

### ジェスチャ認識を用いた仮想空間上における 操作情報取得システムの構築

菅原, 甫史 / Sugawara, Masafumi

---

(発行年 / Year)

2011-03-24

(学位授与年月日 / Date of Granted)

2011-03-24

(学位名 / Degree Name)

修士(工学)

(学位授与機関 / Degree Grantor)

法政大学 (Hosei University)

# 2010 年度 修士論文

ジェスチャ認識を用いた仮想空間上における  
操作情報取得システムの構築

指導教授 赤松 茂 教授

法政大学大学院 工学研究科  
システム工学専攻

**09R6112**

スガワラ マサフミ  
菅原 甫史

## 概要

本研究では、マウスやキーボードのような既存のデバイスを使うことなく、人間のジェスチャだけで特定の操作を行えるシステムを構築した。

構築したシステムは、まず特定の操作情報が入った玉をユーザーに取り込む。その後、取り込んだ玉に割り当てられている操作がユーザーのジェスチャによって動作できるというものである。

ユーザーには接触型センサやマーカーのような特殊な装置を装着させる事なく、カメラによる画像処理で制御されたシステム構成となっている。

システムの実用例として、ブラウザの進む、戻る、更新といった操作を実現している。各々の玉の中にはブラウザの更新や進む、戻るといった情報が割り当てられている。そして、各々の玉に対応付けられたジェスチャを行う事により、本来はマウスで行う動作をユーザーのジ手でも動作できるようになっている。

システムの構築のほかに、仮想空間上の手のアイコンと実空間の手の動きの操作性に関する実験を行った。

キーワード：ユーザーインターフェース、仮想空間、ジェスチャ認識、ステレオビジョン

## **Abstract**

In this paper, without the use of existing devices such as mice and keyboards, has developed a particular operation system running on human gestures.

The developed system is import the user with a ball that contains information specific operation. After that the user can operate the gestures of operation assigned to the import ball.

Special equipment such as a user of the contact sensors or markers attached to it rather than be controlled by the configuration of the camera image processing.

As practical examples of the system, the browser forward, back, refresh and operation has been achieved. Each ball in browser refresh, forward of the browser, the browser's back has been assigned.

By gestures that are associated with each ball the user can work on the operation of the mouse hand.

In addition to building the system, carried out the experiment on the usability of hand motion in the real space and hand icon on the virtual space.

**Key Words** : User Interface, Virtual Space, gesture recognition, Stereo Vision

## 目次

第1章	はじめに	1
第2章	環境設備	3
第3章	画像処理ソフトウェア	
3-1	使用ソフトについて	4
3-2	画像処理ソフトウェア HALCON	4
第4章	操作情報取得システム	
4-1	システム概要	8
4-2	玉を取り込むまでのシステム操作	11
4-3	玉を取りこんでからのシステム操作	14
第5章	処理概要	
5-1	手の3次元位置の推定	15
5-1-1	三角測量の原理	17
5-1-2	カメラキャリブレーション	21
5-2	ジェスチャ形状の認識（ゲー形状・パー形状）	22
5-2-1	背景画像取得	23
5-2-2	背景差分	23
5-2-3	肌色抽出	25
5-2-4	形状特徴量の抽出	28
5-2-5	モフォロジー処理	29
5-2-6	真円度の抽出	30
5-2-7	探索範囲の設定	31
5-2-8	交差領域と最大内接円との差分領域の検出	32
5-3	ジェスチャ形状の認識（指形状）	33
5-3-1	検出された領域のレクタングル化	34
5-3-2	レクタングル化された領域の角度検出	34
5-4	実空間と仮想空間のキャリブレーション	35
5-4-1	重回帰分析	36
第6章	実験	
6-1	実験1	37
6-2	実験2	43
第7章	考察	
7-1	分散分析	45
7-2	実験1の考察	47

7-3 実験2の考察.....	48
第8章 むすび.....	49
謝辞.....	50
参考文献.....	51

## 第1章 はじめに

現在、普段の生活の中でのコンピュータとの接点は、大きく分けて公共空間での使用と個人的使用の二つに大別できる。公共空間で使用する代表的なものとしては切符などの券売機や駅の ATM などが挙げられる。これは、誰にでも使いやすいように設計されており操作にはタッチパネルを使用しているものがほとんどである。個人で使用するものとしては、最も代表的なものとして、PC や携帯電話が挙げられる。

PC の操作をする際はマウスやキーボード、携帯電話のメニュー操作などには十字キーや決定ボタンなどを用いている。しかし、人によってはこれらのインターフェースを使いこなすのに時間が掛かる事もある。その場合覚えなければいけない事柄が多く、敷居が低いとは言い難い面もあり、全ての人が使いやすいものとはいえない。最近では、携帯電話の操作にタッチパネルを搭載しているものも増えている。しかし、タッチパネルは指先での細かい操作がしにくい面があるため、意図しない選択の検出がある、キーボードほどの素早く正確な入力はできない、ボタンのような押した感覚がないため入力動作がぎこちなくなるなどの問題点もある。

これら既存の入力装置やその操作方式に取って代わるようなものとして、誰でも直感的に扱う事ができるインターフェースが望ましく、近年ではユーザーの生活環境内でも自然で直感的な操作を行う事ができるシステムの研究や、人間の身振り手振りや音声といった、人間同士のコミュニケーションに用いられる意思伝達手段を用いた PUI (Perceptual User Interface) システムの開発がされている。

それらのシステムは具体的には、カメラなどのセンサを用いる事により、コンピュータがユーザーの動作を認識し、適切な情報表示や支援を行う事ができるものが考えられている。動作には、表情や顔、手などを使用したものがあるが、特に身振り手振りやジェスチャのような手を使用したものは、ユーザーの意図を直感的に伝達する事ができる有効な手段として考えられているため、広く使用されている。

しかし、このようなシステムの大半は、磁気センサや加速度センサなどの接触型センサを取り付けたもの[1][2][3]や、赤外線カメラを使用したもの[4]などが多く提案されている。我が研究室でも、遠方の壁に投影されたカーソル位置とその指示コマンドを指差し動作によって遠隔制御するフィンガーポインターシステム[5]に取り組んできたが、指差しによる遠隔位置の指定の際に補助としてレーザーポインターを使用していた。

特殊な装置の着用や利用を求めるものは、高速でロバストな検出は可能だが、機材費用が高価であったり、設定場所の制約、着脱の繁雑さなどのユーザーへの負担の問題などにより、広く普及には至っていないのが現状である。

そんな中、任天堂の Wii や Microsoft の kinect などゲーム産業では、人間のジェスチャを利用した製品も開発され売り出されており、世間一般にも広く普及されている。また、形のない情報に直接触れる事ができるようにしたタンジブル・ユーザー・インターフェース (TUI: Tangible User Interface) のような新たな次世代インターフェースの提唱と、それらを利用した様々なインターフェースシステム[6][7]も提案されている。十数年前からジェスチャ認識を用いたシステムの研究はなされており、現在では上記の事例のようにゲームやインタラクティブアートの分野でのシステム開発などが多く目立ち、世間での認知度・期待度も上がってきている。これらの分野以外に、ユーザーの実生活にあるような身近なモノと組み合わせた実用的なシステム[8][9]の製品化を目指して開発を進めている企業や研究機関も多くなってきている。現段階では、ユーザーの生活環境に密着した実用的なシステムに関する目立った製品などの登場は無いのだが、今後更なる発展がなされていくと思われる。

そこで本研究では、上記であげた特殊な装置の利用やユーザーに接触型センサやマーカ一などを装着させる事なく、カラーカメラを使った画像処理による手形状ならびに 3 次元位置の推定、ジェスチャの認識を用いた PC の操作に着目したシステムを提案する。普段は既存の入力装置であるマウスやキーボードで PC の操作を行っているが、これらの操作の一部を人間の手で操作する、いわゆるハンドジェスチャでのシステム制御を目指した。なお、画像処理に関してのアルゴリズムの作成・検証は株式会社リンクスの HALCON9.0[10]を使用している。



## 第2章 環境設備

本システムの設備は、カメラの設置に伴いラックとポールを使っている以外は、ディスプレイと机が置いてある一般的な PC 環境化である。使用機材は以下のようになっている。

### ■ハードウェア

OS : Windows XP Professional SP3  
CPU : Intel Core i7 920 2.67GHz  
メモリ : 3GB  
カメラ : ace シリーズ acA640-100gc ×2

### ■ソフトウェア

画像処理 : HALCON (ver.9.0)  
システム実装 : Microsoft Visual C++2005 Express Edition  
Adobe Flash CS3

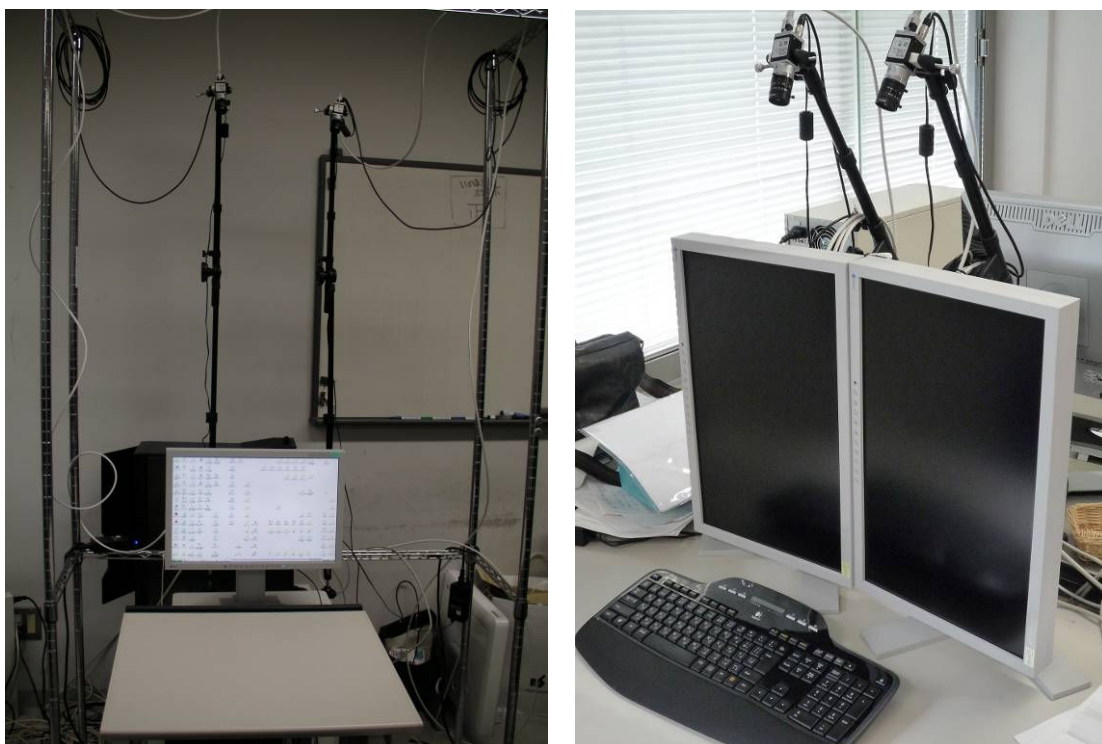


図1. 環境設備 (左：実験環境用 右：一般的な PC 環境用の一例)

## 第3章 画像処理ソフトウェア

### 3-1 使用ソフトについて

本研究では，システム全体のプログラムとは別に画像処理部分のプログラムを行っている．その際使用した画像処理ソフトウェアとして本研究では HALCON を用いている．HALCON は手形状の認識，手の3次元位置の推定，ジェスチャの認識で使われた．画像処理部分に関する詳しい処理概要については第6章で述べる．ここでは，HALCON の主な機能について説明をする[11][12]．

### 3-2 画像処理ソフトウェアHALCON

HALCON には，処理に関するオペレーターが用意されている．それらのオペレーターを作成したい処理の用途に応じて組み合わせていきプログラムを作成していくのが，HALCON の基本的な使い方になっている．処理によっては，数行のオペレーターだけで実現が可能となっている．

HALCON は画像処理に特化したソフトウェアなため，画像処理を行ううえで便利な機能も備わっている．例えば，Graphic Window に表示している画像を拡大することができるので，細かな領域の確認などに便利である．zooming という新たな window が開き，そこに拡大表示される仕様になっている．また，Graphic Window に表示している任意の画像に対して閾値処理が容易にできるので，閾値処理をする際の閾値の設定ができる．Gray Histogram という window にヒストグラムが表示され，バーを動かす事によって任意の閾値処理を確認できる．抽出した個々の領域のデータを見る事もできるので，ノイズ除去などの際に領域の特徴をつかむのに便利である．Region Features という window にデータが表示される．表示されるデータは重心座標や高さや幅，面積などである．これらの機能を使うことにより，処理段階での細かい設定などができるのが利点である．

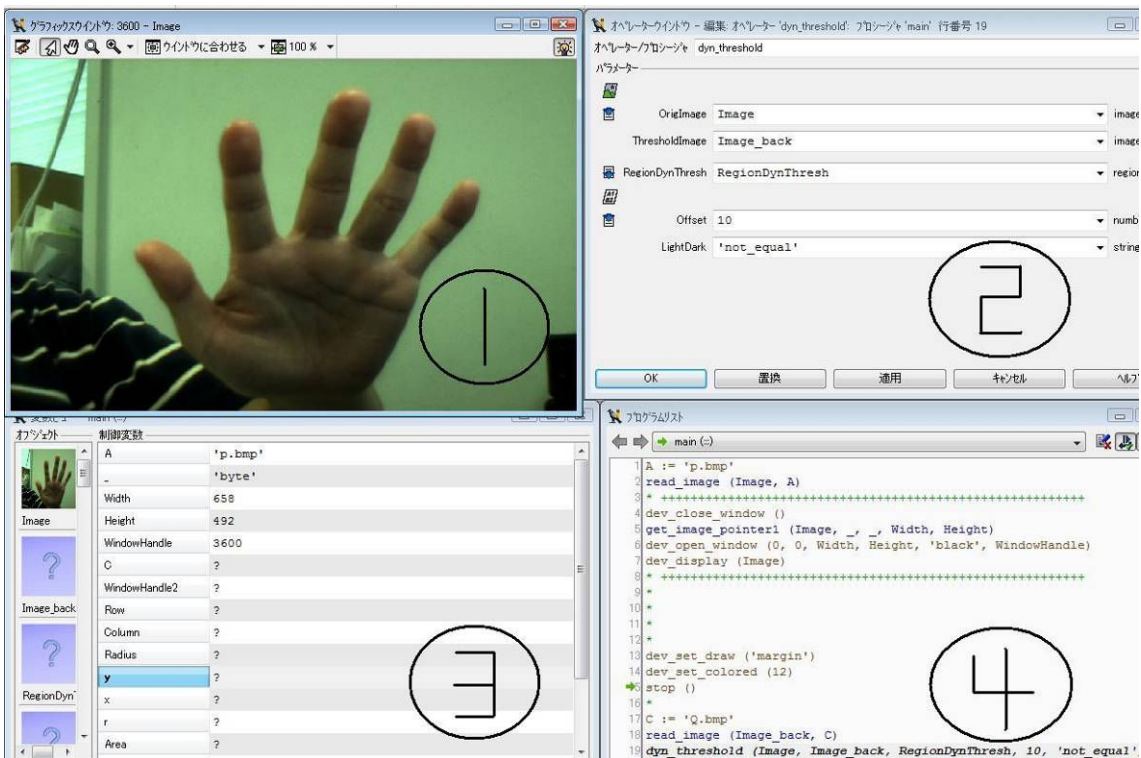


図 2. HALCON の window 画面

表 1. HALCON のサブウィンドウ

図2の番号	名称	用途
①	Graphic Window	処理結果の表示
②	Valuable Watch	処理の結果や数値の表示及び変更
③	Operator	オペレータの選択
④	Program	プログラムの表示

ソフトウェア自体は、表 1 のような 4 つのウィンドウで構成されている。この 4 つのウィンドウ上で作業を行っていくことになる。

Graphics Window にはプログラムを実行した結果の画像が表示される。プログラムの流れに沿った処理結果が画像に反映されて表示されていく。プログラム全体としてだけでなく、各オペレーターごとの実行結果も確認できる。

Valuable Watchではプログラム中で処理される検査対象および変数が管理されている。画像処理によって得られた数値結果や計算結果、領域の持つ値（面積値や重心の行列座標）等が確認できる。また、画像処理の途中結果（最終結果に至るまでの処理結果）を一括で表示している。計算内容や領域の値などの変更もここで自由にできる。任意の画像をダブルクリックすることによってGraphics Windowに表示する事もでき、処理途中の結果も確認できる。

OperatorではHALCONのライブラリに対応するオペレーターの編集作業を行える。また、各オペレーターに対して必要なパラメータなども設定することができる。HALCONには様々なオペレーターが用意されているが、実行したいオペレーターの一部を入力するだけでその文字に含まれるオペレーターが表示される。また、helpを使うことによりオペレーターの処理内容や、パラメータの推奨値等の情報を確認できるので、オペレーターの意味やパラメータの意味を調べる際には便利である。

Programではプログラムの編集作業を行う。Operatorで指定したオペレーターを表示し全体のプログラムとなる。入力したオペレーターの削除や新たなオペレーターの挿入もできる。また、プログラムを任意の部分まで実行する事も可能である。

オペレーターの詳しい内容はそれぞれのオペレーターのhelpに書いてある。さらにパラメータの設定や、各オペレーターの前処理、後処理に適切な候補オペレーターなども確認することができる。表2には、本研究で使用したオペレーターの一部をオペレーター名と共にその機能を示してある。なお、表2に載せたオペレーターの他にも、Graphic Windowに画像を表示させるためのオペレーターや、途中経過を表示させないオペレーターなどいくつか本システムでは使っているが、画像処理には直接関係ない部分のため省略している。

表2. 使用したオペレーターとその機能の一例

オペレータ名	機能
dyn_threshold	局所的な閾値を用いて画像を分割する
decompose3	RGB の1 つ1 つの要素を出力する
trans_from_rgb	RGB 空間から任意の色空間へと変換する
Threshold	任意の値で閾値処理をする
opening_circle	オープニング処理(雑音除去)を行う
closing_circle	クロージング処理(穴埋め)を行う
Inner_circle	領域の最大内接円を求める
gen_circle	中心座標と半径 を指定することで, 1 つもしくは複数の円を生成する
gen_rectangle1	座標軸に平行な矩形領域を生成する
intersection	2 つの領域の交差領域を計算する
difference	2 つの領域画像の差を求める
fast_threshold	与えられたグレイ値範囲内でグレイ値の高速選択をする
connection	領域の連結成分(面積や座標など)を求める
select_shape	形状特徴量に基づいて領域を選択する
shape_trans	領域の形状変換をする
orientation_region	領域の方向を求める
skeleton	領域の骨格を計算する
area_center	領域の面積と重心を求める
reduce_domain	画像のドメイン(解析範囲)を絞り込む

また, HALCON のプログラムは C 言語に用いられるような if 文や for 文などの一般的な制御文を使用することもできる. ある条件で HALCON のオペレーターを実行したり, 繰り返し実行したりする事が可能である.

また, HALCON 上で作成したプログラムを Visual Basic, Visual C++, Visual C#といった汎用コードに変換することもできる. HALCON のプログラムソースはボタン 1 つで, C++ や C#にコード変換できるので, 本研究のようにシステム全体を他のプログラミング言語で構築して, 画像処理の部分だけ HALCON で構築しているなどの場合でも容易に組み込むことが可能である.

## 第4章 操作情報取得システム

### 4-1 システム概要

操作情報取得システムについて概要を説明する。本システムは操作情報の入った玉を人に取り込むと、その取り込んだ玉に割り当てられている操作が人間のジェスチャによって動作できるシステムである。

本来はマウスやキーボードなどの入力装置でPCの操作を行う。これらは、操作によって動作が異なったりボタンの配置も決まっていたりする。例えば、キーボードにはショートカットキーというのがありボタン1つで操作ができるが、どの操作がどのボタンに対応しているかを覚えていないと使えない。これに対し、本システムでは難しい操作などは無く、実行したい操作情報の玉を掴む、離す、ジェスチャを行うという直感的な動作で制御ができる。また、カメラの撮影範囲内であれば、ある程度自由度のきく範囲内での操作が可能となる。

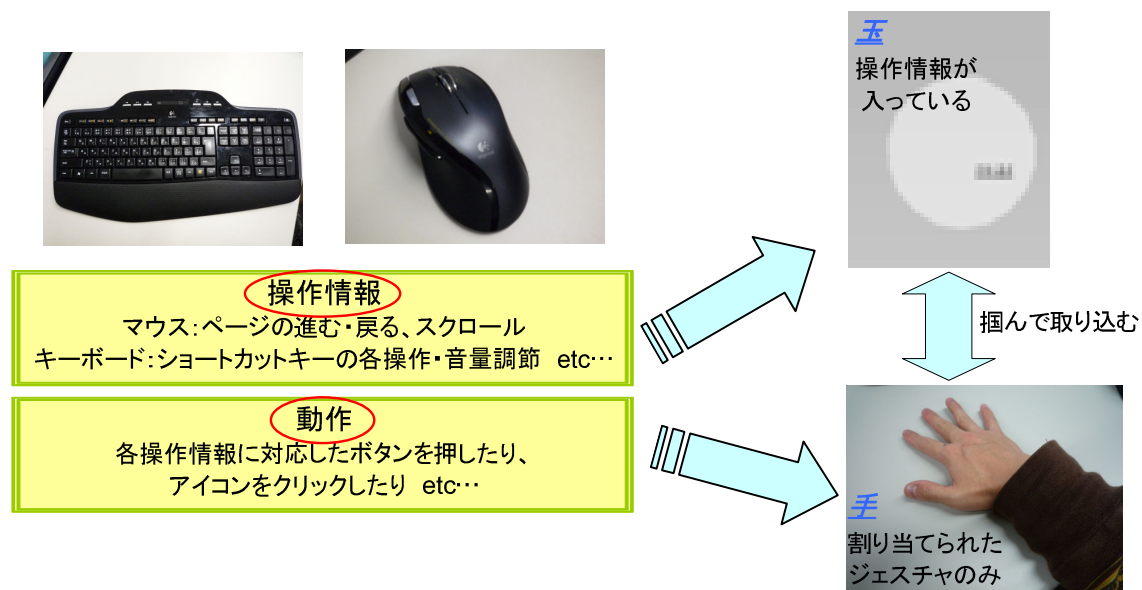


図3. システム概要

ディスプレイには図 4 のような 3 次元仮想空間が映し出されている。仮想空間は 600×600×600 ピクセルの 3 次元仮想空間である。

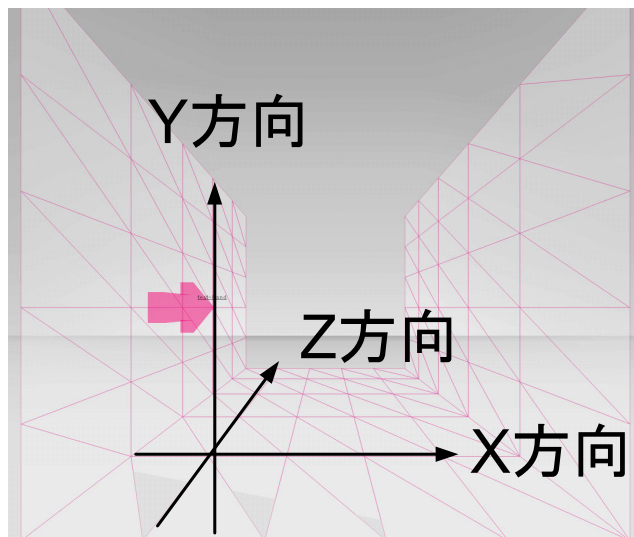


図 4. 3 次元仮想空間

仮想空間上には、特定の操作情報が入っている玉と手のアイコンが配置されている。仮想空間上であれば、玉は自由に配置することも可能である。また、玉と操作情報の対応付けもこちらで自由に変えることが可能となっている。仮想空間上の手のアイコンは、実空間の手の動きと連動して動くようになっている。

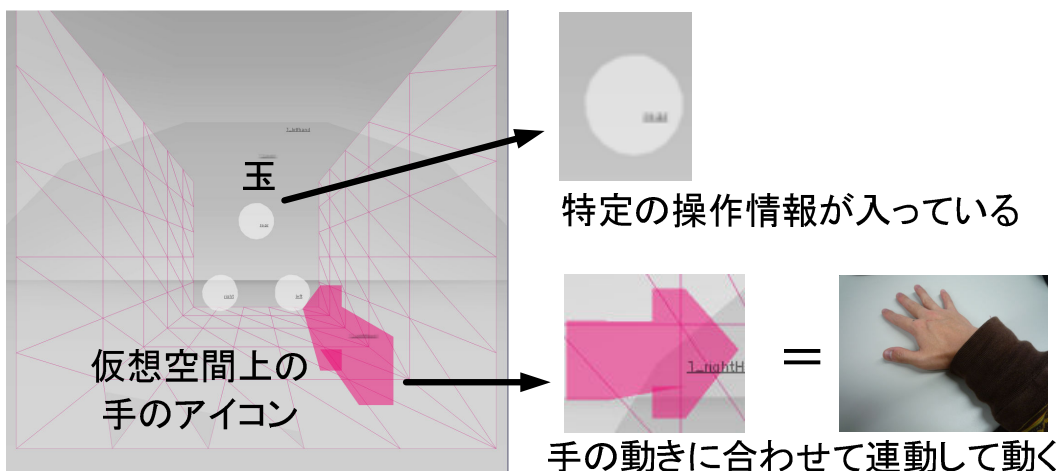


図 5. 仮想空間上の配置物



システムの操作手順としては以下のようにになっている。ユーザーは図 6 のようにディスプレイを見ながら、仮想空間上に浮かんでいる操作情報の入った玉を手で掴み・取り込むという動作をカメラに認識させる。これが認識された段階で、各操作情報に対応付けられたジェスチャを行う事で玉に割り当てられている操作が反映されることになる。手のアイコンは実際にユーザーの手の動きに付随して動くようになっているので、ユーザーはディスプレイを見ながらアイコンを操作することになる。

本システムの基本操作の部分でもある、玉を掴む・取り込むという操作部分のジェスチャであるが、玉を掴む動作は手をグー形状、玉の中の操作情報を取り込む動作はグー形状を 3 秒間保持、掴んだ玉を離す動作は手をパー形状にすることでそれぞれの操作ができるようになっている。

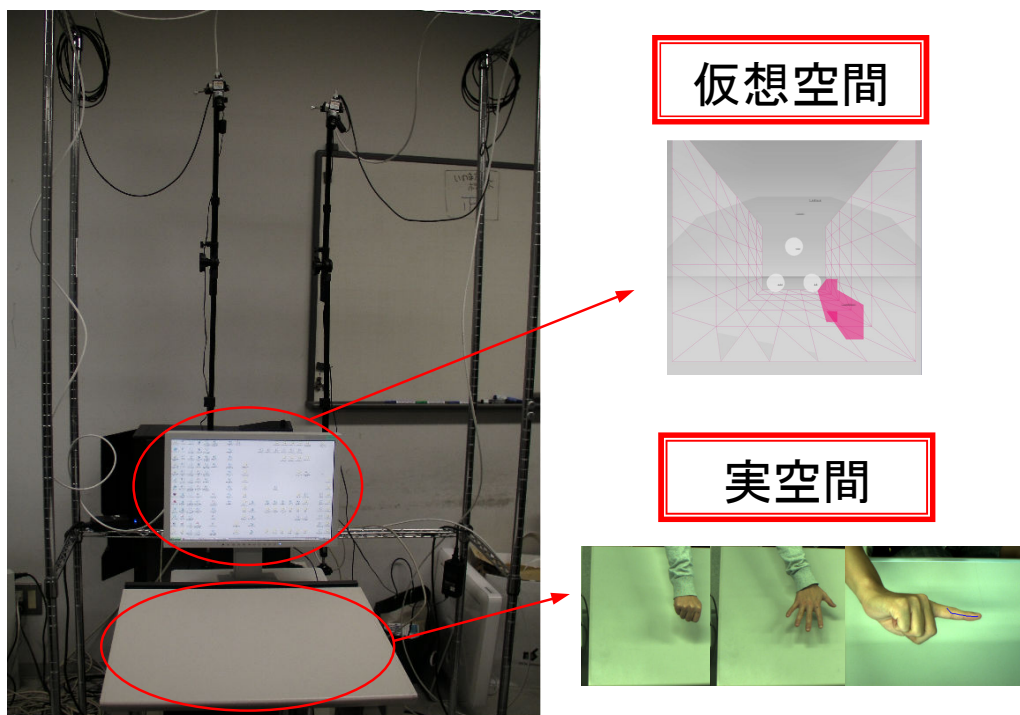


図6. 仮想空間と実空間



## 4-2 玉を取り込むまでのシステム操作

4-1で述べたシステムの操作手順を、「仮想空間」と「実空間」に分けて細かく説明していく。

図7は仮想空間、実空間ともに何も無い状態である。この場合、仮想空間には操作情報の入った玉が浮かんでいるだけの状態となっている。また、システムを初めて起動する際は、何も写りこんでいない実空間の作業領域を撮影する必要がある。この詳しい処理に関しては、次の5章で説明する。

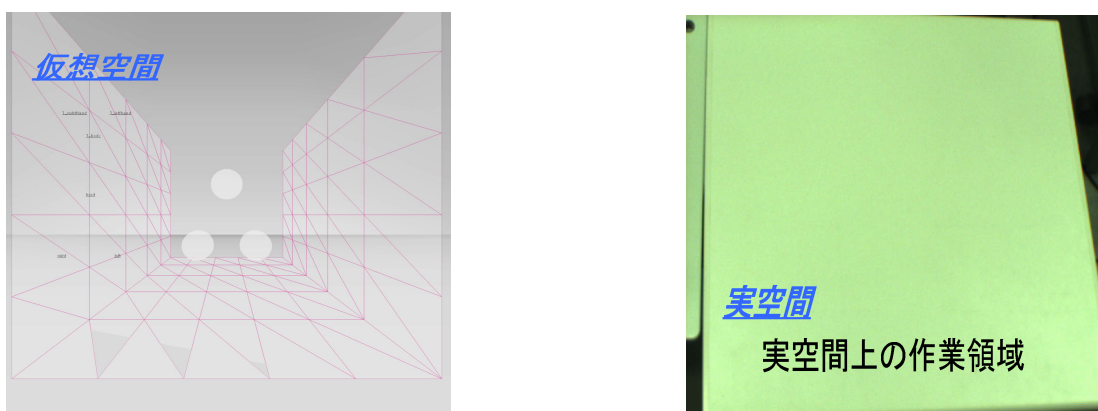


図7. 仮想空間と実空間の状態①

図8は、ユーザーが実空間に手を入れている時の状態である。実空間に手が入ると仮想空間には手のアイコンが出現し、実空間の手の動きと連動して動く。



図8. 仮想空間と実空間の状態②

図9は、手のアイコンと玉が触れている状態である。実空間で玉のある位置まで手を動かす。仮想空間上の手のアイコンが玉に触れると、玉とアイコンが振動状態になる。

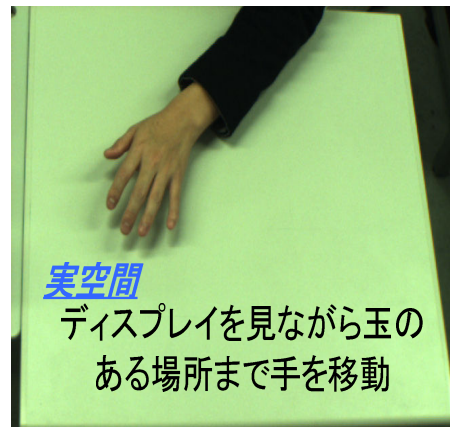
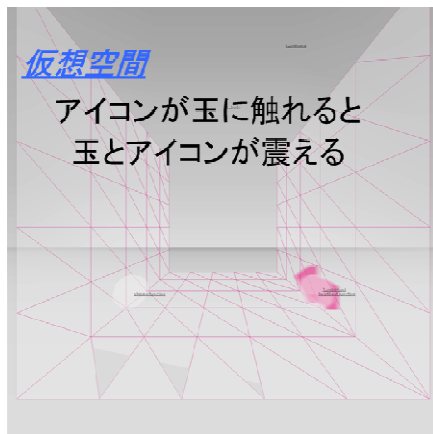


図9. 仮想空間と実空間の状態③

図10は、玉を取り込んでいる最中の状態である。図9のように玉と手のアイコンが振動した後に、実空間上で手をグー形状にするとアイコンと玉が重なって、玉のほうは本来の色から赤へと変化します。



図10. 仮想空間と実空間の状態④

図11は、玉の取り込みが完了した状態である。図10のように玉を取り込んでいる状態から、実空間でゲー形状を3秒間保持することで玉の取り込みが完了される。この後は、玉の中の操作情報と連動したジェスチャを行っていくことになる。それについては4-3で説明する。



図11. 仮想空間と実空間の状態⑥

図12は、玉の取り込みを解除した状態である。図10のように玉を取り込んでいる状態から、途中で玉の取り込みを解除する際には、実空間で手をパー形状にすることで、玉の取り込みを解除することが可能である。図11のように玉が取り込まれた状態から、玉の取り込みを解除する際には、実空間でパー形状を3秒間保持することで玉の取り込みを解除することが可能である。



図12. 仮想空間と実空間の状態⑦

### 4-3 玉を取りこんでからのシステム操作

本システムは玉の中の操作情報とユーザーのジェスチャを割り当てる事により、マウスやキーボードで操作できるものは、システムに組み込む事ができるようになっている。システムの実用例として本研究では、マウスのジェスチャにブラウザの更新や進む、戻るといった操作が割り振られているFirefoxのアドオン「FireGesures」のような機能を実現している。玉の中にはブラウザの更新や進む、戻るといった情報が入っており、これらに対応付けられたジェスチャを行う事により、本来はマウスで行う動作を人間のジェスチャで制御できるようになっている。今回の実用例では3操作を実装しているため、仮想空間上に玉は3個配置されているが、操作情報の玉は仮想空間上にはいくつでも配置することができ、玉の配置はユーザーの置きたいところへの自由な配置が可能となっている。

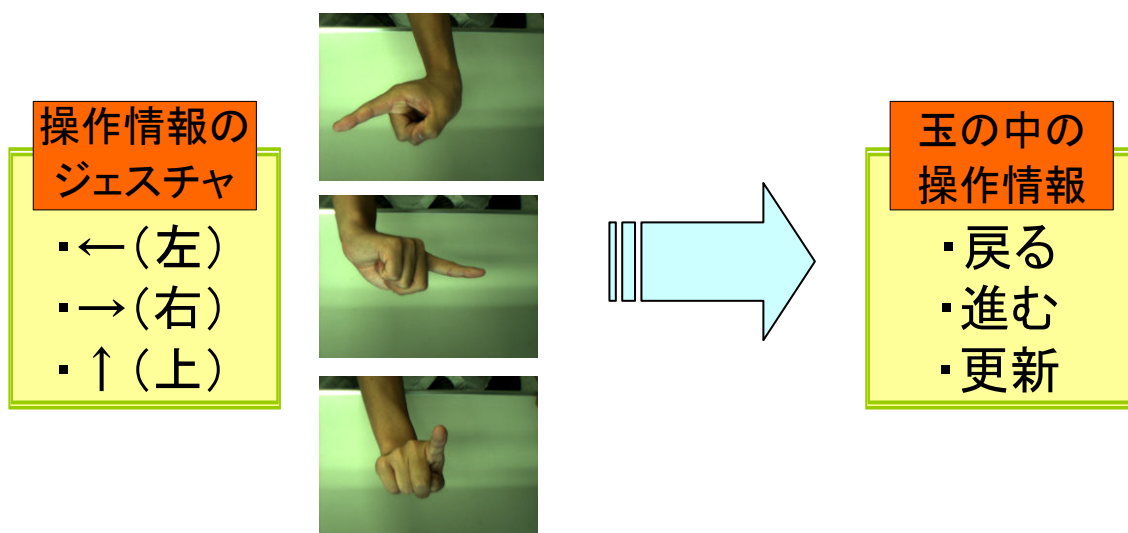


図13. システム操作の内容

## 第5章 処理概要

### 5-1 手の3次元位置の推定

手の3次元位置はステレオビジョンにより推定している。ステレオビジョンとは、三角測量の原理を用いて測定対象の3次元空間での位置座標を測定する手法である。異なる位置から測定対象を観測し、各視点での測定対象の投影位置の違い（視差）を求めると、三角測量の原理を用いて測定対象の3次元位置座標を得ることができる。図14のように左側のカメラと右側のカメラで手領域を撮影した入力画像中の手領域の見え方のズレ及び重心位置を元に、三角測量の原理に基づいて手の3次元位置を計算する。対象物の同じ位置の検出にはグレイ値マッチングを用いている。

世界座標系である3次元空間における実在の位置 $P$ は $P_L$ を延長した直線上のどこかに存在すると定義できる。そして、もう一方のカメラからの実在の位置 $P$ もサーチすると、 $P_R$ の直線上に存在する事になる。これにより、直線の交差する点を抽出する事により、3次元空間における世界座標系の位置 $P(x_p, y_p, z_p)$ を抽出する事ができる。

具体的には以下のように求める。カメラキャリブレーションで得られたカメラパラメータをもとに、各々の入力画像に対する投影中心 $P_L$ 、 $P_R$ が決定され、この $P_L$ 、 $P_R$ と計測点からなる三角形を利用して以下の式で求める事ができる。ただし、左右画像の点の座標は、それぞれの画像の中心を原点とする相対座標である。

$$x_p = \frac{d \cdot x_L}{x_L - x_R} \quad (1)$$

$$y_p = \frac{d \cdot y_L}{x_L - x_R} = \frac{d \cdot y_R}{x_L - x_R} \quad (2)$$

$$z_p = \frac{d \cdot f}{x_L - x_R} \quad (3)$$

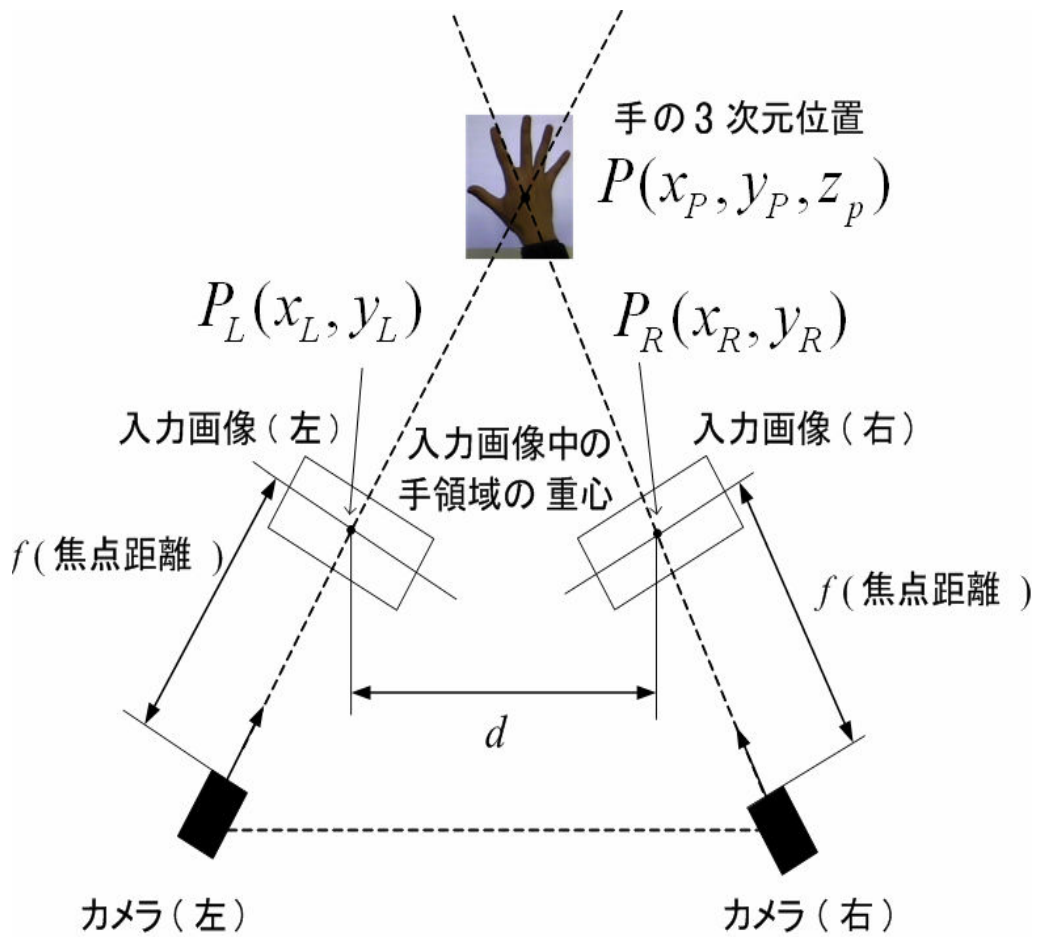


図14. 手の3次元位置推定

### 5-1-1 三角測量の原理

三角形の原理を使って離れた地点との距離を計測する手法が三角測量と呼ばれている。ある2点間の正確な距離が分かっている場合その2点から離れた場所のある地点との距離は、その2点との角度が分かれば、「三角形の一辺とその両端角が分かれば三角形が確定する」という性質により確定することができる。(図15)

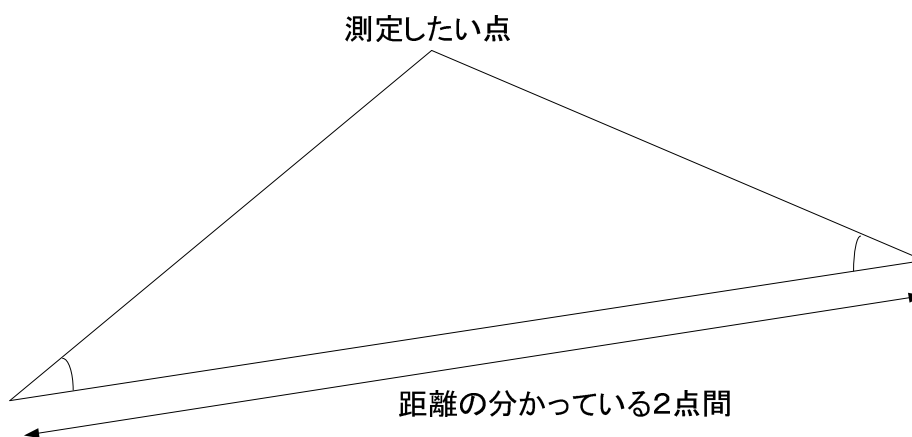


図15. 三角測量の概念図

例えば木の高さを測定したい時に木から少し離れた場所に立ち、その木との距離と測定場所から木のとっぺんを結んだ線と地表の角度が分かれば、木の高さを計算できる。これは三角測量の応用である。(木と地表の角度は90度)(図16)

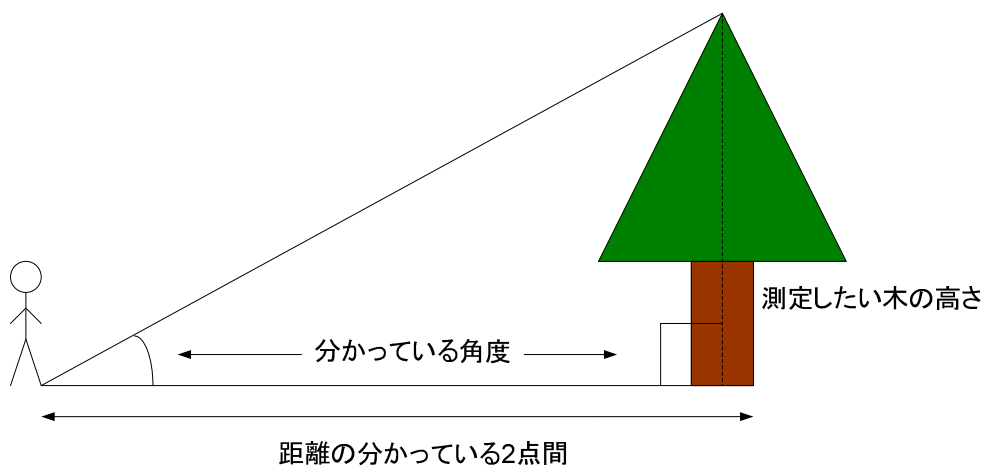


図16. 三角測量の応用例



この三角測量の原理を利用することにより、離れた場所にある物体の三次元座標を測定することが可能となる。

複数のカメラを使って 1 つの対象を異なる角度から撮影した場合、一定数以上の注目点がそれぞれにカメラに写っていて、その点の座標情報からカメラの位置関係を求めることができる。(図 17)

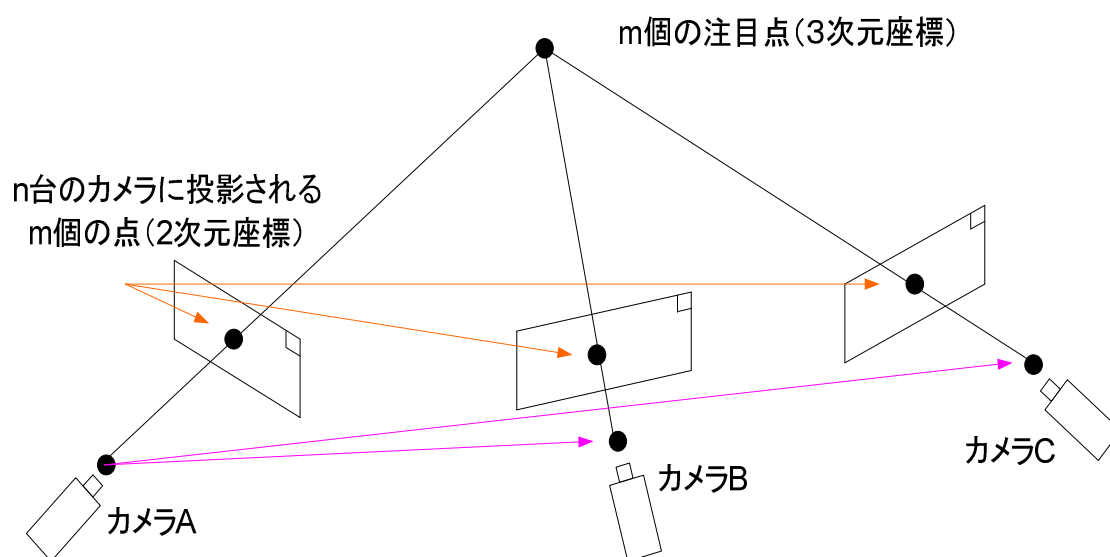


図 17. カメラ位置の特定図

カメラに写った物体上の点 P の座標  $p(x_p, y_p)$  は、

カメラの位置  $(X_c, Y_c, Z_c)$

カメラの方向  $(\omega_c, \phi_c, \kappa_c)$

点 P の位置  $(X_p, Y_p, Z_p)$

の関数として表せる。これを式にすると、

$$x_p = f(X_c, Y_c, Z_c, \omega_c, \phi_c, \kappa_c, X_p, Y_p, Z_p) \quad \dots(4)$$

$$y_p = g(X_c, Y_c, Z_c, \omega_c, \phi_c, \kappa_c, X_p, Y_p, Z_p) \quad \dots(5)$$

となる。



図 17 のように、点が  $m$  個、カメラが  $n$  台 (=写真  $n$  枚) の場合を考える。この時、 $n$  枚の写真上の  $m$  個の点について上記の式を  $2mn$  個たてることができる。また、未知の情報(未知数の数)は  $6n+3m$  個(カメラの位置+方向が  $6n$ , 点の位置が  $3m$ )となる。この時点では、座標系の取り方は任意のため、1つのカメラの位置と方向を基準座標とすることによって未知数を 6 個、さらに幾何学的にスケールが不定となるため、未知数はさらにもう 1 つ減らすことができる。よって、未知数は計  $6n+3m-7$  個となる。したがって、 $2mn$  個の連立方程式が解くためには、式の数は未知数の数より多ければよい。すなわち、

$$2mn \geq 6n + 3m - 7$$

を満たせばよいことになる。

写真が 2 枚 ( $n=2$ ) の場合で考えると、上記条件は

$$m \geq 5$$

となり、5 個以上の点が 2 枚の写真に共通に写っていれば理論的には連立方程式を解くことができ、カメラの位置関係が決定することができる。

これらを踏まえ2台のカメラを用いた場合を考える.2台のカメラを用いて物体を撮影し,それぞれのカメラの位置関係が求められているとすると,物体上のある注目点とカメラに投影される画像の関係は図18のようになる.

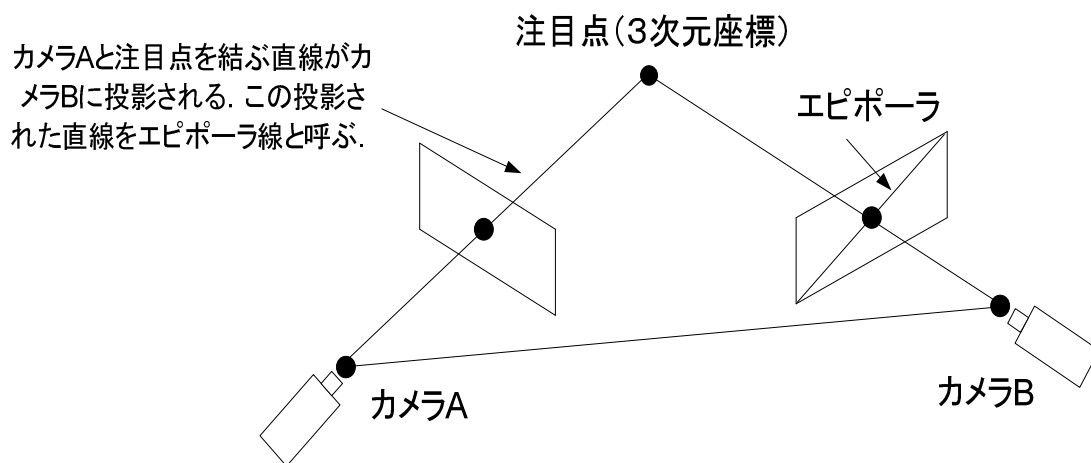


図18. 2台のカメラを使った3次元座標特定図

注目点がカメラAに写る点とそのレンズ中心を結ぶ直線と,カメラBに写る点とそのレンズ中心を結ぶ直線が交差する点が目的の注目座標として得られる.ただし,ここではカメラA,カメラB,注目点を結ぶ三角形の相対的な関係が求められるだけなので,絶対スケールを求める為には最低1カ所の実寸スケールが判明している必要がある.

カメラAと注目している点を結ぶ直線は,カメラBに直線として投影される.この直線をエピポーラ線と呼ぶ.カメラBと注目している点を結ぶ直線も,同様にカメラAに投影される.注目している点の座標が分からない場合は,カメラAに写っている点はカメラBではこの直線(エピポーラ線)上にあることが分かる.

## 5-1-2 カメラキャリブレーション

本研究のようにステレオビジョンを用いる場合、測定対象の各点への視線方向が分からなければいけない。そのために2台のカメラの相対位置を事前に取得する必要がある。そこで、カメラの内部パラメータと外部パラメータを取得する。

内部パラメータとは、カメラとレンズの組み合わせによって決まるパラメータのことで、例えばレンズの焦点距離、画素サイズの縦横比、カメラの角度係数、レンズの歪み係数などのことをいう。カメラ固有のものなので、カメラの設置場所や方向には依存しない。外部パラメータとは、撮影したカメラの3次元上での設置情報のことで、例えば撮影したカメラの被写体からの位置座標、上下の角度（チルト）、左右の角度（パン）、回転の角度（ロール）などのことをいう。カメラ固有のものではなく、カメラの設置場所や方向に依存する。

内部パラメータ、外部パラメータの校正法としては、形状が既知の基準物体に対して複数の視点から物体の位置や姿勢を変えて撮影し、得られた画像から推定する。基準物体に関しては正方形の格子パターンや円パターンなど様々なものが利用されているが、本研究では、図19のような正方形のキャリブレーションプレートを使用した。

本システムでは、キャリブレーションプレートを2台のカメラ両方に写るように設置して、角度および位置を変えて20枚程度撮影をする。そして、撮影したキャリブレーションプレート上のドット位置の計測を行い、その情報からカメラの内部パラメータ及び外部パラメータ、そして2台のカメラの相対位置を計測する。このカメラキャリブレーション作業は、システム環境を構築した際に一度行うだけで、システムを使うたびに毎回計測する必要はない。

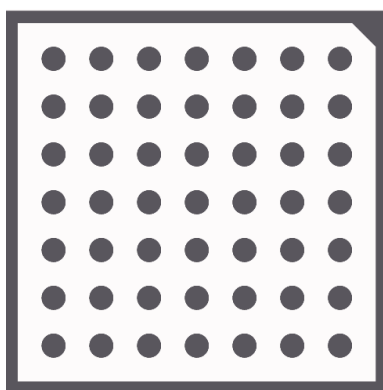


図19. キャリブレーションボード

## 5-2 ジェスチャ形状の認識（グー形状・パー形状）

ここでは仮想空間上の物体を，取り込んだり離したりするときに用いる，手のジェスチャ形状の認識について説明する．第3章で述べたようにHALCON画像処理ソフトを使用する．ジェスチャ形状を認識する処理の流れは，図20のようなフローチャートの通りである．各処理の説明は順次示す．

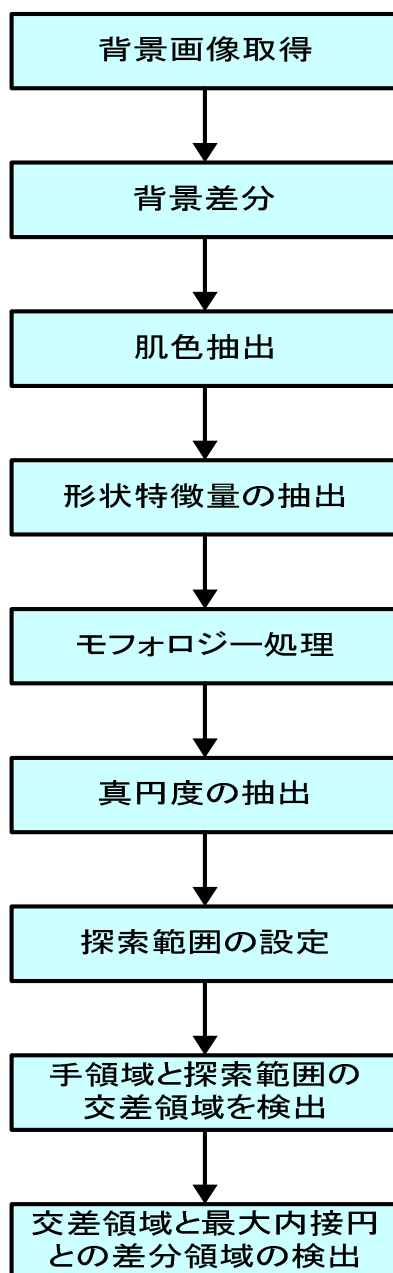


図20. 手形状認識のフローチャート

### 5-2-1 背景画像取得

システム初期準備として実空間上の作業領域の背景画像を取得する。システムの起動後は毎回、数秒間は作業領域に手やモノなどはない状態を保持する必要がある。その後はここで取得した背景画像を、実空間でのリアルタイムに取得される入力画像との背景差分の際に使用される。

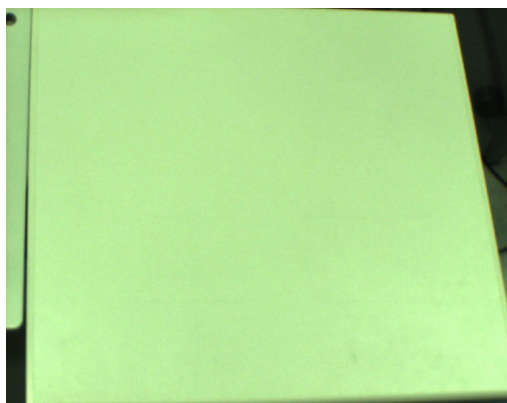


図21. 作業領域の背景画像

### 5-2-2 背景差分

まず背景差分を行う前に、リアルタイム化での作業領域の画像入力を行う。画像入力とは、カメラから得られた作業領域上を撮影した画像を取得するためのものである。このときに得られた画像を背景差分に使用する。そして画像入力で得られた画像と、システム初期準備で得られた背景画像を用いて背景差分を行う。ここで背景差分を用いる目的は、手の周辺の画像領域を絞り込むためである。

そもそも背景差分とは、あらかじめ背景画像（移動物体がない画像）を撮影しておき、入力画像（カメラで撮影した現在の画像）を撮影との差分処理（背景画像と入力画像の画素ごとの輝度値を比較し、その値が一致しなければその画素を移動物体領域とみなす処理）を施し、移動物体を検出する方法である。この方法は輝度変化によって対象部分以外も多少検出されるが、移動物体が存在すればその物体を一塊として検出できるため検出精度は高いと考えられる

図22のように，入力画像を取得する．

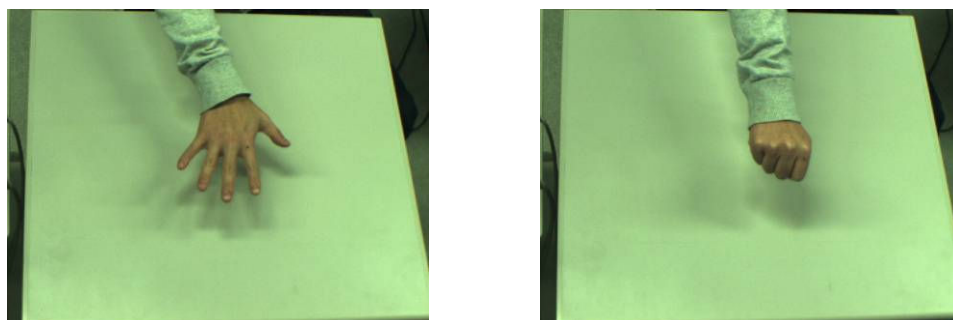


図22. 入力画像

図21の背景画像と，図22の入力画像を用いて背景差分を行う．これにより，画像領域が絞り込まれた差分画像（図23・24右端）が取得される．

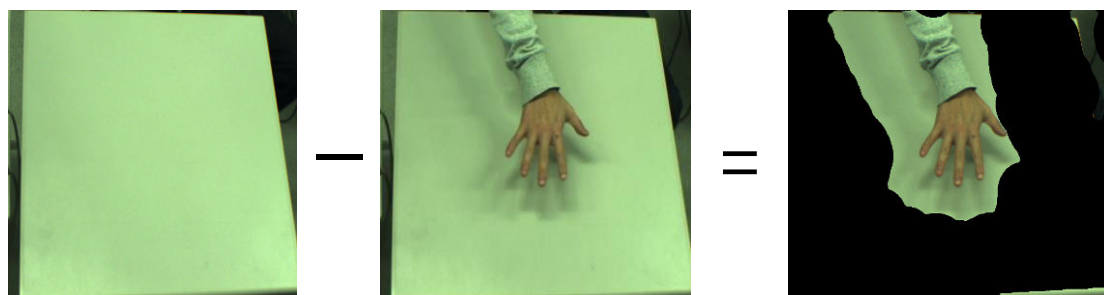


図23. 背景差分の結果①



図24. 背景差分の結果②

背景差分法はその性質から検出性が高い反面，いくつかの問題点もある．1つ目は，背景画像には移動物体が写りこんでいないものを使用しなければならない．そのため，移動物体が多く存在する区域において背景画像を取得する事は極めて難しくなるという点である．2つ目は，差分処理は各画素の輝度値の差を比較して移動物体の領域を検出しているので，光の変化が激しい屋外での検出は困難であるという点である．

### 5-2-3 肌色抽出

背景差分により絞り込まれた画像領域に対し、HSV 表色系を用いた肌色抽出を行う。そのため、画像を RGB のカラー空間から HSV のカラー空間に変換する必要がある。

HSV表色系とは、色相 (Hue) , 彩度 (Saturation) , 明度 (Value) により色を数値的に定義する表色系であり、オストワルト表色系に基づいた六角錐の表色系である (図25右)。HSIと呼ばれることもある。

オストワルト表色系とは、1920年頃ドイツのノーベル賞化学者オストワルトによって考案された表色系である (図25左)。何らかの原色を決めてこれらの混色量で色を表示する混色系の表色系のことで、理想的な白の白色量 (W)、理想的な黒の黒色量 (B)、理想的な純色の純色量 (F) を適当な面積比で回転混色することであらゆる色ができるというものである。

HSV表色系は、色相と明度が独立しているため、撮影した画像の明るさを考慮する必要がない。色相、彩度、明度の定義は以下のようにになっている。

- ・ 色相 … 色の種類を表す尺度。単位は角度で、0~360度の範囲である。
- ・ 彩度 … 色の鮮やかさを表す尺度。単位は%で、0%~100%の範囲である。  
色の彩度の低下につれて灰色さが顕著になり、くすんだ色が現れる。
- ・ 明度 … 色の明るさを表す尺度。単位は%で、0%~100%の範囲である。

HSV表色系はRGB 表色系と違い輝度が独立した単位として分離しているので、アルゴリズムを組む際輝度変化をあまり考慮しなくても済む。

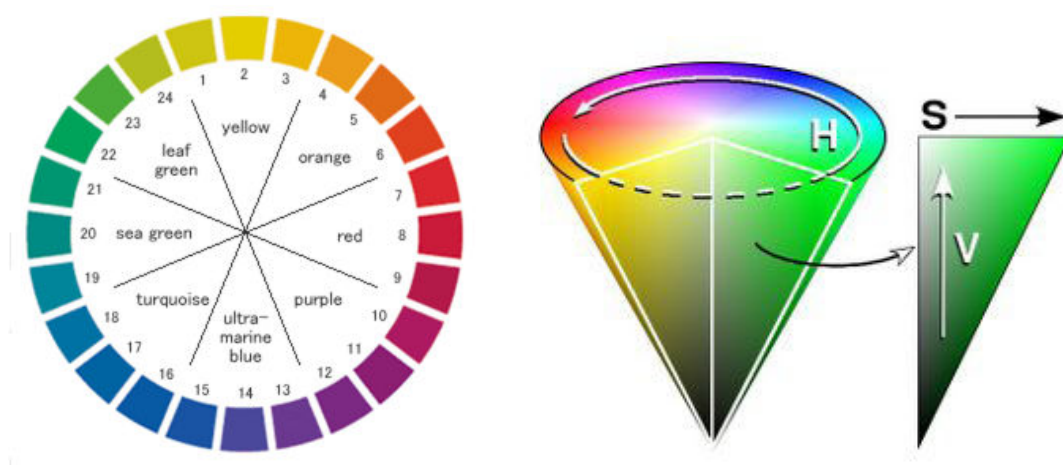


図25. オストワルト表色系 (左) , HSV色系 (右)

RGBからHSVへの変換式は以下のようになっている。

$\max = \max(R, G, B)$ ,  $\min = \min(R, G, B)$ とすると,

$$\begin{cases} H = 0 & (\max = \min \text{の場合}) \\ H = \left( \frac{G - B}{\max - \min} \right) \times 60 & (R = \max \text{の場合}) \\ H = \left( 2 + \frac{B - R}{\max - \min} \right) \times 60 & (G = \max \text{の場合}) \\ H = \left( 4 + \frac{R - G}{\max - \min} \right) \times 60 & (B = \max \text{の場合}) \end{cases} \quad (6)$$

$$\begin{cases} S = 0 & (\max = \min \text{の場合}) \\ S = \left( \frac{\max - \min}{\max} \right) & (\max \neq \min \text{の場合}) \end{cases}$$

$V = \max$

今回肌色領域を抽出するのに色相・彩度・明度の内、色相を用いることにした。これは人間の肌の色が人種にかかわらず、色相の0~30の閾値に収まるからである[13]。これを利用することで、肌色を容易に抽出することができる(図26)。

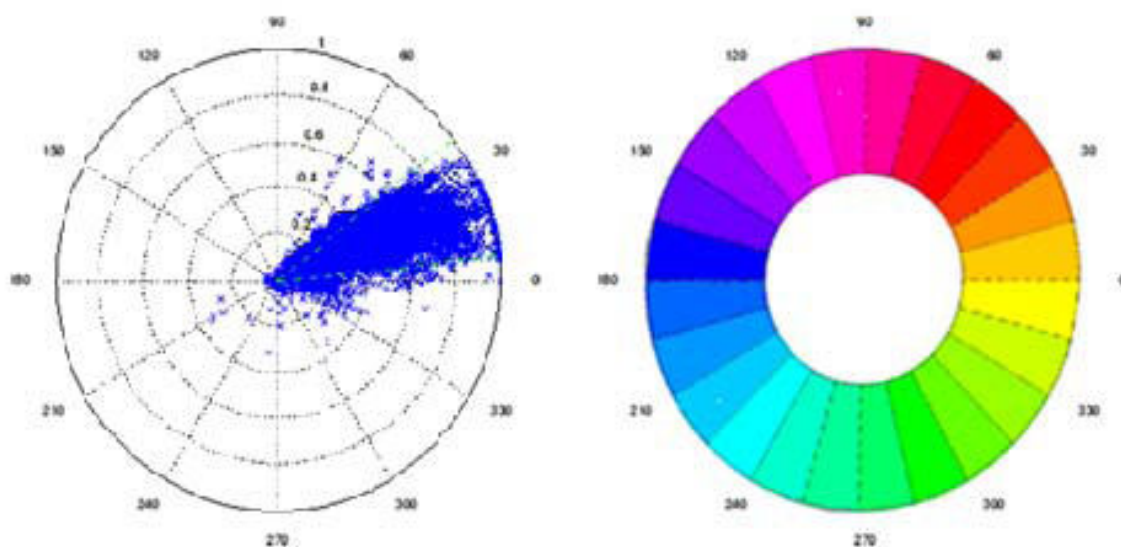


図26. 肌色のサンプルと色相関



RGBからHSVに変換すると図27のようになる。

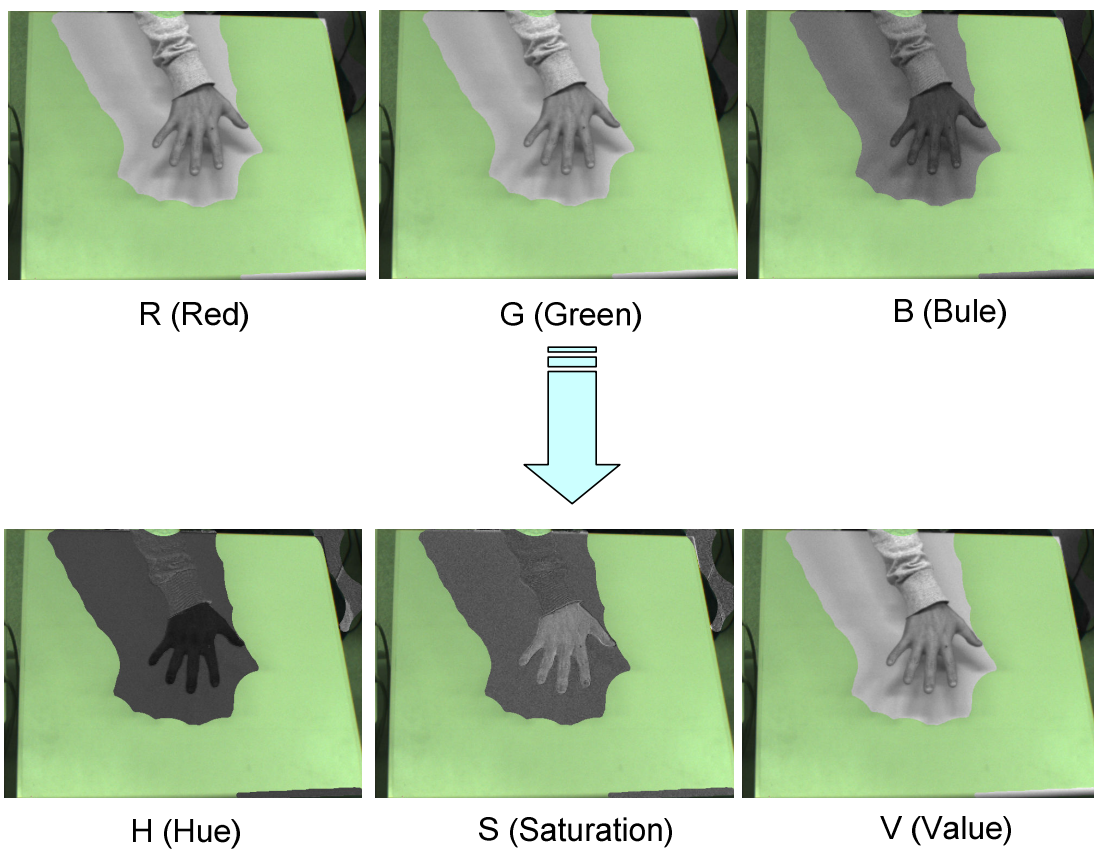


図27. RGBからHSVへの変換の結果

HSVに変換した後、色相（Hue）を用いて肌色領域を抽出した結果は図28のようになる。

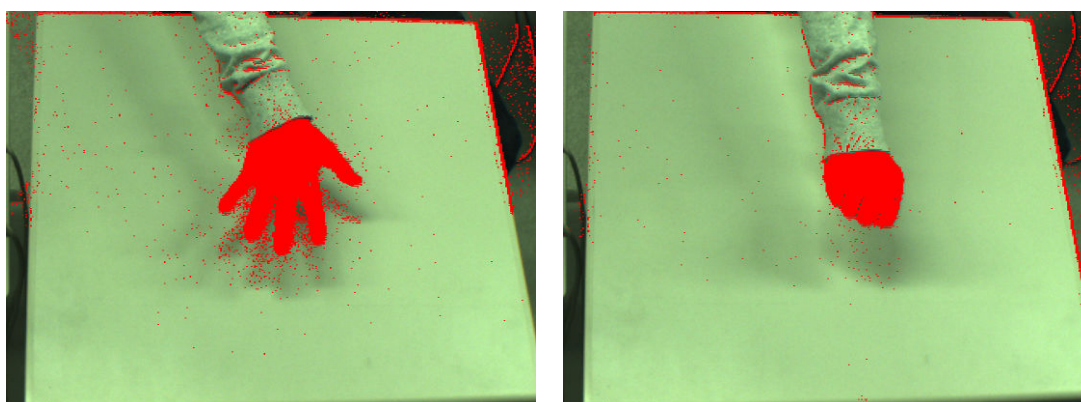


図28. 肌色抽出の結果

#### 5-2-4 形状特徴量の抽出

肌色抽出をしたら、次は手領域だけを抽出したい。そのために、形状特徴量に基づいて必要としている領域の抽出を行っている。本研究で用いた形状特徴量は領域の面積（画素数）を使用した。図29のように抽出された肌色領域のすべての領域の面積（画素数）が、こちらの指定した範囲内にある領域だけが抽出されるようになっている。これにより、手領域以外の細かいノイズはほぼ破棄して、手領域だけを抽出できる。

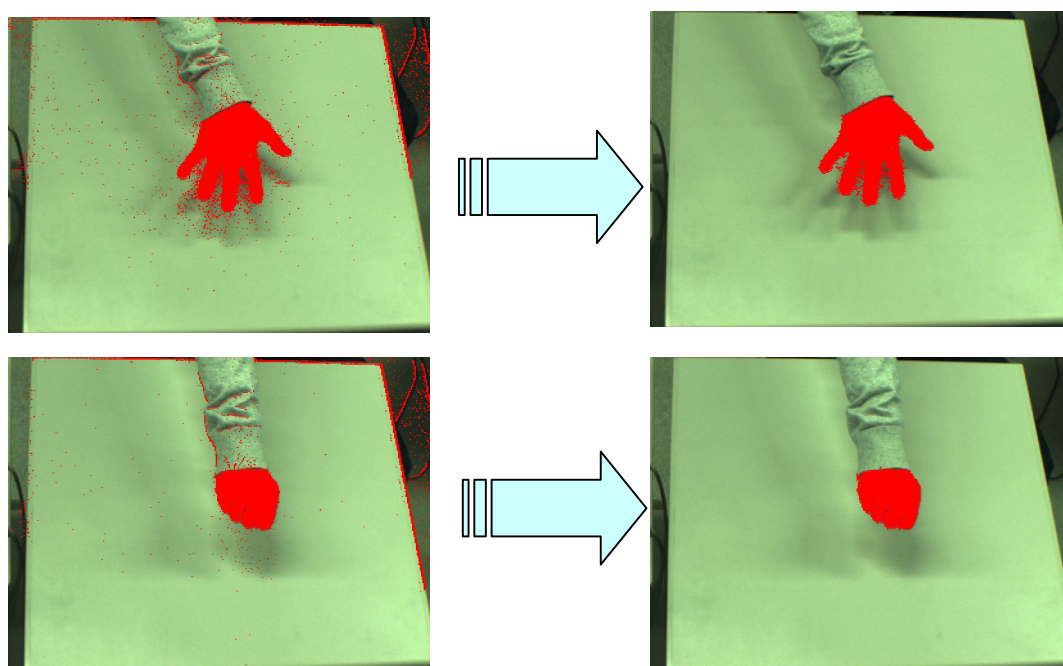


図29. 形状特徴量の抽出結果

### 5-2-5 モフォロジー処理

図28を見ても分かるように肌色抽出を行っただけだと、細かいノイズがのっていたり、抽出された領域が粗く次の処理の妨げになってしまったり、欲しい領域が1つにまとまっていないことがある。そこでモフォロジー処理を行い、領域を滑らかにして塊として手領域を抽出する。

モフォロジー処理とはn回の収縮とn回の膨張を繰り返し、領域を削ったり領域の穴を埋めたりする処理である。

図形を1画素分だけ細くする、あるいは太くする処理をそれぞれ図形の収縮・膨張という。図形の収縮は、1つの画素に対して上下左右に1画素ずつ収縮させ図形の境界点の値を0にする操作であり、図形の膨張は1つの画素に対して上下左右に1画素ずつ膨張させ、背景の境界点の値を1にする操作である。

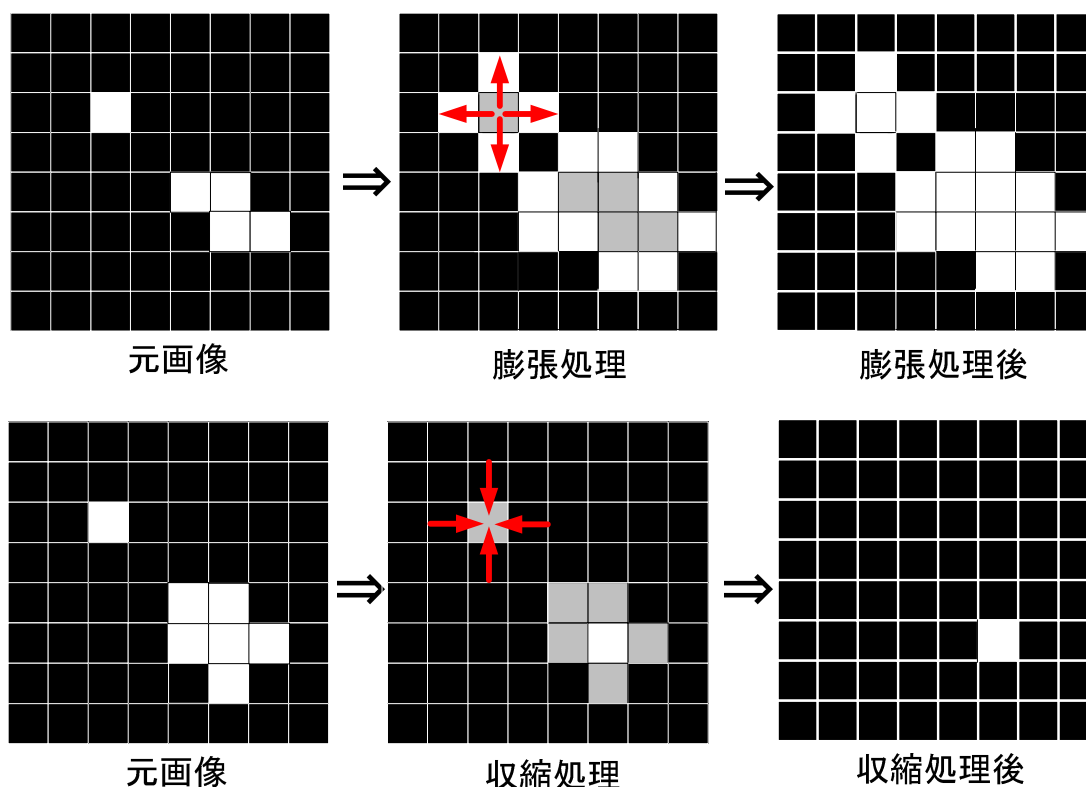


図30. 膨張・収縮処理

収縮の次に膨張を行う処理を「オープニング処理」といい、これは細かいノイズなど不要な部分を削ることができる処理である。そして、膨張の次に収縮を行う処理を「クローズング処理」といい、これは図形の穴を埋めたり切れている箇所をつなぐことができる処理である。

モフォロジー処理を施すと，図31のように綺麗に整えられた手領域が抽出できる。

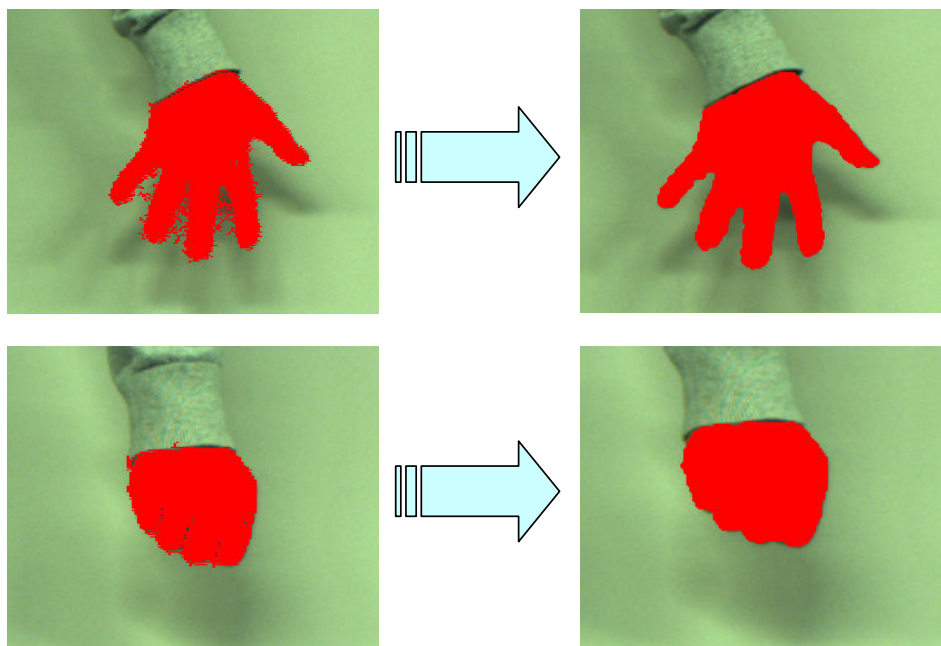


図31. モフォロジー処理の結果

### 5-2-6 真円度の抽出

肌色抽出及びモフォロジー処理で得られた領域内の，真円度が一番高い領域（最大内接円）を抽出する．そして得られた最大内接円の中心座標と半径を検出する．この中心座標は，指の探索範囲を限定するのに用いる．ここで抽出された最大内接円は手領域の拳部分にあたる．

また図32の最大内接円は，手領域の拳を覆い隠せるようにするため本来の大きさの1.5倍になるように設定している．このように設定している理由としては，後処理に当たる探索範囲の限定の際に，指領域だけを抽出するためである．

真円度の抽出結果は図32のようになる。

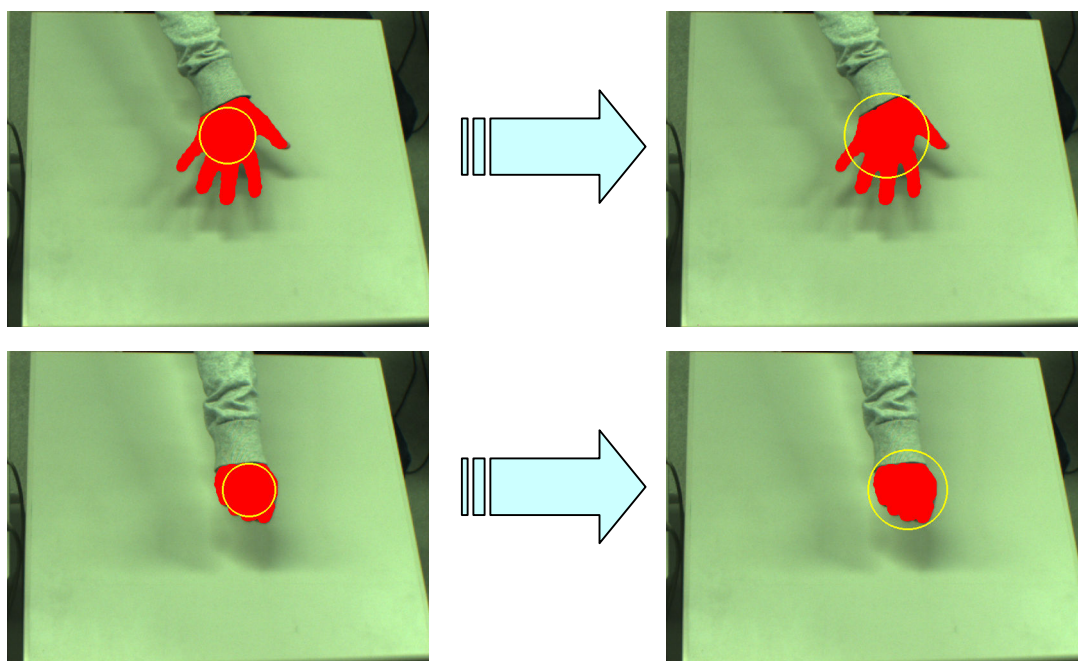


図32. 真円度の抽出

### 5-2-7 探索範囲の設定

図32のように手領域の最大内接円の中心座標を基準として、探索範囲（図33. 青い長方形部分）を設定する。そして、手領域と探索範囲の交差領域を検出する（図34）。個人によって手の大きさが異なるので、最大内接円の半径を変数として用い探索範囲を限定している。

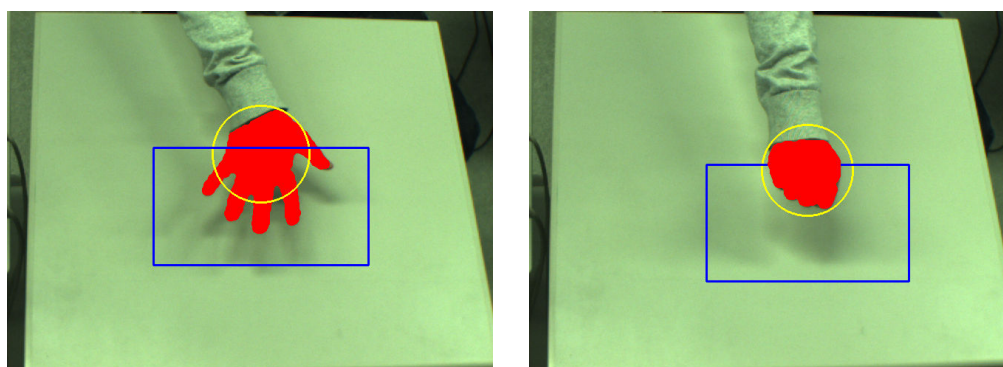


図33. 探索範囲の設定



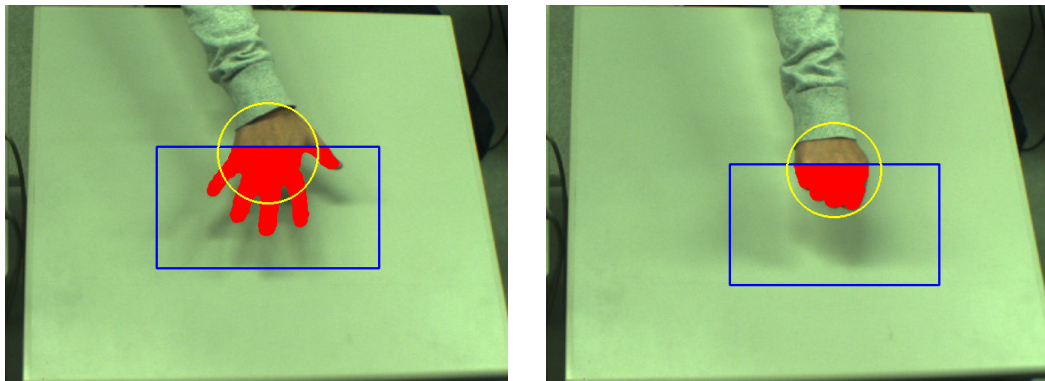


図34. 交差領域の検出

### 5-2-8 交差領域と最大内接円との差分領域の検出

5-2-6で抽出した最大内接円（図32 黄色枠）と5-2-7の交差領域（図34）との差分領域を求める。この差分処理後に残った領域数が、5だった場合はパー形状、0だった場合はグー形状とすることにより、グーパーの手形状が判別される。

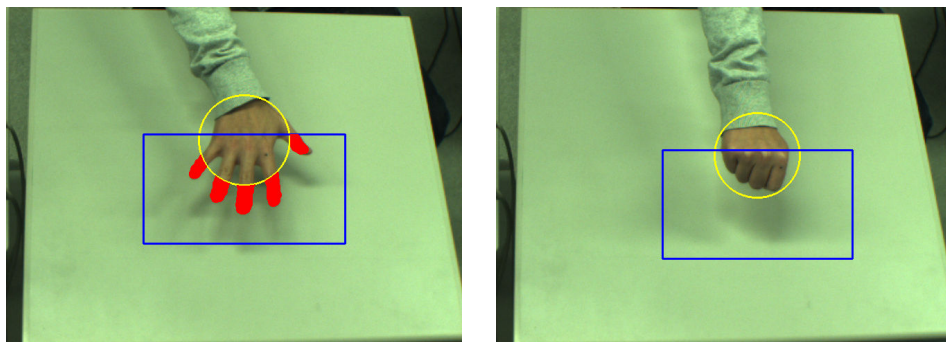


図35. 差分領域の検出

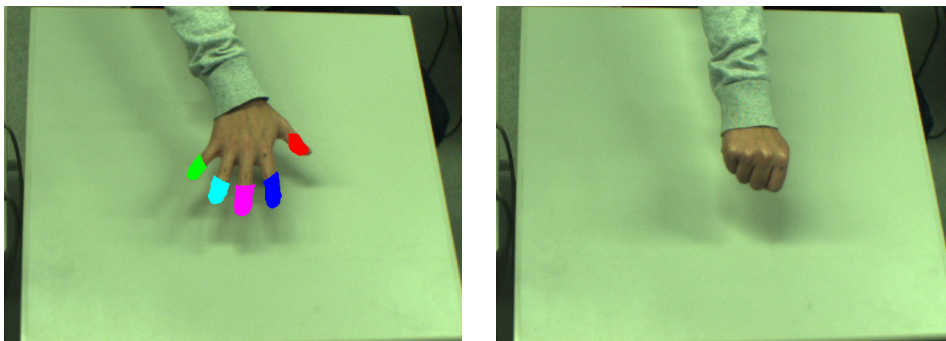


図36. パー形状（左），グー形状（右）

### 5-3 ジェスチャ形状の認識（指形状）

指形状の検出で行われる処理の流れは以下のようなフローチャートの通りである。各処理の説明は順次示すことにする。

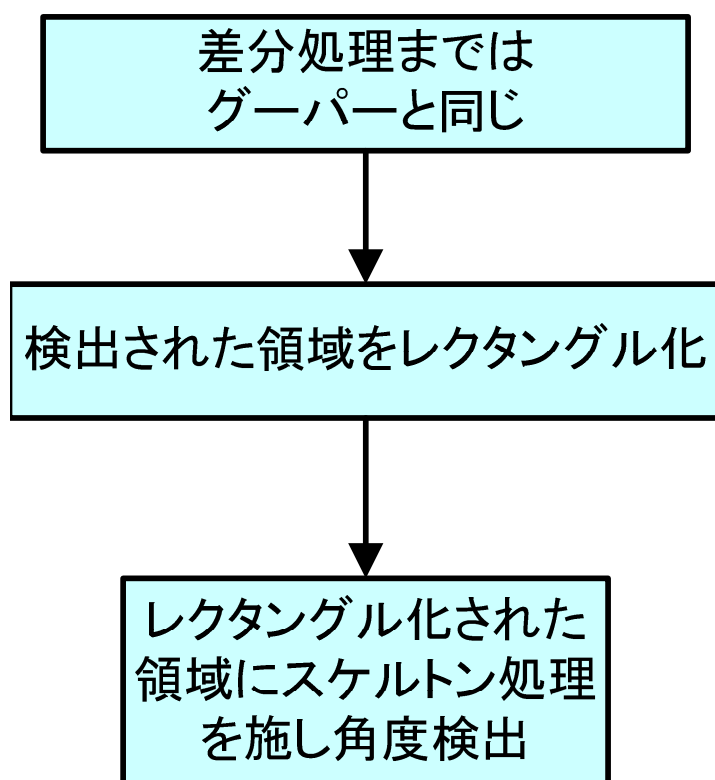


図37. 指形状認識のフローチャート

### 5-3-1 検出された領域のレクタングル化

抽出された指の肌色領域から、角度とその領域の中心座標を検出する前準備として領域をレクタングル化（領域を長方形に整えること）する。

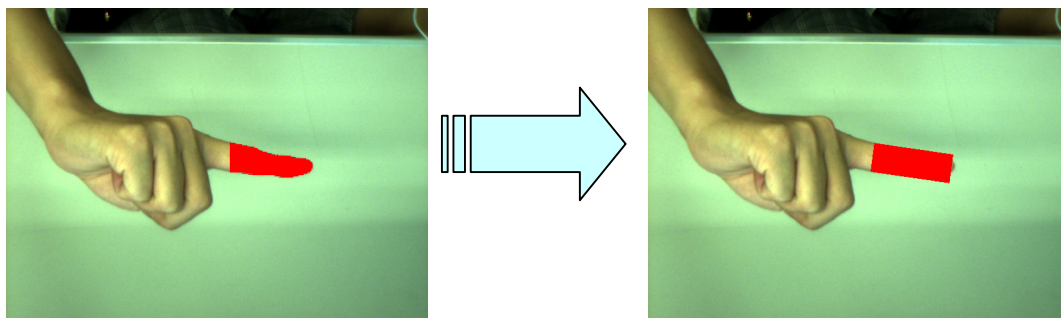


図38. レクタングル化の結果

### 5-3-2 レクタングル化された領域の角度検出

レクタングル化（細線化）された領域にスケルトン処理を施し指の角度を検出する。スケルトン処理とは、広がりのある連結成分をその芯線に相当する線図形変換を施す処理のことを言う。アルゴリズムの基本は、図形をその幅が1になるまで連結成分の周辺の1画素を消去可能なものから順に削っていく（輝度を0に置き換える）操作である。ある1画素が消去可能であるとは、その画素を消去しても画像内の連結成分の個数および穴の個数が変化しないことを言う。スケルトン処理によって施された芯線の位置は、もとの図形のほぼ中心にくる。

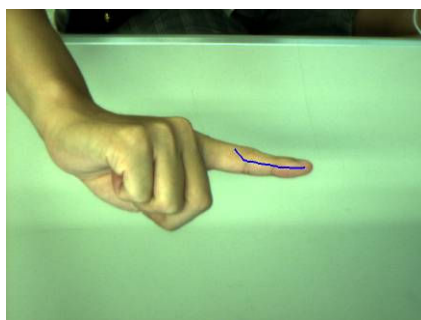


図39. スケルトン化の結果



## 5-4 実空間と仮想空間のキャリブレーション

実空間にあるユーザーの手と仮想空間上の手のアイコンが追従して動くためには、両空間のキャリブレーションが必要である。そこで、机の領域（x-z 方向）と 2 台のカメラが撮らえた領域でなす範囲（y 方向）を実空間の作業領域（図 40）とする事により、ディスプレイ上の 3 次元仮想空間と対応付けている。

実空間と仮想空間のキャリブレーションの方法としては、実空間の作業領域に手を置いて、その際の手の入力画像から画像処理によって得られる位置特徴と、仮想空間上の手のアイコンの位置座標との関係を表す線形モデルを重回帰分析によって推定している。手を置く場所としては、図 41 のように実空間を全 3 層、1 層を 30 分割の計 90 箇所である。90 箇所それぞれは、全 10 データ位置特徴を取得しておき、全部で 900 データを使って重回帰分析を行っている。なお、このデータ取得に関してはシステムを使う度に行う必要は無いが、各ユーザーごとの個人差低減のためシステム起動の際にユーザーごとの拳のデータを取得し、微調整を行う必要はある。

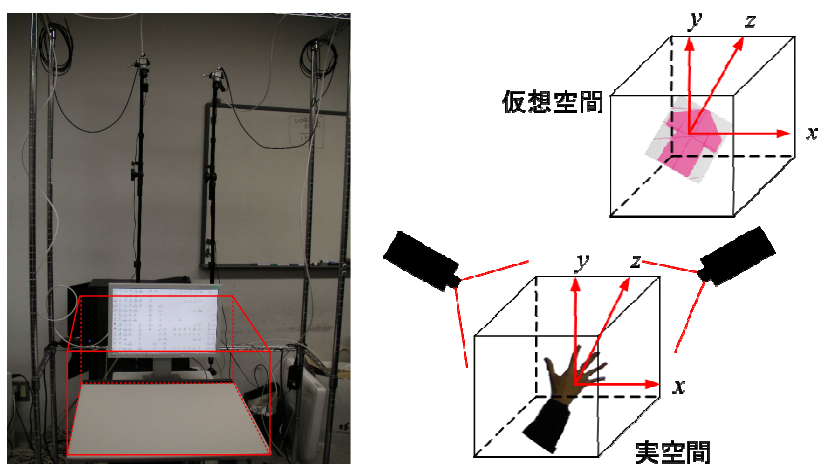


図 40. 実空間の作業領域（左）、実空間と仮想空間の対応付け（右）

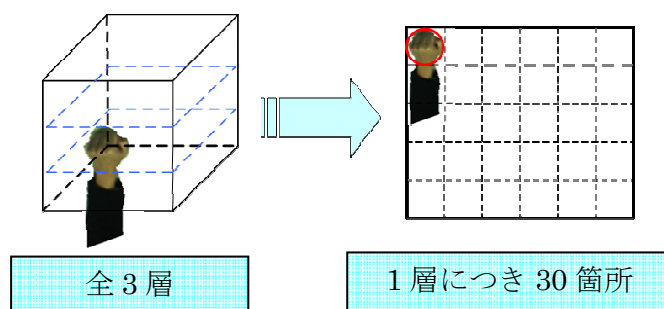


図 41. 作業領域の位置特徴

## 5-4-1 重回帰分析

重回帰分析とは、複数の説明変数（独立変数）から1つの目的変数（従属変数）を予測・説明するための分析法である。回帰分析の説明変数が複数になったもので、適切な変数を複数選択することで、計算しやすく誤差の少ない予測式を作ることができる。また、どの説明変数がどの程度目的変数に、影響を与えているかを知ることができる。

目的変数  $y$  とそれに影響を与える複数の説明変数  $x_1, x_2, x_3, \dots, x_n$  のあいだにおいて

$$y = a_0 + a_1x_1 + a_2x_2 + \dots + a_nx_n \quad \dots (7)$$

を設定し、この式に基づいて予測する。なお、 $a_1, a_2, \dots, a_n$  を偏回帰係数、 $a_0$  を定数項という。

\* 偏回帰係数：他の説明変数の影響を除き（一定にする）、ある説明変数の値が1だけ増加させたとき、目的変数の予測値がどれだけ変化するかを表す。

今回は本システムの重回帰分析にて、目的変数を手のアイコンの推定位置座標（3変数）、説明変数を実空間の手の拳の3次元位置座標と拳領域の面積（4変数）とした。画像処理により得られる「(取得画像の) 拳の x 座標」「拳の y 座標」「拳の z 座標」「拳の面積」の4つの特徴量を（6）式の説明変数として使用する。偏回帰係数（係数  $b$ ）は、重回帰分析によって学習させることで算出される。

これを「手のアイコンの x 座標」「手のアイコンの y 座標」「手のアイコンの z 座標」を目的変数とした式にそれぞれ当てはめ使用する。

しかしながら、回帰分析を行うと実測値を近似する回帰式（予測値）が得られるが、回帰式は目的変数ではなく、あくまで説明変数に基づいて予測した値で誤差（残差）が生じることもある。したがって、得られた回帰式が常によく当てはまっているとは限らない。

## 第6章 実験

仮想空間上の手のアイコンと実空間の手の動きの操作性に関する評価実験を2種類行った。実験1は、奥行き動作の評価と仮想空間での玉の位置情報を取得する時、奥行きのある中で人間の視覚と実際の座標とで違いがあるのかを調べる実験である。実験2は、提案したインターフェースと既存の入力装置とを比較した場合の特徴について評価する実験である。

### 6-1 実験1

この実験は、人間が仮想空間での物体の位置情報を取得するとき、奥行きのある中で、視覚と実際の座標とで誤差があるか調べる目的のものである。

実験方法としては、仮想空間に浮かんでいる玉を被験者が利き手で掴み、指定したマーカーの位置まで運んだあとその位置で玉を離してもらい（図42）。その際の、指定位置の位置座標と実際に玉が離れた位置座標とのズレを計測する。指定位置にはマークがあるので、ユーザーは自分の感覚でマークの位置にきたと思ったらそこで玉を離してもらい。玉は奥行き方向（z方向）へ運ぶものとする。実験は、高さ（y方向）0ピクセルの状態と高さ100ピクセルの状態の2種類に対し、それぞれ4条件、計8種類の実験を行った（表3）。

表3. 実験条件（単位：ピクセル）

	奥行き(z方向)の移動	高さ(y方向)
条件1	手前→中間(0→300)	高さ0
条件2	手前→奥(0→600)	
条件3	奥→中間(600→300)	
条件4	奥→手前(600→0)	
条件5	手前→中間(0→300)	高さ100
条件6	手前→奥(0→600)	
条件7	奥→中間(600→300)	
条件8	奥→手前(600→0)	

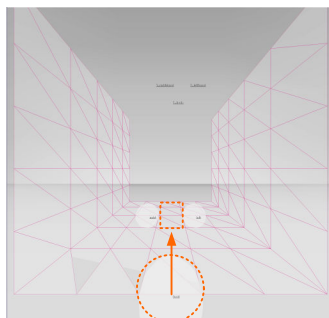


図42. 実験1の内容

各 8 条件は以下のようにになっている.

**【条件 1】**

高さなし (0 ピクセル) の状態で, 手前 (0 ピクセル) にある物体を中間 (300 ピクセル) へ持っていく.

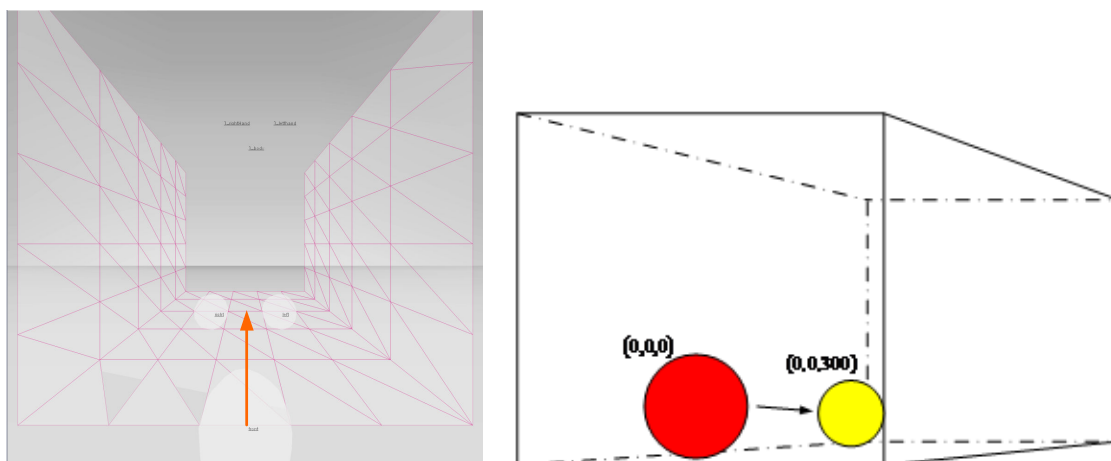


図 43. 条件 1 の移動実験

**【条件 2】**

高さなし (0 ピクセル) の状態で, 手前 (0 ピクセル) にある物体を奥 (600 ピクセル) へ持っていく.

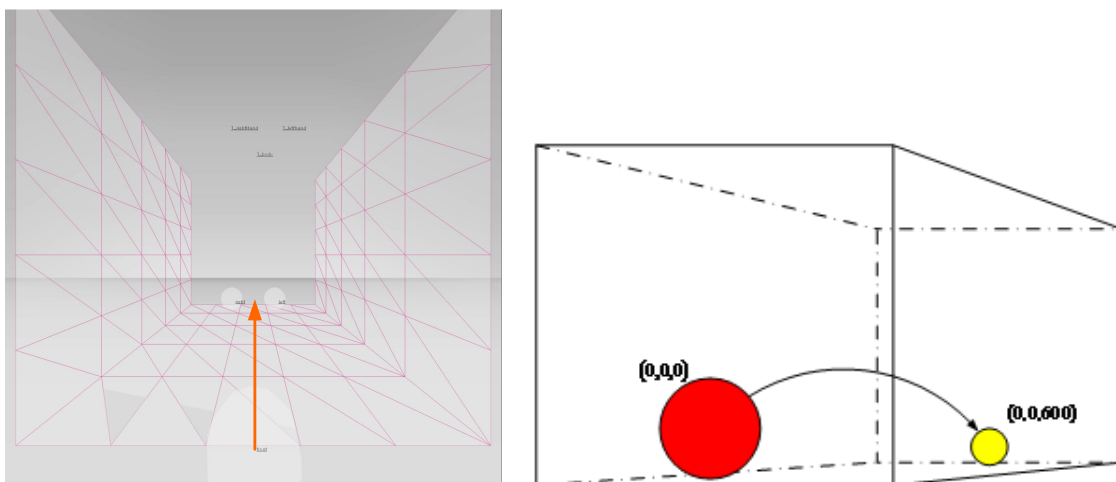


図 44. 条件 2 の移動実験

**【条件 3】**

高さなし (0 ピクセル) の状態で, 奥 (600 ピクセル) にある物体を中間 (300 ピクセル) へ持っていく.

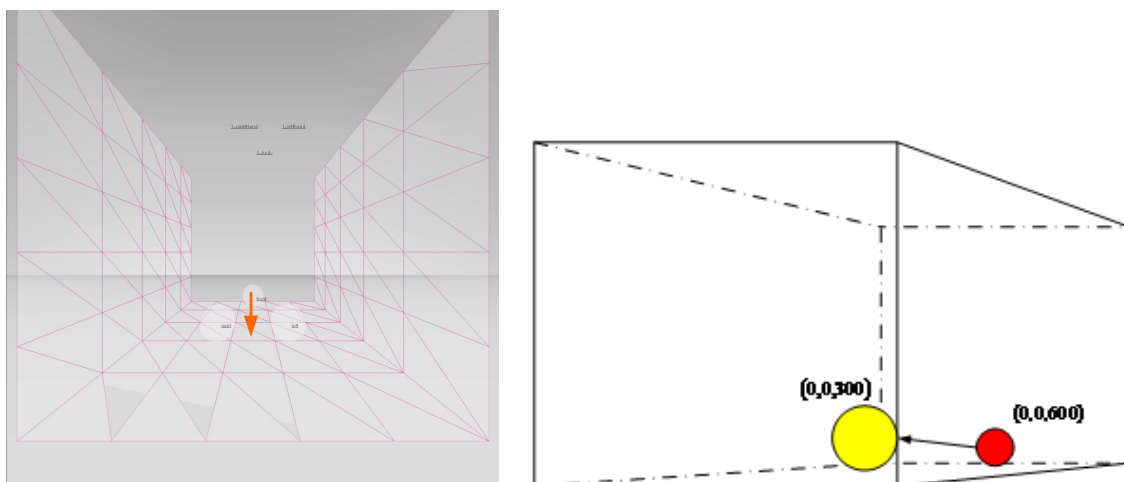


図 45. 条件 3 の移動実験

**【条件 4】**

高さなし (0 ピクセル) の状態で, 奥 (600 ピクセル) にある物体を手前 (0 ピクセル) へ持っていく.

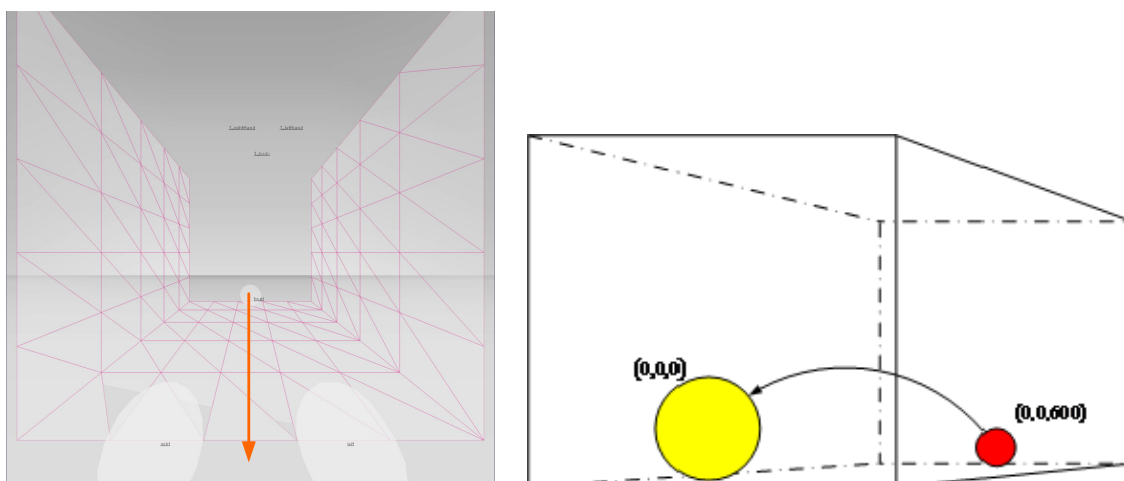


図 46. 条件 4 の移動実験

**【条件 5】**

高さあり（100 ピクセル）の状態で，手前（0 ピクセル）にある物体を中間（300 ピクセル）へ持っていく．

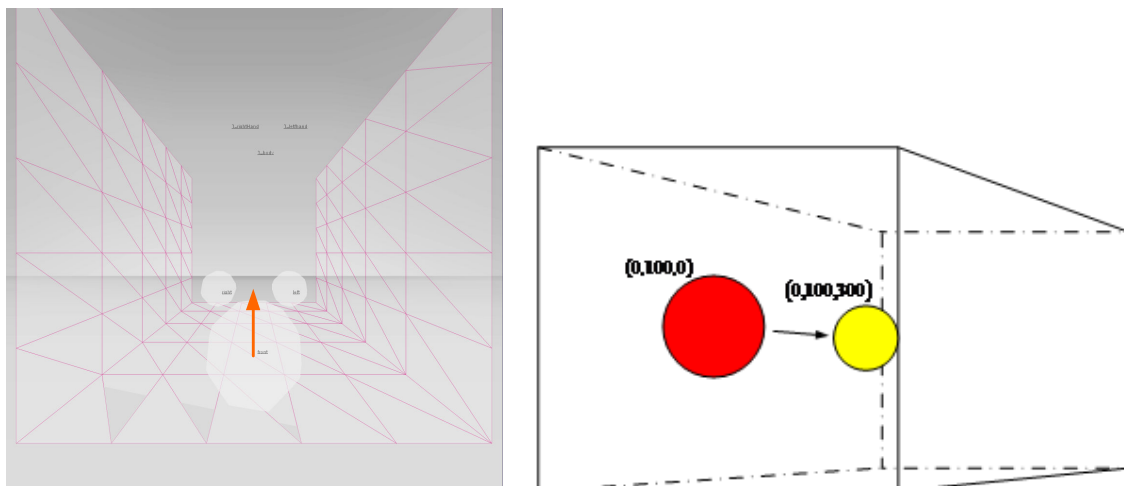


図 47. 条件 5 の移動実験

**【条件 6】**

高さあり（100 ピクセル）の状態で，手前（0 ピクセル）にある物体を奥（600 ピクセル）へ持っていく．

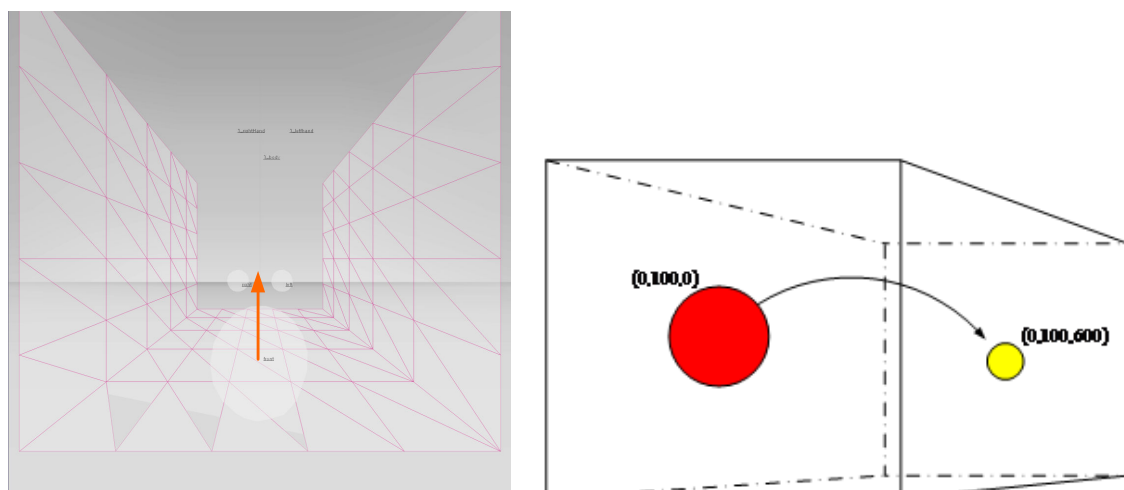


図 48. 条件 6 の移動実験

**【条件 7】**

高さあり（100 ピクセル）の状態で，奥（600 ピクセル）にある物体を中間（300 ピクセル）へ持っていく．

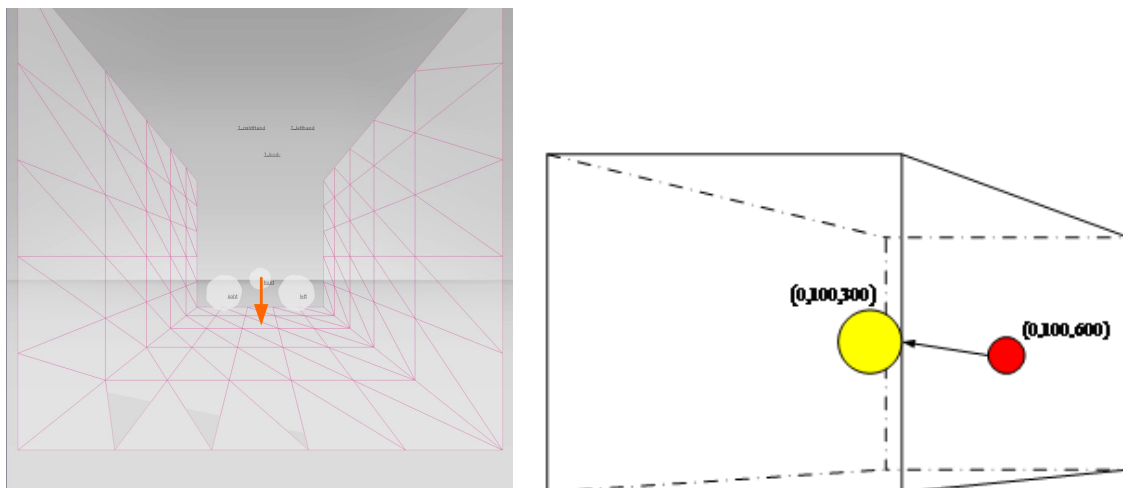


図 49. 条件 7 の移動実験

**【条件 8】**

高さあり（100 ピクセル）の状態で，奥（600 ピクセル）にある物体を手前（0 ピクセル）へ持っていく．

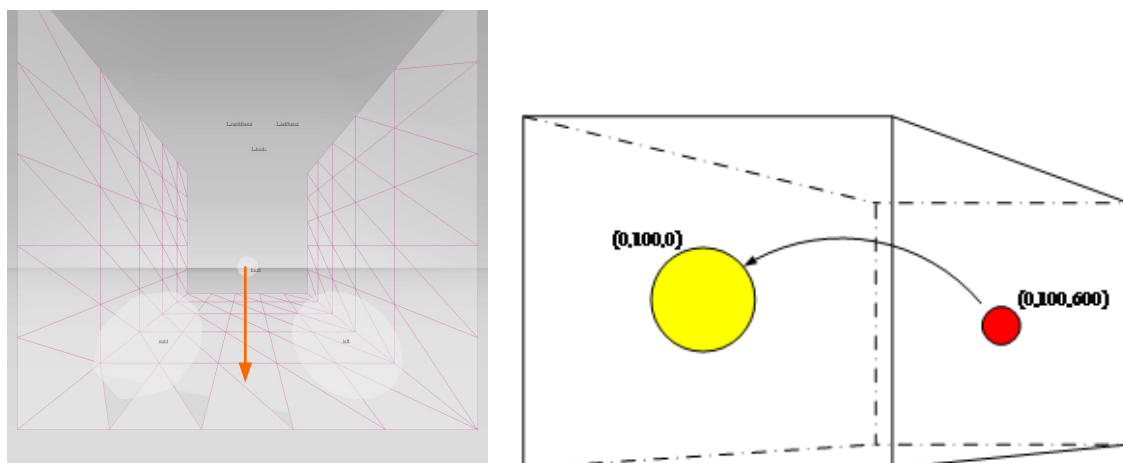


図 50. 条件 8 の移動実験

計 8 条件の実験は被験者 14 人に行ってもらった。被験者 1 人あたり全 8 条件をそれぞれ 5 回ずつ行ってもらい、計 560 データ取得を取得した。表 4 は被験者ごとに、各 8 条件の座標のズレ値の平均をまとめたものである。図 51 は、各 8 条件のズレ平均を棒グラフにしたものである。

表 4. 実験 1 結果 (単位 : ピクセル)

	条件1	条件2	条件3	条件4	条件5	条件6	条件7	条件8
被験者1	52.8	23.4	54.2	52.6	26.8	33.8	14.8	27.4
被験者2	73.2	34.6	67.0	45.2	31.6	53.0	59.2	30.6
被験者3	88.2	10.0	82.6	26.4	39.8	19.4	25.4	21.4
被験者4	15.0	83.8	12.4	14.6	38.6	81.8	12.8	29.0
被験者5	25.8	58.4	16.8	21.0	56.6	63.0	55.6	3.2
被験者6	75.8	89.6	113.4	46.4	158.2	41.6	51.8	34.8
被験者7	29.2	31.2	24.0	27.2	22.2	64.0	12.4	18.6
被験者8	20.0	38.8	35.6	39.0	111.8	23.2	59.8	34.0
被験者9	17.4	38.8	16.0	13.0	33.6	48.8	7.4	11.6
被験者10	34.2	29.4	16.2	17.0	12.2	97.4	26.4	6.8
被験者11	7.4	29.4	18.2	13.6	43.2	82.2	15.6	3.4
被験者12	20.6	22.2	31.2	38.4	88.2	90.4	15.4	12.4
被験者13	25.4	43.2	34.2	10.0	50.4	162.6	18.0	19.0
被験者14	27.0	41.4	40.6	4.0	41.2	74.6	50.8	19.8
ズレ平均	36.57	41.01	40.17	26.31	53.89	66.84	30.39	19.43

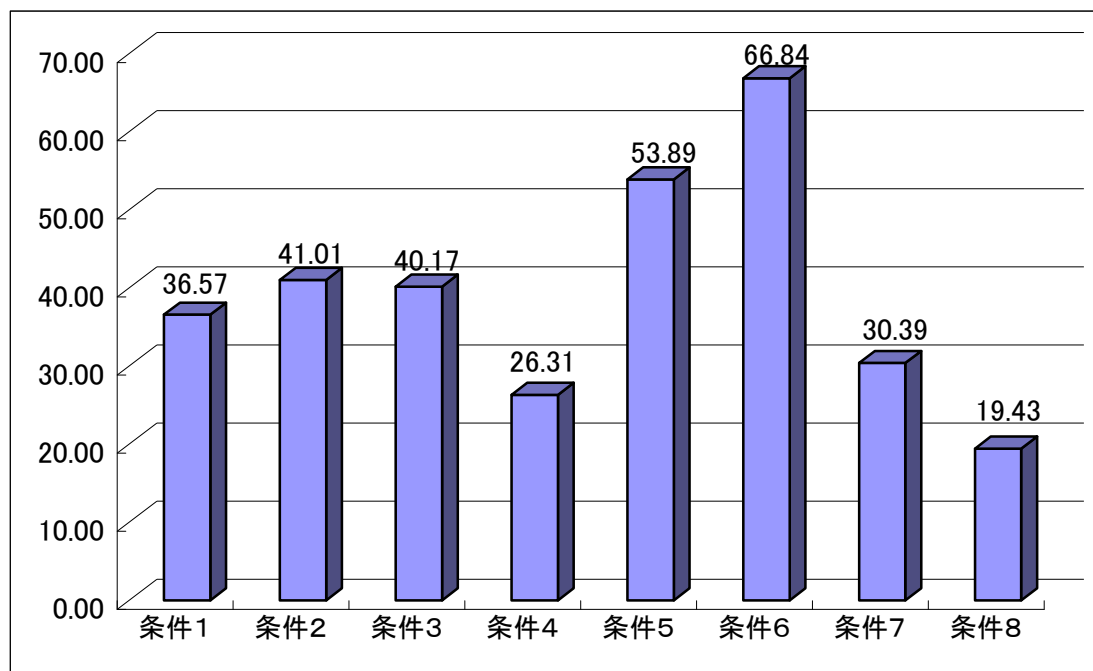


図 51. 実験結果のズレ平均① (単位 : ピクセル)



## 6-2 実験 2

本システムで提案した操作情報取得システムはいわゆるハンドジェスチャによる操作である。そこで、一般的に PC を操作する場合に用いられる入力装置と比較して、マウスとハンドジェスチャでの操作の比較実験を行った。この実験で使用した入力装置はマウスを使用した。

実験方法としては、実験 1 と同様に仮想空間に浮かんでいる玉を被験者が掴み、指定した位置まで運んだあとその位置で玉を離してもらう。図 51 のような手順で、まずは右下にある玉を掴んで①の方向に玉を移動させる。左下の位置まできたら玉を離す。離れた玉はまた掴んで②の方向に玉を移動させる。あとはその繰り返しで、最終的に右下の位置で玉を離したら実験終了である。この試行をハンドジェスチャとマウスの 2 種類に対して行い、試行が終了するまでの時間を計測した。

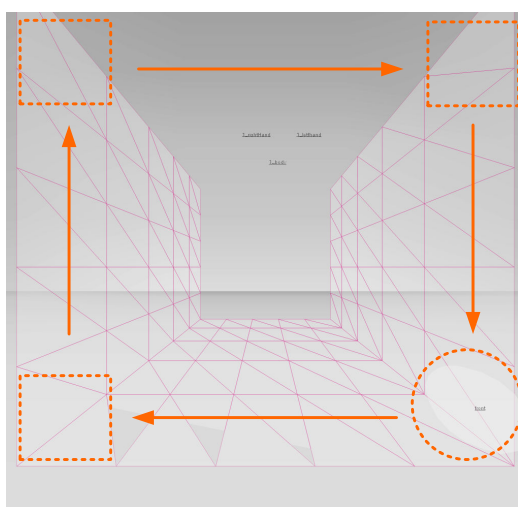
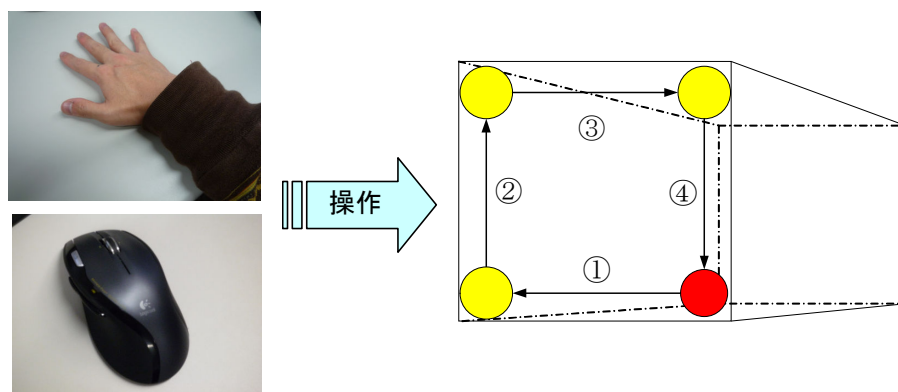


図 52. 実験 2 の内容

実験は被験者 10 人に行ってもらった。被験者 1 人あたりハンドジェスチャでの操作，マウスでの操作を 1 回行ってもらった。表 5 は被験者ごとの各操作インターフェースの操作時間をまとめた表である。

表 5. 実験 2 結果 (単位: 秒)

	操作インターフェース	
	ハンドジェスチャ	マウス
被験者A	33.5	23.9
被験者B	33.6	22.5
被験者C	39.1	21.7
被験者D	33.5	23.4
被験者E	33.1	25.1
被験者F	32.4	22.9
被験者G	41.8	21.7
被験者H	36.4	20.9
被験者I	34.8	22.2
被験者J	30.6	23.8
平均操作時間(秒)	34.9	22.8

## 第7章 考察

### 7-1 分散分析

取得した数種類の実験データを比較するために、分散分析を用いる。この分析によって得られた有意差などから、考察をしていく。

分散分析 (Analysis of variance : ANOVA) とは、母集団においてグループの平均値に差があるかどうかを決定するために実施され、各グループの平均値に有意差 (統計的に意味のある差) があるかを証明する分析である。各グループに明らかな差がある場合や全く差がない場合ではなく、分布が重なり微妙な差がある場合に有効である。また、2つのグループを比較する場合 t 検定が一般的だが、データが3つ以上のグループに分けられた場合に、分散分析で比較を行うことができる。

この検定は、Excel でできるが今回はより細かい指定のできる「ANOVA4」という WEB 上のソフトを用いる。

#### 【検証方法】

1. 人間の視覚と実際の座標とのズレに注目し、Excel で表を作成。
2. ANOVA4 を起動し、「被験者間要因」、「被験者内要因」の数を記入。
3. それぞれの要因名と水準数を記入。
4. 被験者数を記入。
5. Excel の調べたいデータを貼り付ける。

#### \* 被験者間要因 (Between-Subjects Factor)

複数の条件を、別々の被験者が実験を行うときに用いる。Ex)性別の違い

#### \* 被験者内要因 (Within-Subject Factor)

複数の条件を、同一の被験者が実験を行うときに用いる。Ex)時間の前後による違い

上記の方法で、簡単に有意差があるかどうか調べることができる。

表 6. 分散分析結果

source	SS	df	MS	F	p
A:高さなし奥へ(条件 1, 2)	2699.382	1	2699.382	14.980	0.0019 ***
B:高さあり奥へ(条件 5, 6)	17600.925	1	17600.925	26.540	0.0002 ****
C:高さなし手前へ(条件 3, 4)	10298.006	1	10298.006	10.936	0.0057 **
D:高さあり手前へ(条件 7, 8)	2155.120	1	2155.120	8.159	0.0135 *
E:手前から中間へ(条件 1, 5)	1872.257	1	1872.257	4.983	0.0438 *

+ p<.10, \* p<.05, \*\* p<.01, \*\*\* p<.005, \*\*\*\* p<.001

分散分析の結果、表 6 のものに高い有意差があった。p 値に着目し、p<0.05 ならば有意差ありといえる。

\*SS (sum of squares) : 変動, 偏差平方和. 観察値  $X_i$  から平均値  $Xbar$  を引いた偏差の 2 乗和  $\sum (X_i - Xbar)^2$ . これを自由度で割れば, 不偏分散 (平均平方) が得られる.

\*df : 自由度. 互いに独立に動けるデータの個数のようなもの統計量  $X_1 + X_2 + \dots + X_n$  の自由度は n. だが,  $(X_1 - Xbar)^2 + (X_2 - Xbar)^2 + \dots + (X_n - Xbar)^2$  の自由度は n-1 となります.

\*ms : 不偏分散. 標本分散は, その期待値が母分散 (母集団の分散) よりも若干小さくなる。そこで、期待値が母分散に等しくなるように補正をかけた

$$u^2 = \frac{1}{n-1} \sum_{i=1}^n (\bar{x} - x_i)^2$$

が用いられることも多く, これを母分散の不偏推定量であるとの意味で不偏分散 (unbiased variance) と呼ぶ.

\*F : F 値 =  $\frac{[\text{要因変動}] \text{の平均平方和}}{[\text{誤差変動}] \text{の平均平方和}}$  F 値が大きいほど P 値が小さくなる.

\*P : P 値 有意確率.

## 7-2 実験1の考察

まず、実験 1 の各条件での位置座標の誤差平均を比べてみる。まず、手前から奥方向への移動（条件 1・2・5・6）になるにつれ誤差が大きくなり、奥から手前方向への移動（条件 3・4・7・8）になるにつれ誤差が小さくなっていった。特に高さあり・なしどちらも手前から奥への移動（条件 2・6）が一番誤差が大きいという結果になった。分散分析でも、条件 1・2 と条件 5・6 に一番高い有意差が確認できた。これは、手前より奥になるにつれて手のアイコンやマークの表示が小さくなるため、認識しづらくなったのではないかと考えられる。

また、奥から手前方向への移動である条件 3・4 と条件 7・8 を比べると、高さ有りの条件 7・8 の方が誤差が小さくなっている。条件 7・8 だけ誤差が小さくなった理由として考えられるのは、ユーザーの目線の高さに玉があり、視覚的に認識しやすかったためではないかと考えられる。目線の高さにあると物体が捉えやすいというのは、球技系のスポーツなどでは言われている事である。例えばテニスではボレーを打つとき、ラケットと目線が同じ高さになるようにすると良いと言われている。本実験でも、目線の高さにあり、かつ奥から手前へ玉を運ぶ条件 7・8 が一番誤差が少なかったのはこの事より言えるのではないだろうか。

また、実験をやらしてもらった被験者のアンケートでは、実験 1 がやりやすく、実験 4 がやりづらいという意見が多かった。実際の誤差平均（図 53）を比べてみると、やりづらいと感じていた実験 4 の誤差は小さく、やりやすいと感じていた実験 1 の誤差は大きかった。この結果から、人間の感覚との違いが生じたのではないだろうか。もちろんすべての被験者が実験 1 がやりやすく、実験 4 がやりにくいと感じた訳ではないので、個々によって感覚の捉え方に違いはあるとは思われる。また、3次元座標の正確な取得ができていない点があるので、それに伴うシステムの感度の悪さも影響しているのではないかとと思われる。

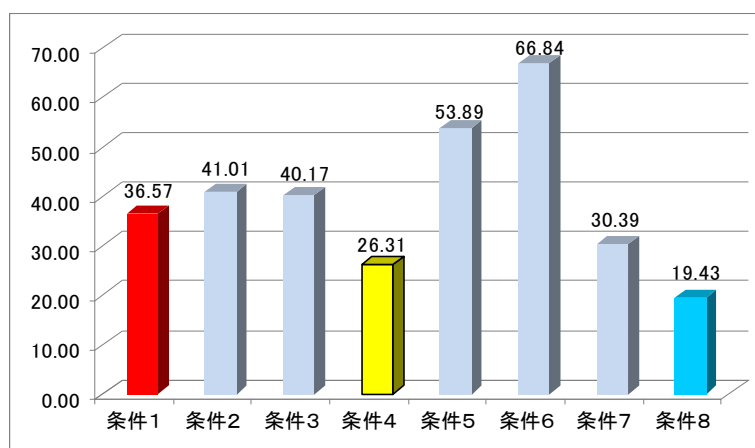


図 53. 実験結果のズレ平均② (単位：ピクセル)

### 7-3 実験2の考察

実験 2 の考察である。それぞれの試行にかかった時間を平均したものが表 4 である。ハンドジェスチャとマウスを比較すると、操作時間に 10 秒近くの差が出ている。マウスの場合、動かせばカーソルも動き、動きを止めればカーソルもぶれる事なく止まるものである。

しかし、本システムのハンドジェスチャは、一箇所で動きを止めても、仮想空間上の手のアイコンには若干のぶれが生じている。手の正確な位置関係が上手く認識されていない事が、今回の操作比較に大きく影響していると考えられる。実験をやってもらった被験者のアンケートでは、やはりマウスより使いにくいという結果が多かったが、手で操作できるのは面白い、操作性が向上すれば使いやすくなるなどの意見もあった。

このアンケート結果から、操作性の向上というのは、システム全体に対する印象を変える重要な要素だと思われる。今回はマウスのような既存の入力装置と比べるとまだ不十分である本システムで、操作性の向上ができれば、評価も改善される可能性があるのではないだろうか。

表 7. 各操作インターフェースの平均操作時間

操作インターフェース	ハンドジェスチャ	マウス
平均操作時間(秒)	34.9	22.8

## 第8章 むすび

ユーザーに接触型センサやマーカなどを装着させる事なく、カメラを使った画像処理による手形状ならびに3次元位置の推定、ジェスチャの認識を用いたシステムの研究を進めてきた。3次元の仮想空間を使った人間のジェスチャを使ったシステムの場合、ユーザーにとって使いやすい操作性の追及、ユーザーが認識しやすいような表示などの視覚的な考慮が重要だと思われる。これらの要因はすべて、ユーザーがストレスフリーを感じる事ないシステムにつながるからこそ、考えなければいけない点である。本システムはまだ完璧ではない点が多く、特に操作性の点で手の正確な3次元の位置座標の取得、システム全体の安定したfpsの確保によるシステムの操作性の安定に努める必要などが考えられる。

今回仮想空間と実空間のキャリブレーションに関しては、重回帰分析に関して言えば個人々のデータ取得はしなかった。ただ、個人差低減のための微調整は行った。今回、手の正確な3次元の位置座標の取得にアルゴリズム以外の問題点を挙げるとすれば、キャリブレーションの個人依存性が考えられる。手形状のような手の大きさの違い、指の長さの違いというような個人差を抱える特徴情報をキャリブレーションに用いると、データ取得者と操作者は同じのほうが好ましいのではないかと思われる。

また、既存の入力装置に比べると、玉を掴む・離す認識に関してはそれほど差が無いものの、スムーズな動きに関してはまだ劣っている。もちろん、どちらの面もできていなければユーザーに使いやすいシステムとはいえない。そのために、プログラムの改善によるユーザービリティの向上も考えなければいけない。

改善する余地はまだあるが、ストレスフリーの改善、ユーザービリティの向上ができれば、カメラを使った画像処理によるジェスチャ認識システムの構築の実現は十分可能であると考えられる。

## 謝辞

本研究の一部は日本学術振興会の科研費（基盤研究(B)18300076）の助成を得た．ここに記して謝意を表す．

また，本研究を進めるにあたり，多大なご指導を頂きました法政大学理工学部応用情報工学科赤松茂教授に深く感謝いたします．

本研究のシステム構築の際に，全般的にサポートをして頂きましたシステム計画研究所濱哲平氏に感謝いたします．

本研究の評価実験の際に，準備段階からデータのまとめなどにご協力頂きました法政大学工学部システム制御工学科4年 山田直樹氏に感謝いたします．

そして，評価実験の被験者にご協力頂きました皆さんに感謝いたします．



## 参考文献

- [1] 木村朝子, 鶴田剛史, 酒井理生 他, “広視野電子作業空間に関する考察システム試作～マイノリティ・レポート型 I/F とその発展形”, 情報処理学会シンポジウム論文集, vol2005 No.4 Page.143-150 February 2005
- [2] 所洋平, 寺田努, 塚本昌彦, “2 つの加速度センサを用いたポインティング手法”, 第 16 回インタラクティブシステムとソフトウェアに関するワークショップ論文集, November 2008
- [3] Ubi-Finger: モバイル指向ジェスチャ入力デバイスの研究, “情報処理学会論文誌, vol.43 No.12 Dec.2002
- [4] 山本豪志朗, 徐会川, 佐藤宏介, “PALMbit-Silhouette: 掌シルエットの重畳表示によるデスクトップアクセス”, 情報処理学会シンポジウム論文集, vol2008 No.4 Page.109-116 March 2008
- [5] 大森敬由, 赤松茂, “画像処理による空間ポインティング・インタフェースの構築,” Media Computing Conference 2007 予稿集, pp. 33-34, June 2007
- [6] 石井裕, “タンジブル・ビット - 情報と物理世界を融合する, 新しいユーザー・インタフェース・デザイン -”, 情報処理, vol43.No3,Page.222-229 2002
- [7] 蔵田武志, 酒田信親, 葛岡英明, 興梠正克, 大隈隆史, 西村拓一, “遠隔協調作業のためのウェアラブル・タンジブルインタフェース”, SICE 第 69 回パターン計測部会研究会, 11-18 2006
- [8] 塚田浩二, “情報家電制御に適した次世代インタフェース”, 情報処理, vol43.No3,Page.222-229 2002
- [9] 小池秀樹, 小林貴訓, 佐藤洋一, “机型実世界指向システムにおける紙と電子情報の統合および手指による実時間インタラクションの実現”, 情報処理学会論文誌, vol.42 No.3 Mar.2001
- [10] 株式会社リンクス <http://www.linx.jp/index.shtml>
- [11] 株式会社リンクス 画像システム事業部: “HALCON 活用法”, 株式会社リンクス 出版事業部 2004-12
- [12] 株式会社リンクス 画像システム事業部: “画像処理アルゴリズムと実践アプリケーション”, 株式会社リンクス 出版事業部 2008-6
- [13] Jamie Sherrah, Shaogang Gong, ” Skin Colour Analysis”  
[http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL\\_COPIES/GONG1/](http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/GONG1/)