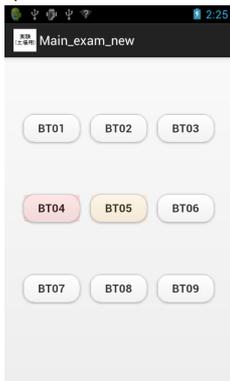


図 6.11: すべての目標ボタンの色が元に戻る

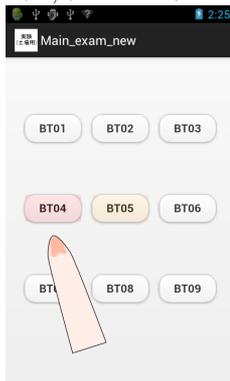


図 6.12: 目標をポインティングできなかった場合

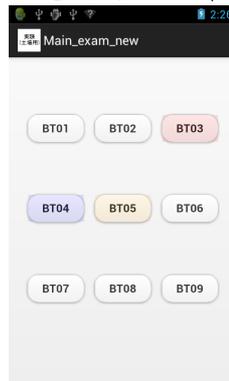


図 6.13: 目標をポインティングできなかった結果

# 第7章 本実験結果と考察

## 7.1 補正方向の決定

### 使用する機械学習ソフトウェア

機械学習にはオープンソースのデータマイニングソフトウェア Weka[10] を使用した。X 軸方向 (画面水平方向) のずれ、Y 軸方向 (画面垂直方向) のずれ、それぞれに対して X 軸加速度、Y 軸加速度、Z 軸加速度を特徴量として、補正方向を、正方向、0、負方向の 3 つの値として算出する。

## 7.2 補正の決定

予備実験の結果より、実際にポインティングした位置に対して、

- ポインティングのばらつきから被験者の「ずれる量」を決定する。
- 被験者のポインティング特性に合わせて、固定値分ポインティング位置を移動させる。
- 機械学習から得た結果と、200ms 分の加速度の値を元に補正方向を決定する。
- 3 つを合わせて、最終的なポインティング位置の補正を行う。

## 7.3 結果の算出方法

ポインティング目標であるボタンの中心位置を  $X, Y$ 、実際のポインティング位置を  $x, y$ 、補正したあとのポインティング位置を  $hx, hy$ 、補正距離を  $hX, hY$  とした時、以下の計算式によってポインティングが成功したかどうかを判定する。なお、ボタンの大きさは  $38*78$ pixel である。

$$(x + hX) = hx \quad (7.1)$$

$$(y + hY) = hy \quad (7.2)$$

$$|hx - X| < 39 \text{ かつ } |hy - Y| < 19 \text{ ならば、成功} \quad (7.3)$$

$$\text{そうでないならば失敗} \quad (7.4)$$

## 7.4 結果

### 7.4.1 ポインティング補正の成否

ポインティング位置補正の結果は以下のようになった。

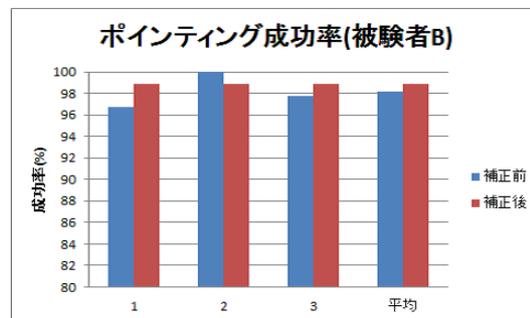
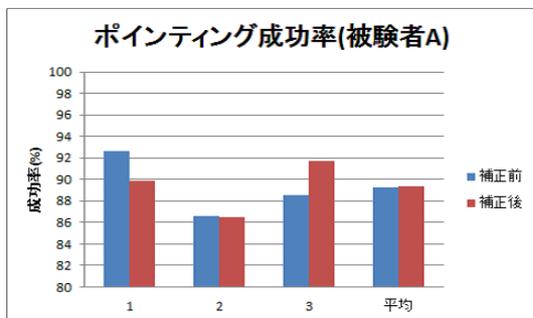


図 7.1: ポインティング成功率 (被験者 A)

図 7.2: ポインティング成功率 (被験者 B)

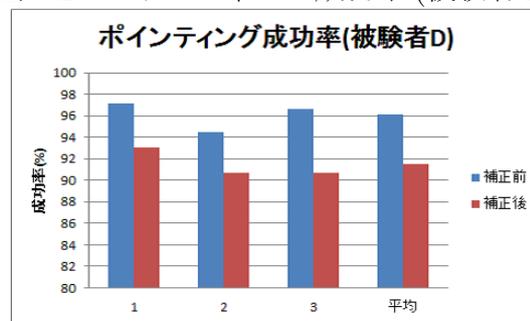
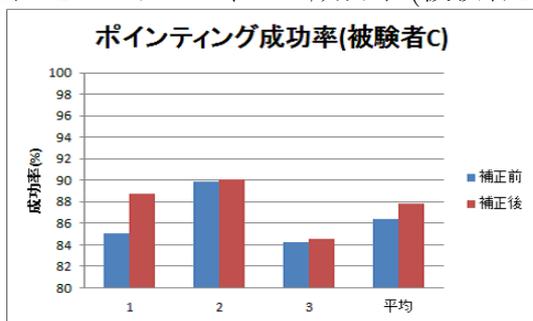


図 7.3: ポインティング成功率 (被験者 C)

図 7.4: ポインティング成功率 (被験者 D)

詳細な値を表にまとめると以下のようになる。

表 7.1: 補正前のポインティング成功率

	被験者 A	被験者 B	被験者 C	被験者 D
1 回目	92.6	96.7	85.1	97.1
2 回目	86.5	100	89.8	94.5
3 回目	88.5	97.7	84.2	96.6
平均	89.2	98.1	86.4	96.0

被験者 A,B,C に関しては有意差はなかったが、わずかに補正後のほうがポインティング成功率が高くなるという結果になった。

被験者 D のみ結果が悪いのでログデータを解析してみたところ、機械学習を行うために収集したデータの分布と、実験で取得したログデータの分布とが大きく異なっていることが判明した。またその人特有のポインティング特性であるずれも、セッションごとに異なっているということがわかった。これらの点から、被験者 D にとって本システムは相性が良くなかったと考えられる。

表 7.2: 補正後のポインティング成功率

	被験者 A	被験者 B	被験者 C	被験者 D
1 回目	89.9	98.9	88.8	93.0
2 回目	86.5	98.9	90.1	90.7
3 回目	91.7	98.9	84.6	90.7
平均	89.4	98.9	87.8	91.5

被験者 D を除いた 3 人のデータから平均ポインティング成功率を求めたところ、以下のようになった。

表 7.3: 全体のポインティング成功率

	ABC のみ
補正前	91.27
補正後	92.02

全体で見ても僅かではあるが補正後のほうがポインティング成功率が高くなるという結果になった。

## 7.5 問題点と改善案

被験者 D に関しては、ポインティング特性の値と、ずれの分布がセッションごとに異なるという結果になった。このような「ポインティング特性の値が一定しない」ユーザーは現状のシステムでは相性が悪いと考えられ、対応を模索する必要がある。

# 第8章 追加実験

## 8.1 目的

提案したシステムの有用性を向上させるために、補助の補正方法を考案した。

## 8.2 補助補正方法概要

加速度によって補正した座標より、± 10pixel の範囲にある要素を検索し、最も範囲の広い要素を選択する。

手順は以下のとおり。

- 画面上をポインティングする (ボタンからずれた位置をポインティングしたと仮定する)。図 8.1
- 加速度の値によってポインティング位置の補正を行う。図 8.2
- 補正したポインティング位置から一定範囲内の要素を検索する。図 8.3
- 一定範囲内にある要素が
  - ー 1つならば、その要素へイベントを送る。図 8.4
  - ー 2つ以上あるならば、最も範囲の大きい要素へイベントを送る。



図 8.1: ずれた位置をポインティングする  
図 8.2: 加速度によって補正する  
図 8.3: 一定範囲内の要素を検索

図 8.4: 最も範囲の大きい要素へイベントを送る

### 8.3 追加実験被験者

本研究室の学生、2名に対して実験を行った。

### 8.4 追加実験内容

実験内容に関しては本実験と同じである。

### 8.5 追加実験結果

追加実験の結果、補助補正を使い、加速度による補正の有無で比較したところ、以下のようになった。

表 8.1: 補助補正を行ったポインティング成功率

	被験者 A	被験者 B	平均
補正前 (%)	96.3	92.5	94.3
補正後 (%)	97.5	93.6	95.5

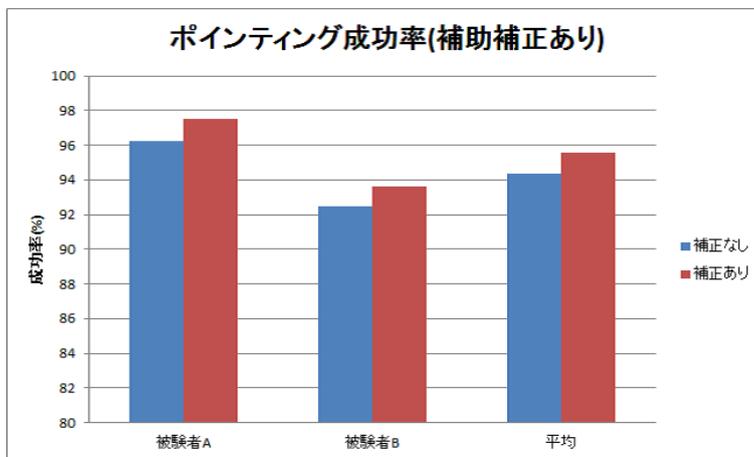


図 8.5: 追加実験結果

P 値は  $0.043 < 0.05$  となり、有意差があった。

また加速度補正の有無と、補助補正の有無をまとめた結果、以下のようになった。

表 8.2: 加速度補正の有無と補助補正の有無の比較

	加速度補正無し	加速度補正有り
補助補正無し (%)	91.2	92.1
補助補正有り (%)	94.3	95.5

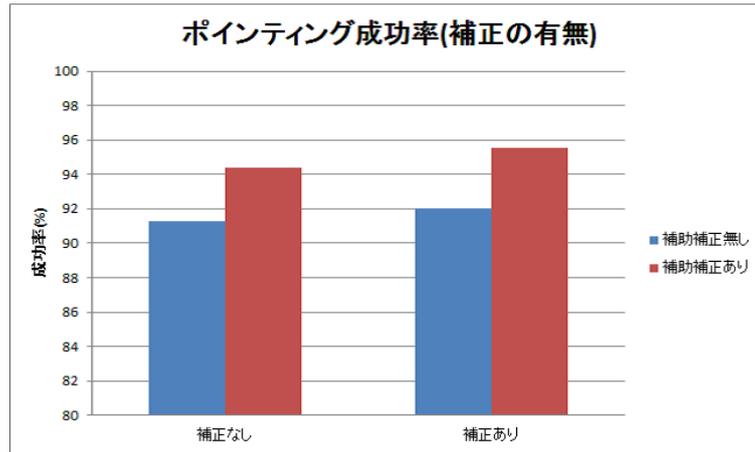


図 8.6: 本実験と追加実験の比較

加速度補正による有意差は、P 値が  $0.137 > 0.1$  となり、有意差はなかったが、成功率は若干上昇している。

## 8.6 追加実験考察

補助補正と加速度による補正を組み合わせることで、ポインティング成功率は上昇し、有用性が増すことがわかった。

## 第9章 おわりに

本研究では、小サイズのモバイル端末を、歩行中や電車・バスに乗車している等の移動中に使用し、その状況でブラウジングする際に、体が揺れることにより発生する「手元のぶれ」を補正し、ポインティングを正確に行えるようにするシステムを提案、実装した。

予備実験により被験者のポインティング特性、及び加速度の値に対してポインティングがどの程度ずれるかを調べ、機械学習によって、加速度とずれの方向の関係性を調べ、二つを組み合わせることで「手元のぶれ」を補正することができた。

結果は有意差は無いものの若干のポインティング成功率上昇が見受けられた。しかし、本来成功していたポインティングを、補正によって失敗にしている場合もあり、改良の余地はあるだろう。

### 9.1 展望

#### 9.1.1 より精度の高い補正

本システムでは加速度を手元のぶれとして扱い、その値によって補正を行った。今回は加速度しか用いなかったが、加速度の他にも存在するいくつかのセンサーからの値を組み合わせることで、より高い精度で補正を模索する。

#### 9.1.2 既存のインターネットブラウザアプリへの組み込み

今回はポインティング実験で本システムの有用性を評価した。実際のウェブアプリケーションに組み込んで使用すると、また異なる評価になると思われる。

## 参考文献

- [1] NoShake: Content stabilization for shaking screens of mobile devices. PerCom 2009, p1-6
- [2] TiltText: Using Tilt for Text Input to Mobile Phones UIST 2003, p81-90
- [3] GesText: Accelerometer-Based Gestural Text-Entry Systems CHI 2010, p2173-2182
- [4] WalkType: Using Accelerometer Data to Accommodate Situational Impairments in Mobile Touch Screen Text Entry CHI 2012, p2687-2696
- [5] Gait Phase Effects in Mobile Interaction CHI 2005, p1312-1315
- [6] Getting Off the Treadmill: Evaluating Walking User Interfaces for Mobile Devices in Public Spaces MobileCHI 2008, p109-119
- [7] Observational and Experimental Investigation of Typing Behaviour using Virtual Keyboards on Mobile Devices CHI 2012, p2659-2668
- [8] 振動負荷時におけるタッチ画面への人体のポインティング特性関西支部講演会講演論文集 2003(78), "12-29"- "12-30"
- [9] スイッチのサイズ及び形状が抵抗膜式タッチパネル携帯端末の操作性に及ぼす影響について日本機械学会論文集 (C 編) 2011(77), p84-94
- [10] Weka 3: Data Mining Software in Java <http://www.cs.waikato.ac.nz/ml/weka/>

# 謝辞

二年間に渡り御指導いただきました角田先生、赤池先生を始め、実験に快く協力していただいた研究室の方々に心から感謝を申し上げます。