

ウェアラブル拡張現実感による情報端末の仮想化

加茂 浩之^{†1} 田中 二郎^{†1}

近年, ネットワークを介した Web アプリケーションが普及しており, ユーザはネットワークに接続可能な情報端末があれば, 端末の違いを意識することなく情報端末を使えるようになった. しかし, ユーザは利用場所に合った情報端末を利用するために, 複数の情報端末を所有することが多く, 常に情報端末の存在を意識している必要がある. そこで, 本論文ではユーザがネットワークに接続可能なウェアラブルコンピュータのみを着用することで, 情報端末の存在を意識することなく仮想的な情報端末を利用可能にする「情報端末の仮想化」を提案した. そして我々は, 仮想的な情報端末を操作するためのインタフェース「AiR surface」を開発した. AiR surface とは, ユーザが矩形を描く手の姿勢を作ることによって, 自由な場所に自由な大きさの仮想平面を作り出し, この仮想平面をタッチパネルのような入出力デバイスとして扱うことができるインタフェースである. さらに, AiR surface を用いて画像の選択・閲覧を行うケーススタディを行い, ユーザから使用感についてのコメントを得た.

Virtualization of Information Terminal with Wearable Augmented Reality

HIROYUKI KAMO^{†1} and JIRO TANAKA^{†1}

In recent years, Web applications using the network have become widespread. The user has become able to use terminals without any noticeable difference in the terminal, if the terminal can be connected to the network. However, in order to take advantage of the terminals in various places, the user must be aware of the existence of the terminals, because (s)he has a number of terminals for own use. In this paper, we propose a “Virtual terminal” where we apply virtualization technology to terminals. The user can use a virtual terminal by wearing only a wearable computer, like a head mounted display. We developed the interface for manipulating the virtual terminal. The interface is called “AiR surface”. The user can create a virtual surface using a hand posture like taking a picture. The surface is free size, can be created anywhere and is manipulated like a touch panel interface. In addition, we performed a case study for choosing and browsing images and we gathered comments regarding manipulation from the user.

1. はじめに

近年, ネットワークを介したアプリケーションサービスや, リモートデスクトップシステムが普及している. これらはネットワークに接続可能な情報端末さえあれば, ユーザが端末の違いを意識すること無くアプリケーションやデスクトップ環境を利用することを可能にする.

ユーザがこれらを利用するために用いる情報端末には様々なものがあり, 1人のユーザが複数の端末を持っていることが多く, 利用シーンによって使い分けている. 情報端末を大きく2つに分類すると, デスクトップ環境とモバイル環境がある. デスクトップ環境においては, ユーザは大きなディスプレイを用いて作業を行うことができるが, ユーザ自身が環境側に拘束されてしまうという問題点がある. 一方, モバイル環境においては, 利用場所が制限されないが, ユーザが小さなディスプレイを用いて作業を行わなければならないか, 端末自体の処理性能が低かったりといった問題点がある. ネットワークを介したサービスの普及により, ユーザは端末環境の違いを意識する必要がなくなったが, その反面, 環境側に拘束されたり, 情報端末を常に持ち歩いたりするように, 常に情報端末の存在を意識しなければならない.

本研究の目的は, ユーザが情報端末の存在を意識することなく, どこでも必要に応じた端末環境を利用可能にすることである. そこで, 我々はウェアラブルコンピュータのみ装着することで, ネットワークを介して必要に応じた端末環境を利用可能にする「情報端末の仮想化」を提案する. 本論文では, ユーザが仮想的な端末環境を利用するためのインタフェース AiR surface¹⁾ について述べる.

以下, 本論文の構成を示す. 第2章では我々が考える情報端末の仮想化と, 実現するために利用する技術について述べ, 第3章で仮想化された端末環境を利用するためのインタフェース「AiR surface」について述べる. 第4章ではシステムの実装について述べる. 第5章ではシステムの試用とその考察について述べる. 第6章では関連研究について述べ, 第7章でまとめと今後の課題について述べる.

^{†1} 筑波大学システム情報工学研究科コンピュータサイエンス専攻

2. 情報端末の仮想化

仮想化技術とは、ユーザ側から見た機能に影響を与えずに、システムの構成を柔軟に変更することができる仕組みのことである。サーバの仮想化を例に挙げると、1台の物理的なサーバマシンに複数台の仮想サーバが構築されている状況では、ユーザはあたかも物理的なサーバマシンが複数あるかのように複数台のサーバを利用することができる。

我々は、上述した仮想化技術を情報端末に応用することが有益であると考え、我々が考える情報端末の仮想化とは、ユーザがネットワークに接続可能なウェアラブルコンピュータを装着し、サーバ上の仮想マシンに接続することによって、情報端末に依存しないコンピューティング環境を構築することである。ここでのウェアラブルコンピュータとは、頭部に眼鏡のように取り付けるものであり、ディスプレイ機能、通信機能、カメラ機能を搭載しているものを想定している。ウェアラブルコンピュータは、ユーザからの入力を受付と、出力のみを行い、計算処理は全て仮想サーバ上で行われる。入出力を行うインタフェースについては次章にて述べる。次に、情報端末の仮想化を実現するために用いる、ウェアラブルコンピュータについて述べる。

現在のヘッドマウントディスプレイをはじめとするウェアラブルコンピュータは大きく重い、既存の携帯情報端末よりも性能が低い、着用している姿が社会的に受け入れられがたいといった問題がある。そのため、一般への普及はまだ進んでいない。しかし、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ほか、デザイン性を意識した製品も登場してきている。そこで、我々は近未来においてはヘッドマウントディスプレイが一般に普及するものだと考える。

3. AiR surface: 拡張現実感を用いた仮想平面インタフェース

AiR surface とは、拡張現実感²⁾を用いて実世界に重畳表示される仮想平面を利用するインタフェースである。既存の情報端末の表示インタフェースであるディスプレイやタッチパネルは平面型であることがほとんどである。そのため、表示インタフェースが平面型であることはユーザにとって馴染みやすいものと考え、AiR surface の表示インタフェースも平面型に設計した。次に、本インタフェースに用いた拡張現実感について述べる。

3.1 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。拡張現実感の特徴は、仮想物体があたかも実世界の特定の場

所にあるかのように表示することであり、ユーザは実世界を眺めながら同時に仮想物体を知覚することができる。この拡張現実感に対し、ユーザがコンピュータの作り出した情報のみを知覚できる技術、人工現実感 (VR: Virtual Reality) がある。

我々は、端末を操作する上で、現実世界の情報を知覚できることが重要であると考え、なぜならば、現実世界から知覚する情報を端末操作に活用したり、反対に情報端末から得た情報を現実世界の営みに活用したりすることがあるためである。例えば、前者では紙に印刷された資料に基づいて情報端末への入力を行うという行動が、後者では地図を閲覧しながら目的地に向かって歩くといった行動が例として挙げられる。

そこで、我々はユーザが実世界を眺めると同時に、コンピュータによって作り出された情報を知覚することができる拡張現実感を用いてインタフェースを設計することにした。また、拡張現実感をユーザに提示する手段としては、ヘッドマウントディスプレイやスマートフォンを使う手段があるが、表示領域が広いことと、端末を手を持つ必要がないという理由から、ヘッドマウントディスプレイを用いることにした。次に、拡張現実感を用いた仮想平面の特徴と、その利点について述べる。

3.2 拡張現実感を用いた仮想平面の特徴とその利点

既存の情報端末を持ち運んで利用するシーンを想定した場合、表示・操作領域の大きさと持ち運びやすさがトレードオフとなるほか、端末を手を持つか机の上に置いて操作する必要があるため、持ち手に負担がかかることや、端末を置くための机が必要であるといった問題がある。しかし、仮想平面は空気中に浮遊した状態で存在することができるため、ユーザは情報端末を手を持つ必要が無く、情報端末を置くための机も必要ない。したがって、ユーザはどのような場所でも大きさの制約を受けることなく端末操作が可能になるという利点がある。また、仮想平面は数の制約が無いため、ユーザは任意の数の端末操作が可能になるという利点もある。

3.3 仮想平面とのインタラクション手法の検討

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ものがある。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。

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。たとえば、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³⁾がある。一方、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ため、仮想化された情報端末により複数の端末を持ち歩く必要がなくなるという利点を損なってしまう。

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ったり装着したりする必要が無いため、手の自由度に一切の制約を受けないという利点がある。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⁴⁾がある。素手を用いる方法は、ユーザは利用に備えてデバイスを持ち歩く必要が無いため、提案手法の利点を活かすことができる。したがって、我々は素手を用いて仮想平面とインタラクションを行う方法を採用した。

3.4 仮想平面の作成

我々は素手による仮想平面の作成方法の検討を行った。我々は、仮想平面を作成する方法は、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。

- 仮想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を採用した⁵⁾。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1(a))、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1(b))。

ユーザが矩形姿勢を作り、仮想平面を作成したい平面領域で一定時間静止すると、その場所に仮想平面が作成される。このように、矩形姿勢の状態を維持し、仮想平面を作成する操作を「Make 操作」と呼ぶ。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ることによって、自由な大きさの仮想平面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、自由な数の仮想平面を創り出すことが可能である。

3.5 仮想平面への入力

前節で述べた通り、仮想平面は複数枚作成されることが想定されている。そのため、操作する場合に操作対象となる仮想平面を明らかにする必要がある。我々は操作方法を検討する際、既存の情報端末に搭載されているタッチパネルインタフェースを参考にした。タッチパネルインタフェースの操作方法を参考にしたことには2つの理由がある。1つめはタッチパ

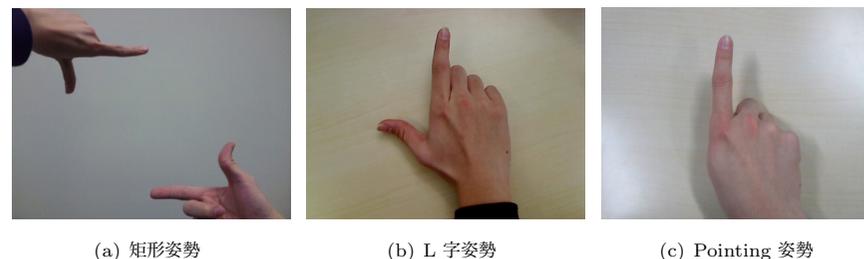


図1 操作に用いる手の姿勢

ネルインタフェースの操作が直接操作であり、操作対象となる仮想平面を明示することができるためである。2つめは近年タッチパネルインタフェースが搭載された携帯端末が普及しており、操作方法がユーザに馴染み深いものであると考えたからである。

机上に置かれているタッチパネルに対してユーザがタッチ操作を行うとき、人差し指を立てて画面に触れる場合が多い。このような、人差し指を立てている手の姿勢(図1(c))を「Pointing 姿勢」と呼ぶ。仮想平面に対する入力操作は、この Pointing 姿勢をとって行う。次に、仮想平面に対する入力操作について述べる。

Pointing 操作

Pointing 操作とは、指先を仮想平面上の1点と隣接させる操作を指す(図2(a))。隣接している1点を入力点とする。この操作はタッチパネルインタフェースにおける、タッチパネル上に指先を置く操作にあたる。

Through 操作

Through 操作とは、指先が仮想平面を突き抜ける操作を指す(図2(b))。指先が仮想平面の手前から奥に向かって移動した時の、突き抜ける瞬間に通過した1点を入力点とする。指先が仮想平面の奥から手前に向かって移動した場合は Through 操作としてみなさない。この操作はタッチパネルインタフェースにおける、タップ操作にあたる。

Drag 操作

Drag 操作とは、Pointing 操作を行っている状態で、指先を仮想平面上の上下左右へ移動させる操作を指す(図2(c))。Pointing 操作を行っている状態が継続している間、隣接している点を入力点とする。この操作はタッチパネルインタフェースのドラッグ操作と同等の操作である。

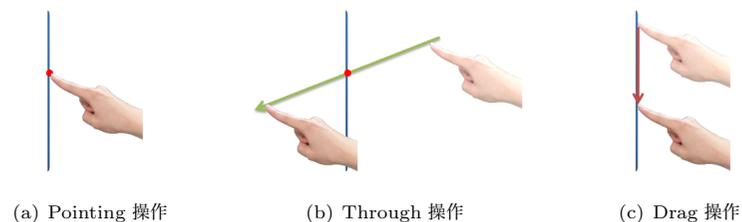


図 2 仮想平面への入力操作

3.6 利用シナリオ

AさんはT大学の学生であり、11時30分から行われるT先生との面接までT大学内のコーヒESHOPで休憩している(図3)。Aさんは面接に遅刻しないよう、仮想平面に大きな時計を表示させた。また、Aさんは面接に備え、T先生の研究室のWebサイトを閲覧し、研究室についての情報収集を行うことにした。また、ソーシャルネットワーキングサービスを見ながら、T先生の研究室の学生がどのようなことに感心があるのかを同時にチェックすることにした。事前の情報収集をしっかりと行うことができたため、AさんはT先生との面接を無事終えることができた。

このストーリーはAiR surfaceの利点を有効に活用した利用例である。AさんはコーヒESHOPという端末を置くスペースが限られた環境で、3つの仮想端末を利用している。これは、物理的な端末が存在しないという特徴を活用している。また、仮想平面は表示させる情報や、起動するアプリケーションに合わせた大きさになっている。これは、ユーザが指定した大きさの仮想平面を作り出せるという特徴を活用している。

4. AiR surface の実装

我々は仮想化された情報端末を操作するためのインタフェース「AiR surface」のプロトタイプを実装した。本章ではAiR surfaceの具体的な実装方法について述べる。

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

*1 eMagin社 (<http://www.emagin.com/>)

メラを取り付けた(図4)。このUSBカメラにはLogicool社^{*2}の2-MP Portable Webcam C905mを用いた。

なお、プロトタイプに用いたヘッドマウントディスプレイは非透過型であり、取り付けられたUSBカメラから入力された画像をそのままヘッドマウントディスプレイに表示させる、ビデオスルー方式を採用した。これには2つの理由がある。1つめに、現在のヘッドマウントディスプレイは透過型よりも非透過型の方が解像度が良いからである。仮想平面には多くの文字情報を表示する状況が想定されるため、高解像度であることが有効である。2つめは、カメラ入力画像を画像解析して得られた指先などの位置情報と、ユーザが意図している指先位置との整合性をとるためである。透過型ヘッドマウントディスプレイでは、ユーザの目に入る現実世界の情報をそのまま知覚するため、カメラ入力画像とは誤差が生じる。



図 3 AiR surface 利用シーン



図 4 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

本システムは、ヘッドマウントディスプレイのカメラ入力画像が与えられてから、ヘッドマウントディスプレイに表示させるまでの流れを、以下の5つの処理部により行っている。それぞれの処理部が行っている処理の詳細について述べる。

なお、以下に述べる処理の実装には3つのライブラリを使用した。画像解析には

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

OpenCV^{*1}を用いた。実世界の特定位置の検出のために AR ToolKit⁽⁶⁾⁷⁾を用いた。カメラ入力画像に仮想平面を重畳表示するために OpenGL^{*2}を用いた。

- 画像解析部：カメラ入力画像から、指先と Pointing 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
- 操作解析部：ユーザの動作を環境側に反映
- 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、カメラ入力画像から肌色検出を行い、検出された領域から指先、Pointing 姿勢、L 字姿勢、矩形姿勢の検出を行う。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。

- 肌色領域検出：カメラ入力画像から肌色領域を検出
- 指先検出：検出した肌色領域を基に指先の検出と Pointing 姿勢を検出
- L 字姿勢検出：検出した指先を基に L 字姿勢を検出
- 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 肌色領域抽出

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検出する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⁸⁾⁹⁾。別の表色系に変換するのは、RGB 表色系における肌色は広範囲に分布するため、閾値を定めて肌色のみを抽出することが難しいからである。我々はこの表色系に、色が持つ明るさの情報と、色味の情報を分離して表される YUV 表色系を用いた。RGB 表色系から YUV 表色系に変換する式を式 1、式 2、式 3 に示す。肌色領域を検出するための閾値を式 4 に示す。式 4 は室内において試行を繰り返すことで得られた値である。照明環境や太陽光の当たり方によって、精度高く肌色領域のみを検出することができる閾値が変化するため、今回はスライダを用いて柔軟に閾値を調整することを可能にした。しかし、スライダを用いた閾値の調整はユーザが試行錯誤を繰り返さなければならないという問題点がある。ユーザが閾値を調整する必要のないキャリブレーションシステムを導入するといった対策を検討する必要がある。

*1 OpenCV(<http://opencv.jp/>)

*2 OpenGL(<http://www.opengl.org/>)

ある。

上述した方法で肌色領域を抽出した後、膨張と収縮処理を行うことによってノイズを除去し、残った肌色領域に対してラベリングを行う。ラベリングには井村が提供するラベリングクラス^{*3}を用いた。そして、面積が最も大きい領域と、次に大きい領域のみを残し、残りの領域は除去する。2つの肌色領域としてユーザの右手と左手が認識されることを想定している。

$$Y' = (0.256 \times R + 0.504 \times G + 0.098 \times B) + 16 \quad (1)$$

$$U' = (-0.148 \times R - 0.291 \times G + 0.439 \times B) + 128 \quad (2)$$

$$V' = (0.439 \times R - 0.368 \times G - 0.071 \times B) + 128 \quad (3)$$

$$64 < Y' < 240 \quad 94 < U' < 125 \quad 140 < V' < 255 \quad (4)$$

4.2.2 指先検出

肌色領域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。我々は指の形状の特徴を利用し、肌色領域において凸型の形状をしている場所を指として検出することにした。最初に、肌色領域の輪郭を OpenCV の関数によって検出する。次に輪郭に沿って、一定の間隔 d を開けた 3 点 P_1 , P_2 , P_3 のなす角度 θ を計算する (図 5)。 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 5) と $\cos \theta$ (式 6) について閾値 (式 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 6)。それぞれの集合に対し、ラベリングを用いて重心を求めることで、凸型をしている部分に対する 1つの端点と、2つの側点をそれぞれ 1 点に定める。1つの端点と 2つの側点の組を 1本の指として認識する。検出された指が 1本であった場合、手は Pointing 姿勢であると認識する。

$$\sin \theta = \frac{P_2P_1 \times P_2P_3}{|P_2P_1| |P_2P_3|} \quad (5)$$

$$\cos \theta = \frac{P_2P_1 \cdot P_2P_3}{|P_2P_1| |P_2P_3|} \quad (6)$$

$$0 \leq \sin \theta \leq \frac{1}{2} \quad \frac{\sqrt{3}}{2} \leq \cos \theta \leq 1 \quad (7)$$

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

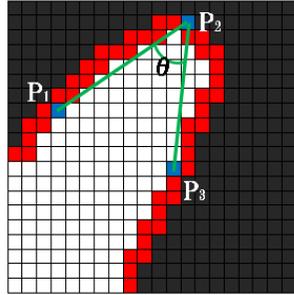


図 5 指先の検出方法

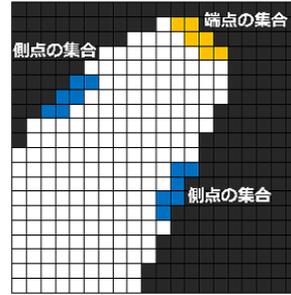


図 6 側点の集合と端点の集合

4.2.3 L字姿勢検出

次に、検出された指の情報から、2本の指の組み合わせがL字姿勢を構成しているものを検出する。図7を用いて説明する。図中の親指にあたる指先情報を Finger A、人差し指にあたる指先情報を Finger B とする。最初に、Finger A と Finger B について、2つの側点 side1 と side2 の中点 C_A 、 C_B を求める。また Finger A、Finger B それぞれの端点を T_A 、 T_B とすると、直線 $C_A T_A$ と直線 $C_B T_B$ の交点 V は、それぞれの直線の傾き d_A 、 d_B (式8) を用いて式9で表すことができる。

次に、ベクトル VT_A とベクトル VT_B がなす角 θ の余弦 (式10) を計算し、これが式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1に示す。この結果から、親指の長さ÷人差し指の長さはおおよそ0.45から0.55の間になることが予想できる。この値を閾値として、式12を用いてフィルタリングをおこなう。この条件をみたした組み合わせの、2本の指の端点と、交点をL字姿勢情報として認識する。

$$d_A = \frac{T_{Ay} - C_{Ay}}{T_{Ax} - C_{Ax}} \quad d_B = \frac{T_{By} - C_{By}}{T_{Bx} - C_{Bx}} \quad (8)$$

$$V_x = -\frac{(T_{Ay} - d_A T_{Ax}) + (T_{By} - d_B T_{Bx})}{d_A - d_B} \quad V_y = d_A V_x + d_B T_{Bx} \quad (9)$$

$$\cos \theta = \frac{VT_A \cdot VT_B}{|VT_A| |VT_B|} \quad (10)$$

$$\frac{1}{2} \leq \cos \theta \leq 1 \quad (11)$$

$$0.45 \leq \frac{|VT_A|}{|VT_B|} \leq 0.55 \quad (12)$$

表 1 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

4.2.4 矩形姿勢検出

最後に、検出されたL字姿勢情報の組み合わせから矩形姿勢の検出を行う。2つの手が矩形姿勢をつくる時、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8を例に挙げる。図における左手にあたるL字姿勢をL-posture A と L-Posture B とし、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13求め、式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 $|V_{By} - V_{Ay}|$ とした。

$$\theta = \cos^{-1} \left\{ \frac{V_A I_A \cdot V_B I_B}{|V_A I_A| |V_B I_B|} \right\} \quad (13)$$

$$\frac{7}{8}\pi \leq \theta \leq \pi \quad (14)$$

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、Pointing 姿勢、L字姿勢、矩形姿勢についての情報が得られた。しかし、データ中の座標は、すべてスクリーン座標系であり、2次元の情報しか持っていない。

ステレオ画像処理部では2つの2次元データを統合し、Pointing 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、カメラ座標系における3次元データを求めることができる。

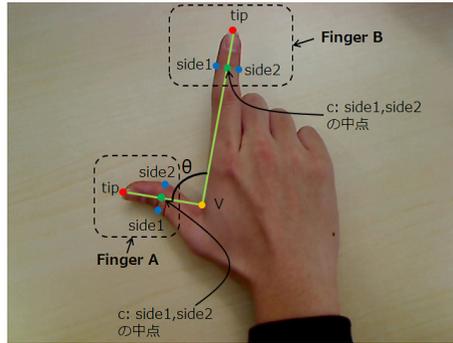


図 7 L 字姿勢検出

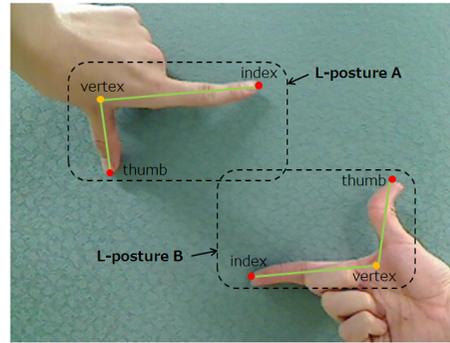


図 8 矩形姿勢検出

ステレオ法を用いて 3 次元座標を求める方法は文献¹⁰⁾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 B 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 15 によって求めることができる。

$$Z = \frac{B \cdot f}{x_l - x_r} \quad X = Z \cdot \frac{x_l + x_r}{2} \quad Y = Z \cdot \frac{y_l + y_r}{2} \quad (15)$$

4.4 座標解析部

座標解析部では、ステレオ画像処理部において統合した 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する。我々は、実世界の特定の位置の検出を、AR ToolKit⁽⁶⁾⁷⁾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、これを 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仮想平面の位置を実世界の位置と対応づけることができる。AR ToolKit を用いることによって、マーカの中心座標をカメラ座標系における座標に変換する座標変換行列を得ることができる。この座標変換行列の逆行列はすなわち、カメラ座標系からマーカ座標系に変換する座標変換行列となる。これを利用することによって、

3 次元データを実世界に配置された AR マーカに対応させることができる。

4.5 操作解析部

本インタフェースには 4 つの操作がある。Make 操作と、Pointing 操作、Through 操作、Drag 操作である。これらの操作を検出する方法について述べる。

4.5.1 Make 操作の検出

ユーザが 5 フレームの間継続して矩形姿勢を行っており、かつ矩形姿勢の位置に大きな変化がなかった場合、Make 操作が行われたと判断する。Make 操作が行われたら、5 フレーム間における矩形姿勢の位置の平均を求め、その場所に仮想平面を作成する。フレーム数が少ない場合、ユーザが意図した場所以外に仮想平面が作成されてしまう恐れがある。また、フレーム数が多い場合、ユーザが矩形姿勢を維持する時間が長くなり、操作性に影響を及ぼす。

4.5.2 Pointing 操作、Through 操作、Drag 操作の検出

ユーザがこれらの操作を行うとき、ユーザは Pointing 姿勢の状態で行う。最初に、Pointing 操作が検出された場合、仮想平面との衝突判定に利用する 2 つの線分を作成する。それぞれの線分と、仮想平面との衝突判定を行うことによって操作を検出する。

1 つめの線分は、指先の 3 次元座標からカメラ方向 (手前) へ一定距離の座標を始点とし、その反対側へ同一の距離の座標を終点とする。この線分はユーザの指先の動きに関係なく、常に指先の周辺において衝突判定を行うため、この線分と仮想平面が衝突していた場合、Pointing 操作であると認識する。また、この状態において指先を仮想平面上に上下左右へ移動させた場合、Drag 操作であると認識する。

2 つめは、前フレームにおいても Pointing 姿勢が検出されていた場合、前フレームにおける指先の座標を始点とし、現在のフレームにおける指先の座標を終点とする線分を作成する。この線分と仮想平面が衝突していた場合、Through 操作であると認識する。

それぞれの線分と、仮想平面の衝突判定は以下の 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、平面の法線ベクトルと線分の始点・終点から、線分を延長した直線が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 9(a) 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 16 で行うことができる。

$$(v_1 \cdot n) \cdot (v_2 \cdot n) \leq 0 \quad (16)$$

次に、線分の始点・終点から平面までの距離から、内分比を求め、貫通点の座標を求め。図 9(b) 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 17 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 18) を用いて式 19 と表すことができる。

$$d_1 = n \cdot v_1 \quad d_2 = n \cdot v_2 \quad (17)$$

$$a = \frac{d_1}{d_2} \quad (18)$$

$$v_3 = av_1 + (1 - a)v_2 \quad (19)$$

最後に、求められた貫通点の座標が仮想平面の内側にあるかどうかの判定を行う。図 9(c) 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。

4.6 表示画像生成部

表示画像生成部では、カメラ入力画像に仮想平面を重ね合わせ、HMD に表示する画像の作成をおこなう。仮想平面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。

オクルージョンとは、ユーザの手が仮想平面よりも手前にある場合、ユーザの手によって仮想平面の一部が隠れることである。ユーザの手が仮想平面よりも手前にあるか判定するために、仮想平面の法線ベクトル 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。指先が仮想平面よりも手前にある場合、 $n \cdot OF$ の値は正で

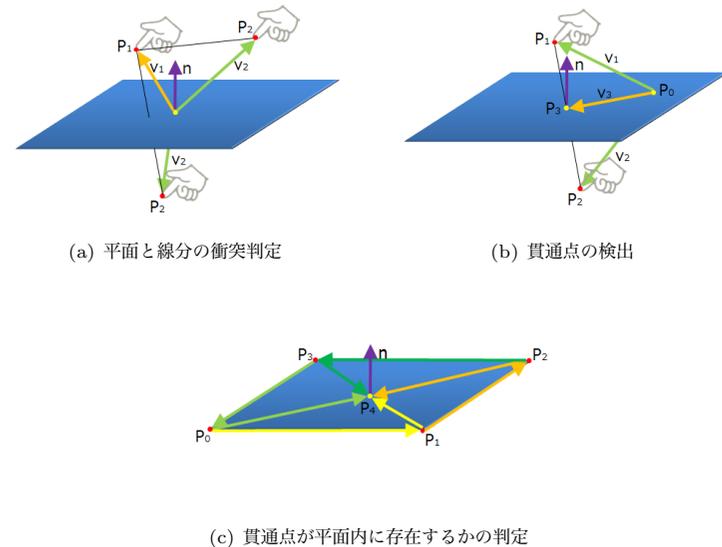
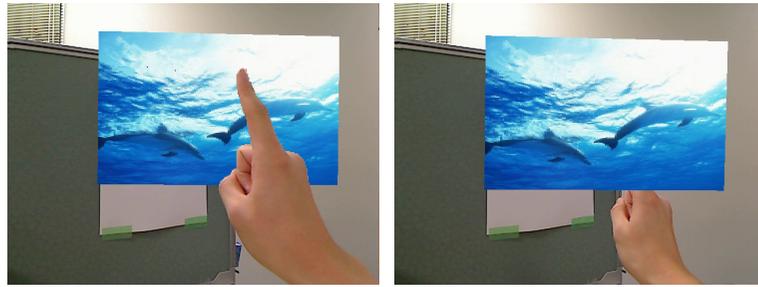


図 9 衝突判定

あり、指先が仮想平面より奥にある場合、 $n \cdot OF$ の値は負となる。手が仮想平面より手前にあると判定された場合、カメラ入力画像に仮想平面を描画した後、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。手が仮想平面よりも手前にある場合、図 10(a) のように表示され、手が仮想平面よりも奥にある場合、図 10(b) のように表示される。

4.7 画像ビューワアプリケーション

情報端末では様々なアプリケーションが利用可能であるが、数あるアプリケーションの一例として、画像の選択と閲覧を行うアプリケーションを開発した。ユーザはヘッドマウントディスプレイを装着し、Make 操作を行う (図 11(a)) と、所有している画像のサムネイルが一覧表示される (図 11(b))。サムネイル画像に対して Through 操作を行うと、操作を行った対象の画像が仮想平面全体に拡大表示される (図 11(c))。Through 操作を行った後、手を仮想平面の奥から手前に移動させる場合は、Through 操作としては認識されない (図 11(d))。そして、再び Through 操作を行うと、一覧表示に戻る (図 11(e))。また、所有している画像の数が多き場合、サムネイルが仮想平面に収まりきらないため、Drag 操作によつ



(a) 手が仮想平面よりも手前にある場合 (b) 手が仮想平面よりも奥にある場合

図 10 オクルージョンの再現

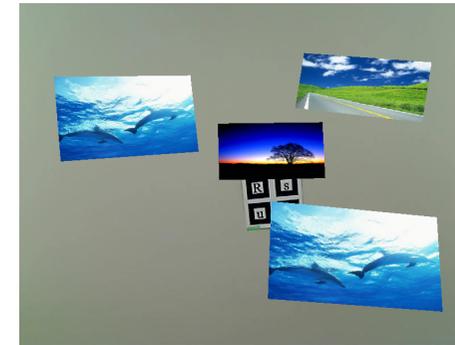


図 12 複数の仮想平面で画像を閲覧している様子

て一覧画面の表示領域を切り替えることができる (図 11(f)(g)).

ユーザは複数の仮想平面を作成することによって、複数の画像を同時に閲覧することができる。複数の仮想平面を用いて画像を閲覧している様子を図 12 に示す。

5. ケーススタディ

我々は AiR surface を用いた画像ビューワアプリケーションを、3名の被験者に自由に使ってもらったケーススタディを行った。被験者は日頃から情報端末に使い慣れている大学生3名である。得られたコメントは、操作感についてのコメントと、被験者が着用したヘッドマウントディスプレイについてのコメントに分類された。それぞれのコメントと考察について述べる。

5.1 操作感についてのコメントと考察

Make 操作について、2名からは「ほぼ意図した通りの場所と大きさに作ることができた。」、「好きなだけ仮想平面を作ることができて楽しい。」、「場所が若干ずれたが、それほど気にならなかった。」といった、ポジティブなコメントが得られた。1名からは「Make 操作の認識が思うようにされなかった。」というコメントが得られた。この被験者の Make 操作を観察したところ、L字姿勢がカメラと並行な向きで行われておらず、人差し指が前の方を向いていたことが原因であった。現在の実装 (4.2.3) では、L字姿勢がカメラと並行に行われることが前提となっている。しかし、今回のように人差し指が前を向ってしまうことを想定し、視覚的フィードバックを与えるか、閾値を見直すといった対策を検討する必要がある。

Pointing 操作, Through 操作, Drag 操作については、3名全員から「仮想平面までの距離がわかりにくい。」、「触った感じがしないので Pointing 操作をできているのかわからない。」といった距離の知覚やフィードバックに関するコメントが得られた。これらのコメン

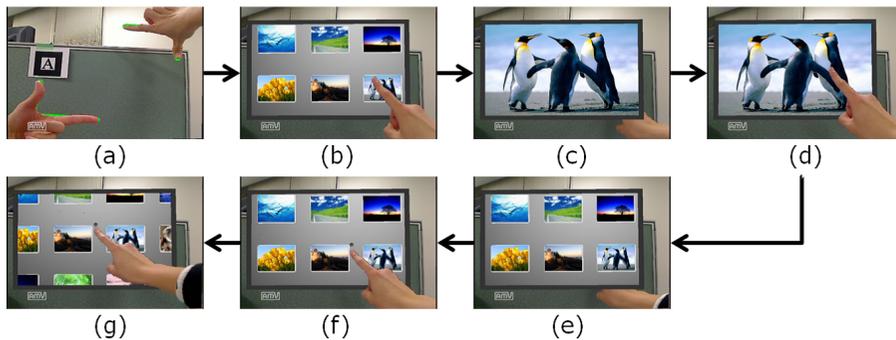


図 11 画像ビューワアプリケーション



図 13 視覚的フィードバックの例

トから、オクルージョン(4.6)による手と仮想平面の前後関係だけでは、ユーザが仮想平面の位置を知覚するための情報が不足していると言える。ユーザに触覚フィードバックを与える場合、ユーザの手にデバイスを装着する必要がある、物理的な情報端末を持ち歩く必要がないという利点を損なってしまう。そこで、我々は視覚的フィードバックによって、ユーザが仮想平面の位置を知覚することを補助する方法が適切であると考えた。例えば、仮想平面に指先との距離に応じた濃さの影を表示(図 13(a))したり、指先と仮想平面の距離を数値で表示(図 13(b))したり、仮想平面の影を環境側に表示する(図 13(c))といった方法がある。

5.2 ヘッドマウントディスプレイについてのコメントと考察

ヘッドマウントディスプレイに関しては、「視野が狭い」、「体を動かしてから表示される映像が動くまでに若干のタイムラグがあり酔う」、「装置が重い」といったコメントが得られた。視野が狭いことと、表示される映像にタイムラグがあることは、ヘッドマウントディスプレイがビデオシースルー方式であり、ユーザの視野角がカメラの視野角に依存しているためである。現在は 4.1 節で述べた理由で非透過型のヘッドマウントディスプレイを用いているが、透過型ヘッドマウントディスプレイの性能が向上し、小型化されればこれらの問題は解決することができる。

6. 関連研究

6.1 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¹¹⁾¹²⁾。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⁴⁾。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」それら进行操作することに焦点を当てているのに比べ、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。

6.2 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¹³⁾。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て、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³⁾。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。

7. まとめと今後の課題

本論文では、ユーザがネットワークに接続可能なウェアラブルコンピュータのみを着用することによって、情報端末の存在を意識することなく仮想的な情報端末を利用可能にする「情報端末の仮想化」を提案した。そして我々は、仮想的な情報端末を操作するためのインタフェース「AiR surface」を開発した。AiR surface を利用することで、ユーザは自由な場所に自由な大きさの仮想平面を作り出し、この仮想平面をタッチパネルのような入出力デバイスとして扱うことができ、仮想化された情報端末を操作することが可能となった。さらに、AiR surface を用いて画像の選択・閲覧を行うケーススタディを行い、ユーザから使用感についてのコメントを得た。今後の課題として、ケーススタディで得られたユーザのフィードバックを実装に反映したい。

参 考 文 献

- 1) 加茂浩之, 田中二郎. 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. 情報処理学会第 73 回全国大会講演論文集 (4), pp.229-230, 2011.
- 2) TAzuma. A survey of augmented reality. *Teleoperators and Virtual Environments* 6, 4, pp.355-385, 1997.
- 3) 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.
- 4) 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.
- 5) 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.
- 6) 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.
- 7) 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.
- 8) 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.
- 9) 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.
- 10) 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.
- 11) 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.
- 12) 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.
- 13) Rekimoto J and Saitoh M. Augmented surfaces: A spatially continuous workspace for hybrid computing environments. *CHI'99*, pp.378-385, 1999.