平成25年度

筑波大学情報学群情報科学類

卒業研究論文

題目
大型タッチ画面を備える携帯情報端末向けの片手ポインティング手法

主専攻　　ソフトウェアサイエンス主専攻

著者　　大西主紗

指導教員　志築文太郎 高橋伸 田中二郎 三末和男

要　旨

大型タッチ画面を備える携帯情報端末上における片手ポインティング手法を示す．本手法では，ユーザは大型タッチ画面を備える携帯情報端末を片手にて把持し，その親指にて端末画面の全領域に対するポインティングを行う．直接指が届かない画面領域に対する操作を実現するために，本研究では画面端において反射を行うカーソルを用いる手法，およびドラッグ操作にて画面の指定した領域にタッチ点を転送する手法を提案する．これらの手法は画面に強くタッチを行うことよって起動されるため，タッチ画面上における既存の操作方法と共存することが可能である．提案手法のポインティング性能を検証するため，既存手法との比較実験を行った．結果，Android 端末におけるホーム画面のようにアイコンが規則正しく並ぶ環境においては，提案手法は既存手法に近い性能を発揮しうることが判明した．

目 次

第 1 章　序論 **1**

1.1 背景 1

1.2 目的 2

1.3 本論文の構成 2

第 2 章　関連研究 **3**

2.1 大端末上におけるポインティング 3

2.1.1 直接ポインティング 3

2.1.2 間接ポインティング 3

2.2 巨大画面環境におけるポインティング 4

2.3 携帯情報端末における入力語彙の拡張 5

第 3 章　提案手法 **6**

3.1 TouchOver 6

3.1.1 TouchOver の設計 6

3.1.2 TouchOver の実装 7

3.1.3 TouchOver の問題点 7

3.2 操作手法 9

3.2.1 TouchWarp 9

3.2.2 Reflection 11

3.3 起動手法 14

3.3.1 Forcetouch の設計 14

3.3.2 Forcetouch の実装 17

第 4 章　実験 1：Forcetouch 予備実験 **19**

4.1 実験設計 19

4.2 分析項目 19

4.3 結果 20

4.4 考察 21

第 5 章　実験 2：被験者実験 **23**

5.1 実験設計 23

5.2 実験手順 24

5.3 分析項目 25
5.4 結果：所要時間および失敗率 26
5.5 所要時間および失敗率の考察 27
5.6 理想起動環境における失敗率に基づく分析 27
5.7 ダミーポインティングについての分析 28

第 6 章 実験 3：TouchOver を用いた性能比較予備実験 31
6.1 実験設計 31
6.2 分析項目 31
6.3 結果：所要時間および失敗率 31
6.4 所要時間および失敗率の考察 32
6.5 理想起動環境における失敗率に基づく分析 33
6.6 ダミーポインティングについての分析 35

第 7 章 結論および今後の課題 36
7.1 操作手法（TouchWarp） 36
7.2 操作手法（Reflection） 37
7.3 起動手法 38
7.4 TouchOver 38

謝辞 39

参考文献 40

付 録 A 付録 43
A.1 実際に使用した書類 43
A.1.1 誓約書 44
A.1.2 実験手順書 45
A.1.3 アンケート 48
A.2 アンケート結果 50
A.2.1 操作感 50
A.2.2 Forcetouch の感想 51
A.2.3 Touch の感想 51
A.2.4 CornerSpace の感想 52
A.2.5 TouchWarp の感想 52
A.2.6 Reflection の感想 53

図 目 次

3.1 TouchOver の使用例．a : Forcetouch による起動．b : タッチ点の転送．c : タッチアップによる TouchWarp の終了． 6
3.2 TouchOver におけるタッチ点及び転送後のタッチ点の位置の関係． 8
3.3 TouchOver の問題点．a : タッチが不可能な手元領域およびタッチ点の転送が不可能な領域． b ：ポインティングが不可能な領域． 8
3.4 TouchWarp の使用例．a : Forcetouch による起動．b : ドラッグ操作による転送先の決定．c : タッチアップによる TouchWarp の解除． 9
3.5 TouchWarp におけるタッチ点及び転送後のタッチ点の位置の関係． 10
3.6 カーソルを引き戻さず行う再ポインティング．a : ターゲットを通り越したカーソル．b : 反射を行い再度ターゲットをポインティングするカーソル． 11
3.7 カーソルの表示位置演算のフローチャート． 12
3.8 カーソルの表示位置演算の例． 12
3.9 Reflection の使用例．a : Forcetouch による Reflection の起動．b : ドラッグ操作によるカーソルの移動．c : 画面端におけるカーソルの反射． 13
3.10 Reflection におけるタッチ点及びドラッグ操作後のカーソルの位置の関係 . . . 14
3.11 キャリブレーション画面． 16
3.12 キャリブレーション時に使用したターゲットと分割した領域の大きさ． . . . 16
3.13 閾値 F_S 及び F_W の関係． 18

4.1 実験の様子． 20
4.2 閾値設定のための分割領域． 20
4.3 予備実験結果：strong 条件の分類成功率および分類失敗率． 21
4.4 予備実験結果：weak 条件の分類成功率および分類失敗率． 21
4.5 予備実験の結果：被験者ごとの分類失敗率． 22

5.1 実験時の様子． 23
5.2 ダミーなし条件． 25
5.3 ダミー有り条件． 25
5.4 実験結果：1 タスクの所要時間． 26
5.5 実験結果：失敗率． 27
5.6 実験結果：分類不能率． 28
5.7 実験結果：理想起動環境における失敗率． 29

5.8 実験結果：ダミーポインティング率. 30

6.1 実験結果：1 タスクの所要時間. 32
6.2 実験結果：失敗率. 33
6.3 実験結果：分類不能率. 34
6.4 実験結果：理想起動環境における失敗率. 34
6.5 実験結果：ダミーポインティング率. 35

7.1 転送先のヒント案. 37

A.1 誓約書. 44
A.2 実験手順書 1. 45
A.3 実験手順書 2. 46
A.4 実験手順書 3. 47
A.5 アンケート 1. 48
A.6 アンケート 2. 49

第1章　序論

本章では，まず研究の背景と目的を述べる．次に，本研究の目的を述べる．最後に，本論文の構成を述べる．

1.1　背景

近年，大型のタッチ画面を備える携帯情報端末（以降，大端末）が市場に登場し，ユーザに使用されるようになった．また，Parhi ら [PKB06] によると，ユーザの大部分は片手操作，すなわち端末を把持した手の親指のみを用いた携帯情報端末の操作を望んでいる．

しかし，片手操作によって大端末を操作する場合，ユーザの指が届かない画面領域（以降，困難領域）に対する操作が困難であるという問題が存在する [YHHH13]．そのため，大端末に対する片手操作の研究が盛んに行われている．研究の例として，画面上に備えたカーソルを用いたポインティングを可能とする研究 [KYL12]，画面上でのドラッグ操作によって生成されるタッチパッドを用いたオブジェクトの選択を可能とする研究 [KB08]，背面ジェスチャを用いた困難領域の手元への移動を可能とする研究 [土佐 13] などが挙げられる．また，携帯情報端末のタッチ入力の語彙を拡張する研究が行われている．例として，画面に対するタッチの強弱を用いる研究 [HL11b]，端末にかかる圧力を用いる研究 [GWP12]，タッチに用いた指の部位を用いる研究 [HSH11] などが挙げられる．本研究は，以下に示す操作手法を用いることにより，困難領域のポインティングを可能とする．

- Reflection : ドラッグによる操作が可能な，画面の端において反射するカーソルを用いる操作手法
- TouchWarp : タッチ点を指が届かない画面領域に転送させる操作手法

また，本研究は Forcetouch と呼ばれる起動手法を用いてこれらの操作手法を起動する．これは Forcetap[HL11b] の原理を応用した，タッチの強弱を検出する手法であり，ユーザが画面を強くタッチした際に提案操作手法を起動する．この Forcetouch を起動手法とすることにより，ユーザは操作手法を，タッチ画面を備える端末にて従来から使用されてきたダブルタップやピンチなどのジェスチャ操作と使い分けることが可能である．以降，これらの起動手法およびそれに続く操作手法の組み合わせをポインティング手法と呼ぶ．また，2 種類の操作手法を用いるポインティング手法をそれぞれ Forcetouch & TouchWarp，Forcetouch & Reflection と呼ぶ．

また，提案ポインティング手法の実装に要する入力デバイスは，シングルタッチの検出が可能なタッチ画面，および加速度センサである．このため，提案ポインティング手法は多くの帯情報端末にて動作可能である．

1.2 目的

本研究の目的は，片手による大端末のポインティングを可能とする手法の提案および実装を行い，既存の操作手法との比較を行うことである．

1.3 本論文の構成

第 1 章では，本研究の背景と目的を述べた．第 2 章では，関連研究と本研究の位置付けを述べる．第 3 章では，提案した起動手法および操作手法，TouchWarp のプロトタイプである TouchOver について，設計および実装を述べる．第 4 章では，Forcetouch の性能評価を行った予備実験（以降，実験 1）を述べる．第 5 章では，提案ポインティング手法の性能評価を行った被験者実験（以降，実験 2）を述べる．第 6 章では，Forcetouch & TouchOver の性能評価を行った予備実験 (以降，実験 3) について述べる．第 7 章では，本論文の結論および今後の課題を述べる．

第2章　関連研究

本章では，本研究の関連研究および位置付けを述べる．初めに，本研究と同じく大端末上におけるポインティングやターゲットの選択を対象とした研究を挙げる．なお，ユーザのポインティング対象を，以降ターゲットと呼ぶ．また，大画面を持つデスクトップ並びにテーブルトップ環境（以降，巨大画面環境）においても，カーソルやユーザの立ち位置から遠く離れたターゲットのポインティングを対象とした研究を行っている．よって、これらの研究も本研究と関連すると言える。最後に，起動手法に関連する研究として，携帯情報端末における入力語彙を拡張する研究を挙げる．

2.1　大端末上におけるポインティング

大端末上における直接ポインティングおよび間接ポインティング手法の研究を挙げる．

2.1.1　直接ポインティング

困難領域に存在するターゲットを指の届く範囲に移動させることにより，直接選択を可能とするポインティング手法がこれまでに研究されてきた．LoopTouch[土佐 13] では，ユーザは，大端末の裏面に人差し指を，表面に親指をタッチした状態で，両指の相対位置が近づく方向にスワイプする．この操作により画面領域全体がループするため，ユーザはターゲットを指の届く範囲に移動させることが可能となる．この操作を検出するために，LoopTouch では裏面にタッチパネルを装着した大端末を用いる．Sliding Screen[KYL12] では，画面を広い面積でタッチ，もしくは BezelSwipe[RT09] を行った後のドラッグ操作によって，ユーザは画面領域全体を指の方向に移動させ指の届く位置にターゲットを移動させることが可能である．永田らは，ベゼルスワイプを行うことにより，困難領域にあるオブジェクトを指の届く領域に移動させる手法を提案している [永田 12]．一方，本研究はこれらと異なり間接ポインティング手法を用いる．また [永田 12] とは異なりオブジェクト以外の任意の箇所のポインティングも可能とする．

2.1.2　間接ポインティング

ThumbSpace[KB08] は，ユーザのドラッグ操作によって画面領域の縮小画像が表示されたタッチパッドを任意の位置に生成する．ユーザはこのタッチバッドを操作することによって

指の届かない位置のオブジェクトを選択することが可能である．ユーザは，タッチパッドをタッチすることによりオブジェクトにフォーカスを当て，指をドラッグすることによりフォーカスを別のオブジェクトに移し，指を離すことによりフォーカスされたオブジェクトを選択する．一方，我々の手法はオブジェクト以外の任意の箇所のポインティングも可能である．

また，カーソルを用いてポインティングを行う手法も存在する．MagStick[RHL08] および Shift[VB07] は”fat fingers”[SRC05] を解決するためにカーソルを用いているのに対して，本研究の Reflection では指の届かない画面領域に存在するターゲットの選択を行うためにカーソルを用いる．CornerSpace[YHHH13] は，カーソルの出現位置を決定するためのウィジェットを使用するポインティング手法である．ベゼルスワイプによって画面を 4 領域に分割する円形ウィジェットが生成され，その分割されたいずれかの領域のタッチにより，その領域に対応した画面隅にカーソルを表示する．ユーザはこのカーソルをドラッグ操作によって移動させ，ターゲットを選択する．このポインティング手法は起動および操作手法間に 1 度タッチアップを要するため，ポインティングを 1 連の操作として行うことができない．これに対して，本研究の 2 手法ではどちらも 1 段階の操作のみ要する．Extendible Cursor[KYL12] は，広い面積でのタッチ，もしくは BezelSwipe が行われたとき，ドラッグ操作によって移動させることのできるカーソルを画面に表示する．ユーザはこれを用いてターゲットの選択を行う．本研究における Reflection にて用いられるカーソルは，画面に対する強いタッチが行われたとき表示される．

2.2 巨大画面環境におけるポインティング

大画面を持つデスクトップ環境におけるポインティングを行う手法がこれまでに研究されてきた．Ninja Cursors[KI08] は，画面上に複数のカーソルを表示することにより，カーソルとターゲットの平均距離を縮めている．これによりポインティングに必要なカーソルの移動距離を減少させることが可能である．本研究における TouchOver は，1 つのタッチ点に対してポインティング先を 4 種類用意することにより，直接タッチが必要な領域を減少させている．TorusDesktop[HCD11] は，カーソルがデスクトップ画面の端から画面外に移動する際，逆の画面端から出現するような移動を提案した．これにより，カーソルを画面の端から端へ移動させる際のカーソルの移動距離を減少させることが可能である．一方本研究における Reflection は画面外の領域に移動する際，画面端にて反射する．

また，タッチ画面を備えるテーブルトップ環境における，手の届かない領域に対するポインティング手法についても研究がなされている．I-Grabber[ALMP09] は，マルチタッチインタラクションにより操作するウィジェットを用いて遠隔地をポインティングする．HandyPointing[YMK+13] は，引き出しジェスチャ[YST12] を用いて操作するカーソルを用いて遠隔地をポインティングする．一方本研究においては，親指一本にてポインティングを行う．また，手の接触形状を用いた仮想マウスによるポインティング手法を提案した研究もある [BHOH11, MGLF09]．本研究はこれらと異なり，単一のタッチ点のみを用いてポインティングを行う．

2.3 携帯情報端末における入力語彙の拡張

端末に組み込まれたセンサを用いて，携帯情報端末における入力語彙の拡張を行う研究が存在する．BezelSwipe[RT09] は，ユーザがスワイプを画面の縁から行うことによって，通常のスワイプと異なる機能の起動を可能とした．GripSense[GWP12] は，組み込みのジャイロセンサおよび振動モータを用いて端末にかかる圧力を検出した．これによりユーザは，端末の握り具合を変えることにより様々な機能を使い分けることが可能である．Bezel Swipe には既存の操作が割り当てられていることが多い．また，大端末は手で握りこむことが困難であるため，握り具合を変えることが困難である．したがって，本研究においてはタッチの強さを入力語彙の拡張に用いることとした．Forcetap[HL11b] は組み込み加速度センサの値を用いてユーザのタップの強さを検出した．これにより，タップに複数の機能を割り当てることを可能とした．本研究はタップではなくタッチの強さを検出し，タッチに複数の機能を割り当てることとした．

また，外部センサを用いて入力語彙を拡張する研究も存在する．Force Gestures[HL11a] は，ケースに取り付けた圧力センサを用いて端末にかかる圧力を検出することにより，タッチスクリーンにおける新たなジェスチャを可能とした．TapSense[HSH11] は，端末に取り付けたマイクを用いて，タッチに用いた指の部位を検出した．これによりユーザは，タッチの際に指の部位を使い分けることにより，様々な機能を使い分けることが可能である．一方本研究では組み込みのセンサのみを用いるため，外部センサを用いることなく実装可能である．

第3章　提案手法

本章では，まず TouchWarp のプロトタイプである TouchOver について述べる．その後，提案操作手法である Reflection，TouchWarp，および，起動手法である Forcetouch について述べる．

3.1　TouchOver

本節では TouchOver の設計，実装，その問題点について述べる．

3.1.1　TouchOver の設計

TouchOver は，ユーザが画面下の領域に入力したタッチ点を，画面上領域に転送する操作手法である．これにより，ユーザの指が届く画面下領域における操作のみを用いて，指の届かない画面上領域に存在するターゲットの選択を可能とする．

TouchOver は，まずユーザが画面に対して強いタッチを行い，タッチアップによって困難領域をポインティングする．TouchOver の使用方法を図 3.1 に示す．ユーザはまず，図 3.1a のように画面に対して強いタッチを行う．これにより，図 3.1b のようにタッチ点を画面上領域に転送する．このとき，TouchOver はユーザに対しする転送先のタッチ点のフィードバックとしてカーソルを表示する．ユーザが指をタッチ画面から離したとき，図 3.1c のように TouchOver が終了する．

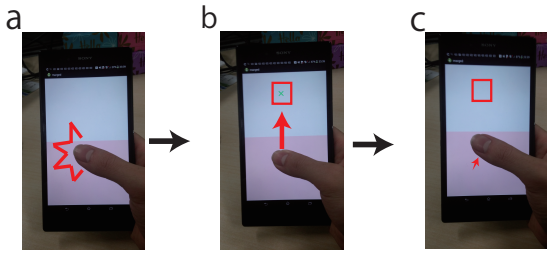

図 3.1: TouchOver の使用例．a：Forcetouch による起動．b：タッチ点の転送．c：タッチアップによる TouchWarp の終了．

3.1.2 TouchOver の実装

本手法により転送されるタッチ点の位置 $P(x_P, y_P)$ の算出法を詳述する．ユーザのタッチ点を $S(x_S, y_S)$，画面の縦方向の長さを h とした場合，図 3.2 に示すように以下とする．

$$(x_P, y_P) = (x_S, y_S - \alpha \times h) \tag{3.1}$$

この α を，直接タッチが必要な領域が最も小さくなるように決定する．指が最も届きにくい領域である画面上端にタッチ点を転送する際に直接タッチを行う領域が，画面下端の領域からタッチ点を転送する領域に一致する場合，直接タッチが必要な領域が最も小さくなる．

まず，式 3.1 の転送量にしたがった場合，転送後のタッチ点の y_P 座標は

$$y_S - \alpha \times h \tag{3.2}$$

である．したがって，タッチ画面の上端，すなわち y_P=0 となる領域をポインティングするためには，

$$y_S = \alpha \times h \tag{3.3}$$

を満たす領域をタッチする必要がある．この領域が，タッチ点を転送させたときにポインティング可能な最も下端の領域となるとき，すなわち

$$y_S = h \tag{3.4}$$

を満たす座標 y_P が

$$y_P = (1 - \alpha) \times h = \alpha \times h \tag{3.5}$$

となるとき，直接タッチの必要な領域の縦幅が最も小さくなる．したがって，α は式 3.5 から

$$\alpha = 0.5 \tag{3.6}$$

と定めることとした．また，TouchOver は，直接タッチが必要な領域を薄く表示しユーザにヒントを与えることとした．この領域を以降手元領域と呼ぶ．

3.1.3 TouchOver の問題点

実装を進める中で，TouchOver の設計指針を得るために実施した予備実験の際，図 3.3a にて緑色で示す困難領域が存在するという知見が得られた．TouchOver にて困難領域が存在する場合，図 3.3a にて赤色で示す領域にタッチ点を転送することが不可能となり，図 3.3b にて青色で示す，画面左端の領域に対するポインティングが不可能となる．結果，ユーザの手の大きさに対して本研究の対象となる大端末は幅が広すぎるため，TouchOver は不適当であるという結論が得られた．この問題を解決するため，タッチ点の転送領域を上領域のみから，左，左上，上の 3 領域に拡張した．転送領域を拡張した手法を TouchWarp と呼び，詳細を次節にて述べる．また，同じく操作手法である Reflection も合わせて述べる．

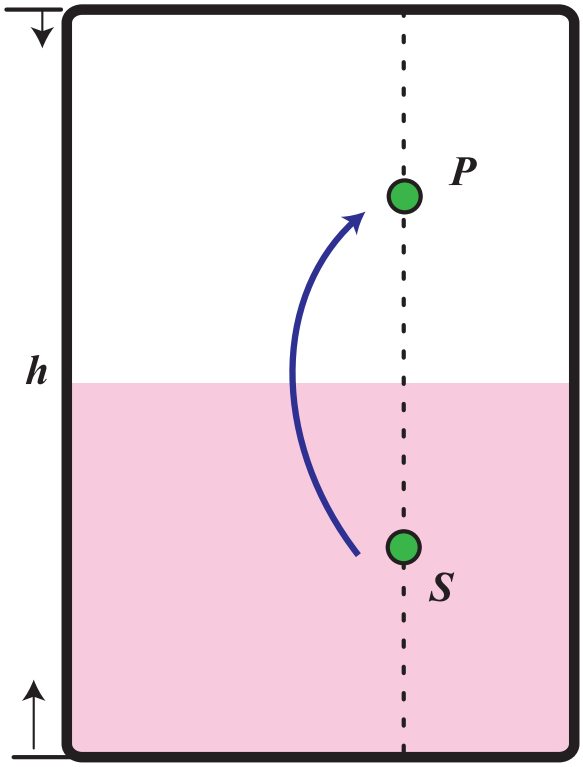

図 3.2: TouchOver におけるタッチ点及び転送後のタッチ点の位置の関係.

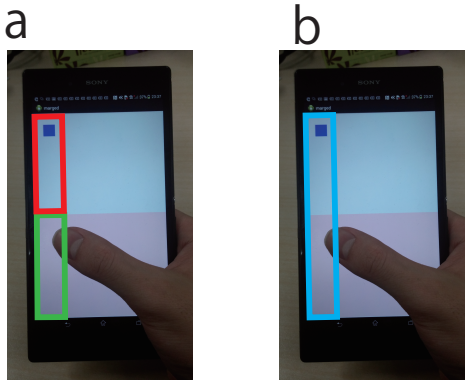

図 3.3: TouchOver の問題点．a：タッチが不可能な手元領域およびタッチ点の転送が不可能な領域． b ：ポインティングが不可能な領域.

3.2 操作手法

本節では，操作手法である TouchWarp および Reflection について述べる．

3.2.1 TouchWarp

本節では，TouchWarp の設計及び実装について述べる．

TouchWarp の設計

TouchWarp は画面領域を 4 分割し，ユーザが画面右下の領域に入力したタッチ点を，左，左上，上のいずれかの画面領域に転送する．これにより，指が届く画面右下領域における操作のみを用いて，指の届かない画面領域に存在するターゲットのポインティングを可能とする．転送対象となる画面領域の選択にはドラッグ操作を用いる．ただし，このドラッグ操作には画面領域の選択に必要な最小限の指の移動のみにて可能となるよう TouchWarp を設計し，本操作手法に最低限要する指の移動量を 3px とした．この結果，カーソルを用いる手法に比べて必要な指の移動量が減少したため，高速なポインティングが可能となり得る．なお，この移動量は著者が実験的に求めた値である．

TouchWarp は，まずユーザが画面に対して強いタッチを行い，次にドラッグ操作を用いてタッチ点の転送領域を選択し，最後にタッチアップによって困難領域をポインティングする．TouchWarp の使用例を図 3.4 に示す．ユーザはまず，図 3.4a のように画面に対して強いタッチを行う．その後ユーザは画面をタッチした指を左上方向にドラッグすることより，図 3.4b のようにタッチ点を画面左上領域に転送する．このとき，TouchWarp はユーザに対しする転送先のタッチ点のフィードバックとしてカーソルを表示する．ユーザが指をタッチ画面から離したとき，図 3.4c のように TouchWarp が終了する．

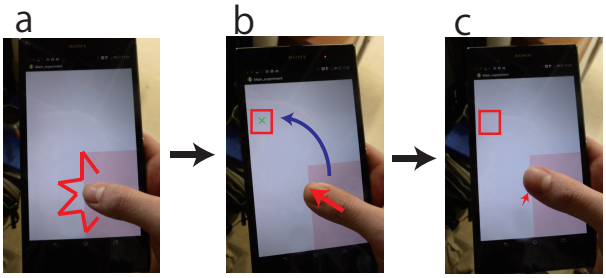

図 3.4: TouchWarp の使用例．a：Forcetouch による起動．b：ドラッグ操作による転送先の決定．c：タッチアップによる TouchWarp の解除．

TouchWarp の実装

本手法により，転送されたタッチ点の位置 $P(x_P, y_P)$ の算出法を詳述する．ユーザのタッチ点を S（x_S, y_S），画面の横方向の長さを w，縦方向の長さを h とした場合，図 3.5 に示すようにユーザのドラッグ方向に応じて以下とする．

$$(x_P, y_P) = \begin{cases} (x_S, y_S - \alpha \times h) & A_1\text{方向にドラッグ操作を行った場合} \\ (x_S - \alpha \times w, y_S - \alpha \times h) & A_2\text{方向にドラッグ操作を行った場合} \\ (x_S - \alpha \times w, y_S) & A_3\text{方向にドラッグ操作を行った場合} \end{cases}$$

TouchWarp における α の値は TouchOver 同様に 0.5 とし，また，手元領域の表示も行うこととした．

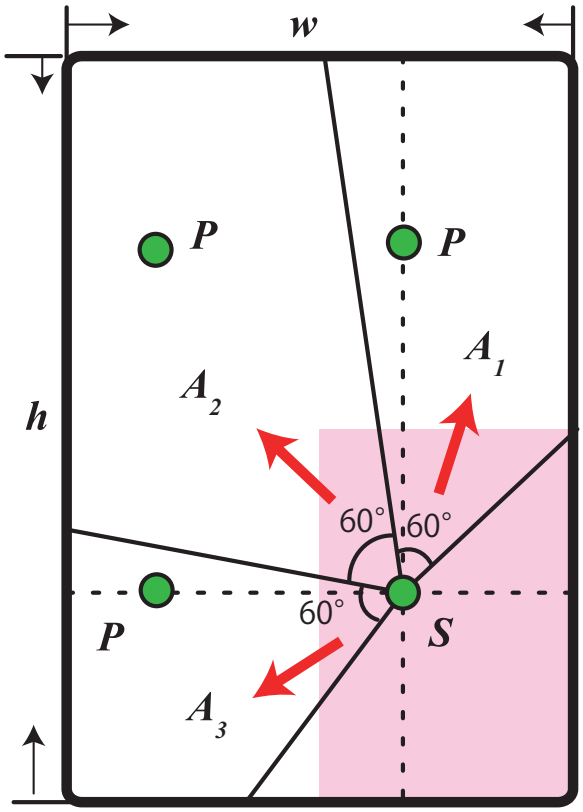

図 3.5: TouchWarp におけるタッチ点及び転送後のタッチ点の位置の関係．

3.2.2 Reflection

本節では Reflection の設計及び実装について述べる．

Reflection の設計

Reflection では画面端において反射するカーソルを表示する．このカーソルはユーザのドラッグ操作によって困難領域に移動させることができるため，ユーザは片手のみを用いた困難領域のポインティングを行うことができる．カーソルが画面端において反射するため，図 3.6a のようにカーソルがターゲットを通り越してしまった際もカーソルを引き戻さず，図 3.6b のように同じ方向の操作のみを用いて再度ターゲットをポインティングすることが可能である．

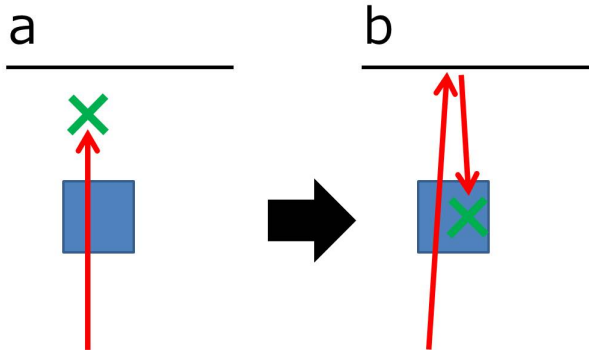

図 3.6: カーソルを引き戻さず行う再ポインティング．a：ターゲットを通り越したカーソル．b：反射を行い再度ターゲットをポインティングするカーソル．

複数回の反射を実現するため，カーソルの表示位置の演算にはレイトレーシング法を応用したアルゴリズムを用いた．すなわち，図 3.7 に示すように，初めに Reflection は，タッチ開始点とカーソルを結んだ線分（cursor-ray）の距離及び方向，cursor-ray の開始点を算出する．cursor-ray が画面端を通る線分（bezel-ray）と接触した場合，その交点を新たな cursor-ray の開始点とし，再度 cursor-ray の距離及び方向を算出する．この演算を，cursor-ray と bezel-ray が接触しなくなるまで繰り返すことにより，Reflection はカーソルの表示位置を決定する．この演算を，図 3.8 に示される例を用いて説明する．まず，図 3.8a のように，Reflection は cursor-ray の距離，方向，および開始点を算出する．このとき，cursor-ray が画面左の bezel-ray と接触している．このとき，図 3.8b のように，Reflection はこの 2 線分の交点を新たな cursor-ray の開始点とし，破線にて示される線分を画面領域方向に反転させることにより新たな cursor-ray とする．同様に，図 3.8c のように，Reflection は新たな cursor-ray およびその開始点を算出する．このとき，cursor-ray は bezel-ray と接触しないため，カーソルの表示位置が決定される．

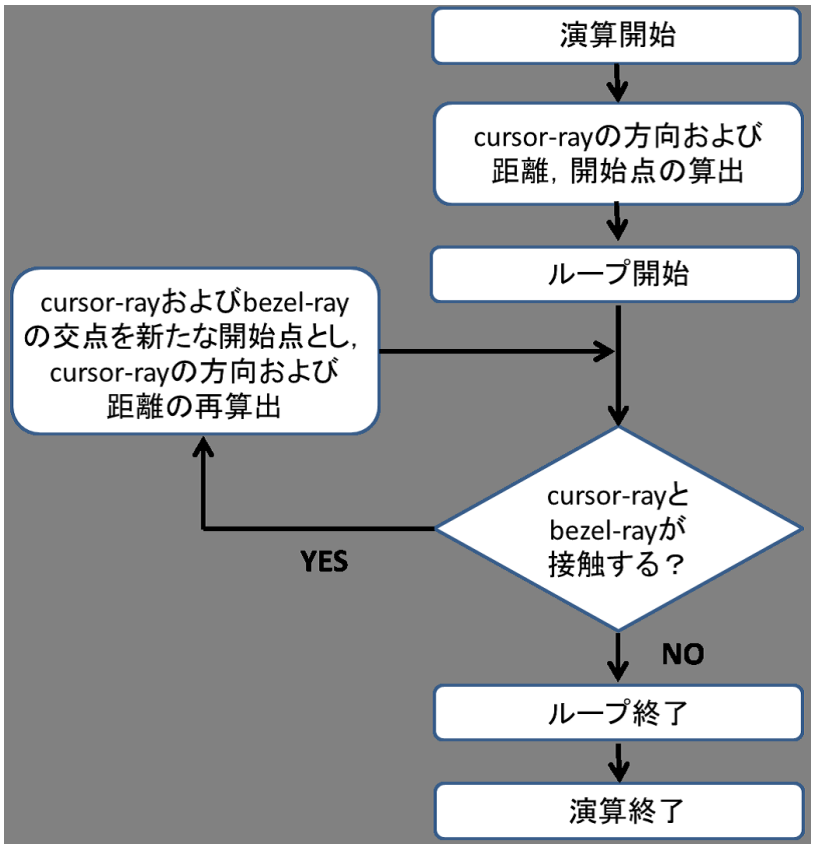

図 3.7: カーソルの表示位置演算のフローチャート.

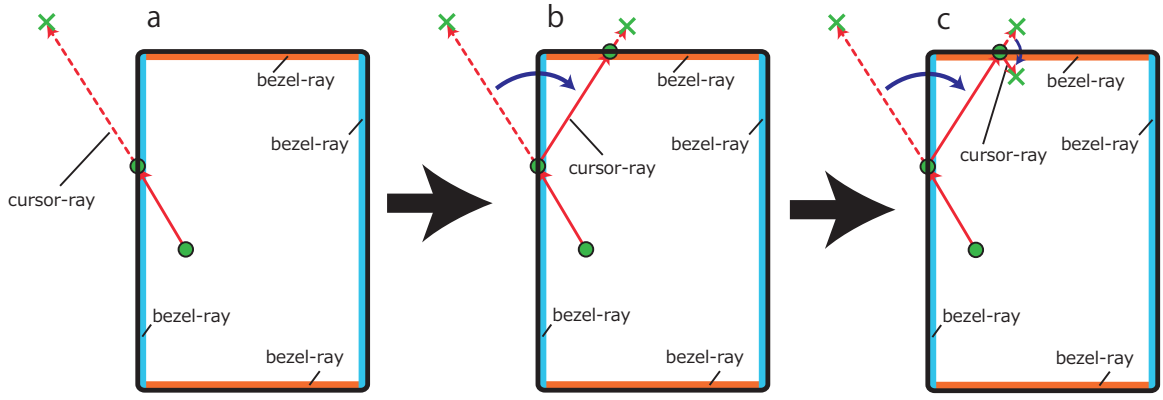

図 3.8: カーソルの表示位置演算の例.

また，このカーソルの control-display（C-D）比は，CornerSpace[YHHH13] にならい 2 とした．しかし著者が操作を行ったところ，C-D 比が大きすぎるため細かいポインティングが困難であった．そこで，大まかにターゲットの方向にカーソルを移動させたのちに細かい操作を用いてポインティングを行うことを可能とするために，C-D 比を反射後に減少させることとした．反射後の C-D 比の値は実験的に 1 とした．

Reflection の使用方法を図 3.9 に示す．ユーザはまず，図 3.9a のように画面に対して強いタッチを行う．その後ユーザは，画面をタッチした指をドラッグすることによって，図 3.9b のようにカーソルを移動させることができる．また，カーソルが画面外へ移動する操作を行った場合，カーソルは図 3.9c のように画面端において反射する．

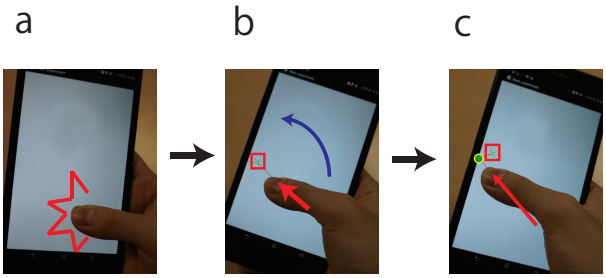

図 3.9: Reflection の使用例．a：Forcetouch による Reflection の起動．b：ドラッグ操作によるカーソルの移動．c：画面端におけるカーソルの反射．

Reflection の実装

本手法により表示されるカーソルの位置 $P(x_P, y_P)$ の算出法を詳述する．反射を行わない場合，タッチ画面に最初に触れた時のタッチ点を $S(x_S, y_S)$，ドラッグ操作後のタッチ点を $S'(x_{S'}, y_{S'})$ とすることにより，

$$(x_P, y_P) = (2 \times (x_{S'} - x_S) + x_S, 2 \times (y_{S'} - y_S) + y_S) \tag{3.7}$$

とする．また，図 3.10 における V_1〜V_4 に示される bezel-ray と SP に示される cursor-ray が接触した場合のカーソルの表示位置 $P'(x_{P'}, y_{P'})$ は，V と SP の交点を $G(x_G, y_G)$ とした場合以下となる．

V_1 または V_3 と SP が接触した場合

$$(x_{P'}, y_{P'}) = (1 \times \frac{2 \times (x_{S'} - x_S) - (x_G - x_S)}{2} + x_G, -1 \times \frac{2 \times (y_{S'} - y_S) - (y_G - y_S)}{2} + y_G) \tag{3.8}$$

V_2 または V_4 と SP が接触した場合

$$(x_{P'}, y_{P'}) = (-1 \times \frac{2 \times (x_{S'} - x_S) - (x_G - x_S)}{2} + x_G, 1 \times \frac{2 \times (y_{S'} - y_S) - (y_G - y_S)}{2} + y_G) \tag{3.9}$$

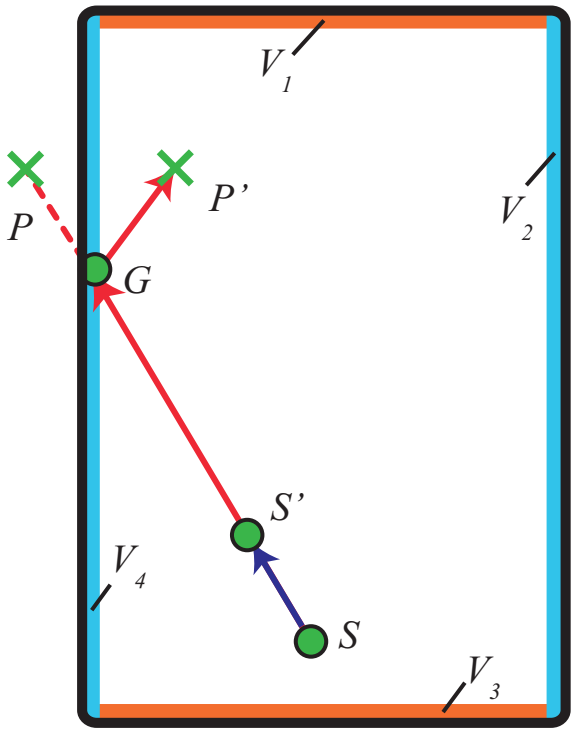

図 3.10: Reflection におけるタッチ点及びドラッグ操作後のカーソルの位置の関係

3.3 起動手法

本節では，起動手法である Forcetouch の設計及び実装について述べる．

3.3.1 Forcetouch の設計

以下に，Forcetouch の設計について述べる．

タッチの強さの検出

Forcetouch はタッチの強さを分類する手法である．その分類は，端末に組み込まれた加速度センサの値を取得し，ユーザが行ったタッチの強さを検出することにより行われる．先行研究である Forcetap[HL11b] は，画面に対するタップの強弱を分類するため，10 ms 間隔にて加速度センサの値を取得し，ユーザの指が端末の画面に接触している間に取得された値すべての合計を用いて分類を行った．一方本手法においては，続く操作手法の起動に用いるため，

ユーザのタッチが行われた瞬間にそのタッチの強弱の分類を行うこととした．また，本ポインティング手法は，起動手法である Forcetouch がタッチの強さの分類に失敗した場合，誤ったターゲットをポインティングする可能性がある．そこで分類の失敗を減らすため，あいまいな強さのタッチは分類不能とし，そのタッチによる処理を行わないこととした．

閾値の設定

[HL11b] においては，画面領域ごとにユーザのタッチによる加速度に差があると述べ，端末画面を縦 6 領域 × 横 4 領域の計 24 領域に分割し，強いタップに分類するための加速度の閾値をそれぞれの領域に定めていた．本手法もそれにならい，画面を 24 領域に分割しそれぞれの領域に閾値を設定することとした．ただし，我々が実装を進めるにあたって，タッチ時の加速度にユーザごとの個人差が見られたため，ユーザに関わらない定数の閾値を用いた場合の分類精度が低かった．したがって本手法においては，予めキャリブレーションを行い，ユーザごとに各領域に対して閾値を設定することとした．

キャリブレーション

キャリブレーションの手順を以下に示す．キャリブレーションの際，ユーザは 3 種類の強さ（以降，タッチ条件）にて 24 領域に表示されるターゲットをタッチする．3 種類タッチ条件とは，画面を強く押す程度の強さ（strong），通常のタッチを行う程度の強さ（normal），画面に触れるか触れないか程度の強さ（weak）である．ユーザにはそれぞれのタッチ条件にて，各領域に表示されるターゲットに対して 6 回ずつタッチを行ってもらった．このとき図 3.11 に示すように，画面に表示された文字とは別に，ユーザがタッチする際に求められるタッチ条件がターゲットの大きさ及び色を用いて示される．すなわち図 3.12 のターゲットは左から，strong 条件のタッチを行う際のターゲット，normal 条件のタッチを行う際のターゲット，weak 条件のタッチを行う際のターゲットである．ターゲットを表示する領域の順番はランダムとしたが，タッチ条件の提示順については strong，normal，weak の順番とした．キャリブレーションによって閾値を決定する際には strong 条件のタッチによる加速度及び weak 条件のタッチによる加速度の値のみを用いる．この理由は，キャリブレーションの設計指針を得るために実施した予備実験の際，被験者に strong 条件のタッチ及び weak 条件のタッチのみを行ってもらった際の 2 種類のタッチにおける加速度の差が小さかったことによる．そこで被験者に両条件を明確に使い分けてもらうため，タッチ条件に normal 条件を加えることとした．

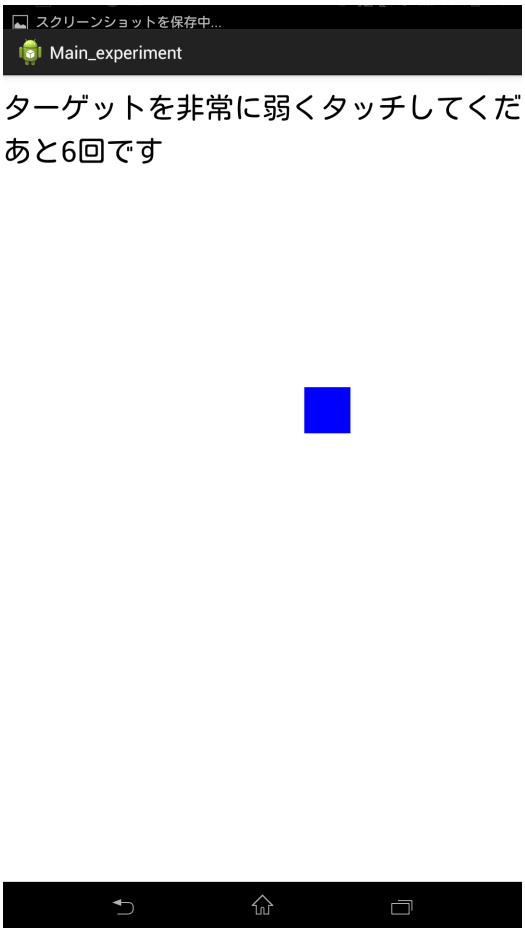

図 3.11: キャリブレーション画面.

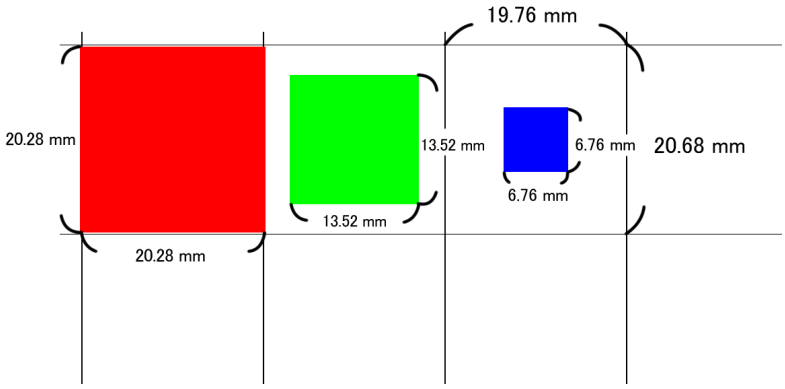

図 3.12: キャリブレーション時に使用したターゲットと分割した領域の大きさ.

3.3.2 Forcetouch の実装

以下に，Forcetouch の実装について述べる．

タッチの強さの検出

先行研究は取得された加速度の推移とタッチイベントの関係の調査を行い，以下の知見を得た．

- 加速度のピークがタッチイベントの発生前に起こっている．
- ユーザがタップを行う際，指が画面に接触している時間が平均 96 ms である．

そこで本手法においては，タッチイベントが発生する直前の加速度の値の合計を分類に用いた．これはタッチイベントの直前に加速度のピークがあるため，Forcetap と同様にタッチ時の加速度が得られると考えたためである．また加速度の取得時間は，指が画面に接触している時間と同じ 96 ms とし，本手法における加速度の取得間隔は，実験的に 5 ms とした．

本実装を行うにあたり，使用する加速度について以下の 3 種類の値を検証した．

- z 軸方向のみの絶対値
- x，y，z 軸方向の絶対値の合計
- x，y，z 軸方向の絶対値によって貼られるベクトル長

検証の結果，z 軸方向のみの絶対値を用いた場合の分類精度が最も高かったため，使用する加速度は z 軸方向のみの値とした．

閾値の算出法

閾値の算出法を示す．まず，キャリブレーションによって得られた strong 条件のタッチの加速度及び weak 条件のタッチの加速度の分布がそれぞれ正規分布であるとみなし，図 3.13 に示される両分布の交点 F を算出する．次に，分類の誤りを減らすため，両分布から得られた標準偏差 σ_S，σ_W を用いて F_S，F_W を領域毎に次式にて求める．

$$F_S = F + \beta \times \sigma_W \tag{3.10}$$

$$F_W = F - \beta \times \sigma_S \tag{3.11}$$

これらを用いることにより，あるタッチの加速度を f とした場合，そのタッチ条件は以下のように分類される．

strong 条件：f > F_S

weak 条件：f < F_W

このとき上記のどちらにも当てはまらない場合には，そのタッチによる処理を行わないようにした．なお β の値は実験から 0.5 とした．

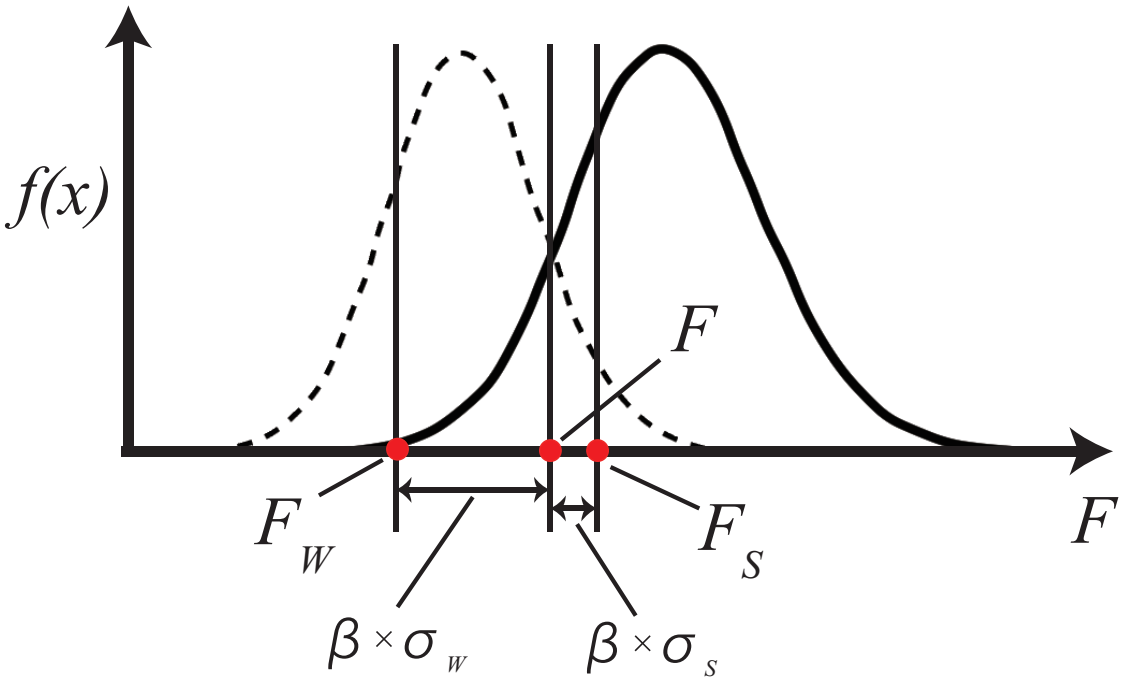

図 3.13: 閾値 F_S 及び F_W の関係.

第4章　実験1：Forcetouch予備実験

本章では，Forcetouchのキャリブレーション機能の性能評価を行うために行った予備実験について述べる．

4.1　実験設計

Forcetouchのプロトタイプを，Android 4.2.2上で動作するアプリケーションとして実装した．また実験用の大端末としてSony Xperia Z Ultra（Android 4.2.2，サイズ92.2 × 179.4 × 6.5 mm，解像度1080 × 1920 px）を用いた．被験者は22歳から24歳の大学生及び大学院生3名であり，全員右利きであった．安定した大端末把持のため，被験者には机に利き腕の肘を付けた状態にて，利き手を用いて図4.1のように大端末を把持し，その親指のみを用いて操作を行ってもらった．実験風景を図4.1に示す．この時指の届かないターゲットを選択する際には，非利き手の使用を行わせないため，利き手のみを用いて端末を回転させ，ターゲットをタッチするよう指示した．この間ユーザは非利き手を膝の上におき続けるよう指示した．

被験者にはまず，図4.2に示す24領域に対してキャリブレーションを行ってもらった．その後被験者には，図4.2の赤枠にて示される領域5，6，7，9，10，11，13，14，15，17，18，19，21，22，23に1度ずつ表示されるターゲットをstrong条件にて順番にタッチしてもらった．その後，同じ条件のターゲットをweak条件にてタッチしてもらった．このとき指示した条件にてタッチを行うことができなかった場合，その条件にてタッチに成功するまで同じターゲットに対してタッチを行い続けてもらった．被験者一人当たりの所要時間は約5分間であった．

4.2　分析項目

本実験の分析項目は分類失敗率である．分類失敗率とは分類失敗回数 / タッチ回数であり，分類失敗とは，指示と異なるタッチ条件に分類されることである．本ポインティング手法は共に，起動手法であるForcetouchに失敗した場合誤ったターゲットをポインティングする可能性がある．例えば直接タッチにてターゲットをポインティングしようとした際，誤ってTouchWarpを起動させた場合，タッチ点の転送先の領域にあるターゲットに対するポインティングを行ってしまう．したがってForcetouchの結果として示される分類成功，分類不能，分類失敗のうち，分類失敗率の分析を行う．また，分類成功率（分類成功回数 / タッチ回数）も併せて示す．

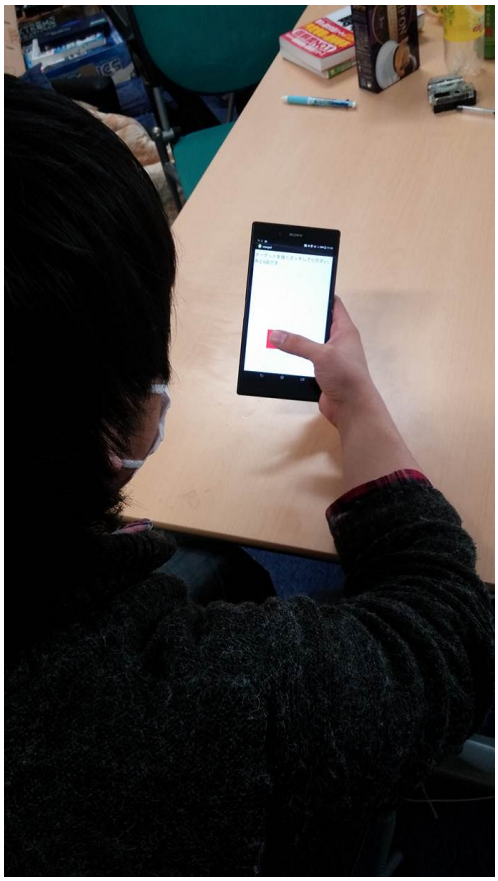

図 4.1: 実験の様子.

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15
16	17	18	19
20	21	22	23

図 4.2: 閾値設定のための分割領域.

4.3 結果

strong 条件におけるターゲットごとの分類成功率および分類失敗率を図 4.3 に，weak 条件におけるターゲットごとの分類成功率および分類失敗率を図 4.4 に，被験者ごとの分類失敗率を図 4.5 に，それぞれ示す.

strong 条件に関しては領域 17 の分類成功率が最も低く，また領域 17 の分類失敗率が最も高かった．また，weak 条件に関しては領域 22 の分類成功率が最も低く，領域 15 の分類失敗率が最も高かった.

被験者ごとの分類失敗率を領域全体の平均としてみた場合，個人差はあったが，平均すると strong 条件においては 5.46%であり，weak 条件においては 6.18%であった.

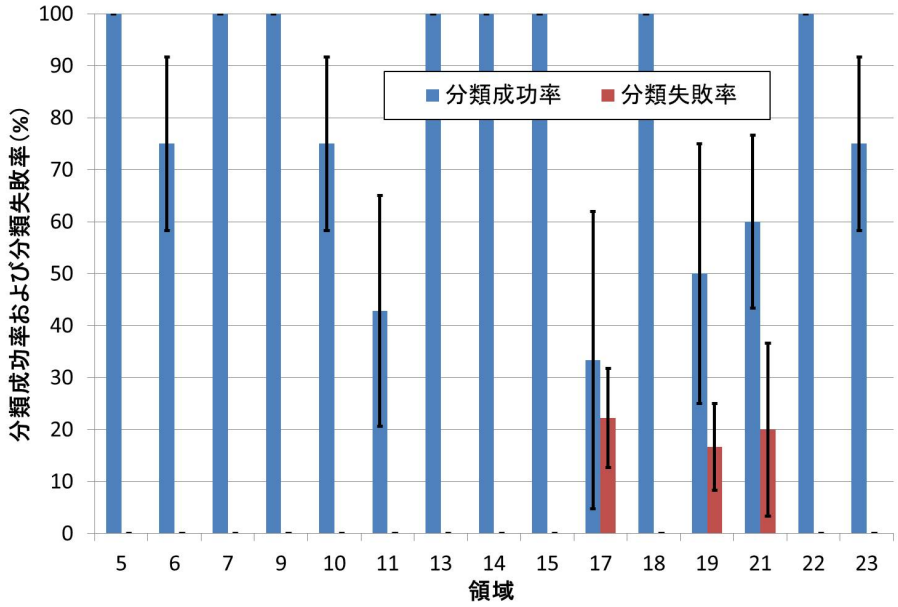

図 4.3: 予備実験結果：strong 条件の分類成功率および分類失敗率.

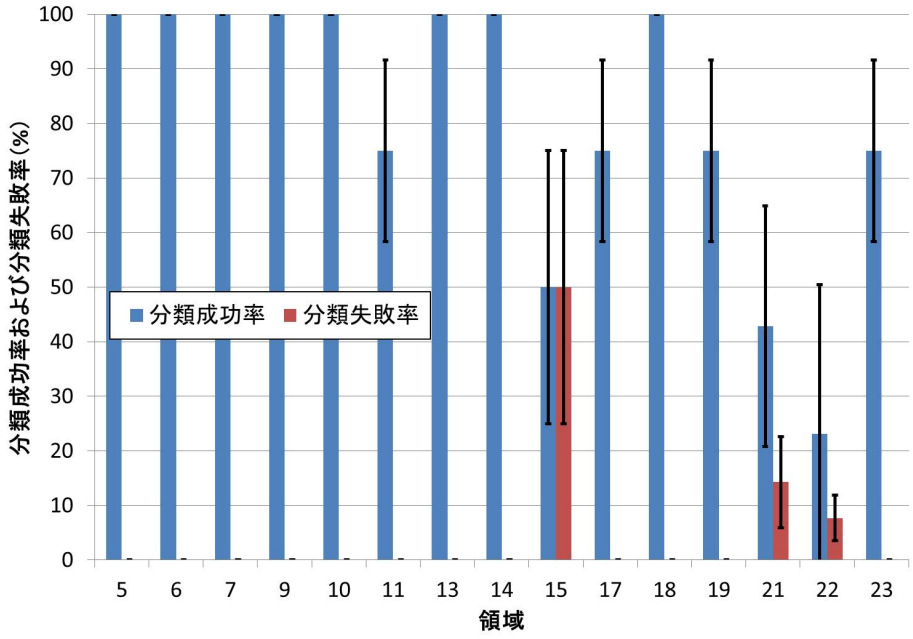

図 4.4: 予備実験結果：weak 条件の分類成功率および分類失敗率.

4.4 考察

上述のように，分類失敗率は領域ごとに差が見られた．最も分類失敗率の高い領域 15 に関して，キャリブレーションから得られた閾値を比較したところ，領域全体の閾値 F_S の平均値

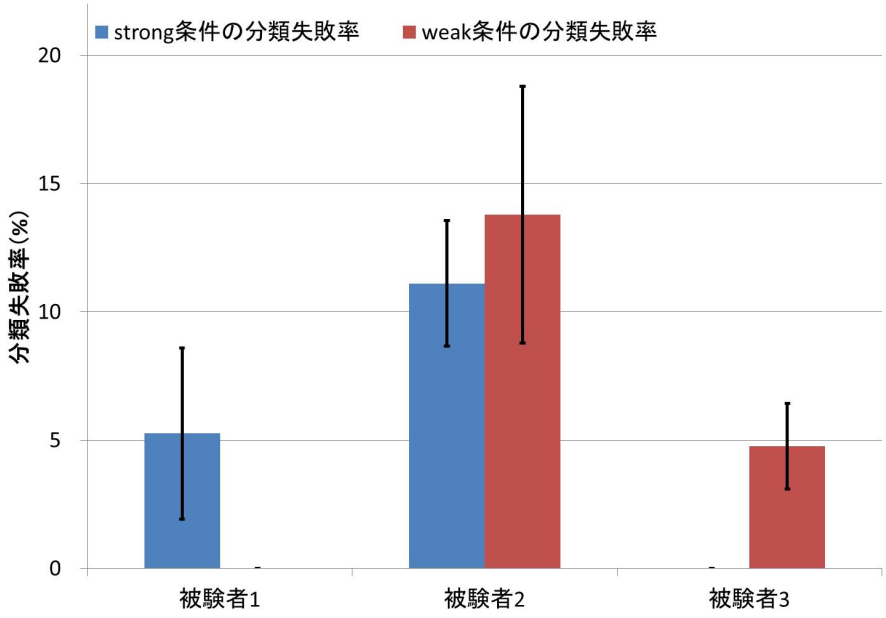

図 4.5: 予備実験の結果：被験者ごとの分類失敗率.

は 14.50 であり，閾値 F_W の平均値は 8.52 であるのに対し，領域 15 における F_S の平均値は 9.39 であり，F_W の平均値は 5.29 であった．F_S と F_W の差は全領域の平均値が 6.98 であるのに対し領域 15 が 4.10 と小さいことから，領域 15 は被験者が端末を把持する手の親指の付け根に近いため，タッチの際に力の使い分けが困難であるためと考えられる．したがって，今後閾値決定の際に用いる値 β，もしくは式そのものについての再検討を行う．

しかし，領域全体の平均を見た場合，strong 条件における分類失敗率は 5.46%であった．また weak 条件における分類失敗率は 6.18%であった．これらの識別精度と Forcetap の識別精度を比較するため，Forcetap の平均分類失敗率 5.83 %を用いて 1 サンプル t 検定を行った．その結果，strong 条件および Forcetap の間に有意差が見られなかった（$p > .10$）．また，weak 条件および Forcetap の間にも有意差が見られなかった（$p > .10$）したがって Forcetouch は Forcetap と同程度の識別精度を持つと言える．

第5章　実験2：被験者実験

本章では，提案手法と，直接タッチおよび先行研究である CornerSpace の性能とを比較するための被験者実験について述べる．

5.1　実験設計

提案手法のプロトタイプを，Android 4.2.2 上で動作するアプリケーションとして実装した．実験用大端末として Sony Xperia Z Ultra（Android 4.2.2，サイズ 179.4 × 92.2 × 6.5 mm，解像度 1080 × 1920 px）を用いた．被験者は 20 歳から 24 歳の大学生及び大学院生 8 名であり，全員右利きであった．被験者には予備実験と同様の条件にて端末を操作してもらった．実験時の様子を図 5.1 に示す．

図 5.1: 実験時の様子．

実験内容はターゲットのポインティングタスクである．この実験に用いるターゲットの大きさは Android 端末におけるアイコンの大きさと同じ 96 px × 96 px とした．

本実験は，独立変数の 1 つとしてダミー条件を設けた．これは，ポインティング目標として表示されるターゲットの他に，ポインティングしてはいけないダミーターゲットを表示させるか否かという条件である．以降，ダミーターゲットを表示するタスクをダミー有り条件，表示しないタスクをダミーなし条件と呼ぶこととする．ダミーターゲットは，例えば誤って押してはいけないボタンや選択してはいけないアイコンを模したものであり，被験者の意図しないターゲットに対するポインティングの分析に用いた．

実験終了後，各被験者に実験についてのアンケートを行った．被験者には実験終了後に謝礼として固定報酬 820 円を渡した．被験者 1 人あたりの実験時間は約 1 時間であった．

5.2 実験手順

被験者にはまず，予備実験同様に Forcetouch の閾値を決めるためのキャリブレーションを行ってもらい，その後画面上に表示されるターゲットのポインティングタスクを行ってもらった．このタスクにおいて被験者は，指示された手法条件に従い画面上の目標ターゲットをポインティングする．この手法条件は，Touch 条件，CornerSpace 条件，Forcetouch & TouchWarp 条件，Forcetouch & Reflection 条件の 4 種である．Touch 条件においては，被験者は直接タッチのみを用いて目標ターゲットのポインティングを行う．その他の手法条件においては，被験者は指示された手法及び直接タッチを任意に使い分け，目標ターゲットのポインティングを行う．このときターターゲットは Forcetouch の閾値設定に用いた 24 領域それぞれの中心に配置され，ランダムにひとつが目標ターゲットとして青く表示される．ダミーなし条件において画面上に表示されるのは目標ターゲットのみであるが，ダミー有り条件においては残りのターゲット 23 個がすべて灰色のダミーターゲットとして表示される．図 5.2，5.3 にダミーなし条件，ダミー有り条件のそれぞれの様子を示す．

タスクは被験者が待機画面に対してタッチを行い，タッチに用いた指が画面から離れたときに開始される．被験者は目標ターゲット 24 個をすべてをポインティングすることによって 1 回のタスクを完了する．ただしダミー有り条件において，被験者がダミーターゲットをポインティングした場合，新たな目標ターゲットが表示される．このタスク間に，被験者は任意に休憩を取ることができた．また 2 度の本番タスクを行う前に，被験者には本番と同じ条件で 1 タスク分の練習タスクを行ってもらった．

以上より本実験の独立変数は，手法条件（4），ダミー条件（2），タスク（2），ターゲット数（24）であり，ポインティング回数は独立変数の積である $4 \times 2 \times 2 \times 24 = 384$ 回である．なお，学習効果を打ち消すため，手法条件が指示される順番を被験者ごとにランダムとした．また，ダミー条件の順番をすべてダミー有り条件，ダミーなし条件の順とした．被験者は 2 回の本番タスクを行う前に，本番タスク 1 回に相当する練習タスクを行った．

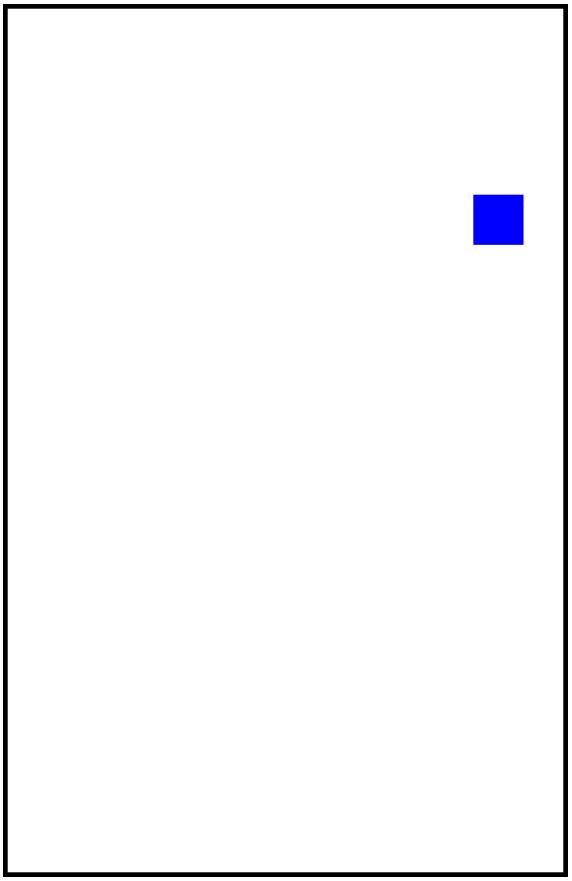

図 5.2: ダミーなし条件.

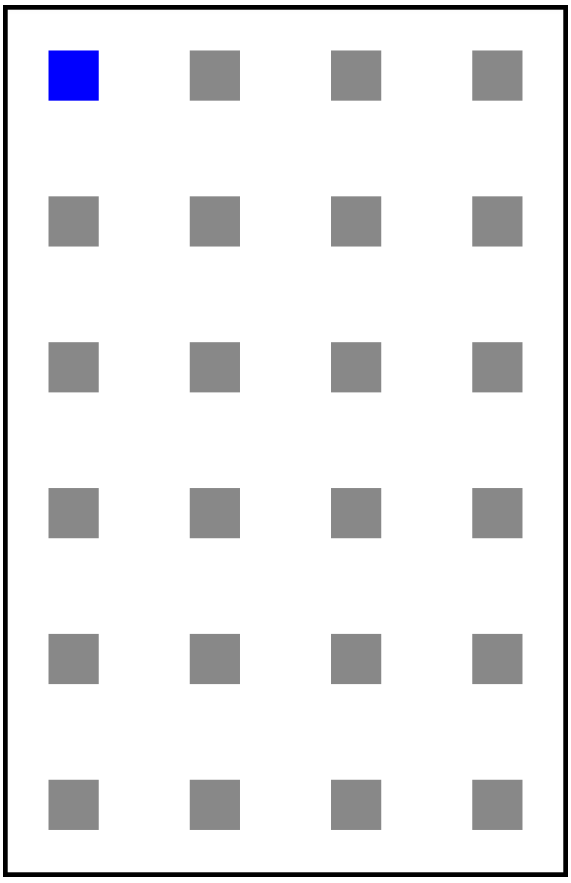

図 5.3: ダミー有り条件.

5.3 分析項目

本実験の分析項目を以下に示す.

所要時間

1 回のタスクに要する時間

失敗率

(タッチ回数 - ターゲット数) /タッチ回数

理想起動環境における失敗率

Forcetouch が分類失敗したポインティングを除いた失敗率

= （(タッチ回数 - Forcetouch が分類失敗したタッチ回数） - ターゲット数） / （タッチ回数 - Forcetouch が分類失敗したタッチ回数）

ダミーポインティング

ダミーターゲットに対するポインティング

まず，ポインティング手法としての性能評価を行うため，所要時間および失敗率に関する分析を行う．次に，操作手法の性能評価を行うため，理想起動環境における失敗率の分析を行う．最後に，ダミーターゲットに対するポインティングについて分析する．これにより，ダミーターゲットに対するポインティングの原因を特定する．

5.4 結果：所要時間および失敗率

1 タスクの所要時間の平均値及び平均失敗率をそれぞれ図 5.4，5.5 に示す．

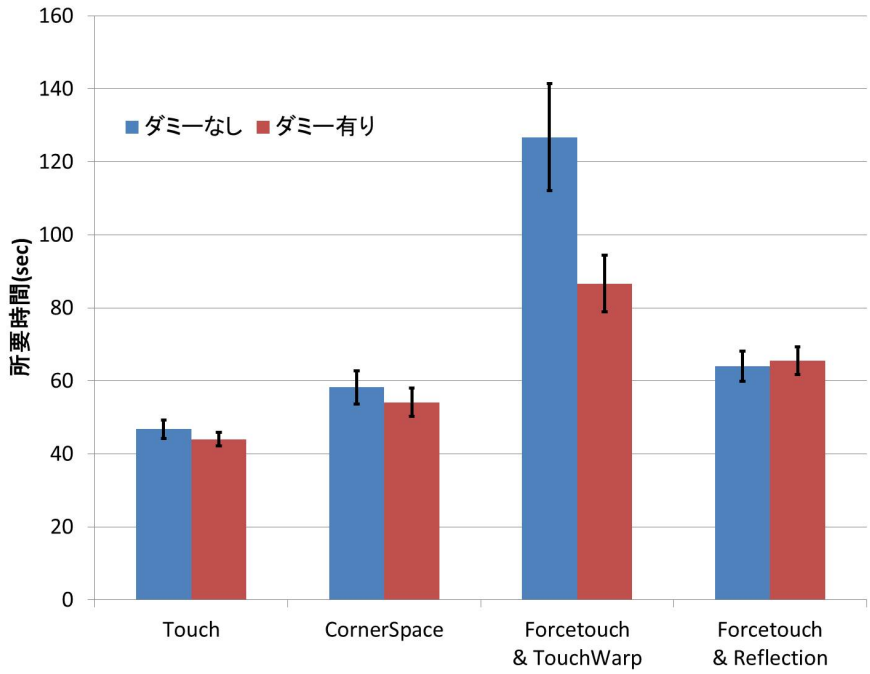

図 5.4: 実験結果：1 タスクの所要時間．

分散分析の結果 Forcetouch & TouchWarp 条件は，他の手法条件と比べ所要時間が有意に長いことがわかった（$p < .05$）．また失敗率に関しても，Forcetouch & TouchWarp 条件は他の手法条件に対し有意に失敗率が高かった（$p < .05$）．

両結果に対して多重検定を行い交互作用を調べたところダミー条件において交互作用が認められ，ダミー有り条件における Forcetouch & TouchWarp の所要時間はダミーなし条件に比べて有意に短く（$p < .05$），失敗率は有意に低かった（$p < .05$）．また，手法条件について交互作用が認められ，ダミー有り条件においては Forcetouch & Reflection 条件及び Forcetouch & TouchWarp 条件における失敗率に有意差が認められなかった（$p > .10$）．

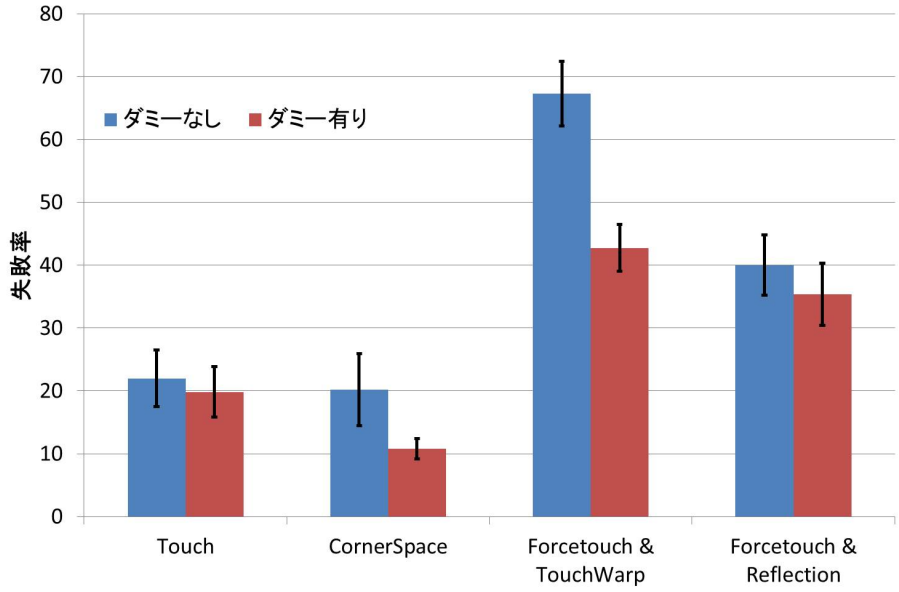

図 5.5: 実験結果：失敗率.

5.5 所要時間および失敗率の考察

ダミー有り条件にて Forcetouch & TouchWarp 条件の失敗率が低下し，所要時間が減少している．この理由は，ダミーとして表示されたターゲットがタッチ点の転送先のヒントとして機能しているからと考えられる．したがって Android 端末におけるホーム画面など，十分なヒントの存在する環境における Forcetouch & TouchWarp は有用であると考えられる．

また，所要時間が長く，失敗率が高かった Forcetouch & TouchWarp 条件および Forcetouch & Reflection 条件について，実験後のアンケートに「Forcetouch が使いづらかった」とのコメントが多く見られた．したがって，失敗率および所要時間が増加した理由は Forcetouch の分類が成功しなかったことにあると考えられる．これを確かめるため，タッチ条件の分類不能率を調べた．結果を図 5.6 に示す．分散分析の結果，ダミー条件にかかわらない，手法条件間の平均値における有意傾向が見られた（$p < .10$）．これは，Reflection は任意のタッチ画面領域からタッチ画面全体に対してポインティングを行うことができるため，被験者はよりタッチ条件を使い分けやすい領域にて Forcetouch を行ったためと考えられる．

5.6 理想起動環境における失敗率に基づく分析

Forcetouch の性能によらない提案操作手法の性能を比較するため，Forcetouch が分類失敗したポインティングを除いた理想起動環境を設定し分析を行った．理想起動環境における失敗率を図 5.7 に示す．ダミー有り条件において，Touch 条件，Forcetouch & TouchWarp 条件，

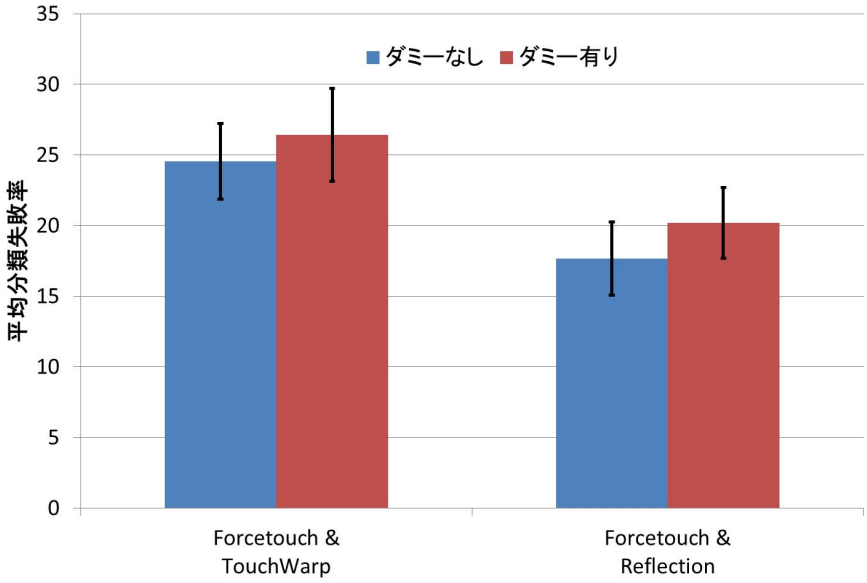

図 5.6: 実験結果：分類不能率.

Forcetouch & Reflection 条件の失敗率の差が 3%以下となった.

分散分析を行い多重検定を用いて交互作用を調べたところ，ダミーあり条件において，各手法間に有意差が認められなかった（$p > .10$）．したがって，タッチ転送先のヒントが存在する環境において，Forcetouch の分類が成功するならば提案ポインティング手法は直接タッチと同程度の精度にてポインティングが可能であるといえる.

5.7 ダミーポインティングについての分析

ダミーターゲットに対するポインティングの原因を特定するため，ダミーポインティング率の分析を行った．これはダミーターゲットに対するポインティング回数 / ターゲット数によって求められる．ダミーポインティング率を図 6.5 に示す．TouchWarp のダミーポインティング率が最も高く，他手法に比べ約 5 倍となっている.

Forcetouch & TouchWarp 条件におけるダミーポインティングのうち，約 85%が weak 条件によるものであることがわかった．すなわちこのダミーポインティングは，TouchWarp の起動に失敗したためタッチ点を転送することができず，直接タッチによってタッチ点の転送先のヒントとなっているダミーターゲットがポインティングされていたことよるものである．この問題は起動手法の分類精度を向上させることによって解決されると考えられる.

また実験時の様子から，正しく起動を行ったにも関わらずダミーターゲットをポインティングしている場合も見られた．現在，TouchWarp によってタッチ点を移動する領域は，タッチ開始時およびタッチアップ時の座標間の関係によって決定される．したがってタッチアッ

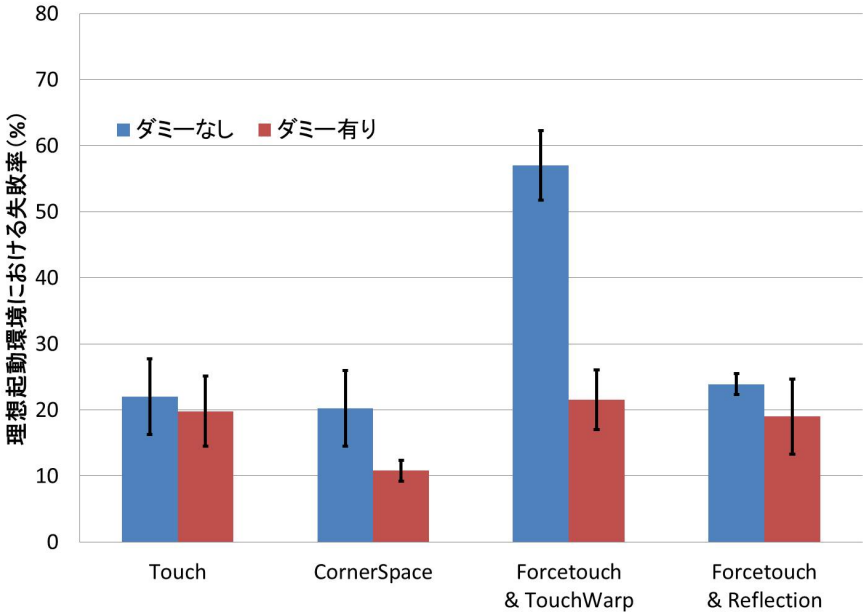

図 5.7: 実験結果：理想起動環境における失敗率.

プ時に指が画面上を移動した場合，タッチ点の転送領域が被験者の意志と関係ない方向に定まる．これにより，ダミーターゲットのポインティングが行われていたことが，実験時の様子から観察された．この問題は，転送先の決定方法を見直すことにより解決されると考えられる．そのほか，誤ったタッチ点転送先のヒントを用いて Forcetouch & TouchWarp を起動し，ダミーターゲットをポインティングする場合もあった．したがって，タッチ点転送先のヒントについても今後検討を行う．

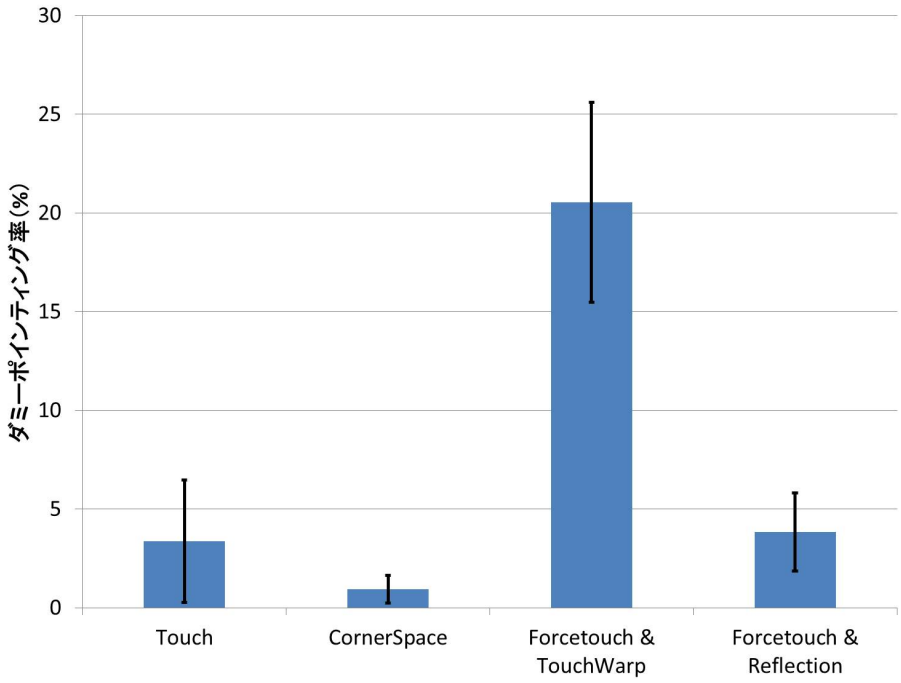

図 5.8: 実験結果：ダミーポインティング率.

第6章　実験3：TouchOverを用いた性能比較予備実験

本章では，プロトタイプとして実装したTouchOverの性能評価を行った予備実験について述べる．この予備実験の設計は，5章にて述べた被験者実験とほぼ同じ内容のものである．

6.1　実験設計

Forcetouch & TouchOverを，Android 4.2.2上で動作するアプリケーションとして実装した．Forcetouch & TouchOverを用いるため，実験用の端末には横方向に対して直接タッチが行え，かつ縦方向には直接タッチが行えない端末（以降，中端末）が適当である．そこで実験用の端末として，G2 L-01F（Android 4.2.2，サイズ 139 × 71 × 9.2 mm，解像度 1080 × 1920 px）を用いた．被験者は著者1名であり，右利きであった．被験者には予備実験と同様の条件にて端末を操作した．実験手順は被験者実験と同様のものであったが，手法条件は，Touch条件，CornerSpace条件，Forcetouch & TouchOver条件，Forcetouch & Reflection条件の4種とした．手法の順番は，ランダムに決定した結果，Touch条件，CornerSpace条件，Forcetouch & Reflection条件，Forcetouch & TouchOver条件の順となった．実験時間は約30分であった．

6.2　分析項目

本実験の分析項目は，被験者実験と同様に以下のものである．

- 所要時間
- 失敗率
- 理想起動環境における失敗率
- ダミーポインティング

6.3　結果：所要時間および失敗率

1タスクの所要時間および失敗率を，それぞれ図6.1，6.2に示す．所要時間に関しては，実験2に比べて全体的に短くなっている．また，CornerSpaceおよびForcetouch & TouchOver条

件間の差が 0.45 秒となっており，ダミー有り条件に関しては Forcetouch & TouchOver 条件の所要時間の方が短い．失敗率に関しても同様に，実験 2 に比べて全体的に減少している．また，CornerSpace および Forcetouch & TouchOver 条件間の差が実験 2 に比べて小さくなっている．Forcetouch & Reflection 条件に関しては，他の手法条件に比べて相対的に所要時間が長くなり，失敗率は増加している．

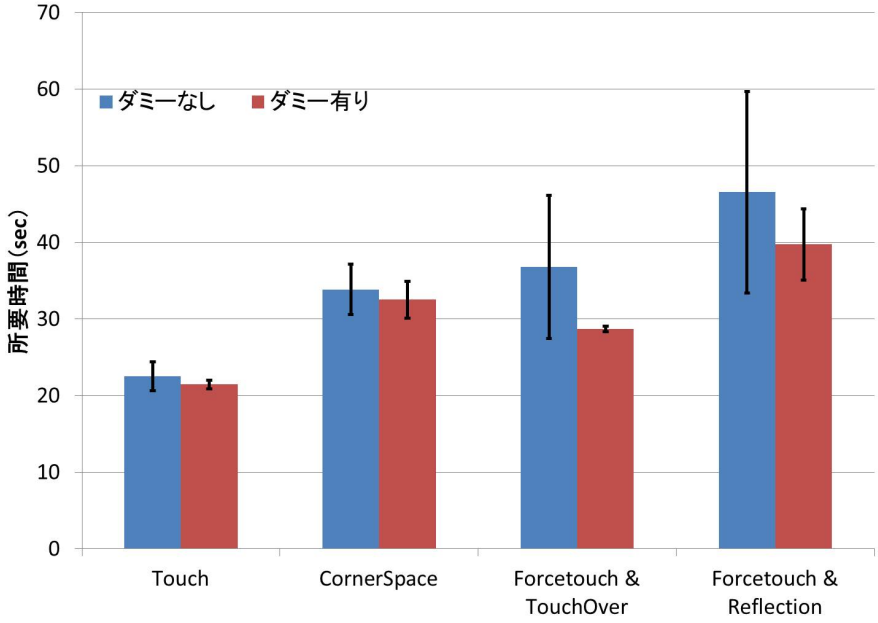

図 6.1: 実験結果：1 タスクの所要時間．

6.4 所要時間および失敗率の考察

所要時間の全体的な減少に関しては，端末の大きさが小さくなったため，端末を持ち替えることなく直接タッチできるターゲットが増加したためと考えられる．CornerSpace および Forcetouch & TouchOver 条件間の結果については，上記の理由に加え，著者が実験 2 の被験者に比べて Forcetouch & TouchOver に習熟しているため，相対的に Forcetouch & TouchOver の結果が向上したと考えられる．

CornerSpace は 2 段階の操作を用いるため，ポインティングに多くの時間を要する．一方，Forcetouch & TouchOver は一連のタッチダウンおよびタッチアップのみを用いるため，わずかな時間でポインティングを行うことが可能である．したがって，特にタッチ点転送先のヒントが存在する環境を模したダミー有り条件においては，Forcetouch & TouchOver 条件は CornerSpace

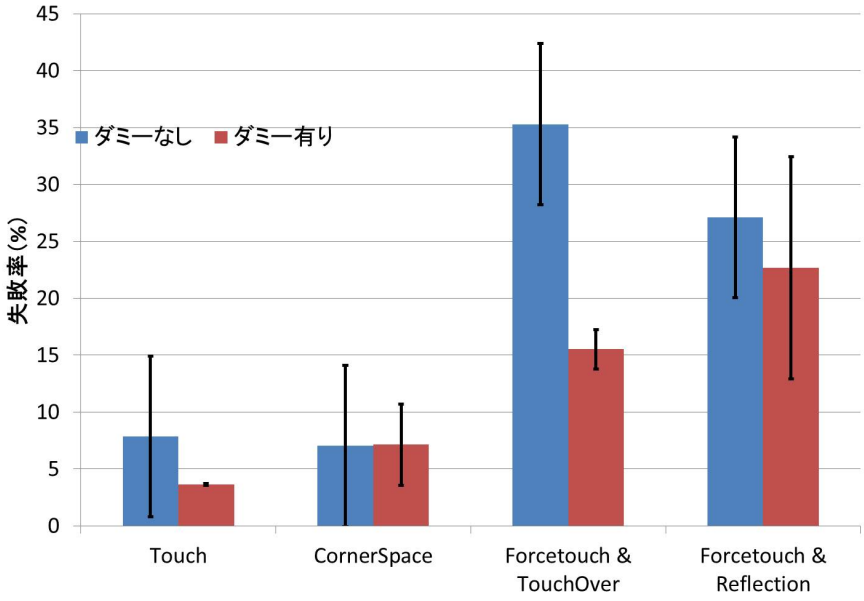

図 6.2: 実験結果：失敗率.

条件よりも素早くポインティングを行うことができたと考えられる.

また，相対的に所要時間および精度が低下した Forcetouch & Reflection 条件に関しては，実験中起動の失敗が多くみられた．この原因を特定するため，タッチ条件の識別不能率を調べた．結果を図 6.3 に示す．図 6.3 より，失敗率の高いダミーなし条件における分類不能率が低いため，多くの分類失敗が起きていることが読み取れる．これを確かめるため，6.5 節にて理想起動環境における失敗率の分析を行う.

また，被験者実験と同様，6.6 節にてダミーポインティング率の分析を行う.

6.5　理想起動環境における失敗率に基づく分析

理想起動環境における平均失敗率を図 6.4 に示す．Forcetouch & Reflection の失敗率が最も低い結果となった．これにより，先行研究である CornerSpace よりも高いポインティング精度が示された.

CornerSpace の失敗率が高い原因として，ベゼルスワイプの失敗が挙げられる．また，Forcetouch & Reflection の失敗率が低い原因として，被験者である著者は反射による C-D 比の減少を多用したことが考えられる．したがって今後は反射を容易に用いることができるよう Forcetouch & Reflection の設計を改良する．この改良には，C-D 比を増加させ，よりカーソルが画面端に届きやすくすることが考えられる.

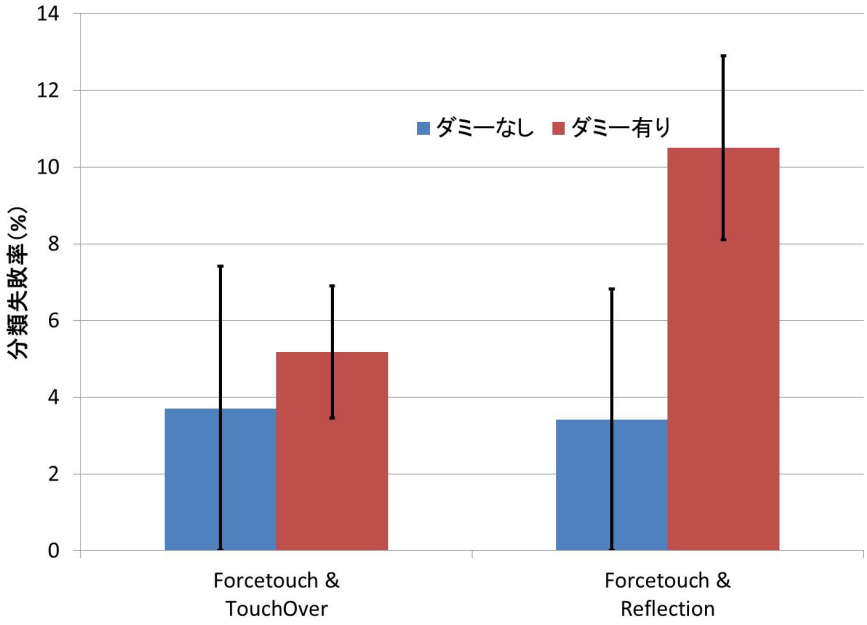

図 6.3: 実験結果：分類不能率.

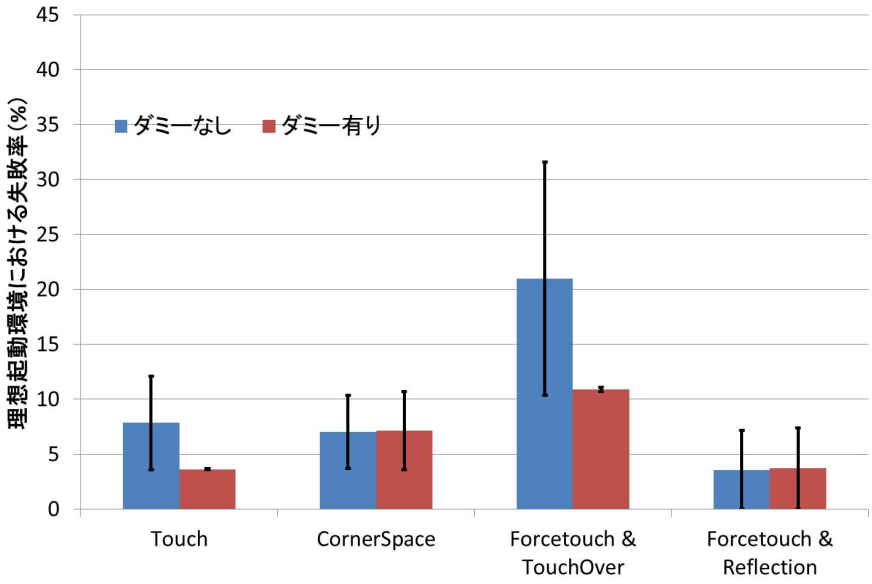

図 6.4: 実験結果：理想起動環境における失敗率.

6.6　ダミーポインティングについての分析

ダミーポインティング率を図 6.5 に示す．実験 2 と同様，Forcetouch & TouchOver 条件のダミーポインティング率が最も高くなっている．

実験 2 と同様に，TouchOver の起動に失敗したため，タッチ点転送先のヒントに対するポインティングが原因の多くを占めている．したがって，起動手法の見直しを行い，分類精度の向上を図る．

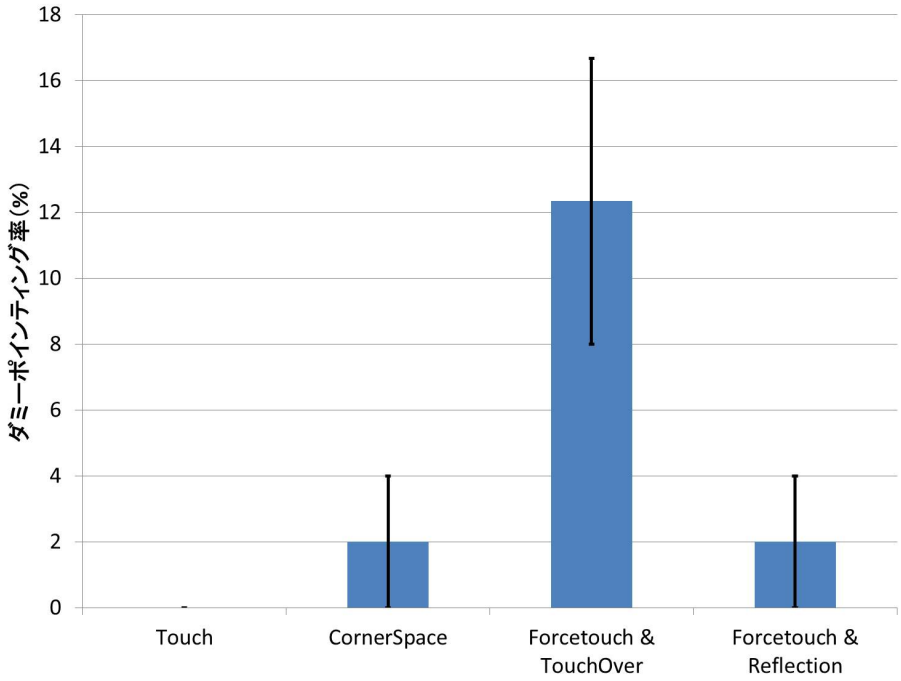

図 6.5: 実験結果：ダミーポインティング率.

第7章　結論および今後の課題

本研究では，大型のタッチ画面を備える携帯情報端末における片手操作手法の提案を目的とし，起動手法及び操作手法からなるポインティング手法を 2 種類提案した．この提案手法を Android アプリケーションとして実装を行い，その性能を評価するための被験者実験を行った．具体的には，既存の操作手法である直接タッチおよび先行研究である CornerSpace を比較対象とし，ポインティングタスクを行った．実験の結果，本実験に用いた起動手法に問題があることが分かった．しかし，起動手法が安定した場合直接タッチと同程度の精度が出る可能性が示された．また，TouchWarp のプロトタイプとして実装した TouchOver の性能を比較するため，著者による予備実験を行った．結果，先行研究よりも素早いポインティングを行うことが可能だという知見が得られた．

また，実験より得られた知見をもとに，2 種類の操作手法，起動手法，TouchOver についてそれぞれの課題を示す．

7.1　操作手法（TouchWarp）

今後の課題は，タッチ点の転送後の操作の実装である．現在はタッチ点を転送した後，ユーザの指と連動したタッチアップイベントを発生させるのみであり，タッチ点の転送領域の決定がタッチアップ時に行われている．この実装が，ポインティングタスクにおける失敗の原因の 1 つとなっていた．したがって今後は，タッチ点の転送方向を決定する最低限のドラッグ操作を行った後の操作はすべて転送領域に対するイベントとして処理するよう実装を変更する．これにより，タッチアップ時にタッチ点の転送領域が誤って決定される現在の問題が解消され，ポインティング精度の向上が考えられる．

また，誤ったヒントを用いたポインティングを無くし，同時に，ヒントの示されない環境における精度を向上させるため，TouchWarp によるヒントの提示を検討する．これにより，ユーザはヒントのない環境においても，TouchWarp を用いて正確なポインティングを行うことができると考えられる．ヒントの例としては，ホバー機能を持つ大端末上に TouchWarp を実装し，ユーザのホバーを用いて，転送先のタッチ点のフィードバックを与える方法が考えられる．このフィードバックを図 7.1 に示す．ホバーにより取得された座標をタッチ予想座標として赤く表示し，その座標から転送可能なタッチ点のフィードバックを緑色に表示している．このときユーザは，ターゲットに最も近いカーソルをホバー状態にてターゲットに合わせ，そのまま画面に対して Forcetouch を行う．これにより TouchWarp を起動したのち，ユーザはドラッグ操作にて目的のタッチ点転送領域を決定することによってポインティングを行うこと

が可能となる．

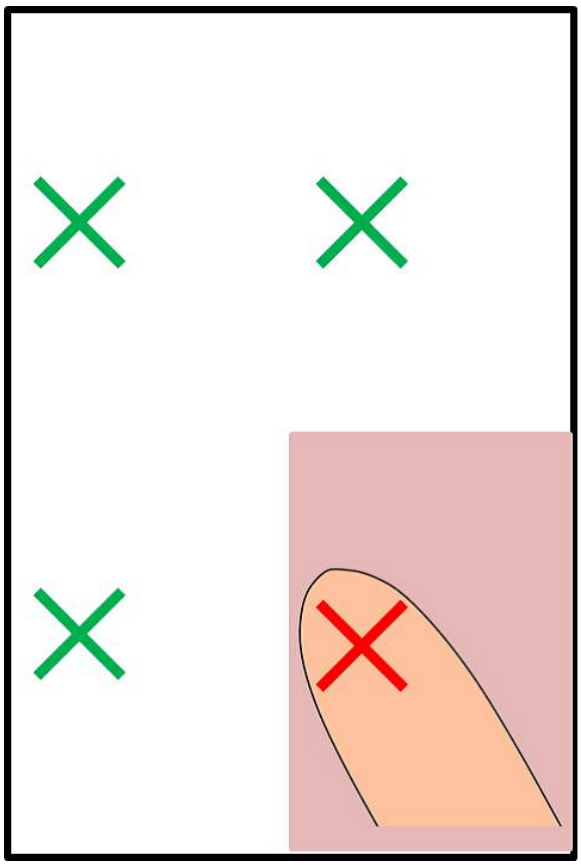

図 7.1: 転送先のヒント案．

このヒントを実装した場合，ユーザは TouchWarp を Ninja Cursors のように用いることができる．したがって，困難領域に対して，カーソルを用いた手法と同程度の精度をもつ，TouchWarp を用いた素早いポインティングが可能となる．

7.2 操作手法（Reflection）

今後の課題は，より適した C-D 比の設定である．実験 2 のアンケートにおいて，Reflection の反射機能を用いた被験者はいなかった．これは C-D 比が小さすぎるため，カーソルが画面端まで移動することがなかったためであると考えられる．また，アンケートの中に，カーソルの C-D 比が足りないという意見が見られたことからも，本研究の実装に用いた C-D 比は小さすぎたことがわかる．したがって本研究の実装よりもより大きな C-D 比を用いることにより，より少ない指の移動を用いたポインティングを実現すると同時に，反射を用いる新たなポインティングの提案を行う．

7.3 起動手法

今後の課題は，Forcetouch における閾値算出方法の再考である．第 4 章にて述べた実験 1 にて，ユーザがタッチを使い分けるのが困難な領域が存在するという知見が得られた．したがって今後は，領域ごとのタッチの強さをより詳しく分析し，いずれの領域においても安定して起動する閾値の設定を行う．例えば，閾値を求める際に用いた式 3.10，3.11 における β の値を，定数から領域に応じた変数とすることである．これにより，閾値 F_S および F_W の値をより分類精度が上がるように設定する．また，式自体の見直しも検討する．

7.4 TouchOver

今後の課題は，実験 3 を複数の被験者にて行うことである．実験 3 にて，TouchOver が中端末向けの操作手法として適している可能性が示されたため，より詳しい評価を行う．また TouchWarp 同様，誤ったヒントを用いたポインティングを減らすため，TouchOver によるヒントの提示を検討する．

謝辞

本論文の執筆に当たり，志築文太郎先生をはじめ，高橋伸先生，田中二郎先生，三末和男先生には多くのご助言やご指導を頂きました．特に志築文太郎先生には，研究の方針，論文の執筆，そして私生活に至るまで，厳しくも丁寧なご指導を頂きました．心より感謝いたします．また，インタラクティブプログラミング研究室の皆様にも，大変お世話になりました．日頃から貴重な助言やご意見を頂きました．ここに深く感謝いたします．特に WAVE チームの皆様には，チームゼミでの意見のみならず研究生活全体に渡っても多くのご意見を頂きました．ここに深く感謝いたします．本研究に欠かすことのできないすべての実験に協力してくださった被験者の皆様に深く感謝いたします．友人たちには，連日にわたって多くの励ましを頂き，精神面で大きく支えていただきました．心より感謝いたします．最後に，様々な面で私を支えてくださった家族に，深く感謝致します．皆様へ心から感謝の気持ちと御礼を申し上げたく，謝辞にかえさせていただきます．

参考文献

[ALMP09] Martha Abednego, Joong-Ho Lee, Won Moon, and Ji-Hyung Park. I-Grabber: expanding physical reach in a large-display tabletop environment through the use of a virtual grabber. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '09, pp. 61–64, New York, NY, USA, 2009. ACM.

[BHOH11] Tom Bartindale, Chris Harrison, Patrick Olivier, and Scott E. Hudson. SurfaceMouse: supplementing multi-touch interaction with a virtual mouse. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*, TEI '11, pp. 293–296, New York, NY, USA, 2011. ACM.

[GWP12] Mayank Goel, Jacob Wobbrock, and Shwetak Patel. GripSense: Using Built-in Sensors to Detect Hand Posture and Pressure on Commodity Mobile Phones. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, pp. 545–554, New York, NY, USA, 2012. ACM.

[HCD11] Stéphane Huot, Olivier Chapuis, and Pierre Dragicevic. TorusDesktop: Pointing via the Backdoor is Sometimes Shorter. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pp. 829–838, New York, NY, USA, 2011. ACM.

[HL11a] Seongkook Heo and Geehyuk Lee. Force Gestures: Augmented Touch Screen Gestures Using Normal and Tangential Force. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '11, pp. 1909–1914, New York, NY, USA, 2011. ACM.

[HL11b] Seongkook Heo and Geehyuk Lee. Forcetap: Extending the Input Vocabulary of Mobile Touch Screens by Adding Tap Gestures. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '11, pp. 113–122, New York, NY, USA, 2011. ACM.

[HSH11] Chris Harrison, Julia Schwarz, and Scott E. Hudson. TapSense: Enhancing Finger Interaction on Touch Surfaces. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pp. 627–636, New York, NY, USA, 2011. ACM.

[KB08] Amy K. Karlson and Benjamin B. Bederson. One-handed Touchscreen Input for Legacy Applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pp. 1399–1408, New York, NY, USA, 2008. ACM.

[KI08] Masatomo Kobayashi and Takeo Igarashi. Ninja Cursors: Using Multiple Cursors to Assist Target Acquisition on Large Screens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pp. 949–958, New York, NY, USA, 2008. ACM.

[KYL12] Sunjun Kim, Jihyun Yu, and Geehyuk Lee. Interaction Techniques for Unreachable Objects on the Touchscreen. In *Proceedings of the 24th Australian Computer-Human Interaction Conference*, OzCHI '12, pp. 295–298, New York, NY, USA, 2012. ACM.

[MGLF09] Justin Matejka, Tovi Grossman, Jessica Lo, and George Fitzmaurice. The design and evaluation of multi-finger mouse emulation techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pp. 1073–1082, New York, NY, USA, 2009. ACM.

[PKB06] Pekka Parhi, Amy K. Karlson, and Benjamin B. Bederson. Target Size Study for One-handed Thumb Use on Small Touchscreen Devices. In *Proceedings of the 8th Conference on Human-computer Interaction with Mobile Devices and Services*, MobileHCI '06, pp. 203–210, New York, NY, USA, 2006. ACM.

[RHL08] Anne Roudaut, Stéphane Huot, and Eric Lecolinet. TapTap and MagStick: Improving One-handed Target Acquisition on Small Touch-screens. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '08, pp. 146–153, New York, NY, USA, 2008. ACM.

[RT09] Volker Roth and Thea Turner. Bezel Swipe: Conflict-free Scrolling and Multiple Selection on Mobile Touch Screen Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pp. 1523–1526, New York, NY, USA, 2009. ACM.

[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 Tenth IFIP TC13 International Conference on Human-Computer Interaction*, pp. 267–280. Springer, sep 2005.

[VB07] Daniel Vogel and Patrick Baudisch. Shift: A Technique for Operating Pen-based Interfaces Using Touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pp. 657–666, New York, NY, USA, 2007. ACM.

[YHHH13] Neng-Hao Yu, Da-Yuan Huang, Jia-Jyun Hsu, and Yi-Ping Hung. Rapid Selection of Hard-to-access Targets by Thumb on Mobile Touch-screens. In *Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services*, MobileHCI '13, pp. 400–403, New York, NY, USA, 2013. ACM.

[YMK+13] Takuto Yoshikawa, Yuusaku Mita, Takuro Kuribara, Buntarou Shizuki, and Jiro Tanaka. A Remote Pointing Technique Using Pull-out. In *Proceedings of the 15th International Conference on Human-Computer Interaction: Interaction Modalities and Techniques - Volume Part IV*, HCI'13, pp. 416–426, Berlin, Heidelberg, 2013. Springer-Verlag.

[YST12] Takuto Yoshikawa, Buntarou Shizuki, and Jiro Tanaka. HandyWidgets: local widgets pulled-out from hands. In *Proceedings of the 2012 ACM international conference on Interactive tabletops and surfaces*, ITS '12, pp. 197–200, New York, NY, USA, 2012. ACM.

[永田 12] 永田和信, 村田和義, 渋谷雄. モバイル機器における片手操作時の選択可能範囲外にあるオブジェクトの引き寄せ手法. シンポジウム モバイル'12, pp. 35–40, 6 pages, 2012.

[土佐 13] 土佐伸一郎, 田中二郎. LoopTouch：画面ループを用いたモバイル端末片手操作手法. インタラクション 2013, 8 pages, 2013.

付 録A　付録

A.1　実際に使用した書類

第 5 章における実験の際に使用した誓約書，実験手順書およびアンケートを付録として以下に示す.

A.1.1 誓約書

誓約書

大端末におけるポインティングに関する実験におけるデータ収集

文責：大西主紗

この度は実験にご協力いただき、ありがとうございます。

○本実験は、従来の操作手法及び我々の提案した操作手法におけるポインティングの特性を調査することを目的とします。

○実験途中には、任意のタイミングで休憩を取ることが可能です。

○実験への参加は協力者の自由意志によるものであり、
実験への参加を随時拒否・撤回することが可能です。

○この実験によって得られたデータは、個人が特定できないように処理を行います。

○学内外にて発表する論文に実験結果を利用することがありますが、
いかなる場合においても協力者のプライバシーは保全されます。

○実験風景の撮影を行う場合があり、また、その映像および静止画を論文並びに対外発表に掲載する可能性があります。
この掲載は拒否することが可能です。

実験に関して、上記内容を十分に理解し、同意していただけましたら下部の署名欄に
署名をお願いいたします。

平成　　年　　月　　日

所属＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

署名＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

実験風景の掲載を　許可する　・　拒否する

説明者　所属　情報学群情報科学類

氏名＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

図 A.1: 誓約書.

A.1.2　実験手順書

実験手順書

ポインティングに関する実験手順

文責：大西主紗

この度は実験にご協力いただき、ありがとうございます。
実験は 4 種類あり、各実験の前にはそれぞれ練習タスクを行っていただきます。
全ての実験が終わった後にアンケートを記入して頂きます。
なお、実験の所要時間は合計で 1 時間から一時間半程度です。

◎実験手順

・本実験の手順を以下に記載いたします。
1. 説明者によって口頭説明を行います。
2. 操作手法に用いるため、タスクを行う前にタッチパネルをタッチする際の指の加速度を計測します。
   ※これにより計測された加速度は実験タスクにて用いるため、再現可能なタッチを行ってください。
3. 1 タスク分の練習タスクを行っていただきます。
4. 本番タスクを 2 回行っていただきます。
5. 各タスクの間には任意に休憩を取っていただいて構いません。

◎タスク内容

本実験では、携帯情報端末上に表示されたターゲットの選択を行っていただきます。実験の際には、タッチのみを用いた選択、及び操作手法とタッチ操作を任意に使い分けて行う選択の 2 種類があります。また選択時には、画面にターゲットのみが表示された状態および、ターゲットおよびダミーターゲットが表示された状態の二種類があります。
これらのタスク間は、利き手のみを用いて携帯情報端末を把持し、その親指のみを用いて操作を行ってください。利き手は肘を机につけた状態でならば、自由に動かしてかまいません。携帯情報端末は、机など、利き手以外に触れないように把持してください。また、非利き手は机の下においてください。
携帯情報端末は、縦方向下辺および左辺に利き手掌が接触するようにならば持ち替えてかまいませんが、その際も利き手のみを用いてください。

◎操作手法

図 A.2: 実験手順書 1.

・本実験に用いる操作手法は 3 種類存在します。

●Corner Space

ベゼルから指をスワイプし画面から離すと、画面上にウィジェットが表示されます。この中心は、指を離した位置に固定されます。このウィジェットによって区切られた領域内のいずれかをタッチすると、その領域における画面角にポインタが表示されます。このポインタをドラッグ操作によってターゲットの上に移動させ、指を離すと、ポインタが重なったターゲットが選択されます。誤ってウィジェットを出現させた場合には、画面内をタップすることで消すことができます。また、ターゲット選択の成否にかかわらず、一度ポインタを動かした場合、画面から指を離すとウィジェットは消えます。

●Reflection

画面を強くタッチすると、親指の真下にポインタが出現します。その親指をドラッグすることで、指を動かした方向にポインタを移動させることができます。このポインタをドラッグ操作によってターゲットの上に移動させ、指を離すと、ポインタが重なったターゲットが選択されます。このポインタを画面領域の外に移動させようとした場合、画面端において反射移動を行います。また、一度反射移動を行ったポインタの移動量は減少します。

●TouchOver

画面を強くタッチし、指をドラッグすることによって、ポインタをドラッグ方向に応じた領域に出現させることができます。指とポインタの位置関係は、下図のように定まります。指を離すと、ポインタが重なったターゲットが選択されます。

図 A.3: 実験手順書 2.

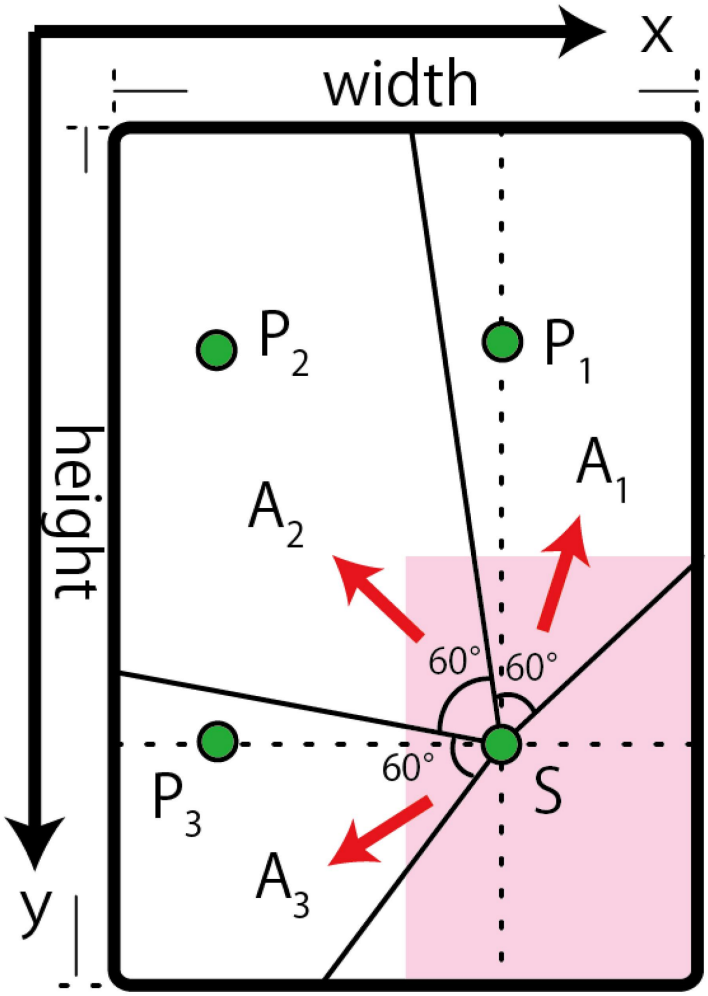

図 A.4: 実験手順書 3.

A.1.3　アンケート

ポインティング実験 アンケート

ポインティング実験にご協力いただきありがとうございます。
使用した感想・意見についてのアンケートにご協力をお願いします。

性別　男 ・ 女　　利き手　右 ・ 左　　年齢

<携帯情報端末の入力の経験について>

1. タッチパネルを持つ携帯情報端末の入力を行ったことがありますか？

はい・いいえ

2. （はいの場合）それは、どのようなデバイスにおいてですか？（複数可）

スマートフォン・タブレット・その他（　　）

3. また、その際片手把持状態において操作をし辛いと思ったことはありますか？

はい・いいえ

<提案手法について>

各操作手法について操作感はいかがでしたか？ 5 段階評価でお答えください。

Touch	1・2・3・4・5
ConerSpace	1・2・3・4・5
TouchOver	1・2・3・4・5
Reflection	1・2・3・4・5

起動手法に関する質問です。
ForceTap を使った感想をお答えください。

図 A.5: アンケート 1.

2. 操作手法に関する質問です。

Touch を使った感想をお答えください。

ConerSpace を使った感想をお答えください。

TouchOver を使った感想をお答えください。

1. Reflection 使用時、画面端での反射移動を行いましたか？

はい・いいえ

2. Reflection を使った感想をお答えください。

図 A.6: アンケート 2.

A.2　アンケート結果

実験の際得られたアンケートの結果を以下に示す.

A.2.1　操作感

各操作手法ごとの 5 段階評価の平均値の表を示す.

Touch	3.5
CornerSpace	4.75
TouchWarp	1.25
Reflection	3.5

A.2.2 Forcetouch の感想

Forcetouch の感想を以下に示す.

- Reflection においては使いやすく感じた.
- 弱の操作が，普段自分が使用しているもの（Galaxy S2）での反応より鈍く，何度も押してやっと反応するという感じで，使いにくさを感じました.
- 慣れるまでに時間がかかった．強弱の再現性がない人もいるので，個人的には難しかった.
- 強弱をつけているつもりでも，ちゃんとつけられているのか分からなかった.
- 画面を強くタッチするときと普通にタッチするときの区別がつらかった.
- タッチの加減が難しかった．なれれば使いやすいと思います.
- 強くタップすることは意識せずにできたが，弱くタップする方は安定せず使いにくかった.
- あまり安定してない．使いづらい.

A.2.3 Touch の感想

Touch の感想を以下に示す.

- 直接的，シンプルで分かりやすい．しかし，左上など，遠すぎる所や近すぎる所は使いにくい.
- 画面の左上，左下の角は持ち替えをしないとタッチ出来ず，また，画面の端に近いターゲットをタッチする際，右下に手のひらでの誤動作が多くなり使いにくいと感じました.
- 普通だった．今回，端末が大きかったので届かない場所があったのが面倒だった.
- 画面端まで指を届かせるのが困難.
- 端の方をタッチするのが難しいと感じた.
- 普段とあまり変わらないので楽だった.
- 画面が大きいので左端特に角はタッチしにくい．左に指を伸ばすと右側に手のひらが触れてしまうので注意する必要があった.
- 片手だけでは届かない場所が多くて不便.

A.2.4 CornerSpace の感想

CornerSpace の感想を以下に示す.

- 非常に使いやすかった．しいて言えば，一番初めに出てくる円がもっと画面の中央に出てほしい（円を出す時点で大きな端末の操作しにくさがある）.
- 画面中央上部以外は自然に位置を取れて操作できました.
- 画面端へのタッチが最も楽だった.
- 便利だった．試したものの中でもっとも使いやすく感じた.
- 通常では難しい画面端のタッチが容易にできてやりやすかった.
- 位置関係の把握が TouchWarp よりも楽だった．強弱を使い分ける必要がなかったので手もさほど疲れなかった.
- 使いやすい．正確さも早さも一番良かったと思う．しいて言えば，左端真ん中あたりに使用するときも角から出すしかないのが気になった.
- 起動方法が安定していた．思った通りの動作がしやすかった.

A.2.5 TouchWarp の感想

TouchWarp の感想を以下に示す.

- 強くタップした後にポインタを他の 3 つの領域へ動かす操作がし辛かった．どうしても指のスライドよりも端末を傾けるなどの操作をしてしまう.
- カーソルを出現させる操作がやや手間に感じましたが，カーソルの出現後は角の近くのターゲットも楽に選択できて楽でした.
- 狙ったところにタッチできるのに何回かかかった.
- 遠くのものを選択するのには便利だが，他のものに比べて難しく感じた.
- タッチしたところに対応する座標を見極めるのが難しいと感じた.
- 射影する先の位置を把握するのに手間取った.
- 初めはよくわからなかったが，慣れてからは使いやすくなった．起動の強くタッチはうまくいくが，通常タッチの弱いタッチがうまくいかない.
- ダミーなしでは狙いが定まらない．タッチ強弱のっ失敗が起こりやすくて使いづらかった.

A.2.6 Reflection の感想

Reflection の感想を以下に示す.

- 指に近すぎる所を選択しづらい．手自身が影になり見えなくなってしまった.
- とても自然に操作でき，快適でした．カーソルの伸びがもう少し大きく伸びると個人的にはより良いかなと感じました.
- 使いやすかった.
- 使いやすかったが，的が大きかったため反射を使う機会がなかった印象.
- 反射移動を使う必要がある場面がなかった.
- 画面真ん中からほとんど指を動かさなくていいので手がつかれなかった.
- ポインタ表示後の位置合わせに時間がとられた．TouchWarp と同様に通常タッチがうまくいかなかった.
- 起動が安定しなかった．CornerSpace と同じ起動方法だと使いやすそうだと感じた.