

高速ビジョンを用いた 指入カシシステムの研究

新倉雄大

目次

第 1 章	序論	1
1.1	はじめに	1
1.2	携帯機器向けインタフェースにおける制約	2
1.3	手を用いた携帯機器向け入力	2
1.4	本研究の目的	3
1.5	本論文の構成	4
第 2 章	携帯機器向け入力におけるインタフェース設計論	6
2.1	はじめに	6
2.2	既存の携帯機器形状における新たなインタフェース	6
2.3	形状に制約のない携帯機器向けインタフェースの研究	9
2.4	音を利用したジェスチャ認識に関する先行研究	14
2.5	空中操作における触覚フィードバックに関する先行研究	14
2.6	仮想物体とのインタラクションに関する先行研究	15
2.7	求められる携帯機器向け入力インタフェース	16
2.8	まとめ	19
第 3 章	空中操作インタフェース：空中を操作領域とする非装着型手入力システム	20
3.1	はじめに	20
3.2	空中操作インタフェース	22
3.2.1	概要	22
3.2.2	システム設計・開発	23
3.3	指先トラッキング・入力動作認識	24
3.3.1	概要	24
3.3.2	歪み補正	24
3.3.3	指先領域抽出・指先位置姿勢推定	25
3.3.4	入力動作検出	28
3.3.5	振動フィードバックの提示	29
3.3.6	アプリケーション例	30
3.4	システム性能評価	30

ii 目次

3.4.1	操作範囲	31
3.4.2	指先の位置合わせ精度	31
3.4.3	指先の奥行き位置認識精度	33
3.4.4	入力動作検出精度	33
3.4.5	背景に対するロバスト性	34
3.4.6	計算速度	35
3.5	空中操作インタフェースを用いた操作性に関する実験	37
3.5.1	実験1：フレームレートがシステムの操作性に与える影響に関する実験	37
3.5.2	実験2：振動フィードバックが操作性に与える影響に関する実験	40
3.5.3	タスク	40
3.6	まとめ	44
第4章	Anywhere Surface Touch：実環境の面を操作領域とする装着型手入力システム	46
4.1	はじめに	46
4.2	Anywhere Surface Touch	48
4.2.1	概要	48
4.2.2	ハードウェア構成	48
4.3	ジェスチャー入力認識	50
4.3.1	概要	50
4.3.2	指先トラッキング	51
4.3.3	音処理	53
4.3.4	画像処理と音処理の統合	54
4.3.5	ジェスチャー入力の決定	56
4.3.6	アプリケーション例	56
4.4	システム評価実験	57
4.4.1	タスク	57
4.4.2	被験者	60
4.4.3	結果	61
4.4.4	考察	62
4.5	まとめ	66
第5章	空中操作における仮想物体とのインタラクション	67
5.1	はじめに	67
5.2	裸眼立体視ディスプレイを用いた仮想物体とのインタラクション	68
5.2.1	システム設計	68
5.2.2	指先トラッキング	69
5.2.3	裸眼立体視ディスプレイを用いたインタラクションシステム	71
5.3	仮想物体とのインタラクションにおける同期精度向上	72

5.3.1	高速ビジョンを用いることによるレイテンシへの影響	73
5.3.2	高速ビジョンを用いることによる動作予測精度への影響	74
5.3.3	実験システム	74
5.3.4	ディスプレイ座標とカメラ座標の対応付け	75
5.3.5	指先位置認識	76
5.3.6	カルマンフィルタを用いた指先位置予測	77
5.3.7	指の動きのモデル化	78
5.3.8	レイテンシの計測	79
5.3.9	カルマンフィルタのノイズパラメータ設定	80
5.4	実験：指先位置予測精度向上による操作への影響	81
5.4.1	タスク	81
5.4.2	被験者	83
5.4.3	結果	83
5.4.4	考察	83
5.5	まとめ	84
第 6 章	結論	91
	謝辞	94
	参考文献	96

第 1 章

序論

1.1 はじめに

近年、携帯機器小型化の技術進歩に伴って、携帯電話やスマートフォン、PDA などが広く世の中に普及している。かつて大型計算機によって行われていたような計算処理も今や手元の小型端末によって実現されるようになり、もはや我々の日常に欠かせないものとなっている。

このような携帯機器のサイズにおける技術進化は未だ途上であり、究極的には身体の一部となるほどに小型化していくと考えられる。しかし、このように劇的に携帯機器のサイズが小型化する一方で、携帯機器における入力方法は本質的に変化していない。

最近では、携帯機器の小型化が進んだ影響で、図 1.1 に示すように機器表面に広い操作領域を確保した入力インタフェースの実現は難しくなっている。それに伴い、操作性の低下が問題となっている。

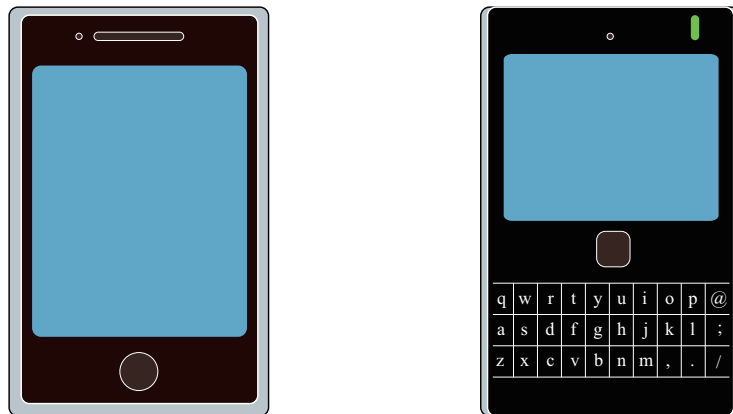


図 1.1. 近年の小型化した携帯機器

例えば、図 1.1 の左側のような携帯機器では、一つ一つの物理ボタンが小さいため、間違っ隣りのボタンを押してしまうといった問題が起こりやすい。また、右側の携帯機器のようにタッチパネルを採用した機器では、アプリによってその都度ボタンの配置を工夫することは可能ではあるが、実際には使い慣れていないキー配置で操作することは難しい。また、仮に特殊

2 第1章 序論

なボタン配置を利用したとしても、タッチパネルに触れる自分の指が画面を隠してしまうため、自分の指が今何に触れているかを正確に確認することが困難である。このような問題を Fat Finger Problem[1] と呼ぶ。Fat Finger Problem に取り組んだ様々な研究が報告されている。

1.2 携帯機器向けインタフェースにおける制約

前節で述べた Fat Finger Problem は、ユーザーインタフェースのハードウェア制約問題と考えられる。例えば、キーボードは人間の指でタイプ出来るほどの大きさでなければならない、といった制約問題である。

この制約問題は、すべて人間の身体的な特徴に由来する問題と考えられる。人間の身体的な特徴は大きく変化するものではない。したがって、この問題を解決するためには、新たな入力インタフェースを開発することが必要不可欠である。詳しくは後述するが、この問題に取り組み、新たなインタフェースを提案した多くの先行研究が存在する。しかし、それらの先行研究の多くは特定の使用環境で使用可能なものや、特別な機器を装着する必要があるものなど、使用にあたって制約があるものが多い。

1.3 手を用いた携帯機器向け入力

人間の身体的な特徴に由来する制約問題を解決する新たなインタフェースを実現する概念として、ナチュラルユーザーインターフェース (NUI) と呼ばれる概念がある。NUI は、キーボードやマウスといった装置を主体とせず、身振り手振りや発話といった、より人間にとって自然な動作を使用して入力を行うインタフェースの事である。

NUI のコンセプトは我々人間が自然に行う動作や発話を基本としてインタフェースの入力に利用するというコンセプトである。したがって、このコンセプトに基づいて提案されたインタフェースでは人間の身体的な特徴に由来する制約問題は発生しないと考えられる。

このように、携帯機器向け入力においては、自然な動作に基づく入力インタフェースが身体的な特徴に由来する問題を解決すると考えられる。しかし、一言で自然な動作とは言っても、具体的にどのような入力方法を利用するかという選択は十分考慮する必要がある。例えば、家電向けの入力システムでは、手を振るなどのジェスチャ入力を入力する手法 [2] が提案されている。また、人間の発話した言葉を認識し入力とするような手法も多く提案 [3] されている。実際、近年発売されているスマートフォンなどにも搭載されるなど実用化も進んでいる。

しかし、ジェスチャによる入力は、我々の普段の動作の中でも比較的大袈裟と考えられるものが多く、公共の場で行うには目立ってしまうという課題が存在する。実際、公共の場におけるジェスチャ入力は社会的な受容性が低いという評価結果が報告されている [4]。同様の理由から、音声による入力も社会的受容性が低いと考えられる。また、音声による入力は、簡単な操作は可能ではあるものの、一言一言が命令にあたるため、CLI と同様に複雑な入力を高速に実行することが難しい。したがって、複雑な操作が一般的となった種々のアプリケーションでの普及は困難であると考えられる。

そこで本研究では、自然な入力として、手を用いた入力システムを考える。従来の携帯機器においても、操作のほぼすべてが手を用いた入力であるため、社会的受容性は高いと考えられる。また、我々の身体の中でも、手は最も器用な動作が可能な部位であるため、複雑な操作を実現することも可能と考えられる。

しかし、手を用いた入力においても考慮すべき課題が存在する。それは、人間の手の動きが高速なため、その動作を捉えることが難しいという課題である。Erol[5]らの論文によれば、手の並進方向の運動は最高で5[m/s]、手首の回転運動は300[\pm /s]になると言われている。また、タッピング動作に関しても、指の屈曲回数は一秒間に6回ほど可能であると言われている[6]。入力インタフェースにおいては、より速くスムーズに携帯機器を使いこなしたいというユーザーの欲求に応えるためにも、このような高速な動作を用いた入力に対応することが望ましい。

1.4 本研究の目的

ここまで、携帯機器向け入力における課題や制約を紹介した。さらに、それらの課題や制約を克服する方法を検討し、その結果手を用いた入力が最も自然かつ適切な手法であると結論づけた。しかし、手の動きは高速であり、快適な入力を実現するにはシステムが高速な動作も認識する必要がある。

詳しくは2章で後述するが、手の高速な動きを認識することや、携帯機器としての大きさ、ユーザーに対する負担などを考慮すると、ビジョン技術を用いた入力インタフェースが最も適切な手法であると考えられる。

そこで本研究では、携帯機器における新たなインタフェースとして、高速ビジョンを用いた手入力に関するインタフェースに着目し、従来の課題を克服する新たな携帯機器向けインタフェースを提案することを目的とする。

具体的には、広い操作空間実現する新たなインタフェースとして、空中を操作領域とする空中操作インタフェースと、あらゆる面を入力面とする Anywhere Surface Touch を提案する。また、空中操作における操作の実体感を問題とし、裸眼立体視ディスプレイを用いたインタラクションの提案、及び仮想物体とのインタラクションにおける実体感の向上に関する研究を行う。

本論文における研究の位置づけを図1.2に示す。本論文では、既存の携帯機器を大きく変更することなく、操作領域の拡張、高精度な入力を実現するものとして、空中操作インタフェースを提案する。(図1.2 右上) また、空中操作における実体感の向上に関する研究も、携帯機器における高精度な入力を実現する研究の一部である。一方 Anywhere Surface Touch は、より小型化した携帯機器において、操作環境を多様化し、複雑な操作を高速に行えるインタフェースの実現を目指す。(図1.2 右下)

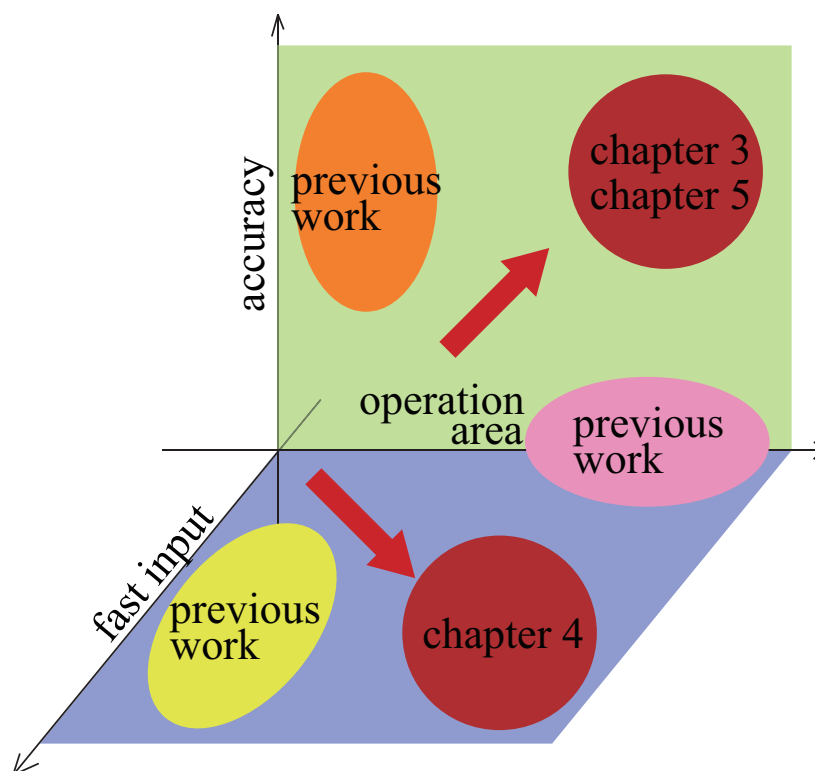


図 1.2. 本論文の位置づけ

1.5 本論文の構成

本論文の構成を図 1.3 に示す。まず本章では本研究に至る背景を述べ、目的を明確にした。

第2章では、より具体的に先行研究を紹介し、先に提案されたインタフェースのメリット・デメリットを検討する。検討したメリット・デメリットに基づき、求められる新たな携帯機器向けのインタフェースとはどのようなインタフェースであるかを述べる。また、そのような携帯機器向けのインタフェースを実現する上で考慮すべき問題についても述べる。

第3章では、求められるインタフェースの1つとして、空中を操作領域として広い操作領域、高精度な入力を実現する空中操作インタフェースを提案する。また、空中操作インタフェースのシステム評価や、提案システムを用いた空中操作における振動フィードバックの影響について実験を行い、その効果を検証する。

第4章では、求められるインタフェースのもう1つの形として、身近にある様々な面を入力面とする、Anywhere Surface Touch (AST) を提案する。ASTにより、複雑な操作を高速に行うことが可能となる。また、ASTを用いて様々な面上で入力が可能であることを確認する実験を行う。

第5章では、空中操作における操作の実体感の向上を目的とした研究を行う。まず、空中で操作している際に、従来の情報提示では表現が2次元に限定されてしまい操作の実体感を得ることが出来ないということを課題とし、裸眼立体視ディスプレイを用いて仮想物体とのインタ

ラクション可能なインタフェースを提案する。次に、仮想物体とのインタラクションにおいて指先位置と仮想物体の位置の同期精度を問題として取り上げ、高速ビジョンを用いて指先位置予測を行う。これにより、より実体感の高い仮想物体とインタラクションを実現する。最後に、第6章では本研究の結論を述べる。

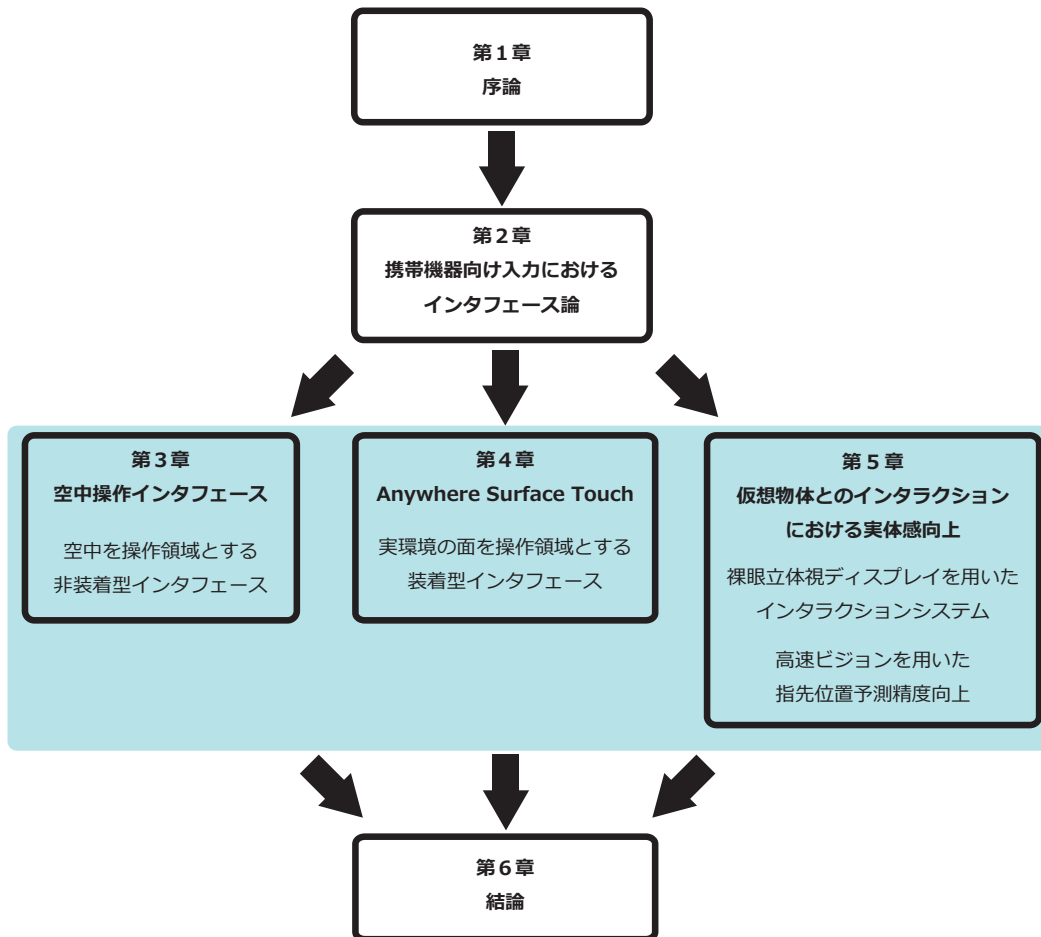


図 1.3. 本論文の構成

第 2 章

携帯機器向け入力におけるインタフェース設計論

2.1 はじめに

本研究では、携帯機器における入力において人間の身体的な特徴による制約の克服を課題とし、問題解決のための新たなインタフェースを提案する。そのために、まずは先行研究について議論し、それぞれのコンセプトのメリット・デメリットを検討する。

携帯機器向けに新たなインタフェースを提案した先行研究は数多く存在するが、ここではそれらを大きく 2 つに分けて紹介する。一つは、既存の携帯機器の形状を大きく変更することなく Fat Finger Problem などの問題に取り組んだ先行研究である。もう一つは、既存の携帯機器の形状に囚われることなく、より小型化した機器やウェアラブルデバイスを利用した携帯向けインタフェースの先行研究である。ここでは、それらの先行研究についてそれぞれメリットとデメリットをまとめる。また、携帯機器における入力を考察する上で、入力に付随する問題がいくつか考えられる。ここではそれらについても先行研究を引用して考察する。

そして最後に、先行研究を踏まえて、どのような携帯機器向けインタフェースが求められるかを検討する。

2.2 既存の携帯機器形状における新たなインタフェース

携帯機器が小型化したことで、機器表面の操作領域が縮小し、操作が行いにくいという問題が存在する。この問題に対し、既存の携帯機器の形状を大きく変化させることなく問題の解決を図った研究が存在する。

Vogel ら [7] は Shift と呼ばれる手法を導入することによって、タッチパネルにおける指のオクルージョンの問題の解決を図った。この手法では、指で触れている画面の画像を指先の上になぞらして表示することで、指によるオクルージョンを低減させる。しかし、一方で本来表示されていたものが見えなくなるため、必要な情報が隠れてしまう可能性がある。

同様に、Wigdor ら [8] も指で操作したい対象物を隠してしまうことが小さいタッチスクリー

ン上での操作を困難にしていると考えた。そこで、図 2.1 にあるような LucidTouch という新しい小型機器を提案した。このシステムではディスプレイの背面にタッチパネルを配置し、指が接触している箇所を検出する。また、ディスプレイの後方にディスプレイの方を向いたカメラが取り付けられており、このカメラでそれぞれの指のシルエットと指先に表示されたカーソルの画像を取得する。この手法によって、Fat Finger Problem のオクルージョン問題が解決出来るが、指の画像を安定して得ることが難しいという課題が残されている。また、システム構成上、カメラが後方に突き出しているようなシステムは携帯機器としては大き過ぎると考えられる。

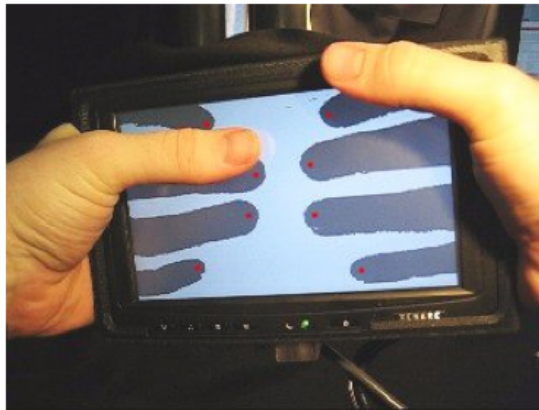


図 2.1. Wigdor らの先行研究 : LucidTouch[8]

また、Baudisch[9] らは、Fat Finger Problem を解決する方法として、機器の背面にタッチパネルを取り付け、タッチによるオクルージョンが発生しないインタフェースとして、nanoTouch を提案した。この手法は LucidTouch のコンセプトを発展させたものであり、大きさの面でも携帯機器として実用的であるといえる。しかし、一方で背面にタッチパネルがある場合タッチしようとしている指を直接見ることが出来ないため、正確なタッチが困難になるという課題が存在する。言い換えれば、Fat Finger Problem の本質である指のオクルージョンを解消するために、オクルージョンが発生しない手法を提案したが、指全体を見ることができないために正確なタッチが難しくなるというジレンマが存在する。

ここまでに紹介した先行研究は、タッチパネルを利用しつつ、オクルージョンの問題を解決することを目的とした携帯機器向けの新たなインタフェースである。しかし、携帯機器向けのインタフェースとして手を入力に用いる事を考えると、タッチパネルに拘らず、空中に操作領域を拡張することで問題の解決を図ることが考えられる。手に何もつけることなく空中で動く手の位置姿勢を認識することが出来れば、

例えば、Butler ら [10] は、SideSight というシステムを提案した。この手法では、携帯電話の側面に赤外 LED とセンサを直線上に設置し、接近した指が反射した赤外 LED の光をセンサが検出する。これにより、携帯機器の側面の外側に広がる空間を操作領域とすることを可能とした。言い換えれば、機器表面の操作領域の大きさに囚われることなく、広い空間を操作領域とすることが可能とした。しかし、一方でこの入力インタフェースは、通常の携帯機器で最

8 第2章 携帯機器向け入力におけるインタフェース設計論

も普及した操作方法である、携帯機器を持ちながら操作するという操作方法を実現することが出来ない。また、現状ではひとつの側面で一点しか認識が出来ないため、タッチパネルのような様々な入力を行うには本体を平面に置き、両側面で両手を使って行う必要がある。

その他にも、主にコンピュータビジョンの分野を中心に、手の三次元位置姿勢を高精度に推定しようとする研究は昔から多数行われている。例えば、Rehgら[11]は、手のモデルとして直方体や円柱、半球などを組み合わせた手の三次元モデルを作成し、手のモデルと得られた画像とのマッチングを取ることで手の姿勢を認識する手法を提案した。この手法では2台のカメラを用いて手を撮影しており、それぞれのカメから見えを最適化するようにパラメータを求めることによって位置姿勢推定を実現している。

星野ら[12]は、予め撮影した4500枚の手指画像と、同時にデータグローブで計測した指関節姿勢を対応付けたデータベースにより、短い処理時間で(150-160 fps)で姿勢推定を行う手法を提案した。しかし、操作自由度の高い操作をユーザーに提供することを考慮した場合、使用出来る指の数や屈曲の自由度など数多くの種類の姿勢を撮影しておく必要がある。また、データベースにおける姿勢角は離散的なものとなっており、ユーザーの連続的な姿勢変化を自然な操作に対応付けるためには高い分解能で姿勢角を変化させたデータベース画像を用意しなければならないという課題がある。また、Athitsosら[13]も、あらかじめ大規模な手の形状のデータベースを構築し、取得した画像とデータベースの照合を高速に行うことで、リアルタイムに手の形状を推定する手法を提案しているが、同様の課題を抱えている。

Wangら[14]は、色分けされたグローブを装着し手に、それをカメラで撮影することで手の位置姿勢推定を行うシステムを提案した。この手法では、手指の様々な姿勢に対して、色分けされたグローブ画像の見えを対応づけたデータベースをあらかじめ作成し、カメラで取得した画像とデータベースを比較することで手の状態認識を行なっている。インタフェースとして想定する手指運動における画像を用いて認識する場合は認識のロバスト性を保ちやすいと考えられる。しかし一方で、グローブの装着は煩雑であり、利便性を下げると考えられる。また、グローブを使用しない場合、ほぼ均一な肌色の手指では入力画像に対してテクスチャを用いて画像比較を行うことは困難であると考えられる。

複数のカメラを用いて手の三次元形状を取得し、そのデータを用いた位置姿勢推定を行う研究も行われている。例えば、Causo[15]らは、立方体状の金属フレームに4つのカメラをとりつけ、様々な方向から手を見ることで立体交差法を用いて手の三次元形状を復元し、あらかじめ作製した手のモデルと比較することで位置姿勢推定を行う手法を提案した。

また、Schlattermanら[16]は、上方から3台のカメラで手を撮影し、立体交差法を用いて手の三次元形状を大まかに推定する手法を提案した。具体的には、手の全部の自由度である26自由度のうちある特定の6自由度のみ推定し、推定結果から4つのジェスチャ入力を認識する手法を提案した。

Oikonomidisら[17, 18]の提案手法も、同様に8台のカメラを配置し手を様々な角度から撮影する。ただしこの手法では、手のモデルと同時に手で把持するための物体をもモデル化し、手が何か物体を把持していてオクルージョンが発生しているような状況で手の26自由度と物体6自由度の位置姿勢推定を行う手法を提案した。この手法を用いることができれば、素手で

機器を操作するだけでなく、何か物体を持って機器を操作するといったことも可能になると考えられる。

しかし、立体交差法を行うためには、手の周囲の様々な方向からカメラで手の画像を取得する必要がある。携帯機器においてそのような構成でカメラを配置することは困難であり、システム構成上携帯機器では実現することが難しい。

システム構成上の観点から言えば、携帯機器で最も実現が容易な方法は、単眼のカメラを用いた手指の三次元位置姿勢推定であると考えられる。単眼のカメラであれば既存のほとんどの携帯機器にも搭載されており、実用化に向けたコストも低い。その点が大きなメリットとなる。

実際、Chang ら [19] は、手のひらの側を単眼カメラで撮影し、指の曲げ状態を推定する手法を提案している。Chang らの手法は、手の姿勢を表す状態モデルとカメラでの手の見えを表す見え状態モデルを作成し、それらの対応をパーティクルフィルターを用いることで確率的に推定する手法である。

また、Athitsos ら [20] によって、単眼カメラを用いて手の全 26 自由度を考慮した位置姿勢推定の手法が提案されている。この手法では、あらかじめ手の見え方の変化をデータベースとして保持しておく。得られた画像からエッジ情報を抽出し、確率的なエッジのマッチングを行うことで手の姿勢を推定している。

立体交差法での手の三次元位置姿勢推定手法を提案した Oikonomidis らは、Kinect を利用することで単眼カメラで推定を行う手法 [21] も提案している。この手法では、kinect を用いて得られたデプスデータ、カメラから得られた画像データに対して、あらかじめ作製した手の三次元モデルのマッチングを行うことで、手の位置姿勢推定を行う。なお、マッチングには Particle Swarming Optimization を利用し、15fps 程度の処理速度も実現している。しかし、手の動きは一般に高速であるため、15fps 程度では高速な手の動きを推定することは難しいと考えられる。

このように、単眼カメラで手の位置姿勢推定を行う手法がいくつも存在する。しかし、これらの手法はどれも計算量が多く、一枚の画像に対し指の位置姿勢を推定するのに数秒以上かかってしまう。従って、単眼カメラで手の全自由度を考慮して三次元位置姿勢を推定しようとした場合、計算量が膨大になってしまうため、計算資源が限られている携帯機器向けインタフェースにこのような手法を取り入れることは困難であると考えられる。

このように、既存の携帯機器の形状を大きく変えることなく、Fat Finger Problem に取り組んだ種々の研究が存在する。これらは、より高精度な入力を実現する、もしくは操作領域を拡張することを目標とした研究である。しかし、これらを両立させた研究は存在していない。

2.3 形状に制約のない携帯機器向けインタフェースの研究

携帯機器が小型化したとき、機器そのものの操作領域が狭くなるという問題が発生する。この問題の解決を試みた先行研究が存在するが、それらは大きく 2つのアプローチにわけられる。一つは空中の広い空間を操作領域として利用する方法であり、もう一つは周囲の面を入力

領域として利用するという方法である。

空中を操作領域として拡張する先行研究として、SixthSense[22]がある。Mistryらによって提案されたSixthSenseは、図2.2の左上にあるように、カメラ、プロジェクタ、ミラー、指先のマーカで構成されるシステムである。カメラによってマーカ的位置を認識し、その状態にあわせてプロジェクタから投影された映像をミラーで実空間に投影する。こうすることで、どんな場所でも空中で指を動かすことで操作を行うことが可能となる。



図 2.2. Mistry らの先行研究:SixthSense[22]

また、Harrisonら[23]は、Abracadabraと呼ばれるシステムの中で、磁石を用いて空中での機器操作を実現した。これは、指先に磁石の取り付けられた小型の装置を身に付け、その磁石の発生させる磁場の状態や変化を認識し、腕時計型の小型機器を操作することが可能なシステムである。同様に、Chenら[24]のuTrackも指先に磁石を付けて空中での操作を行うシステムである。また、Chanら[25]のFingerPadは、磁石を用いて指の腹を入力面とするシステムである。磁場の検出は古くから様々なセンサを用いて行われており、指先位置の認識も行い易い。しかし、これらのシステムはポインティング操作は行いやすいものの、メニュー操作などの複雑な操作を高速に行うには適さないと考えられる。また、携帯電話など電波を送受信するような機器では、機器自身が発生させる磁場などノイズの原因となる要因が多数考えられるため、高精度に操作することが難しいと考えられる。

その他、ウェアラブルカメラを用いた研究が多数報告されている。Watanabeら[26, 27]はカメラを脚に取り付け、カメラから得られる外界の変動を入力とすることで、歩行動作の推定が可能な手法を提案した。また、Shiratoriら[28]らは、体中にカメラを取り付け画像の変化から関節角や関節位置を推定しモーションキャプチャを実現するシステムを提案した。

また、KimらのDigits[29]では、カメラと赤外線LED、赤外線レーザーで構成されたデバイスを手首から前腕にかけて装着する事で、指の三次元的な屈曲状態の推定を可能にした。ただし、この手法ではあらかじめ手にレーザーが照射された時のカメラからの見え方と指の屈曲状態が一意に対応することを仮定して推定を行っている。そのため、使用するためにはあらか

じめ見えと指の屈曲状態の対応関係をキャリブレーションする必要がある。また、一般生活でレーザーを利用する事は危険で、特にあらゆる方向を向くことのある手にレーザーを取り付けることは思わぬ事故を誘発する危険性が存在すると考えられる。

ここまでウェアラブルカメラを用いた人間の動作認識に関する手法を紹介した。しかし、これらウェアラブルカメラを用いた研究は、身体や手の姿勢を推定することを主な目標としている。本研究は新たなインタフェースの提案を目的としており、これらの研究は我々とは目的が異なる。

ウェアラブルカメラを用いて空中を操作環境としたインタフェースとして、インタフェースを紹介する。例えば、Howardら[30]は、図2.3にあるようにLightgloveというシステムを提案した。このシステムでは、手首にLEDアレイ、受光センサアレイ、2次元加速度計で構成されたデバイスを身に付ける。図2.3のように、指先が曲がると手首のデバイスから放出された光を反射し、それを受光センサが検出することによって空中での指先の動作を認識する。また、加速度計を用いることで手全体の動きを検出し、カーソル移動などに利用することが可能となっている。こうすることでいかなる場所でも空中で指を動かして入力動作を行うことが可能となる。しかし、各指に関して曲げているか曲げていないかの2状態しか認識出来ず、複雑な操作を行うことは難しい。

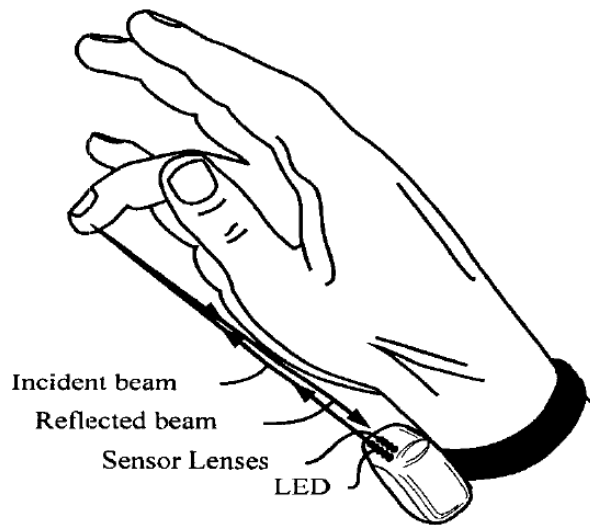


図 2.3. Howard らの先行研究:Lightglove[30]

UbiHand[31, 32]では、手首の内側にカメラを装着して指先の位置を検出し、指先位置の時系列データから手の状態を認識している。この手法では、空中で指を動かすことで入力を実現しているが、画像のみ用いた場合の認識精度は高くなく、言語モデルなどの修正・予測モデルを入れることで認識精度の改善を図っている。

以上、空中を利用した装着型携帯機器向けインタフェースの先行研究を検討した。これらのメリットは、先にも述べたように、空中での操作を導入することで、携帯機器表面よりも広い領域を実現した点である。また、一部の手法では手や指の三次元的な位置姿勢が認識可能なた

12 第2章 携帯機器向け入力におけるインタフェース設計論

め、アプリケーションによっては従来より便利な入力方法が実現可能になると考えられる。しかし、多くはポインティング操作などに留まっており、複雑な操作を高速に行うことは難しい。

上記のように、空中を操作領域とする手法には一定のメリットが存在する。しかし一方で、我々が使い慣れている入力は直接何かに触れて入力するようなインタフェースであり、空中で操作していることで操作している指への触覚フィードバックが得られない事は、操作の実体感に乏しいという課題を生む。従って、空中を操作環境とするインタフェースにおいては、触覚フィードバックの欠如を補償する異なるフィードバックの与え方を考慮する必要がある。(詳しくは2.5節で後述する。)

触覚フィードバックの欠如という問題に対し、身体の様々な場所を入力面とし直接指で触って操作することで、操作領域を拡張しつつ触覚フィードバックも提示可能なウェアラブルなインタフェースが提案されている。例えば、OmniTouch[33]では、肩にデプスカメラと小型のプロジェクトを装着し、手のひらや腕、手に持った紙などを入力面として利用する。

また、Skinput[34]では上腕にピエゾセンサとプロジェクトを取り付け、上腕や手のひら上でのタップ入力を検出する。ピエゾセンサは、人体の皮膚の固有振動数などを考慮した特殊な振動特性を持つものを幾つか用意し、それらをアレイ状に配置することで、前腕のどの部分で入力が行われたかを検出する。また、プロジェクトを用いて前腕部に映像を投影し、情報提示を行う。Ogataら[35]も、腕に2つのアームバンドを取り付けることで、上腕での入力を可能とするインタフェースを提案している。

Makinoら[36]は、腕時計型のシステムを装着し、手の甲を入力面とするインタフェースを提案した。このインタフェースは他のシステムと比較しても小型であり、携帯機器向け入力として望ましいと考えられる。しかし、現状ではポインティング操作やクリックに相当する動作など単純な操作しか認識することが出来ない。

これらの研究は、身体という常に存在する平面を入力平面として利用しているという点で優れている。しかし、OmniTouchのように上部から観測する場合、操作面と指先の接触認識を高精度に行うことが難しく、面と指との距離が僅かに離れている場合でも接触したと誤認識されてしまう可能性がある。特に、高速な入力を行う場合は、接触点と指先の距離は数[mm]程度の距離を維持して入力が行われると考えられるため、この誤認識は重要な問題となる。また、SkinputやSenSkinなどの皮膚の特性を利用したインタフェースでは、使用する際に洋服を着用することが困難という制約があり、一般的な利用が困難と考えられる。

これらの機器装着型インタフェースでは、指先や手首、腕などに直接機器を取り付けることにより、様々な環境で入力が可能になるといったメリットが存在する。しかし、携帯機器を日常的に利用するといった側面を考慮した場合、大型装置を身に付けることの煩雑さは十分考慮しなければならない。

また、機器を特定の場所に設置・利用することで様々な場所を入力面とする研究も行われている。例えば、Wilsonらによって提案されたPlayAnywhere[37]は、通常の平面を操作可能な平面に変えるコンセプトを提案したインタフェースである。PlayAnywhereはステレオカメラと赤外照明を平面の上方500[mm]程度の高さに設置する。あらかじめ平面とシステムの位置関係をキャリブレーションし、ステレオカメラによる視差の情報を用いることで指と面との接

触を検出する。こうすることで、PlayAnywhere が設置された平面をタッチパネルのような操作可能な平面とすることが可能になる。また、Agarwal[38] からも、同様のシステム構成によるタッチ認識システムを提案し、より高精度に指の接触姿勢を認識する手法を提案している。しかし、これらのシステムではステレオカメラを平面から少し離れた位置に設置しなければならず、システム構成の観点からこの手法を携帯機器に利用するのは難しい。

Roeber ら [39] は、Canesta Projection Keyboard というシステムを提案した。このシステムは、小型プロジェクタとカメラ、ラインレーザーからなる小型のデバイスを平面に設置して利用する。これにより、図 2.4 のように様々な平面にキーボードを描画し、赤外線の反射率が変化する事を利用して指先の位置を検出し、キーボード入力を実現した。



図 2.4. Roeber らの先行研究:Canesta Projection Keyboard[39]

また、Mouse 2.0[40] の論文中においても、ラインレーザーを利用して平面との接触を認識する新たなマウスが提案されている。Canesta Projection Keyboard や Mouse2.0 で提案された手法のメリットは、使い慣れているキーボードなどと同じインタフェースが様々な平面上で利用可能なことである。また、平面であれば高精度にタッチ検出を行うことも可能である。しかし、一方で利用には機器設置やキーボードを投影するために広くて平らな面が必要である。更に、曲面や凹凸のある面上ではラインレーザーが遮蔽されてしまい、タッチ認識が失敗すると考えられる。また、一般的に仮想キーボードはブラインドタッチによる入力が困難なため、高速に入力することが難しい。携帯機器を利用する際に、常に平らで広い面が周囲に存在するとは限らないため、いつでも利用可能であることをこの手法で実現することは難しい。

また、Minput[41] では、小型ディスプレイの裏面に 2 つの光学センサを置く手法が取られている。これらの光学センサはマウスと同様の原理で利用され、デバイスを面に沿って動かすことで入力を可能にするインタフェースである。マウスと異なり 2 つの光学センサを用いたことで、並進運動だけでなく回転運動も認識することが可能になった。この手法であれば、利用する面が平らでなくても利用可能であるという意味で優れているといえるが、一方で単純な入力しか行えず、文字入力など複雑な入力には適していないという課題が存在する。

以上、様々な形状で携帯機器向け入力を実現した先行研究を検討した。それぞれ、様々な場

所で入力が可能といったメリットや、既存の入力方法と同等に高速に複雑な入力が可能といったメリットが存在する。しかし、この2つのメリットを同時に達成した研究は存在しない。

2.4 音を利用したジェスチャ認識に関する先行研究

ここまで、携帯機器向け入力インタフェースの先行研究を紹介した。ここまでの先行研究の課題の1つとして、携帯機器としてのシステムの大きさや処理の容易さ、いつでもどこでも使えるなどを考慮すると、認識可能な入力が限られてしまい、複雑な操作を高速に行うことが難しいということがあげられる。それに対し、Skinputが利用していたように、音を追加的に用いる事によってより複雑な人間の動作を認識するということが可能になると考えられる。そこで、ここでは音を利用して人間の動作を認識する先行研究をいくつか紹介する。

例えば、BodyScope[42]ではマイクを首にとりつけ、種々の動作を区別する手法を提案している。この手法では12種類もの動作認識を実現しているが、一方で、一つ一つの動作は数秒以上継続するものが対象であるため、携帯機器の入力のように高速な入力を認識する方法として利用する事は難しい。また、この研究ではオフラインでの動作認識にとどまっている。

Hambone[43]では、手首や足首など身体の様々な場所に圧電素子を取り付け、各場所において得られた信号からジェスチャの認識を行なっている。この研究では複数のジェスチャをオンラインで認識する事に成功しているが、それぞれのジェスチャを認識するために2秒録音してから処理を行う必要があるため、動作から2秒程度の遅延が必ず発生するという問題がある。

Scratch Input[44]では、聴診器を平面に取り付け、面を引っ掻く音で入力を行っている。このシステムは主に音の振幅とピークカウントで動作をリアルタイムに認識し、タップやスワイプなどのジェスチャを区別する事に成功している。

このように、音を用いる事でもシステムが人間の複数の動作を認識する事が可能である。ただしカメラと異なり、どの指が動作を行ったかといった情報を取得する事は難しい。また、音だけでは突発的なノイズで誤認識される可能性もある。

2.5 空中操作における触覚フィードバックに関する先行研究

空中で非接触の操作をする場合、操作している手が入力動作を行なっても、スマートフォンを操作している時のように直接機器に触ることがない。そのため、操作している手に触覚フィードバックを与えることが出来ない。このような操作時の触覚フィードバックは操作性向上に寄与していると考えられる。したがって、空中操作において、どのようにして触覚フィードバックを代替し、操作性を向上させるかについて考慮する必要がある。

携帯機器において多様な振動フィードバックを与え、それにより操作効率を向上させようとする先行研究がいくつか存在する。例えば、Active Click[45]では、タッチパネルを触った指に触った箇所に応じた触覚フィードバックを与えたり、機器を把持している手に多様な触覚フィードバックを与えるたりする新たなアクチュエータが提案されている。しかし、先程述べたように、本研究ではカメラによる非接触の操作を実現することを想定しているため、このよ

うな方法による触覚フィードバックの提示は難しい。

Rekimoto[46] は、指先に振動素子と受光素子などを組み合わせた小型の機器を身につけ、空間に投影されたパターンに合わせて多様な触覚フィードバックを指先に与える手法を提案している。しかし、この手法では空間にパターンを投影するために複数の方向からプロジェクタを用いてパターンを投影する必要があるため、立体交差法と同様に携帯機器としてのシステム構成上実現が難しい。

また、星ら [47, 48] は超音波を用いて遠隔的に空中の指へ触覚フィードバックを与える研究を提案している。しかし、現状ではこのシステムは大型であり携帯機器に搭載可能な大きさではないと考えられる。

このように、空中操作に対する触覚フィードバック提示方法としては、指先に特別な装置を装着するなどして提示する方法と、超音波などを用いて遠隔的に触覚フィードバックを与える方法がある。しかし、触覚フィードバックのためだけに指先にウェアラブルデバイスを身につける事はユーザーにとって煩雑であると考えられる。また、一般的な動作を行う際に邪魔になる可能性が高いという課題も考えられる。一方、遠隔的に触覚フィードバックを与える方法では、携帯機器としての実装が難しいという問題が存在すると考えられる。

2.6 仮想物体とのインタラクションに関する先行研究

タッチパネルを用いることよって、使用する文脈に応じた様々なメニューやボタンなどをユーザーに提示することが可能になっている。しかし、Siek ら [1] が指摘した Fat finger problem に見られるように、現在の小型機器においては狭い機器表面で操作を行う必要があり、細かい操作を正確に行うことが難しいという問題が存在する。

それに対し、我々は Fat Finger Problem を解決する一つの手法として、空中を操作環境とする空中操作インタフェースを提案する。だが、操作が3次元的に行えるのに対して、現状ではボタンなどの表示が二次元平面上に限定されてしまうため、ボタンなどのオブジェクトを直接操作しているという実体感に乏しいという課題が存在する。

この課題に対し、実体感のあるインターフェースとして、実空間上で実物体を操作するタンジブルユーザーインタフェースの研究が行われている。情報に物理的な実体を与えることによって、ユーザーが操作を行う実空間と計算機内の情報空間の差を縮めることが出来ると考えられている [49]。例えば、Jorda ら [50] は、机の上にマーカーの描かれた物体を置き、それらを机の下からカメラを用いて認識することで、即興の音楽を演奏することが出来る reacTable を提案した。

また、Patten らの提案する Sensetable[51] では、電磁石を埋め込んだ物体を机の上に置き、それらを動かすことで操作を行う。物体に埋め込まれた電磁石によって物体の位置や向きを検出し、システム情報にあるプロジェクタを用いて机の上に情報を提示する。これらのシステムは、異なる物体を机上の様々な位置に置くことで、直感的に操作を行うことが可能である。しかし、操作を行うために実物体を利用する場合には、用途ごとのシステムの設計が必要であり、汎用性が低いという問題がある。

また、VR技術を用いて三次元仮想物体を提示し、実体感の伴うインタラクションを実現する研究も多数提案されている。例えば、指先に機器を装着し、特別な機器を用いて空中に結像された仮想物体とのインタラクションを実現するシステムとして、RePro3[52]というシステムが提案されている。このシステムでは、プロジェクターアレイと再帰性反射スクリーン、ハーフミラーを利用することで、実空間上に仮想物体を結像する。また、カメラで指先の位置を検出し、指先に特別な機器を装着することで、仮想物体と接触が起きた際に指先に触覚が与えられる。

Benkoらの提案するmirageTable[53]は、ステレオプロジェクターと専用の眼鏡、及びKinectを用いたインタラクションシステムである。ユーザー自身の前面に用意されたスクリーンに仮想物体が立体表示され、ユーザーはそれを素手で操作する。また、このシステムでは、ユーザーの視点位置をトラッキングし、正確な三次元映像を投影している。

また、指にマーカを取り付け、ホログラフィーを利用して空中に投影されたオブジェクトとインタラクションするシステムも提案されている。[54]。しかしこれらの手法では、利用にあたって立体視用の眼鏡やセンサなどの特殊な装置を着用する必要があり、その煩わしさが問題となる。

このような煩わしさを解消するためには、カメラを用いて非接触で手を計測し、また裸眼立体視可能なディスプレイを用いてインタラクションを実現することが望ましいと考えられる。Hilligesらが提案したHoloDesk[55]は、そのようなシステムの一つである。このシステムでは、デブスカメラを用いてユーザーの手の動きを認識し、仮想物体を動かす、持ち上げるといったインタラクションを可能にした。しかしながら、一般的なカメラは出力までの遅れ（レイテンシ）が大きいため、仮想物体と実物体の間に位置ずれが生じるという問題がある。

仮想物体とのインタラクションにおけるレイテンシの研究も様々なものが行われている。例えば、Meehanら[56]による研究では、HMD（ヘッドマウントディスプレイ）を装着して仮想環境下で作業する場合において、50-90[ms]の間でレイテンシが変化することによって実体感が異なる事を示されている。また、タッチパネルにおいては、Jotaら[57]が1[ms]でもレイテンシを短縮することでポインティングタスクのスループットが向上するといった結果を示している。

一般に、システムのレイテンシとユーザーのパフォーマンスの関係性は、操作方法やタスクによって異なると考えられている。従って、特定の操作方法やタスクに応じて評価を行う必要がある。

2.7 求められる携帯機器向け入力インタフェース

ここまで携帯機器向け入力に関する様々な先行研究を紹介した。上記で述べたように、既存の手法にはそれぞれ優れた面もある一方で、課題も存在する。しかし、情報提示のあり方から考えると、求められる携帯機器には大きく分けて2つのシナリオが考えられる。

1つは、既存の携帯機器のように4インチ前後のディスプレイを中心とした形状が大きく変わらないシナリオである。この場合、より広い範囲で高精度な入力の実現が求められる。そ

して、先行研究での検討内容から、空中で3次元的に手を動かすことによって操作を行うことが望ましいと考えられる。なぜなら、これによりディスプレイ面よりも広い空間で高精度な操作が可能になると考えられるからである。また、三次元的な動きを認識することによって、情報提示も三次元化し、立体的なオブジェクトの操作など、これまでとは異なった新たなアプリケーションが実現可能になると考えられる。

空中で3次元的に手を動かして操作する場合、情報を提示する端末を把持している手を使って操作することは著しく視認性を低下させると考えられる。従って、図2.5のように、機器を把持している手とは別の、もう片方の手を用いて入力を行うことが自然であると考えられる。



図 2.5. 空中操作による入力のイメージ図

空中で操作する場合、操作する手に何もつけることなく、携帯機器単体で手の動きを認識可能であるほうが、ユーザーの利便性は高いと考えられる。従って、非装着型のインタフェースであるほうが望ましいと言える。

非装着型の新たな携帯機器向けインタフェースを設計するにあたっては、以下の3つの条件を満たす必要があると考えられる。

- 設計方針として装置を体に装着する必要がない
- 使用環境を限定しない
- 細かい操作を高速に行うことが出来る

これらを満たすことにより、携帯機器の利便性を損なうことなく、より高度な操作を高速に行うことが出来るようになると考えられる。

上記の先行研究から、この3つの要求を満たすインタフェースを開発するには、ビジョン技術を用いることが最も有効な手法であると考えられる。カメラを用いてユーザーの指の動きを広い空間で撮影することによって、特別な環境や特殊な装置を身につけることなく、機器の操

作を行うことが出来ると考えられる。また、システム構成上の観点から、単眼のカメラを用いて指の3次元位置姿勢を推定する手法を実現することが望ましい。最後に、どのような場所でも利用可能であるように、複雑な背景環境の変化に対してロバストに指先領域を抽出可能である必要がある。

このような要求を満たすインタフェースとして、我々は空中操作インタフェースを提案する。提案システムは単眼のカメラを用いてユーザーの指の3次元位置姿勢を推定し、認識結果を用いてポインティング操作や入力動作の認識を行うシステムである。提案システムは、特殊な機器を装着すること無く、空中で指を動かすことのみでシステムの操作を行うことが可能である。

空中操作インタフェースに関する詳細は第3章で述べる。また、空中操作において、触覚フィードバックの欠如による操作性への影響や、レイテンシの変化による操作性への影響について上記で述べた。これらの問題について、本研究では空中操作インタフェースを用いて、実験を行い検証した。これらの実験についても、第3章で詳細に述べる。

一方、もう1つの想定される携帯機器向け入力のシナリオは、メガネ型デバイスや腕時計型のモニターのように、情報提示デバイスがより小型化するシナリオである。従来はディスプレイの大きさが入力端末における一つの制約となっていたが、情報提示デバイスがより小型化した世界では、そのような制約は存在しない。従って、情報と人間をつなぐ入力のあり方も変化し、より複雑な操作を高速に行えることが求められる。先行研究を踏まえると、携帯機器が身近な環境の平面や曲面を操作環境と認識することが可能になれば、携帯機器が小型化することによって得られるメリットを最大限享受することが出来ると考えられる。

このコンセプトを実現するには、ユーザーの指の動きや指と面の接触を常に捉えるために、画像情報を用いて認識を行う事が必要である。同時に、指と面の接触状態を認識するために、設置面を伝搬する音を検出する必要がある。

このような要求を満たすシステムとして、本論文ではカメラとマイクを手首の内側に装着するウェアラブルな構成を提案する。本システムでは、カメラを指先の方向に向けて取り付ける事で、平面や曲面など様々な面上で指先を認識する事が可能である。

そこで、本研究では様々な面上での複雑な操作を実現する携帯機器向けインタフェースとして Anywhere Surface Touch を提案する。AST は、実環境の平面や曲面などあらゆる面上でタップやスワイプなどのジェスチャをし、入力を行う事を目標とする。具体的には、どの指でどのようなジェスチャを行ったかを認識する。また、面上で操作した時に面を伝搬する音を積極的に利用し、撫でる、軽く触る、叩くなど、面と手との接触状態の違いを認識する事を目標とする。

従来のタッチパネルのような入力では、面に接触している指先の点数で操作を識別している。しかし、面と指の接触点を認識するだけでは入力が限られてしまい、複雑な操作を高速に行う事が難しいと考えられる。そこで本システムは、以下の3つの手法によってこの問題を解決する。まず、カメラを用いて面に接触している指がどの指であるかを認識する。次に、接触時における指の動きを見る事で、ジェスチャを認識する。更に、本システムは面の材質の特性を利用し、面上で操作した時に面を伝搬する音をマイクで取得する。面には様々な材質が存在

するが、本システムはいくつかの面上で軽く触る、叩くなど、面と手との接触の仕方（以下、接触状態と呼ぶ）を認識する。これらの3つの手法によって入力自由度を大幅に向上させ、複雑な操作を実現する事を目指す。ASTに関する詳細は第4章で述べる。

最後に、仮想物体とのインタラクションにおける実体感についての研究を提示する。空中操作においては、操作を3次元的に行っているにも関わらず、ボタンなどの表示が2次元平面上に限定されてしまうため、ボタンなどのオブジェクトを直接操作しているという実体感に乏しいという課題が存在する。そこで、本研究では高速カメラを用いて手の動きを認識し、裸眼立体視ディスプレイを利用することで、立体表示したオブジェクトを直接操作することが可能なインタフェースを提案する。

さらに、仮想物体を操作する際に、高速に動く指先の位置と仮想物体との位置が同期されていないと操作の実体感が低下するといった問題が考えられる。そこで、本研究では高速ビジョンを用いて取得したデータから予測フィルタを適用することによって、従来よりも高い精度で指先位置を予測し、操作において高い実体感を実現する手法を提案する。これらの詳細は、第5章において述べる。

2.8 まとめ

本章ではまず先行研究について考察し、それぞれのコンセプトの利点と克服すべき課題について述べた。

それらの議論を踏まえて、求められる入力インタフェースのあり方を情報提示の観点から2つのシナリオに沿って示した。1つは、既存の携帯機器のような形状を維持するシナリオを想定し、空中を操作領域とすることで、より広い範囲で高精度な入力の実現を目指す。もう1つは、携帯機器がより小型化し、複雑な操作を高速に行うことが求められるというシナリオを想定し、様々な面上での複雑な操作を実現する携帯機器向けインタフェースの実現を目指す。これらについては、後の章でそれぞれ詳細に説明する。

また、空中操作における操作の実体感の乏しさを課題とし、裸眼立体視ディスプレイを用いて立体表示したオブジェクトを直接操作可能なインタフェースを提案する。さらに、仮想物体と指先位置との同期精度が低いことによる実体感の低下を課題とした。これに対し、高速ビジョンを用いて取得したデータに対して予測フィルタを適用することによってそれぞれの位置の同期精度を向上させ、高い実体感を実現する手法を提案した。これらについては第5章において詳細を述べる。

第 3 章

空中操作インタフェース：空中を操作領域とする非装着型手入力システム

3.1 はじめに

近年、小型携帯機器の技術進歩に伴ってスマートフォンやタブレット PC などが広く世の中に普及するようになってきている。また、ネットワーク技術の進歩やクラウドコンピューティングの発展により、携帯機器を用いたサービスがより高度化した。その結果、Web サイトの閲覧やメールの読み書きといった従来デスクトップ PC で行なっていたような作業を、携帯機器を用いて行う機会が増えてきている。

しかし、Siek ら [1] が指摘した Fat finger problem に見られるように、現在の小型機器においては狭い機器表面で操作を行う必要があり、細かい操作を正確に行うことが難しいという問題が存在する。この問題は、人間の身体的な制約に由来するユーザーインタフェースのハードウェア制約問題である。人間の身体は大きく変化するものではないため、この問題を解決するためには新たな入力インタフェースが必要不可欠である。

この問題に取り組んだ様々な先行研究が存在するが、それらは大きく 3 つの手法に分けることができる。1 つ目の手法は、機器装着型である。具体的には、携帯機器向けの入力操作を実現するために、何か特別な機器を操作者の身体に装着する手法 [30, 32, 23] のことである。しかし、携帯機器が日常的に利用するものであることを考えると、機器を利用するたびに装着する必要がある事は煩わしいといえる。

二つ目の手法である投影型では、何らかの映像投影装置を用いて既存の入力手法を仮想的に再現し、それによって携帯機器向けの入力インタフェースとする [39, 22, 33]。しかし、映像を投影するための平面が必要であることや、プロジェクタを設置する平面、もしくはプロジェクタを身体に装着することが必要となるため、携帯機器での利用に向かないという欠点がある。

最後の手法は、非接触型（ジェスチャー型）である [2, 10]。この手法ではユーザーが慣れ親しんだ既存の入力システムを模倣した入力システムが実現可能である。しかし、多くのジェス

チャー入力型インタフェースはPCや家電などを遠隔操作するためのものであり、携帯機器向けに細かい操作を高速に行うことを目的としていない。

上で述べたように、既存の手法の多くが特別な環境を必要とする、特殊な装置を装着する必要がある、単純な操作しか行うことが出来ない、といった課題を抱えている。そこで、新たな携帯機器向けインタフェースを設計するにあたって、以下の3つの条件を満たす必要があると考えられる。

- 設計方針として装置を体に装着する必要がない
- 使用環境を限定しない
- 細かい操作を高速に行うことが出来る

これらを満たすことにより、携帯機器の利便性を損なうことなく、より高度な操作を高速に行うことが出来るようになると考えられる。

この3つの要求を満たすインタフェースを開発するには、ビジョン技術を用いることが最も有効な手法であると考えられる。なぜならば、カメラを用いてユーザーの指の動きを広い空間で撮影することによって、特別な環境や特殊な装置を身につけることなく機器の操作を行うことが出来ると考えられるからである。

主にコンピュータビジョンの分野を中心に、手の三次元位置姿勢を高精度に推定しようとする研究は多数行われている。ある研究者たちはデータベースを用いた認識手法 [12, 58, 13] を提案し、また別の研究者たちは複数のカメラを用いたシステムによる認識手法 [15, 16] を提案した。しかし、前者は推定する手の姿勢の候補を有限個に限定してしまうという課題がある。また、後者は携帯機器において複数の方向から手を見るようなシステム構成を実現することが難しいという意味で携帯機器の手法とは言いがたい。

システム構成の観点から言うと、単眼のカメラを用いて手の3次元的位置姿勢を取得する手法が現実的である。当然、単眼のカメラを用いて手の3次元的位置姿勢を推定しようとした研究 [17, 19, 20] が存在する。しかし、手の全ての自由度を考慮してその位置姿勢を再構成するには非常に多くの計算が必要となる。そのため、限られた計算資源しか持たない携帯機器上ではこれらの手法を実行することが難しいと考えられる。

加えて、多くのジェスチャー認識システムでは、色情報を用いて手指領域の抽出を行っている。しかし、仮にユーザーが機器を片手で把持しもう片方の手を機器の手前で動かして操作すると考えると、機器の前面に取り付けられたカメラは図3.1のようにユーザーや他人の顔も撮影してしまう。そのため、肌色抽出を用いた指先領域検出が困難になる。また、指先領域を抽出する別の方法として、白色照明をカメラのフレームに合わせて点滅させ、照明が点灯している時に得られた画像と消灯している時に得られた画像の差分を取ることで指先領域を抽出するといった手法が考えられる。しかし、図3.1(b)に示すように、背景に蛍光灯が写り込んでしまうと、蛍光灯の光がサチュレーションを起こして指先領域に侵食してしまうため、指先領域を安定して抽出することが困難になってしまう。

以上の議論をまとめる。新しい携帯機器向けインタフェースを開発するにあたって、我々は単眼のカメラを用いて指の3次元位置姿勢を推定する手法を考える必要がある。また、複雑な

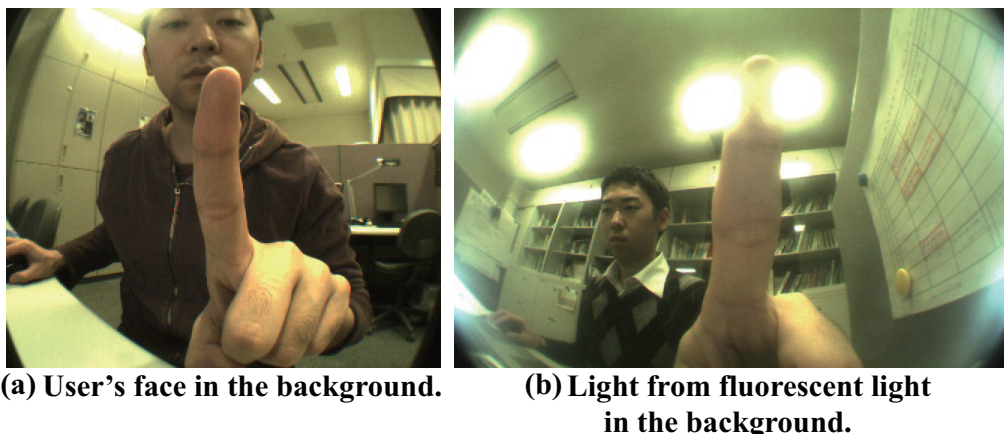


図 3.1. 指先領域抽出: (a) ユーザーの顔が背景にあることで、肌色抽出を用いて指先領域を検出することが困難となる。(b) 蛍光灯の光が写り込むことでサチュレーションが起きてしまい指先領域の検出が困難となる。

背景環境の変化に対してロバストに指先領域を抽出可能である必要がある。

この問題を解決するものとして、我々は空中操作インタフェースを提案する [59, 60, 61, 62, 63]。提案システムは単眼のカメラを用いてユーザーの指の三次元位置姿勢を推定し、認識結果を用いてポインティング操作や入力動作の認識を行う。提案システムは、特殊な機器を装着すること無く、空中で指を動かすことのみでシステムの操作を行うことが可能である。

加えて、操作を行ったユーザーに適切なフィードバックを与えることは操作性を向上させる上で重要である。特に、触覚フィードバックによって入力動作がシステムに認識されたことをユーザーに伝え、次の操作にスムーズに移行出来るようにすることは重要である。しかし、本提案システムでは、操作している指が空中にあるため、指に触覚フィードバックを提示することが困難である。そこで本システムでは、機器の操作性を改善するために、振動素子を機器本体の背面に取り付け、機器を把持している手に振動フィードバックを提示することにした。片方の手を用いて空中で操作し、機器を把持しているもう片方の手に対して振動フィードバックを与えることによる操作性への影響については評価がなされていないため、本研究では実験を通してその効果を検証した。

本研究の主たる貢献は大きく2つある。1つは Fat finger problem を解決する新たな携帯機器向けインタフェースを提案したことであり、もう1つは機器を把持している手に与える振動フィードバックの効果を検証したことである。

3.2 空中操作インタフェース

3.2.1 概要

空中操作インタフェースは、単眼の高速カメラを用いて指先の動きを3次元追跡し、空中での動作を利用したポインティングや入力が可能なインタフェースである。ユーザーは片方の手

で機器を持ち、もう片方の手の指先を機器の前面に付けられたカメラの前の空間で動かす事によって入力動作を行う。ユーザーの縦横の手の動きに合わせて画面内のカーソルが移動し、指を軽く前後に動かすことで選択操作を実行することが可能である。

本システムでは、携帯機器として一般的な単眼のカメラのみを用いて指の三次元位置を取得する。携帯機器を片手に持ち、もう片方の手で操作を行う場合、カメラと指の間の距離はおおよそ 10cm 程度となる。このようにカメラと指の距離が近い場合、カメラに映る指の大きさから奥行き相対位置を知ることが可能である。

カメラ画像から指の位置や大きさを取得するには、まず指領域と背景領域を分離する必要がある。本システムでは、赤外 LED を点滅させたものをアクティブ照明として使用し、LED 点灯時の画像と消灯時の画像の差分を取ることで指領域の抽出を行う。カメラの前面には赤外透過フィルタを設置しており、カメラには照明の反射光のみが映る。このようにすることで、白色照明を用いた画像差分だけでは残ってしまう蛍光灯の影響を軽減する。

本インタフェースでは、細かい操作を可能にする必要があるため、指の三次元位置取得において高い精度が要求される。一方で、そのような三次元位置推定は単眼カメラのみで実現可能で、かつ実時間で処理可能でなければならない。そこで本システムでは、抽出された指領域に対し、ユーザー自身の指形状を用いたテンプレートマッチングを行うことにより、位置姿勢の検出を行う。なお、ユーザーの指形状は操作を開始する際に指を所定の位置に置いて自動的に登録する。フレーム毎に前フレームの情報を利用したトラッキングを行うことで、安定した推定を実現する。

さらに、指が高速に動いてもトラッキングが外れないように、通常のカメラよりもフレームレートの高いカメラを使用する。これにより、安定したトラッキングが実現されるとともに、低遅延・高応答のインタフェースシステムが実現される。

最後に、空中での操作では、キー入力などを行った際に物理キーボードやタッチパネルにおいて入力を行った時のような触覚的なフィードバックを感じる事が出来ない。そこで本システムではその代用として、機器背面に振動素子を取り付け、キー入力時に視覚効果と合わせて素子を振動させることでユーザーへのフィードバックとする。

なお、本研究は廣部の研究 [64] を元にし、携帯機器向けのシステムで特別な機器を装着すること無く空中での操作を実現した、世界で初めてのシステムであると考えられる。

3.2.2 システム設計・開発

提案インタフェースの有効性を検証するため、PC と小型ディスプレイを用いてプロトタイプシステムを構築した。システム構成を図 3.2 に示す。PC (Intel Core 2 Quad Q9650 搭載) に接続した 4.3 インチ USB 接続サブモニタ (480 × 800 pixels, 5.3 × 9.5cm, Century 社製 LCD-4300U) の前面に小型の IEEE1394 カメラ (Point Grey Research 社製 Firefly MV) を、背面に振動素子を取り付けている。カメラには焦点距離 1.9mm の広角レンズと赤外線フィルタ、赤外線 LED4 個を取り付けた。広角レンズの利用により、画角 90° を確保した。カメラの取得画像サイズは 752 × 180pixels で、取得レートは 145fps 程度である。また、USB 接続

24 第3章 空中操作インタフェース：空中を操作領域とする非装着型手入力システム

サブモニタのリフレッシュレートは 60Hz である。振動素子は FM34F という型番の偏平型振動モータを用いた。

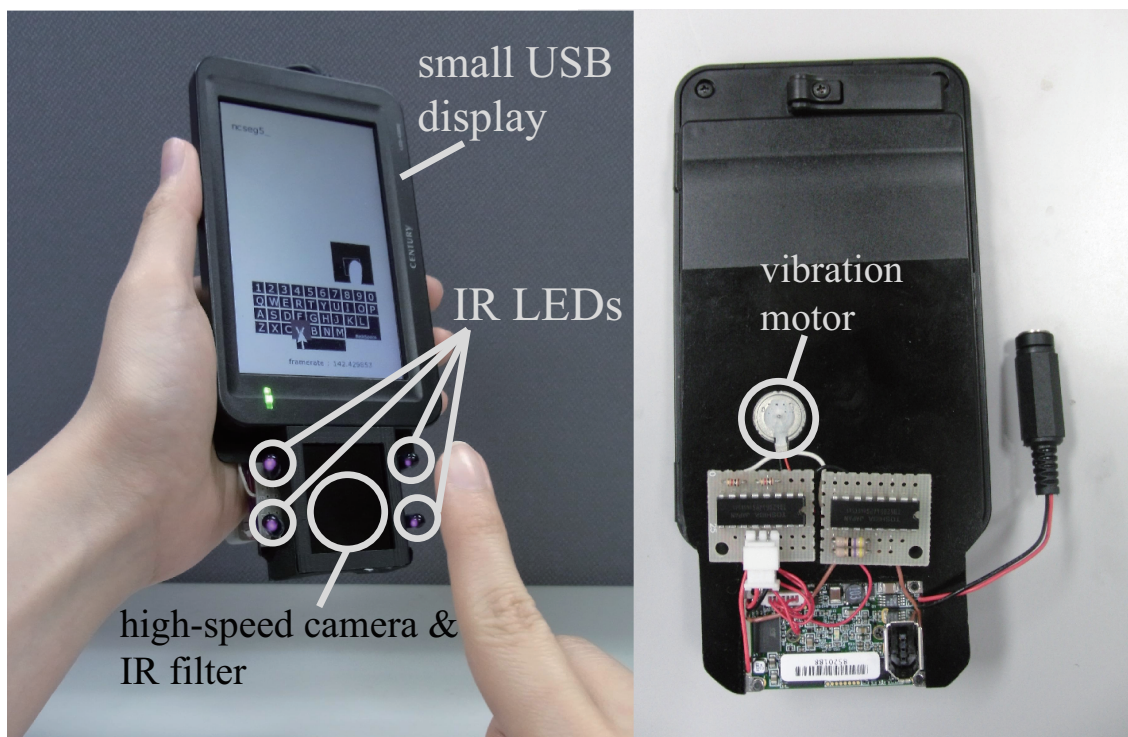


図 3.2. 空中操作インタフェースのシステム構成

3.3 指先トラッキング・入力動作認識

3.3.1 概要

ここでは本システムにおける指先トラッキング、および入力動作認識のアルゴリズムについて説明する。それぞれの処理のフローをまとめた図を図 3.3 に示す。続く節において、それぞれの処理の詳細について述べる。

3.3.2 歪み補正

本システムでは、広い視野を確保するため広角レンズを利用した。しかしこれにより、図 3.4 のようにレンズ歪みによってカメラで取得出来る画像が大きく歪んでしまう問題が発生する。この問題に対し、本システムはレンズ歪み補正を行う事で対応する。なお、画像の歪み補正は比較的計算時間を要するため、画像全体を歪み補正するのではなく、前フレームで指先があると推定された位置の近傍領域だけを歪み補正し、抽出する。

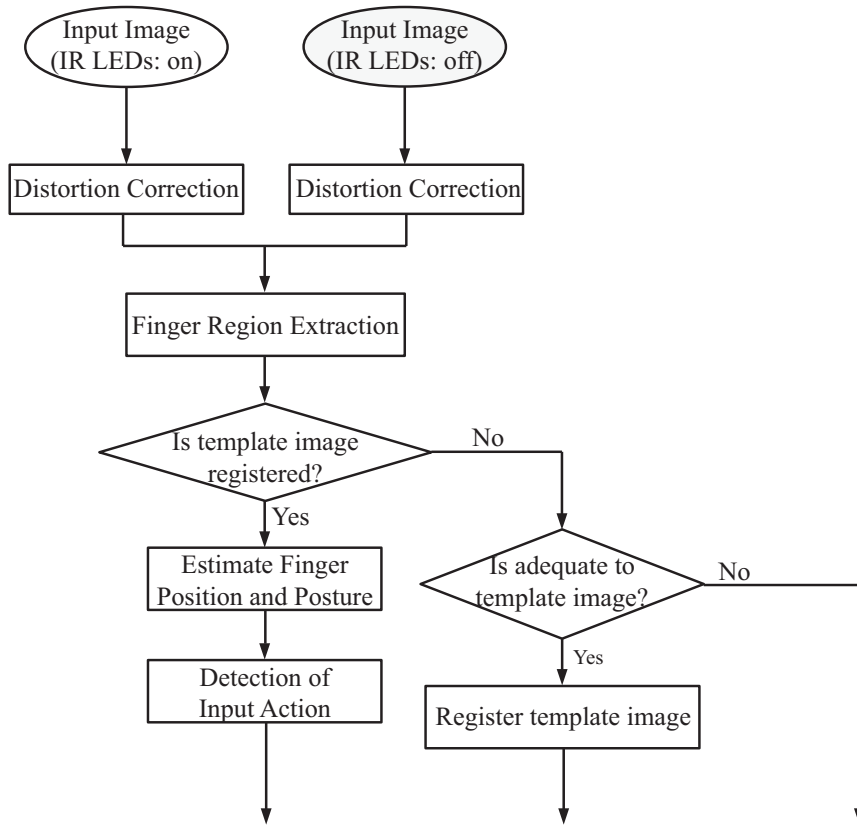


図 3.3. 指先トラッキング、及び入力動作認識の処理のフロー図

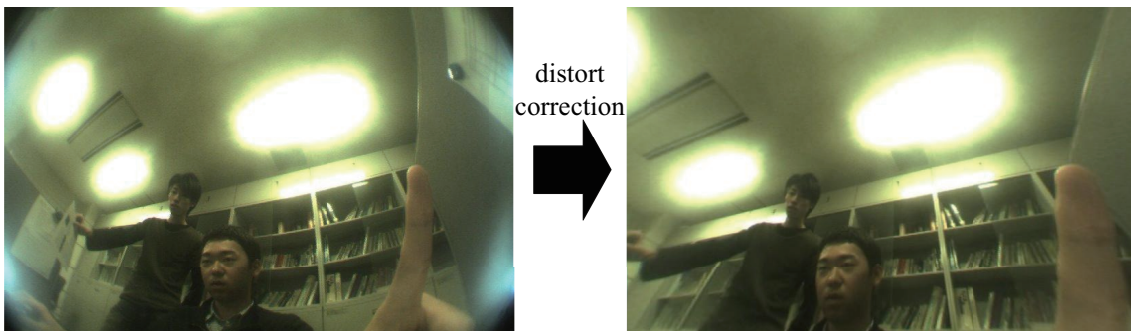


図 3.4. 歪み補正

3.3.3 指先領域抽出・指先位置姿勢推定

指の位置姿勢を推定するための前処理として、まず指領域の抽出を行う。本システムでは、赤外線 LED をアクティブ照明として利用し、指先領域の抽出を行う。赤外線 LED をカメラのフレームに合わせてオン・オフ制御することにより、連続する 2 フレームで点灯時・消灯時の 2 枚の画像を取得し、両画像の差分を取る事で指先領域の抽出を行う。図 3.5 左のように、

26 第3章 空中操作インタフェース：空中を操作領域とする非装着型手入力システム

カメラに近い物体のみ赤外線 LED で照らされるため、2 枚の画像の差分を取る事で図 3.5 右のように顔よりもカメラに近い指のみを抽出する事が可能となる。

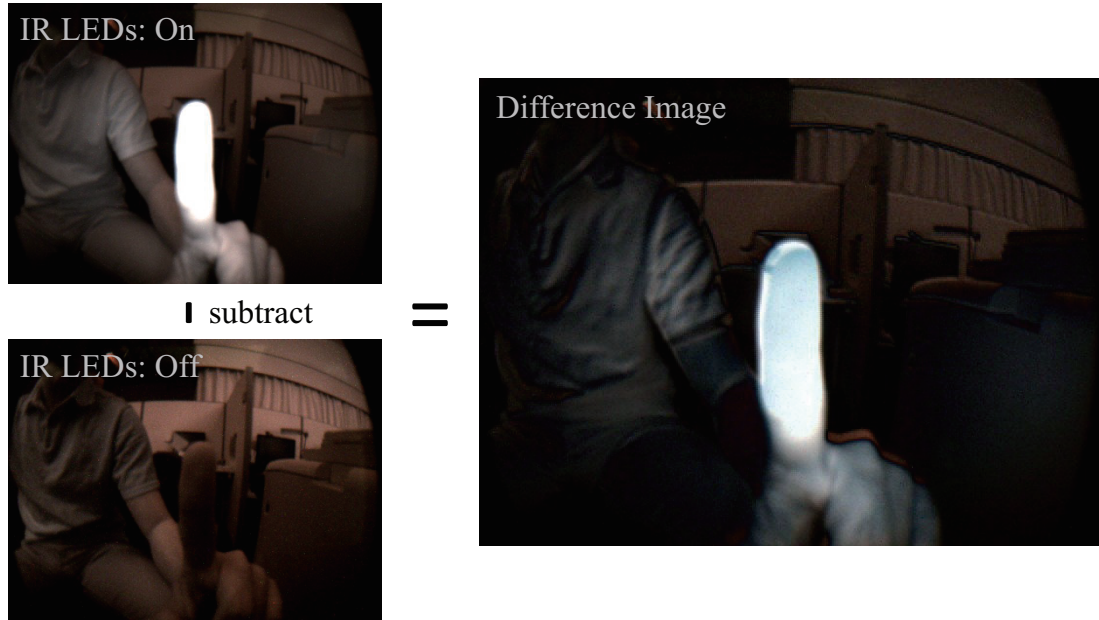


図 3.5. 赤外線 LED を利用した背景差分画像の取得

次に、抽出した指領域画像から指の位置姿勢推定を行う手法について説明する。本システムでは、テンプレートマッチングにより指のトラッキングを行う。テンプレート画像は、トラッキング開始時に自動的に取得する。図 3.6 左のように、画面には予め正方形の枠と、指先の輪郭をかたどった画像が表示される。指先をカメラの前に持ってくると、図 3.6 中央に示すように 2 値化された指先領域が白く表示される。図 3.6 右のように予め表示された指先の輪郭画像に自身の指先領域が沿うように指先を空中で動かすと、システムが正方形の枠内の指先領域の面積と重心を計算し、一定の条件を満たした時に正方形内の枠内の画像がテンプレート画像として自動的に登録される。

このようにテンプレート画像を取得する事により、予めユーザーの指先のモデルを仮定した場合と異なり、ユーザーの指の形状や大きさの違いなどに影響される事なくユーザーの指の位置姿勢推定を行う事が可能となる。

3次元空間における指の運動は、指を剛体と仮定した場合でも 6 自由度、指の曲げを考慮するとさらに高い自由度になり、画像上の指の形状はさまざまに変化する。

しかし、事前に取得したテンプレートから推定可能な指形状にはある程度の制約がある。今回は本インタフェース使用時に取りうる指の形状をなるべく広くカバーしつつ、一方で安定したトラッキングを実現するため、指の運動をカメラ光軸に垂直な平面上での並進、光軸周りの回転、光軸方向の並進に限定する。また、それらに対応する画像変形のパラメータ (t_x, t_y, θ, s) を用いて、指の位置姿勢を表現することとした。

ここで、 t_x, t_y はそれぞれ画像平面に対する x 軸、 y 軸方向の並進移動量、 s は指先画像のス



図 3.6. 指先画像のテンプレート登録の様子

ケール、 θ はカメラ光軸周りの回転量を表す。本システムでは、これらのパラメータを用いて、指の運動による画像変形を以下のように表す。

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = s \begin{pmatrix} \cos \theta & \sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x - t_x \\ y - t_y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \quad (3.1)$$

なお、 (x, y) はテンプレートにおける座標、 (x', y') はテンプレートを変形した後の座標であり、入力画像と比較される。入力画像と変形後のテンプレート画像との差分二乗和が最小になるようなパラメータを、反復解法である Lucas-Kanade アルゴリズム [65] により推定する。

具体的には、指先状態の 4 パラメータを決定し指先の位置姿勢を推定するために、以下の式を最小化するパラメータ \mathbf{p} を求める。

$$\sum_{\mathbf{x}} [I(\mathbf{W}(\mathbf{x}; \mathbf{p})) - T(\mathbf{x})]^2 \quad (3.2)$$

I はカメラから取得した入力画像、 T はテンプレート画像である。 \mathbf{p} を決定するには、まず以下の式を最小化する $\Delta \mathbf{p}$ を求める。

$$\sum_{\mathbf{x}} [I(\mathbf{W}(\mathbf{x}; \mathbf{p} + \Delta \mathbf{p})) - T(\mathbf{x})]^2 \quad (3.3)$$

(3.3) 式を $\Delta \mathbf{p}$ に関して一次テイラー展開すると、

$$\sum_{\mathbf{x}} \left[I(\mathbf{W}(\mathbf{x}; \mathbf{p})) + \nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta \mathbf{p} - T(\mathbf{x}) \right]^2 \quad (3.4)$$

となる。ここで、 $\nabla I = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right)$ は $\mathbf{W}(\mathbf{x}; \mathbf{p})$ における画像の勾配を表す。また、 $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ は、 $x' = x - x_0, y' = y - y_0$ と置くと、以下の式で表される $\mathbf{W}(\mathbf{x}; \mathbf{p})$ の \mathbf{p} に対するヤコビアンである。

$$\frac{\partial \mathbf{W}}{\partial \mathbf{p}} = \begin{bmatrix} 1 & 0 & x' \cos \theta - y' \sin \theta + x_0 & -s(x' \sin \theta + y' \cos \theta) \\ 0 & 1 & x' \sin \theta + y' \cos \theta + y_0 & s(x' \cos \theta - y' \sin \theta) \end{bmatrix} \quad (3.5)$$

次に、(3.4) 式を $\Delta \mathbf{p}$ について微分して 3.3 式を最小化する $\Delta \mathbf{p}$ を求める。

$$2 \sum_{\mathbf{x}} \left[\nabla \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^T \left[I(\mathbf{W}(\mathbf{x}; \mathbf{p})) + \nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta \mathbf{p} - T(\mathbf{x}) \right] = 0 \quad (3.6)$$

$$\Delta \mathbf{p} = H^{-1} \sum_{\mathbf{x}} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^T [T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x}; \mathbf{p}))] \quad (3.7)$$

H は以下の式で表される行列である。

$$H = \sum_{\mathbf{x}} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^T \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right] \quad (3.8)$$

この式を用いて求めた $\Delta \mathbf{p}$ より $\mathbf{p} \leftarrow \mathbf{p} + \Delta \mathbf{p}$ として \mathbf{p} の値を更新する。

これにより、指先の位置姿勢推定を実現した。変形したテンプレート画像を取得したカメラ画像上に重畳して描画した画像を図 3.7 に示す。これらの画像から、指先の運動に対して推定が精度よく行われていることが確認できる。

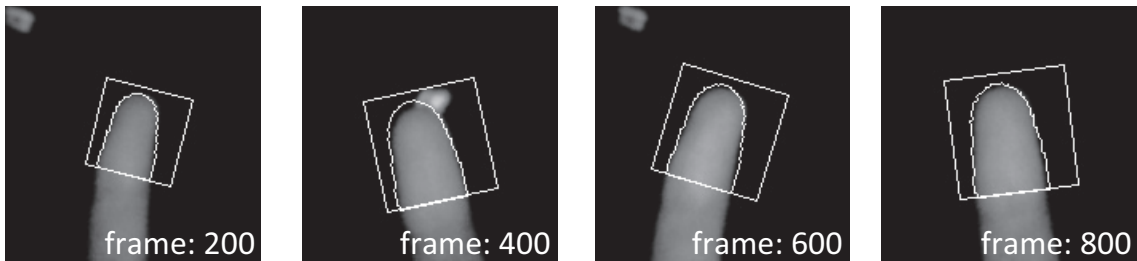


図 3.7. 指先トラッキングの様子：変形したテンプレート画像を取得したカメラ画像上に重畳している。

3.3.4 入力動作検出

本システムでは、キータイピングにおけるキーを打つ動作や、GUI 画面におけるボタンなどのオブジェクトを選択するための動作として、実際のキータイピングの動作に近い、指を軽く前後に動かす動作を採用した。以下ではこの動作を入力動作と呼ぶことにする。本インタフェースでは、指とカメラの距離が近いので、微小な指の動きでもその変化が指の大きさの変化として画像上に現れる。そのため、指先画像のスケールの変化を用いる事で入力動作を認識する事が可能である。

しかし、実際の入力動作では、カーソルを目標位置まで移動させながらタイピング動作を行うため、指の移動に伴う変化と前後運動による変化が同時に発生する。したがって、これらの変化を分離する必要がある。

本システムでは、指先のスケールの値 s の逆数である光軸方向の相対位置座標 z の変化について周波数解析したところ、指の移動による z の変化は 3Hz 以下の帯域のあったのに対し、タイピング動作による変化は 3~7Hz の帯域にあることがわかった。そこで、3~7Hz を通過帯域とするバンドパスフィルターを z に適用し、その出力値が設定した閾値を超えた場合にタイピング動作が行われたと検出することとした。図 3.8 にバンドパスフィルターを用いた入力動作検出の様子を示す。

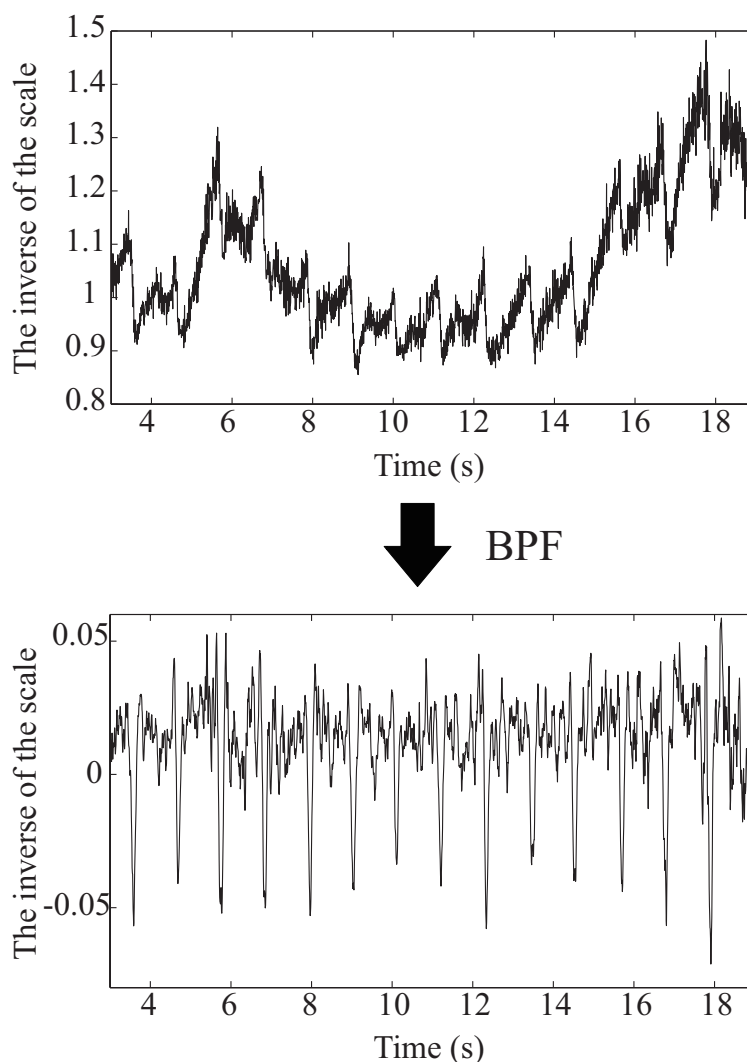


図 3.8. バンドパスフィルターを用いた入力動作検出

3.3.5 振動フィードバックの提示

一般に、インタフェースにおいて操作に対して適切なフィードバックを提示する事は、ユーザビリティを向上させる大きな要因である。フィードバックには視覚、聴覚、触覚など様々な種類が考えられるが、特に触覚情報を用いるフィードバックは視覚や聴覚に比べて環境に左右されずに知覚されると考えられ、その効果は大きいと言える。

先にも述べたように、本研究では、図 3.2 のように機器本体に振動素子を取り付けた。空中で操作している指によって入力動作が行われた際に、機器を把持している手に振動を与えることで操作性の向上を図った。なお、振動素子によるフィードバックは、入力動作が検出されたと同時に与えるものとした。また、振動の継続時間は適切なフィードバックの時間を考慮し 150[ms] とした。

3.3.6 アプリケーション例

本システムを用いて作成したいくつかのアプリケーションの例を図 3.9 に示す。図 3.9(a) は、指先の位置に合わせて動くカーソルを画面に表示し、画面上のソフトウェアキーボードを空中でタイプして文字入力を行うというものである。空中で指を軽く前後に動かすと、タイピング動作として検出される。

加えて、提案システムが指の三次元位置を認識出来る事を利用した新たなアプリケーションを2つ紹介する。1つ目は、図 3.9(b) に示すように、写真をズームしたりスクロールしたり出来る画像ビューアである。入力動作が検出されると、写真を動かすことが出来るようになる。また、再度入力動作が検出されると、写真が画面に固定される。

2つ目は三次元空間に直接線を描画出来る三次元ペインタである。三次元ペインタの様子を図 3.9(c) に示す。ユーザーは自分の指を用いて3次元空間上に直接線を引く事が出来る。また、描画した絵は、指の動きにより視点を変えて表示させることが出来る。

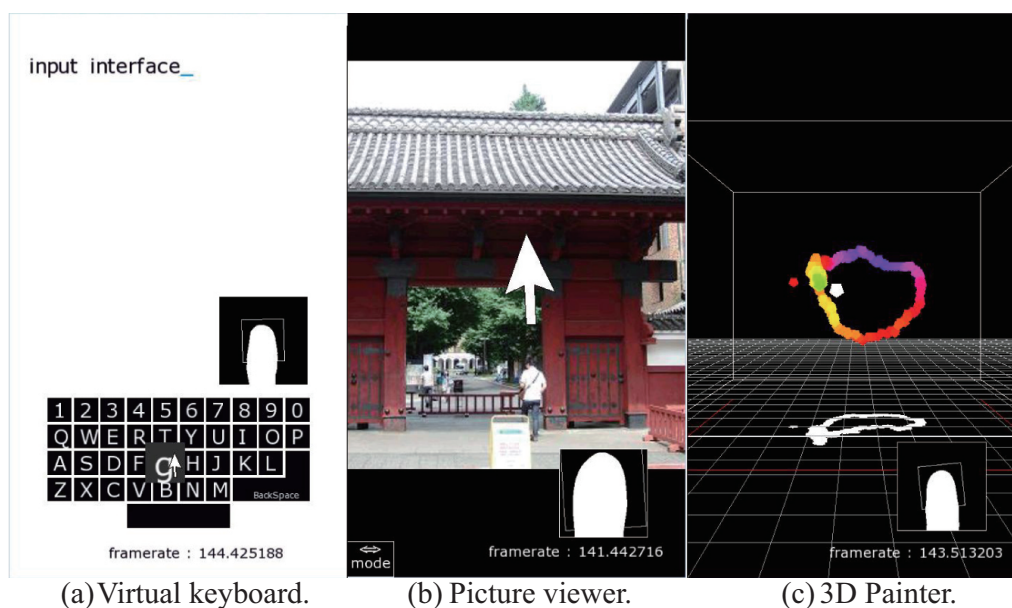


図 3.9. Sample applications.

このように、単眼カメラを用いて三次元情報が取得出来るため、携帯機器の操作領域の問題を解決するだけでなく、これまでとは異なった新たな携帯機器向けアプリケーションを実現することが可能である。

3.4 システム性能評価

本節では、提案システムの性能評価について述べる。なお、本節での性能評価において、以下の実験はシステムそのものの性能評価であるため、著者自身がユーザーとして評価を行った。

- 操作範囲
- 指先の奥行き位置認識精度
- 背景に対するロバスト性
- 計算速度

また、以下の2つの実験に関しては、ユーザーの操作に依存した評価であるため、10人のボランティア（男性8名、女性2名）に実験に参加してもらい評価した。

- 指先の位置合わせ精度
- 入力動作検出精度

3.4.1 操作範囲

本システムは広い操作領域を実現することを目標としている。そこで、本システムの空中での操作範囲を計測した。なお、本システムはカメラを用いて指先をトラッキングしているため、指先とカメラの距離によって操作範囲が異なる。今回は、指先を機器から約11.5cmの位置に起き、カメラの光軸方向に対して垂直な面内で平行移動させ、操作範囲を計測した。

実験の結果、機器からの距離が約11.5cmのところ、高さ約5.5cm、幅約12cmの操作範囲が実現出来たことを確認した。これは、ディスプレイの面積に対して約131%の大きさの操作範囲に相当する。また、ディスプレイの幅5.3cmと対応させて考えると、空中で横に1[cm]移動することが、ディスプレイ上で約4.4[mm]の移動に相当し、細かい操作が実現可能であると言える。

3.4.2 指先の位置合わせ精度

本システムは高精度な操作が可能であることを目標としている。そこで、簡単な評価アプリを用いてシステムのカーソル位置合わせ精度を評価した。

図3.10に評価アプリの様子を示す。タスク開始前には、図3.10(a)のようにディスプレイに9個のターゲットが描画される。各ターゲットは、半径10pixelの白い円の中に半径5pixelの赤い円で描画されたもので構成される。中央のターゲットの座標は、トラッキング開始時のカメラ座標の位置の中心に対応する。また、その他のターゲットはトラッキング開始時のカメラ座標の位置を基点として、カメラ座標系で上下左右方向にそれぞれ50pixelずつ移動した場所に対応する場所に描画する。また、カーソルの初期位置は中央のターゲットの中心とする。

タスクが始まると、図3.10(b)のように9つのターゲットのうち1つだけがランダムに表示される。ターゲットは2秒おきにランダムに変化するようにした。ユーザーは移動するターゲットの中央に順次カーソルの先端を合わせるようにする。ターゲットが移動する際には、画面が一瞬白く光り、ターゲットの変更があったことを視覚的に知らせた。今回は、ターゲットが次の位置に移動する直前のカーソルの位置をカメラ座標系で記録し、ターゲットとの位置の誤差を位置合わせ精度とした。なお、計測は連続して100回行った。

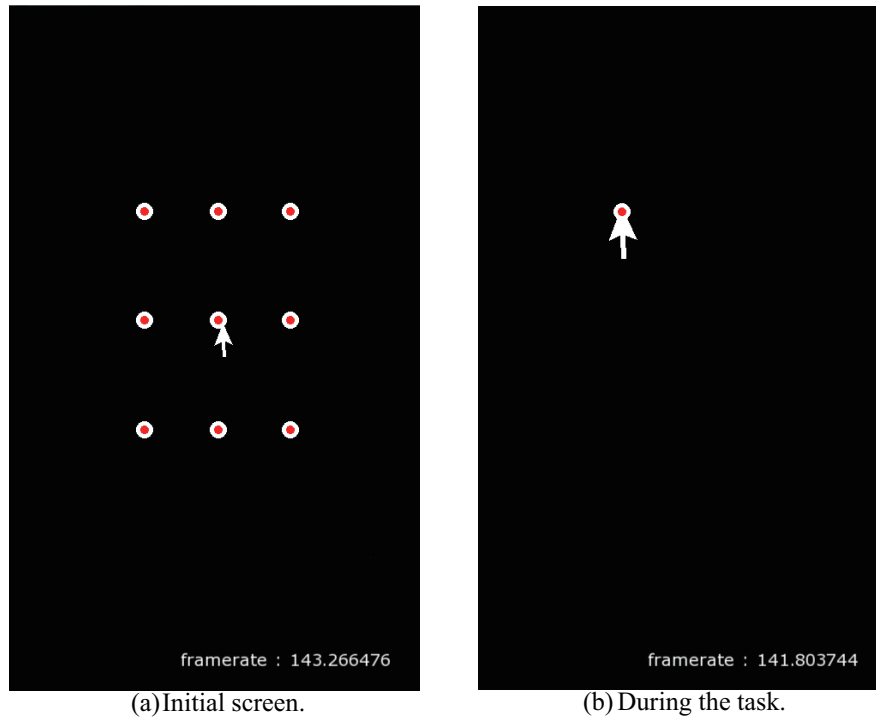


図 3.10. 位置合わせ精度計測用アプリ

実験の結果を表 3.1 に示す。実験の結果、 x, y 座標とも高い精度でカーソルの位置合わせが可能なのことがわかった。また、各マーカー位置における位置合わせ精度の偏り等の特別な傾向は見られなかった。

被験者数: 10 名 (男性: 8 名、女性: 2 名)

	x	y
平均誤差 (pixels)	-0.11	1.518
誤差の標準偏差 (pixels)	2.48	3.79

表 3.1. 指先の位置合わせ精度の実験結果

本システムでは、空中で指先を操作するため、指先がディスプレイ平面から離れた位置にある。そのため、仮に眼と指先とマーカーが同一直線上にあったとしても、両眼視によってターゲットをオクルージョン無く視認することが出来る。すなわち、Fat finger problem において問題となっていた、指先によるオクルージョンが存在しない。また、3.4.1 で示したように、本システムでは細かい操作を行うことが可能である。以上のような理由から、高い精度でカーソルの位置合わせが実現出来たと考えられる。

3.4.3 指先の奥行き位置認識精度

続いて、本システムの指先奥行き位置認識精度を評価した。実験では、まず最初に指先のテンプレート画像登録を赤外透過フィルタから 11.5cm の距離で行った。次に、カメラの光軸方向に指先を平行移動させ、機器から一定距離の場所で指先を静止させた。取得された画像において推定された指先のスケール s を記録し、奥行き方向の指先位置認識精度を評価した。

図 3.11 に実験結果のグラフを示す。赤いマーカーが各距離において推定された指先のスケールの平均値である。また、各推定値にエラーバーとしてそれぞれの標準偏差を表示した。青線及び青のマーカーは、各奥行き位置で推定されるべきスケールの理論値を示している。

実験の結果、7.5[cm] から 12.5[cm] にかけて、ほぼ正確に奥行き位置がスケールから推定できていることがわかった。一方で、13.5[cm] と 14.5[cm] では理論値から外れた結果が得られた。これは、指がシステムから離れたことで、赤外線 LED の光が十分に指先に届かず、指先に当たった光の光量が足りなくなったためだと考えられる。実際、13.5[cm] よりも 14.5[cm] の方が理論値との乖離が大きく、光量が距離の 2 乗に反比例して減衰する影響が現れていると考えられる。

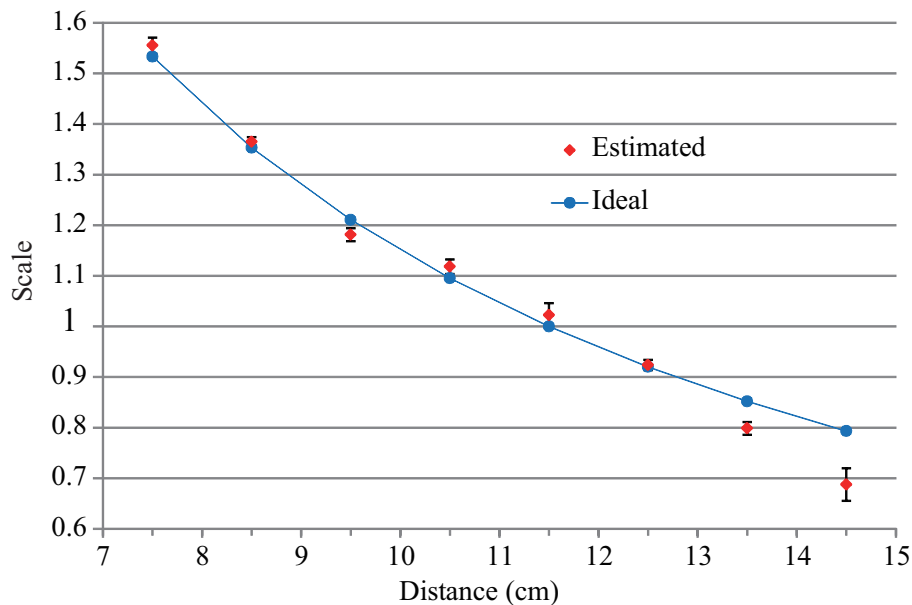


図 3.11. 各距離における奥行き位置推定精度

3.4.4 入力動作検出精度

本システムの入力動作検出精度について検証した。検証は、指先の位置合わせ精度検証実験と同じアプリを用いて行った。ユーザーは、表示されたターゲットにカーソルを合わせながら入力動作を行う。入力動作が認識された瞬間のカーソルのカメラ座標系での位置を記録し、

34 第3章 空中操作インタフェース：空中を操作領域とする非装着型手入力システム

ターゲットとの位置のずれを入力精度とした。また、計測中に誤認識された回数、及び入力動作が認識されなかった回数も計測した。ターゲットは一回の試行で20回移動することとし、入力動作に慣れるために試行を4回繰り返した。その後、5回目の試行においてデータを記録した。

表3.2に実験結果を示す。位置合わせ精度と比較すると、 x, y 座標とも平均誤差が少し大きくなっている。これは、ターゲットに対して入力動作を行う際に指を前後に動かすが、その動作の影響でカーソルも多少移動してしまうため、このような結果になったと考えられる。また、標準偏差についても x, y 座標とも大きくなっている。取得画像サイズとディスプレイの大きさをもとに考えると、検出された入力動作の x 方向の標準偏差は実世界で0.6mmに相当する。マイクロソフトがタッチパネル上でのボタンの配置などの指針を示しているGuidelines for targeting[66]では、実際のタッチ可能なターゲットの大きさは最小でも 9×9 mm以上であるべきだと述べている。また、ノキアのDesign Guideline[67]も同様に 7×7 mm以上であるべきだと記している。AppleのiOS Human Interface Guidelines[68]には具体的な大きさが言及されていないが、公開されている資料から換算するとおよそ 6.8×6.8 mm以上のターゲットで設計すべきと判断することが出来る。これらの数字と比較しても、本システムの入力動作検出精度は十分に高いということが可能である。また、入力動作の検出精度については、誤検出(第一種過誤)が6回(3%)、見逃し(第二種過誤)が12回(6%)となった。

入力動作検出は、カーソルを移動しながらの入力であるため、しばしば検出ミスが発生する。この問題はフィルタ設計の問題と考えることも出来る。しかし現実的には、高速な手指の動作と奥行き位置推定のノイズの影響により、バンドパスフィルタでは高精度な入力動作検出が難しいと考えられる。そこで、今後はより高度な時系列処理を導入することにより、指先の移動と入力動作をリアルタイムに判別することが必要になると考えられる。

被験者数: 10名(男性:8名、女性:2名)		
	x	y
入力動作の平均誤差 (pixels)	-0.25	1.49
誤差の標準偏差 (pixels)	8.60	9.80

表 3.2. 入力動作位置検出精度

3.4.5 背景に対するロバスト性

本システムは赤外線LEDをアクティブ照明として利用し、赤外透過フィルタをカメラに設置することによりロバストな指先トラッキングを実現することを目標としている。ここでは、まず背景に顔や蛍光灯などがあった場合についてのロバスト性について検証する。

図3.12に、赤外透過フィルタを設置する前とした後の比較画像を示す。この画像からわかるように、赤外透過フィルタを用いることにより、顔が背景として映らなくなっていることがわかる。また、背景にあった照明の画像上での輝度も大きく減衰していることが確認出来る。

さらに、我々はシステムのロバスト性を暗い室内や、直接照明で照らされている環境、及び日光下において評価した。直接照明で照らされている環境については、LED バー照明を機器から 60cm の距離に設置し、機器のカメラを向くように設置した。各環境下の照度はそれぞれ 5lx、2,130lx、50,000lx であった。なお、本システムを自然に利用した際のユーザーの顔と機器の距離がおおよそ 60cm 程度であるため、これ以上近い位置に照明がある環境下で利用することは通常考えにくいといえる。また、日常生活において 2,000lx を超える室内環境は相当強く照明がたかれた特殊な環境であり、室内におけるロバスト性を評価する意味では十分であると考えられる。

図 3.13 に、各環境下において指先が赤外線 LED に照らされた時の画像、照らされていない時の画像、及びそれらの差分画像を示す。暗い室内や直接照明環境下では、安定して指先領域を抽出することが出来ており、指先をロバストにトラッキングすることが可能だった。しかし、日光下では太陽からの赤外線が強く、赤外線 LED で照らされていない時でもほとんどサチュレーションを起こしてしまっている。そのため、本システムは日光下では十分な差分画像を得ることができず、指先をトラッキングすることは出来なかった。

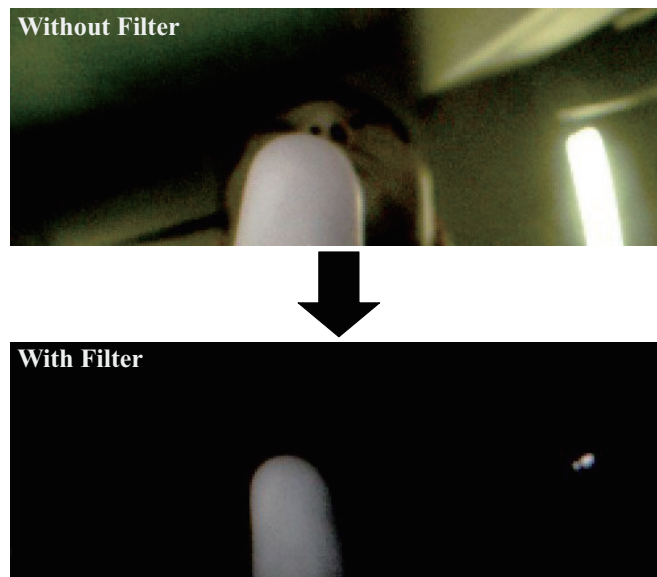


図 3.12. 赤外線 LED と赤外線透過フィルタによるロバスト性の向上

3.4.6 計算速度

本システムの画像処理について、計算速度を計測した。なお、本システムの画像処理は大きく分けて 4 つの処理に分類出来る：順に、カメラから取得した画像から推定領域のみ歪み補正をしつつ抽出する処理、指先領域を抽出する処理（差分画像生成と画像の 2 値化）、2 値化した画像に対して Lucas-Kanade アルゴリズムを用いてトラッキングを行う処理、トラッキングにより推定した指先の状態から入力動作を検出する処理である。今回は、これら 4 つの処理に

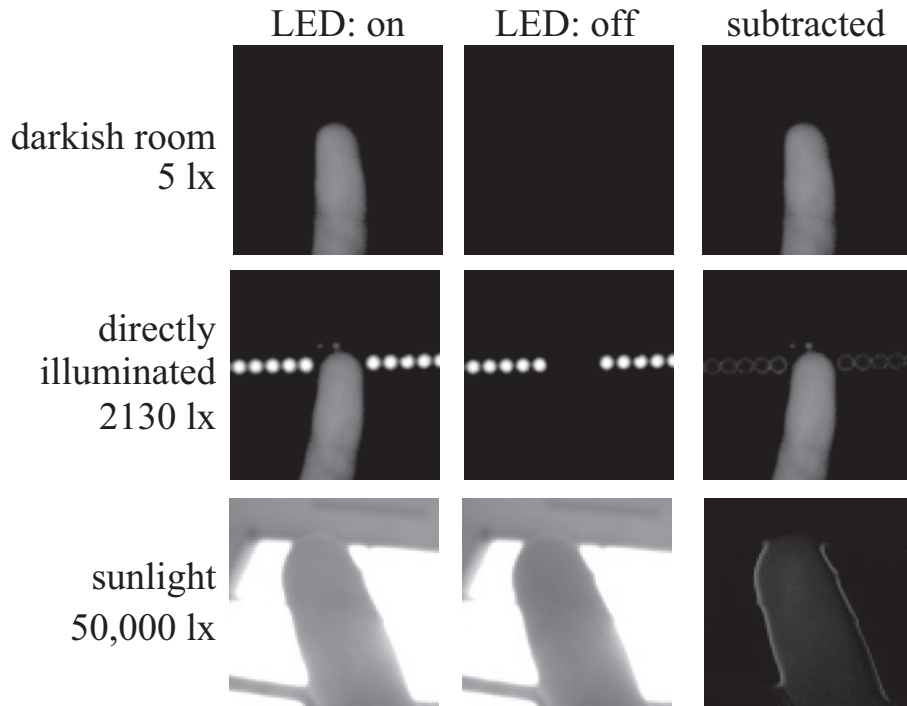


図 3.13. 各環境下におけるロバスト性の評価：左から順に暗い室内、直接照明下、日光下

ついて、実際に指先をトラッキングしている最中の連続した 100 フレームにおける処理時間を計測した。

実験の結果を表 3.3 に示す。どの処理も 5ms 以内での計算が実現されており、全体でも 7.2ms で処理が行えている。この程度の計算負荷であれば、携帯機器において実装することも十分可能であると考えられる。

処理	平均処理時間 (μ s)
歪み補正	4788.7
指先領域抽出	219.6
LK 法による指先トラッキング	2167.5
入力動作検出	2.6
合計	7160.5

表 3.3. 各処理の平均処理時間

3.5 空中操作インタフェースを用いた操作性に関する実験

3.5.1 実験1：フレームレートがシステムの操作性に与える影響に関する実験

ここでは、フレームレートとレイテンシに関する実験について述べる。空中で自由に動く人間の手の動きは高速である。したがって、その動作を安定してトラッキングするためには高フレームレートに撮像する必要がある。

しかし、仮に撮像が高速であったとしても、情報が操作者に素早くフィードバックされなければ、ユーザビリティの低下を引き起こすと考えられる。具体的には、操作者のストレスとなることや作業効率の低下が考えられる。従って、視覚フィードバックをいかに高速に与えるかということも重要であると考えられる。

本システムにおいて視覚フィードバックを高速に与えるためには、カメラのフレームレートを向上させることと、システムのレイテンシを小さくする事が必要であると考えられる。そこで、本実験ではこの2つに注目して検証実験を行った。

なお、実験の前に一つ注意しておくべき点がある。それは、フレームレートとレイテンシは必ずしも独立ではないという点である。一般に、カメラのフレームレートが向上すると、画像読み出しや転送時間が短縮する。また、撮像回数が増える事で、撮像までのレイテンシも短くなる。

システム設定

本実験では視覚フィードバックのフレームレートとレイテンシを変えて実験を行う。しかし、視覚フィードバックのフレームレートに合わせてカメラのフレームレートも低下させると、トラッキングの安定性が大きく異なってしまう、同等の条件で実験を行う事が困難となる。しかし、一方でそれぞれのフレームレートに合わせたトラッキング手法を利用すると、システムの性能が変化してしまい、やはり同等の実験条件を確保しているとは言えない。

そこで、今回カメラは常に 120fps で撮影するように設定し、それぞれの画像から計算された座標を間引いて表示する事により、仮想的に表示のフレームレートを変化させた。具体的には、120fps の他に 60fps、30fps、15fps をそれぞれ実験条件として設定した。

また、レイテンシに関しても意図的に変化させる必要がある。先に述べたように表示のフレームレートが高い場合にはレイテンシが短くなるため、各フレームレートにおいてレイテンシがどれほどの長さなのかを予備実験を通して確認する必要がある。

そこで、今回は事前に簡単なアプリケーションを用意し、予備実験を行った。アプリケーションは、認識した指先の位置に合わせて画面上にポインタが表示されるものである。予備実験では、用意したアプリケーションを使用している様子を外部から高速カメラ（CASIO 製 EX-F1）で撮影し、指およびその観測値を表示した点の両方を検出した。撮影した動画像を用いて、指が動いてから画面上でポインタが動くまでのフレーム数の平均を求める事によってレイテンシを確認した。予備実験の結果、各フレームレートにおけるレイテンシの平均は表 3.4

38 第3章 空中操作インタフェース：空中を操作領域とする非装着型手入力システム

のようになった。なお、120fps と 60fps についてはレイテンシの差がほとんど存在していな

フレームレート (fps)	平均レイテンシ (ms)
120	70.16
60	69.03
30	83.03
15	95.65

表 3.4. 各フレームレートにおける平均レイテンシ

い。これはディスプレイのリフレッシュレートが 60[Hz] であったためと考えられる。

タスク

被験者には単純なタスクを課した。被験者は、図 3.14 に示すように、まず画面中央に表示されたスタートボタンの上にカーソルを移動させる。3 秒間のカウントダウンの後、スタートボタンが消え、5 つの○で囲まれた数字 (直径 30pixel) が互いに重なる事無くランダムな位置に表示される。図 3.15 に示すように、1 番小さい数字にカーソルが重なるとその数字は消え、新たに大きな数字が 1 つランダムな位置に追加して表示される。また、表示されている数字の中で 1 番小さい数字は赤く表示される。被験者は 50 まで順に出来るだけ早くカーソルを重ねるよう要求される。本実験では、スタートボタンが消えてからタスクが終了するまでの時間を計測した。

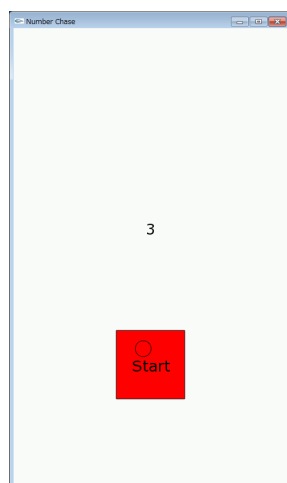


図 3.14. タスクのスタート時の様子

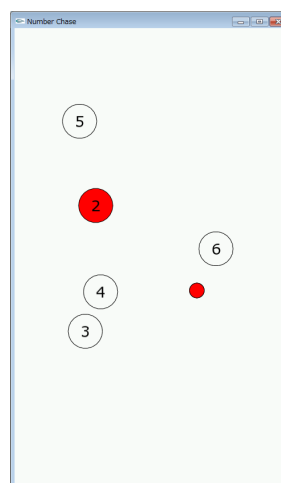


図 3.15. タスク実行中の様子

実験計画

今回の実験は、フレームレート (120fps、60fps、30fps、15fps) とレイテンシ (70ms、83.3ms、96.7ms、110ms) の組み合わせで 13 通りの実験を行った。13 通りの実験を行う順序はラテン

方格法を用いて決定し、ランダム性を確保した。なお、フレームレートが低い場合ではレイテンシの短縮に限界があるため、3つの組み合わせ（30fps×70ms, 15fps×70ms, 15fps×83.3ms）は実現する事が難しい。従って、これらの組合せについては今回実験を行っていない。

タスクを始める前に、被験者はインタフェースとタスクについて説明を受けた。5分間の練習、2分間の休憩の後にタスクを行った。練習は全員 60fps×70ms の組み合わせで行った。実験終了後、被験者にアンケートを行い幾つかの質問をした。なお、本実験には 20 代の男性 13 名が被験者として参加した。

結果

図 3.16 に各条件でのタスク終了までの時間の平均を示す。図 3.16 の枠で囲まれた 1、2、3 のそれぞれについて分散分析を行い、詳細を調べた。その結果、レイテンシは全ての枠で有意な効果がある（枠 1、2 で $p<0.001$ 、枠 3 で $p<0.05$ ）ことがわかった。また、フレームレートは枠 3 でのみ有意な効果がある ($p<0.001$) という結果が得られた。被験者に実験中何か変化を感じたかを尋ねたところ、7人が感じたと答えたものの残りの 6人は何も気づいていなかった。また、7人のうちの数人はカーソルの滑らかさなどフレームレートに関して変化しているように感じたと答えた。

framerate \ latency	70ms	83.33ms	96.66ms	110ms
120fps	21.40s	22.77s	22.39s	23.92s
60fps	22.01s	21.89s	23.79s	23.89s
30fps	-	22.18s	22.72s	23.75s
15fps	-	-	24.95s	25.40s

図 3.16. タスクを終了するまでの時間（秒）の平均

考察

レイテンシについては、全ての条件においてタスク終了までの時間に影響がある事が確認できた。しかし、被験者の半分が実験条件の変化に気づいていなかった。また、変化に気づいている被験者も、フレームレートの変化のみ気づいた被験者が多かった。

フレームレートについては、15fps ではタスク終了までの時間が伸びる影響が確認できたものの、それ以外のフレームレートでは影響は確認できなかった。60fps と 120fps についてはディスプレイのリフレッシュレートが 60fps であったため差が出なかったと考えられる。また、30fps と 60fps や 120fps についても差が出なかった。これは、人間の目の時間分解能は一般に数 10fps 前後と言われており [69]、これ以上の早いレートで描画しても知覚に影響がなかったものと考えられる。しかし、先にも述べたようにフレームレートとレイテンシは独立ではなく、フレームレートを向上させる事はレイテンシの短縮に繋がる。従って、フレームレートを向上させレイテンシを短縮させることによって操作効率を向上させる事は可能であると考

えられる。

3.5.2 実験2：振動フィードバックが操作性に与える影響に関する実験

次に、本研究では、簡単なタスクを用いて振動フィードバックが操作性に与える影響について実験を行い検証した。本実験では、入力動作時に機器を把持している手に振動フィードバックを与え、それが操作性にどのような影響を与えるかを検証した。

先にも述べたように、スマートフォンなどのタッチパネルでは入力動作を行った手にも触覚フィードバックが存在するが、本システムではそのようなフィードバックが存在しない。しかし、その代わりに機器を把持する手にのみ触覚フィードバックを与えるようにした。このような触覚フィードバックの提示は過去に検証されていないため、今回の振動フィードバックが操作性にどのような影響を与えるかを検証する必要がある。

3.5.3 タスク

本実験では、簡単なタスクを被験者に課して振動フィードバックの影響を検証した。図 3.17 に示すように、画面にはテンキーとスタートボタンが表示される。被験者がスタートボタン上で入力動作をすると、100 個の数字がランダムに画面上に表示される。被験者は表示された数字を、出来るだけ早く順にテンキーのボタン上で入力動作を行う。

また、入力された文字は画面上で赤く表示され、未入力の文字との区別がつきやすいようにした。各ボタンはクリックされると約 150ms 間赤く光り、視覚的なフィードバックを与える。また、入力ミスに対して修正動作などのペナルティは何も課さなかった。本実験では、振動はクリック動作が認識された時のみ約 150ms 間与える事とした。なお、150ms という時間については、入力動作への適切なフィードバックの長さを考慮して決定した。また、このタスクは振動あり・なしのいずれの場合にも共通するタスクである。

実験計画

タスクは振動あり、振動なしの2条件下で行った。実験を始める前に、被験者はインタフェースについての説明を受けた。被験者は振動有りの操作と振動なしの操作に慣れるため、休憩を挟んでそれぞれ数分間ずつ練習した。なお、振動フィードバックが与えられたのは、入力動作が認識された場合のみである。

その後、被験者は各条件で5回ずつタスクを交互に試行した。各タスクの試行前には、約1分程度の休憩が与えられた。また、振動フィードバック有り、無しのどちらを先に実験を行うかについては被験者ごとに入れ替え、実験順序によるバイアスを回避した。

実験の最後には、被験者に対して簡単なアンケートを行った。実験は一人あたり全部でおよそ30分程度である。なお、アンケートでは以下の項目について質問した。

- 提案インタフェースの使い方はすぐに理解出来ましたか？ (Learnability)
- 提案インタフェースの使い方はすぐに覚えられましたか？ (Memorability)

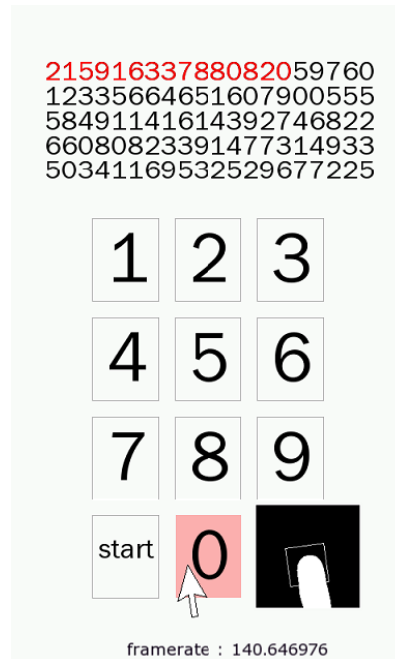


図 3.17. 振動フィードバックの効果検証用アプリケーション

- 提案インタフェースを思ったように操作することが出来ましたか？ (Task Attainment)
- 提案インタフェースが使いやすいインタフェースだと感じましたか？ (Usability)
- 提案インタフェースを操作していて楽しいと感じましたか？ (Satisfaction)
- 振動があったほうが操作しやすいと感じましたか？ (Vibration Feedback)

各項目は、-3 から 3 までの数字で 7 段階評価してもらった。(3 に近づくほど質問に対し同意している、-3 に近づくほど同意していないと考えた事に相当する。) また、実験について自由に感想を既述してもらう欄も設けた。

本実験では、タスクを完了させるまでの時間、及びミスタイプ回数(目標とするボタンとは違う場所をタイプしてしまった回数)を測定し、各タスク完了後に被験者にも提示した。

被験者

22 歳から 34 歳までの 10 名のボランティア(男性:8 名、女性:2 名)が実験に参加した。なお、この被験者は 3.4 に参加した被験者と同一である。被験者は先に 3.4 で行った実験に参加し、その後この実験に参加した。すべての被験者は実験以前に提案インタフェースを操作した経験はなく、初心者であった。

結果

まず、振動フィードバックの有無による操作時間への影響について分析した。図 3.18 に、振動フィードバック有り、無しの場合について各試行におけるタスクを終了するまでの平均時間を示す。平均して見ると、被験者はどの環境においても試行を繰り返すごとに上達し、タス

クを終了するまでの時間が短くなっていったのがわかる。

また、振動フィードバックがある場合とない場合について、タスクを終了するまでの平均時間に対して対応のある t 検定を行った。その結果、振動フィードバックがある場合 ($M=134.22$, $SD=31.55$) と振動フィードバックがない場合 ($M=141.23$, $SD=33.26$) にはタスクを終了するまでの平均時間に有意な差があることがわかった。($t(49)=-1.80$, $p = 0.05$) すなわち、振動フィードバックがある場合の方が有意に操作時間が短かったと言える。

次に、各被験者ごとの平均ミスタイプ回数を図 3.19 に示す。被験者間でミスタイプの回数にばらつきがあることが見て取れる。また、振動フィードバックがない場合に比べて、振動フィードバックがある場合の方がミスタイプの回数が少ない傾向にあることが伺える。

そこで、振動フィードバックがある場合とない場合について、一回の試行あたりの平均ミスタイプ回数に対して、対応のある t 検定を行った。その結果、振動フィードバックがある場合 ($M=4.66$, $SD=2.80$) と振動フィードバックがない場合 ($M=5.94$, $SD=3.97$) にはタスクを終了するまでの平均時間に有意な差があることがわかった。($t(49)=-2.12$, $p = 0.05$) すなわち、振動フィードバックがある場合の方が有意にミスタイプ回数が少なかったと言える。

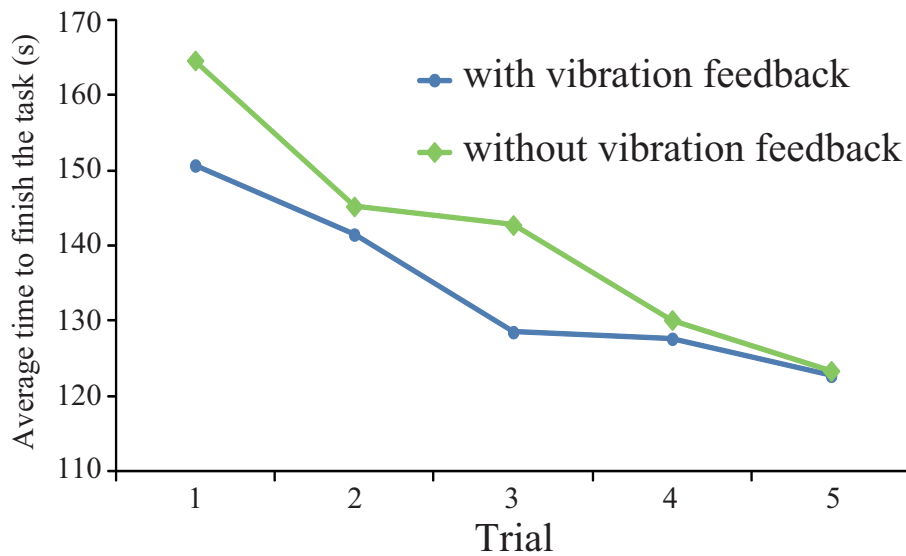


図 3.18. 各試行におけるのタスク終了までの平均時間

最後に、図 3.20 にアンケートの結果を示す。全体を通して提案システムに対する評価は好意的であるが、task attainment と usability に対する評価は他の項目に比べると低い評価となっている。表 3.5 に、アンケートの回答の標準偏差、アンケートの回答とタスク終了までの平均時間の間の相関係数、およびアンケートの回答とミスタイプの平均回数の間の相関係数を示す。task attainment と平均時間、usability と平均時間、learnability とミスタイプ回数に強い相関があることがわかる。

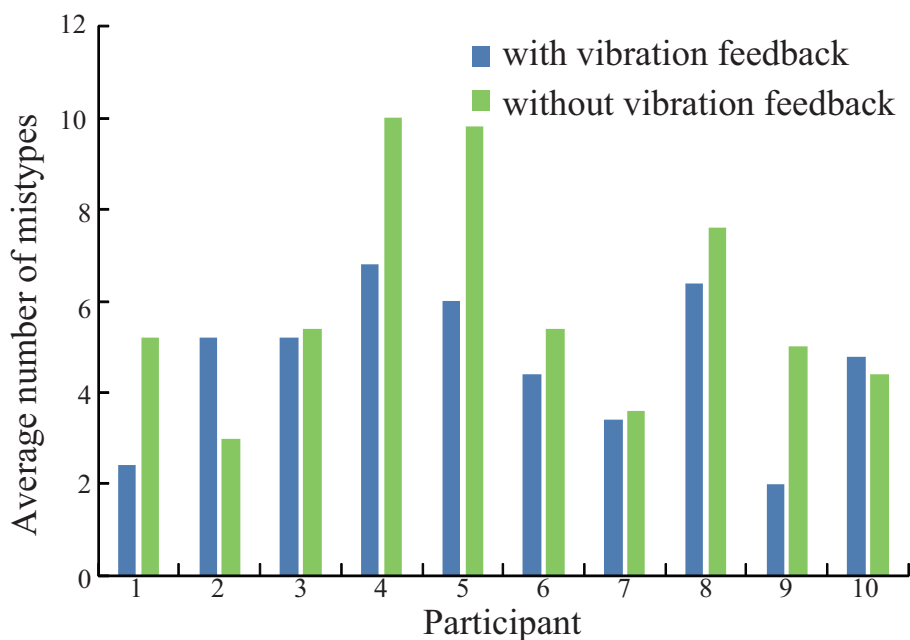


図 3.19. 各被験者における平均ミス回数

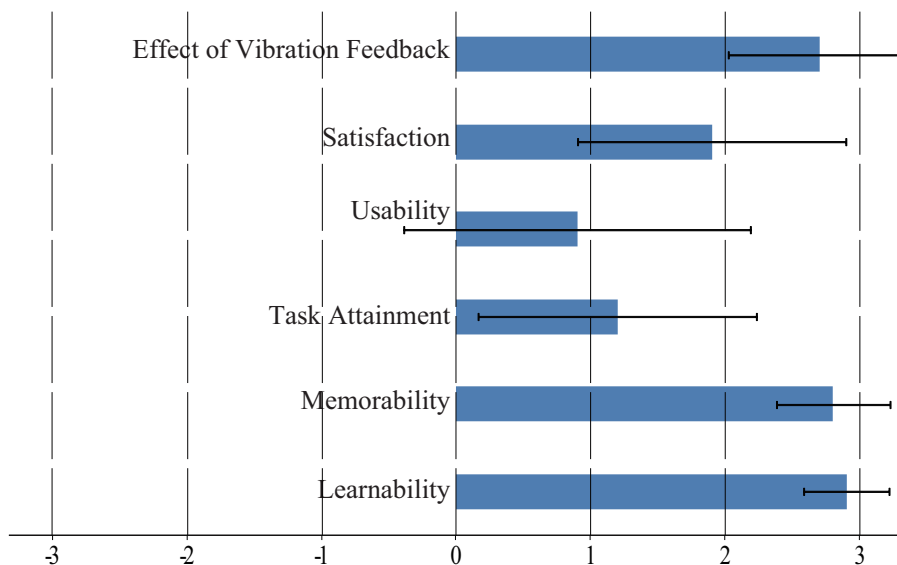


図 3.20. アンケート結果

考察

本実験では、振動フィードバックの有り無しによって、タスクを終了するまでの時間に有意な差があることを検証した。また、振動フィードバックが一回の試行あたりのミスタイプ回数を有意に減らす効果も検証した。アンケートにおいて、振動フィードバックとタスク終了までの平均時間、および振動フィードバックとミスタイプ回数の相関は弱かったが、これは被験者

44 第3章 空中操作インタフェース：空中を操作領域とする非装着型手入力システム

	標準偏差	Correlation coefficients	
		平均時間	ミスタイプ回数
Learnability	0.32	0.28	-0.60
Memorability	0.42	0.20	-0.39
Task Attainment	1.0	-0.64	0.32
Usability	1.3	-0.76	0.28
Satisfaction	1.0	-0.05	-0.13
Vibration Feedback	0.67	0.11	-0.28

表 3.5. 各アンケート項目の標準偏差、各アンケート項目と平均時間の相関係数、および各アンケート項目とミスタイプ回数回数の相関係数

全員を通じて振動フィードバックの高かったためだと考えられる。実際、アンケートにおける振動フィードバックの標準偏差を見ても小さな値になっていることが確認出来る。これらの事実から、機器を把持している手にのみ振動フィードバックを与えるだけでも機器の操作性を向上させることが可能であると言える。

振動フィードバックによりミス回数が有意に減少した理由は以下のように考えられる。被験者は、なるべく高速に入力を行うために、入力すべき数字をいくつか記憶し、その数字を連続して入力した後に入力結果を確認する。この時、振動フィードバックがある場合は、特定のボタン上で入力し損なった場合に振動が得られないため、次のボタンに移る前にもう一度押しそこねたボタンを入力することが出来る。一方、振動フィードバックがない場合は、入力している最中にボタンを押しそこねていたとしても振動によって押しそこねたことを知覚することが出来ないため、そのまま次の数字のボタンを押してしまう場合が存在する。これにより、振動フィードバックがない場合に入力ミスが増えたと考えられる。

アンケートの結果、learnability、task attainment、usability の3項目がシステムの操作性を向上させる上で重要なポイントであることがわかる。多くの被験者がシステムの操作方法を簡単に理解し、また記憶したにもかかわらず、task attainment と usability の評価はそれらの評価よりも比較的低かった。このことから、task attainment と usability の評価は入力動作の実行に関連して評価が少し下がったものと考えられる。したがって、入力動作検出手法をより改善することによって、より操作のしやすい携帯機器向けインタフェースが実現出来ると考えられる。最後に、ユーザーが振動フィードバックに対して好意的な評価をしたことも重要である。

3.6 まとめ

本提案システムでは、小型機器において細かい操作を高速に行うことの困難性を問題として取り上げた。また、この問題を扱う先行研究を紹介し、新たなインタフェースに求められる要求をまとめた。そして、それらの要求を満たす新たなインタフェースとして、単眼の高速カメラを用いて指先の動きを3次元トラッキングし空中での操作や入力動作が可能な、空中操作イ

インタフェースを提案した。また、高速な単眼カメラを用いて指先領域の安定した抽出、指先のトラッキング、入力動作の検出を実現する手法を具体的に提案した。

システム評価の結果、提案システムがディスプレイの大きさに対して131%の大きさの操作領域を実現したことを確認した。特に、水平方向についてはディスプレイの2倍以上の操作領域を実現した。また、指先位置の推定において、水平方向に平均誤差2pixel以内と高精度に行えることを示した。同様に、奥行き方向についても高い精度で推定が行えていることを確認した。この結果は、提案システムが従来よりも遥かに高精度な操作を実現したことを示すといえる。また、赤外線LEDと赤外線フィルタを用いることで、提案システムは背景に写り込んだ顔や照明に対して極めてロバストに動作することを確認した。また、暗い部屋の中でもロバストに指先トラッキングが行えることも確認した。加えて、1フレームあたり約7.2msという高速な処理も実現した。入力動作検出についても、高い認識率を実現した。

フレームレートに関する実験では、15fpsでは有意にタスク終了までの時間が増加するという結果を得た。レイテンシについてはいずれの場合においても操作時間に有意な差が生まれ、非常に小さいレイテンシの差であってもレイテンシを減少させる事が操作効率の向上に有意に影響を与えるという結果を得た。レイテンシを減少させるためにはフレームレートを高くする事も必要となるため、間接的に高フレームレートの必要性も確認できた。

振動フィードバックの実験の結果、振動フィードバックを付加することによってタスクを終了するまでの時間、およびミスタイプの回数を有意に減少させることが出来る事を確認した。従って、機器を把持している手にのみ振動フィードバックを提示することによってもシステムの操作効率を改善することが可能であるといえる。

今後はより広い操作範囲を実現するための工夫や、入力動作検出のためのより高度な時系列解析の導入などを旨とする。また、多指による入力など、より複雑で高度な入力を可能にするインタフェースの実現を目指す。

第 4 章

Anywhere Surface Touch : 実環境の面を操作領域とする装着型手入力システム

4.1 はじめに

近年、携帯電話やスマートフォンなどの小型情報端末が広く普及している。また、ネットワーク技術やクラウド技術の発展により、様々な場所でより高度なサービスを受ける事が可能になってきている。これに伴い、携帯機器を用いて文字入力やメニュー操作、あるいはゲームなど、常に多くの入力命令候補が存在するアプリケーションが利用可能になっている。例えば、文字入力であればアルファベットや記号など、常に数十以上の入力命令候補が存在する。これらのアプリケーションでは、多数ある入力命令候補を適切に選んで順に実行するといった複雑な操作を行う必要がある。

一方、理想的な携帯機器を突き詰めて考えると、機器が身体と同一化するほどに小型化し、利用を意識する事無く、いつでもどこでも自然な一連の動作で操作できる事が望ましい。ここで、小型化に伴う操作領域の縮小は大きな問題であると考えられる。携帯機器の小型化に影響を受けない操作方法の設計を導入しなければ、理想的な携帯機器がどれだけ複雑な計算や処理を行えたとしても、提供しうる機能は大きく制限されてしまう可能性がある。

この問題に対し、我々は環境の様々な面を操作領域として利用する事に着目する。携帯機器が身近な環境の平面や曲面を操作領域として利用する事が可能になれば、小型化によるメリットを最大限享受する事ができると期待できる。

そこで本研究では、机や壁、ソファなどの身近な面や身体の一部など、手を置く事ができる様々な面上で入力操作を実現する Anywhere Surface Touch(AST) を提案する [70]。AST はカメラと照明、コンタクトマイクから構成される腕時計サイズの小型システムである。ユーザーは利き腕の手首の内側に AST を装着し、機器を周囲の面に接触させて入力操作を行う。

従来のタッチパネルのような入力では、面に接触している指先の点数で操作を識別している。しかし、面と指の接触点を認識するだけでは入力が限られてしまい、複雑な操作を高速に



図 4.1. Anywhere Surface Touch のコンセプト

行う事が難しいと考えられる。そこで本システムは、以下の3つの手法によってこの問題を解決する。まず、カメラを用いて面に接触している指がどの指であるかを認識する。次に、接触時における指の動きを見る事で、ジェスチャを認識する。更に、本システムは面の材質の特性を利用し、面上で操作した時に面を伝搬する音をマイクで取得する。面には様々な材質が存在するが、本システムはいくつかの面上で軽く触る、叩くなど、面と手との接触の仕方（以下、接触状態と呼ぶ）を認識する。これらの3つの手法によって入力自由度を大幅に向上させ、複雑な操作を実現する事を目指す。

片手で手元を見ないブラインドの状態でも、例えば、ASTを利用することで図4.1のように、本を読んでいる最中や友人との会話中にちょっとしたメモを取るといった事や、ソファでくつろいでいる時に手元で身近にある機器を操作するなどの応用が可能になる。また、実世界の面に接触するという行為を積極的に利用するアプリケーションとして、壁面に掲示されたメディア（ポスターや地図）や NFC などを利用し、関連する情報の取得や操作方法、操作面の特性などを取得する形態も考えられる。このように、特定の場所では、面に触れる行為をきっかけとして、入力操作を直感的に開始することができると考えられる。更には、満員電車の中など、従来ではほとんど身動きが取れず複雑な操作が困難だった状況下でも、身体表面を入力面とする事で操作が可能になる。

本論文の主たる貢献は以下の2点である。まず、いつでもどこでも複雑で高速な入力ができる新たな携帯機器向けインタフェースとして、Anywhere Surface Touch を提案し、そのプロトタイプを実現した。また、プロトタイプを用いて様々な面上で実験を行い、身近な環境で複雑な操作が行える事を示した。

本章では、次の構成で議論を進める。まず、第2節では、新たなインタフェースの設計指針をまとめ、その上で提案システムの詳細について述べる。第3節では、提案システムが先に述べた特徴を実現した事を示すために行った実験について説明し、その結果について考察する。最後に、第4節で本論文のまとめを述べる。

4.2 Anywhere Surface Touch

4.2.1 概要

本章では様々な面上での複雑な操作を実現する携帯機器向けインタフェースとして Anywhere Surface Touch を提案する。AST は、実環境の平面や曲面などあらゆる面上でタップやスワイプなどのジェスチャをし、入力を行う事を目標とする。具体的には、どの指でどのようなジェスチャを行ったかを認識する。また、面上で操作した時に面を伝搬する音を積極的に利用し、撫でる、軽く触る、叩くなど、面と手との接触状態の違いを認識する事を目標とする。

このコンセプトを実現するには、ユーザーの指の動きや指と面の接触を常に捉えるために、画像情報を用いて認識を行う事が必要である。同時に、指と面の接触状態を認識するために、設置面を伝搬する音を検出する必要がある。

このような要求を満たすシステムとして、本論文ではカメラとマイクを手首の内側に装着するウェアラブルな構成を提案する。本システムでは、カメラを指先の方向に向けて取り付ける事で、平面や曲面など様々な面上で指先を認識する事が可能である。また、面と手の接触状態を認識するために、コンタクトマイクを面に接触するシステム下部に取り付け、設置面からの音を取得する。以上の構成の下、画像情報と音情報を統合する事で複雑な操作を認識する。

カメラは指先の方向を向いて取り付けることで、どのような平面・曲面上でも指先を認識することが可能である。また、面と手の接触状態を認識するため、コンタクトマイクをカメラのすぐ横に取り付け、設置面から音声を取得する。本システムは、視覚情報と音情報を統合することで、様々な入力を認識する。

なお、プロトタイプシステムでは、3種類のジェスチャ（タップ、スワイプ（左）、スワイプ（右））を認識する。また、システムはどの指が面と接触しているかを認識する。更に、今回は最も基本的な接触の仕方として、特定の面上で軽くタップした（指の腹でタップした）か、強くタップした（爪でタップした）かの2種類の接触状態を認識する。

従って、指の組合せは全部で31通り、またそれぞれの指の組合せに対して指の腹によるタップ、爪によるタップ、左右のスワイプの4通りのジェスチャが認識可能なため、最大 $31 \times 4 = 124$ 通りの入力が認識可能である。現在一般的に使用されている日本語キーボードのボタン数は106もしくは109であり、片手のみでもキーボード以上に複雑な操作が実行可能な自由度を備えていると考えられる。

4.2.2 ハードウェア構成

図4.2に開発したプロトタイプシステムを示す。本システムは、小型のIEEE 1394高速カメラと4つの赤外線LED、コンタクトマイク、及びPC (Intel Core 2 Quad Q9650搭載)で構成される。カメラはFirefly MV (Point Grey Research社製)を用い、取得画像サイズは600×180pixelsとした。コンタクトマイクは中央の針がピエゾ素子となっており、設置面の振動を電気信号に変換し音を検出する。なお、情報の提示には既存のディスプレイや携帯機器、

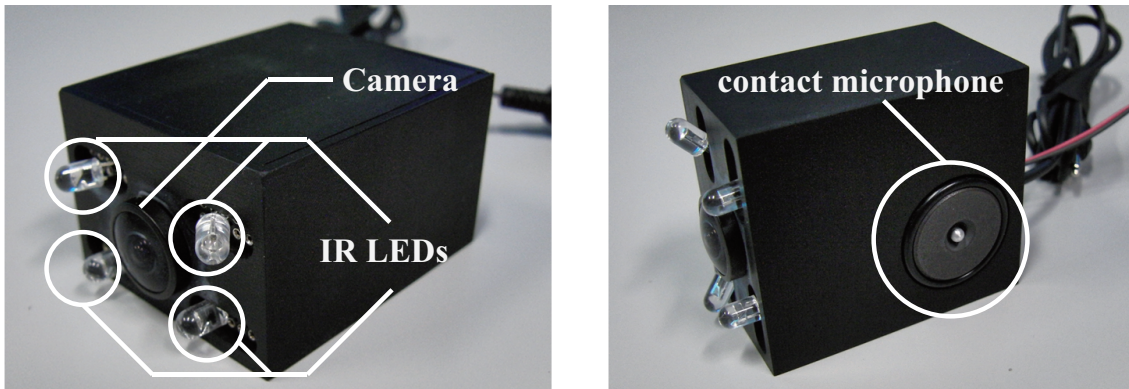


図 4.2. AST のプロトタイプシステム

または 2 インチ程度の小型ディスプレイや小型プロジェクタを利用する。

また、ピアノの打鍵などでは 1 秒間に 10 数回の入力が行われる事があり、ユーザーが高速な連打で入力することも考慮すると、カメラのフレームレートは高速である方が良い。そこで、今回はフレームレートは約 145fps(frames per second) とし、高速な入力も十分撮影出来るようにした。また、コンタクトマイクは 44100Hz でサンプリングする。本システムはマジックテープを用いて手首に固定する。

本システムでは、赤外線 LED をアクティブ照明として利用し、各フレームごとに点滅するように制御した。これにより、カメラでは設置面及び指先が照明で照らされた画像と、何も照らされていない画像を交互に得る。これらの連続する 2 枚の画像の差分を取ることで背景差分をとり、設置面と指先のみが写った画像を得ることが出来る。

しかし、手首の付け根から指全体を照らそうとすると、非常に広角な範囲を照らす必要がある。仮に、親指の第二関節から小指の付け根の長さを 10cm, 手首から親指の第二関節までの長さを 6cm とすると、左右方向に約 80° 照らす必要がある。そこで、今回は指向半値角が 30° と広く、放射強度も 80mW/sr と強い LED を使用した。また、すべての指と設置面を照らすために、図 4.3 に示すように上段の LED は左右にそれぞれ 40° ずつ傾いた向きに取り付け、主に親指や小指と左右の設置面を照らすようにした。一方、下段の LED は正面を向いて取り付け、主に人差し指、中指、薬指と設置面の中央を照らすようにした。また、カメラも焦点距離 1.9mm の広角なレンズを用い、水平方向に 120°、垂直方向に 90° 程度の画角を実現しており、すべての指を十分に撮影可能である。

なお、本システムはカメラに対してオクルージョンが発生し、指先と面との接触が撮影できない状況においては利用する事は難しい。具体的には、図 4.4 にあるように、指先が水平な面に接触する位置がシステムから 110mm とすると、図 4.4 のオレンジ色の範囲内であれば、障害物があったとしても指先と面との接触を認識する事が可能である。また、照明として LED を用いている事から、鏡面反射物体の面上では設置面をカメラで適切に撮影する事が難しく、指先と面との接触を認識する事が難しい。

AST は機器が身体と一体化し、使用していることを意識しないような小型のインターフェー

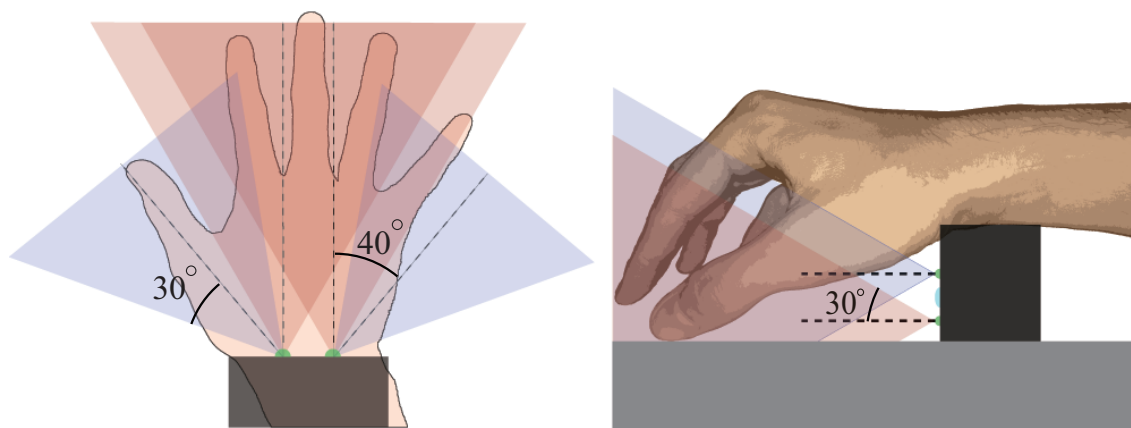


図 4.3. 赤外光 LED による照明

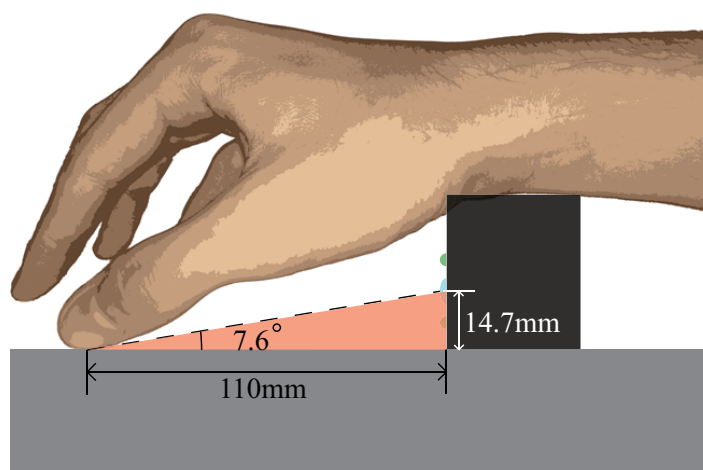


図 4.4. 障害物が存在しても指先と面との接触が認識可能な範囲

ス为目标としているが、本プロトタイプはその目標と比較すると大きい構成になっている。しかし、各センサが小型化し、一つの組込みシステムとして開発することで将来的には腕時計程度のサイズで実現可能であると考えている。

4.3 ジェスチャー入力認識

4.3.1 概要

ここでは、プロトタイプにおけるジェスチャー入力認識の概要を説明する。本システムでは、画像処理と音処理をそれぞれ独立に行う。画像処理は、得られた画像から、どの指がどのジェスチャーを行なっているかを認識し、それを記録する。一方音処理では、音の振幅の変化を元に

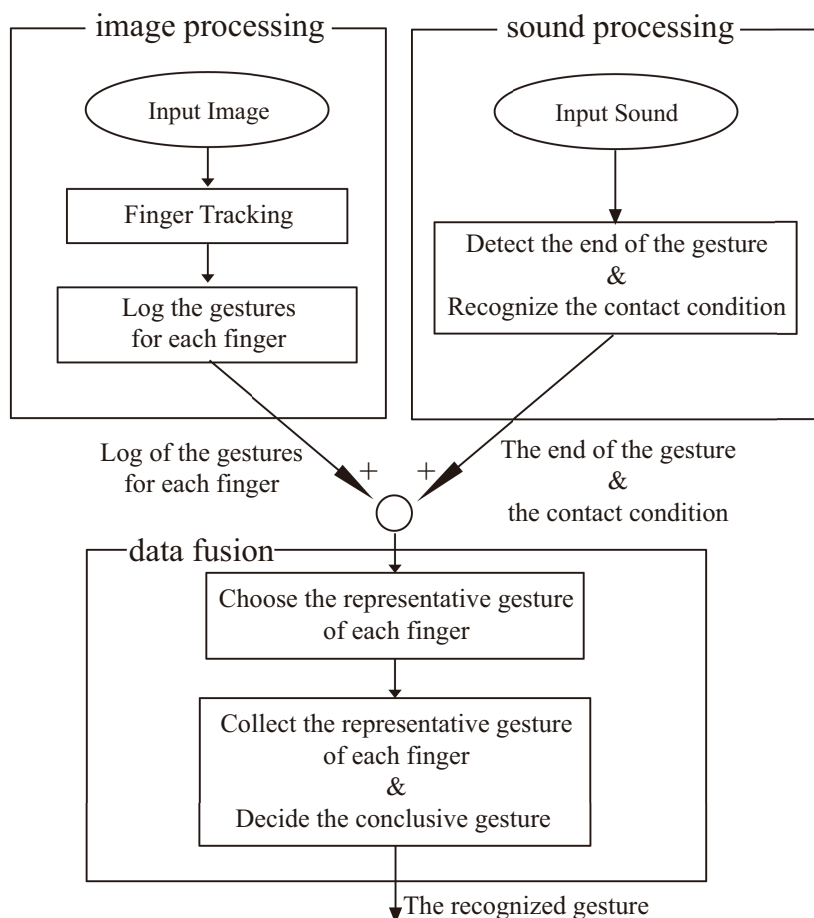


図 4.5. ジェスチャー入力の認識フロー

ジェスチャが終わったタイミングを、高周波成分の大きさを元に接触状態の認識を行う。

次に各処理の認識結果を統合する。具体的には、音処理においてジェスチャ終了を検出した時点で、画像処理によって記録された各指のジェスチャのログを確認し、各指のジェスチャ候補を一つ決定する。最後に、各指のジェスチャ候補の情報を整理し、ジェスチャの種類、ジェスチャを行った指、接触状態を決定する。

4.3.2 指先トラッキング

ここでは、カメラを用いた指先位置トラッキング、及びジェスチャ認識について説明する。指先トラッキングのための前処理として、先にも述べたように、本システムでは赤外線 LED をアクティブ照明として利用し、差分をとる事で指先と設置面の領域を抽出する。また、ここで背景差分をとった画像を 2 値化する。なお、予めカメラの視野を設定する事で手のひらが写らないようにし、各指が分離した状態で撮影する。

本システムでは、以下のように指先トラッキングを行う。なお、各指が設置面にタッチする



図 4.6. 接触面に右手の薬指が触れている時に得られる画像：タッチしている薬指の重心位置のみ異なる色で表示されている。

以前には、各指領域について、フレーム $t-1$ における重心位置 $(px_i, py_i)(i = 1, \dots, 5)$ が認識されているとする。図 4.6 に、右手での操作時に薬指が設置面にタッチしている様子を示す。各指領域内の点は各指の重心位置を示す。タッチしている薬指の重心位置のみ異なる色で表示されている。フレーム t において指が設置面にタッチした場合、画像ではタッチした指領域と設置面領域が連続した領域として表れる。従って、ラベリングされる領域数が 6 (設置面 + 指の領域数) 以下に減少し、タッチした指の重心位置の情報が失われる。これに基づき、本システムでは以下の順序でタッチ認識、指のトラッキングを行う。

1. 領域面積が最大の領域を設置面領域 (とタッチした指先) として領域番号を 0 とする
2. 得られた残りの領域についてそれぞれ重心位置 $(x_{j,t}, y_{j,t})(j = 1, \dots, n; n \leq 5)$ を計算する。
3. もし $n = 5$ なら、 $x_{j,t}$ が昇順になるようにソートし、順に (px_i, py_i) の値を更新する。
4. もし $n < 5$ なら、それぞれ (px_i, py_i) とのユークリッド距離 d_{ij} を計算する。 d_{ij} が一定以下となる (i, j) の組が見つかった場合、 j 番目の領域は i 番目の指に対応した領域を持つとし、タッチしていないと認識する。また、 (px_i, py_i) を $(x_{j,t}, y_{j,t})$ の値で更新する。
5. もし d_{ij} が一定以下となる (i, j) の組が見つからない i 番目の領域があった場合、 i 番目の指は設置面領域にタッチしていると認識する。また、この時 (px_i, py_i) は更新しない。

タッチした指に関しては、タッチ以前の指領域の重心位置情報は上記のように更新しない。しかし、もしタッチしていた指が設置面から離れれば、再び指領域は設置面と異なる領域として表れ領域数が 6 となるため、先に述べた手法で各指のトラッキングが行われる。

最後に、トラッキングした指位置の情報を用いたジェスチャ認識について述べる。ここで認識すべき事は、どの指がジェスチャを行っているかと、どのようなジェスチャを行っているかである。予備実験の結果、タップ時にはタッチした指以外の指はほとんど左右に動かないのに対し、スワイプ時にはタッチしていない指も左右に大きく動く事がわかった。そこで、本システムでは、主に 2 つの情報をを用いてジェスチャ認識を行う。一つは、どの指が設置面と接触しているかどうかであり、もう一つは、設置面に接触していない指がどのような動きをしているか、である。

ここで、説明のため設置面に接触している指の集合を Ω とする。すなわち、もし i 番目の指

が設置面に接触していた場合、 $i \in \Omega$ と表記する。システムは、 $i \notin \Omega$ を満たす i 番目の指について、重心の x 方向の速度 v_i を求め、その平均 v を求める。

次に、すべての指 $i(i = 1, \dots, 5)$ について以下のようにジェスチャーの判定を行う。

1. $i \notin \Omega$ の場合、その指ではジェスチャーは行われていないとする
2. $i \in \Omega$ かつ $v > v_{threshold}$ の時、その指は右にスワイプしていると認識する
3. 同様に、 $i \in \Omega$ かつ $v < -v_{threshold}$ の時、その指は左にスワイプしていると認識する
4. $i \in \Omega$ かつ $-v_{threshold} < v < v_{threshold}$ の時、その指はタップしていると認識する

このようにして各指に対してジェスチャーの判定を行い、それらのログを記録する。

4.3.3 音処理

音処理では、ジェスチャーが終了したタイミングと接触状態を認識する。

最初に、ジェスチャー終了の検出について述べる。詳しくは後述するが、画像処理の結果と音処理の結果を統合するために、同じ現象、すなわちジェスチャー入力を画像処理、音処理がそれぞれ独立に認識する必要があるため、音処理においてもジェスチャーを認識する。

本システムに取り付けたマイクは設置面の音を直接取得する事ができるため、ユーザーがタップやスワイプなどのジェスチャーを行った際には、何も行われていない時と比較して大きな音が聞こえる。そこで、本システムでは予め設定した値よりも大きな振幅の音が一定時間以上連続して検出された場合に、ジェスチャー入力が行われたと認識する。なお閾値とする時間は、予備実験として様々な環境でジェスチャー入力を複数回計測し、それらの認識結果のうちで最も短かった時間を下回る時間として、10ms を設定した。

また、本システムでは音によるタップとスワイプの区別は行わず、大きな振幅が収まるまでを一つのジェスチャーとし、振幅が収まった瞬間にジェスチャーが終了したと認識する。

次に、接触状態の認識について説明する。本システムでは、接触状態の認識として、特定の面上で軽いタップ（指の腹）と強いタップ（爪）の認識を行う。図 4.7 に、机（金属）の上において指の腹でタップした場合と爪でタップした場合のスペクトログラムを示す。図 4.7 のように、指の腹でタップした場合に比べて、爪でタップした場合の方が高周波数成分を強く含んでいる事がわかる。

そこで本システムでは、256 サンプル（5.8 ミリ秒, 172Hz）ごとに高速フーリエ変換を適用し、各周波数成分の大きさを取得する。得られた成分のうち、11025~22050Hz の和を求め、ジェスチャー終了が検出されるまでに、その和が予め設定した閾値を超えた場合は爪でのタップが発生していると認識する。また、データのオーバーラップはなく、窓関数にはハン窓 ($\omega = 0.5 - 0.5 * \cos(2\pi n/256)$) を利用した。なお、次の段落以降で詳細に述べるが、爪によるタップが認識可能な面においては、指の腹によるタップの場合は 11025~22050Hz の信号成分をほとんど含んでいない。そこで、予備実験の結果を元に、適切な値を接触状態認識のための閾値として設定した。

なお、接触状態の認識は、接触面の音の伝搬特性によって認識可能な面と不可能な面が存在

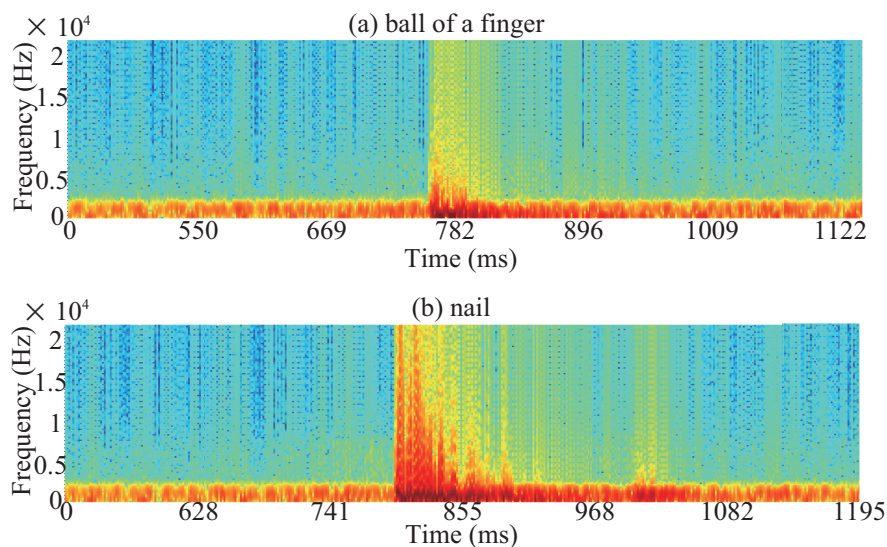


図 4.7. 指の腹と爪によるタップのスペクトログラム

する。今回は、予備実験として、比較的身近な材質からなる机（金属）、木材、プラスチック、ゴム、発泡スチロール、コルク、クッション、脚の 8 種類の面上で指の腹と爪によるタップを各 10 回ずつ行った。なお、机は厚さ 2cm 程度の金属板で、木材、プラスチック、ゴム、発泡スチロール、コルク、クッションはそれぞれ厚さ約 1cm 程度の板を用意し、机の上に置いてタップを行った。

図 4.8 に、11025~22050Hz のみを抽出した信号の、タップによる振幅のピークの平均・標準偏差を示す。この結果から、机（金属）、木材、プラスチック、ゴムの上では指の腹によるタップと爪によるタップにおいて 11025~22050Hz のみの信号の振幅が大きく変化するため、タップの識別が可能と考えられる。一方、発泡スチロール、コルク、クッション、脚の上では、異なるタップを行っても 11025~22050Hz のみの信号の振幅はほとんど変化しないため、タップの識別は困難と考えられる。

今回は比較的身近な材質からなる 8 種類の面を用いて網羅的に予備実験を行い、それぞれの面の音伝搬特性を検証した。なお、現実の面は複数の材質が混在している、中が空洞になっているなど物質の材質や形状が様々あり、それらによっても伝搬特性が変化するため、一般的な議論は難しいと考えられる。従って、面の音伝搬特性が不明な場合には、数回指の腹と爪によるタップを行い、音伝搬特性を確認する必要がある。

4.3.4 画像処理と音処理の統合

本システムでは、画像処理によって認識されたジェスチャの種類や入力している指と、音処理によって認識された接触状態の情報を統合する。この統合処理によって、もしどちらか一方でのみ入力が誤検出されても、もう片方の処理で入力が検出されていなければ、システムは最終的な入力として採用しないため、誤入力を防ぐ事が可能である。ここでは、その統合手法に

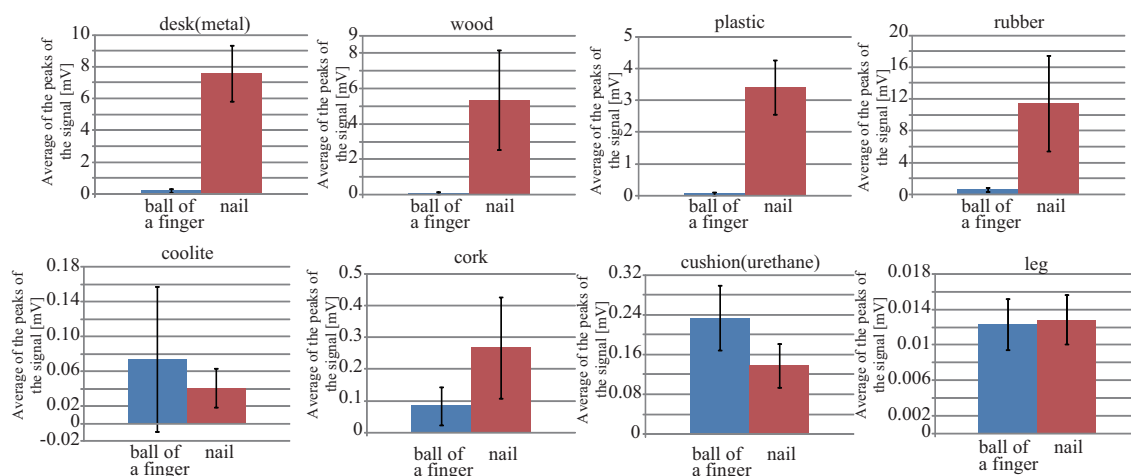


図 4.8. 各環境でのタップによる 11025~22050Hz の信号のピークの平均・標準偏差

ついて述べる。

本システムでは、画像処理と音処理はサンプリングレートが異なるため、それぞれ独立に処理を行なっている。しかし、Atrey[71]らが論文で述べたように、これらの融合において最も問題になるのは、それぞれの処理結果の同期をいかに取るかである。

そこで予備実験として、どのタイミングで情報の統合を行うべきかを検討するための簡単な実験を行った。実験は、画像処理及び音処理を独立に実行した状態でタップやスワイプを行い、それぞれの処理における認識結果、及び認識された時刻のログを記録し、オフラインで解析した。解析の結果、いずれのジェスチャにおいても、音処理によってジェスチャが終了したと認識されるタイミングは、画像処理において全ての指が設置面から離れたと認識された時刻から 30 ミリ秒後までの間に行われる事がわかった。これは、音でジェスチャ終了を検出する際に、一定時間大きな振幅が収まっている事を確認しているため、その分の遅れが現れたと考えられる。

そこで、本システムでは音処理によるジェスチャ終了検出をトリガーとして情報の統合を行う事とした。具体的には以下の通りである。まず、音処理によってジェスチャ終了が検出されると、システムは画像処理によって記録された各指のジェスチャ認識のログを確認する。例えば、もし人差し指においてジェスチャが記録されていた場合、人差し指のジェスチャ候補を決定する。ジェスチャ候補は、画像処理によって記録されていたジェスチャのログの中で、最も出現回数の多いジェスチャをジェスチャ候補とする。

なお本システムでは、一度ジェスチャ候補が決定されるごとに、ログにその情報を記録する。これにより、新たなジェスチャが行われた際にはシステムは以前ジェスチャ候補が決定された時点まで遡って新たなジェスチャ候補を決定する事が可能である。

4.3.5 ジェスチャー入力の決定

最後に、各指のジェスチャー候補の情報を整理し入力を決定する。例えば i 番目の指でスワイプをしていた時に意図せず j 番目の指で少し設置面に接触してしまった場合に、 j 番目の指がタップしたと誤認識される事や、二本指でスワイプをしていたと誤認識される事を防ぐための処理である。

具体的には、以下の2つの条件を満たすものはジェスチャーとして認識しない。

1. ある指がスワイプと認識されている時にタップと認識されている場合
2. ある指のジェスチャーのログが記録されていた回数が、別の指のログが記録されていた回数に比べて十分短い時

1. はスワイプとタップが同時に発生する事はないという理由からである。また、2. はある指が間違えて一瞬設置面に触った場合は、ジェスチャーをしようとしていた指に比べて設置面に触れていた時間が短いだろうという理由からである。

このようにしてジェスチャーを行っていた指とジェスチャーの種類を決定し、最後に、音処理によって認識された接触状態の情報を加え、最終的なジェスチャーの認識結果とする。

4.3.6 アプリケーション例

Anywhere Surface Touch によって、様々な新しいアプリケーションを実現する事ができる。AST の特徴は、複数指の組み合わせやジェスチャー、接触状態の違いなどを組み合わせる事で、多くの種類の入力が認識可能な事である。そこで、実装したものも含め、そのようなアプリケーション例を幾つか紹介する。

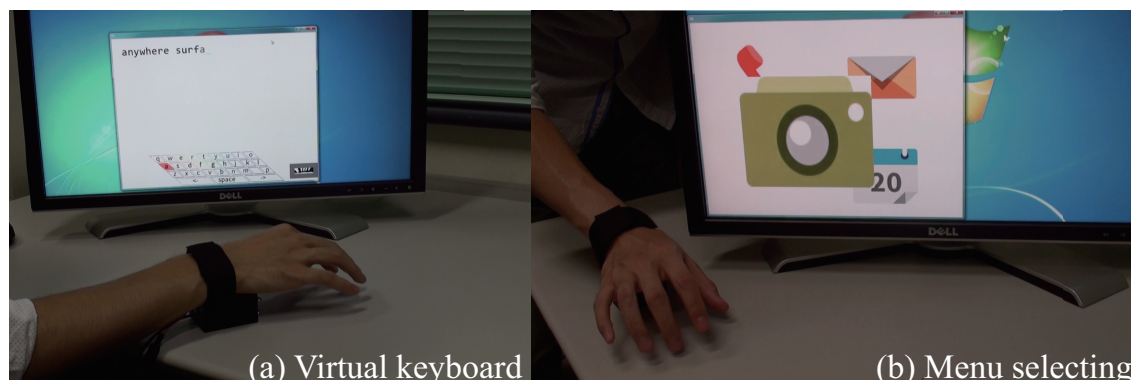


図 4.9. アプリケーション例

まず一つ目のアプリケーションは、複数指の組み合わせによる入力である。我々が携帯機器で行う文字入力やメニュー選択、画像閲覧などは常に多くの入力命令候補があるが、目的の入力を出来るだけ少ない命令数で行える事が望ましい。それに対しプロトタイプは最大 124 通

りもの入力が認識可能であり、少ない命令数で目的の入力を行うことが可能である。

図 4.9(a) に文字入力アプリケーションを示す。本アプリケーションでは、文字入力を 1 本もしくは 2 本の指によるジェスチャ入力によって、すべてのアルファベットとスペース、バックスペースの入力を実現した。ジェスチャと指の組合せはまだ数多く残っているため、その他句読点や記号などの入力に割り当てる事も可能である。

記憶するまでに時間が必要ではあるが、ブラインドでの入力が可能であるため、身近な面や身体を利用してちょっとしたメモを取るといったインタラクションシナリオが考えられる。

AST のもう一つの特徴を生かしたアプリケーションとして、アイコンやメニューなどの選択操作を紹介する。例えば、通常の操作とは別に、最終的な決定を行う場面で間違いのない入力が必要となるが、接触状態の違いを利用する事でそのような入力を実現できる。その一例として、図 4.9(b) のメニュー操作では、人差し指のスワイプで目的のアイコンを探し、アイコンを爪でタップした場合のみ選択されるようなアプリケーションを実装した。

最後に、将来的なアプリケーションの可能性を提案する。まず、AST に新たなセンサを加える事で、システムが設置面と接触する事自体をアプリケーションの一部とする事ができる。例えば、光センサを追加する事で、システムそのものを面上で動かす事でマウスのような機能を追加する事が可能である。

また、QR コードや NFC を利用する事で機器が接触した時に設置面から情報を取得するといった事も考えられる。例えば、ソファの肘掛けのある場所に AST を置くと、NFC からテレビの操作方法の取得が行われ AST によるテレビの操作が可能になり、また別の場所に置くと今度はオーディオ機器の操作が可能になる、というような事が実現できる。また、本の文字や画像を手がかりとする事で、本の特定の箇所に仮想的にメモを記録するといった事も考えられる。

4.4 システム評価実験

最後に提案システムの評価実験を行う。本システムは、画像情報を積極的に用いた入力としてジェスチャ認識、音情報を用いた入力として接触状態認識がある。そこで、これらの認識率をそれぞれ評価するため、様々な面上で入力を行い、その認識率を評価した。

4.4.1 タスク

本実験では机（金属・水平面）、クッション（ウレタン・水平面）、壁（木材・垂直面）、脚（人体・カメラから見て横方向にゆるく湾曲）の、材質や形状が異なる 4 つの環境で実験を行った。各実験環境の様子を図 4.10 に示す。なお机と壁は爪によるタップが認識可能な面、クッションと脚は爪によるタップが認識不可能な面の代表と考え、それぞれ面の材質や形状、操作時の姿勢が異なるようにした。また、壁は厚さ 1cm 程度の檜の木材を金属製の壁に貼り付けて実現した。また、どの実験環境においても、被験者は椅子に座った状態で入力を行った。なお、環境によって操作しやすい姿勢が微妙に異なるため、被験者には自由に椅子の高さを調整

してもらった。

なお、画像の2値化における閾値は、以下の手法でキャリブレーションし、自動的に決定した。まず、予め机の上で操作を行い、操作しやすい閾値 θ 、及びその時の設置面領域の面積 S を記録する。次に、各環境において面から少し指を浮かせた状態で姿勢を維持させ、設置面領域の面積を求める。なお、2値化のための閾値の初期値は θ とする。得られた設置面領域の大きさを計算し、大きさが S の上下5%以内に収まるように閾値を変更する。設置面領域が S の上下5%以内に収まるまで閾値の修正を行い、修正が終わった際の閾値をその環境における閾値とした。

また、音によるジェスチャ認識のための閾値も、あらかじめ環境のノイズを1秒間取得し、環境ノイズの音の大きさの2倍を閾値とした。爪によるタップ認識のための閾値は環境によらず一定とした。

実験で行うジェスチャはタップ（指の腹）、タップ（爪）、左右のスワイプとした。また、入力に用いる指の組合せは、各指1本ずつに加え、親指と人差し指、人差し指と中指、中指と薬指、薬指と小指の組合せを含む9通りとした。すなわち、本実験で行う入力は、9(指の組合せ) \times 4(ジェスチャ)=36通りである。ただし、先の予備実験で示したように、脚及びクッション上では爪でタップしても高音を出す事が難しいため、これらの環境では爪によるタップを要求せず、指の腹でタップするよう指示した。

本実験では、22インチのディスプレイにタスクを表示し、それを被験者が見ながら入力動作を行う。図4.11に、ディスプレイに表示したタスクの画像を示す。画面中央に手の画像が表示されており、入力すべきジェスチャーがそれぞれの指先に表示される。具体的には、指先に緑色の○（丸）が表示された場合はタップ（指の腹）、指先にオレンジ色の△（三角）が表示された場合はタップ（爪）、指先の左に左方向を向いた赤い矢印が表示された場合はスワイプ（左）、指先の右に右方向を向いた赤い矢印が表示された場合はスワイプ（右）とした。これらの指示の内容は予め被験者に伝えられ、実験開始までに記憶してもらう。

本システムでは、操作中に指領域同士が接触してしまったり、指領域がカメラの画角から外れてしまった場合に誤認識を起こしてしまう可能性がある。実際、例えば人差し指と中指の領域が1つにくっついてしまった場合、検出される領域数が1つ減少しているため、仮にどの指も設置面に接触していなくてもどちらかの指が設置面に接触していると誤認識される可能性がある。（ただし、音声情報を用いた入力認識も同時に行っているため、その誤認識が直ちに誤入力に繋がるわけではない。）

そこで、本実験では画面左下に取得した2値画像を表示し、被験者に指がどのように写っているかを提示することで、どのように指を動かすと認識されやすいか、もしくは誤認識されやすいかを被験者に伝えるようにした。また、2値画像上に各指の認識されている重心位置を小さな円で表示した。重心位置は、指が設置面に接触している時は青色、接触していない時は緑色で表示される。また、設置面領域の重心位置は赤い円で表示される。このような表記や注意についても、予め実験前に被験者に伝え、入力動作に出来るだけ早く慣れることが出来るようにした。

また、同様に画面右下に音の大きさを表示した。音の大きさは256サンプルの平均をその都

度表示した。また、音の大きさについては画面に設定されているしきい値を表示し、どれくらいの大きさの音が検出されると入力として認識されるかを提示した。

なお、被験者が入力を行うと、次の入力すべきジェスチャーが画面中央の手の画像に表示される。同時に、入力したジェスチャーの認識結果が簡易的に右上に表示される。この表示は、表示されている円が左から順に親指、人差し指、中指、薬指、小指に対応しており、灰色であればその指は何も入力に関係していないことを示す。また、それ以外の色（緑、オレンジ、赤、青）であった場合には、対応する色のジェスチャーがその指で入力されたことを示す。被験者には、この入力結果を見ることによって操作に対する学習を促し、なるべく早く機器の操作に慣れることが出来るようにした。

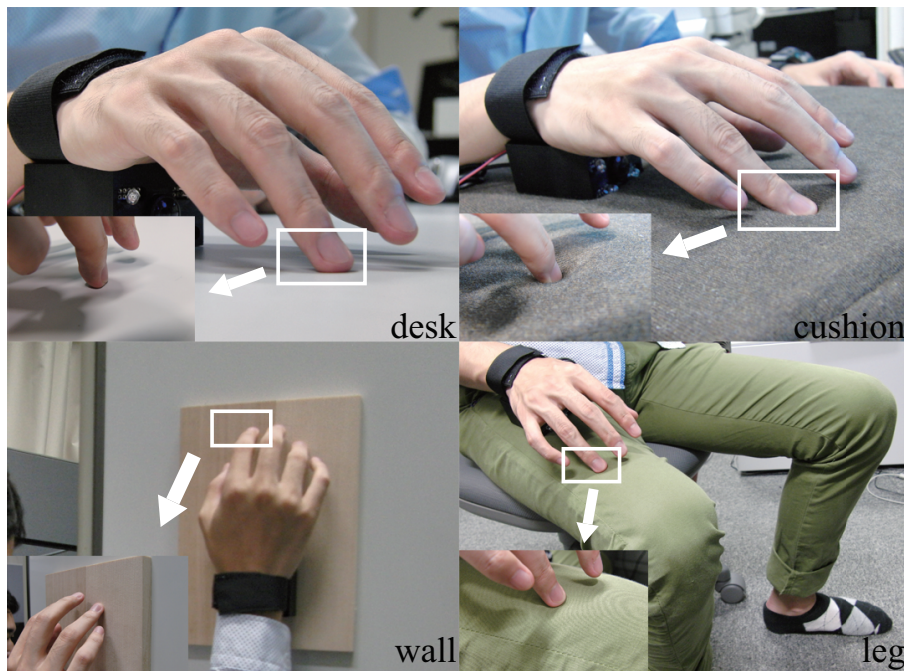


図 4.10. 各実験環境の様子

被験者は、最初に実験についての説明を受ける。次に、机の上で実際にジェスチャー入力を行いつつ、実際にシステムを操作しシステムの操作に慣れる。慣れるまでの時間はおよそ5分から10分程度とし、被験者が慣れたと申告した時点で練習を終了した。そして、どの被験者もまずは机の上で実験を行いデータを記録した。また、その後それぞれの環境において再度ジェスチャー入力の練習を行い、データを記録した。環境ごとに練習を行ったのは、環境が変わるごとに機器の操作感が微妙に異なるためである。なお、机の上で実験をした後の実験環境の順序は、ラテン方格法を用いてランダムに順序を入れ替えて実験した。

実験は、図 4.11 のように画面にランダムに提示されたジェスチャーに従って入力動作を繰り返し行いそれらを記録する。1回の試行で36通りの入力を1回ずつ行い、それを5回繰り返す。5回の試行が終わると、環境を変え同様に実験を行う。本実験では、実験データとして、入力された時の指の組合せ、ジェスチャー、指の腹/爪の情報を記録した。実験は一人あたり全

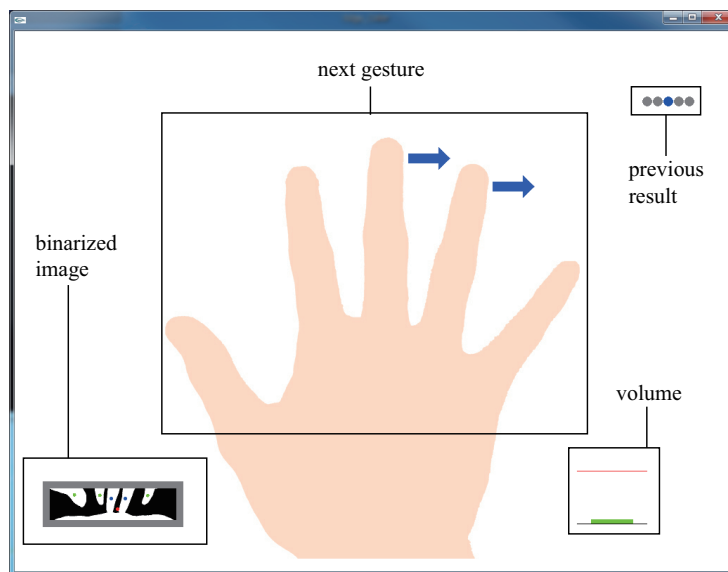


図 4.11. タスク画像

部でおよそ 45 分程度であった。

また、実験の最後に簡単なアンケートを行い、システムの主観的な評価を聞いた。

- 提案インタフェースの使い方はすぐに理解出来ましたか？ (Learnability)
- 提案インタフェースの使い方はすぐに覚えられましたか？ (Memorability)
- 提案インタフェースを思ったように操作することが出来ましたか？ (Task Attainment)
- 提案インタフェースが使いやすいインタフェースだと感じましたか？ (Usability)
- 提案インタフェースを操作していて楽しいと感じましたか？ (Satisfaction)

各項目は、-3 から 3 までの数字で 7 段階評価してもらった。(3 に近づくほど質問に対し同意している、-3 に近づくほど同意していないと考えた事に相当する。) また、実験について自由に感想を既述してもらおう欄も設けた。

4.4.2 被験者

被験者として、情報系の学生 12 名 (男性 10 名、女性 2 名) がボランティアとして参加した。被験者は全て機縁法により募集した。なお、11 名が右利きで右手でシステムを操作したが、1 名は左利きであったために左手で操作した。提案システムでは、右利きと左利きによって左右のスワイプの操作の意味が異なる。具体的には、図に示すように、例えば右利きの人が右手の親指で左にスワイプ動作を行うのと同等のジェスチャ入力、左利きの人左手の親指で右に向かってスワイプ動作である。(手の外側に向かって指や手全体を動かしているという意味で同等という表現をしている。) 今回の実験では、各平面における入力動作の特性などにも興味があるため、得られたデータをそのまま集計してしまつては正しい結果が得られない。そこで、左利きの被験者のデータについては、上記で説明した通り左右を反転させた形で

データの集計を行った。なお、実験以前に AST を操作したことのある被験者はいなかった。

4.4.3 結果

図 4.12 に各環境における各ジェスチャの認識率を示す。各棒グラフは被験者ごとの認識率の平均、エラーバーはそれらの標準偏差を表している。順に、机、クッション、脚、壁における認識率を示している。また、各環境下のジェスチャー認識率の平均は机：77.3%、クッション：85.8%、壁：74.7%、脚：75.8% だった。また、アンケートに自由に感想を記述してもらった結果、壁での入力は腕が疲れやすく操作しづらかったとの意見が多く見られた。

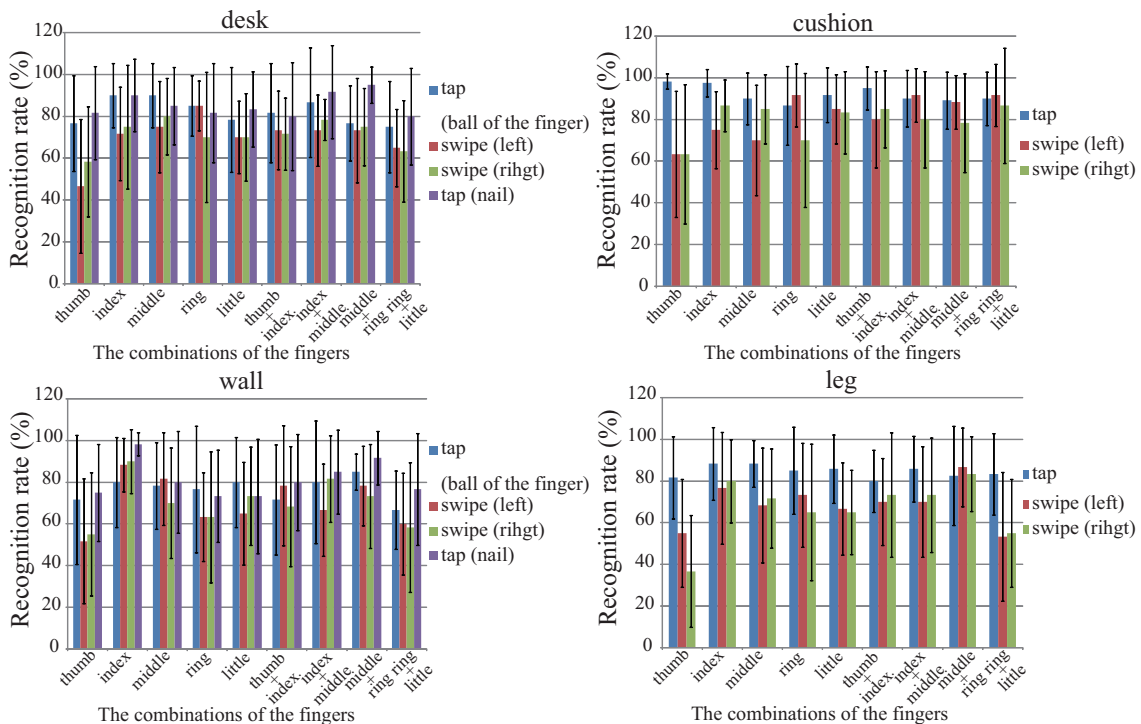


図 4.12. 各ジェスチャの認識率：机（左上）、クッション（右上）、壁（左下）、脚（右下）

また、図 4.13 にアンケートの結果を示す。表 4.1 には、アンケート結果の平均と分散を示す。アンケート結果のグラフを見てわかるように、Learnability や Memorability といったシステムの操作方法を理解し覚えることに関しては高い評価を得た。また、Satisfaction の値も比較的高く、被験者が操作に対して満足していたことが伺える。しかし一方で、Task Attainment や Usability の評価はそれらに比べると比較的低く、また分散も大きいことがわかる。

各環境における認識率と、アンケートの回答について被験者ごとに対応付け、相関係数を求めたものを表 4.2 に示す。表 4.2 を見ると、Task Attainment とクッション上での認識率に強い相関があることがわかる。また、他の環境下における認識率と Task Attainment との間など、いくつかの項目においてやや強い相関を見ることが出来る。

アンケートに自由に感想を記述してもらった結果、壁での入力は腕が疲れやすく操作しづら

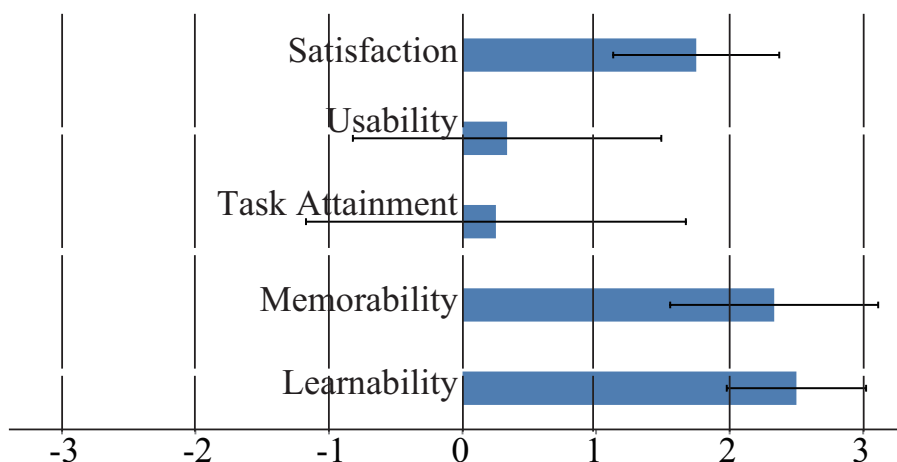


図 4.13. Anywhere Surface Touch を用いた実験でのアンケート結果

被験者数: 12 名 (男性: 10 名、女性: 2 名)

項目	平均	標準偏差
Learnability	2.5	0.52
Memorability	2.3	0.78
Task Attainment	0.25	1.4
Usability	0.33	1.2
Satisfaction	1.75	0.62

表 4.1. Anywhere Surface Touch の実験アンケート結果の詳細

	机	クッション	壁	脚	全環境の平均
Learnability	-0.064	-0.065	-0.44	0.076	-0.10
Memorability	0.33	0.019	-0.22	0.46	0.22
Task Attainment	0.45	0.77	0.34	0.46	0.62
Usability	0.22	0.43	0.13	0.27	0.33
Satisfaction	0.30	0.41	-0.23	0.46	0.35

表 4.2. 各アンケート項目と認識率の相関係数

かったとの意見が多く見られた。また、Learnability や Memorability はどの環境の認識率においてもほとんど相関が見られないという結果を得た。

4.4.4 考察

実験の結果、様々な環境でジェスチャ入力を行う事が可能であることを確認した。各環境ごとの認識率について、対応のある t 検定を行ったところ、以下のような結果を得た。

- クッション (M=84.01, SD=21.91) の方が机 (M=74.63, SD=23.49) より有意に認識率が高い ($t(322) = -5.87, p = 5.516 \times 10^{-9} < 0.01$)
- クッションの方が脚 (M=73.46, SD=26.60) より有意に認識率が高い ($t(322) = -6.26, p = 6.043 \times 10^{-10} < 0.01$)
- クッションの方が壁 (M=72.47, SD=26.52) より有意に認識率が高い ($t(322) = 6.43, p = 2.34 \times 10^{-13} < 0.01$)
- 机 (M=77.31, SD=10.57) のほうが壁 (M=74.72, SD=11.52) よりも有意に認識率が高い ($t(430) = -6.35, p = 0.03152 < 0.05$)

なお、爪によるタップが認識可能な面と不可能な面での検定では、爪によるタップのデータを削除した上で検定を行った。以上の結果から、クッションは他のどの環境よりも有意に認識率が高い環境であると言える。

本実験では、要求されたジェスチャに対して実際どのようなジェスチャが認識されたか、という事をすべてログに記録した。ログを確認した結果、本実験で起きた誤認識は大きく分けて以下の3つに分類できる事がわかった。

1. スワイプがタップと誤認識される
2. ジェスチャを行うべき指と隣り合った指も設置面に接触していると誤認識される
3. ジェスチャを行うべき指と隣り合っていないが、親指や小指が設置面に接触していると誤認識される

これらの誤認識は、すべて実験以前の段階で把握していた誤認識である。なお、これらの誤認識は予め被験者にも説明した上で練習・実験を行った。

1. は、被験者が丁寧に操作しようとしたあまり、指をゆっくり動かしてしまった事が原因である。実際、実験開始前の練習では、慣れない機器を初めて操作という事もあり、ゆっくり丁寧に操作しようとする被験者が多く存在した。しかし、本システムは指の移動速度を用いてスワイプの認識を行うため、ゆっくりとした動作ではシステムがスワイプと認識する事が困難になってしまい、タップに誤認識されてしまう。被験者には予めスワイプのやり方について説明を行ったが、実験を繰り返す中で、ミスをしないうにという気持ちからゆっくりと指を動かしてしまったと考えられる。

2. は、ジェスチャ入力時に誤って他の指も面に接触してしまった、もしくは面とは接触していないがジェスチャ入力を行っている指と接触してしまったために誤認識が発生したものである。特に後者の誤認識は、入力したい指と隣合う指との第二関節から指の根元までを接触させた状態でジェスチャ入力を行うと、このような誤認識が発生する。

3. は、操作に集中するあまり親指や小指が伸びてしまい、カメラの視野から外れてしまった事で発生する誤認識である。本システムのタッチ認識は、一時刻前のフレームと重心位置を比較し、近い位置に重心位置が見つからなかった場合にその指をタッチしていると認識する方法を採用しているが、指先が画面から消えてしまった場合も重心位置が検出されなくなるため、このような誤認識が発生する。

64 第4章 Anywhere Surface Touch：実環境の面を操作領域とする装着型手入力システム

これらの誤認識は、被験者が機器の操作に慣れるほど自然と減るミスと考えられる。しかし、より使いやすいインタフェースを実現するという意味では、今後タッチ認識やジェスチャ認識のアルゴリズムを改良する事で、ユーザーの意図した操作を認識可能なシステムの実現を目指す必要がある。

また、親指のスワイプはどの環境でも認識率が60%前後、もしくは60%以下と、平均に比べて低い認識率であった。これは、親指が他の指と構造が違う事に由来していると考えられる。具体的には、本システムではタッチしていない指の動きを見る事でスワイプ認識を行っているが、親指の場合他の指を動かさなくても左右に指を動かす事ができてしまうため、タップと認識されてしまうケースが多かったと考えられる。ただし被験者によっては問題なく入力できている事や親指+人差し指では認識率が低い事を考えると、親指によるスワイプも十分に練習する事で認識率が向上する事が期待できる。

環境別に見ると、脚の上での薬指+小指のスワイプの認識率の低さが目立つ。これは、脚の上では設置面が平坦ではなく、凸状に丸みを帯びている事が原因であると考えられる。複数の指で入力する場合それぞれの指を同時に面に設置させる必要があるが、脚の上などの凸面では薬指に比べて小指を設置面に接触させる事が難しく、認識率が低下したと考えられる。実際に実験時に記録されたジェスチャ入力の記録を確認したところ、誤認識の多くが小指のみによるジェスチャ、もしくは薬指のみによるジェスチャと誤認識されたものであった。この問題は、予め面の形状を大まかに計測するなどし、面の形状に応じて認識方法を適応的に変化させる事で認識率の向上が図る事が可能であると考えられる。

タップの指の腹と爪の認識について見ると、机の上ではどちらの認識率もほとんど差がないのに対して、壁で入力を行った際には爪の認識率のほうが高いことが見て取れる。これには大きく分けて2つ理由があると考えられる。

1つは、入力を行った面の材質の特性であると考えられる。机は金属製ではあるものの、正確には金属の板であり、また表面に塗装加工も施されている。それに対し、本実験で用いた壁の木材は、机の金属板より厚い木の板であった。また、木材も比較的軽量のものを用いたため、木材の中でも高い音が出やすい材質であった。それゆえ、指の腹を用いてタップを行った場合でも比較的高い接触音が出てしまい、爪でタップ動作をしたと誤認識されてしまった事が考えられる。この問題については、入力を開始する前にあらかじめ面の音響伝搬特性を調べ、指の腹と爪の認識を区別するためのしきい値をキャリブレーションするなどの方法を取ることで解決可能であると考えられる。

そして、もう1つの理由は、壁で入力している際の姿勢によるものと考えられる。被験者は提案システムをマジックテープによって手首に固定しているが、より正確に言うとは前腕部の手首に近いところから手のひらにかけてのあたりである。図4.14に示すように、提案システムを平面で用いる際には手のひらと前腕部がほぼ水平となるため、特に違和感なく操作することが出来る。しかし、壁で入力しようとする際には、手のひらと前腕部の間に一定の角度が生じる。

このため、提案システムを壁に設置させつつ入力動作を行おうとすると手首の固定している部分に必要以上に強い力がかかる、あるいは少し前腕部を手のひらと同じ方向に向ける必要が

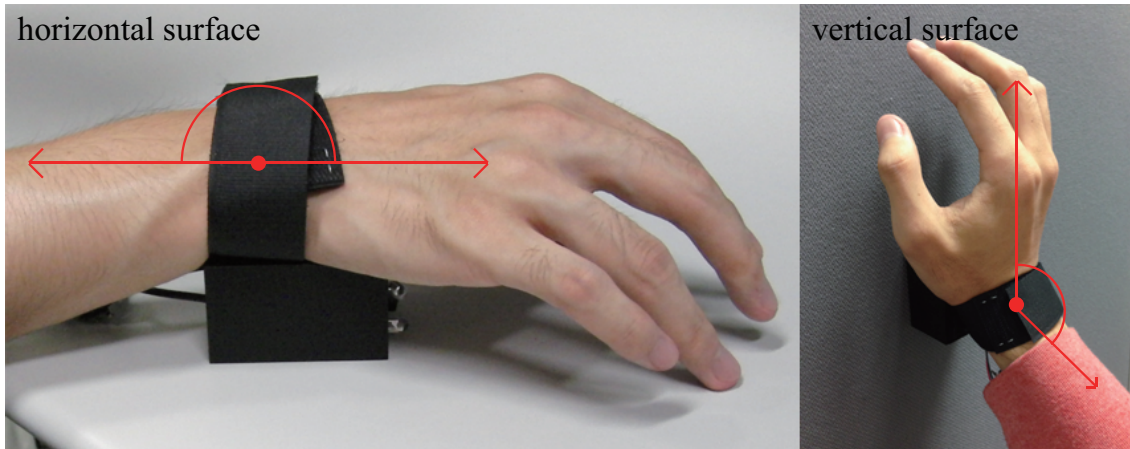


図 4.14. Anywhere Surface Touch を利用する際の姿勢：(左) 水平な面上で利用する場合 (右) 垂直な面で利用する場合

あるなど、やや難しい体勢になりながら入力を行わざるを得なくなる。このように壁での入力はやや不自然な体勢での入力を余儀なくされたため、タップ入力を行う際に不必要な緊張が筋肉に走り、強く壁をタップしてしまった、といった事が考えられる。なお、アンケートにおいても壁での入力が腕が疲れやすいという意見が多くあったが、その原因の1つもこの不自然な体勢での入力があると考えられる。(単純に腕を上げ続ける意味での疲労も当然あると考えられる。) これについては、システム本体の物理的な構成を再度検討する他、手首の曲げに対応して変形するような機構を取り入れるなどする事で壁においてもより自然な体勢での入力が可能になると考えられる。

ここからは、アンケートの結果について考察する。まず、Learnability や Memorability の評価が高かった、すなわち多くの被験者にとって操作方法が理解しやすく、また覚えやすかったことはポジティブな結果であると言える。特に、タップやスワイプといった動作はスマートフォンやタブレットなどのタッチパネルを用いたデバイスにおいて行い慣れている動作であるため、被験者にとっても馴染みの動作であったことが大きかったと考えられる。また、同様に Satisfaction も比較的高い評価を得た。被験者が操作していて楽しいと感じる事は重要であり、特に機器に慣れるまでの時間は操作していて楽しいと思えるほど熱中して機器を操作出来るため、機器に慣れるまでの時間も短縮出来ることが期待出来る。

一方、Task Attainment や Usability の評価は被験者によって割れた。その結果、平均値も Learnability や Memorability に比べて平均値が低く、また標準偏差の値が大きいという結果になっている。これは、特にスワイプ動作の習得にやや慣れが必要であったことが影響したと考えられる。実際、スワイプ動作にはある種のコツがあるようで、ある被験者はアンケートの自由記述欄で「タップして移動するように指を動かすと上手くいったような気がする」という感想を述べている一方で、別の被験者は最後までスワイプのやり方が上手くつかめなかったと述べている。当然、スワイプ動作の習熟には個人差があるため、ある程度このような結果になる事は致し方無いが、スワイプ動作の認識手法の改善を検討する必要があると考えている。ま

た、今回得られた意見を参考に、実験開始時の練習の際に与えるアドバイスをよりの確にすることによっても習熟を早めることが出来ると考えている。

それぞれのアンケートの項目と、各環境における認識率の相関係数についても考察する。Learnability や Memorability は、どの環境における認識率に対してもほとんど相関が見られなかった。しかし、これはどの被験者もこれらのアンケート項目に対して高い評価を与えたためであると考えられる。実際、これらの項目は標準偏差の値も小さく、評価が総じて高かったことがわかる。また、クッション上での認識率と Task Attainment の評価に強い相関が出たことは興味深い。各環境におけるジェスチャー入力の認識率を見ても、平均ではクッションでの入力が認識率が一番高かった。従って、比較的ジェスチャーに早く慣れることが出来た被験者は、クッション上でうまく入力出来たと考えた人が多く、その結果 Task Attainment の評価も高い評価を与えたと考えられる。

以上の結果から、今後より Anywhere Surface Touch の操作性を向上するためには、ジェスチャー入力の認識手法を改善し、より多くの人にとってすぐに入力が可能になるようなインタフェースを実現することが必要であると考えられる。

4.5 まとめ

本章では、我々は小型化する携帯機器がどのようにあるべきかを述べた。また、そのような携帯機器のコンセプトを実現するものとして、小型カメラとコンタクトマイクを一体化して手首に取り付ける Anywhere Surface Touch を提案した。AST により、様々な面上での複雑な操作を実現した。AST では画像を用いて指やジェスチャーの認識、音を用いた接触状態の認識を行いそれらの情報を統合する事で複雑な操作を実現している。

様々な環境で利用可能である事を確認するために実験を行った結果、机やクッション、壁、脚の上など身近にある様々な面上で入力可能である事を確認した。また、環境や姿勢などによっては実行しやすい、もしくは実行しにくいジェスチャー入力がある事を発見した。将来的に本システムを用いてアプリケーションを開発する場合には、人間の行い易い動作を考慮してアプリケーションを開発するといった事が必要になる。

将来的には、指の三次元形状を推定しつつ設置面との接触を判定する事で奥行方向へのジェスチャー入力の認識を実現する事を目標としている。また、音と画像の統合についても、よりロバストかつ高精度な認識を実現したいと考えている。

第 5 章

空中操作における仮想物体とのインタラクション

5.1 はじめに

近年、小型携帯機器に関する技術が進歩し、それに伴ってスマートフォンやタブレット PC などが広く世の中に普及するようになってきている。これらの機器の多くは、インタフェースとしてタッチパネルを採用している。タッチパネルを用いることによって、使用する文脈に応じた様々なメニューやボタンなどをユーザーに提示することが可能になった。

しかし、Siek ら [1] が指摘した *Fat finger problem* のように、現在の小型機器においては狭い機器表面で操作を行う必要があり、細かい操作を正確に行うことが難しいという問題が存在する。それに対し、我々は *Fat Finger Problem* を解決する一つの手法として、空中を操作環境とする空中操作インタフェースを提案した。だが、提案システムはボタンなどの表示が二次元平面上に限定されており、ボタンなどのオブジェクトを直接操作しているという実感に乏しいという課題が存在する。

この課題に対し、三次元空間上で物体を操作して入力を行うシステムの研究が提案されている。例えば、実物体を用いて操作を行うインタラクションシステムが提案されている [50, 51]。しかし、操作を行うために実物体を利用するには用途ごとのシステムの設計が必要であり、汎用性が低いという問題がある。

また、VR 技術を用いた三次元仮想物体とのインタラクションシステムも多数提案されている。例えば、指先に機器を装着し、特別な機器を用いて空中に結像された仮想物体とのインタラクションを実現するシステム [52] や、ステレオプロジェクターと専用の眼鏡、及び Kinect を用いて、自身の前面に用意されたスクリーンに投影されたオブジェクトを操作するシステム [53] などが提案されている。また、指にマーカーを取り付け、ホログラフィーを利用して空中に投影されたオブジェクトとインタラクションするシステムも提案されている [54]。しかしこれらの手法では、利用にあたって立体視用の眼鏡やセンサなどの特殊な装置を着用する必要があり、その煩わしさが問題となる。

このような煩わしさを解消するためには、カメラを用いて非接触で手を計測して操作を実現

することが望ましいと考えられる。Hilliges らが提案した HoloDesk[55] は、そのようなシステムの一つである。このシステムでは、デプスカメラを用いてユーザーの手の動きを認識し、仮想物体を動かす、持ち上げるといったインタラクションを可能にした。しかしながら、一般的なカメラは出力までの遅れ（レイテンシ）が大きいため、仮想物体と実物体の間に位置ずれが生じるという問題がある。

そこで、本研究では高速カメラを用いて手の動きを認識し、裸眼立体視ディスプレイを用いて立体表示したオブジェクトとインタラクション可能なシステムを提案する。高速カメラを用いることで、撮影後のレイテンシや撮影までの平均レイテンシが小さくなることが期待出来る。また、カメラで取得した手の動きと連動させてオブジェクトを移動させることで、実際の物体に触っているような感覚が得られるインタフェースシステムを提案する。

5.2 裸眼立体視ディスプレイを用いた仮想物体とのインタラクション

5.2.1 システム設計

本システムの全体像を図 5.1 に示す。今回は簡単のため、ユーザーから見てディスプレイの右側に黒い背景を設置し、左側に設置した高速カメラから安定して手指領域の認識が行えるようシステムを構成した。

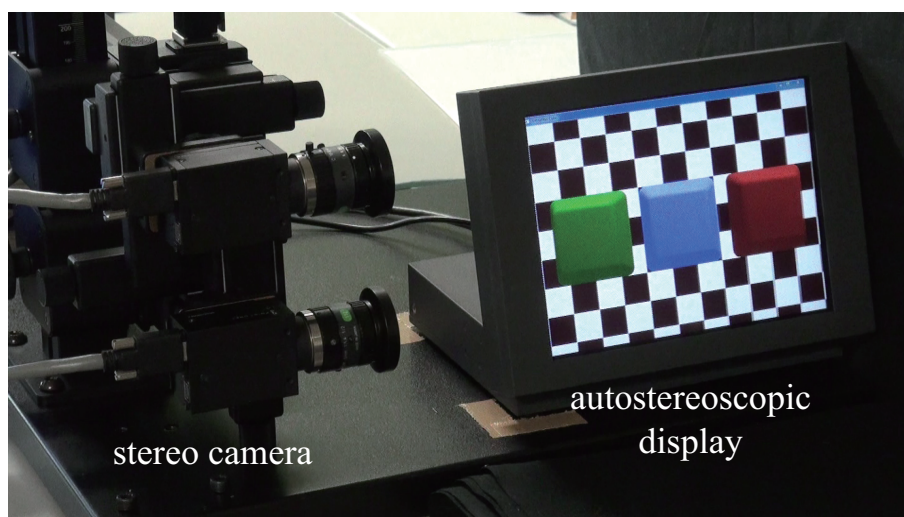


図 5.1. 裸眼立体視ディスプレイを用いたインタラクションシステム

今回のシステムでは、カメラに Point Grey Research 社の Grasshopper (GRAS-03K2M) を使用した。取得画像サイズは 640×480 [pixel] でモノクロ画像、フレームレートは 200fps である。一般に、高速カメラは画像出力までの遅延が一般的なカメラに比べて小さいため、仮想物体と実物体の間に位置ずれを生じにくい。すなわち、実物体と仮想物体との同期を実現し、実体感を高めることが出来ると考えられる。

項目	仕様
サイズ (インチ)	8.4
バリア方式	parallax barrier
レゾリューション	1024 × 768
表示タイプ	XGA
レスポンスタイム (ms)	25
ディメンション (W/H/D、mm)	223×190×148
視点数	5
3D 最適表示距離 (m)	0.9
最小最大 3D 表示距離 (m)	0.65-1.50

表 5.1. 提案システムで利用した裸眼立体視ディスプレイの仕様

また、裸眼立体視を実現する手法は多数あるが、本システム用に裸眼立体視ディスプレイを複数選定し、それらの立体視の安定性や解像度、最適視聴距離などを考慮した結果、今回は VMJ 社のパララックスバリア方式のディスプレイ (8.4" PB 3D Display) を採用した。ここで、パララックスバリア方式の裸眼立体視ディスプレイの原理について簡単に説明する。

パララックスバリア方式では、通常のディスプレイの手前側にパララックスバリアと呼ばれる遮光フィルタを配置する。一方、ディスプレイでは右目用画像と左目用画像を交互に表示する。このバリアにより、ディスプレイに表示された映像は左目用の映像と右目用の映像に分離されるため、立体映像として提示される。

ディスプレイの詳細な仕様を表 5.1 に示す。最小 3D 表示距離が 0.65m であるため、ディスプレイに提示される仮想物体に対して、立体視をしながら仮想物体とのインタラクションを行うことが出来る。なお、本ディスプレイを用いた三次元映像の描画は VMJ 社の提供している SDK を利用して行い、OpenGL で描画したい映像を作製するとディスプレイに最適化された立体視用映像が生成、表示される。

5.2.2 指先トラッキング

今回は簡単のため、ユーザーと仮想物体とのインタラクションは、指を一本のみ伸ばした状態で行われ、指がディスプレイ面に向いた状態で行われるものとした。本システムでは、ステレオカメラを用いて取得した 2 枚の画像からそれぞれ指先の位置を検出し、平行ステレオ法に基づき指先の三次元位置を推定する。以下に詳細を述べる。

まず、あらかじめステレオカメラに対してステレオキャリブレーションを行い、カメラ間の並進と回転行列を求めておく。なお、キャリブレーションには Camera Calibration Toolbox for Matlab[72] を利用した。

図 5.2 に処理の流れを示す。提案システムでは、まず左右それぞれのカメラで背景差分用の画像を一枚撮影する。これにより、ディスプレイの側面と黒い背景のみが写った画像が取得さ

れる。

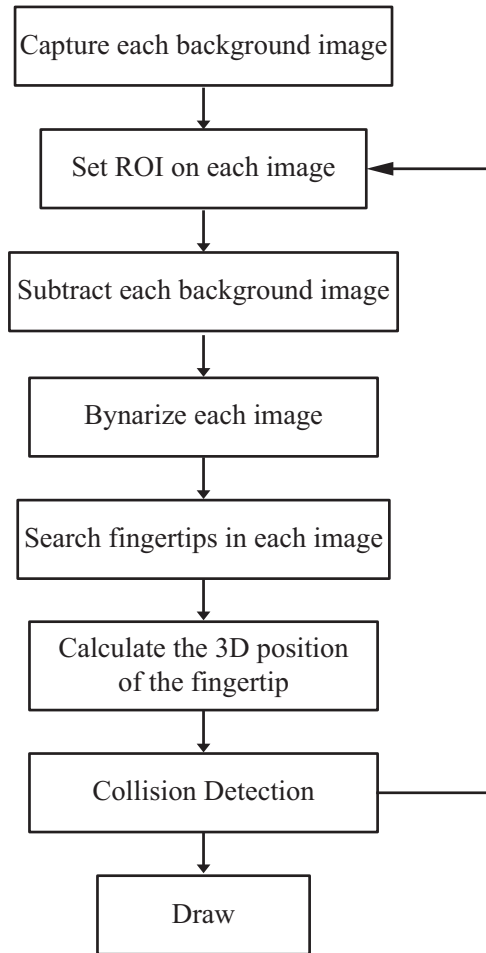


図 5.2. 画像処理の流れ

次に、取得した画像から指先領域を抽出し、指先のトラッキングを行う。トラッキング手法の詳細を以下に述べる。まず、取得した画像に対し指先のある位置を中心とした ROI (Region of Interest) を設定し、その領域のみを抽出する。また、対応する ROI を背景画像にも設定し、同様にその領域のみ抽出する。なお指先のある位置が検出されていない場合は、ROI の初期値として、得られた画像上でディスプレイと反対の側（指先が現れる側）にある画像の一部を設定する。

次に、取得した画像と背景画像の差分を取り、適切な閾値で二値化することによって、指先部分のみ白いピクセルで表示された画像を得る。得られた二値画像に対し、以下の条件に基づき探索を行うことで指先位置を検出する。

1. 指先の先端がある側の画像の列について上から順に画素値を見る。
2. もし白の画素値が一定 pixel 以上連続した場合、その座標を指先位置とする
3. もし連続する pixel が探索中の画像列に無かった場合、次の画像列に移り同様に上から探索する。

図 5.3 に指先位置検出アルゴリズムの探索の概要を示す。

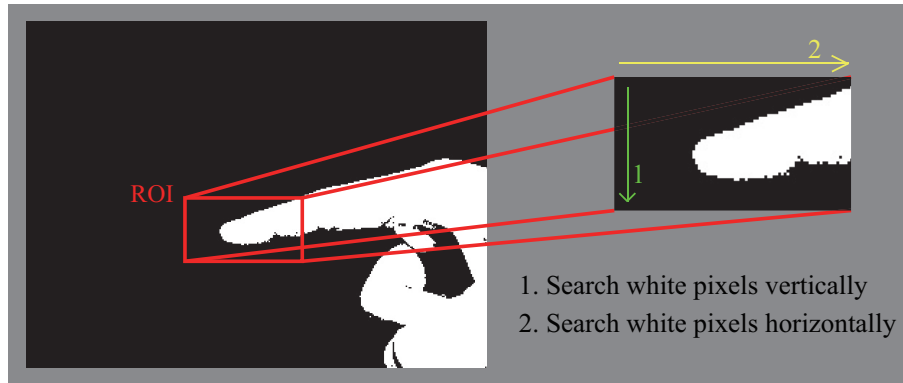


図 5.3. 指先位置検出の探索方法

以上の方法で、左右のカメラを用いて指先位置を検出する。最後に、それぞれのカメラで検出した指先位置の座標を元に、指先の三次元位置を推定する。

仮想物体との接触判定は、認識された指先の座標を用いて行う。具体的には、検出された指先座標の x 座標が画像座標中である値よりも小さくなった時に仮想物体との接触が発生したと判定する。また、 z 座標（カメラから見て奥行方向）の位置に合わせて、接触判定を行う仮想物体を適宜変更する。接触後は、指先の位置に合わせて仮想物体の位置を移動させることで、指が仮想物体に触れているかのような視覚効果を実現する。

5.2.3 裸眼立体視ディスプレイを用いたインタラクションシステム

設計したシステムをもとに、本研究では三次元グラフィカルユーザーインターフェースを実現するものとして、ボタンを押しこむことが可能なデモシステムを開発した。図 5.4 にデモシステムを操作している様子を示す。

ディスプレイにはボタンが立体的に表示され、ユーザーにはそれぞれがディスプレイから飛び出して見えている。ユーザーが中央のボタンに対して指を近づけていくと、空中のある位置でボタンとの接触判定が起き、指を押しこむことでボタンも連動して奥に向かって押し込まれる。指はディスプレイ面に接触するまで押しこむことが可能であり、押し込んだ指を手前に引くと押し込んだ時と同様に指先と連動してボタンが手前に移動する。

本システムを ITS 2012 のデモ発表等 [73, 74] で対外的に発表し、多くの人に操作してもらった。その結果、視覚的にボタンの表面に接触したと感じた位置でディスプレイ面にも接触したと錯覚するが、そのまま指を押しこんでいく事で自身の指でボタンを押し込んでいるという実感を得ることが出来るという感想を得た。また、ディスプレイに接触するまで押しこむことで指先に触覚を得ることができ、ボタンを押し込み切ったという実感を得ることも出来るという感想もあった。高速カメラを用いたことにより、通常のカメラを利用した時と比較して、指先がボタンに接触した際のボタンの応答がより同期していると感じることが出来るといった意見もあった。

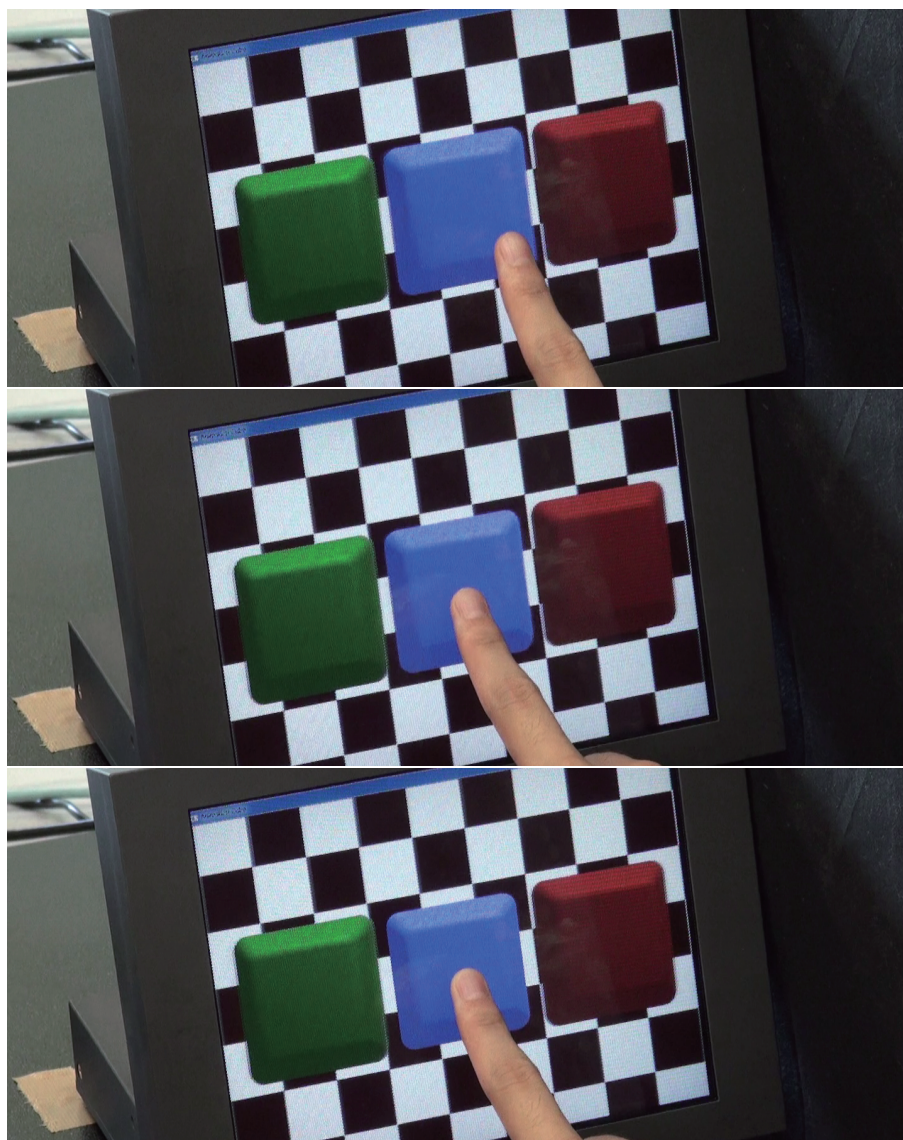


図 5.4. 操作の様子：指がディスプレイに近づくと連れてボタンが押し込まれていく

しかし一方で、ボタンを押し込んだ後に指を手前に引くといった切り返しの動作など、指先の速度が急激に変化するような動作を行った場合、指先の動きに対してボタンが追従していないと感じられるといった問題が指摘された。そこで、次節からは高速ビジョンを用いた指先位置予測を行い、仮想物体とのインタラクションにおける指先位置と仮想物体の位置との同期精度向上を目的とした実験を行う。

5.3 仮想物体とのインタラクションにおける同期精度向上

カメラを用いて仮想物体とのインタラクションを認識する場合、一般的なカメラでは出力までのレイテンシが大きいと、実物体と仮想物体との位置ずれが生じるという課題が存在する。

そこで、本研究では高速カメラを用い、人間の動作の予測を行うことでこの問題の解決を目指す。高速カメラを用いることで、この問題に対しては次のようなメリットがあると考えられる。

- 人間が動作を行ってから画面に提示するまでのレイテンシの減少
- 人間の動作予測の精度の向上

以下、それぞれについて簡単に説明する。

5.3.1 高速ビジョンを用いることによるレイテンシへの影響

高速ビジョンを用いることによるレイテンシへの影響について説明する。仮想物体とのインタラクションにおいては、システムのレイテンシが大きいほど実物体と仮想物体との位置ずれが発生しやすい。このレイテンシを更に細かく分解すると、次の4つのレイテンシに分割することが出来る。

1. 指先をカメラで撮影する際のレイテンシ
2. 撮影した画像を PC に転送する際のレイテンシ
3. PC で指先位置を計算する際のレイテンシ
4. 指先位置の動作を仮想物体の位置に反映させ、ディスプレイに表示する際のレイテンシ

1つめのレイテンシは、カメラのフレームレートの逆数（1フレームを撮像するにあたり何ミリ秒かかるか）に左右される。また、2つめのレイテンシはカメラの性能に依存する。ただし、リアルタイムに画像を転送可能なカメラであれば、最低でも1つめのレイテンシと同等、もしくはそれ以下であることが保障される。3つめのレイテンシは処理の内容に依存するため、一般的な評価は難しい。しかし、全てのフレームに対して毎回同じ処理を行っているのであれば、このレイテンシも1つめのレイテンシと同等かそれ以下であると考えられる。最後のレイテンシは、ディスプレイのフレームレートに依存する。なお、このレイテンシは他のレイテンシと比較してばらつきやすい。例えば、もし描画命令が出た直後にディスプレイの更新タイミングが来た場合、レイテンシは殆ど存在せずに画面に反映される。しかし、もし描画命令が出た直前にディスプレイで描画が発生した場合はディスプレイの更新周期とほぼ同じ時間だけのレイテンシが発生する。

以上のように、仮想物体とのインタラクションにおいては大きく分けて4つのレイテンシが存在する。1つめや2つめのレイテンシはカメラのフレームレートと密接な関係があることがわかる。従って、高速カメラを用いることでレイテンシを削減し、仮想物体とのインタラクションにおける同期精度が向上出来ると考えられる。

5.3.2 高速ビジョンを用いることによる動作予測精度への影響

高速カメラを用いることでシステム全体のレイテンシを削減し、仮想物体とのインタラクションにおける同期精度が向上可能であることを述べた。しかし、レイテンシ削減の効果以外にも、人間の動作予測を行った際に動作の予測精度が向上するというメリットが考えられる。

ここで、人間の手の動きは局所的には何らかの単一の運動モデルに従っていると仮定する。仮に、ある動作において人間の指先が図 5.5 のように、時刻 t_0 まである速度 v_a で等速直線運動をし、その後速度 v_b で等速直線運動をしたとする。

予測フィルタによる予測が運動モデルの転換点、すなわち時刻 t_0 の前後においては、フレームレートカメラの高低によらず、同等に精度よく動作の予測が出来ると考えられる。

しかし、運動モデルの転換点 t_0 付近について考えると、その予測精度に差が生じると考えられる。なぜなら、高フレームレートカメラは速度が変化した後の情報を即座に取得し、モデルの修正が可能であるのに対し、低フレームレートカメラの場合はその取得が遅れ、モデルの修正が遅れる可能性が高いと考えられるからである。

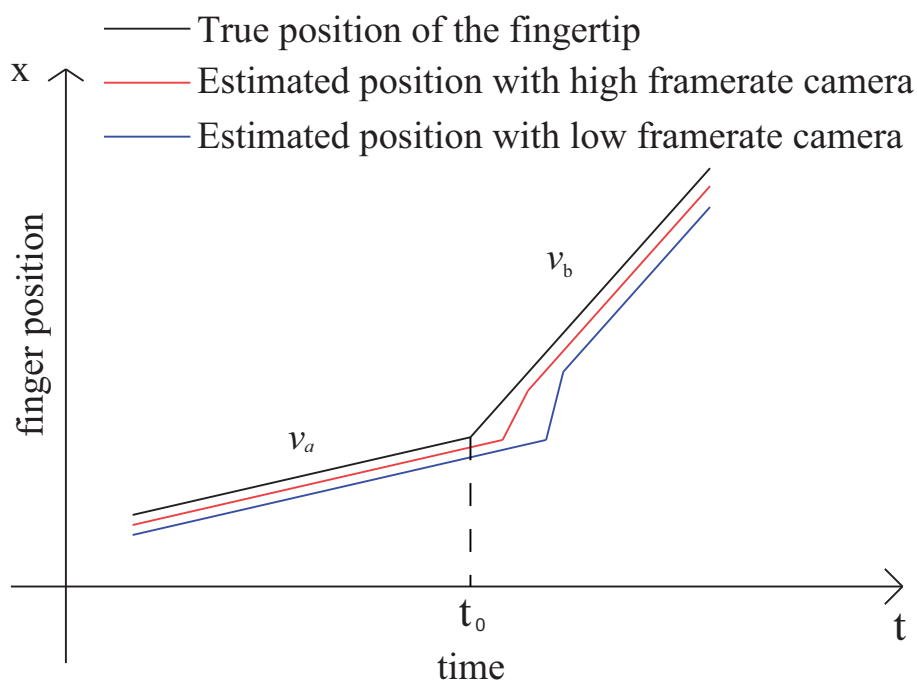


図 5.5. フレームレートが動作予測に与える影響

5.3.3 実験システム

本実験では、通常のディスプレイおよびその手前に設置したカメラからなるシステムで実験を行う。本研究は、裸眼立体視ディスプレイを用いた仮想物体とのインタラクションにおける同期精度向上を目的としたものではあるが、裸眼立体視ディスプレイを用いたシステムでは手

前に飛び出した仮想物体をカメラで捉えることが出来ないため、客観的な同期精度の評価が難しい。そこで、今回は通常のディスプレイを用いて同期精度の評価を行う。

図 5.6 に構成したシステムの様子を示す。今回のシステムで用いるディスプレイは、システムのレイテンシのばらつきを極力低減させるためなるべく高いフレームレートのディスプレイが望ましい。そこで、今回は W2363D-PF (23", 120Hz, LG 電子製) というディスプレイを使用した。通常のディスプレイは 60Hz 程度であることが多く、5.3.1 で述べた 4 つめのレイテンシをおよそ半分に減少することが可能である。

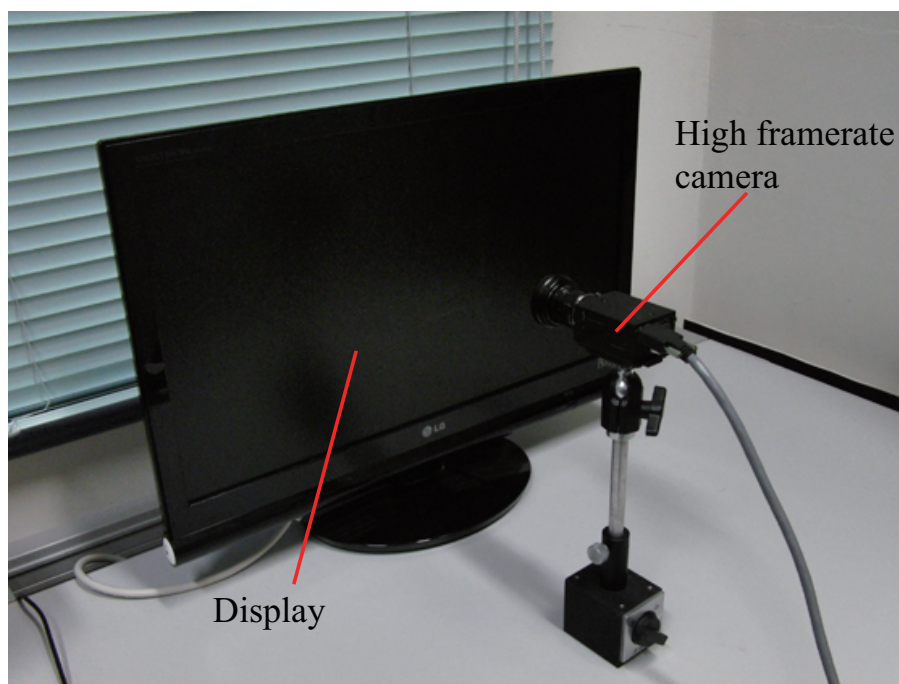


図 5.6. 指先位置予測システムの構成

また、今回のシステムで用いるカメラとして、Point Grey Reseach 社の Grasshopper (GRAS-03K2C-C) を使用した。このカメラは取得画像サイズが最大で 640×480 [pixel]、またフレームレートが最大で 200[fps] まで設定することが可能である。今回は取得画像サイズを 640×480 [pixel] とした。また、フレームレートは実験に応じて 200[fps] と 30[fps] を切り替えて構成した。また、カメラはディスプレイからおおよそ 200[mm] 程度離れた位置に設置した。

5.3.4 ディスプレイ座標とカメラ座標の対応付け

今回のシステムでは、カメラ画像から得られた手指の位置情報を用いて、ディスプレイ上に描画された仮想物体とのインタラクションを行う。その際、ディスプレイ上の仮想物体がカメラ画像中ではどこに位置するのかを調べるため、ディスプレイ座標とカメラ座標の対応付けを行う必要がある。

今回のシステムでは、ホモグラフィ行列による射影変換によって座標間の対応付けを行う。ホモグラフィ行列とは、三次元空間上のある平面を、別の二次元平面に変換する行列である。

変換前の座標を (x, y) 、変換後の座標を (x', y') とし、ホモグラフィ行列を H とすると、座標変換は次式のように書ける。

$$s \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (5.1)$$

$$sx' = \mathbf{H}x \quad (5.2)$$

ここで、 s はスケール係数である。この式より、次の3本の方程式が導出される。

$$\begin{cases} h_{11}x + h_{12}y + h_{13} = sx' \\ h_{21}x + h_{22}y + h_{23} = sy' \\ h_{31}x + h_{32}y + 1 = s \end{cases} \quad (5.3)$$

s を消去し、次の独立な方程式を2本得る。これを行列の形式で表すと次の式になる。

$$\begin{cases} h_{11}x + h_{12}y + h_{13} - h_{31}xx' - h_{32}yx' = x' \\ h_{21}x + h_{22}y + h_{23} - h_{31}xy' - h_{32}yy' = y' \end{cases} \quad (5.4)$$

$$\begin{pmatrix} x & y & 1 & 0 & 0 & 0 & -xx' & -yx' \\ 0 & 0 & 0 & x & y & 1 & -xy' & -yy' \end{pmatrix} \begin{pmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{pmatrix} = \begin{pmatrix} x' \\ y' \end{pmatrix} \quad (5.5)$$

この連立方程式を解くことにより、ホモグラフィ行列の8つのパラメータ h_{11} から h_{32} が求められる。1組の対応点から連立方程式が2式得られるため、ホモグラフィ行列を求めるためには最低でも計4組の対応点を調べる必要がある。

本システムでは、指先トラッキングを開始する前に、図5.7のようにディスプレイ上に緑色の円形のマーカを描画し、カメラで取得する。取得した画像をRGB画像に分割し、G画像に対して特定の閾値で2値画像に変換する。2値画像に対しラベリング処理を行い、4点の画像上での中心を求める。あらかじめディスプレイ座標においてどこに円形のマーカを描画したかは既知であるため、この4点の対応付けを行うことで、ホモグラフィ行列を求めることが出来る。

5.3.5 指先位置認識

指先位置認識は、5.2.2で説明したアルゴリズムとほぼ同等の手法を用いる。ただし、今回はディスプレイが背景となっているため、背景画像の取得や背景差分の利用は行わず、ディスプレイに表示されるウィンドウの背景を黒くすることで処理を簡単にする。また、今回のシス

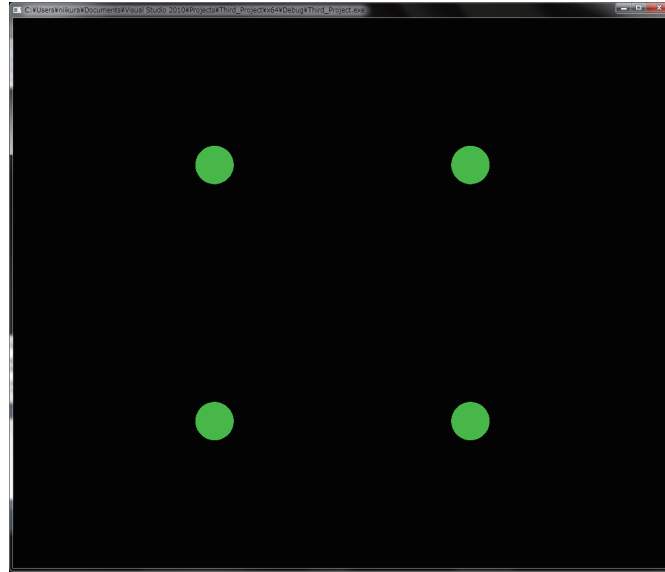


図 5.7. ディスプレイ座標とカメラ座標の対応付け

テムは単眼のカメラを用いているため、指先の3次元位置を求める処理も行っていない。その代わりに、指先は平面と同じ面上に存在すると仮定し、実験を行う際にはディスプレイの面上で操作を行うこととした。

5.3.6 カルマンフィルタを用いた指先位置予測

人間の動作を予測するために、予測フィルタとしてVRやARの分野を中心にカルマンフィルタが広く利用されている [75, 76, 77, 78]。そこで本実験でも、指先位置の予測に対してカルマンフィルタを用いた予測を行う。ここではまず、カルマンフィルタの概要を説明する。

画像を通じて観測した目標物体の位置を、観測ベクトル z_t で表す。画像のノイズ等によって、観測データにはノイズが付加されていると考えられる。従って、観測データから真の位置を直接知ることが出来ない。また、位置以外の情報、例えば指先の速度や加速度も直接知ることが出来ない。そこで、そのような要素を内部状態と呼び、状態ベクトル x_t の要素とする。

x_t に関して、次の観測値が得られるまでの状態遷移を、何らかの線形遷移で仮定し、その遷移行列を F_t とする。これはすなわち線形な運動モデルを導入することである。この際、制御ベクトルとして外力の作用をモデルに組み込むことも可能であるが、本実験で扱う指先の動きに関しては現れないため、ここでは割愛する。

また、通常モデルは完璧ではないため、遷移ノイズ w_t を加算する。なお、このノイズはガウス分布に従うと仮定する。この仮定によって、 w_t の平均と分散さえ既知であれば、状態の事前分布から予想される状態の分布を得ることが出来る。以上の説明をまとめると、時刻 k における状態方程式は以下のように表せる。

$$\mathbf{x}_t = F_t \mathbf{x}_{t-1} + \mathbf{w}_t \quad (5.6)$$

カルマンフィルタは、観測値と状態値には線形な関係があるシステムを扱っている。従って、状態ベクトルに観測行列と呼ばれる行列 G がかった状態で観測値を得る。観測値にも外力が関係する場合はこの影響を考慮する必要があるが、これについても本実験では扱わないため以後省略する。先にも述べたように、観測にはノイズがのるが、この観測ノイズも正規分布と仮定し、これを n とすると、観測方程式は次のような方程式となる。

$$z_t = H_t x_t + n_t \quad (5.7)$$

以上の式に従って、カルマンフィルタは各時刻での予測・および状態の更新を行う。以下では、 $\hat{x}_{l|m}$ は、時刻 m における時刻 l の状態の推定値を表すものとする。 t を現在の時刻とすると、カルマンフィルタはまず時刻 $t-1$ の状態推定値から時刻 t での状態を予測する。

$$\hat{x}_{t|t-1} = F_t \hat{x}_{t-1|t-1} \quad (5.8)$$

その予測の正確さを、時刻 $t-1$ での誤差の共分散行列の推定値から、以下の式を用いて誤差の共分散行列の予測値として求める。

$$P_{t|t-1} = F_t P_{t-1|t-1} F_t^T + Q_t \quad (5.9)$$

最後に、予測値と現在時刻の観測値を用いて、状態の修正を行う。

$$\tilde{y} = z_t - H_t \hat{x}_{t|t-1} : \text{観測残差} \quad (5.10)$$

$$S_t = H_t P_{t|t-1} H_t^T + R_t : \text{残差の共分散} \quad (5.11)$$

$$K_t = P_{t|t-1} H_t^T S_t^{-1} : \text{最適カルマンゲイン} \quad (5.12)$$

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t \tilde{y} : \text{更新された状態の推定値} \quad (5.13)$$

$$P_{t|t} = (I - K_t H_t) P_{t|t-1} : \text{更新された誤差の共分散行列} \quad (5.14)$$

このようにして $\hat{x}_{t|t}$ 、 $P_{t|t}$ を得る。次の時刻 $t+1$ の観測値が得られると、これらの推定値を元に同様に推定を行う。

5.3.7 指の動きのモデル化

カルマンフィルタを指の動きに適用するためには、指の動きの運動モデルを考える必要がある。今回のシステムで特に注目しているのは、仮想物体を押し込み、戻すといった動作など急激な速度の変化を伴った動作である。そこで、本研究では指は等加速度運動によって仮想物体とのインタラクションを行うと仮定し、運動モデルとして等加速度運動を採用する。また、簡単のためここでは指の動きは1次元的な動きであると仮定する。

カメラ画像の座標における時刻 t での指の真の位置を $x = p_t$ 、速度を $v = v_t$ 、加速度を $a = a_t$ とする。等加速度運動をする指先の時刻 $t + \Delta t$ における位置 $p_{t+\Delta t}$ は

$$p_{t+\Delta t} = p_t + v_t \Delta t + \frac{(\Delta t)^2}{2} a_t \quad (5.15)$$

と書くことが出来る。ここで、状態ベクトルを

$$\mathbf{x} = \begin{pmatrix} x \\ v \\ a \end{pmatrix} \quad (5.16)$$

とおくと、このモデルの状態方程式は以下のように表すことができる。

$$\mathbf{x}_t = F\mathbf{x}_{t-1} + \mathbf{w}_t \quad (5.17)$$

ただし、 \mathbf{w}_t は $\mathbf{w} \sim N(0, Q)$ からなる遷移雑音である。また遷移行列 F は、

$$F = \begin{pmatrix} 1 & 1 & \frac{1}{2} \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \quad (5.18)$$

で表される。また、観測値を $z = z_t$ とすると、観測方程式は以下の式になる。

$$z_t = H\mathbf{x}_t + \mathbf{n}_t \quad (5.19)$$

ただし、 \mathbf{n}_t は $\mathbf{n} \sim N(0, R)$ からなる観測雑音である。また、観測行列 H は

$$H = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \quad (5.20)$$

で表される。

5.3.8 レイテンシの計測

本研究では、高速カメラを利用することで指先の位置予測精度を向上させ、指先位置と仮想物体との位置ずれを減少させることを目的としている。しかし、そのためにはあらかじめ補償すべき遅延（レイテンシ）を把握しておく必要がある。具体的には、予測すべき時間は遅延フレーム分なので、今回は 200fps と 30fps で撮影した場合のそれぞれのレイテンシを計測した。

レイテンシの計測に用いた簡単なアプリケーションについて説明する。カメラは、先に説明した方法で指先位置を検出する。また、指先位置に合わせて、円形のマーカーが画面に表示される。マーカーは、指先の左右の動きを追従するように移動する。なお、補償すべきレイテンシの計測が目的であるため、この予備実験においてはカルマンフィルタによる予測は行わない。なお、5.3.4 節で述べたディスプレイ座標とカメラ座標の対応付けは事前に行う。

次に、構築したシステムとは別に、高速カメラ（Casio EXILIM EX-F1）を用意する。これを用いて、指先および画面に表示されたマーカーの両方を 300[fps] で撮影する。指先を繰り返し運動させ、指先に遅れて切り返すマーカーの様子を外部の高速カメラで撮影することで、各フレームレートにおけるレイテンシの計測を行った。

最後に外部の高速カメラで撮影した画像から、切り返しの際に遅れたフレーム数を計測する。この作業を複数回繰り返し、そのフレーム数の平均をとることで、描画タイミングのばらつきに起因するレイテンシのばらつきを解消する。今回は切り返しの際に遅れたフレーム数を図る作業を 11 回繰り返し、平均を取った。レイテンシ計測の様子を図 5.8、計測したレイテンシの平均を表 5.2 に示す。

この結果を元に、30fps での予測フレーム数は 2、200fps での予測フレーム数は 5 とした。

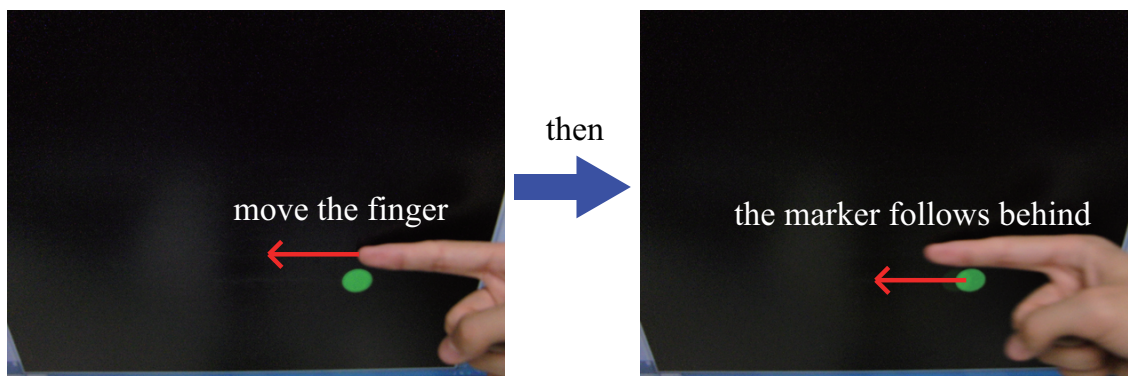


図 5.8. レイテンシ計測の様子

	30fps	200fps
遅れたフレーム数の平均 (300fps)	19.7	7.90
時間 (ms)	65.8	26.4
フレーム数 (30fps 換算)	1.97	(0.79)
フレーム数 (200fps 換算)	(13.2)	5.27

表 5.2. フレームレート毎のレイテンシ計測結果

5.3.9 カルマンフィルタのノイズパラメータ設定

カルマンフィルタのパラメータについて述べる。ここで、プロセスノイズ、観測ノイズは時間に依存しないため、フィルタ計算の前にはあらかじめ把握しておく必要がある。本節では 30fps、200fps のそれぞれに対して、それぞれ 27 種類のプロセスノイズを設定し、予測精度がどのように変化するかを比較することによって、より指先の予測精度が向上するよう調節する。

プロセスノイズの共分散行列はパラメータ別に $Q1 \dots 27$ とし、それぞれ以下のように設定する。

$$\begin{aligned}
 Q1 &= \text{diag}(10^{-1}, 10^{-1}, 10^{-1}), Q2 = \text{diag}(10^{-1}, 10^{-1}, 10^{-2}), Q3 = \text{diag}(10^{-1}, 10^{-1}, 10^{-3}) \\
 Q4 &= \text{diag}(10^{-1}, 10^{-2}, 10^{-1}), Q5 = \text{diag}(10^{-1}, 10^{-2}, 10^{-2}), Q6 = \text{diag}(10^{-1}, 10^{-2}, 10^{-3}) \\
 Q7 &= \text{diag}(10^{-1}, 10^{-3}, 10^{-1}), Q8 = \text{diag}(10^{-1}, 10^{-3}, 10^{-2}), Q9 = \text{diag}(10^{-1}, 10^{-3}, 10^{-3}) \\
 Q10 &= \text{diag}(10^{-2}, 10^{-1}, 10^{-1}), Q11 = \text{diag}(10^{-2}, 10^{-1}, 10^{-2}), Q12 = \text{diag}(10^{-2}, 10^{-1}, 10^{-3}) \\
 Q13 &= \text{diag}(10^{-2}, 10^{-2}, 10^{-1}), Q14 = \text{diag}(10^{-2}, 10^{-2}, 10^{-2}), Q15 = \text{diag}(10^{-2}, 10^{-2}, 10^{-3}) \\
 Q16 &= \text{diag}(10^{-2}, 10^{-3}, 10^{-1}), Q17 = \text{diag}(10^{-2}, 10^{-3}, 10^{-2}), Q18 = \text{diag}(10^{-2}, 10^{-3}, 10^{-3}) \\
 Q19 &= \text{diag}(10^{-3}, 10^{-1}, 10^{-1}), Q20 = \text{diag}(10^{-3}, 10^{-1}, 10^{-2}), Q21 = \text{diag}(10^{-3}, 10^{-1}, 10^{-3}) \\
 Q22 &= \text{diag}(10^{-3}, 10^{-2}, 10^{-1}), Q23 = \text{diag}(10^{-3}, 10^{-2}, 10^{-2}), Q24 = \text{diag}(10^{-3}, 10^{-2}, 10^{-3}) \\
 Q25 &= \text{diag}(10^{-3}, 10^{-3}, 10^{-1}), Q26 = \text{diag}(10^{-3}, 10^{-3}, 10^{-2}), Q27 = \text{diag}(10^{-3}, 10^{-3}, 10^{-3})
 \end{aligned}
 \tag{5.21}$$

ここで、 $diag(v)$ は v を対角成分に持つ対角行列を表す。また観測ノイズの共分散行列 R はどちらのフレームレートにおいても等しく $R = diag(1)$ とした。

今回、ユーザーはカメラに対して左右方向に指先を移動させ、その動作を 200fps に設定したカメラで 1 秒間トラッキングした。トラッキングした指先位置のうち、 x 軸方向の指先位置をデータとして使用した。また、全く同じ動作を人間が行うことは難しいため、30fps のデータについては、200fps で撮影したデータを適切な時間間隔で間引くことによって仮想的に実現した。

まず最初に、図 5.9 から 5.13 に 200fps、30fps のそれぞれのフレームレートにおける各パラメータを用いた指先位置予測結果を示す。なお、200fps においては 5 フレーム先、30fps においては 2 フレーム先の指先位置を予測しているため、これらのグラフでは予測された指先位置を予測フレーム分だけそれぞれ後にずらして描画している。

また、表 5.3 に各パラメータにおける観測値と予測値の二乗平均平方根 (Root Mean Square) を示す。

以上の結果から、カメラのフレームレートを高速にすることによって、指先位置の推定精度が向上していることが確認出来る。この結果をもとに、これ以降の実験では 200fps では $Q18$ を、30fps では $Q25$ をカルマンフィルタのパラメータとして用いることとした。

5.4 実験：指先位置予測精度向上による操作への影響

本実験では、高速ビジョンを用いることによって指先位置予測精度を向上させ、指先位置と仮想物体位置の位置ずれを解消し、システムの操作性や操作の実体感を向上させることを目的としている。そこで、以下の様な簡単なタスクを実装し、評価実験を行った。

5.4.1 タスク

被験者には単純なタスクを課した。被験者は、図 5.14 に示すように、ディスプレイ右方向から人差し指を伸ばした状態でカメラの前に手を差し出す。指先が差し出されると、システムは自動的に被験者の指先位置をトラッキングし、フレームレートに応じて指先位置の予測を行う。

図 5.15 に示すように、被験者の指先位置に合わせて、画面上にはマーカーが表示される。予め被験者には指を移動させ、マーカーを画面中央に移動させる。マーカーが中央に移動したことを確認すると、ディスプレイ上方から緑色の円形のオブジェクトが一定速度で落下してくる。なお、オブジェクトが落下し始める位置から落下し終わるまでの距離は、実空間上で 200[mm] である。

被験者にはオブジェクトが落下開始する前に掛け声で合図を送るが、どの位置からどの速度でオブジェクトが落下してくるかは被験者には伝えない。被験者には、落下してくるオブジェクトに対し、マーカーを移動させてマーカーの中央でオブジェクトを受け止めるよう指示した。

RMS	30fps	200fps
Q1	18.0013	4.7585
Q2	30.2788	2.7671
Q3	36.3855	2.2681
Q4	17.9660	4.7417
Q5	17.9660	2.6314
Q6	47.6008	1.9590
Q7	17.9623	4.7382
Q8	33.8686	2.621
Q9	50.1798	1.9189
Q10	17.7156	4.8298
Q11	29.8826	2.7979
Q12	36.2049	2.2682
Q13	17.6594	4.8168
Q14	33.0806	2.6692
Q15	49.0887	1.9523
Q16	17.6534	4.8153
Q17	33.5541	2.6578
Q18	52.4798	1.9151
Q19	17.6863	4.8390
Q20	29.8410	2.8009
Q21	36.1853	2.2684
Q22	17.6278	4.8249
Q23	33.0449	2.6738
Q24	49.2703	1.9523
Q25	17.6216	4.8244
Q26	33.5209	2.6626
Q27	52.7713	1.9155

表 5.3. 200fps, 30fps における指先位置予測の二乗平均平方根

オブジェクトが落下してくる位置は全部で9箇所とし、マーカーのある中央を中心に左右に均等に4箇所ずつ配置した。なお、実空間における落下位置の距離は、中央を基準として順に1.75[cm]間隔で配置し、図5.15にあるように、左から順にA、B、C、…、Iとする。また、マーカーの落下速度も5通り設定した。実空間における速度に換算すると、遅い順に200[mm/s]、240[mm/s]、280[mm/s]、320[mm/s]、360[mm/s]である。

今回は、全ての落下位置と速度の組合せ、及びフレームレートを代えて一回ずつ実験を行っ

た。すなわち、あるフレームレートの条件下で、落下してくるオブジェクトを受け取るタスクは45回行ったため、都合90回タスクを行った。なお、全ての実験の開始時に、マーカーは必ず一度画面中央に移動させてから実験を行った。また、45回の落下の順序は、ランダムに決定した。

今回の実験では、オブジェクトが落下しきった瞬間の、カメラで取得した画像におけるオブジェクトの中心位置とその時予測されていた指先位置との距離 [pixel] を誤差として記録した。

5.4.2 被験者

男性4名の被験者で実験を行った。被験者は、どちらかのフレームレートで先に45回オブジェクトを受け止めるタスクを行い、その次にフレームレートを代えて再度タスクを行った。どの被験者も右利きで、右手を使って操作を行った。

5.4.3 結果

まず最初に、被験者ごとの記録された誤差の二乗平均平方根を表5.4に示す。表を見てもわかるように、30[fps]の時のほうが200[fps]の時に比べて二乗平均平方根が大きな値を取っている。

	30fps	200fps
被験者1	16.259	9.7938
被験者2	14.027	12.961
被験者3	17.946	14.263
被験者4	18.413	9.9021

表 5.4. オブジェクトと予測された位置の二乗平均平方根 (pixel)

また、オブジェクトの落下位置ごとの二乗平均平方根を図5.16、落下速度ごとの二乗平均平方根を図5.17に示す。オブジェクトの落下位置について、ほとんどの箇所において差がない、もしくは30fpsの方が二乗平均平方根が大きいことが読み取れる。一方、落下速度ごとの二乗平均平方根では、どの速度においても30fpsの方が200fpsよりも二乗平均平方根の値が大きいという結果を得た。また、速度が200[mm/s]と360[mm/s]の場合、フレームレートを変えた時の二乗平均平方根に大きな差があることもわかる。

5.4.4 考察

実験の結果、フレームレートを向上させることで、どの被験者においても仮想物体とインタラクションを行う際の位置精度を向上させることが出来た。

より詳細に見ると、オブジェクトの落下位置ごとの二乗平均平方根では、HやIにおける二乗平均平方根の差がほかの位置に比べて大きいことがわかる。これは、被験者にとって指を右

に動かす方が指の動きを止めやすかった事が原因と考えられる。指先を左に動かす場合は右に動かす場合に比べて力も入れやすく動かしやすいものの、一方で力が入りすぎてしまった場合指先を制止するのも困難であると考えられるからである。一方、指先を右に動かす場合は上腕や肩を利用して手の動きを止めることも容易であるため、このような結果が得られたと考えられる。

また、オブジェクトの移動速度ごとの二乗平均平方根では、200[mm/s]と360[mm/s]の場合の二乗平均平方根の差が大きかった。360[mm/s]については、オブジェクトの移動速度が速かったため、200fpsではカーソルが指先位置を正確に予測できたのに対し、30fpsではオーバーシュートしてしまい、二乗平均平方根に差が生まれたと考えられる。一方200[mm/s]の場合は、他の速度に比べて比較的位置合わせをする余裕があったものの、指先位置を固定しようとした結果振戦が発生してしまい、予測位置が大きくずれたことが理由として考えられる。

5.5 まとめ

本章では、空中操作における仮想物体とのインタラクションの実体感の向上を目的とし、高速ビジョンと裸眼立体視ディスプレイを用いたインタラクションシステムを提案した。提案システムをデモ発表等した結果、フレームレートを向上させたことによって、ボタンに触れた時の仮想物体の応答が早くなり、実体感が向上したという意見を得た。しかし一方で、指を前後に動かした場合など、急激な速度変化を伴う指先の動きに対しては、仮想物体の動きが指先の動きに追従せず、実体感が低いという課題も見つかった。

そこで、本研究では高速ビジョンを用いて得たデータに対してカルマンフィルタを用いた指先位置予測を行うことで実際の指先位置と描画される仮想物体との位置ずれを解消し、操作の実体感を向上させることを目的とした実験を行った。

実験の結果、高速な指先の動きを伴う操作において、高速ビジョンを用いた指先位置予測を行うことでより正確に仮想物体とのインタラクションが行えることが確認できた。

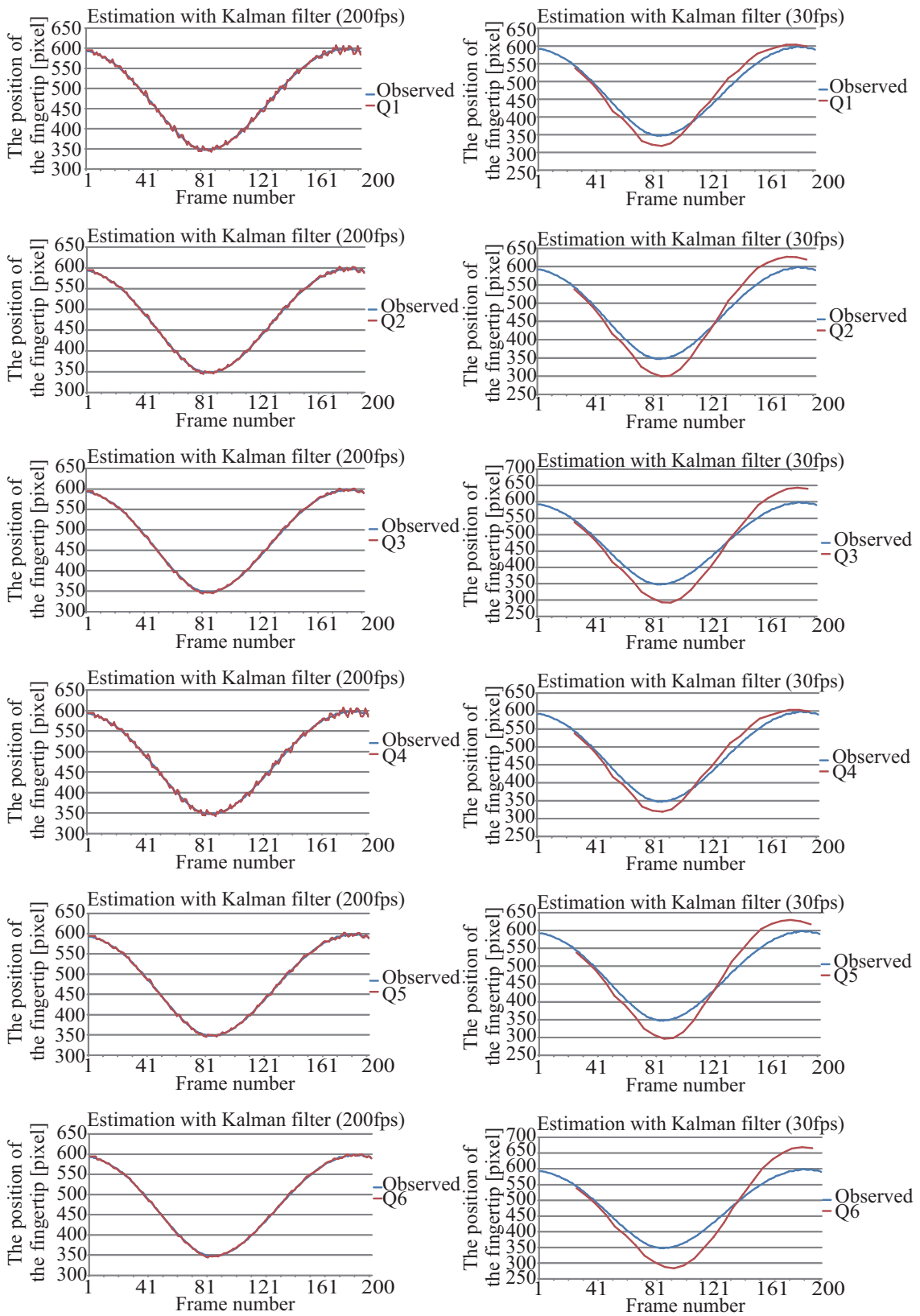


図 5.9. Q1 ~ Q6 下における指先位置予測

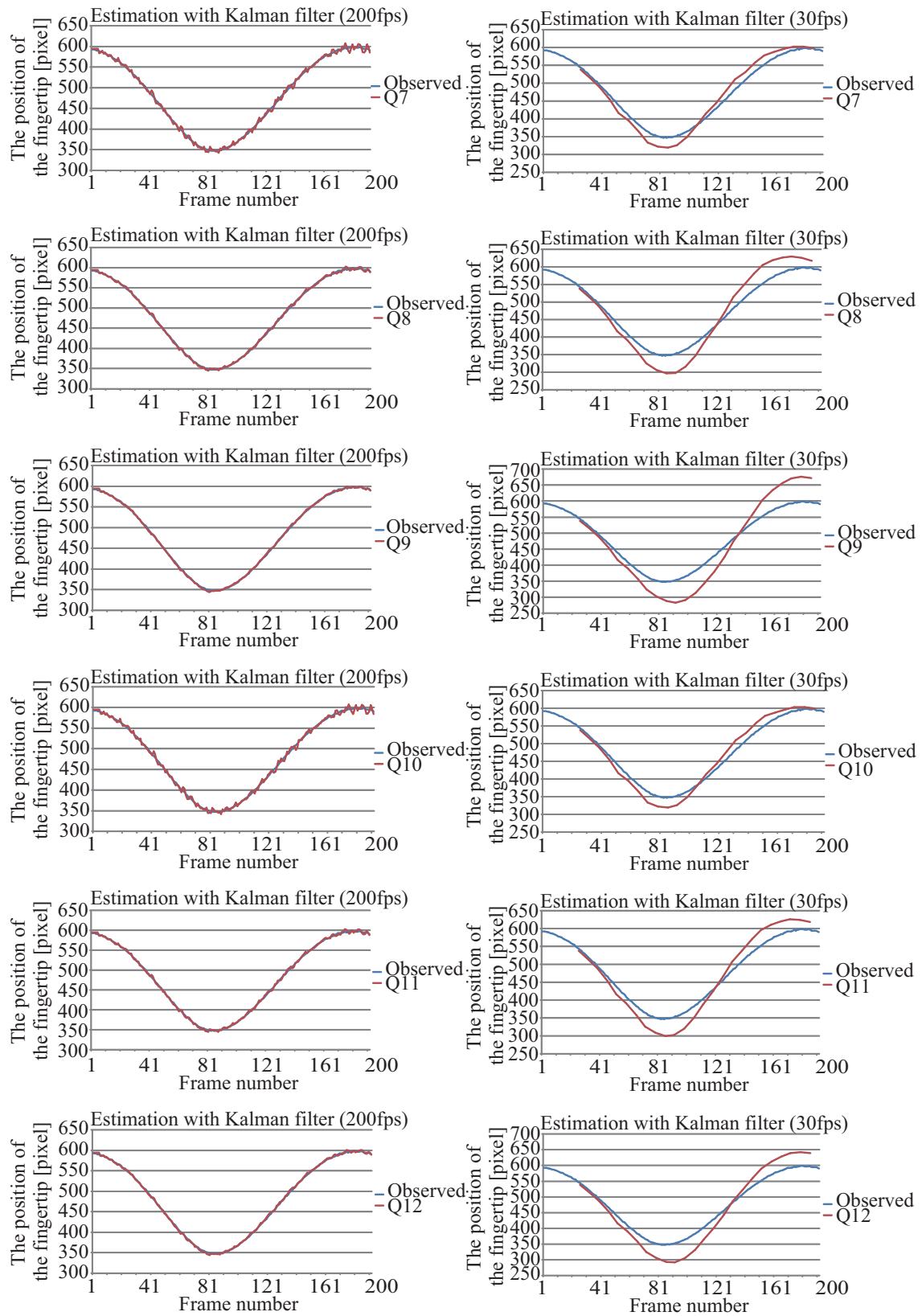


図 5.10. Q7 ~ Q12 下における指先位置予測

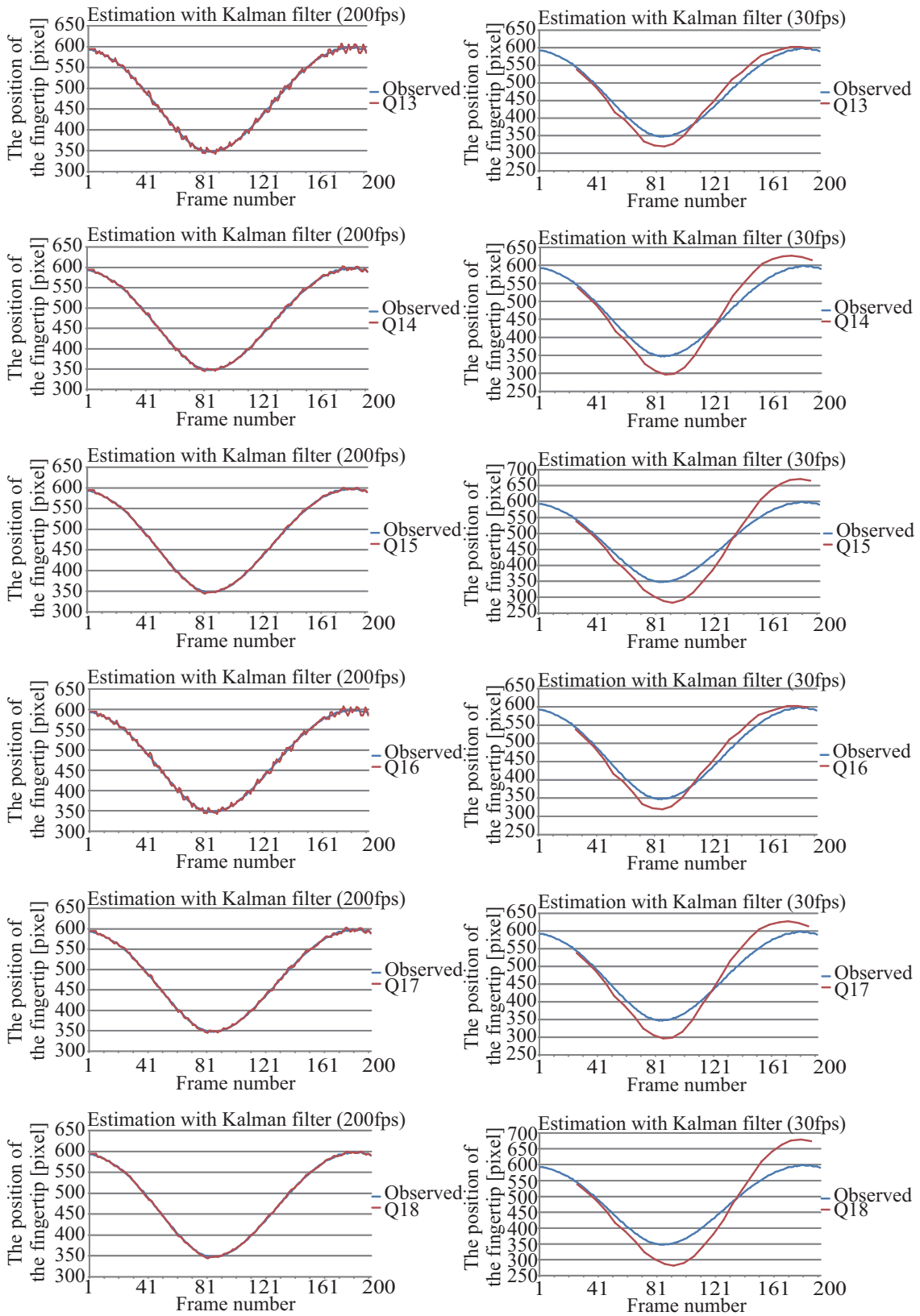


図 5.11. Q13 ~ Q18 下における指先位置予測

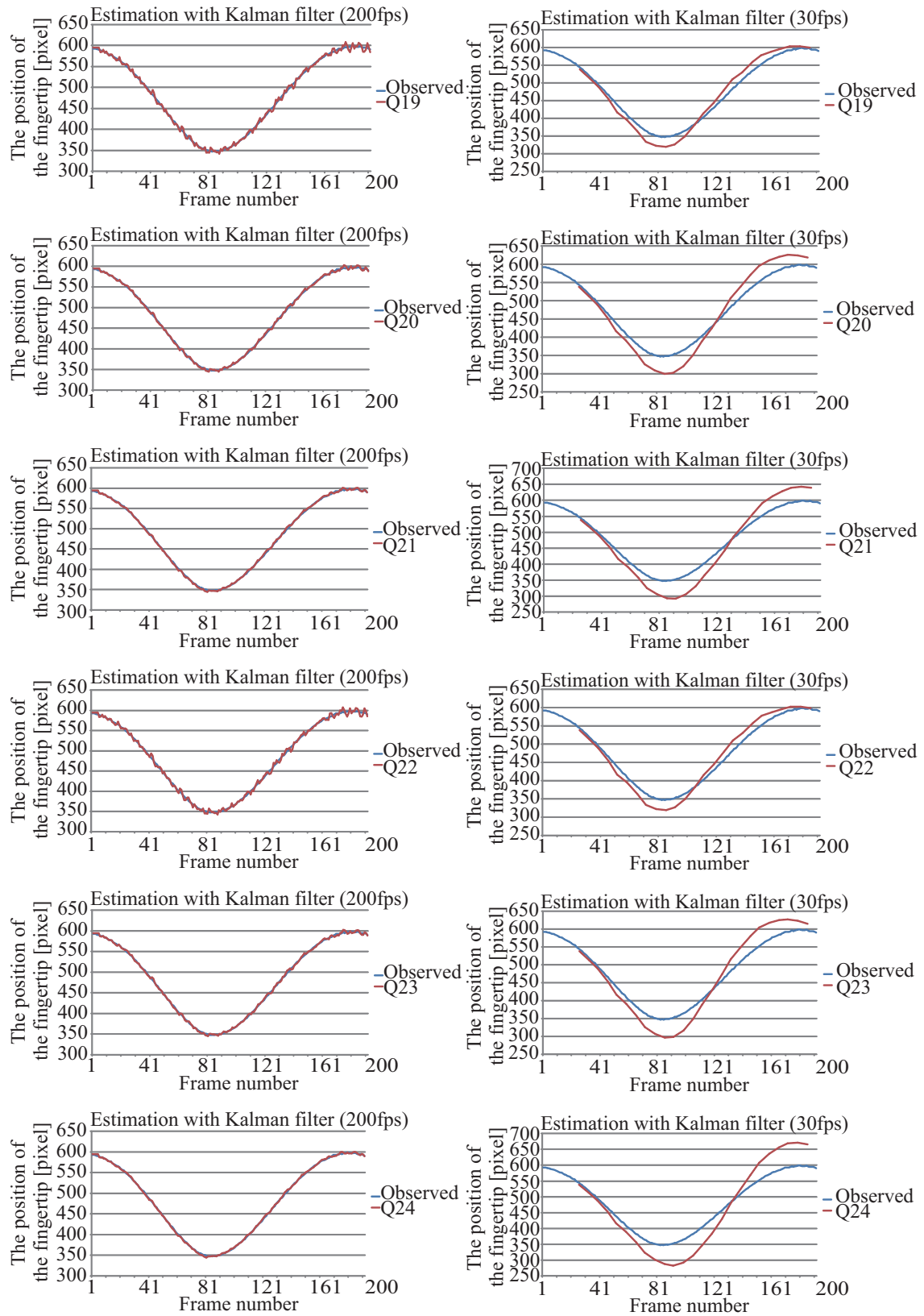


図 5.12. Q19 ~ Q24 下における指先位置予測

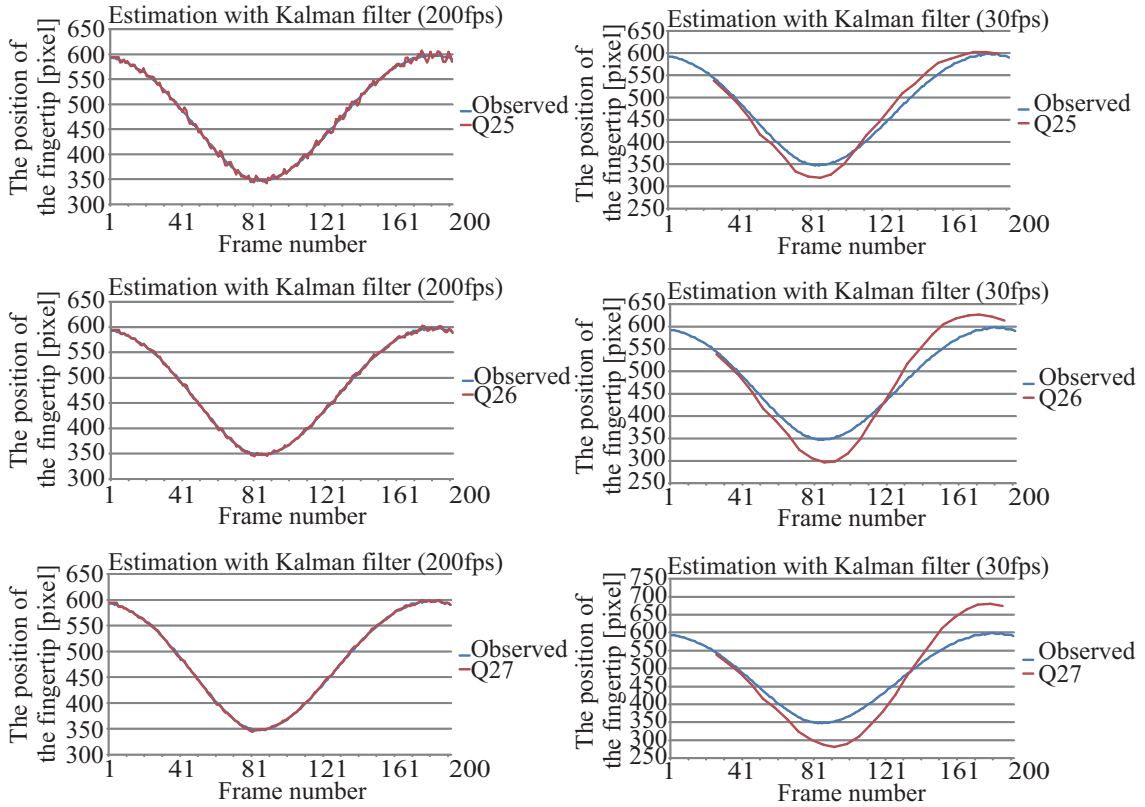


図 5.13. Q25 ~ Q27 下における指先位置予測

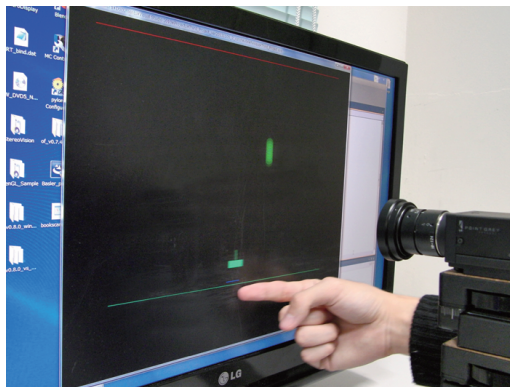


図 5.14. 実験環境

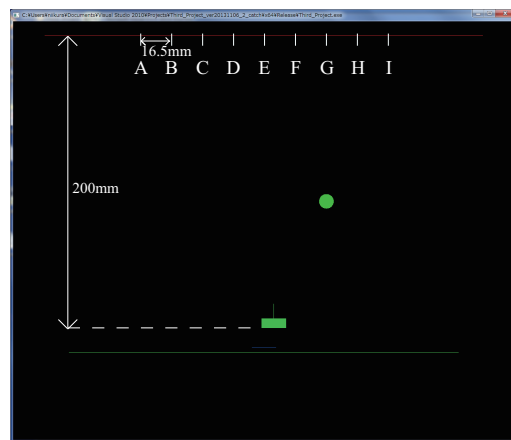


図 5.15. タスクの様子

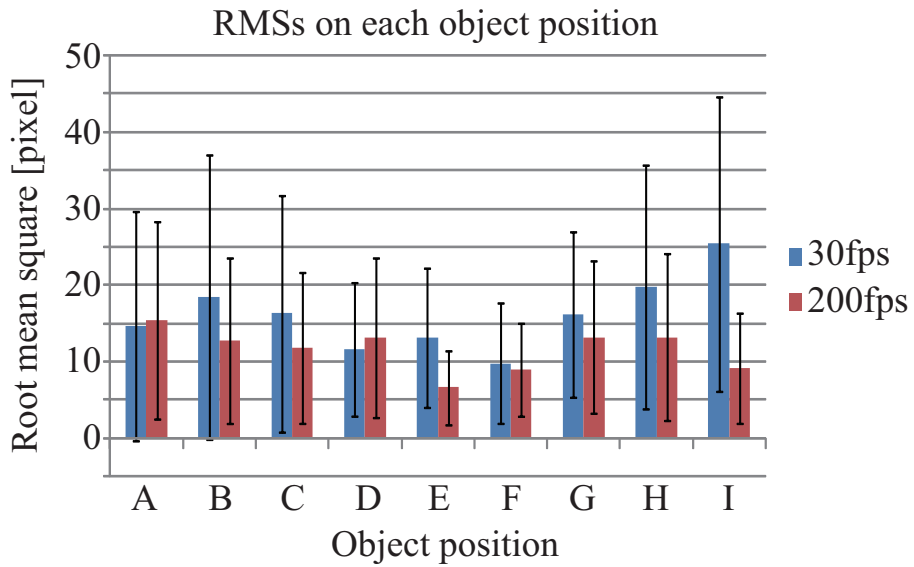


図 5.16. オブジェクトの落下位置ごとの二乗平均平方根

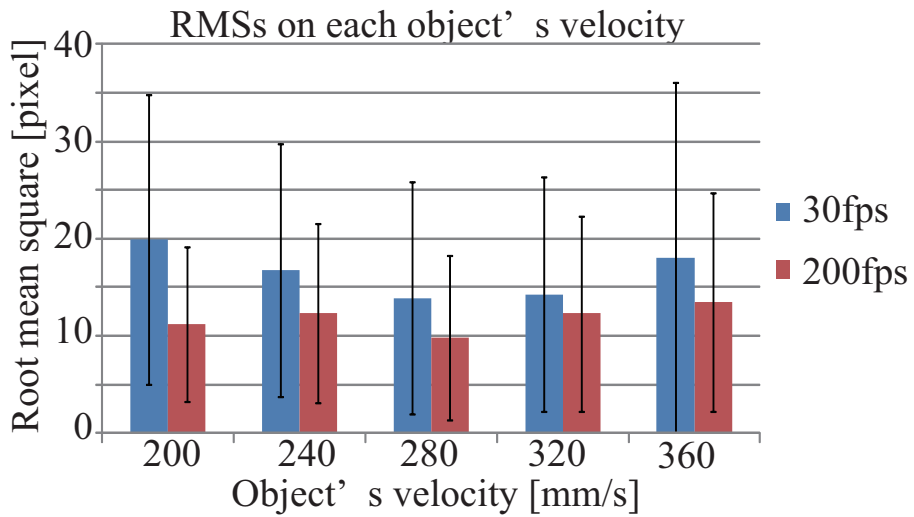


図 5.17. オブジェクトの落下速度ごとの二乗平均平方根

第6章

結論

本論文では、携帯機器での入力における問題を取り上げ、高速ビジョンを用いた手入力システムの必要性、および有効性を論じた。

以下、本章以前に議論・検討を行った内容を章ごとにまとめる。

第1章

第1章では、携帯機器向けインタフェースにおける入力の課題や制約などをまとめ、それらの課題や制約を克服する方法を検討した。検討の結果、手を用いた入力が最も自然かつ適切な手法であると結論づけた。そして、手を用いた入力を実現する方法として、高速ビジョンを用いた手入力システムが最適な手法であることを述べた。

最後に、そのようなシステムとして、空中操作インタフェースと Anywhere Surface Touch を提案することを述べた。また、空中操作における仮想物体とのインタラクションをテーマに、操作の実体感を向上させることを目的とした研究を行うことを述べた。

第2章

第2章では、第1章で述べた携帯機器向け入力における課題や制約を解決することを目標とした種々の研究を紹介し、それらの手法のメリット・デメリットを議論した。この議論に基づき、求められる新たな携帯機器向けインタフェースを論じ、情報提示の観点から2つの考える解決方法を述べた。具体的には、既存の携帯機器の形状を維持した場合における解決手法と、より小型化した携帯機器における解決手法があることを述べた。

既存の携帯機器のような形状が維持される場合は、空中を操作領域とすることが望ましいことを示した。また、使用形態などを考慮すると非装着型のインタフェースが望ましいことを示した。そして、そのようなインタフェースとして空中操作インタフェースを提案した。一方で、より小型化した携帯機器において高速に複雑な操作を行うためには、ウェアラブルなシステムで実空間の様々な面を利用することが望ましい事を指摘した。そして、そのような入力を実現するインタフェースとして Anywhere Surface Touch を提案した。

また、空中操作においては、操作が三次元空間で行われるにもかかわらず、現状では情報の提示が2次元に制約されているなど実体感に乏しいという課題が存在することを述べ、課題解決のために裸眼立体視ディスプレイを用いたインタラクションシステムの提案を行った。また、仮想物体とのインタラクションにおいて指先位置と仮想物体との位置ずれが生じることによっても実体感が損なわれることを課題とし、高速ビジョンを用いた指先位置予測によって操作性を向上させることを提案した。

第3章

本章では、非装着型の新たな携帯機器向けインタフェースに求められる要求をまとめた。それらの要求を満たす新たなインタフェースとして単眼の高速カメラを用いて指先の動きを3次元トラッキングし空中でのタイピングが可能な、空中モーションインタフェースを提案した。また、高速な単眼カメラを用いて指先領域の安定した抽出、指先のトラッキング、入力動作の検出を実現する手法を具体的に提案した。

システム評価の結果、提案システムがディスプレイの大きさに対して131%の大きさの操作領域を実現したことを確認した。特に、水平方向についてはディスプレイの2倍以上の操作領域を実現した。また、指先位置の推定において、水平方向に平均誤差2[pixel]以内と高精度に行えることを示した。同様に、奥行き方向についても高い精度で推定が行えていることを確認した。この結果は、提案システムが従来よりも遥かに高精度な操作を実現したことを示すといえる。また、赤外線LEDと赤外線フィルタを用いることで、提案システムは背景に写り込んだ顔や照明に対して極めてロバストに動作することを確認した。また、暗い部屋の中でもロバストに指先トラッキングが行えることも確認した。加えて、1フレームあたり約7.2[ms]という高速な処理も実現した。入力動作検出についても、高い認識率が確認できた。

フレームレートに関する実験では、15[fps]では有意にタスク終了までの時間が増加するという結果を得た。レイテンシについてはいずれの場合においても操作時間に有意な差が生まれ、非常に小さいレイテンシの差であってもレイテンシを減少させる事が操作効率の向上に有意に影響を与えるという結果を得た。レイテンシを減少させるためにはフレームレートを高くする事も必要となるため、間接的に高フレームレートの必要性も確認できた。

振動フィードバックの実験の結果、振動フィードバックを付加することによってタスクを終了するまでの時間、およびミスタイプの回数を有意に減少させることが出来る事を確認した。従って、機器を把持している手のみ振動フィードバックを提示することによってもシステムの操作効率を改善することが可能であるといえる。

第4章

本章では、小型化する携帯機器がどのようにあるべきかを述べ、装着型の携帯機器のコンセプトを実現するものとして、小型カメラとコンタクトマイクを一体化して手首に取り付けるAnywhere Surface Touchを提案した。ASTにより、様々な平面上での複雑な入力を実現した。

AST では画像を用いて指やジェスチャの認識、音を用いた接触状態の認識を行いそれらの情報を統合することで多様な入力を実現した。

様々な環境で利用可能であることを確認するために実験を行った結果、机やクッション、壁、脚の上など身近にある様々な面上で入力可能であることを確認した。また、環境や姿勢などによっては実行しやすい、もしくは実行しにくいジェスチャ入力があることを発見した。将来的に本システムを用いてアプリケーションを開発する場合には、人間の行い易い動作を利用してアプリケーションを開発するといった事が必要になる。

将来的には、指の三次元形状を推定しつつ設置面との接触を判定することで奥行方向へのジェスチャ入力の認識を実現することを目標としている。また、音声と画像の統合についても、よりロバストかつ高精度な認識を実現出来ると考えている。

第5章

本章では、空中操作における仮想物体とのインタラクションの実体感の向上を目的とし、高速ビジョンと裸眼立体視ディスプレイを用いたインタラクションシステムを提案した。提案システムをデモ発表等した結果、フレームレートを向上させたことによって、ボタンに触れた時の仮想物体の応答が早くなり、実体感が向上したという意見を得た。しかし一方で、指を前後に動かした場合など、急激な速度変化を伴う指先の動きに対しては、仮想物体の動きが指先の動きに追従せず、実体感が低いという課題も見つかった。

そこで、本研究では高速ビジョンを用いて得たデータに対してカルマンフィルタを用いた指先位置予測を行うことで実際の指先位置と描画される仮想物体との位置ずれを解消し、操作の実体感を向上させることを目的とした実験を行った。

実験の結果、高速な指先の動きを伴う操作において、高速ビジョンを用いた指先位置予測を行うことでより正確に仮想物体とのインタラクションが行えることが確認できた。

謝辞

修士課程での2年間、そして博士課程での3年間と、石川（小室）奥研究室での充実した研究生活を思い返すと、様々な出来事があったなと感慨深い気持ちになります。日々研究室で暮らしていた時はあんなにも長かったのに、こうして思い返すとあっという間の出来事のようにも思えます。この5年間、また学部時代も含めれば9年間、学内・学外を問わず、ここには書ききれないほどの、本当に多くの方々にお世話になりました。この場を借りて厚く御礼申し上げます。

石川 正俊教授、このような貴重な研究の場を提供していただき、本当にありがとうございました。例えば、修士で進学する研究室に悩んでいた時に、石川先生のお話を伺って石川研に進学したいと思った事を覚えています。当時は、まさか博士課程まで進学し、5年間も石川研に在籍することになるとは思ってもいませんでした。石川先生も大変お忙しいため、直接ご指導頂く機会はそれほど多くは得られませんでした。折にふれて頂くアドバイスにその都度考えさせられ、研究の糧となりました。また、今年に入ってからは、就職活動に関してもアドバイスを頂きました。石川先生には大変感謝しております。今後は卒業し、社会人としての道を歩みますが、これからなにとぞよろしく願いいたします。

渡辺 義浩助教には、ビジョンアーキテクチャグループのリーダーとして、研究を進めるにあたって直接的な指導をして下さり、大変お世話になりました。渡辺さんは、僕が博士課程に進学すると同時にビジョンアーキテクチャグループのリーダーになりました。ビジョンアーキテクチャグループは修士の学生や卒論生など指導が必要な学生も多く、非常に大変なお仕事だったのではないかと感じています。正直に言うと、最初は指導教員が小室先生から渡辺さんへ変わったことや、僕自身修士から博士に進んだことで、研究の進め方などに関して、どのように進めていくのがいいのかなど、少し戸惑っていました。（逆の立場に立つと、渡辺さんにとっても、グループのリーダーを引き継ぎ環境が変わる中で、どうするのがいいのか迷っていたのかもしれないと、今になって思います。）それでも、僕に対して博士課程学生としての様々なアドバイスや、評価などを下さるうちに、少しずつ研究の進め方などを理解し、実行出来るようになったのではないかと考えています。渡辺さんの研究に対する真剣な姿勢や楽しさは、僕にとっても大きな刺激となりました。本当にありがとうございました。これからもどうぞよろしく願いいたします。

共にビジョンアーキテクチャグループのメンバーとして研究を進めてきた、山田 雅宏様、望戸 雄史先輩、郷原 啓生君、多田 圭佑君、野口 翔平君、松本 康平君、宮下 令央君、勝木 祐伍

君、平野 正浩君、渡邊 千紘さんにも大変お世話になりました。同じグループのメンバーとして、実験への協力など、心から感謝しています。また、既に卒業したメンバーですが、久保 伸太郎君や松谷 淳史君、三浦 洋平君とは同じ実験室で過ごしたメンバーとして、楽しくまた充実した研究室での日々を送る事が出来ました。彼らにも、この場を借りて感謝の意を表したいと思います。

埼玉大学の小室 孝准教授には、修士時代の指導教員として大変お世話になりました。また、博士課程に進んでからも、埼玉大学で産学官連携研究員として雇用していただき、博士課程での研究や論文を進める上でも多くの助言を頂きました。僕が博士課程に進もうと考えたのは、間違いなく、修士時代に小室先生にご指導いただいて研究の面白さを知ったからだと考えています。小室先生には感謝してもしきれません。本当にありがとうございました。今後ともよろしくお願いいたします。

満洲邦彦教授、篠田裕之教授、奥寛雅講師、牧野泰才講師には、お忙しい中本博士論文の副査をお引き受け頂き、お世話になりました。それぞれの分野において第一線でご活躍されている先生方から貴重なご指摘やご助言を頂いたことを大変光栄に思っております。先生方のご助言があったからこそ、博士論文を仕上げる事ができたと思います。ここに深くお礼申し上げます。

最後に、これまでの学生生活を応援して下さった家族、両親に心より感謝いたします。特に、博士課程1年の冬に体調を崩し、以降1年以上に渡って原因不明の体調不良に悩まされた時期は、本当に心配をかけたと思います。そんな時も変わらず支えてくれた家族には、感謝の言葉しかありません。これまで、本当にありがとうございました。

参考文献

- [1] K. A. Siek, Y. Rogers, and K. H. Connelly. Fat finger worries: how older and younger users physically interact with pdas. In *Proceedings of the 2005 IFIP TC13 international conference on Human-Computer Interaction*, INTERACT' 05, pp. 267–280. Springer-Verlag, 2005.
- [2] C. Shan. *Multimedia Interaction and Intelligent User Interfaces*. Advances in Pattern Recognition. Springer London, 2010.
- [3] M. F. BenZeghiba, R. De Mori, O. Deroo, S. Dupont, T. Erbes, D. Jouviet, L. Fissore, P. Laface, A. Mertins, C. Ris, R. C. Rose, V. Tyagi, and C. J. Wellekens. Automatic speech recognition and speech variability: A review. *Speech Communication*, Vol. 49, No. 10-11, pp. 763–786, 2007.
- [4] J. Rico and S. Brewster. Usable gestures for mobile interfaces: evaluating social acceptability. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pp. 887–896. ACM, 2010.
- [5] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly. Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding*, Vol. 108, No. 1-2, pp. 52–73, 2007.
- [6] 矢部京之助, 大築立志, 笠井達哉. 入門運動神経生理学—ヒトの運動の巧みさを探る—. 市村出版, 2003.
- [7] D. Vogel and P. 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. ACM, 2007.
- [8] D. Wigdor, C. Forlines, P. Baudisch, J. Barnwell, and C. Shen. Lucid touch: a see-through mobile device. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, UIST '07, pp. 269–278. ACM, 2007.
- [9] P. Baudisch and G. Chu. Back-of-device interaction allows creating very small touch devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pp. 1923–1932. ACM, 2009.
- [10] A. Butler, S. Izadi, and S. Hodges. Sidesight: multi-”touch” interaction around small devices. In *Proceedings of the 21st annual ACM symposium on User interface software and*

- technology, UIST '08, pp. 201–204. ACM, 2008.
- [11] J. M. Rehg and T. Kanade. Visual tracking of high dof articulated structures: an application to human hand tracking. In *Proceedings of the Third European Conference-Volume II on Computer Vision - Volume II*, ECCV '94, pp. 35–46. Springer-Verlag, 1994.
- [12] K. Hoshino and T. Tanimoto. Real time search for similar hand images from database for robotic hand control. In *13th IEEE International Workshop on Robot and Human Interactive Communication*, ROMAN '04, pp. 377–382. IEEE, 2004.
- [13] V. Athitsos, H. Wang, and A. Stefan. A database-based framework for gesture recognition. *Personal and Ubiquitous Computing*, Springer-Verlag, Vol. 14, No. 6, pp. 511–526, 2010.
- [14] R. Y. Wang and J. Popović. Real-time hand-tracking with a color glove. *ACM Transactions on Graphics*, Vol. 28, No. 3, pp. 63:1–63:8, 2009.
- [15] A. Causo, M. Matsuo, E. Ueda, K. Takemura, Y. Matsumoto, J. Takamatsu, and T. Ogasawara. Hand pose estimation using voxel-based individualized hand model. In *Proceedings of the 2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, AIM '09, pp. 451–456. IEEE, 2009.
- [16] M. Schlattman and R. Klein. Simultaneous 4 gestures 6 dof real-time two-hand tracking without any markers. In *Proceedings of the 2007 ACM symposium on Virtual reality software and technology*, VRST '07, pp. 39–42. ACM, 2007.
- [17] I. Oikonomidis, N. Kyriazis, and A. Argyros. Markerless and efficient 26-dof hand pose recovery. In *Proceedings of the 10th Asian Conference on Computer Vision - Volume Part III*, ACCV '10, pp. 744–757. Springer-Verlag, 2011.
- [18] I. Oikonomidis, N. Kyriazis, and A. A. Argyros. Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pp. 2088–2095. IEEE, 2011.
- [19] W. Y. Chang, C. S. Chen, and Y. D. Jian. Visual tracking in high-dimensional state space by appearance-guided particle filtering. *IEEE Transactions on Image Processing*, Vol. 17, No. 7, pp. 1154–1167, 2008.
- [20] V. Athitsos and S. Sclaroff. Estimating 3d hand pose from a cluttered image. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR '03, pp. 18–20. IEEE, 2003.
- [21] I. Oikonomidis, N. Kyriazis, and A. Argyros. Efficient model-based 3d tracking of hand articulations using kinect. In *Proceedings of the British Machine Vision Conference*, pp. 101.1–101.11. BMVA Press, 2011.
- [22] P. Mistry and P. Maes. Sixthsense: a wearable gestural interface. In *ACM SIGGRAPH ASIA 2009 Sketches*, SIGGRAPH ASIA '09, pp. 11:1–11:1. ACM, 2009.
- [23] C. Harrison and S. E. Hudson. Abracadabra: wireless, high-precision, and unpowered finger input for very small mobile devices. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, UIST '09, pp. 121–124. ACM, 2009.

- [24] K. Y. Chen, K. Lyons, S. White, and S. Patel. utrack: 3d input using two magnetic sensors. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, pp. 237–244. ACM, 2013.
- [25] L. Chan, R. H. Liang, M. C. Tsai, K. Y. Cheng, C. H. Su, M. Y. Chen, W. H. Cheng, and B. Y. Chen. Fingerpad: Private and subtle interaction using fingertips. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, pp. 255–260. ACM, 2013.
- [26] Y. Watanabe, T. Hatanaka, T. Komuro, and M. Ishikawa. Human gait estimation using a wearable camera. In *Proceedings of the 2011 IEEE Workshop on Applications of Computer Vision*, WACV '11, pp. 276–281. IEEE, 2011.
- [27] Yoshihiro Watanabe, Tetsuo Hatanaka, Takashi Komuro, and Masatoshi Ishikawa. Human gait estimation using a wearable camera(special issue: Real world imaging). *Transactions of the Virtual Reality Society of Japan*, Vol. 17, No. 3, pp. 219–229, 2012.
- [28] T. Shiratori, H. S. Park, L. Sigal, Y. Sheikh, and J. K. Hodgins. Motion capture from body-mounted cameras. *ACM Transactions on Graphics*, Vol. 30, No. 4, pp. 31:1–31:10, 2011.
- [29] D. Kim, O. Hilliges, S. Izadi, A. D. Butler, J. Chen, I. Oikonomidis, and P. Olivier. Digits: freehand 3d interactions anywhere using a wrist-worn gloveless sensor. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, UIST '12, pp. 167–176. ACM, 2012.
- [30] B. Howard and S. Howard. Lightglove: wrist-worn virtual typing and pointing. In *Proceedings of the 5th IEEE International Symposium on Wearable Computers*, ISWC '01, pp. 172–173. IEEE, 2001.
- [31] F. Ahmad and P. Musilek. Ubihand: a wearable input device for 3d interaction. In *ACM SIGGRAPH 2006 Research posters*, SIGGRAPH '06. ACM, 2006.
- [32] F. Ahmad and P. Musilek. A keystroke and pointer control input interface for wearable computers. In *Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications*, PERCOM '06, pp. 2–11. IEEE, 2006.
- [33] C. Harrison, H. Benko, and A. D. Wilson. Omnitouch: wearable multitouch interaction everywhere. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, UIST '11, pp. 441–450. ACM, 2011.
- [34] C. Harrison, D. Tan, and D. Morris. Skininput: appropriating the body as an input surface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pp. 453–462. ACM, 2010.
- [35] M. Ogata, Y. Sugiura, Y. Makino, M. Inami, and M. Imai. Senskin: Adapting skin as a soft interface. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, pp. 539–544. ACM, 2013.
- [36] K. Nakamatsu and H. Shinoda. Wristband-shaped input interface using user's back of hand. In *Proceedings of the IEEE World Haptics Conference Demo*, WHC. IEEE, 2011.

- [37] A. D. Wilson. Playanywhere: a compact interactive tabletop projection-vision system. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*, UIST '05, pp. 83–92. ACM, 2005.
- [38] A. Agarwal, S. Izadi, M. Chandraker, and A. Blake. High precision multi-touch sensing on surfaces using overhead cameras. In *Proceedings of the Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, TABLETOP '07, pp. 197–200. IEEE, 2007.
- [39] H. Roeber, J. Bacus, and C. Tomasi. Typing in thin air: the canesta projection keyboard - a new method of interaction with electronic devices. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '03, pp. 712–713. ACM, 2003.
- [40] Nicolas Villar, S. Izadi, D. Rosenfeld, H. Benko, J. Helmes, J. Westhues, S. Hodges, E. Ofek, A. Butler, X. Cao, and B. Chen. Mouse 2.0: multi-touch meets the mouse. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, UIST '09, pp. 33–42. ACM, 2009.
- [41] C. Harrison and S. E. Hudson. Minput: enabling interaction on small mobile devices with high-precision, low-cost, multipoint optical tracking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pp. 1661–1664. ACM, 2010.
- [42] K. Yatani and K. N. Truong. Bodyscope: a wearable acoustic sensor for activity recognition. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, pp. 341–350. ACM, 2012.
- [43] T. Deyle, S. Palinko, E.S. Poole, and T. Starner. Hambone: A bio-acoustic gesture interface. In *Proceedings of the 2007 11th IEEE International Symposium on Wearable Computers*, ISWC '07, pp. 3–10. IEEE, 2007.
- [44] C. Harrison and S. E. Hudson. Scratch input: creating large, inexpensive, unpowered and mobile finger input surfaces. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, UIST '08, pp. 205–208. ACM, 2008.
- [45] M. Fukumoto and T. Sugimura. Active click: tactile feedback for touch panels. In *CHI '01 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '01, pp. 121–122. ACM, 2001.
- [46] J. Rekimoto. Senseablerays: opto-haptic substitution for touch-enhanced interactive spaces. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '09, pp. 2519–2528. ACM, 2009.
- [47] T. Hoshi, M. Takahashi, T. Iwamoto, and H. Shinoda. Noncontact tactile display based on radiation pressure of airborne ultrasound. *IEEE Transactions on Haptics*, Vol. 3, No. 3, pp. 155–165, 2010.
- [48] K. Yoshino and H. Shinoda. Contactless touch interface with visuo-acoustic screen. In *The 18th Annual Conference of the Virtual Reality Society of Japan*, pp. 472–475. The Virtual Reality Society of Japan, 9 2013.

- [49] H. Ishii and B. Ullmer. Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, CHI '97, pp. 234–241. ACM, 1997.
- [50] S. Jordà, G. Geiger, M. Alonso, and M. Kaltenbrunner. The reactable: exploring the synergy between live music performance and tabletop tangible interfaces. In *Proceedings of the 1st international conference on Tangible and embedded interaction*, TEI '07, pp. 139–146. ACM, 2007.
- [51] J. Patten, H. Ishii, J. Hines, and G. Pangaro. Sensetable: a wireless object tracking platform for tangible user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '01, pp. 253–260. ACM, 2001.
- [52] T. Yoshida, S. Kamuro, K. Minamizawa, H. Nii, and S. Tachi. Repro3d: full-parallax 3d display using retro-reflective projection technology. In *ACM SIGGRAPH 2010 Emerging Technologies*, SIGGRAPH '10, pp. 20:1–20:1. ACM, 2010.
- [53] H. Benko, R. Jota, and A. Wilson. Miragetable: freehand interaction on a projected augmented reality tabletop. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pp. 199–208. ACM, 2012.
- [54] T. Hoshi, M. Takahashi, K. Nakatsuma, and H. Shinoda. Touchable holography. In *ACM SIGGRAPH 2009 Emerging Technologies*, SIGGRAPH '09, pp. 23:1–23:1. ACM, 2009.
- [55] O. Hilliges, D. Kim, S. Izadi, M. Weiss, and A. Wilson. Holodesk: direct 3d interactions with a situated see-through display. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pp. 2421–2430. ACM, 2012.
- [56] M. Meehan, S. Razzaque, M.C. Whitton, and Jr. Brooks, F.P. Effect of latency on presence in stressful virtual environments. In *Proceedings of the 2003 IEEE Virtual Reality*, IEEE VR '03, pp. 141–148. IEEE, 2003.
- [57] R. Jota, A. Ng, P. Dietz, and D. Wigdor. How fast is fast enough?: A study of the effects of latency in direct-touch pointing tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pp. 2291–2300. ACM, 2013.
- [58] R. Wang, S. Paris, and J. Popović. 6d hands: markerless hand-tracking for computer aided design. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, UIST '11, pp. 549–558. ACM, 2011.
- [59] Y. Hirobe, T. Niikura, Y. Watanabe, T. Komuro, and M. Ishikawa. Vision-based input interface for mobile devices with high-speed fingertip tracking. In *ACM UIST 2009 Demonstrations*, UIST '09. ACM, 2009.
- [60] T. Niikura, Y. Hirobe, Y. Watanabe, T. Komuro, and M. Ishikawa. 3d input interface for mobile devices. In *VRIC 2010 - Laval Virtual Demo Session*, VRIC '10, pp. 297–298. ACM, 2010.
- [61] T. Niikura, Y. Hirobe, A. Cassinelli, Y. Watanabe, T. Komuro, and M. Ishikawa. In-air typing interface for mobile devices with vibration feedback. In *ACM SIGGRAPH 2010 Emerging*

- Technologies*, SIGGRAPH '10, pp. 15:1–15:1, New York, NY, USA, 2010. ACM.
- [62] T. Niikura, Y. Watanabe, T. Komuro, and M. Ishikawa. In-air typing interface: Realizing 3d operation for mobile devices. In *Proceedings of the 2012 IEEE 1st Global Conference on Consumer Electronics*, GCCE '12, pp. 223–227. IEEE, 2012.
- [63] 小室孝, 新倉雄大. 携帯機器向け3次元ジェスチャインタフェースの研究, 次世代ヒューマンインタフェース開発最前線 第3編 第1章 第5節. NTS, 2013.
- [64] 廣部祐樹. 単眼カメラを用いた携帯機器向け入力インターフェースの研究. 東京大学工学部 計数工学科 卒業論文, 2009.
- [65] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, Vol. 56, No. 3, pp. 221–255, feb 2004.
- [66] Microsoft. Guidelines for targeting: Dev center - windows store apps, 2013.
- [67] Nokia. Symbian design guidelines, 3 2012.
- [68] Apple Incorporated. Layout, 2013.
- [69] 前野隆司. ヒトの触覚受容機構: 力学・アナロジー・錯覚という視点から. *The Transactions of the Institute of Electrical Engineers of Japan. E, A Publication of Sensors and Micromachines Associated Society*, Vol. 122, No. 10, pp. 469–473, 2002.
- [70] 新倉雄大, 渡辺義浩, 石川正俊. Anywhere surface touch: 実環境のあらゆる平面を入力平面とするインタフェースの提案. 第17回日本バーチャルリアリティ学会大会大会論文誌, VRSJ2012, pp. 385–388. 日本バーチャルリアリティ学会, 2012.
- [71] P. Atrey, M.A. Hossain, A. El Saddik, and M. Kankanhalli. Multimodal fusion for multimedia analysis: a survey. *Multimedia Systems*, Vol. 16, No. 6, pp. 345–379, 2010.
- [72] J. Y. Bouguet. Camera calibration toolbox for matlab, 2013.
- [73] T. Niikura and T. Komuro. 3d touch panel interface using an autostereoscopic display. In *Proceedings of the 2012 ACM international conference on Interactive tabletops and surfaces*, ITS '12, pp. 295–298. ACM, 2012.
- [74] 新倉雄大, 小室孝. 裸眼立体視ディスプレイを用いた空中操作による三次元グラフィカルユーザーインタフェース. 第6回新画像システム・情報フォトニクス研究討論会 予稿集, 第6回新画像システム・情報フォトニクス研究討論会, pp. 13–14, 2012.
- [75] J. Liang, C. Shaw, and M. Green. On temporal-spatial realism in the virtual reality environment. In *Proceedings of the 4th annual ACM symposium on User interface software and technology*, UIST '91, pp. 19–25. ACM, 1991.
- [76] A. van Rhijn, A. and J. D. Mulder. An analysis of orientation prediction and filtering methods for vr/ar. In *Proceedings of the 2005 IEEE Conference 2005 on Virtual Reality*, VR '05, pp. 67–74. IEEE, 2005.
- [77] G. F. Welch. History: The use of the kalman filter for human motion tracking in virtual reality. *Presence: Teleoperators and Virtual Environments*, Vol. 18, No. 1, pp. 72–91, 2009.
- [78] J. R. Wu and M. Ouhyoung. On latency compensation and its effects on head-motion trajectories in virtual environments. *The Visual Computer*, Vol. 16, No. 2, pp. 79–90, 2000.