

平成22年度

筑波大学情報学群情報科学類

卒業研究論文

題目
AiR surface : 拡張現実感を用いた仮想タッチパネル
インタフェース

主専攻 ソフトウェアサイエンス主専攻

著者 加茂 浩之

指導教員 田中二郎 三末和男 高橋伸 志築文太郎

要 旨

現在のタッチパネルインタフェース端末は「物理的制約問題」と「アクセスコントロール問題」が存在する。物理的制約問題とは、端末固有の大きさや重さに起因して持ち運びやすさが損なわれたり、利用場所が制限されるといった問題である。また、アクセスコントロール問題とは、ユーザがデバイスに表示されているコンテンツを共有するかしないかをコントロールできないといった問題である。

そこで本研究では、これらの問題を解決するために「仮想タッチパネルインタフェース」を提案する。仮想タッチパネルインタフェースとは、ヘッドマウントディスプレイを装着した状態において、実世界に拡張現実感を用いて重畳表示される仮想平面を、タッチパネルのように素手で操作できるインタフェースである。このインタフェースの特徴は、ユーザが仮想的なタッチパネルを自由に創り出し、大きさや場所の柔軟な調整や、ユーザ同士での共有ができることである。

仮想タッチパネルインタフェースの概念実証として、実世界に仮想平面を創り出し、タッチ入力を行うことができるシステム「AiR surface」を実装した。本システムでは、ユーザが両手で矩形を描く姿勢をとることによって仮想的な平面物体を創り出し、タッチ入力を行うことができる。素手による操作によって、ユーザはデバイスを手に持つ必要がないため、物理的制約問題を解決することができた。また、ヘッドマウントディスプレイに表示される内容をユーザ以外が見られないことによって、アクセスコントロール問題の一部を解決した。

目次

第1章	はじめに	1
1.1	タッチパネルインタフェース端末	1
1.2	タッチ端末の問題点	2
	物理的制約問題	2
	アクセスコントロール問題	3
1.3	本研究の目的	3
1.4	本研究のアプローチ	3
1.5	本論文の構成	4
第2章	仮想タッチパネルインタフェース	5
2.1	想定する未来環境	5
2.2	仮想タッチパネルインタフェースとは	5
2.2.1	物理的なタッチパネルが存在しない	6
2.2.2	自由に surface を創り出し、操作することが可能である	6
2.2.3	複数人数で surface を利用することが可能である	7
2.3	想定する利用シーン	7
第3章	仮想タッチパネルインタフェース プロトタイプ「AiR surface」	9
3.1	操作手段の検討	9
3.1.1	専用の操作デバイスを用いる方法	9
3.1.2	素手を用いる方法	9
3.2	surface の作成方法	10
3.3	surface のタッチ入力	12
第4章	実装	14
4.1	開発環境とシステム構成	14
4.2	画像解析部	15
4.2.1	肌色領域検出	15
4.2.2	指先検出	16
4.2.3	L 字姿勢検出	20
4.2.4	矩形姿勢検出	21

4.3	ステレオ画像処理部	22
4.3.1	指の同一視	23
4.3.2	3次元カメラ座標検出	23
4.4	座標解析部	24
4.4.1	実世界の定点を原点とした座標系への変換	24
4.4.2	タッチ判定	24
4.5	表示画像生成部	27
4.5.1	オクルージョンの再現	27
第5章	関連研究	28
5.1	仮想物体とのインタラクション	28
5.2	拡張現実感を用いたワークスペースの拡張	28
第6章	議論	29
第7章	結論	30
	謝辞	31
	参考文献	32

図目次

1.1	タッチ端末を手に持ち操作する様子	2
1.2	タッチ端末を机に置いて操作する様子	2
2.1	仮想タッチパネルインタフェースを利用しているユーザのイメージ	5
3.1	矩形姿勢	10
3.2	L字姿勢	10
3.3	矩形の創出方法	11
3.4	複数創り出された surface	11
3.5	タッチ姿勢	12
3.6	指先が surface に触れている状態	12
3.7	指先が surface を貫通した状態	12
3.8	タッチ入力	13
4.1	2つのカメラが取り付けられたヘッドマウントディスプレイ	14
4.2	ラベリング前の画像	16
4.3	ラベリング後の画像	16
4.4	最大の肌色領域	17
4.5	2番目に大きい肌色領域	17
4.6	カメラ入力画像から得られた手の領域	18
4.7	肌色検出された部分	19
4.8	得られた輪郭データ	19
4.9	指先の検出方法	19
4.10	側点の集合と端点の集合	19
4.11	L字姿勢	20
4.12	両手で矩形を描く姿勢	22
4.13	定義された矩形領域	22
4.14	左右カメラの取得画像	23
4.15	カメラ座標系からマーカ座標系への変換	25
4.16	平面と線分の衝突判定	26
4.17	平面と線分の衝突判定	26
4.18	貫通点が surface 内にあるかどうかの判定	27

4.19 オクルージョンの再現	27
---------------------------	----

第1章 はじめに

1.1 タッチパネルインタフェース端末

近年、PDA と携帯電話を融合したスマートフォンと呼ばれる携帯情報端末や、スレート端末が急速に普及している。これらの携帯情報端末は操作インタフェースと表示インタフェースを兼ね備えたタッチパネルを搭載しているものが多い。以降、タッチパネルを搭載した携帯情報端末を**タッチ端末**と呼ぶ。

タッチ端末の特徴として、入力操作が直感的でわかりやすいことや、端末を複数人で共有しやすいこと、アプリケーションが豊富であることが考えられる。これらの特徴と、それぞれの利点を以下に述べる。

入力操作が直感的でわかりやすい

タッチパネルを搭載していない携帯情報端末では、キーボードのような操作用のデバイスを用いて入力を行う。一方、タッチ端末の入力は、操作対象となる場所を直接手で触れることで行う。この入力操作は、キーボードのように、操作用のデバイスを用いて間接的に入力を行うよりも、直感的でわかりやすい。

端末を複数人数で共有しやすい

タッチ端末はタッチパネルを搭載していない携帯情報端末と比較して、広い領域を用いて操作を行うことができる。そのため、複数人数で操作を行いやすい。また、複数人数で操作を行えることによって、情報共有や共同作業に活用することができる。

利用可能なアプリケーションが豊富である

タッチ端末を用いて利用可能なアプリケーションには様々な種類が存在する。その代表的な例として、メールの送受信や、Web ページの閲覧、テキストの編集を行うアプリケーションが挙げられる。このように、タッチ端末を用いて多様な作業を行えることはユーザにとって有益である。

1.2 タッチ端末の問題点

既存のタッチ端末には物理的制約問題、アクセスコントロール問題という2つの問題点があるとする。これらの問題点について以下に述べる。

物理的制約問題

物理的制約問題とは、タッチ端末が固有の大きさと重さを持つことによって、作業効率が低下したり、タッチ端末を利用できる状況が限られてしまう問題である。

最初にタッチ端末の大きさに着目する。大きなタッチパネルをもつ端末は、ユーザに多くの情報を提示することができることや、広い作業領域を提供できるという利点がある。しかし、タッチ端末自身のサイズが大きくなるため、持ち運びやすさが損なわれる。一方、小さなタッチパネルをもつ端末は、持ち運びやすいが、ユーザに提示できる情報の量が少ない。そのため、ユーザはスクロールや画面の切り替えといった操作を頻繁に行わなければならない、作業効率が低下する可能性がある。

次にタッチ端末を利用するユーザの状況に着目する。ユーザはタッチ端末を利用するとき、片手でタッチ端末を持つか(図 1.1)、タッチ端末を机に置いて操作を行う(図 1.2)必要がある。ユーザがタッチ端末を片手に持って操作する時、ユーザはもう一方の手で操作を行うため、両手を用いた操作を行うことができない。また、ユーザがタッチ端末を机に置いて操作する時、ユーザは両手を使って操作を行うことができるが、机のない場所ではタッチ端末を利用することができない。そのため、タッチ端末を利用できる状況が限られてしまう。

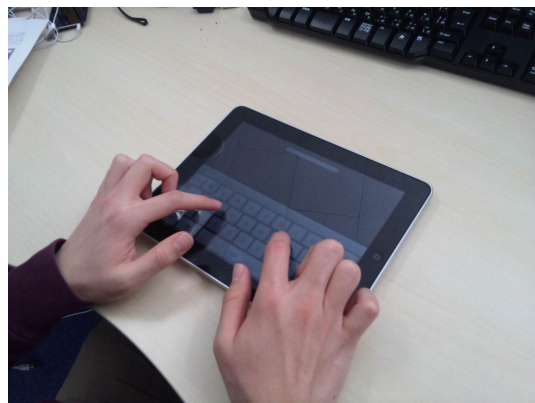
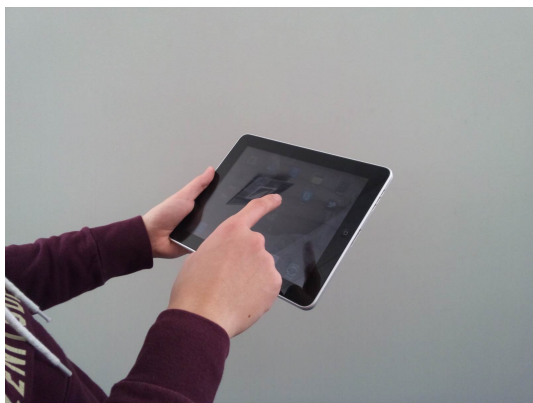


図 1.1: タッチ端末を手に持ち操作する様子 図 1.2: タッチ端末を机に置いて操作する様子

アクセスコントロール問題

アクセスコントロール問題とは、タッチパネルに表示されている内容を、ユーザ本人だけが見ることができる状態と、ユーザ本人以外も見ることができる状態を切り替えることができないという問題である。タッチ端末はタッチパネルに表示されている内容をユーザ本人以外から覗き見られることがある。例えば、ユーザがタッチ端末で機密情報を扱っている時、周囲から覗き見られることによって情報が漏えいする恐れがある。また、個人的なメールを読んでいる時に他人から内容を覗き見られることは望ましくない。この問題に対処するために、周囲からタッチパネルを見えにくくするフィルムが開発されている。しかし、これを用いても完全に第三者から見えない状態にすることは困難である。

1.3 本研究の目的

本研究では、既存のタッチ端末の問題点である、物理的制約問題とアクセスコントロール問題を解決し、かつタッチ端末の利点を継承したインタフェース「仮想タッチパネルインタフェース」を実現することを目的とする。

1.4 本研究のアプローチ

本研究では、ウェアラブルコンピューティングと拡張現実感技術を用いる。筆者は、ユーザがヘッドマウントディスプレイを着用することで、拡張現実感を用いて実世界に重畳表示される平面を、タッチパネルのように扱うことのできるインタフェース「仮想タッチパネルインタフェース」を提案する。拡張現実感を用いて実世界に重畳表示される物体は、ヘッドマウントディスプレイを用いることによって知覚することができるが、実物体は存在しない。そのため、既存のタッチ端末における物理的制約問題を解決できる。また、ヘッドマウントディスプレイに表示される映像は、着用しているユーザ本人のみが見ることができる。これにより、既存のタッチ端末のアクセスコントロール問題を解決できる。

拡張現実感 (AR:Augmented Reality)

拡張現実感とは、現実の環境から与えられる情報に、コンピュータが作り出した情報を重畳表示する技術のことをいう。概念や技術的に共通点が多いことから、拡張現実感を実現する手段としてウェアラブルコンピューティングが用いられることが多い。拡張現実感についての研究は近年盛んに行われてきている [1] [2] [3]。拡張現実感の実世界の物体に関連する付加的な情報を付与する場合などに用いられている [4][5][6]。

ウェアラブルコンピューティング

ウェアラブルコンピューティングとは、情報端末を体に装着して利用するスタイルを指す[7]。装着する情報機器としてヘッドマウントディスプレイが代表的である。情報機器を体に身に着けることで、次のような利点がある。

- 常にコンピュータやディスプレイが稼働状態になるため、ユーザが意図した時にいつでもコンピュータを利用できる。
- ユーザの生活に密着した、ユーザの日常生活をサポートするアプリケーションを構築することができる。

1.5 本論文の構成

本章では、既存のタッチ端末についての現状と問題点を挙げ、研究の目的と問題を解決するためのアプローチについて述べた。続いて第2章では本研究で提案する「仮想タッチパネルインタフェース」のコンセプトについて述べる。第3章では仮想タッチパネルインタフェースのプロトタイプシステムである「AiR surface」について述べ、第4章では実装について詳細に述べる。第5章で関連研究について述べ、第6章で試用から得た知見と今後の発展について議論する。最後に、第7章で結論を述べる。

第2章 仮想タッチパネルインタフェース

2.1 想定する未来環境

本研究は、現代人が眼鏡を掛けることと同様の感覚でヘッドマウントディスプレイを着用するような、ウェアラブルコンピューティングが一般化した近未来環境を想定している。既存のヘッドマウントディスプレイは、大きく重いことや、解像度が低いこと、価格が高価であること、着用している姿が社会的に受け入れられがたいことといった問題がある。そのため、軍事分野では実用化されている例があるものの、一般には普及していない。しかし、近年ヘッドマウントディスプレイの技術は進歩しており、小型化と高性能化が進んでいる。筆者は、今後ヘッドマウントディスプレイの小型化と高性能化が更に進み、デザイン性も向上することによって、一般に普及すると考える。

2.2 仮想タッチパネルインタフェースとは

仮想タッチパネルインタフェースとは、拡張現実感を用いて実世界に重畳表示する仮想的なタッチパネル(以降、surfaceと呼ぶ。)を利用するインタフェースである。仮想タッチパネルインタフェースを利用しているユーザのイメージを図2.1に示す。仮想タッチパネルインタフェースの特徴とそれぞれの利点を以下に述べる。



図 2.1: 仮想タッチパネルインタフェースを利用しているユーザのイメージ

2.2.1 物理的なタッチパネルが存在しない

ユーザはヘッドマウントディスプレイを着用することによって、surface を拡張現実感を用いて実世界に重畳表示して見ることができる。surface は仮想物体であり実物体ではないため、surface は手に持つことなく空気中に浮遊した状態で存在することができる。そのため、ユーザはタッチ端末を手に持つことなく両手で操作可能であり、タッチ端末を置くための机が無い場所でも利用可能である。したがって、ユーザはタッチ端末を利用することが困難な場所においても surface の利用が可能であるという利点がある。

2.2.2 自由に surface を創り出し、操作することが可能である

ユーザは自由な大きさや数の surface を創り出すことができる。また、創り出した surface はタッチ入力が行えるほか、リサイズや移動といった柔軟な調整が可能である。

自由な大きさの surface を創り出すことが可能である

必要な surface の大きさは、ユーザが行いたい作業の種類によって異なる。そこで、ユーザは行う作業に応じた大きさの surface を作成することができる。例えば、ユーザが文章を入力するシーンを想定すると、電話番号程度の短い文字列の場合、小さな surface が適切である。一方、プレゼンテーションのスライド編集など、扱う情報が多い場合は、大きな surface が適切である。

自由な数の surface を創り出すことが可能である

ユーザは複数の作業を並行して行うとき、1つの surface では使用するアプリケーションを切り替えながら並行して作業を行う必要があり、効率的に作業することができない。例えば、ユーザが Web ページを参照しながら文書を編集するシーンを想定すると、ユーザが1つの surface で作業を行う場合、ブラウザとエディタを何度も行き来しなければならない。しかし、ユーザが必要な数の surface を作成した場合、複数のアプリケーションを切り替える必要が無く、効率的に作業を進めることができる。

タッチ入力が可能である

ユーザは surface をタッチすることによって、計算機のマウスのような専用の入力デバイスを用いた間接的な入力操作ではなく、直観的な入力を行うことができる。

surface の柔軟な調整が可能である

ユーザが作成した surface は作成した後に、移動、リサイズ、削除といった操作を行うことができる。ユーザはこれらの操作によって、常に最適な状態の surface を利用することができる。

2.2.3 複数人数で surface を利用することが可能である

surface は仮想的に実空間上に存在しているため、surface に表示されているコンテンツを複数ユーザで共有することができる。また、ユーザは作成した surface のアクセスコントロールを行うことができる。

surface に表示された内容を共有することができる

surface を用いて友人同士で写っている写真を閲覧している時など、周囲の人々と surface に表示されている内容を共有したい場合がある。このような時、大きな surface を複数人数で利用することで、写真の共有を容易に行うことができる。また、surface に情報を表示させた状態を維持することによって、掲示板のように利用することも可能である。

surface のアクセスコントロールが可能である

surface を用いてプライベートのメールを作成している時や、個人的な趣味の動画を鑑賞している時など、surface に表示されている内容を他人に見られたくない場合がある。このような時、surface を本人だけが見える状態に設定することで、他人に内容を見られることなく surface を利用することができる。

2.3 想定する利用シーン

物理的なタッチパネルが存在しない利点を活かした利用シーン

Yさんは、妹のUさんに頼ってばかりいて家事がとても苦手である。Yさん達の両親が海外旅行中で不在の時、Uさんが高熱に見舞われ寝込んでしまった。そこで、Yさんは妹の体調を気遣い体に優しい料理を作ることにした。キッチンに立ったYさんは surface を作り、レシピを表示させた。レシピを見ながら料理を進めていくと、Yさんは砂糖と醤油の場所が分からないことに気づき、新たな surface を作成し、Uさんにチャットで「砂糖と醤油はどこだっけ？」と尋ねた。砂糖と醤油の場所を知ることができたYさんはレシピを見ながら慎重に進めて行ったことで、無事にUさんにおいしい料理を食べさせることができた。

この例では、キッチンは調理器具や食材が置かれており、タッチ端末を置くスペースが無いと想定される。このような状況においてYさんは、物理的なタッチパネルが存在しないことの利点を活かして surface を利用している。

自由に surface を創り出し、共有できる利点を活かした利用シーン

軽音楽部に所属する Y さんは自分のギターを購入するにあたり、どのようなモデルが良いか、部員の M さんに相談した。Y さんと M さんはそれぞれの surface で Web にアクセスし、ギターについて調べていると、Y さんがあるギターのデザインに非常に心を惹かた。そして Y さんはギターの画像が表示された surface を拡大し、2 人が見やすい場所に移動させた。M さんはそのギターの画像を見て、「ネックがもう少し細い方が初心者に向いているよ。」とアドバイスをした。しかし、Y さんは拡大したギターの画像に夢中になり、すぐに別の surface で EC サイトを開きそのギターを購入した。

この例で Y さんは仮想タッチパネルインタフェースの、自由な数の surface を作成することができることの利点を活かして、2 枚の surface を用いたギターの画像の閲覧と、EC サイトにおける買い物を行っている。また、surface の内容を共有できる利点を活かし、ギターの画像を M さんと共有している。

アクセスコントロールが可能であることを活かした利用シーン

恥ずかしがり屋の作詞家である M さんは、所属するバンドが次回のライブで演奏する楽曲の歌詞を考えていた。surface でテキストエディタを開き、思い浮かんだアイデアを書きこんでいるところに、バンドメンバーの A さんが来た。未完成の歌詞を見られたくないと感じた M さんは、surface を自分だけに見える状態にした。その後、他のバンドメンバーも来たが、M さんが歌詞を書いていることには気づかず、各自で自分の楽器の練習をしていた。M さんは他の人に作業を妨げられることなく歌詞を書きあげることができた。

この例で、M さんはアクセスコントロールが可能であることの利点を活かして、他のバンドメンバーに surface の内容を見られることなく作業を行っている。

第3章 仮想タッチパネルインタフェース プロトタイプ「AiR surface」

前章で述べた仮想タッチパネルインタフェースのコンセプトのうち、「物理的なタッチパネルが存在しない」という特徴と、「自由に surface を創り出し操作することが可能である」という特徴のうちの、surface を創り出し、タッチ入力を行う部分を実現したプロトタイプシステム「AiR surface」を開発した。また、タッチ端末を用いて利用可能なアプリケーションの一例として、本システムを用いて画像の表示と切り替えを行うことができるアプリケーションを開発した。

3.1 操作手段の検討

仮想的な物体を操作する手法については、多くの研究がされている。アプローチの具体例として、操作用に専用のデバイスを用いる方法や、素手でインタラクションを行うといったアプローチがある。それぞれのアプローチについて、操作の自由度と、触覚フィードバック、デバイスの持ち運びやすさに着目して述べる。

3.1.1 専用の操作デバイスを用いる方法

操作用のデバイスを用いる方法は、操作の自由度をあえて抑えることによって操作を単純化する。ユーザの行動の自由度は、デバイスの形状から多くの制約を受けるが、システム側は限られた自由度に対してのみフィードバックを実現すれば良いという利点がある。また、ユーザは専用の操作デバイスを常に持ち歩かなければならない。具体例として、棒状のデバイスを用いて仮想物体を指すことによって操作を行う方法 [8] などがある。

3.1.2 素手を用いる方法

素手を用いる方法は、ユーザがデバイスを手に持つか、装着する必要が無い、手の自由度に一切の制約を受けないという利点がある。また、ユーザは利用に備えてデバイスを持ち歩く必要が無い。例えば、素手を用いてマウスにおけるクリックや選択操作を行う方法 [9] がある。筆者は、素手を用いる方法は操作用デバイスが必要無く、物理的制約問題を解決するアプローチとして有効であると考えた。また、ユーザの手を用いた操作は、専用のデバイ

スの操作方法を学習する必要が無く、人間の知識や経験に基づいた自然な操作であり、より操作をスムーズに行うことができると考えた。したがって本研究では、素手を用いて surface とインタラクションを行う方法を採用する。

3.2 surface の作成方法

前節で述べた通り、本システムでは素手による surface の操作を採用する。そこで、素手による surface の作成方法の検討を行った。筆者は、surface を作成する方法は、以下の3つの要件を満たしていることが望ましいと考えた。

- surface となる平面の領域を定義することができる
- 操作を素早く行うことができる
- インタラクションと自然な対応付けであり理解しやすい

これらの要件を満たすものとして、淵らが範囲選択に用いた、両手で矩形を描く手の姿勢 [10] を採用した。この姿勢は写真を撮る動作などに用いられ、平面の領域を選択する手法として自然な対応付けであり、理解しやすいと考えた。以降、両手で矩形を描く手の姿勢を「矩形姿勢」と呼び(図 3.1)、矩形姿勢を作るそれぞれの手がなす、親指と人差し指を広げて L 字型を描く手の姿勢を「L 字姿勢」と呼ぶ(図 3.2)。



図 3.1: 矩形姿勢



図 3.2: L 字姿勢

ユーザが矩形姿勢を作り、surface を作成したい平面領域で一定時間静止すると、その場所に surface が作成される(図 3.3)。ユーザは手を動かして矩形姿勢が囲む領域の大きさを変えことによって、自由な大きさの surface を作成することが可能である。また、場所を変えて矩形姿勢を作ることによって、自由な数の surface を創り出すことが可能である(図 3.4)。

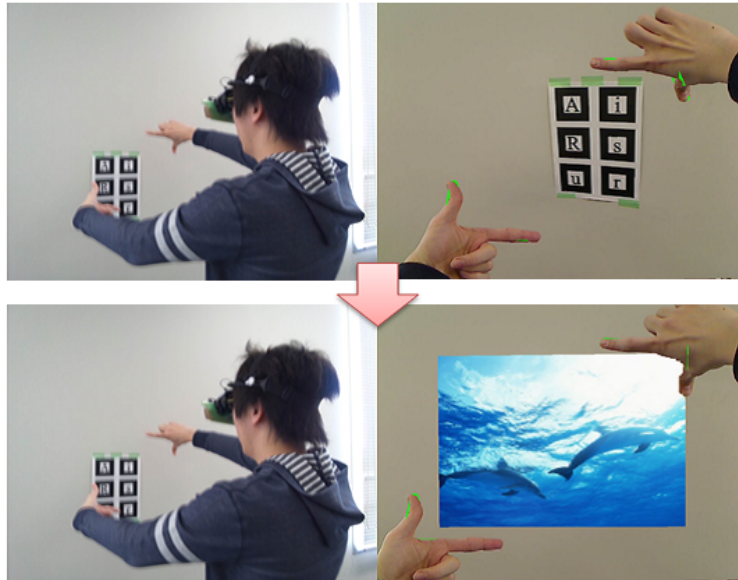


図 3.3: 矩形の創出方法



図 3.4: 複数創り出された surface

3.3 surface のタッチ入力

ユーザが既存のタッチ端末を使用する時，タッチ入力を行う場合は，人差し指を立てて画面に触れる場合が多い．このように，人差し指を立てている手の姿勢 (図 3.5) を「タッチ姿勢」と呼ぶ．surface のタッチ入力は，タッチ姿勢の状態で行う．

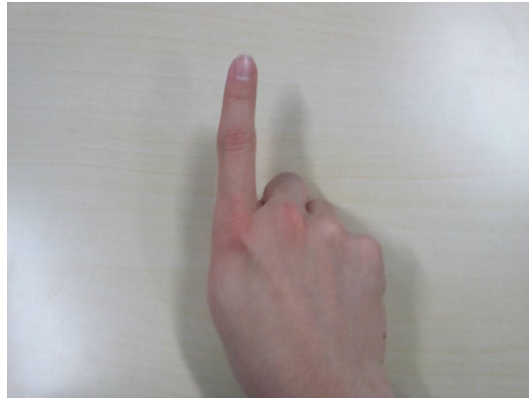


図 3.5: タッチ姿勢

既存のタッチ端末はユーザがタッチ入力を行うと，ユーザの指先はタッチパネルからの反発を受け，タッチパネルを突き抜けることができない．しかし，surface は実物体でないため，ユーザの指先は surface を突き抜けてしまう．そこで，surface のタッチ入力は，指先が surface に接触して静止している状態 (図 3.6) と，指先が surface を突き抜けた状態 (図 3.7) の両方を用いることにした．前者の場合，ユーザの指先が触れている surface 上の点を入力点とする．後者の場合，突き抜けた瞬間に指先が surface と交わった点を入力点とする．タッチ操作を行っているユーザの様子を (図 3.8) に示す．



図 3.6: 指先が surface に触れている状態

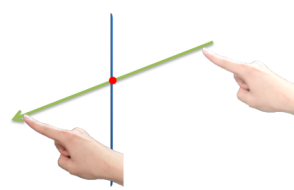


図 3.7: 指先が surface を貫通した状態

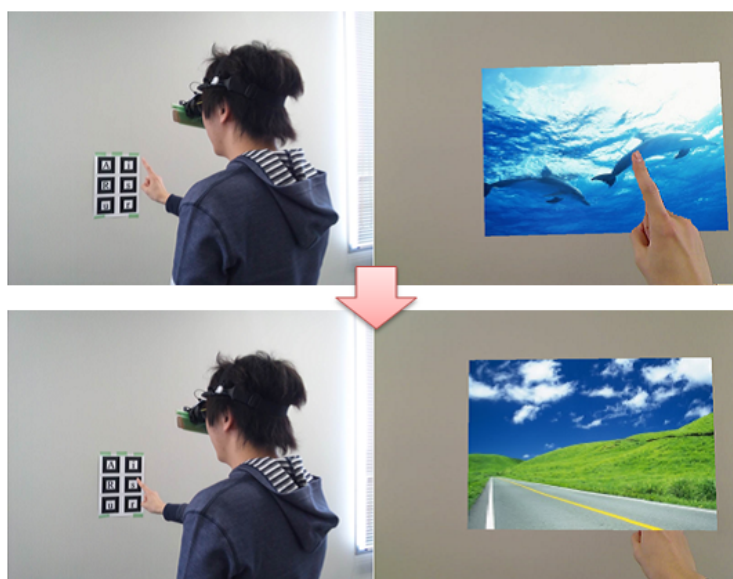


図 3.8: タッチ入力

第4章 実装

4.1 開発環境とシステム構成

開発言語はC++、開発環境には Visual Studio 2008 を使用した。OS は Windows7、CPU は Intel Core 2 Duo 2.66GHz である。ヘッドマウントディスプレイには eMagin 社¹ の Z800 3DVISOR を使用した。また、このヘッドマウントディスプレイには2つの USB カメラを取り付けた (図 4.1)。この USB カメラには Logicool 社² の 2-MP Portable Webcam C905m を用いた。



図 4.1: 2つのカメラが取り付けられたヘッドマウントディスプレイ

実装には3つのライブラリを使用した。画像解析には OpenCV³を用いた。実世界との対応付けのために AR ToolKit⁴を用いた。USB カメラから得られた画像に surface を重畳表示するためには OpenGL⁵を用いた。本システムは以下の4つの処理部からなる。

- 画像解析部：カメラ入力画像から、指先とタッチ姿勢、L字姿勢、矩形姿勢を検出
- ステレオ画像処理部：左右の解析データを基に3次元データを作成
- 座標解析部：3次元データの実世界への対応付け

¹eMagin 社 (<http://www.emagin.com/>)

²Logicool 社 (<http://www.emagin.com/>)

³OpenCV(<http://opencv.jp/>)

⁴AR ToolKit(<http://www.hitl.washington.edu/artoolkit/>)

⁵OpenGL(<http://www.opengl.org/>)

- 表示画像生成部：ヘッドマウントディスプレイに表示する画像を作成

4.2 画像解析部

画像解析部では、以下の手順でカメラ入力画像から矩形姿勢の検出までを行う。この処理を、左右カメラからの入力画像それぞれに対して行う。

1. 肌色領域検出：カメラ入力画像から肌色領域を検出
2. 指先検出：検出した肌色領域を基に指先の検出とタッチ姿勢を検出
3. L字姿勢検出：検出した指先を基にL字姿勢を検出
4. 矩形姿勢検出：検出したL字姿勢を基に矩形姿勢を検出

4.2.1 肌色領域検出

最初に、カメラ入力画像中の肌色領域を検出し、肌色領域を表す2値化画像を作成する。画像中の肌色領域を認識する研究は、盛んに行われてきている。その代表的な手法として、RGB表色系で表されるカメラ入力画像を別の表色系に変換した後、閾値を定めることによって肌色部分を抽出する手法がある[11][12][13]。RGB表色系から変換される表色系として、色を持つ明るさの情報と色見の情報を分離することができるHSV表色系やYUV表色系が用いられる。そこで、本研究ではカメラ入力画像から得られたRGB表色系の画像を、YUV表色系に変換して肌色領域の検出を行った。RGB表色系からYUV表色系に変換する式を式4.1、式4.2、式4.3に、肌色と認識する色の閾値を式4.4に示す。以下に書かれている $Y'U'V'$ は淵らが用いた擬似的なYUV空間[10]となっており、Yに16、UとVにはそれぞれ128を加えることで、一般的なYUV空間に変換できる。

$$Y' = (256 \times R + 504 \times G + 98 \times B) / 1000 \quad (4.1)$$

$$U' = (-148 \times R - 291 \times G + 439 \times B) / 1000 \quad (4.2)$$

$$V' = (439 \times R - 368 \times G - 71 \times B) / 1000 \quad (4.3)$$

$$48 < Y' < 224 \quad -34 < U' < -3 \quad 12 < V' < 127 \quad (4.4)$$

閾値によって2値化画像を作成すると、手以外の微小な領域が誤認識されていることがある。この微小な領域を除去し、手の領域だけを抽出するためにラベリングを行った。ラベリングには井村が提供するラベリングクラス⁶を用いた。ラベリングクラスは肌色として認識された領域(図4.2)について、連続している領域を面積が大きい順番に数値を割り当てる(図4.3)。また、このラベリングクラスを用いることで、それぞれの領域の面積と、重心の座標を得ることができる。

⁶ラベリングクラス (<http://oshiro.bpe.es.osaka-u.ac.jp/people/staff/imura/products/labeling>)

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	1	0
0	0	1	1	0	0	0	0	1	0
0	1	1	1	1	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0
1	1	1	1	1	0	0	0	1	1
1	1	1	1	0	0	0	1	1	1
1	1	1	1	0	0	0	1	1	1
1	1	1	0	0	0	0	1	1	1

図 4.2: ラベリング前の画像

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	4	0	3	0
0	0	1	1	0	0	0	0	3	0
0	1	1	1	1	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0
1	1	1	1	1	0	0	0	2	2
1	1	1	1	0	0	0	2	2	2
1	1	1	1	0	0	0	2	2	2
1	1	1	0	0	0	0	2	2	2

図 4.3: ラベリング後の画像

ラベリングによって得られた各領域の面積を用いて、誤認識された領域のフィルタリングを行い、手の領域の抽出を行う。肌色と認識された領域のうち、正しく認識された手の領域が最も大きく、誤認識された部分の面積は微小であると仮定し、連続している領域のうち、面積が最も大きい領域と、2番目に大きい領域を手の領域として認識した。手の領域として2つの領域を抽出するのは、本システムの操作には両手を用いるためである。図 4.2 の肌色領域から、手の領域として認識した2つの部分を図 4.4, 図 4.5 に示す。カメラ入力画像から、これらの処理を経て生成した手の領域の画像を図 4.6 に示す。

4.2.2 指先検出

肌色検出によって得られた2値化画像をもとに、指先の検出を行う。指の形状の特徴を利用し、肌色領域の凸型の形状をしている場所を指として認識する。最初に、得られた2値化画像の肌色部分の輪郭を検出する(図 4.7)。輪郭の検出には OpenCV を利用した。次に、図 4.8 右の赤い線が示す輪郭に沿って、一定の間隔を開けた3点のなす角度を計算する。図 4.9 のように、走査を開始する点を P_1 とし、一定の間隔 d はなれた点を P_2 、 P_2 からさらに d はなれた点を P_3 とし、 P_1, P_2, P_3 のなす角を θ とする。 θ の角度が30度以下のとき、凹凸型の形状をしていると認識する。ベクトル P_2P_1 とベクトル P_2P_3 の内積と外積から得られる $\sin \theta$ (式 4.5) と $\cos \theta$ (式 4.6) について閾値(式 4.7)を設けることで、凸点のみを抽出することができる。このときの、 P_1 と P_3 を側点、 P_2 を端点と呼ぶ。輪郭に沿って順番に θ を計算していくと、条件を満たす側点と端点は連続的に続き、2つの側点の集合と、端点の集合ができる(図 4.10)。

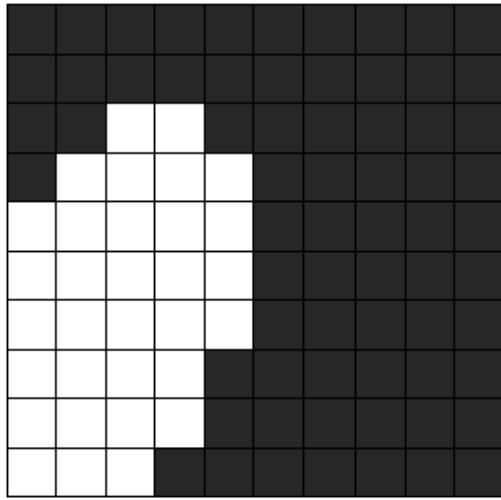


図 4.4: 最大の肌色領域

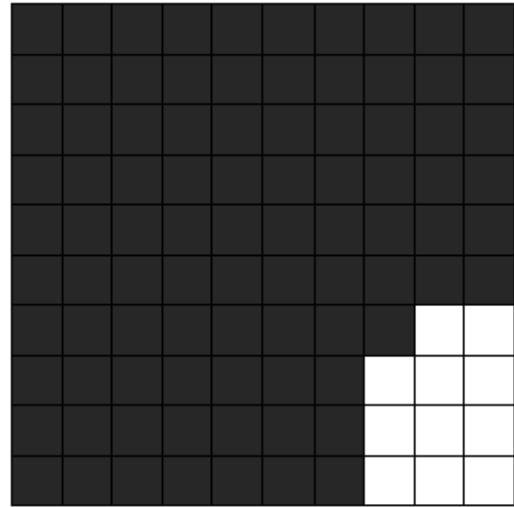


図 4.5: 2 番目に大きい肌色領域

それぞれの集合に対し，先述したラベリングを用いて重心を求めることで，1か所の凸型形状に対する端点と，2つの側点をそれぞれ1点に定める．この3点を用いてFingerクラス(表4.1)のインスタンスを作成する．また，作成されたFingerクラスのインスタンスが1つであった場合，タッチ姿勢であると認識する．

表 4.1: Finger クラスのメンバ

属性名	説明
tip	端点の座標
side1	側点の座標
side2	側点の座標

$$\sin \theta = \frac{P_2 P_1 \times P_2 P_3}{|P_2 P_1| |P_2 P_3|} \quad (4.5)$$

$$\cos \theta = \frac{P_2 P_1 \cdot P_2 P_3}{|P_2 P_1| |P_2 P_3|} \quad (4.6)$$

$$0 \leq \sin \theta \leq \frac{1}{2} \quad \frac{\sqrt{3}}{2} \leq \cos \theta \leq 1 \quad (4.7)$$

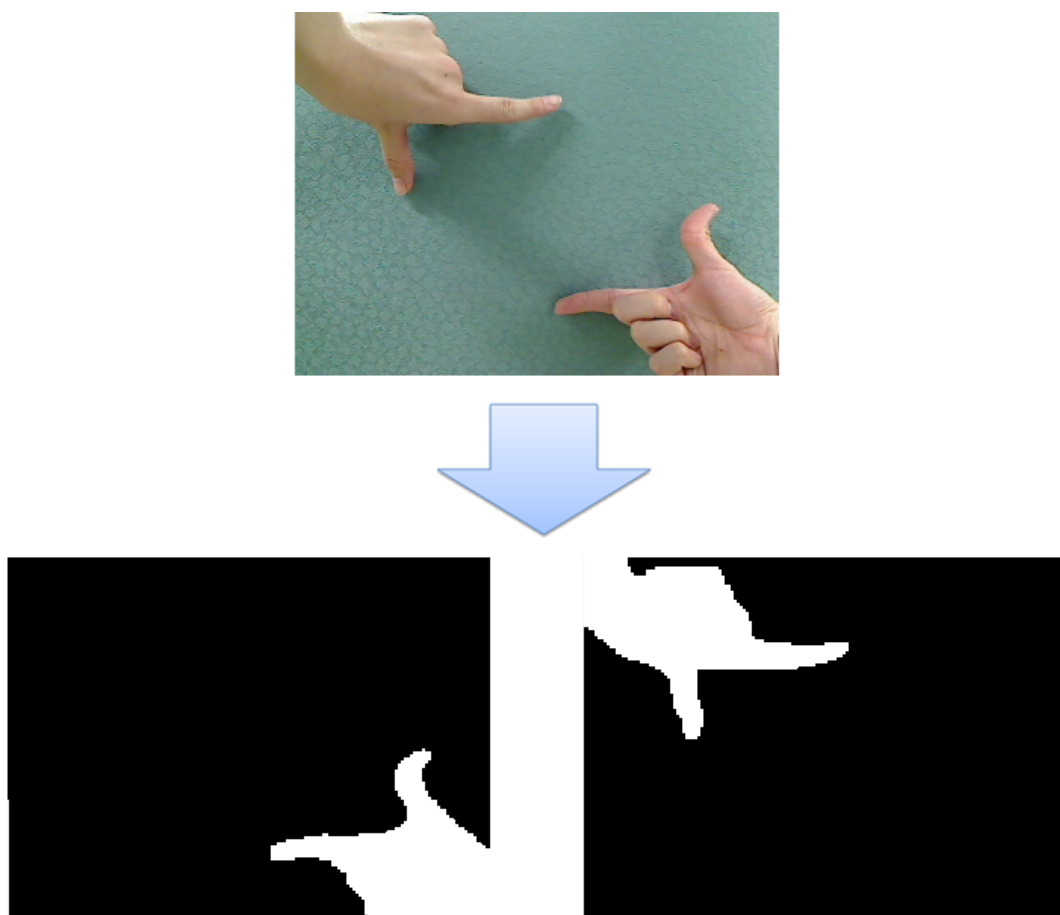


図 4.6: カメラ入力画像から得られた手の領域

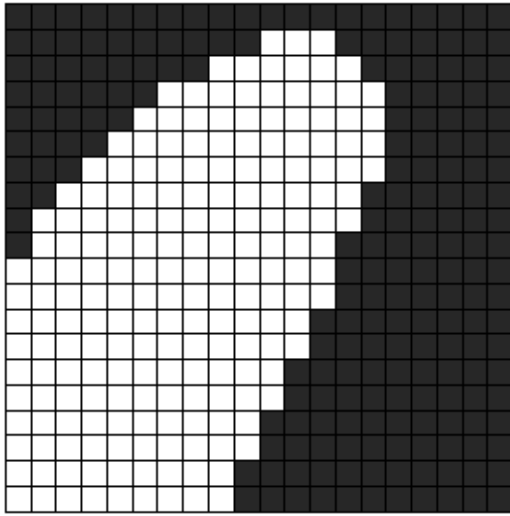


図 4.7: 肌色検出された部分

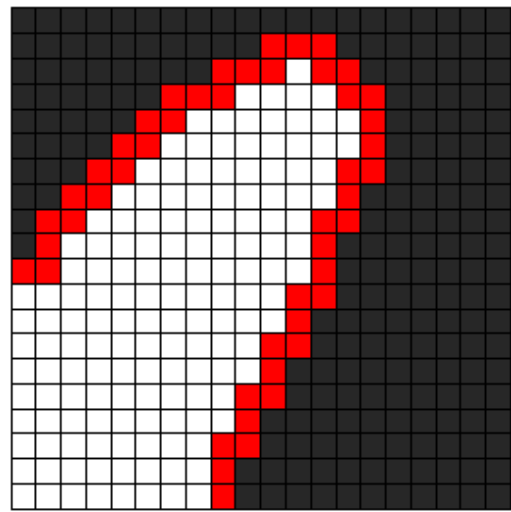


図 4.8: 得られた輪郭データ

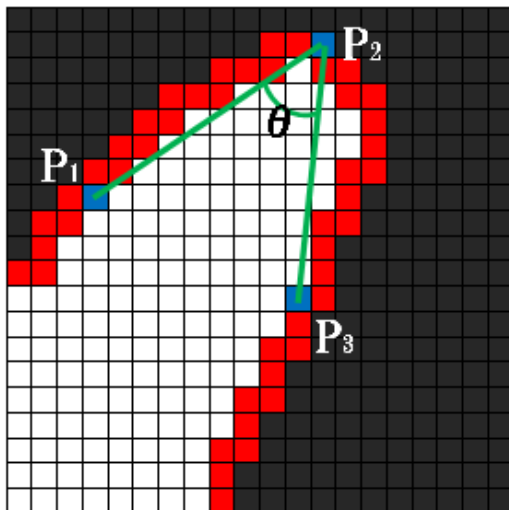


図 4.9: 指先の検出方法

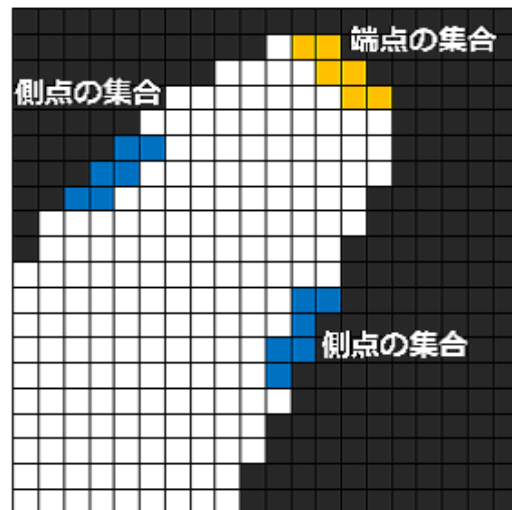


図 4.10: 側点の集合と端点の集合

4.2.3 L 字姿勢検出

作成された Finger インスタンス群から，L 字姿勢の検出を行う．最初に，得られた Finger インスタンス群から，全ての組み合わせが探索されるように，組み合わせの候補を作成し，それぞれの組み合わせについて，L 字姿勢であるかの判定を行う．以下，図 4.11 を例に挙げる．図中の親指に当たる Finger インスタンスを Finger A，人差し指に当たる Finger インスタンスを Finger B とする．最初に，Finger A と Finger B について，side1 と side2 の中点 C_A ， C_B を求める．また Finger A，Finger B それぞれの tip の点を T_A ， T_B とすると，交点の座標 V は指の方向 d_A ， d_B (式 4.8) を用いて式 4.9 で表される．この方法で求められた点 V は，親指の付け根付近となる．

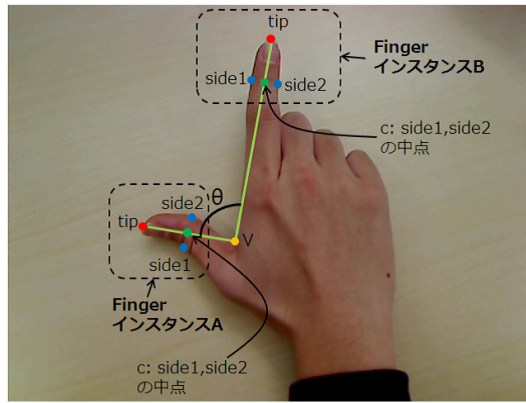


図 4.11: L 字姿勢

$$d_A = \frac{T_{Ay} - C_{Ay}}{T_{Ax} - C_{Ax}} \quad d_B = \frac{T_{By} - C_{By}}{T_{Bx} - C_{Bx}} \quad (4.8)$$

$$V_x = -\frac{(T_{Ay} - d_A T_{Ax}) + (T_{By} - d_B T_{Bx})}{d_A - d_B} \quad V_y = d_A V_x + d_B T_{Ax} \quad (4.9)$$

次に，ベクトル VT_A とベクトル VT_B がなす角 θ の余弦(式 4.10)を計算し，これが式 4.11 を満たすものを候補とする．そして，候補における $|VT_A|$ と $|VT_B|$ を指の長さとし，2本の指の長さの比を計算する．指の長さは人により個人差があるため，指の長さについての傾向を調べるために，大学生4名(うち女性1名)の指の長さを計測した．その結果を表 4.2 に示す．この結果から，親指の長さ÷人差し指の長さはおよそ 0.45 から 0.55 の間になることがわかった．この値を閾値として，式 4.12 を用いてフィルタリングをおこなう．条件を満たしたものについて，親指の座標と人差し指の座標，親指と人差し指の交点の座標から成る L-posture クラス(表 4.3)のインスタンスを作成する．

$$\cos \theta = \frac{VT_A \cdot VT_B}{|VT_A| |VT_B|} \quad (4.10)$$

$$\frac{1}{2} \leq \cos \theta \leq 1 \quad (4.11)$$

$$0.45 \leq \frac{|VT_A|}{|VT_B|} \leq 0.55 \quad (4.12)$$

表 4.2: 人差し指と親指の長さとその比率

	A	B	C	D
親指の長さ	60mm	70mm	55mm	60mm
人差し指の長さ	130mm	130mm	120mm	120mm
親指の長さ / 人差し指の長さ	0.46	0.54	0.46	0.5

表 4.3: L-posture クラスのメンバ

属性名	説明
thumb	親指の座標
index	人差し指の座標
vertex	交点の座標

4.2.4 矩形姿勢検出

最後に、求められた L-posture インスタンス群を用いて矩形姿勢の検出を行う。2つの手が矩形つくるとき、左右の人差し指は平行になる。この特徴を利用して矩形の検出を行う。図 4.12 を例に挙げる。L-posture インスタンス A と L-Posture インスタンス B のそれぞれの人差し指の座標を、 I_A , I_B 、人差し指と親指の交点の座標を V_A , V_B とする。ベクトル $V_A I_A$ とベクトル $V_B I_B$ がなす角 θ を式 4.13 求め、式 4.14 を満たす 2 つの L 字姿勢を矩形姿勢として認識する。矩形の中心座標は、 V_A と V_B の中点の座標とし、矩形の幅は $|V_{Bx} - V_{Ax}|$ 、矩形の高さは $|V_{By} - V_{Ay}|$ とし、Rectangle クラス (表 4.4) のインスタンスを作成する。定義された矩形領域は図 4.13 のようになる。

$$\theta = \cos^{-1} \left\{ \frac{V_A I_A \cdot V_B I_B}{|V_A I_A| |V_B I_B|} \right\} \quad (4.13)$$

$$\frac{7}{8}\pi \leq \theta \leq \pi \quad (4.14)$$

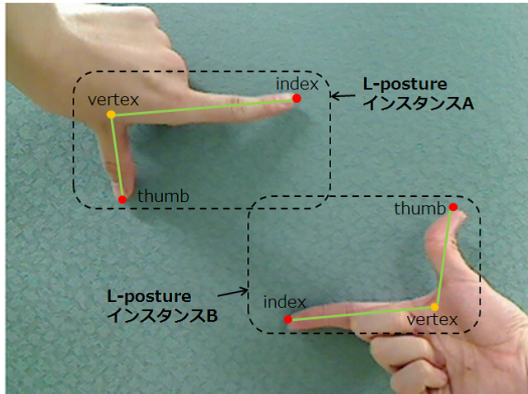


図 4.12: 両手で矩形を描く姿勢

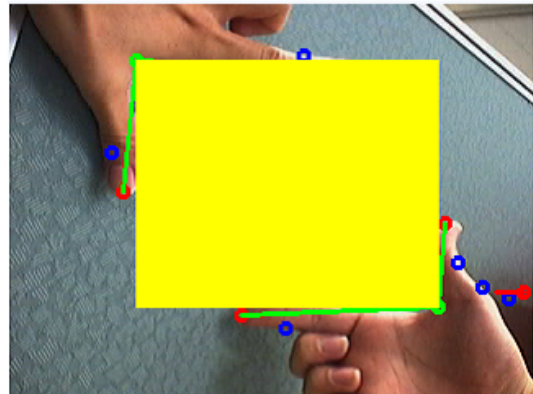


図 4.13: 定義された矩形領域

表 4.4: Rectangle クラスのメンバ

属性名	説明
center	矩形の中心座標
width	矩形の幅
height	矩形の高さ

4.3 ステレオ画像処理部

画像解析部によって、左右カメラの入力画像における指先、タッチ姿勢、L字姿勢、矩形姿勢についてのデータが得られた。しかし、データ中の座標は、すべてスクリーンの中心を原点とした座標系(以下、「スクリーン座標系」と呼ぶ。)であり、2次元である。

座標解析部では2つの2次元データを整理し、タッチ姿勢と矩形姿勢についての3次元データを作成する。本研究では3次元データを求める手法として、ステレオ法を用いた。ステレオ法とは、三角測量の原理を応用し、測定物体を異なる位置から撮影して得た複数の画像から、測定物体の3次元位置を求める方法である。この方法を用いることによって、左右のカメラの中心を原点とする座標系(以下、「カメラ座標系」と呼ぶ。)における3次元データを求めることができる。3次元データを求める手順を以下に示す。

1. 指の同一視：2つの2次元データから同一の指先を検出
2. 3次元カメラ座標検出：同一であると検出された指先の3次元カメラ座標を求める

4.3.1 指の同一視

左カメラ入力画像と右カメラ入力画像のそれぞれで検出された Finger インスタンスは、同一の指であることを認識する必要がある．左右のカメラ入力画像に、画像解析部によって得られた Finger クラスのインスタンスを表示したものを図 4.14 に示す．図 4.14 の入力画像を例とすると、画像解析部における処理の結果、それぞれのカメラ映像から 2 つの Finger インスタンスが得られた．指の角度 θ は、Finger インスタンスの 2 つの側点の中点を C 、端点を T とすると式 4.15 によって求めることができる．左右のカメラ入力画像から得られた Finger インスタンスの指の角度 θ_L, θ_R を求め、 $|\theta_L - \theta_R|$ の値が最小であり、かつ式 4.16 を満たす組み合わせを、同一の指を示す Finger インスタンスであると認識する．図 4.14 の場合、A と A'、B と B' がそれぞれ同一の指を示す Finger インスタンスである．

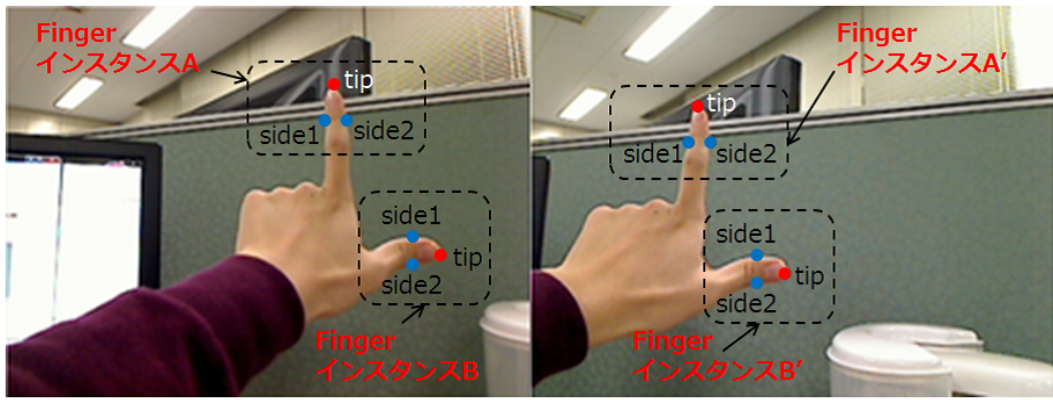


図 4.14: 左右カメラの取得画像

$$\theta = \tan^{-1} \left\{ \frac{T_y - C_y}{T_x - C_x} \right\} \quad (4.15)$$

$$|\theta_L - \theta_R| \leq \frac{\pi}{16} \quad (4.16)$$

4.3.2 3次元カメラ座標検出

同一の指であると認識された指先の座標から、ステレオ法を用いてカメラ座標系における指先の 3 次元座標を求める．ステレオ法を用いて 3 次元座標を求める方法は文献 [5] を参考にした．スクリーン座標系における指先の位置をそれぞれ (x_l, y_l) , (x_r, y_r) とする．また、カメラの焦点距離を f 、左右 2 つのカメラ間の距離を D とする．これらを用いると、左右のカメラの中点を原点とした指先の 3 次元座標 (X, Y, Z) は式 4.17 で求めることができる．

$$Z = \frac{B \cdot f}{x_l - x_r} \quad X = Z \cdot \frac{x_l + x_r}{2} \quad Y = Z \cdot \frac{y_l + y_r}{2} \quad (4.17)$$

4.4 座標解析部

座標解析部では，ステレオ画像処理部によって求めた3次元データを，実世界上の定点を原点とした座標系に変換し，変換した座標をもとにタッチ入力判定を行う。

4.4.1 実世界の定点を原点とした座標系への変換

ヘッドマウントディスプレイに取り付けられたカメラはユーザの動きに追従するため，カメラ座標系のままでは surface もユーザに追従してしまう．拡張現実感技術において，実世界への位置合わせを行う方法について様々な研究がされている [14][15]．例えば，RFID を用いる方法 [16] や，赤外線カメラを用いた方法 [17]，カメラと3次元センサを組み合わせた方法 [18] などがある．本研究では，実世界への位置合わせを，AR ToolKit[19][20] を用いることによって行った．

AR ToolKit は黒枠で囲まれた矩形 (以下，AR マーカと呼び，AR マーカの中心を原点とした座標系をマーカ座標系と呼ぶ．) を認識し，カメラ座標系における AR マーカの位置・姿勢といった情報を得ることができる．カメラ座標系における座標を，マーカ座標系における座標に変換することで surface の位置を実世界の位置と対応づけることができる (図 4.15)．AR ToolKit を用いることによって，マーカの中心位置 (X_m, Y_m, Z_m) をカメラ座標系における位置 (X_c, Y_c, Z_c) に変換する座標変換行列 (式 4.18) を得ることができる．式 (4.18) の座標変換行列における， $r_1 \sim r_9$ は回転成分を表しており， t_x, t_y, t_z はそれぞれ並進成分を表している．式 (4.18) を式 (4.19) のように簡略化すると，カメラ座標系における座標をマーカ座標系に変換する数式は，式 (4.20) のようになる．矩形姿勢についての3次元データを，この方法でマーカ座標系に変換し，変換先の座標に surface を作成する．

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & r_3 & t_x \\ r_4 & r_5 & r_6 & t_y \\ r_7 & r_8 & r_9 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} \quad (4.18)$$

$$A = T \cdot B \quad (4.19)$$

$$T^{-1} \cdot A = B \quad (4.20)$$

4.4.2 タッチ判定

surface のマーカ座標と，タッチ姿勢の指先座標をもとに，指先と surface のタッチ判定を行う．指先が surface に触れている状態 (図 3.6) の判定は，現在のフレームの指先座標から，手

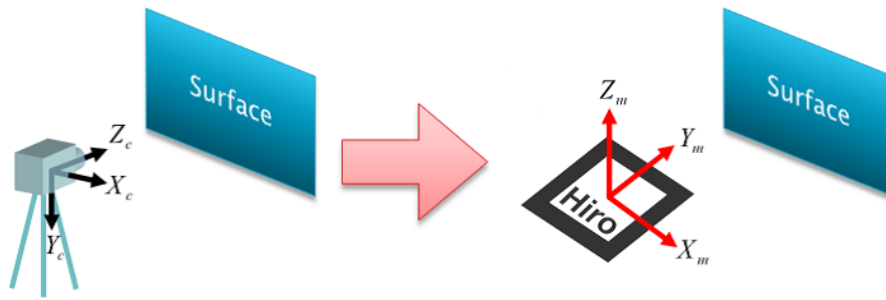


図 4.15: カメラ座標系からマーカ座標系への変換

前に一定距離 d 離れた点と、奥に d 離れた点を結ぶ線分と、surface の衝突判定によって行う。また、指先が surface を貫通した状態 (図 3.7) の判定は、現在のフレームの指先座標と、1 フレーム前の指先の座標を結ぶ線分を作成し、この線分と平面の衝突判定によって行う。線分と平面の衝突判定については〇×つくろーどっとコム⁷を参考にした。衝突判定は以下の3つの手順で行う。

1. 平面の法線ベクトルと線分の始点・終点から、平面を貫通しているかどうか判定する
2. 平面と線分の始点・終点の距離から内分比を求め、貫通点の座標を求める
3. 貫通点が平面上にあるか判定する

最初に、法線ベクトルと線分の始点・終点から、平面を貫通しているかどうかを判定する。ここでの平面とは、領域の制約はなく、無限大に広がる平面のことである。図 4.16 において、 P_0 は平面上の任意点の座標、 P_1 は前フレームの指先の座標を、 P_2 は現在のフレームの指先の座標を表している。また、 v_1 と v_2 はそれぞれ P_0 から P_1 と P_2 へのベクトルを表しており、 n は平面の法線ベクトルを表している。法線ベクトル n と v_1 , v_2 のなす角度に注目すると、線分が平面を貫通していない場合、この角度は必ず鋭角となる。一方線分が平面を貫通している場合、片方の角度は鋭角となり、もう一つの角度は鈍角となる。この性質を利用すると、線分が平面と衝突しているかの判定は式 4.21 で行うことができる。

$$(v_1 \cdot n) \cdot (v_2 \cdot n) \leq 0 \quad (4.21)$$

次に、平面と線分の始点・終点の距離から、内分比を求め、貫通点の座標を求める。図 4.17 において、平面上の任意点を P_0 、線分の始点・終点をそれぞれ P_1 ・ P_2 、線分と平面の交点を P_3 、 P_0 から P_1, P_2, P_3 へのベクトルをそれぞれ v_1, v_2, v_3 とし、 n は平面の法線ベクトルを表している。始点から平面までの距離を d_1 、終点から平面までの距離を d_2 とすると、 d_1 , d_2 は式 4.22 で求めることができる。ここで、 P_3 は P_1 と P_2 の内分点であることから、 v_3 は内分比 a (式 4.23) を用いて式 4.24 と表すことができる。

⁷〇×つくろーどっとコム: <http://marupeke296.com/index.html>

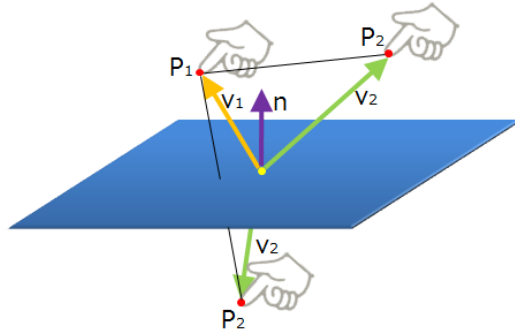


図 4.16: 平面と線分の衝突判定

$$d_1 = n \cdot v_1 \quad d_2 = n \cdot v_2 \quad (4.22)$$

$$a = \frac{d_1}{d_2} \quad (4.23)$$

$$v_3 = av_1 + (1 - a)v_2 \quad (4.24)$$

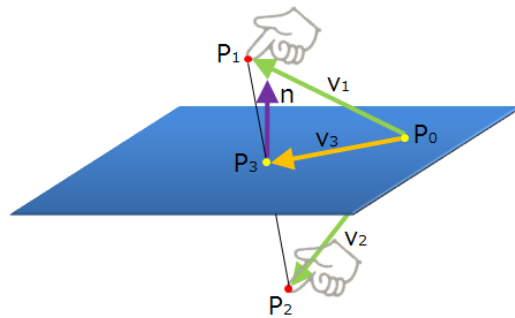


図 4.17: 平面と線分の衝突判定

最後に、求められた貫通点の座標が surface 内にあるかどうかの判定を行う。図 4.18 のように貫通点が surface 上にある場合、同色で示されている 2 つのベクトルに着目すると、いずれの場合においても全てのベクトルが反時計回りとなる。しかし、貫通点が surface の外側にある場合、時計回りをするベクトルの組み合わせが発生する。つまり、色分けしたベクトルの外積を計算し、求められた外積を正規化したものが全て等しい場合に貫通点が surface の内側にあることになる。

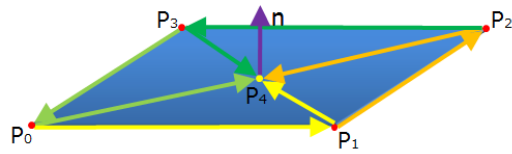


図 4.18: 貫通点が surface 内にあるかどうかの判定

4.5 表示画像生成部

表示画像生成部では、カメラから入力された画像に surface を重畳し、HMD に表示する画像の作成をおこなう。surface はマーカの位置に対応づけられているため、ユーザの位置及び方向に応じて見え方を変える。また、ユーザが surface の空間的位置を知覚することを支援するために、オクルージョンの再現を行った。

4.5.1 オクルージョンの再現

オクルージョンとは、ユーザの手が surface よりも手前にある場合、ユーザの手によって surface の一部が隠れることである。ユーザの手が surface よりも手前にあるか判定するために、surface の法線ベクトル n と、カメラ座標系における原点 O から指先座標 F までのベクトルの内積 $n \cdot OF$ を求める。指先が surface より手前にある場合、 $n \cdot OF$ の値は正であり、指先が surface より奥にある場合、 $n \cdot OF$ の値は負となる。手が surface より手前にあると判定された場合、カメラ入力画像に surface を描画した後、画像解析部によって求められた肌色領域のみ上から描画することによって、オクルージョンを再現することができる (図 4.19)。



図 4.19: オクルージョンの再現

第5章 関連研究

5.1 仮想物体とのインタラクション

椎尾らは実世界に仮想的な文字を書けるシステム「空気ペン」を開発した [16][21]. このシステムはユーザが任意の場所に仮想物体を作成することができる点で本研究と関連している. この研究は人々のコミュニケーションを支援することに焦点を当てている. 一方本研究は, 作成した仮想物体を情報端末として利用することを目的としている.

蔵田らは拡張現実感を用いて, 重畳上表示されている仮想物体に対し, ユーザの指先によってオブジェクト選択を可能とするインタフェース「ハンドマウス」を開発した [9]. ハンドマウスは, マウスにおけるポインティング, クリックに対応する操作を行うことができる. 本研究との相違は, 本研究はユーザが任意の空間上に仮想タッチパネルを「創り出し」それらを操作することに焦点を当てているのに対し, 蔵田らはすでに存在する仮想物体への操作を目的としている点である.

5.2 拡張現実感を用いたワークスペースの拡張

Rekimoto らは机や壁面といった平面に端末内の情報をプロジェクションすることによってワークスペースを拡張するシステム「Augmented Surfaces」を開発した [22]. Augmented Surfaces の実世界に仮想的なディスプレイを作成するという点が本研究と類似している. Augmented Surface は既存の端末の表示領域を拡張し, 情報共有を行うことを目的としていることに対し, 本研究は端末操作自体を仮想的な物体を通じて行う.

坂根らは拡張現実感を用いて, 計算機のデスクトップに表示されるようなアイコンを実世界に表示させ, デスクトップを拡張するシステムを開発した [8]. このシステムは既存端末の作業領域を増やすことに焦点を当てている. 一方本研究は端末自体を必要とせず, モバイルコンピューティングを行うことを目的としている.

第6章 議論

今回提案した仮想タッチパネルインタフェースのプロトタイプシステムである「AiR surface」を試用した。試用を通じて、矩形姿勢は surface の作成方法として有効であると感じられた。この理由を以下に述べる。

矩形姿勢を用いることによって、ユーザは意図した場所に、意図した大きさの surface を直感的に作成することができる。また、矩形姿勢は瞬時に創り出すことができる。その結果、仮想タッチパネルインタフェースのコンセプトである、自由に surface を創り出せることを実現することができ、タッチ端末の物理的制約問題を解決することができたと考えられる。

また、プロトタイプシステムは現状では1人のユーザでのみ利用可能であるため、surface の内容を複数人数で共有することはできない。しかし、surface の内容をユーザ本人だけが見ることができるため、surface の内容を他人に覗き見られることが無い。このため、アクセスコントロール問題の一部を解決することができたと考えられる。複数人数での利用を可能にし、surface をユーザ本人だけが見ることができる状態と、ユーザ本人以外も見ることができる状態の切り替えを実現することは今後の課題である。

一方で、問題点も発見した。今回の実装では、認識のロバストさや装置の手軽さを考慮した結果、手の認識方法として USB カメラによる肌色検出を採用した。しかし、日照条件や照明の影響を受けて肌の色が変化するため、状況によっては手の姿勢の認識が困難であり、その影響は予想以上に大きなものであった。環境の影響を受けずに様々な状況において利用可能にするためには、Kinect センサ¹を利用するといった、本研究により適した手法について再検討する必要がある。

また、拡張現実感を用いて実世界に重畳表示された仮想物体は、奥行きを感じるのが難しいという問題を発見した。オクルージョンの再現によって、surface がユーザの手よりも奥にあるのか、手前にあるのかを知覚することができるが、surface までの距離を知る手掛かりが少ない。原因として、人間は両眼視差によって奥行きを知覚しているが、ヘッドマウントディスプレイに表示される映像は両眼とも等しいことが考えられる。対応策として、ヘッドマウントディスプレイの左右のディスプレイに、左右カメラそれぞれの入力映像の映像を表示させる方法や、surface と手の距離に応じた影を surface 上に表示させるといった、視覚的手がかりを提示する方法が考えられる。

¹Kinect センサ (<http://www.xbox.com/ja-jp/kinect>)

第7章 結論

本研究では、ヘッドマウントディスプレイを装着した状態において、実世界に拡張現実感を用いて重畳表示されるインタフェース「仮想タッチパネルインタフェース」を提案した。

また、仮想タッチパネルインタフェースの概念実証として、実世界に surface を創り出し、タッチ入力を行うことができるシステム「AiR surface」を実装した。本システムでは、ユーザが両手で矩形を描く姿勢をとることによって surface を創り出し、タッチ入力を行うことができる。素手による操作によって、ユーザはデバイスを手に持つ必要がないため、物理的制約問題を解決することができた。また、ヘッドマウントディスプレイに表示される内容をユーザ以外が見られないことから、アクセスコントロール問題の一部を解決した。

アクセスコントロール問題は、本システムを複数人数で利用可能にし、surface の内容を創り出した本人だけが見ることができる状態と、創り出したユーザ本人以外も見ることができる状態を切り替える機能を実装することで完全に解決することができる。

今後は複数人数での利用を可能にし、アクセスコントロール問題の解決に取り組みたい。また、定量的な評価を行い、結果に基づいた改善を行いたい。

謝辞

本論文を執筆するにあたり，指導教員である田中二郎先生をはじめ，志築文太郎先生，三末和男先生，高橋伸先生には丁寧なご指導，有益なアドバイスを頂き，心から感謝を申し上げます。また，田中二郎先生にはテーマ選びから研究の進め方，論文執筆に至るまできめ細かいご指導を頂きました。深くお礼申し上げます。

インタラクティブプログラミング研究室の皆さまにはゼミや日頃の生活の中で，貴重な意見を頂き，大変お世話になりましたことをここに感謝致します。特に WAVE チームの皆さまには，今年度から新設された NERF チームに修士・博士が少ないことを心配し，同じチームの後輩と同じように面倒を見て下さったことを深く感謝申し上げます。

プログラムの不具合に悩んでいる時に助けて下さった，小林敦友先輩，村田雄一先輩，矢崎聖矢先輩，そして 3D プログラミングについて親切にご指導して下さいました伊藤隆朗先輩には深く感謝しております。

そして何よりも，家族が精神面，経済面など，全てにおいて私を支えて下さいました。この場を借りてお礼申し上げます。

最後に，日頃から私を支えてくれた友人たち，そして大学生活でお世話になった全ての方々に心よりお礼申し上げます。ありがとうございました。

参考文献

- [1] T Azuma. A survey of augmented reality. *Teleoperators and Virtual Environments* 6, 4, pp.355-385, 1997.
- [2] Poupyrev I, Tan D.S, Billinghurst M, Kato H, Regenbrecht H, and Tetsutani N. Developing a generic augmented-reality interface. 2002.
- [3] Azuma R, Baillot Y, Behringer R, Feiner S, Julier S, and MacIntyre B. Recent advances in augmented reality. *Computer Graphics and Applications, IEEE*, 2001.
- [4] 森下健, 中尾恵, 垂水浩幸, 上林弥彦. 時空間限定オブジェクトシステム spacetag : プロトタイプシステムの設計と実装. *Transactions of Information Processing Society of Japan*, 2000.
- [5] 神原誠之, 大隈隆史, 竹村治雄, 横矢直和. ビデオシースルー型拡張現実感のための実時間ステレオ画像合成. 電子情報通信学会論文誌 D-II Vol.J82-D-II No.10 pp.1775-1783, 1999, 1999.
- [6] 興梠正克, 蔵田武志, 坂上勝彦, 村岡洋一. パノラマ画像群を位置合せに用いたライブ映像上への注釈提示とその実時間システム. *The transactions of the Institute of Electronics, Information and Communication Engineers. D-II J84-D-II(10)*, 2293-2301, 2001-10-01, 2001.
- [7] 塚本昌彦, 板生知子. ウェアラブルコンピューティングとユビキタスサービス. 経営の科学 オペレーションズ・リサーチ 2004, Vol.49 No.4, pp 210-216, 2004.
- [8] 坂根裕, 塚本昌彦, 西尾章治郎. アイコンの立体表示を可能にするウェアラブル拡張デスクトップシステム. 第4回プログラミングおよび応用システムに関するワークショップ (SPA2001), オンラインプロシーディング, 2001.
- [9] 蔵田武志, 大隈隆史, 興梠正克, 坂上勝彦. ハンドマウス: ビジュアルウェアラブルズが可能にする拡張現実環境に適したインターフェイス. *Technical report of IEICE. PRMU 100(565)*, pp.69-76, 2001.
- [10] 淵一馬, 高橋伸, 田中二郎. ハンドジェスチャによる範囲選択手法を使った撮影システム. 全国大会講演論文集 第70回平成20年(4), "4-221"- "4-222", 2008.

- [11] 呉海元, 陳謙, 谷内田正彦. ファジイパターン照合を用いた色彩画像からの顔検出システム. *The transactions of the Institute of Electronics, Information and Communication Engineers* J80-D-2(7), 1997.
- [12] 本郷仁志, 山本和彦. 動領域内の肌色推定による顔領域および顔部品抽出. *The journal of the Institute of Image Information and Television Engineers* 52(12), 1840-1847, 1998-12-20, 1998.
- [13] 松橋聡, 藤本研司, 中村納, 南敏. 顔領域抽出に有効な修正 hsv 表色系の提案. *The Journal of the Institute of Television Engineers of Japan* 49(6), 787-797, 1995-06-20, 1995.
- [14] 加藤博一, 汐崎徳男, 橘啓八郎. テクスチャ画像からオンライン生成されたテンプレートのマッチングに基づく拡張現実感のための位置合わせ手法. *Transactions of the Virtual Reality Society of Japan* 7(2), 119-128, 2002-06-30, 2002.
- [15] 椎尾一郎. Rfid を利用したユーザ位置検出システム. 情報処理学会研究報告. HI, ヒューマンインタフェース研究会報告 2000(39), 45-50, 2000-05-12, 2000.
- [16] 椎尾一郎, 山本吉伸. コミュニケーションツールのための簡易型 ar システム. インタラクティブシステムとソフトウェア VIII, pp.117-124, 2000.
- [17] 中里祐介, 神原誠之, 横矢直和. ウェアラブル拡張現実感のための不可視マーカと赤外線カメラを用いた位置・姿勢推定. *Transactions of the Virtual Reality Society of Japan* 10(3), 295-304, 2005, 2005.
- [18] 佐藤清秀, 山本裕之, 田村秀行. カメラと 3 次元センサの組み合わせによる現実空間と仮想空間の位置合わせ手法. 日本バーチャルリアリティ学会論文誌 4(1), 295-302, 1999, 1999.
- [19] 加藤博一. 拡張現実感システム構築ツール artoolkit の開発. *Technical report of IEICE. PRMU 101(652)*, 79-86, 2002-02-14, 2002.
- [20] 加藤博一, Mark Billinghurst, 浅野浩一, 橘啓八郎. マーカー追跡に基づく拡張現実感システムとそのキャリブレーション. *Transactions of the Virtual Reality Society of Japan*, 1999.
- [21] 山本吉伸, 椎尾一郎. 空気ペン空間への描画による情報共有. 第 59 回情報処理学会全国大会講演論文集 (4), pp.39-40, 1999.
- [22] Rekimoto J and Saitoh M. Augmented surfaces: A spatially continuous workspace for hybrid computing environments. *CHI'99*, pp.378-385, 1999.
- [23] Klein G and Murray D. Parallel tracking and mapping for small ar workspaces. *In Proc. International Symposium on Mixed and Augmented Reality (ISMAR'07, Nara)*, 2007.