

推薦論文

# AirTarget：光学シースルー方式HMDと マーカレス画像認識による 高可搬性実世界志向インタフェース

入江 英嗣<sup>1,a)</sup> 森田 光貴<sup>2</sup> 岩崎 央<sup>1</sup> 千竈 航平<sup>1</sup> 放地 宏佳<sup>1,†1</sup> 小木 真人<sup>1</sup>  
檜原 裕大<sup>1,†2</sup> 芝 星帆<sup>1,†3</sup> 眞島 一貴<sup>1</sup> 吉永 努<sup>1</sup>

受付日 2013年4月9日, 採録日 2014年1月8日

**概要：**本論文では、光学シースルー方式 HMD 向けのユーザインタフェースとして、仮想オブジェクトや現実オブジェクトを指で直接指定することのできる“AirTarget”システムを提案する。AirTarget は HMD に取り付けられたカメラからユーザの指の位置を検出し、カメラと視線のずれを補正する新規アルゴリズムにより、指先にカーソルを重畳表示する。軽量のアルゴリズムにより指先はマーカレスで検出され、特定の入力デバイスや外部計算機を必要としない。簡単なジェスチャによってコンピュータにコマンドを送ることができ、プログラムの入力インタフェースとして機能する。ユーザは仮想平面上のデスクトップを指差して操作することや、視界に入った現実オブジェクトを指で切り出して画像検索のクエリとすることができる。このような可搬性と操作感から、外出先を含む日常生活のあらゆるシーンにコンピューティングを浸透させるインタフェースとして有効である。Android 端末として機能する HMD 上に提案システムを実装し、システムのリアルタイム性、検出精度、操作感の評価を行った。提案システムは 18 フレーム以上のリアルタイム動作を実現し、また、クリックジェスチャ操作を高い認識率および精度で実現した。

**キーワード：**ユーザインタフェース、ヘッドマウントディスプレイ、拡張現実感、指認識、ウェアラブルコンピューティング

## AirTarget: A Highly-portable Markerless User Interface Using Optical See-through HMD

HIDETSUGU IRIE<sup>1,a)</sup> MITSUTAKA MORITA<sup>2</sup> HIROSHI IWASAKI<sup>1</sup> KOHEI CHIKAMA<sup>1</sup>  
HIROYOSHI HOUCHI<sup>1,†1</sup> MASATO KOGI<sup>1</sup> HIROTAKA KASHIHARA<sup>1,†2</sup> SEIHO SHIBA<sup>1,†3</sup>  
KAZUKI MAJIMA<sup>1</sup> TSUTOMU YOSHINAGA<sup>1</sup>

Received: April 9, 2013, Accepted: January 8, 2014

**Abstract:** In this paper, a system “AirTarget”, which can point virtual and real object directly with user’s finger is proposed for optic see-through HMD devices. The camera attached to the HMD device detects the position of user’s fingertip, calibrates the gap between the sight of the camera and eye, and displays the cursor overlapped to the finger on the virtual plain. Finger detection is done in markerless image processing, so that it does not require specific input devices or external computer. This system enables to send control commands by simple gesture, working as a self-contained interface. The user is able to point a virtual object with their finger, and able to cut out an object of the real sight which can be used as a query of the image searching. Such portability and operability enforces AirTarget as an effective interface to be attached to every scene of daily life including outdoors. We implemented the proposal system on the HMD as an Android device and evaluated its real-time properties, detection accuracy, and operability. The algorithm achieved more than 18 fps which is sufficient to real time operation. Also, the quick gesture operation was achieved in high accuracy and recognition rate.

**Keywords:** user interface, head mounted display, augmented reality, finger detection, wearable computing

## 1. はじめに

計算機の小型高性能化やネットワークサービスの普及を背景に、コンピュータによる支援は日常生活の中へますます浸透しようとしている。個人が携帯する情報端末にはセンサ機能、ネットワーク機能、コンピューティング機能が備えられ、生活の中のあらゆる場所・機会健康モニタリングからクラウドサービスまで様々なサービスを必要に応じて利用することができる。

しかし、個人向けコンピューティングの多くは対話型のサービスであり、その利用中は端末の操作に注力しなければならない。たとえば、普及の進んでいる携帯端末へのタッチ入力では、視界を端末の画面へ向け、片手で端末を把持し、もう一方の手でポイント操作を行わなければならない。外出先など、日常生活のあらゆる機会健康モニタリングからクラウドサービスまで様々なサービスを必要に応じて利用するためには、身体的・視界的に制限が少なく、直観的な操作を実現するユーザインタフェースが求められる。このようなインタフェースはまだ一般的ではなく、実用用途だけでなく、エンタテインメント面でも発展が期待されている。

本論文ではこのような個人の日常生活支援の基本インタフェースとして、光学シースルー方式 HMD を用い、指先での直観的操作を可能とする“AirTarget”を提案する。AirTarget は光学的シースルー方式 HMD から見える視界を指差すことによって、HMD 上に重畳表示される仮想平面および現実視界をポイントすることのできるインタフェースである。指先は HMD に装着されたカメラによって検出されるため、特殊なデバイスや作業場を必要とせず、HMD の可搬性を活かしてどこでも使用することができる。

光学シースルー方式では、ユーザの肉眼視界と表示グラフィックの座標を合わせなければならない。従来、あらかじめ登録された固定平面上への重畳表示手法は提案されていたが、ユーザの指へ AR (Augmented Reality) オブジェクトを表示するような光学シースルー方式向け座標変換はまだ定まった手法がない。我々は、入力操作のために動かす指先はある一定の平面内を動くことに注目し、簡単な座標変換によって自然な操作感を実現する。また、HMD に取り付けられたカメラから見える手の形状の特徴に着目し、頑

健かつ軽量なアルゴリズムを提案する。

以降、本論文は以下のように構成される。2 章では HMD によるユーザ支援技術を紹介し、将来の個人コンピューティングのためのインタフェースに望ましい性質を議論する。3 章では AirTarget 実現のためのアルゴリズム軽量化およびキャリブレーションについて概要を述べる。4 章では AirTarget の実装についてシステム構成やアルゴリズムを詳細に述べる。5 章で評価について述べ、6 章でまとめを示す。

## 2. 関連研究

### 2.1 HMD を使用した MR・AR 環境

HMD (ヘッドマウントディスプレイ) は頭部に装着する表示装置であり、特に密閉型の HMD はその没入感から VR (Virtual Reality) の代表的なデバイスとして知られている。両眼に対応したものはステレオ視による立体表示が可能である。Robinett らは HMD における自然な 3D 表示のための表示ゆがみやステレオ視のモデル化を行っている [1]。

HMD は把持の必要がなく、顔の向きや体勢によらずつねに目の前に仮想画面を表示可能であることから、作業支援のウェアラブルデバイスとしての利用も数多く研究されている。これらの技術では、ユーザが見ている現実空間に対してコンピュータが生成した仮想オブジェクトを重ね表示する、いわゆるシースルー表示を行うことが多い。シースルー表示の実現にはビデオシースルー方式と光学シースルー方式の 2 通りがあり、それぞれ長所と短所を持っている。密閉型の HMD を用いてカメラ画像と CG を合成した画像を表示するビデオシースルー方式では、正確な合成ができる反面、ユーザは直接外界を見ることはできず、視界の制限や表示遅延のため安全に使用できる場所は限られる。これに対し、光学シースルー方式は肉眼の視界に対して CG を重畳する方式であり、視界が広く、現実視界部分には表示遅延がない。この性質は移動中の使用や現実の物体を用いた作業時の使用に適している。一方で、肉眼の視界とカメラの視界にはずれがあるため、CG の表示位置とユーザの視界を合わせるためには表示座標を変換しなければならない。カメラ取得画像の座標へ直接合成すればよいビデオシースルー方式よりも難しい。

HMD を使用した作業支援システムの例として、Billinghurst らは協調作業のための MR (Mixed Reality) 環境を提案している [2]。彼らの手法では、頭の向きをトラッキングして CG の表示位置を調整する。これにより、仮想画面中のオブジェクトがあたかも自分の体や現実空間

<sup>1</sup> 電気通信大学情報システム学研究所  
Graduate School of Information Systems, The University of  
Electro-Communications, Chofu, Tokyo 182-8585, Japan

<sup>2</sup> 電気通信大学電気通信学部  
Department of Communication Engineering and Informatics,  
The University of Electro-Communications, Chofu,  
Tokyo 182-8585, Japan

<sup>f1</sup> 現在、株式会社 DeNA  
Presently with DeNA Co., Ltd.

<sup>f2</sup> 現在、日産車体株式会社  
Presently with Nissan Shatai

<sup>f3</sup> 現在、ACCESSPORT 株式会社  
Presently with ACCESSPORT

a) irie@is.uec.ac.jp

本論文の内容は 2012 年 7 月のマルチメディア、分散、協調とモバイル (DICOMO2012) シンポジウム 2012 にて報告され、ユビキタスコンピューティングシステム研究会主催により情報処理学会論文誌ジャーナルへの掲載が推薦された論文である。

に対して静止しているように見える“Body-Stabilized”および“World-Stabilized”な表示を実現している。この表示法を用いて、遠隔協調者のアバタをユーザのそばの空間に表示する環境や、仮想的な表示窓を空間に浮かべ、同じ部屋にいるユーザとのその仮想画面の位置を共有する環境が示されている。

部屋に「静止」した仮想オブジェクトを仮想窓越しに見る Body-Stabilized なユーザインタフェースは、HMD だけでなく、ハンドヘルドデバイスや小型プロジェクタなどにも応用されている。Yee は PDA 端末の小さなディスプレイを「窓」として、端末を動かすことにより、広い仮想空間を覗き見る環境を実現している [3]。また Cao らは小型プロジェクタの向きを変えることにより、サーチライトで照らすように仮想画面の一部を表示する環境を実現している [4]。

環境に仮想オブジェクトを静止させるだけでなく、ユーザが意のままに動かせる環境も提案されている。吉川らは、3次元位置と姿勢が識別可能なマーカを配置したデバイスを手に持って動かすことで、仮想オブジェクトが同期して動く環境を実現している [5]。このときに奥行き方向の表示を分かりやすく表現するために、仮想的に鏡を配置して距離感をつかみやすくする手法が提案されている。また Ha らは、手に持ったスマートフォンのタッチパネルとジャイロを使って仮想オブジェクトの表示位置を3次的に動かす手法、“AR-Wand”を提案している [6]。

## 2.2 HMD 環境とジェスチャ UI を組み合わせる試み

没入感が高い HMD 環境において、ユーザの自然な手の動きによって入力するジェスチャ UI は親和性が高く、数多くの研究が行われている。Kojima らはユーザの親指、人差し指、その股の3カ所にカラーマーカを付け、マーカの色検出座標をもとに仮想オブジェクトを表示することにより、物体をつかんでいるように見せるビデオシースルー環境を実現している [7]。また、Mistry らの“SixthSense”は、HMD ではなく小型プロジェクタでサーチライトのように仮想オブジェクトを表示する方式だが、同様に手にカラーマーカを用い、胸元のカメラで取得した画像から色検出することによりユーザの指先の動きを追跡し、ジェスチャとして解釈する [8]。

カラーマーカを用いる手法は指先位置の検出を軽量化する利点があるが、背景の色環境によってはかえって外乱が大きくなる。また、マーカを取り付けることなく素手でそのまま動かすことができれば利便性が高い。これらの利点から、マーカレスで指先を認識するジェスチャ UI の研究も行われている。蔵田らの“ハンドマウス”は EM 法により肌色領域を学習し、肌色領域と手の形状モデルとのマッチングによりクリックジェスチャを認識する [9]。同様に、Yao らの指先認識手法では肌色認識と形状モデルマッ

グにより2種類のジェスチャを認識する [10]。またこの研究では、光環境の変化に弱い、という色認識の欠点に対し、最初に明るさのレベル補正を行う手法が提案されている。より柔軟な指先座標検出を目指し、Störring らはモデルとのマッチングではなく形状の認識によって、突き出している指の本数とその位置を検出する手法を提案している [11]。この手法では肌色認識によって2値化した画像に対し、同心円上にトレースしながら肌色の連続領域の箇所を数えることによって指の本数と位置を推定する。さらに正確なアルゴリズムとして、Lee らの“HandyAR”では肌色領域の輪郭線を追跡して角度の鋭角な点を検出し、楕円フィッティングによって指先位置を推定する [12]。加茂らのビデオシースルー方式 AR 環境“AirSurface”では色と輪郭を用いたマーカレス指先認識により、両手で作った矩形ジェスチャの場所に仮想平面が形成される [13]。

また表示フィードバックを用いないジェスチャ UI として、運動記憶を用いたものが提案されている。Li らの“Virtual Shelf”ではユーザの周囲に見えない仮想棚を仮定し、その棚の各場所にプログラムへのショートカットが対応する [14]。ユーザが手を伸ばすことにより、棚の位置に対応するプログラムが動作する。伸ばした手の方向は周囲にカメラを置くことで計測される。ユーザに視覚フィードバックはないが、運動記憶により、意図どおりの場所へ手を伸ばすことができると報告されている。このような運動記憶の評価として、Gustafson らは、視覚フィードバックがないときに空間上に文字が正確にかけられるか評価しており、不可視な空間にも、ユーザは運動記憶により正確に描画ができると報告している [15]。

## 2.3 光学シースルー方式 HMD における重畳表示

光学シースルー方式を用いて現実世界に仮想オブジェクトを重畳表示する研究として、あらかじめ登録した平面上に、あたかも仮想オブジェクトが載っているかのように重畳表示する補正技術がいくつか提案されている。Janin らは、あらかじめ決まった位置にある平面に対して仮想オブジェクトを重畳表示するときの座標変換手法を提案している [16]。同様に、Oishi らの手法では CG を表示すべき位置とサイズの補正手法が提案されている [17]。これらの研究では HMD は固定であり、重畳表示する平面との位置関係が不変な場合の補正手法となっている。大島らは光学シースルー HMD 越しにエアホッケーの台を見ることにより仮想のパッド(球)が表示され、それを打ち合う“AR<sup>2</sup>ホッケー”を提案している [18]。これは現実の対戦相手を肉眼で認識できるなど、光学シースルー方式の利点を活かしたアプリケーションとなっている。AR<sup>2</sup>ホッケーでは台にマーカが埋め込まれており、その位置によって HMD 位置を推定し、パッドの表示位置を補正している。この手法によって、固定平面と、可動 HMD との間での表示補正

を可能としている．Hua らの手法では，現実のグリッド平面を利用して，その上に仮想オブジェクトを重畳表示するが，正確に投影するためにあらかじめ位置関係を登録する [19]．Lee らの手法では，視線入力を利用してマーカを凝視することにより，マーカのそばの平面に仮想オブジェクトが表示される [20]．この平面の正確な重畳表示のために Hua らの手法同様に，平面ごとに位置関係をあらかじめ登録する．また 3D 重畳表示について，Kellner らは 3D マーカデバイスを用いて奥行き方向の見え方の個人差を補正するキャリブレーション手法を提案している [21]．

### 3. AirTarget システムの提案

前節で見たように，光学シースルー方式はユーザの肉眼による視界が阻害されないため，移動中や体を使った作業中など，従来，コンピュータの利用が難しかった状況でも使えるようなプラットフォームとして利用できる．入力手段として，ユーザの指先によるジェスチャを用いれば目を離さずに使え，また特殊な入力機器を探して把持する手間を排除できる．ここで，ビデオシースルー方式で可能となっているような，指先認識によるジェスチャ入力や AR 表示が光学シースルー方式でも実現すれば，ユーザは画面越しではなく，肉眼で見える自分の手で，仮想オブジェクトを直接触って動かすことができる．本研究ではこのようなシステム，AirTarget を提案する．光学シースルー方式 HMD にカメラを取り付け，ユーザの視界を観測し，指先を認識してポインティングデバイスとして用いる．また簡単なジェスチャを定義し，指先がどのように動いたかによってクリック操作として解釈する．図 1 に提案システムの使用イメージを示す．

提案する AirTarget システムを入力として用いることにより，ユーザは入力デバイスを手に持たず，HMD アプリとより直観的なインタラクションが可能となる．たとえば，移動中に時刻や地図などの情報ウィンドウを選んで表示する，ウィンドウを直接手で触ってスクロールさせる，ジェスチャによって表示を消去する，などの操作が行える．また，現実空間に重畳表示した AR オブジェクトに直接触れて動かすゲームなど，新しいユーザ体験を提供できる．さらに，指先だけでなく，カメラがとらえた画像はすべて入力として用いることができる．たとえば，現実空間にある物体を指で指すことによって，その物体の情報を表示することや，紙に書いた文字を指差すことによる文字入力などの応用が可能である．これらのアプリケーションに限らず，ポイント，クリック動作ができれば，ソフトウェアキーボードのような HMD の画角に対して細かい解像度のタッチを要求されなければ，多くの従来アプリケーションを動作させることができる．

提案システム実現のためには，キャリブレーションと計算量の 2 つの課題が存在する．光学シースルー方式では，



図 1 AirTarget イメージ  
Fig. 1 Image of AirTarget.

肉眼の座標と CG 座標を重ねるため座標変換を行わなければならない．前章で紹介した関連研究では，あらかじめ決められた平面への座標変換は提案されているが，様々な位置をとる指に対して正確に CG を重畳する手法は提案されていない．また，マーカを用いずに指先を検出するアルゴリズムは計算負荷の高い画像処理となり，多くの関連研究で外部 PC を必要としているが，可搬性のためには，モバイル機器に備わるプロセッサ程度の計算力で高速動作する軽量なアルゴリズムが望ましく，二律背反の関係にある．

これらの課題について，本論文では以下の手法を提案する．まず，カメラ視界と肉眼視界のずれについて，一般には指と目の距離を測定しなければ補正は不可能である．しかし，ここで，ユーザが画面を指差すとき，指先はある一定の平面上を移動すると仮定すると，関連手法と同様に，HMD から特定の位置にある平面への座標変換とすることができる．具体的には，仮想画面の 4 隅を指差して登録することによりカメラ位置と仮想画面をマッチングさせる．関連研究では身体記憶の持続性が示されており，ユーザの操作時のクセや，登録時に動かした身体記憶により，このような簡単なマッチングでも，ランチャ操作，選択，ジェスチャ解析などに十分な位置精度を得ることが期待できる．

次に，HMD にカメラを取り付けて指や視界を観測する場合，指とカメラの位置関係が限定されることに着目する．カメラは目に近い部分にあり，仮想画面を触る指の向きは必ず指先が上となる．また，仮想画面をユーザが触っているときは，視界内で最も大きい肌色領域は，操作中の手となる．つまり，検出すべき指先座標は最大肌色領域の最大 Y 座標として取得できる．この検出法であれば，モデルマッチングや境界線探索を行うことなく，少ない計算量で実現できる．

さらに，日常生活で使用可能とするために，ロバストな肌色検出手法を提案する．従来手法では，肌色として HSV 空間のある領域を定義し，各画素がその範囲にあてはまるか判定していたが，我々は肌のハイライト部分では彩度の

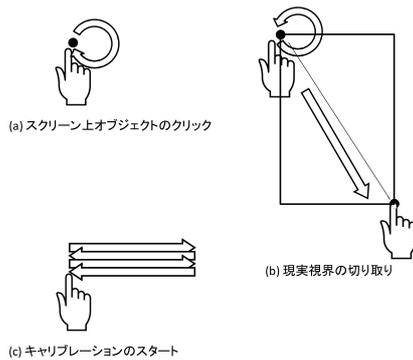


図 2 AirTarget のジェスチャ操作  
Fig. 2 Gesture command of AirTarget.

傾向が変わることに着目し、様々な光環境に対応する肌色検出アルゴリズムを用いる。

本研究で実装したジェスチャは、指先を小さく時計回りさせることによる仮想オブジェクトクリック操作、小さく反時計回りさせることによる現実オブジェクトの切り出し操作、またキャリブレーション登録操作も指先を左右に大きく2度振ることにより開始される (図 2)。

#### 4. 提案システムの実装

##### 4.1 システム構成

AirTarget のシステム構成を図 3 に示す。システムは光学シースルー方式 HMD、小型カメラ、画像解析用のプロセッサから構成される。カメラは視線と同軸に近い角度で HMD に固定され、ユーザの視界に近い視界を確保する。後述するようにキャリブレーションを行うので、厳密にカメラの設置を調整する必要はない。プロセッサはスマートフォンに組み込まれている CPU 程度の計算力があればよく、Android などに対応した HMD であれば、組み込まれているプロセッサによる単体動作が可能である。

システムで行う処理のブロック図を図 4 に示す。計算端末は、まず USB で接続されたカメラから新規フレームを取得する。取得されたフレームは指先検出アルゴリズムに渡され、カメラ画像のどの座標に指先があるかを判定する。得られた座標は履歴として直近 60 フレーム分が保持される。次に履歴を解析して指がどのようなジェスチャをしているかを判断し、インタラクトの内容を決定する。また、カメラから得られた指先座標を、目から仮想平面を通して見たときの指先の座標へ変換する。このようにしてジェスチャから得られた操作モードと変換された指先座標が、HMD で動作するアプリケーションへユーザ入力情報として渡される。以降では、各処理の詳細について示す。

##### 4.2 指先検出アルゴリズム

指先検出アルゴリズムでは、取得されたカメラ画像から指先の位置を検出する。アルゴリズムの流れを図 5 に示す。まず、取得された画像から肌色の領域を抽出する。一

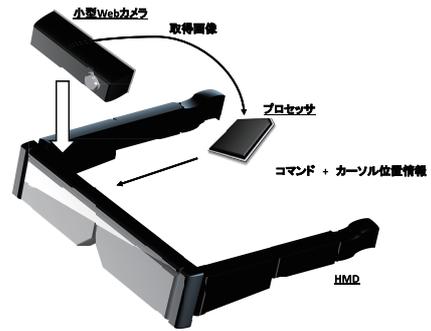


図 3 AirTarget のシステム構成  
Fig. 3 System component of AirTarget.

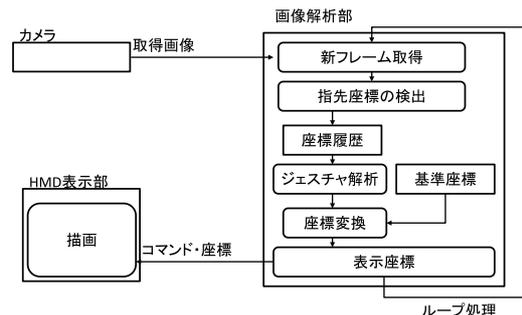


図 4 AirTarget の処理フロー  
Fig. 4 Processing of AirTarget.

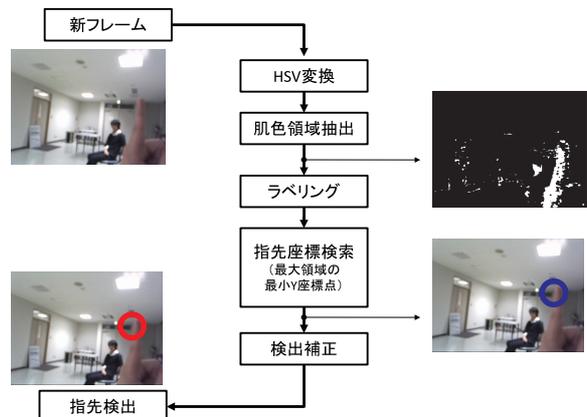


図 5 指先検出アルゴリズム  
Fig. 5 Finger detection algorithm.

般に関連研究で用いられている、色相、彩度、輝度に規定範囲を決めて色を検出する手法は、照明条件の影響を強く受けて検出精度を落とすことが知られている。我々は、照明条件に強い肌色検出として、肌のハイライトに着目する独自アルゴリズムを提案する。様々な照明条件のときに肌が見せる色相と彩度に着目すると、色相は大きく変化しないが、彩度は輝度によって傾向が変化することが観察される。すなわち、輝度の高いところではハイライトで光が反射し、彩度は低くなる傾向があり、一方、輝度が低いところでは肌の色が現れ、彩度が高くなる傾向がある。

この観察に基づき、AirTarget では以下のような肌色検出アルゴリズムを用いる。まず、色相は明るさによる変化

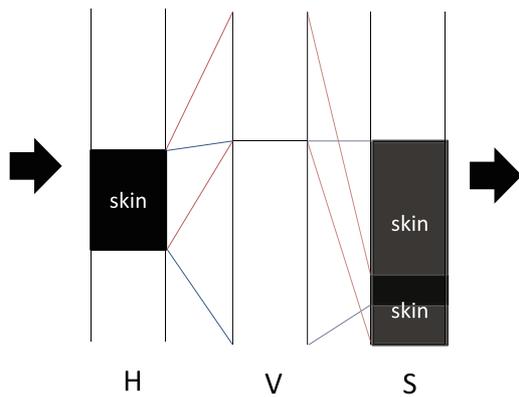


図 6 肌色検出アルゴリズム

Fig. 6 Skin color detection algorithm.

が少ないため、比較的小さい範囲を規定し、この範囲に入る場合のみを肌色候補とする。次に、輝度は照明条件次第で値が大きく変わるため、従来手法と異なり範囲を規定せず、代わりに輝度の値の高低によってその後の判定を変化させる。輝度値が低いときは彩度の肌色判定として高い彩度の領域を用い、輝度が高いときには低い彩度の領域を用いる。以上をまとめると図 6 のようになる。各画素について、このアルゴリズムを用いて判定を行い、肌色領域とそれ以外に 2 値化する。

肌色検出による 2 値化後では目標とする指先のほか、遠方の人の顔やノイズなども肌色領域として残っている。ここで、カメラ位置と手の位置が限定されることを利用する。指先によって仮想画面を指しているとき、視界内で最も大きい肌色領域は指先を含む領域となる。また、指先の座標は、その最大領域内で最も上に存在する点、つまり最も大きい Y 座標値を持つ点として検出できる。この方法であれば形状マッチングなどの処理を必要とせず、高速な検出が可能である。まずラベリング処理を行い、最も大きい連続領域のみを抽出する。ラベリングには井村によるラベリングクラス [22] を利用した。残された最大肌色領域に対して、画面左上から右下に向かって画素をスキャンし、初めに見つけた白色の座標を、そのフレームの指先位置の候補とする。この座標をアルゴリズム中では raw 座標と呼んでいる。

最後に検出補正を行い、ノイズによって偶然検出座標が大ききずれることを防ぐ。また、指先が視界内に存在しない場合でも raw 座標はノイズに基づいて何らかの値をとるため、肌色領域の大きさに応じて指の有無をフィルタリングする。次の条件、i) ラベリングされた最大領域の面積が規定値以上であること、ii) 直前のフレームで検出された raw 座標から一定距離以上離れていないことが満たされていた場合、この座標を新しい指先の位置として検出する。この確定した座標をアルゴリズムでは cursor 座標と呼んでいる。もし条件が満たされていない場合、cursor 座標は以前の値から更新されない。たとえば指先を視界内にいれた

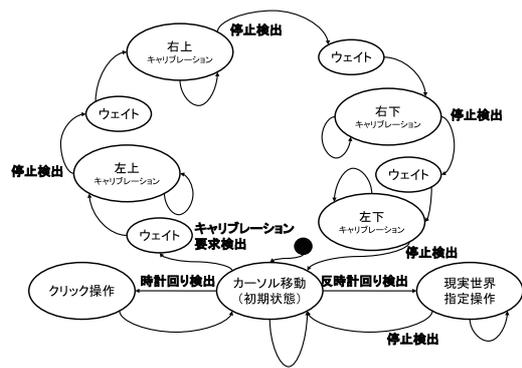


図 7 動作モードのステート遷移図

Fig. 7 State machine diagram of AirTarget.

場合、1 フレーム目の検出では領域の大きさの条件を満たすが、直前の raw 座標からの移動が大きく、cursor 座標は無効のままとなる。2 フレーム目の検出によって、双方の条件が満たされ、cursor 座標として有効となる。フレーム処理は評価で後述するように高速のため、ジェスチャやポインティングのための指の移動では ii) の条件から外れるほどの距離とはならない。

#### 4.3 ジェスチャ解読とステート遷移

検出された cursor 座標は履歴バッファに蓄えられ、ジェスチャ解読が行われる。履歴バッファは Circular FIFO で実装され、最大 60 フレーム分の検出結果を保持する。ジェスチャは履歴バッファをさかのぼって判定する。たとえば、9 時から始まる時計回りを検出するアルゴリズムでは、まず 6 時に相当する位置を探し、存在すれば 3 時を探し、さらに存在すれば 12 時を探し、というようにジェスチャを逆向きに探していき、保持フレーム内にすべてのジェスチャが含まれていれば、ジェスチャによるコマンドが成立する。コマンドが成立した場合は、履歴バッファをクリアする。

ジェスチャによってシステムがどのようなインタラクションを行うかのステート遷移図を図 7 に示す。特にインタラクションをとみなさないカーソル移動状態を初期状態として、大きく 3 通りの操作が実装されている。まず、左右に大きく 2 回指を振る動作によって開始されるキャリブレーション基準点の登録パスがある。キャリブレーションが始まると、左上→右上→右下→左下の順に仮想平面の四隅を登録する。HMD の該当箇所には四角形が表示され、ユーザがその四角形に指を合わせて停止させると基準点とカメラ検出座標との対応が取得される (図 8)。4 点の登録が終わると、システムはカーソル移動モードへと戻る。2 番目に、小さい時計回りジェスチャによって開始される、仮想平面へのドラッグ操作パスがある。時計回りジェスチャを認識するとそのときの座標に対してクリック操作が行われる。3 番目は、小さい反時計回りジェスチャによって開

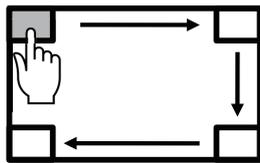


図 8 キャリブレーション基準点の登録操作

Fig. 8 Register operation of calibration parameter.

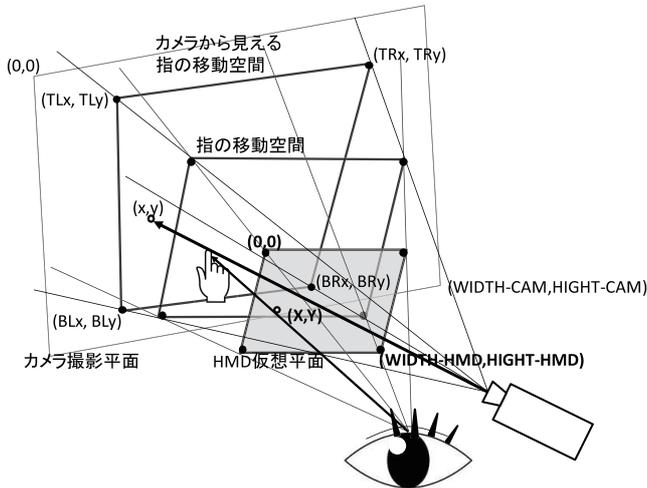


図 9 キャリブレーション

Fig. 9 Calibration.

始される，現実世界の指定パスである．反時計回りジェスチャを検出すると，そのときの座標を基点としてカメラ画像を切り出すことができる．基点から，次に指がしばらく停止される点を終点として，この線分を対角線とする矩形領域がカメラ画像から切り出され，保存される．この画像は写真メモのほか，検索クエリとして用いることができる．

#### 4.4 キャリブレーションアルゴリズム

光学シースルー方式 HMD を通して見た指にカーソルを正確に重畳表示することは難しい．カメラと視線とのずれがあることに加え，指の 3 次元位置によってずれは変化する．AirTarget では，指差しの姿勢について，同一人物が同じ場所を指す場合には，指先が同じような 3 次元位置に来ることを利用し，簡単な座標変換によりキャリブレーションを実現する．

キャリブレーションの様子を図 9 に示す．今，カメラ画像の中で  $(x,y)$  の位置で指先が検出された場合に，仮想平面に対応する  $(X,Y)$  を得ることがキャリブレーションの目的である．キャリブレーションのために，システムはまず，ユーザが仮想平面の 4 隅を指したときにカメラ画像の中でどの位置に指先が見つかるかを登録する．この 4 点の座標をそれぞれ，TL (左上)，TR (右上)，BL (左下)，BR (右下) と呼ぶ．指が仮想平面内の点を指している場合， $(x,y)$  は必ずこの 4 点で作られる四角形の内部に存在する．

変換後の座標  $(X,Y)$  は 4 点と  $(x,y)$  との相対的な位置関

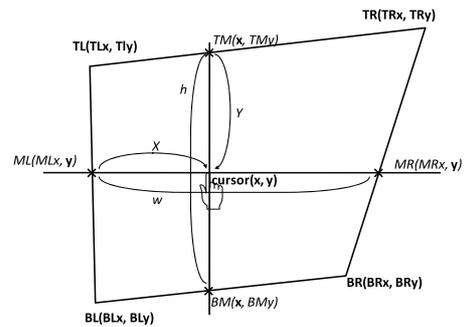


図 10 変換アルゴリズム

Fig. 10 Translation algorithm.

係から簡単な近似計算によって求める (図 10)．検出された座標  $cursor(x,y)$  から水平方向と垂直方向に補助線を引き，TL, TR, BR, BL が作る四角形との交点をそれぞれ TM, MR, BM, ML とする．X を求めるには ML (仮想平面では  $X=0$  に対応する点) と MR (仮想平面では  $X=画面幅$  に対応する点) で形成される線分を  $(x,y)$  がどの比率で内分しているかを求め，その比率を画面幅に対してかけることで近似される．Y も同様に，TM と BM によって形成される線を  $(x,y)$  がどの比率で内分しているかを求め，画面高さに対してかけることによって得られる．

#### 4.5 HMD への表示

画像検出部によって検出された指先座標やコマンドはプログラムへ入力情報として渡され，ユーザインタラクションを実現する．表示方法はプログラムの意図に従って様々だが，後述の実験では検出した  $cursor$  座標に対してカーソル表示を行っている．検出および座標変換が正しく動作すれば，ユーザの指先にカーソルが重なって表示される．HMD では視差を利用した立体視が可能だが，今回の実験では 2 次元表示を用いている．

### 5. 評価

#### 5.1 実機上でのカーソル表示の確認

小型カメラを取り付けた光学シースルー方式 HMD を用いて AirTarget を実装し，指に自然にカーソルが合うことの実証と，ジェスチャ認識精度の評価を行った．使用する HMD として EPSON MOVERIO BT-100 [23] と同様の Android 準拠 HMD を用いた．指先検出のアルゴリズムを OpenCV を用いて記述し，OpenCV for Android SDK フレームワークにより，HMD 上で動くアプリケーションとして実装した．

まず，提案キャリブレーションによって肉眼で見る指と仮想画面のカーソルの位置が一致することを確認した．図 11 (a)~(c) は AirTarget の動作画面を肉眼で見たときの写真である (後述するクリック精度評価のために，画面に 4 分割の線が描かれている)．様々な場所や光環境，また視界内に人物がいる場合でも，菱形のカーソル表示が正

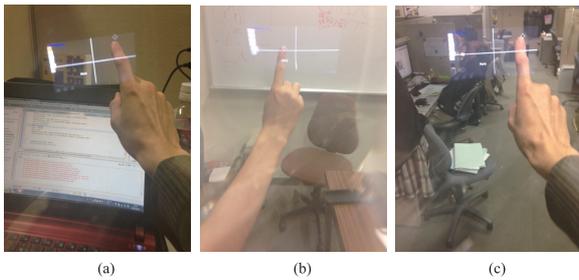


図 11 AirTraget 表示例  
Fig. 11 Sample views of AirTarget.

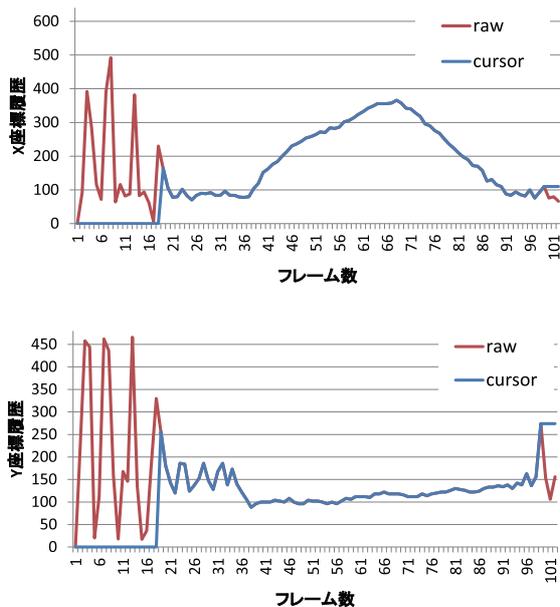


図 12 指先の検出結果  
Fig. 12 Result of finger detection.

確に行われている。

### 5.2 指先検出アルゴリズムの検出安定性および速度

図 12 はユーザが仮想画面内で指先をまっすぐ左→右→左と動かしたときに、アルゴリズムがどのような検出を行ったかを時系列で示したグラフである。グラフでは横軸に経過フレーム数、縦軸に検出位置の X 座標および Y 座標がそれぞれプロットされている。赤い線は肌色検出から推定される指先座標候補、raw 座標の値を、青い線は前フレームの検出位置や肌色領域の大きさから補正した推定指先座標、cursor 座標の値を示している

この実験は指を地面に水平な向きへ左→右→左と動かすものである。このとき、指先座標の X 値は少ない値から上昇し、再び下降する挙動を示す一方で、Y 値は変化が少なく一定であれば指先検出は正確に行われていることとなる。まず X 座標について見ると 36 フレーム目から 66 フレーム目にかけて値が上昇し、その後 91 フレーム目にかけて値が下降しており、正しい追跡となっている。座標の変化は指の動きを反映して不均質ながらもなだらかに推移

表 1 指先検出の処理時間

Table 1 Processing delay of finger detection.

カメラ画像取得時間	30.8 [ms]
肌色検出時間	9.25 [ms]
指先検出時間	14.7 [ms]
キャリブレーション時間	14.3 [us]
1 フレーム合計	54.8 [ms]

しており、指先を見失っていないことが分かる。この間、raw 座標と cursor 座標は一致している。

また、指が画面内に入る 20 フレーム近辺以前では、指が存在しないため、肌色検出は様々な領域をとらえ、raw 座標値が激しく動いている。しかし、大きさおよび位置補正により、アルゴリズムは指が存在しないことを正しく判断し、cursor 座標は無効値になっている。同様に Y 座標に注目すると、36 フレーム目から 91 フレーム目まで、指の水平移動を反映して、Y 座標はほとんど変化していない。X 座標のケースと同様に、指が入っていないフレームでは肌色検出は様々な値をとるが、アルゴリズムは指が存在しないことを正しく判断している。また、正確には、このユーザは指を水平移動させたつもりで、少しずつ指先が上に上がっていることが分かる。このように、AirTarget の指先検出アルゴリズムは指先の動きを正しくとらえている。

表 1 は図 12 の実験時の、HMD 端末におけるアルゴリズム処理時間を示したものである。指先処理の各ステップである肌色抽出、指先座標取得、キャリブレーション処理について、それぞれ全フレームの平均値を算出した。1 フレームの処理時間の約半分は I/O からカメラ画像を取得する時間であり、検出アルゴリズムの時間は軽量であることが分かる。特に、キャリブレーションは単純な線形変換のため、他のステップに比べると 3 桁速い処理時間となっている。1 フレームの合計では 54.8ms となり、フレームレートに換算すると 18 fps のスループットとなった。HMD 単体で動くアプリケーションとしては、十分なりリアルタイム性となっている。高速化について考えると、遅延の多くがカメラ画像の取得時間であることから、マルチスレッドによる実装最適化が効果的である。

### 5.3 クリックジェスチャの評価

次に、クリックジェスチャについて操作速度、認識率、クリック位置精度の評価を行った。HMD プログラムとして、大きめのアイコンをクリックするランチャあるいはアプリケーションを想定し、以下のような実験を行った。まず、各被験者はキャリブレーションジェスチャを入力し、各自の操作平面を登録する。その後、仮想画面を 4 つの領域にわけ、そのうちの 1 つの領域内に “here” の表示をする。被験者はプログラムから指示された領域を直接指さしてクリックジェスチャを行う。クリックジェスチャが認識



図 13 実験環境のカメライメージ  
Fig. 13 Camera view of experimental environments.

され、クリックされた座標が正解の領域に含まれていれば、クリック成功として、プログラムは他の領域に“here”を表示する。異なる領域へのクリックを認識した場合には、プログラムは誤クリックとしてカウントする。このようにして、クリックが4回成功するまでの時間とクリック精度を計測した。また、被験者には自分がジェスチャを行った数をカウントしてもらい、コンピュータが認識したクリック数と比較することで、クリック認識率を計測した。クリックジェスチャは図2に示したように、9時方向を始点とする時計回り1回転の軌跡とした。

実験は4人の被験者を対象に行った。被験者には実験前にクリックジェスチャを説明し、短い練習時間を設けた。日常生活の様々な場所での操作を評価するため、(a)スクリーン、(b)屋外、(c)暗所、(d)研究室、(e)研究室+人の5つの環境で実験を行った。カメラから見える各実験環境を図13に示す。(a)のような色変化の少ない環境に対して生活空間では肌色に近い色領域が出現し、単純な色認識では精度が落ちることが知られている。また色認識は光環境に影響を受けやすいため、(b)屋外や(c)暗所を実験環境に加えて色認識アルゴリズムの頑健さを評価した。また(e)では背景に人を立たせ、その肌色が指先認識に与える影響を評価している。

まず、ユーザがクリックジェスチャを出した回数に対するAirTargetのクリックジェスチャ認識回数の比を認識率として図14に示す。グラフでは被験者A-Dおよびその

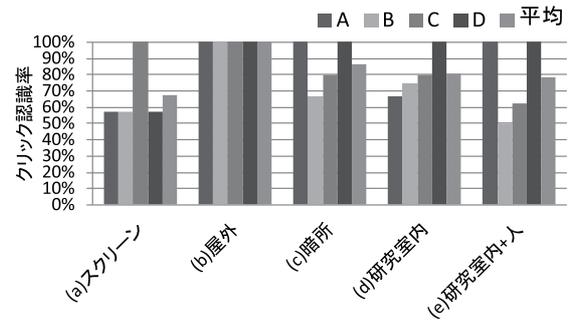


図 14 クリックジェスチャ認識率  
Fig. 14 Recognition rate of click gesture.

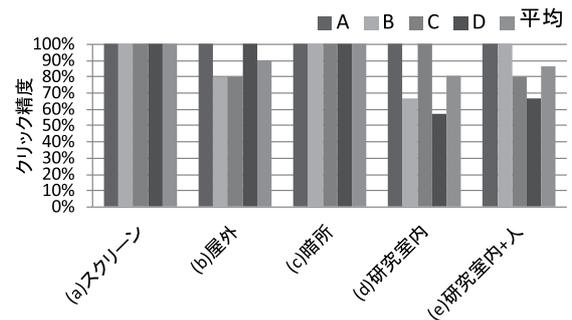


図 15 クリックジェスチャ位置精度  
Fig. 15 Positional accuracy of click gesture.

平均について各実験環境における認識率を示している。平均に着目すると、どのような環境でも良好なジェスチャ認識率を示し、78%から100%の間の値となっている。環境による差では、明るい(b)屋外では被験者全員が100%の認識率となる一方で、検出が難しいとされる(c)暗所でも87%という認識率となり、肌色検出の頑健さが示されている。一方(d)研究室や(e)研究室+人の環境では、高い認識率を示す被験者がいる一方で、低い認識率となった被験者もおり、背景の色配置によってジェスチャ認識が乱されるケースがあることが分かる。ただし、最悪ケースでも認識率50%を下回ることはなく、再試行の簡単なジェスチャであることから、ジェスチャインタフェースとして有効な認識率である。また、被験者Dの(a)スクリーンの認識率が低くなっているが、どのくらいの大きさで回せばクリックとなるかに慣れが必要との感想があった。被験者Dはその後、他の環境ではすべて100%の認識率を示しており、「1回転」のジェスチャには習熟を要することが分かる。この点について、ジェスチャの簡略化が今後の課題である。

次に、認識されたクリックについて、そのクリックが正しい領域内でされていたかを示したグラフが図15である。図14と同様に各被験者および平均について環境ごとに示している。平均に着目すると、クリック位置精度も80%から100%の高い値を示している。環境による差異では、認識率同様に背景に様々な色配置が現れる研究室内の(c)および(e)が他の環境に比べて低くなる結果となっている。ただし、被験者Aのようにすべての環境において100%の

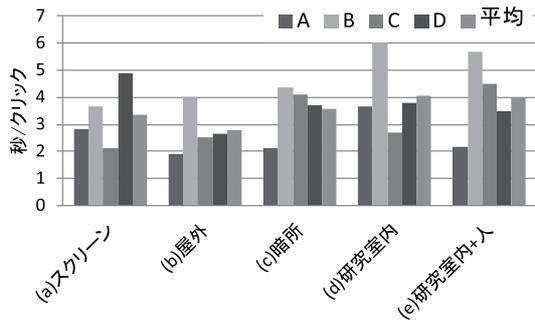


図 16 クリックジェスチャ入力時間  
Fig. 16 Time for a click gesture.

精度を得る被験者もおり、ジェスチャの習熟度が影響することを示している。被験者の感想では、ミスクリックの原因として、時計回りのジェスチャを9時方向から始め、円を描き終わったポイントがクリック座標となる、というクリック点のルールを把握していなかったケースがあげられた。

最後に、1クリックにかかった平均秒数を示したグラフが図 16 である。クリックする領域を示されてからクリックの認識が終わるまでの時間が示されており、ジェスチャ入力にかかる時間のほか、人間の反応速度が含まれた時間となっている。平均に着目すると、画像認識が容易とされる(a)環境から肌色の切り分けの難しい(e)環境になるに従い、クリックにかかる秒数が徐々に増えていくことが分かる。最も難しい(e)環境ではほぼ4秒程度の入力時間となっている。この値は、デスクトップ環境のマウス、キーボードなどによる入力速度に比べると遅くなるが、移動中や作業中に、入力デバイスを取り出すことなく、その場で自分の指による操作ができる時間として十分優位である。また(a)スクリーン環境や、被験者Aの全環境のように、スムーズなケースでは2秒程度の入力時間となっており、ノートPCのタッチ패드などと比べても遜色がない速度を実現している。

#### 5.4 キャリブレーション後の経過時間の影響

提案手法のキャリブレーションは画面の4隅を指定するが、これはユーザにとっての入力仮想平面を登録する作業となっており、Liら[14]の研究のように身体記憶を利用する性質を持っている。この身体記憶がどの程度持続するか、時間の影響を評価した。5.2節と同様の実験を、キャリブレーション1分後、5分後、10分後に行い、認識率、精度、時間の変化を評価した。実験環境としては(a)スクリーンを用いた。

結果を図 17 (認識率)、図 18 (位置精度)、図 19 (1クリックあたりの時間)に示す。認識率には時間による相関は見られない。これはジェスチャ操作自体はカメラ視界と肉眼のずれに影響されず、相対的な動きであることから予想される傾向である。一方で、位置精度はわずかに時間と

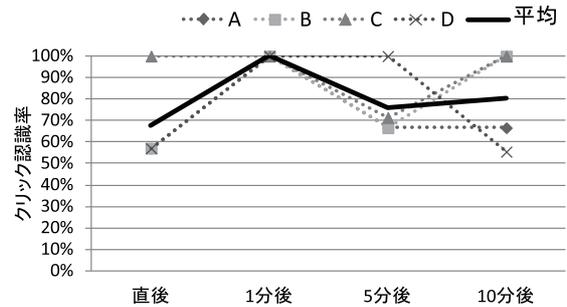


図 17 クリックジェスチャ認識率 (キャリブレーション後経過時間)  
Fig. 17 Recognition rate of click gesture (Time elapsed after calibration).

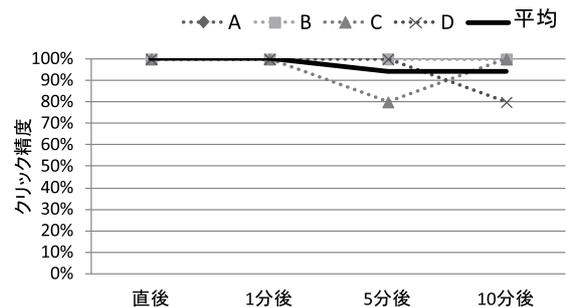


図 18 クリックジェスチャ位置精度 (キャリブレーション後経過時間)  
Fig. 18 Positional accuracy of click gesture (Time elapsed after calibration).

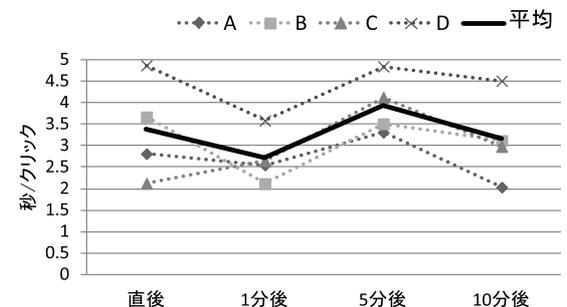
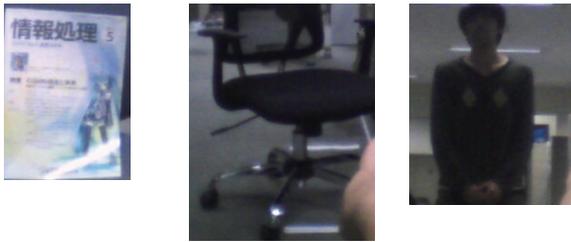


図 19 クリックジェスチャ入力時間 (キャリブレーション後経過時間)  
Fig. 19 Time for a click gesture (Time elapsed after calibration).

ともに低下が見られる。クリックあたりの時間では、キャリブレーション後10分では所要時間が平均で約50%増加している。このことから、身体記憶の持続時間がキャリブレーション精度に影響していることが分かる。10分後でも精度に大きな低下はなく、1回のキャリブレーション後の持続時間は、まとまった作業には十分である。なお、AirTargetではキャリブレーションはジェスチャによっていつでもスタートできるため、使用中にこまめなキャリブレーションが容易である。

#### 5.5 実空間指定操作

さらに、実空間を指定してユーザの視界を切り出す操作



80cm 先オブジェクト (会誌 : 21cm x 30cm) 取得画像 : 139 x 196	160cm 先オブジェクト (椅子 : 99cm x 66cm) 取得画像 : 204 x 264	240cm 先オブジェクト (上半身 : 85cm x 50cm) 取得画像 : 180 x 224
--	---	--

図 20 現実オブジェクト切り出し結果  
Fig. 20 Results of real object coutout.

の評価を行った。ユーザからそれぞれ 80 cm, 1 m60 cm, 2 m40 cm 離れたところに会誌 (21 cm × 30 cm), 椅子 (99 cm × 66 cm), 人物 (上半身 : 85 cm × 50 cm) を配置し, Air-Target の反時計回りジェスチャによる領域指定を用い, 視界の切り出しを行った。取得された画像を図 20 に示す。それぞれ切り出された画像の大きさはカメラ解像度 640 × 480 ピクセルに対し, 139 × 196, 204 × 264, 180 × 224 となった。現実視界の切り出し操作では, カメラと視線のずれの補正を行っていないが, 切り出された画像は, 何を切り出そうとしたか分かり, ジェスチャによりの確な境界線指定ができていない。椅子の例に見られるような見切れが生じる原因として, 切り出しの始点の指定から終点の指定の間まで時間があり, その間に顔を動かしてしまうと, 切り出し領域がずれてしまう点があげられる。ジェスチャの改良では, 始点と終点を一括して指定できるジェスチャが有効である。

## 6. おわりに

光学技術の進化やコンピュータの小型化により, 光学シースルー方式の HMD の可搬性が向上し, 日常生活への浸透が現実的となっている。光学シースルー方式は肉眼視界が維持されることから, 身体を動かしながらの使用が容易となるが, 一方でビデオシースルー方式のような正確な AR 位置合わせが難しいという問題点がある。また, 日常生活で利用するためには, ユーザの指先で直接仮想オブジェクトに触れるようなインタフェースが便利であるが, マーカを用いない検知アルゴリズムは従来外部計算機を必要とすることが多かった。

本研究では操作のための指先はある一定の平面を動くという性質を利用し, 光学シースルー方式のための簡単なキャリブレーション方式と, 指先を頑健かつ軽量に検出するアルゴリズムを提案し, 直接物体を指さして操作可能なシステム “AirTarget” を提案した。

HMD 単体上に実装したシステムを用いて評価を行い, アルゴリズム評価およびポインティング操作の評価を行っ

た。軽量化したアルゴリズムにより, マーカレス認識を含む計算処理は 18 fps 以上の十分なリアルタイム性を示し, 習熟の必要なくポインティング操作が可能であった。今後の課題として, 習熟に依存しない簡単なクリックジェスチャの検討や, 個人差に合わせた肌色閾値の自動調整, ステレオ視を利用してカーソルを指先と同じ立体位置に表示させ, さらに自然な重ね合わせ表示とすることがあげられる。

謝辞 本研究の一部は, 電気通信大学民間等共同研究 (セイコーエプソン株式会社) 「光学透過ヘッドマウントディスプレイにおけるジェスチャ入力の研究」による。

## 参考文献

- [1] Robinett, W. and Rolland, J.: A computational model for stereoscopic optics of a head-mounted display, *Presence: Teleoperators Virtual Environ*, Vol.1, No.1, pp.45–62, MIT press (1992).
- [2] Billingham, M. and Kato, H.: Collaborative Mixed Reality, *Int. Symp. on Mixed Reality*, pp.261–284 (1999).
- [3] Yee, K.: Peephole displays: Pen interaction on spatially aware handheld computers, *SIGCHI Conf. on Human Factors in Computing Systems*, pp.1–8 (2003).
- [4] Cao, X. and Balakrishnan, R.: Interacting with dynamically defined information spaces using a handheld projector and a pen, *Symp. on User Interface Software and Technology*, pp.225–234 (2006).
- [5] 吉川 誠, 亀井銀河, 岡田謙一: 遠隔 MR 協調作業における仮想鏡を用いた実物体と仮想物の重ね合わせ, マルチメディア, 分散, 協調とモバイルシンポジウム, pp.1215–1222 (2011).
- [6] Ha, T. and Woo, W.: ARWand: Phone-based 3D Object Manipulation in Augmented Reality Environment, *Int. Symp. on Ubiquitous Virtual Reality*, pp.44–47 (2011).
- [7] Kojima, Y., Yasumuro, Y., Sasaki, H., Kanaya, I., Oshiro, O., Kuroda, T., Manabe, Y. and Chihara, K.: Hand manipulation of virtual objects in wearable augmented reality, *Int. Conf. on Virtual Systems and Multimedia*, pp.463–469 (2001).
- [8] Mistry, P. and Maes, P.: SixthSense: A wearable gestural interface, *ACM SIGGRAPH ASIA 2009 Sketches*, p.11:1 (2009).
- [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, 信学技法, Vol.PRMU2000, No.156, pp.69–76 (2001).
- [10] Yao, Y., Zhu, M., Jiang, Y. and Lu, G.: A bare hand controlled AR map navigation system, *Int. Conf. on Systems, Man and Cybernetics*, Vol.3, pp.2635–2639 (2004).
- [11] Störring, M., Moeslund, T., Liu, Y. and Granum, E.: Computer vision-based gesture recognition for an augmented reality interface, *Int. Conf. on VISUALIZATION, IMAGING, AND IMAGE PROCESSING*, pp.766–771 (2004).
- [12] Lee, T. and Hollerer, T.: Handy AR: Markerless Inspection of Augmented Reality Objects Using Fingertip Tracking, *Int. Symp. on Wearable Computers*, pp.83–90 (2007).
- [13] 加茂浩之, 田中二郎: ウェアラブル拡張現実感による情報端末の仮想化, マルチメディア, 分散, 協調とモバイルシンポジウム, pp.1223–1233 (2011).

- [14] Li, F., Dearman, D. and Truong, K.: Virtual shelves: Interactions with orientation aware devices, *Symp. on User Interface Software and Technology*, pp.125-128 (2009).
- [15] Gustafson, S., Bierwirth, D. and Baudisch, P.: Imaginary interfaces: Spatial interaction with empty hands and without visual feedback, *Symp. on User Interface Software and Technology*, pp.3-12 (2010).
- [16] Janin, A., Mizell, D. and Caudell, T.: Calibration of head-mounted displays for augmented reality applications, *VRAIS*, pp.246-255 (1993).
- [17] Oishi, T. and Tachi, S.: Calibration method of visual parameters for see-through head-mounted display, *Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems*, pp.447-454 (1994).
- [18] 大島登志一, 佐藤清秀, 山本裕之, 田村秀行:  $AR^2$  ホッケー: 協調型複合現実感システムの実現, 日本バーチャルリアリティ学会論文誌, Vol.3, No.2, pp.55-60 (1998).
- [19] Hua, H., Gao, C. and Ahuja, N.: Calibration of an HMPD-Based Augmented Reality System, *IEEE Trans. Systems, Man and Cybernetics, Part A: Systems and Humans*, Vol.37, No.3, pp.416-430 (2007).
- [20] Lee, J., Prak, H., Lee, S., Kim, T. and Choi, J.: Design and Implementation of an Augmented Reality System Using Gaze Interaction, *Inf. Conf. on Information Science and Applications*, pp.1-8 (2011).
- [21] Kellner, F., Bolte, B., Bruder, G., Rautenberg, U., Steinicke, F., Lappe, M. and Koch, R.: Geometric Calibration of Head-Mounted Displays and its Effects on Distance Estimation, *IEEE Trans. Visualization and Computer Graphics*, Vol.18, No.4, pp.589-596 (2012).
- [22] 井村誠孝: ラベリングクラス, 入手先 (<http://oshiro.bpe.es.osaka-u.ac.jp/people/staff/imura/products/labeling>).
- [23] BT-100 仕様概要 (エプソン製品情報), 入手先 (<http://www.epson.jp/products/moverio/bt100/spec.htm>).

推薦文

この研究は、光学的シースルー HMD を用いた環境において、現実世界に対してハンドジェスチャを用いてさまざまな操作を実現している。単純なシステム提案にとどまらず、アルゴリズムの提案およびシステムの実装を行っており、完成度が高い。

(ユビキタスコンピューティングシステム研究会主査 椎尾一郎)



入江 英嗣 (正会員)

1999 年東京大学工学部電子情報工学科卒業。2004 年同大学大学院情報理工学系研究科電子情報学専攻博士課程修了。博士 (情報理工学)。2004 年科学技術振興機構 CREST 研究員, 2008 年東京大学大学院情報理工学系研究科助教。2010 年電気通信大学大学院情報システム学研究科准教授。計算機システムおよびその応用に興味を持ち, 特にプロセッサアーキテクチャ, デイペンダブルシステム, HCI の研究を進めている。情報処理学会山下記念研究賞 (2010 年), 同学会論文賞 (2010 年)。電子情報通信学会コンピュータシステム研究会幹事 (2011 年~)。電子情報通信学会, IEEE, ACM 各会員。



森田 光貴

電気通信大学電気通信学部にて HMD 向け UI の研究に従事。



岩崎 央

2013 年法政大学情報科学部コンピュータ科学科卒業。同年より電気通信大学大学院情報システム学研究科修士課程にて HMD 向け UI の研究に従事。



千電 航平

2013 年職業能力開発総合大学校長課程電子情報システム工学科卒業。同年より電気通信大学大学院情報システム学研究科修士課程にて HMD 向け UI の研究に従事。



放地 宏佳

2009 年富山商船高等専門学校情報工学科卒業。2011 年豊橋技術科学大学知識情報工学課程卒業。2013 年電気通信大学大学院情報システム学研究科修士課程修了。現在, 株式会社ディー・エヌ・エーにて開発者支援に従事。



小木 真人

2011年電気通信大学電気通信学部電子工学科卒業。2013年同大学大学院情報システム学研究科修士課程修了。



檜原 裕大

2011年電気通信大学電気通信学部量子・物質工学科卒業。2013年同大学大学院情報システム学研究科修士課程修了。現在、日産車体株式会社にて自動車開発に従事。ユビキタスコンピューティングシステム研究会学生奨励賞

(2011年)。



芝 星帆

2011年電気通信大学電気通信学部量子・物質工学科卒業。2013年同大学大学院情報システム学研究科修士課程修了。現在、ACCESSPORT株式会社にてシステム開発に従事。



眞島 一貴

2011年電気通信大学電気通信学部情報通信工学科卒業。2013年同大学大学院情報システム学研究科修士課程修了。



吉永 努 (正会員)

1986年宇都宮大学工学部情報工学科卒業。1988年同大学大学院修士課程修了。同年より宇都宮大学工学部助手。1997年から翌年にかけて電子技術総合研究所客員研究員。2002年電気通信大学大学院情報システム学研究科助教授。現在、教授。博士(工学)。計算機アーキテクチャ、並列分散処理等に興味を持つ。ACM, IEEE, 電子情報通信学会各会員。