

NAIST-IS-DT0161007

博士論文

操作的手動作入インタフェースのための
多視点画像に基づく手形状推定

上田 悦子

2003年9月30日

奈良先端科学技術大学院大学
情報科学研究科 情報システム学専攻

本論文は奈良先端科学技術大学院大学情報科学研究科に
博士(工学) 授与の要件として提出した博士論文である。

上田 悦子

審査委員： 小笠原 司 教授
横矢 直和 教授
木戸出 正繼 教授
松本 吉央 助教授

操作的な手動作入力インタフェースのための 多視点画像に基づく手形状推定*

上田 悦子

内容梗概

本論文は、多視点画像に基づく手形状推定の新しい手法を提案するものである。人間の意識的な手動作は、人間に対する「対話的な手動作」と物体に対する「操作的な手動作」に分けることが出来る。本論文で提案する手法は、後者の操作的な手動作の際の手形状を推定することを目的としており、そのためには手指の各関節の回転・屈曲角度を定量的に推定する必要がある。また、推定した操作的な手動作をコンピュータインタフェースやVRインタフェース、ロボットインタフェースの直接操作入力として応用することを目指している。これらの実現により、従来のポインティングデバイスを用いた入力インタフェースに比べて、より直接的で直感的な入力インタフェースを構築することが出来るという効果をもたらす。

はじめに、本研究のキーアイデアとなる多視点画像に基づく手形状推定手法の提案を行う。提案する推定手法において、入力データは多視点カメラシステムを用いて得られた実空間の手の2次元シルエット画像群である。得られた複数視点のシルエット画像群は統合され、3次元観測データであるボクセルモデルとして再構成される。一方で、人間の手は関節物体として31自由度のリンクと表面形状データで表現されており、このモデルを骨格モデルと呼ぶ。提案する推定手法は、画像キャプチャごとに逐次作成されるボクセルモデルと、骨格モデルの位置関係より3次元フィッティングを行い骨格モデルの指の各関節回転角度を推定することを基本とするものである。シミュレーションにより提案手法の評価を行い、その上で多視点動画像を入力とした連続的な手形状の推定実験を行う。次に、提案した基本手法の推定速度の向上を図るために、手の表面形状データを Catmull-Clark

*奈良先端科学技術大学院大学 情報科学研究科 情報システム学専攻 博士論文, NAIST-IS-DT0161007, 2003年9月30日。

サブディビジョンサーフェスで表現し，基本推定手法に粗密探索法を導入することを提案する．シミュレーションと実画像を入力とした実験により，改良手法の評価とその有効性を明らかにする．最後に，提案手法により推定された操作的な手動作を，物体あるいは形状操作のための入力インタフェースとして応用し，評価を行う．評価のためのアプリケーションとして，提案手法を用いて推定した操作的な手動作によってバーチャルクレイの変形を行うバーチャルクレイモデリングシステムのプロトタイプを構築し，その操作性の検証を行う．

シミュレーションおよび実画像を入力とした実験とそれらの検証結果から，提案した多視点画像に基づく手形状推定手法の有効性・正当性，そして目標とするアプリケーションへの応用が可能であることを示すことが出来た．

キーワード

手形状推定, シルエット画像, 空間分割モデル, モデルフィッティング, サブディビジョンサーフェス, 直接操作入力インタフェース

Hand Pose Estimation based on Multi-viewpoint Images for Direct Manipulative Interface *

Etsuko Ueda

Abstract

This thesis proposes a novel method of hand shape estimation based on multi-viewpoint images. The human hand poses can be classified into the “communicative hand pose” and the “manipulative hand pose.” The proposed method in this thesis aims to estimate the arbitrary hand shape for the manipulative hand movement. For this purpose, the joint angles of all fingers are estimated quantitatively.

In this thesis, a hand pose estimation algorithm is firstly proposed. The proposed algorithm is the key idea of this research. In this method, the hand regions are extracted from multiple images obtained by the multi-viewpoint camera system. By integrating multi-viewpoint silhouette images, a hand shape is reconstructed as a “3-dimensional observation data.” Then joint angles are estimated by fitting a skeletal hand model to the 3-dimensional observation data. Secondly, in order to improve the accuracy of the hand pose estimation, another representation method of the human hand using the subdivision surface is proposed. Finally, the method of representing the clay is described. The virtual clay is modeled by Catmull-Clark subdivision surface, and is deformed based on the free-form deformation technique. A prototype system which can deform the virtual clay using the estimated hand shape is constructed, and the evaluation of the system is performed using this prototype system.

*Doctor's Thesis, Department of Information Systems, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DT0161007, September 30, 2003.

The experimental results which are obtained using hand-shape simulator and real image sequences indicated the validity of the proposed method.

Keywords:

Hand pose estimation, Silhouette image, Voxel model, Model fitting, Subdivision surface, Direct manipulative interface

目次

第1章 序論	1
1.1. 研究の背景	1
1.2. 研究の目的	2
1.3. 人間の手動作の認識と推定に関する従来研究	3
1.3.1 対話的・操作的な手動作認識	3
1.3.2 画像入力に基づく手形状推定	5
1.4. 本研究の位置付け	7
1.4.1 非接触型インタフェース	7
1.4.2 多視点カメラシステム	7
1.4.3 モデルベースの形状推定	8
1.5. 研究がもたらす効果	9
1.6. 本論文の構成	10
第2章 人間の手の機能と操作的な手動作の解析	13
2.1. 本章の趣旨	13
2.2. 人間の手の構造と運動	13
2.3. 操作的な手動作の解析	16
2.4. 本章の結論	26
第3章 多視点シルエット画像を用いた手形状推定手法の提案	27
3.1. 本章の趣旨	27
3.2. 本章で用いる手のモデル	27
3.2.1 骨格データ	28

3.2.2	表面形状データ	31
3.3.	観測データの3次元化	32
3.3.1	オクトリーによる形状再構成法	32
3.3.2	手のボクセルモデル	38
3.4.	手の形状推定	40
3.4.1	提案手法の原理	40
3.4.2	アルゴリズムの詳細	41
3.4.3	推定結果	45
3.5.	実画像を用いた手の形状推定	46
3.5.1	実験システム	46
3.5.2	カメラキャリブレーション	47
3.5.3	初期姿勢の推定	50
3.5.4	連続画像入力による手形状推定	53
3.6.	本章の結論	57
第4章	粗密探索法を導入した多視点シルエット画像を用いた手形状推定	59
4.1.	本章の趣旨	59
4.2.	サブディビジョンサーフェスを用いた手の表面形状データ	59
4.2.1	サブディビジョンサーフェスを用いた3次元形状表現	60
4.2.2	Catmull-Clarkサブディビジョンサーフェス	61
4.2.3	手の表面形状データの設計手順	64
4.3.	自然な手動作における拘束条件の導入	66
4.4.	粗密探索法による手形状推定の高速化	67
4.5.	実画像を入力とした手形状推定改良の検証	69
4.5.1	手の初期位置の決定	69
4.5.2	手形状推定の結果	70
4.6.	本章の結論	73
第5章	手形状推定手法の性能評価	75
5.1.	本章の趣旨	75

5.2.	ボクセルモデルの復元レベルの違いによる推定精度の比較	75
5.3.	カメラの台数の違いによる推定精度の比較	83
5.3.1	偽の指の発生による誤推定	83
5.3.2	一般的なカメラ台数による推定精度の比較	85
5.4.	カメラ位置の違いによる推定精度の比較	87
5.5.	シミュレーションによる手形状推定改良の検証	89
5.6.	本章の結論	90
第6章	コンピュータインタフェースとしての手形状推定の応用	93
6.1.	本章の趣旨	93
6.2.	バーチャルクレイモデリングインタフェースプロトタイプ	94
6.2.1	仕様	94
6.2.2	処理の流れ	94
6.3.	バーチャルクレイのモデル化	95
6.3.1	サブディビジョンサーフェスによるバーチャルクレイの表現	95
6.3.2	FFDを応用したバーチャルクレイの変形	98
6.3.3	提案したモデル化の評価	101
6.4.	実空間の手動作によるバーチャルクレイ変形	105
6.4.1	手の動作可能領域の検証	105
6.4.2	バーチャルろくろシステムプロトタイプ	106
6.4.3	手の位置と姿勢を用いたバーチャルクレイモデリングイン タフェース	108
6.4.4	指形状を用いたバーチャルクレイ変形	112
6.5.	本章の結論	116
第7章	結論	117
7.1.	まとめ	117
7.2.	課題と今後の展開	121
謝辞		123

著者研究業績	127
参考文献	129

目次

1.1	手動作の分類	4
2.1	手の構造と皮線	14
2.2	手の骨格	15
2.3	手の関節運動	16
2.4	じゃんけんの手動作 (0.5 秒間)	17
2.5	物を把持する手動作 (0.5 秒間)	18
2.6	粘土でコップを作る手動作 (0.5 秒間)	19
2.7	物を把持する際の手形状の変化	21
2.8	粘土細工の際の手形状の変化	22
2.9	ろくろによる形状作成の際の手形状の変化	23
2.10	画像中の補助線により計測した MP 関節屈曲角度変化	24
2.11	データグローブにより計測した平均の MP 関節屈曲角度変化	25
2.12	データグローブにより計測した最大の MP 関節屈曲角度変化	25
3.1	骨格データ	28
3.2	中指の関節軸	29
3.3	手の領域分割	32
3.4	オクトリーモデルとその構造	34
3.5	オクタントの投影	35
3.6	オクタントの属性	35
3.7	オクトリーによる形状再構成のフロー	36
3.8	片側距離変換マップ	37

3.9	手形状シミュレータ	39
3.10	シルエット画像	39
3.11	ボクセルモデル	40
3.12	関節角度推定処理フロー	42
3.13	モデルフィッティングの仕組み	43
3.14	トルクの発生	43
3.15	各種手形状における推定過程	45
3.16	実験環境	47
3.17	キャリブレーションボックス	48
3.18	キャリブレーションツール	49
3.19	初期手形状	51
3.20	実画像より作成したボクセルモデル	52
3.21	表面形状モデルの初期状態	52
3.22	初期姿勢の推定	53
3.23	手領域の定義	54
3.24	実画像入力による連続手形状推定の例 (1)	55
3.25	実画像入力による連続手形状推定の例 (2)	56
4.1	サブディビジョンサーフェスの過程	63
4.2	サブディビジョンサーフェス: (a) 初期状態, (b)1 回細分割, (c)2 回細分割, (c)3 回細分割	63
4.3	Shade 6 を用いて作成した手の初期格子	64
4.4	細分割された手形状データ	65
4.5	新しい手の表面形状データ	66
4.6	ボクセルモデルの違いと骨格モデルとの位置関係	67
4.7	粗推定時フィッティングのための力の発生	69
4.8	粗推定を用いた実画像手形状推定 (1)	71
4.9	粗推定を用いた実画像手形状推定 (2)	72

5.1	オクトリーレベルが推定結果に与える影響 (左側: ボクセルモデル, 右側: 推定された手形状)	76
5.2	繰り返しごとの収束率の推移 (Pose-E)	78
5.3	繰り返しごとの収束率の推移 (Pose-F)	78
5.4	繰り返しごとの収束率の推移 (Pose-G)	79
5.5	繰り返しごとの収束率の推移 (Pose-H)	79
5.6	繰り返しごとの推定誤差の推移 (Pose-E)	81
5.7	繰り返しごとの推定誤差の推移 (Pose-F)	81
5.8	繰り返しごとの推定誤差の推移 (Pose-G)	82
5.9	繰り返しごとの推定誤差の推移 (Pose-H)	82
5.10	オクトリーレベルの違いによるボクセルモデル作成に要する時間	83
5.11	カメラの視点とそれぞれのカメラで得られるシルエット画像	84
5.12	カメラの位置と台数が推定結果に与える影響	84
5.13	カメラ台数の違いによる手形状推定の結果	86
5.14	カメラの台数の違いによる推定誤差	87
5.15	カメラの位置	88
5.16	シミュレータで作成した手形状	88
5.17	それぞれのカメラ位置を用いた場合の推定誤差	90
5.18	カメラの位置の違いによる手形状推定結果	91
5.19	推定誤差の比較	92
6.1	バーチャルクレイ変形のフローチャート	95
6.2	direct FFD の過程	100
6.3	正六面体に対するサブディビジョンサーフェス	102
6.4	正六面体の各面を 4 分割した初期格子に対するサブディビジョンサーフェス	103
6.5	正六面体の各面を 16 分割した初期格子に対するサブディビジョンサーフェス	104
6.6	実カメラシステムにおける手の動作可能領域	106
6.7	ろくろによるバーチャルクレイの変形	107

6.8	手領域を厚みのないへらとして扱う	108
6.9	推定に用いる座標系	109
6.10	位置・姿勢の処理の流れ	110
6.11	o-patchに発生する力とモーメント	111
6.12	クレイとのインタラクション	112
6.13	手の位置・姿勢によるバーチャルクレイの変形	113
6.14	指先によるバーチャルクレイの変形(1)	114
6.15	指先によるバーチャルクレイの変形(2)	115

表 目 次

3.1	各指の自由度配置	28
3.2	手首の自由度配置	29
3.3	リンクパラメータ	29
3.4	交差判定の高速化	38
3.5	平均推定時間	54
6.1	3次元オブジェクトのモデリング方法の変形における比較	98

第1章 序論

1.1. 研究の背景

近年のコンピュータパワーの向上とコンピュータグラフィックスの発展に伴って、コンピュータの内部での表現空間は2次元空間から3次元空間へとその次元を上げてきた。2次元表現空間では、空間中のオブジェクトはその次元を落とすことなくグラフィックディスプレイに表示され、ユーザーは直接その形状を、マウスやタブレットなど従来のポインティングデバイスを用いて、作成・変形することができる。しかし3次元形状表現空間においては、空間中のオブジェクトは次元を落としてディスプレイに表示されてしまう。そのためユーザーが従来のポインティングデバイスを用いてオブジェクトとのインタラクションを行う場合、オブジェクトの形状を頭の中で再構成しながら作成・変形等の操作を行わなくてはならない。また、従来の一般的な3次元システムにおけるコマンドベースでの形状入力や変形操作は、物を作るときの手動操作をコマンドでどのように記述するかを考える必要があり直感的でない。

例えば、従来の設計・製造業務に用いられている3次元CAD/CAMシステムや3次元モデリングシステムにおいて、3次元自由形状を入力する作業は非常に困難で、ユーザーにとって大きな負担となっている。また、コンピュータの高性能化と低価格化に伴って、トレーニングされたオペレータによるシステム操作ではなく、設計技術者や意匠デザイナーが直接3次元モデリングシステムを用いて、設計・デザインを行うことが普通になっている。これは、一般の(オペレータとしてトレーニングされていない)ユーザでも、直感的に操作を行うことの出来るインタフェースの必要性を示唆している。

このように、3次元システムにおいて現実世界のオブジェクト操作と同様の方

法で仮想空間のオブジェクトを操作できる，直接的でかつ直感的なインタフェースの構築が望まれている．このような要求は昨今に始まった事ではない．3次元システムへ移行しだしたところから，マンマシン・インタフェースの重要性は高く，より使いやすいシステムへの様々なアプローチがこれまでに研究されている．しかし，それらの研究にも係わらず，現在の3次元システムを用いるユーザーには「入力インタフェースをどうにかして欲しい」という不満を持つものも少なくない．このように，3次元システムにおける直接的かつ直感的な入力インタフェースの研究はまだ発展途上であり，現実的な手法の提案を行うことは，産業界における効果のある研究分野であると言える．

1.2. 研究の目的

「人間が物を作るときには手を使う」ことから，このようなインタフェースを構築するためのアプローチのひとつとして実空間での手動作を認識しインタフェースとして用いる方法が考えられる．人間の手動作は，言葉，感情の表現，形状の表現などのコミュニケーション手段として，また道具や形状の操作手段として多くの機能を有している．この大きな表現力を持つ人間の手動作を，コンピュータと人間の自然なインタフェース手段として応用することは，有効なアプローチであるといえる．

製品設計やデザイン等を行う実務環境では，3次元システムを長時間にわたって使用することが多い．そのような使用環境でのインタフェースとしては，データグローブやカラーグローブまたは力覚提示デバイス(例えば PHANToM Haptic Interface)などの接触型のインタフェースよりも，何も装着する必要のない非接触型のインタフェースのほうがユーザーの拘束感が少ないことから好ましい．

そこで本研究では，操作的な手動作を3次元システム(e.g. バーチャルクレイモデリングインタフェース，VR環境でのオブジェクト操作，ロボット教示システム等)の入力インタフェースとして使用することが可能な，非接触かつリアルタイムでの任意の手形状推定手法の提案を目的におく．その目的の元で，推定のための新しい手形状推定のアルゴリズムを提案・実装することにより実環境で

の動作を確認する。

その上で、本研究で提案する手形状推定手法のアプリケーションへの応用性を検証するために、製品デザインや仮想空間中での陶芸などの芸術作品を作成するための、バーチャルクレイモデリングシステムのプロトタイプを作成する。このバーチャルクレイモデリングシステムは、粘土のような自由に変形できる柔らかな塑性体を仮想空間上でモデル化し、このモデルを人間の手動作によって変形させるものである。このような3次元システムのインタフェースは、デザイナーの自由な発想を阻害することなく、簡単に3次元モデリングシステムでの自由形状の入力や変形が出来るため、本研究の開発背景である直接的で直感的なインタフェースが必要とされるアプリケーションであると言える。このシステムを介して、提案した手形状推定アルゴリズムの有効性と応用性の検証を行うことで、本研究の成果が現実的で実用可能であることを示す。

1.3. 人間の手動作の認識と推定に関する従来研究

1.3.1 対話的・操作的な手動作認識

Quekらはヒューマンコンピュータインタラクションの観点から手動作を図1.1のように分類している[1]。この分類法では、手や腕の動きはジェスチャー (Gestures) と無意識の動き (Unintentional Movements) に分けられる。そのなかでもジェスチャーは把持や形状作成などの操作的 (Manipulative) な動作と、対話的 (Communicative) な動作に分けられる。対話的な動作は、模倣 (Mimetic) や指示動作 (Deictic) などのようにその動き自身に直接的な意味を持つ動作 (Acts) と、言語や言葉の代わりとして用いられる動作 (Symbols) に分けられている。

この分類の上で手動作認識システムを考えると、対話的な動作を認識する事は、現在まで数多くの研究がされている。言語や言葉の代わりに用いられる動作に関しては、ハンドサインや手話などを認識するシステム [2, 3] があり、規定語の認識は現在実時間で可能である。また、動きそのものに意味を持つ動作を認識し、仮想物体などを操作するインタフェースとする研究も数多くされている [4, 5, 6, 7]。多くは、テンプレートマッチングによって形状を決定し、手の動きベクトルによ

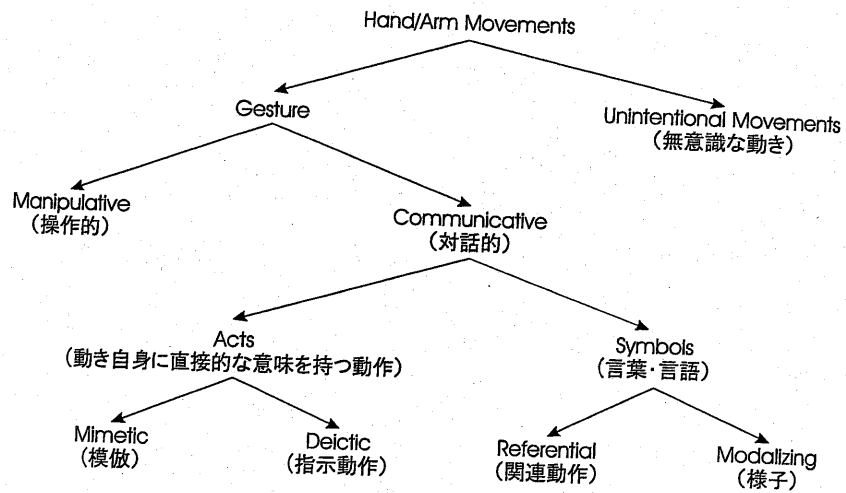


図 1.1 手動作の分類

り操作量などを決めている。例えば、内海らの両手手振りによる仮想空間操作では多視点動画像を用い、手の形状によって決まる8種類のコマンドとスライド操作で仮想空間の物体の操作をおこなっている [8]。また、Maggioniによる Gesture Computer [9] ではシルエット画像のモーメントと指先検出により手形状認識をおこない、インタフェースとして用いている。しかし、これらのアプローチで認識できる手形状は、あらかじめ決められた数種類の形状しか認識する事が出来ず、そのためインタフェースとしても従来のコマンドライクなものになってしまう。

ユーザーの手動作を、直接 VR 空間上での 3次元形状を操作する動作として用いるためには限られたパターンの手形状を認識するのみでは不十分であり、任意の手形状を認識 (図 1.1 では、操作的な動作 (Manipulative)) することが必要である。任意の手の形状を認識するためには、システム内に 3次元の手のモデルを持ち、関節角度や手首の姿勢などによって任意の手形状を表現する必要がある。

3次元形状操作などの入力インタフェースとして用いるために手形状を推定する場合、手の観測データを得るには大きく分けて2種類の方法がある。一つは、位置センサや角度センサ等の機能を内蔵したデータグローブやそれに力覚提示の機能を加えた Cyber Grasp 等を手に装着して、その位置や関節角度情報を直接計測する接触型の方法である。この手法では、位置や関節角度の測定が高精度で行えるという利点がある。舟橋らはデータグローブを装着して仮想空間の物体を把

持するなどの操作を行うインタフェースを構築している [10]. 指先の位置のみを計測し物体と接触した際の力覚を提示する PHANToM Haptic Interface の開発者の一人 Massie は, この力覚提示デバイスを用いて粘土モデルを削って形状を生成するインタフェースを構築している [11]. 少し方向性は異なるが, 北村らは物体とのインタラクションの複雑さを回避するために, 手と物体の間に道具を介したインタフェースの研究を行っている [12].

もう一つは非接触型の方法である. ユーザにとっては接触型の方法よりも拘束感が少ないこと, またデータグローブのような特別な計測デバイスではなく, CCD カメラのような一般的な計測デバイスを用いてシステムを構築することが出来る. 非接触型のインタフェースの観測方法として最もよく用いられている手法は, カメラによる画像を入力としコンピュータビジョンの技術によりその 3 次元形状を計測する方法である [1, 13]. また, 手のシルエット画像を安定して簡単に抽出するために赤外線カメラを用いているシステムもある [14].

通常のカメラ画像を入力とした手形状推定に関する研究については次節で詳しく述べる.

1.3.2 画像入力に基づく手形状推定

このような計測を行うためのシステムをカメラの台数によって分類してみると, 単眼カメラシステム・ステレオカメラシステム・多視点カメラシステムなどのカメラシステムが考えられる.

単眼カメラシステムを用いた手形状推定の研究例を見ていく. 亀田らは 1 枚の単眼視シルエット画像から, あらかじめシステム内部に構築された対象オブジェクトのモデルと 2 次元的にマッチングすることによって関節物体の姿勢推定をおこなっている [15]. 島田らは単眼視時系列シルエット画像によって, 手指の 3 次元形状モデルから生成した可能な姿勢候補の中から, 入力されたシルエット画像にもっともよくマッチするものを探すことで形状推定をおこなっている [16, 17]. 島田らはまた, この手形状推定を PC クラスタを用いることによってリアルタイムに行っている [18]. 岩井らは単眼動画画像から手のモデルを生成し, 画像から得られた特徴量を用いてエネルギー最小化探索を行う手法で手の運動と手形状を推

定している [19]. 遊佐らは単眼静止画像を入力とし, その2値画像と細線化画像を生成した上で指先特徴からの手形状推定を行っている [20]. 中嶋らはマーカーを付けた手の単眼動画を入力とし, モデルマッチングによって形状推定を行っている [21]. 渡辺らはカラーグローブを使用することにより各指節を明確に切り出し, ロバストに指文字を認識する手法を提案している [22]. これら単眼カメラシステムを用いた形状推定システムでは, 距離データを得ることが出来ないこと, またいずれの場合もオクルージョンの影響は避けられず, これらの問題を回避するために様々な手法の提案を行う必要がある.

RehgらのDigitEyesシステムでは27自由度の手モデルのトラッキングをステレオカメラシステムを用いて約10Hzの処理速度で行っている [23]. Delamarreらは, ステレオ画像を用いて手の見えている面の表面を再構成し, その面に対して3次元的にモデルフィッティングを行うことで手形状推定を行っている [24]. Cipollaらは, キャリブレーションしていないステレオカメラを用い, Active Contourを用いてロボットガイダンスのための指差し位置と方向をトラッキングするシステムを構築した [25]. ステレオカメラシステムは, 距離情報を得ることが出来るが, ステレオマッチングの際の誤対応によってその精度が不安定になる. また, 対象物体を一つの方向からしか見ないので, 見えない部分も依然残ったままである.

対象物体をなるべく多くの方向から見るためには, 多視点カメラを用いるのが効果的である. 多視点カメラを用いたシステムでは, 高橋らが正面, 側面, 上面に視点を設定した3眼視による人物姿勢の推定を行っている [26]. 更に内海らは6台のカメラを用い, 最もオクルージョンの少ない3視点を動的に選び推定を行っている [8]. また大野らは複数のステレオカメラを用いて, 3次元テンプレートマッチングにより手のトラッキングを行っている [27]. カメラ台数が多ければ多いほどオクルージョンの影響は少なくなるが, 計算量は増加する.

1.4. 本研究の位置付け

1.4.1 非接触型インタフェース

本研究では、従来の入力デバイスを用いたインタフェースよりも自然な入力インタフェースを構築するための手形状推定を目指している。応用分野の一例は、実務でのコンピュータを用いた3次元設計業務において、入力インタフェースとして用いることである。そのため、接触型や力覚提示型デバイスを用いた接触型インタフェースではユーザーにとって拘束感が強いため、自然なインタフェースであるとは言い難く、非接触型のインタフェースのほうが望ましいと言える。また、形状入力インタフェースとして用いるためには、手動作分類における操作的な動作を行う際の手形状を認識する必要がある。この際の手形状はある決まった種類の手形状ではなく任意の手形状である。しかし従来研究では、このような実時間処理が可能なビジョンベースによる任意の手形状認識手法はまだ確立されていない。

本研究では操作的な動作を入力インタフェースとして用いるために、実時間処理が可能なカメラ画像を入力とした任意の手形状推定を行う手法の提案を行う。ここで、“任意の手形状推定”とは、手指の各関節の回転角度を定量的に推定するということである。

1.4.2 多視点カメラシステム

1.3.2節で、非接触型の方法により観測データを得るためのカメラシステムとして、単眼カメラシステム・ステレオカメラシステム・多視点カメラシステムなどのカメラシステムが考えられることを述べた。単眼カメラシステムを用いて手形状を計測する場合、キャプチャや計算に必要な時間が少なくリアルタイム性に富むという長所を持つ反面、画像から奥行き情報を得ることができないという欠点がある。またステレオカメラシステムでは奥行き情報を得ることができるため単眼カメラシステムの欠点は除去することができる、しかし画像間のマッチングをとる必要があり、避けることの出来ないミスマッチングに起因する精度の低下や

オクルージョンなどの問題が新たに発生する。しかし、複数のカメラを用いた多視点カメラシステムでは、オクルージョンの影響は非常に小さくなる。さらに、得られた画像群から3次元形状を再構成する方法を使えば、画像間のマッチングを取る必要がなくなり、カメラシステムに起因する精度の低下も避けることが出来る。またこの方法ではカメラ間の位置関係を知る必要はなく、3次元空間中の位置が各カメラ画像のどこに射影されるかを知るだけでよい。そのためそれぞれのカメラ配置は自由度が非常に大きくなる。

本研究ではこれらの理由から、多視点カメラシステムを用いて手形状を計測する。

1.4.3 モデルベースの形状推定

ビジョンを用いた一般的な物体認識には大きく2つのアプローチがある。一つはモデルベース(model-based)と呼ばれる3次元特徴モデルでの物体認識である。モデルベースアプローチは、物体の幾何モデルやCADモデルのような幾何学的に表現された対象物体の特徴点と、計測された画像から抽出した特徴量との対応関係を求めることにより物体認識を行う手法である。もう一つの手法はアピアランスベース(appearance-based)と呼ばれる、画像および画像特徴レベルでの照合である。アピアランスベースアプローチは、物体形状などの特徴量を抽出して、計測した画像とあらかじめ記憶しておいた画像データベースとの単純な比較を行い、どの画像と最も似ているかということにより認識を行う手法である。形状推定の対象が自由度の大きい物体であると、アピアランスベースアプローチを適用するためには膨大な量のデータベースが必要となる。このことから、自由度の大きい手の形状を推定するためにはアピアランスベースアプローチよりもモデルベースアプローチのほうが適しているといえる。

本研究では、システム内にあらかじめ作成された3次元の手形状モデルと、多視点カメラシステムによって得られた画像から3次元データとして再構成された観測データとの、3次元でのモデルフィッティングによりその形状を推定するアプローチを提案する。このような形状推定手法は新しい試みであり、本論文で提案する手形状の推定手法は、ビジョンベースの手形状推定手法においてオリジナ

リティの高い手法であると言える。

1.5. 研究がもたらす効果

多くの手動作を用いたインタフェースは、1.3節で述べたように、あらかじめ決められた手形状をコマンドの代わりとして用いているものが多い。しかし本研究で提案するインタフェースでは人間の手動作はコマンドとして取り扱われるのではなく、実空間で物を作るときの動作そのものとして取り扱うために、初心者でも簡単に形状操作を行うことが出来るインタフェースとなることが期待できる。また、従来多く研究されている接触型入力デバイスを用いたインタフェースに比べて、本研究で目標にする非接触型インタフェースは、ユーザーの拘束感が少なく、かつ両手や複数人での協調作業へもデバイスを追加することなく拡張できる。このことから、実際の意匠設計業務で用いる3次元システムの入力インタフェースとして有用かつ現実的な手法の一つとなることが期待できる。

近年ますます低価格化が進む3次元モデリングシステムのインタフェースツールとして、依然高価な接触型の手形状計測デバイスを採用することは、システム構成のバランスを欠いているといえる。本研究で使用する観測システムのカメラは、必要な精度を満たすのに十分な解像度を満たすものであればどのようなものを使っても構わない。キャリブレーションの手間こそ必要であるが、カメラシステムとキャリブレーションツールさえあれば、観測と推定を行うことが出来、コストパフォーマンスは非常に良い。そのためインタフェース研究やバーチャルリアリティの大掛かりなシステムの一環に位置するだけでなく、デザイン事務所や設計事務所など小規模の実務現場でも実用可能なシステムと成りうる。

また、カメラを用いて非接触で手形状の推定を行うため、本研究で検証するバーチャルクレイモデリングシステムの入力インタフェースとしての応用だけでなく、キャリブレーションに多大な時間を要するデータグローブ状のコントローラを用いるヒューマノイドロボットのテレオペレーションシステムにおける、マスタスレーブシステムのマスタ側としてユーザの手の動きを伝える有用なツールにも成ることが期待できる。

このようなことから、本研究は実務環境で実用可能な手形状推定システムとなりうるという効果をもたらすことが出来る。

1.6. 本論文の構成

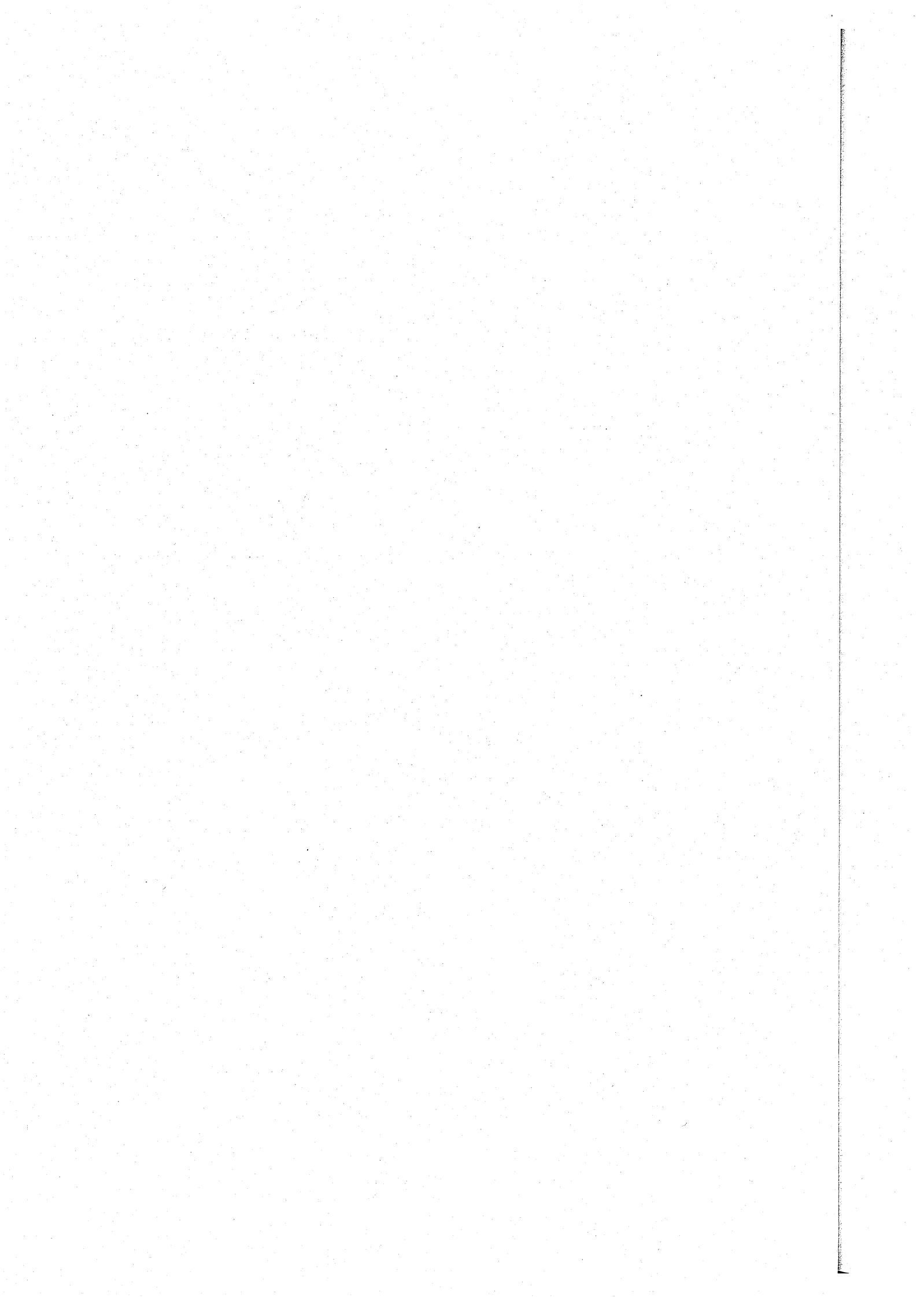
本論文は以下の7章より構成される。

- 第1章: 序論 本研究の背景, 目的, 本研究がもたらす効果を明確にした。さらに, 従来より数多く行われてきた人間の意識的な手動作の際の手形状を推定する研究を整理/分類し, 各々の研究の位置付けを明確にし, その上で本研究の位置づけを明確にした。
- 第2章: 人間の手の機能と操作的な手動作の解析 推定対象となる人間の手の構造について説明を行い, 推定対象となる人間の操作的な手動作を画像より解析し, 本研究で提案する手形状推定の目標スペックを明確にする。
- 第3章: 多視点シルエット画像を用いた手形状推定手法の提案 本研究のキーアイデアとなる多視点画像に基づく手形状推定手法の提案を行う。提案手法では, 観測空間中の手の多視点画像を統合することによって3次元の観測データとして再構成し, あらかじめシステムがもつ手の表面形状データとの3次元フィッティングにより関節角度の推定を行う。シミュレーションにより提案手法の評価を行い, その上で多視点動画像を入力とした連続的な手形状の推定実験を行う。
- 第4章: 粗密探索法を導入した多視点シルエット画像を用いた手形状推定 インタフェースの入力手段として手形状推定を用いるためには, 推定速度と推定精度の向上が不可欠である。それらを実現するために導入した手形状表面データの新しい表現方法と粗密探索法による手形状推定の改良について論じる。
- 第5章: 手形状推定手法の性能評価 提案した多視点シルエット画像を用いた手形状推定手法の性能評価を行う。特に, 観測データの再構成の精度とカメ

ラ位置についての評価を行い，提案手法の特徴について論じる．

第 6 章: コンピュータインタフェースとしての手形状推定の応用 インタラクションの対象となる塑性体の一つである粘土のモデル化を行い，推定した手形状を用いてバーチャルクレイを変形するシステムのプロトタイプを構築し，バーチャルクレイモデリングインタフェースの入力として，提案手法を用いることの有効性の検証を行う．

第 7 章: 結論 本論文を結ぶ．本研究を通して得られた知見を総括し，今後の展開/可能性を述べる．



第2章 人間の手の機能と操作的な手動作の解析

2.1. 本章の趣旨

本章では推定対象となる人間の手の構造と運動学的な観点からの手の機能についてと、推定対象となる操作的な手動作について概観していく。

最初に、人間の手の構造について関節の名称などの定義と、指の運動の名称の定義などを行う。次に、本研究で推定対象とする人間の操作的な手動作の際に、人間の手形状がどのように変化するかを画像とデータグローブを用いて解析する。特に、粘土細工を行う際の手形状の変化について詳細に確認する。最後に、操作的な手形状の解析結果より本研究で提案する手形状推定の目標スペックを明確にする。

2.2. 人間の手の構造と運動

はじめに、推定対象となる人間の手の構造 [28, 29, 30, 31] について概観していく。

「手」とは、手首の関節より先の部分をいう(図 2.1)。これには手根と中手と指の3つの区分がある。手根と中手は区分がはっきりしないので、この2つを合わせた部分の前面を「手掌(てのひら)」といい、反対側を「手背(てのこう)」という。手全体の母指側の縁は橈側縁、小指側の縁は尺側縁と呼ぶ。日本語では「指」は5本の指すべてを指す。しかし母指が形態的にも機能的にもその他の指と異なっていることから、英語では母指が thumb, その他の指が finger で表されることが多い。母指以外の指は3つの区分を持ち、付け根の方から、基節、中節、

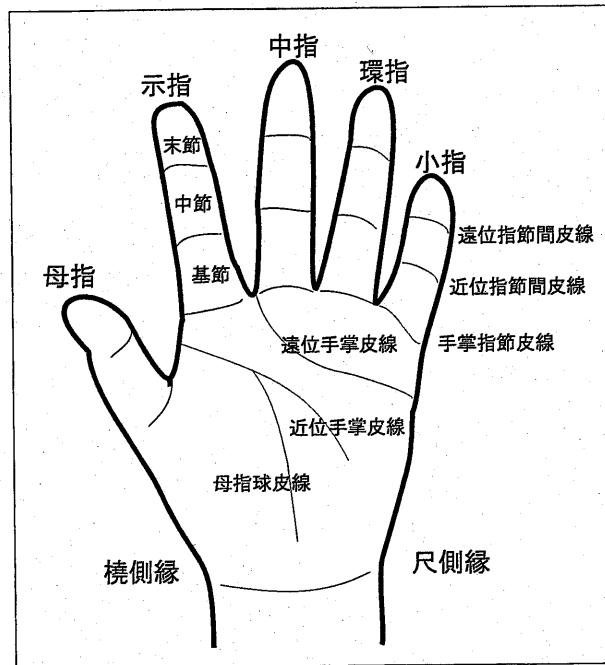


図 2.1 手の構造と皮線

末節と呼ばれる。母指は2区分しかなく、基節、末節のみである。

手の皮膚には運動によって生じたと思われる多数のしわがある。これを皮線という。手掌側面では特に明瞭である。

母指球皮線は、示指基部の手掌の橈側縁にはじまり、母指球の尺側縁を弧を描きながら走り、手首の中央で終わる。遠位手掌皮線は、小指の基部の手掌の尺側縁に始まり、ゆるやかな曲線を描きながら斜めに走って、示指と中指の間で終わる。近位手掌皮線は母指球皮線と同じ場所から始まり、小指球のほぼ中央に向けて斜めにまっすぐ走る。指を伸ばしたままつけ根で曲げてみると、遠位手掌皮線と近位手掌皮線の橈側寄りの部分がいっそうはっきりと刻まれる。

図 2.2 は手の骨格標本を示すものである。母指の支柱としては2個の、その他の指の支柱としては各々の3個の指節骨がある。遠位より、末節骨、中節骨、基節骨ともいう（母指は末節骨、基節骨のみ）、中手は中手骨によって支えられる。これは手背から容易に触れることができる骨である。母指につながる中手骨を第1中手骨といい、以下橈側から順に、第2中手骨、第3中手骨、第4中手骨、第

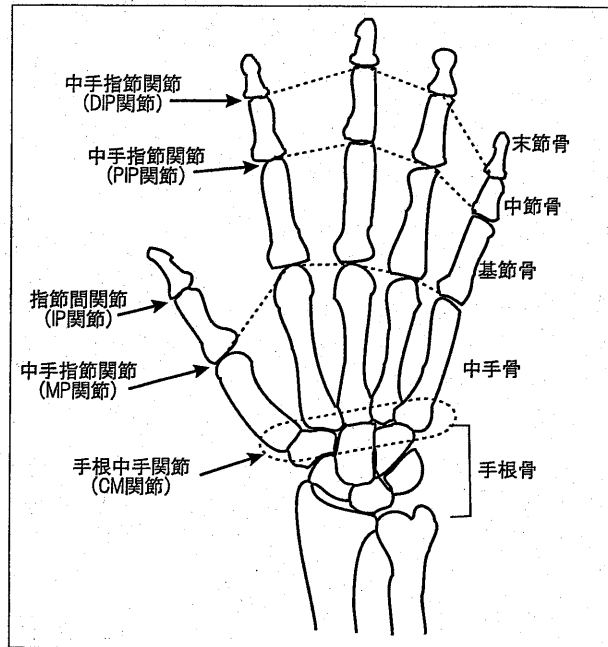


図 2.2 手の骨格

5 中手骨と呼ぶ。各指ごとの指節骨と中手骨のひとつながりを指列という、関節の名は先端より、遠位指節間関節 (distal interphalangeal joint, 以下 DIP 関節), 近位指節間関節 (proximal interphalangeal joint, 以下 PIP 関節), 中手指節関節 (metacarpophalangeal joint, 以下 MP 関節) という。DIP 関節は遠位指節間関節皮線よりもやや遠位に, PIP 関節は近位指節間皮線のほぼ真下に, MP 関節は中手指節皮線と遠位手掌皮線尺側部分および近位手掌皮線橈側部のちょうど間にある。いわゆる手首, すなわち手根部には 8 個の手根骨があって, 靭帯によって密に連結されている。手根骨と中手骨がなす関節を手根中手関節 (carpo-metacarpal joint, 以下 CM 関節) という。

手には多数の関節があり, どのような複雑な手の動きも関節運動の組み合わせであらわすことができる。母指とその他の指の関節運動は図 2.3 のようなものがある。屈曲と伸展はすべての DIP, PIP, MP 関節において生じる。また外転と内転は, 母指の CM 関節と他の指の MP 関節で生じる運動である。対立は複合運動とみなされる。

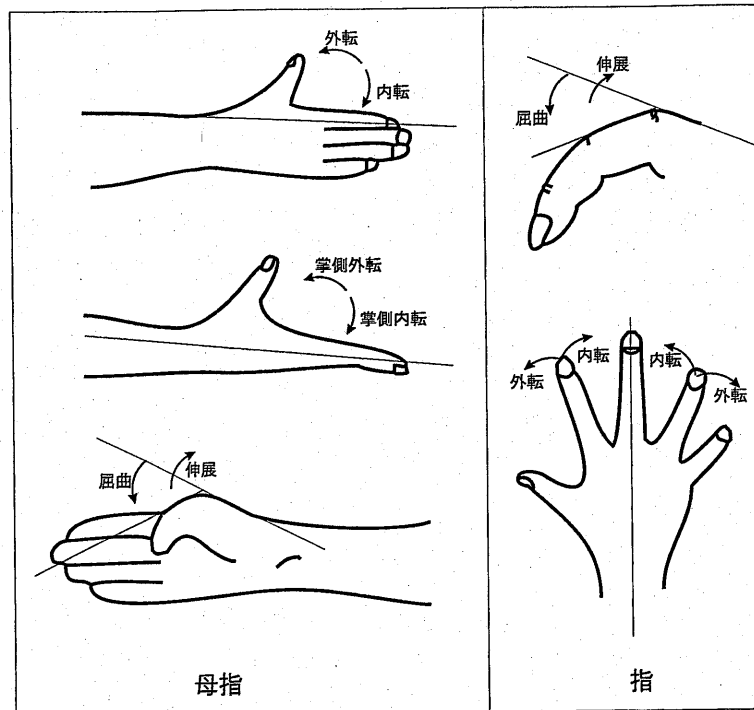


図 2.3 手の関節運動

2.3. 操作的な手動作の解析

人間の手動作には、対話的な手動作と操作的な手動作があることを1.3節で述べた。本節では、人間の手動作がどの程度の速さで行われるのかを、画像を用いて確認する。

まず最初に、対話的な動作の一つである“じゃんけん”について、0.5秒間の手形状の変化を33msec毎(ビデオレート)ごとにキャプチャした画像群を図2.4に示す。“グー”から“パー”に変化する瞬間に大きな画像の変化が現れていることが分かる。このような形状変化には、キャプチャ速度がビデオレートでは足りず、更に推定時間も33msec以下で行う必要があり、非常に高速な処理手法が要求される。

次に操作的な手動作である“物体把持”，“折り紙”，“粘土細工”，“ろくろによる形状作成”を複数人数で行った。その中で“物体把持”，“粘土細工”の例をじゃんけんのときと同様に、図2.5と図2.6に示す。図2.5の物体把持の際の手形状変

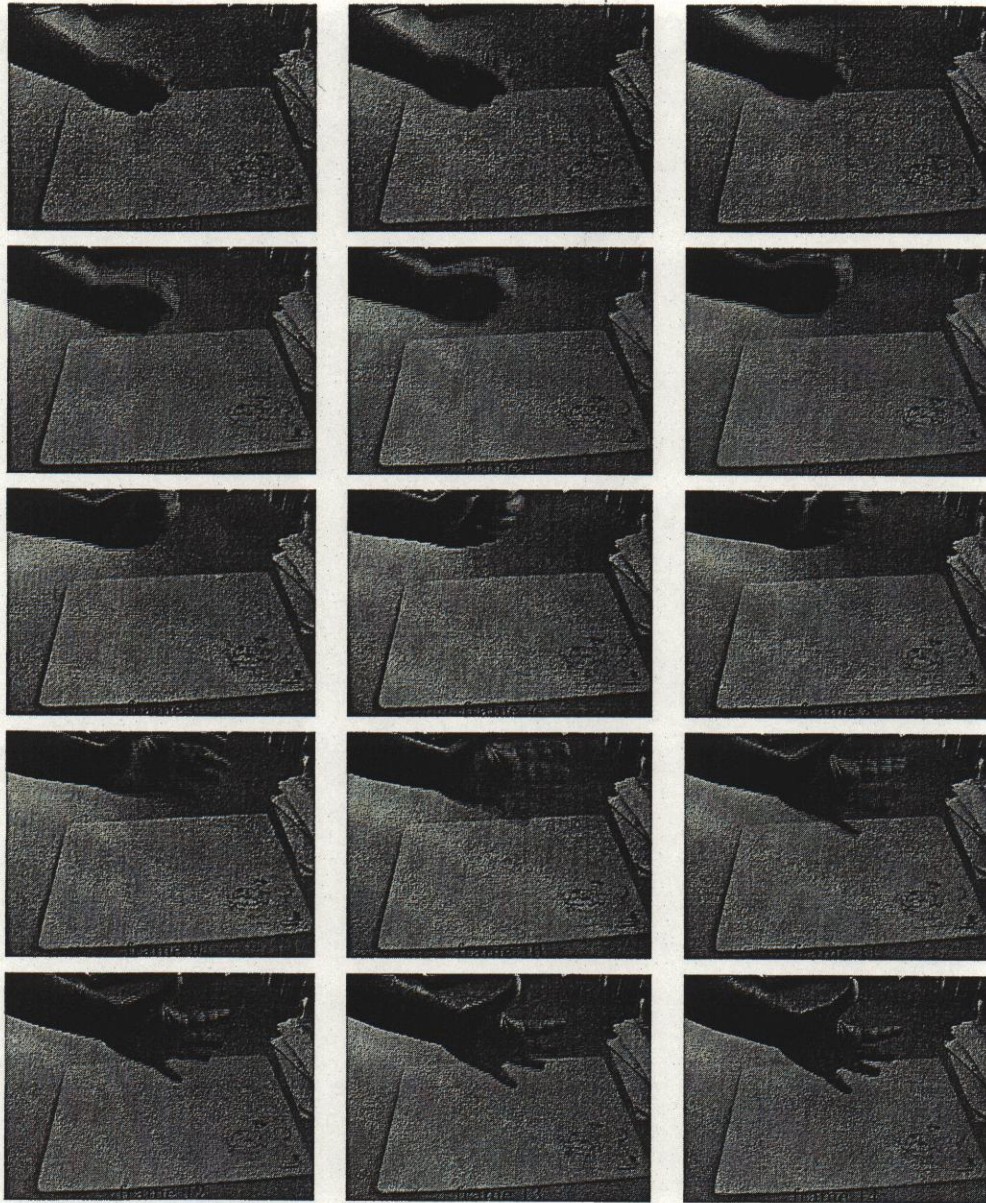


図 2.4 じゃんけんの手動作 (0.5 秒間)

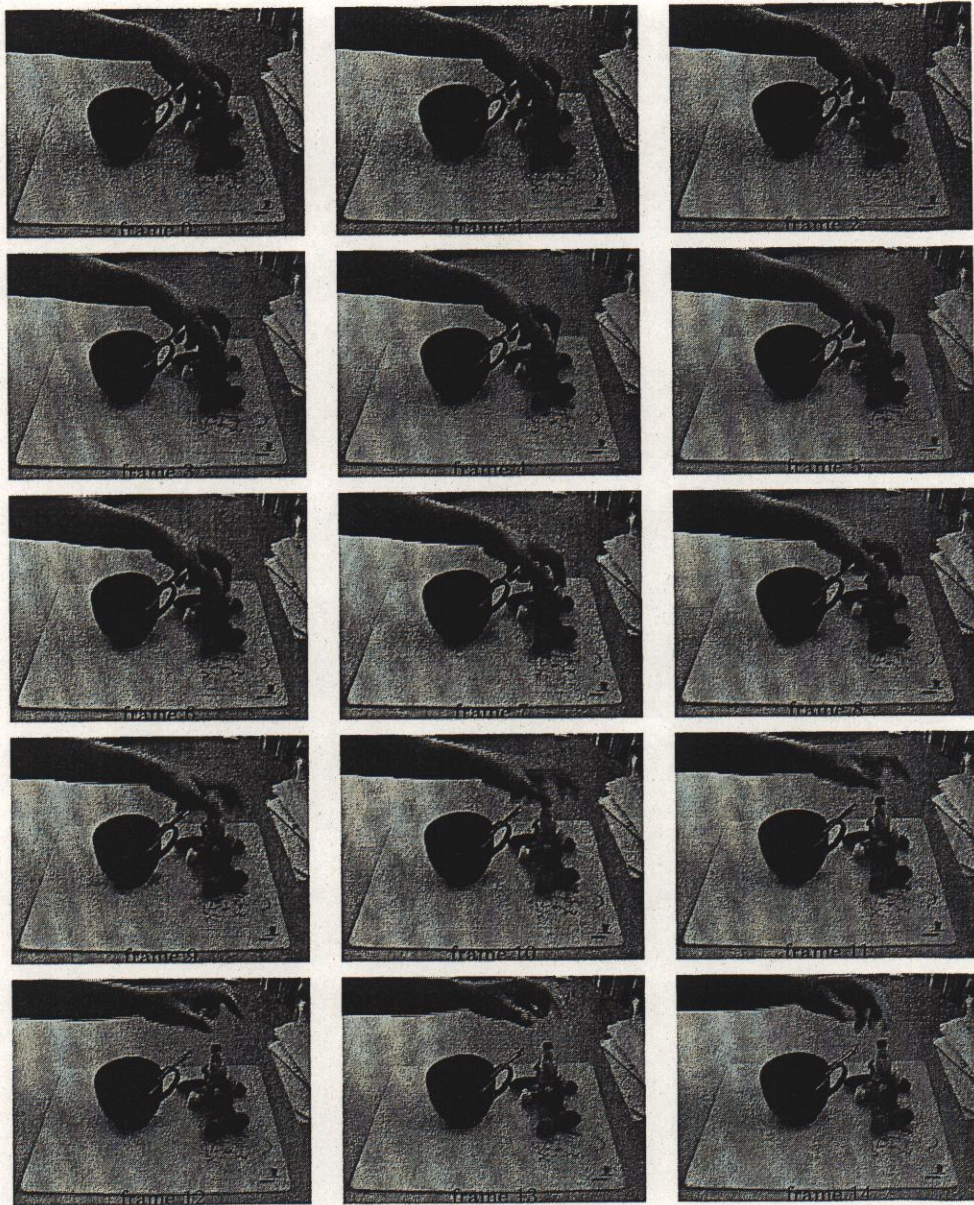


図 2.5 物を把持する手動作 (0.5 秒間)

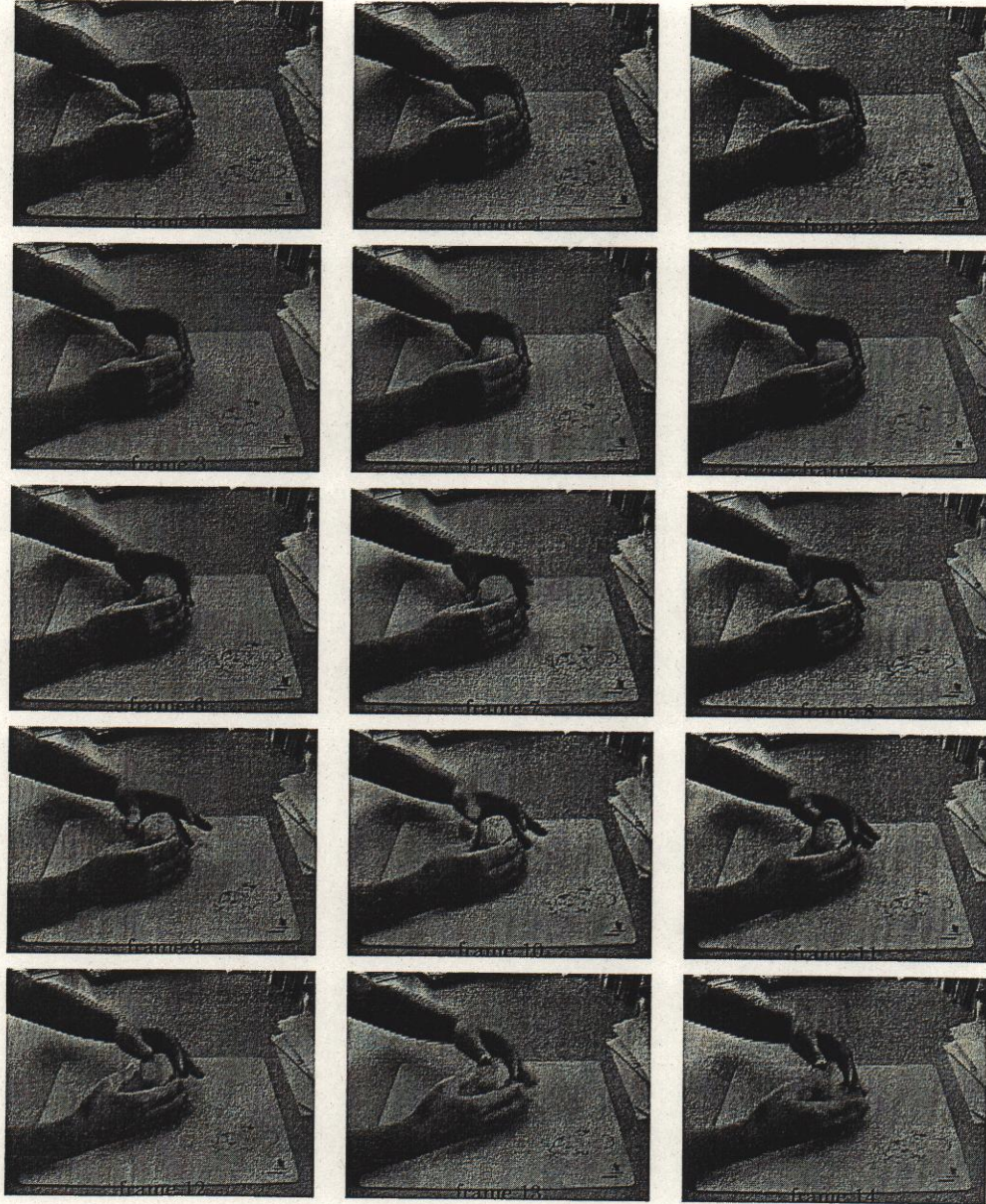


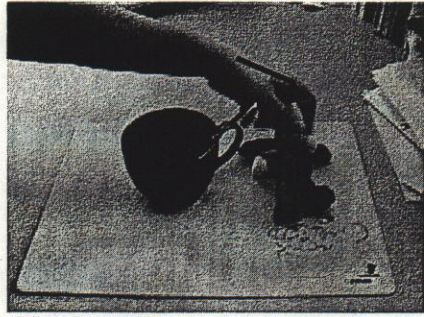
図 2.6 粘土でコップを作る手動作 (0.5 秒間)

化では、その速度が比較的速いため、指先部分の画像に大きなブレが生じているのが分かる。このブレを回避するためには、対話的な動作であるじゃんけんのと
きと同様高速なキャプチャと推定処理が必要になる。一方、図 2.6 の粘土細工の
際の手形状変化は、じゃんけんや物体把持に比べて緩やかな形状変化であること
が分かる。また指先画像の大きなブレが生じている部分はほとんど見られない。

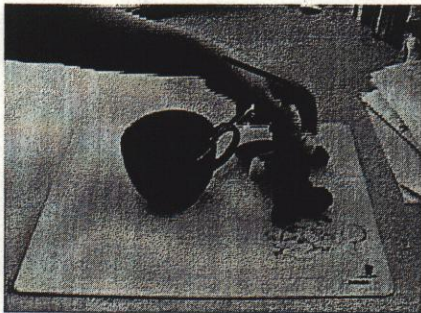
次に、物体把持、粘土細工とろくろによる形状作成についてキャプチャ間隔の
違いによって、どの程度手形状の変形の大きさに違いがあるのかを、基本フレー
ムと 1・2・3・6・10・15 フレーム後（それぞれ処理時間 30, 15, 10, 5, 3, 2[Hz] に
対応する）の間の画像の違いから調べる。図 2.7, 図 2.8, 図 2.9 にそれぞれの手
動作における、基本画像と各時間後の画像を示す。図中には示指の MP, PIP 関
節の屈曲の様子を明確にするために、黒い補助線を追加した。この補助線を用い
て、MP 関節屈曲角度を計測した結果を図 2.10 に示す。この手形状変化の例は、
一例に過ぎないのですべての操作的な手動作に当てはまると言い切ることは出来
ないが、フレーム間隔が大きくなるに従って、関節角度の変化も大きくなってい
ることが分かる。また、ろくろによる形状作成のように操作対象の物体を動かし、
緩やかな手形状の変化を用いて実現する方法では、通常粘土細工と同様に、フ
レーム間隔が小さいときは手形状の変化も少なく、大きくなると手形状の変化も
大きい。全体の手形状変化は小さく、より緩やかな手形状の変化であることが
わかる。

次に、これらの手動作について右手のみではあるが、データグローブを用いて
手動作時の関節角度の変化の計測を以下に示す条件のもとで行った。

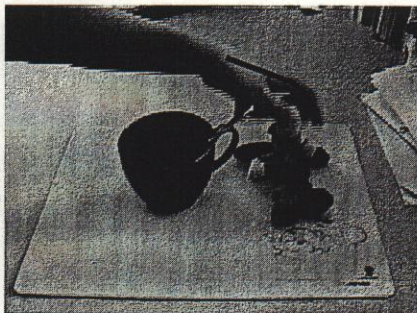
- 被験者は 10 人
- 右手にデータグローブを装着
- “じゃんけん”, “物体把持・移動”, “指先による粘土細工”, “ろくろによる
形状作成” の 4 種類の手動作を行う
- 各動作に要する時間は 6 秒～1 分程度
- 粘土細工, ろくろによる形状作成の際はデータグローブの上からゴム手袋
を装着



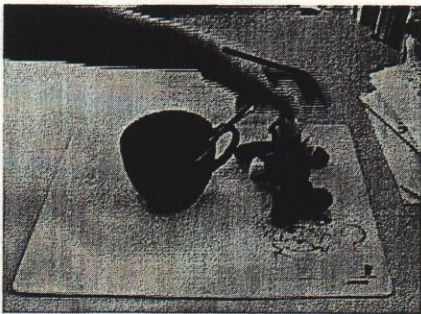
Base frame image



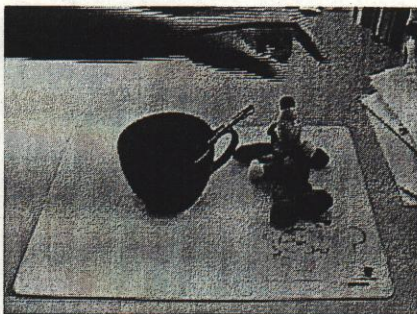
1-frame after



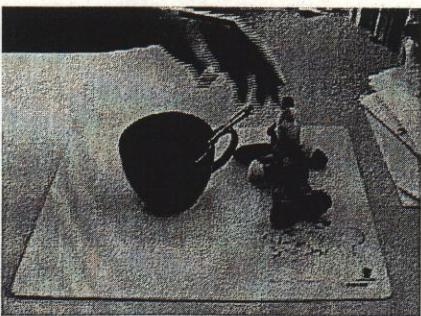
2-frame after



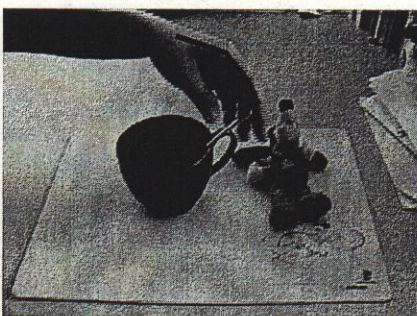
3-frame after



6-frame after



10-frame after

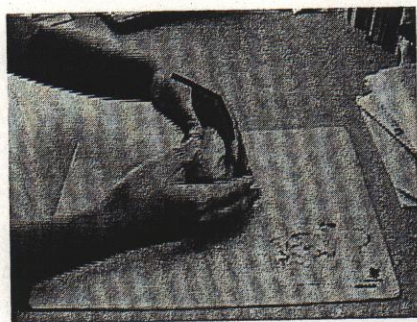


15-frame after

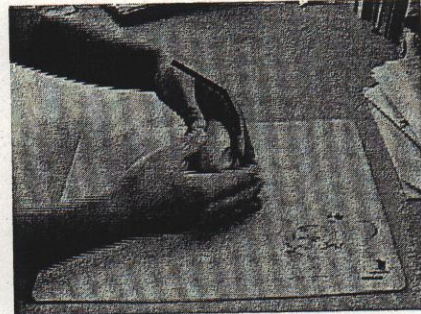
図 2.7 物を把持する際の手形状の変化



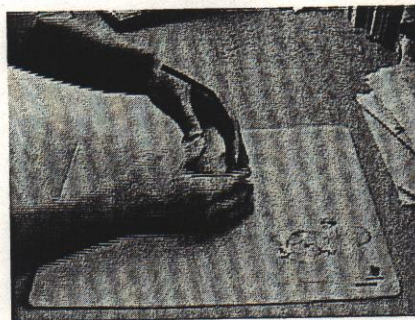
Base frame image



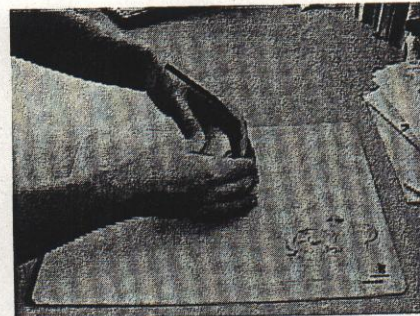
1-frame after



2-frame after



3-frame after



6-frame after



10-frame after



15-frame after

図 2.8 粘土細工の際の手形状の変化



Base frame image



1-frame after



2-frame after



3-frame after



6-frame after



10-frame after



15-frame after

図 2.9 ろくろによる形状作成の際の手形状の変化

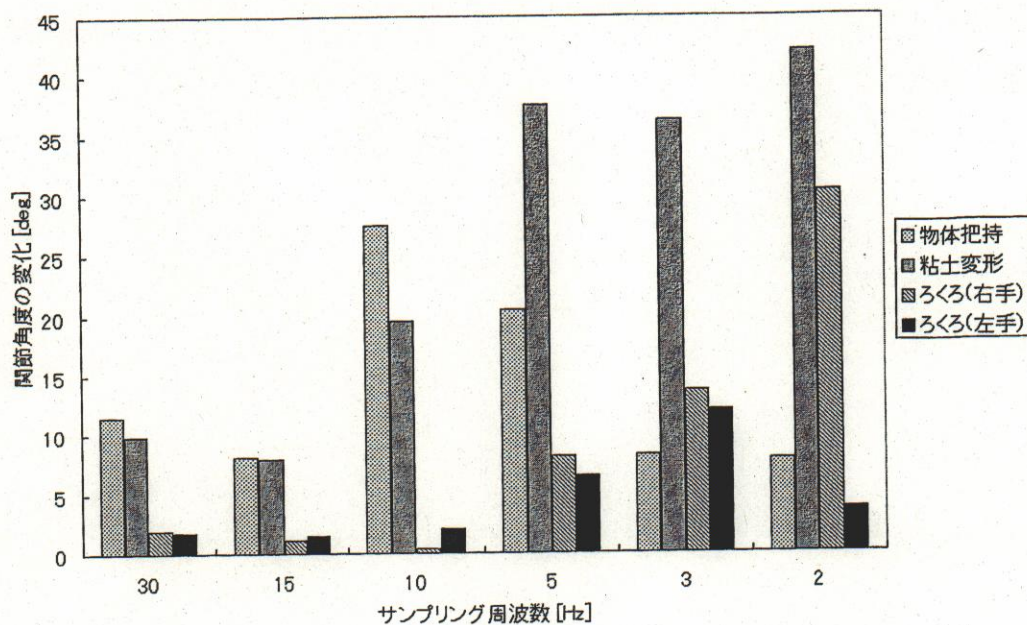


図 2.10 画像中の補助線により計測した MP 関節屈曲角度変化

観測した全員のデータを用いて、サンプリング周波数の違いによる、すべての動作の種類における示指 MP 関節屈曲角度の変化の平均値 (5[deg] 以下の変化は変化 0 として計算) を図 2.11 に示す. 対話的な手動作の一種であるじゃんけんについては、どのサンプリング周波数においても、関節角度の変化が飛びぬけて大きいことが分かる. これは、画像観察から得られた結果を裏付けるものであり、操作的な手動作に比べて関節角度は早く、かつ大きく変化することが示されている. また、図 2.12 に、じゃんけん以外の手動作における示指 MP 関節屈曲角度の変化の最大値を示す. サンプル数は約 10000 程度のうち、サンプリング間隔が 33[Hz] であれば最大 15[deg] の関節角度変化が、また 20[Hz] では最大 20[deg] 程度の角度変化が、サンプリング間隔が 5[Hz] 以下の場合には最大 35[deg] 以上の関節角度変化が生じていることが分かる.

これらの画像とデータグローブを用いた手形状変化の解析結果から、本研究では操作的な手動作の中における緩やかな手形状の変化を精度良く推定することを目標に置く. また処理速度も 20[Hz] 程度のサンプリング間隔であれば、形状変化

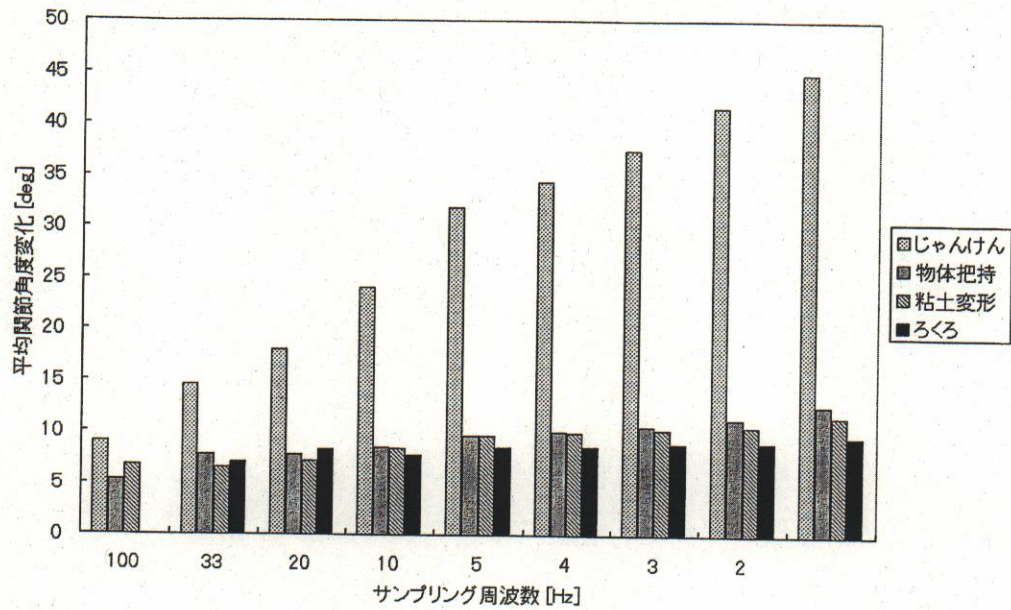


図 2.11 データグローブにより計測した平均の MP 関節屈曲角度変化

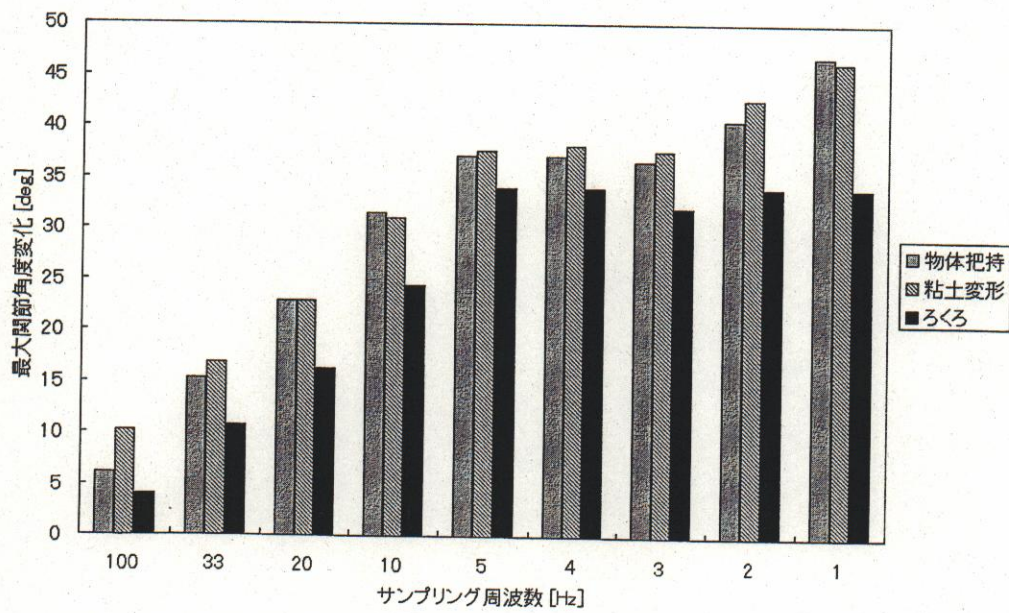


図 2.12 データグローブにより計測した最大の MP 関節屈曲角度変化

が比較的小さいことから、20[Hz]の推定速度を目標にする。また20[Hz]の処理速度が得られない場合は、サンプリング周波数が1~5[Hz]程度での関節角度変化を許容するために、関節角度変化が40[deg]程度あっても推定が可能であることを目標とする。

2.4. 本章の結論

本章では、まず最初に推定対象である人間の手の構造について、本論文で必要となる各部の名称と動作の説明を行った。

次に、本研究で推定対象とする人間の操作的な手動作の際に、人間の手形状がどのように変化するかを画像を用いて解析した。その結果、じゃんけんなどの対話的な手動作における手形状の変化は非常に高速であることが分かった、また操作的な手動作の中でも、物体把持は比較的高速に指先がの形状変化が生じることも示された。一方操作的な手動作の中でも粘土細工などの形状作成は比較的手形状の変化は緩やかであり、特にろくろを用いた形状生成は更に緩やかな手形状の時間変化を持つことが分かった。データグローブを用いて関節角度の変化について更に解析を行った結果、対話的な手動作の一種であるじゃんけんについては、操作的な手動作に比べて関節角度は早く、かつ大きく変化することが示され、画像観察から得られた得られた結果を裏付けるものであった。じゃんけん以外の手動作における示指MP関節屈曲角度の変化の最大値の観測結果からは、物体把持・移動と指先による粘土変形の際の関節角度の変化は画像から得られた結果よりも、粘土細工の際の関節角度変化は大きかったが、操作的な手動作時のサンプリング間隔と関節角度の変化の関係の目安は同じ傾向を示した。

最後に、操作的な手形状の解析結果より本研究で提案する手形状推定を、緩やかな手形状の変化を精度良く推定することに目標を定めた。

第3章 多視点シルエット画像を用いた手形状推定手法の提案

3.1. 本章の趣旨

本章では、本研究のキーアイデアとなる多視点シルエット画像を用いた手形状推定手法の詳細な説明を行う。提案する手形状推定手法は、あらかじめシステム内に構築した手のモデルと、3次元化された観測データの間の3次元的なモデルフィッティングをベースとしている。

最初に、システム内に構築する手のモデルのキネマティクスについて詳細に述べる。次に、複数のカメラから得られた画像を統合して、空間中に占める手の領域をボクセルを用いて表現する方法について詳しく述べる。その上で、この2つを3次元空間上で直接フィッティングさせる手法を提案し、その方法を詳細に述べる。次に、シミュレーションにより提案手法の評価を行い、提案手法が、操作的な手動作を行う際の手形状を推定するのに十分な機能をもっていることを示す。

更に、実際の手動作を観測するために4台のCCDカメラを用いた多視点カメラシステムを構築する。実カメラ環境では初期姿勢の推定が不可欠である、そのため手の形状の知識を利用してアドホックに行う方法について述べる。これらを行ったうえで、提案アルゴリズムによる多視点動画像を入力とする手形状推定を実装し、その動作を検証する。

3.2. 本章で用いる手のモデル

本研究で用いる3次元の手のモデルは、1) 骨格データ、2) 表面形状データの2種類のデータから構成されている。本論文では、この手のモデルを骨格モデルと

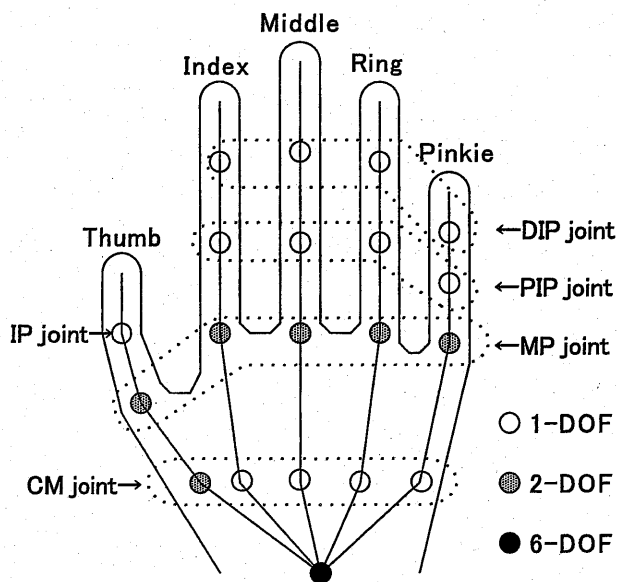


図 3.1 骨格データ

表 3.1 各指の自由度配置

	CM 関節	MP 関節	PIP 関節	DIP 関節	IP 関節	合計
母指	2	2	-	-	1	5
指	1	2	1	1	-	20 (指 4 本分)

呼ぶ。この骨格モデルは安室らが提案した手のモデル構造 [32] を基にしている。

3.2.1 骨格データ

2.2 節で述べたような手の骨格構造と関節運動を考慮した上で、本研究では手を手首に共通のベースを持つ 5 つのマニピュレータの集合としてモデル化する。各指は、リンク（骨）とジョイント（関節）の集合として図 3.1 のように表される。これにより手の動作はマニピュレータ解析の手法を用いて表現できるようになる。図 2.3 に示す運動を表すために、本研究での各リンクの自由度は図 3.1 の各ジョイントに示すように配置した。各関節の自由度は表 3.1, 3.2 に示すとおりである。手のモデル全体の自由度は手首の並進・回転自由度を含めて 31 である。

表 3.2 手首の自由度配置

	回転	並進	合計
手首	3	3	6

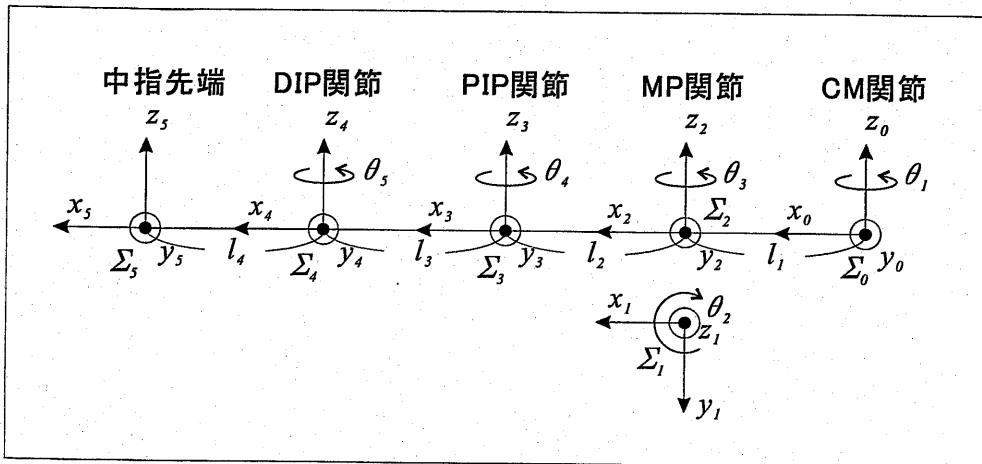


図 3.2 中指の関節軸

表 3.3 リンクパラメータ

ジョイント	θ	d	α	a
1	θ_1	0	$-\frac{\pi}{2}$	l_1
2	θ_2	0	$\frac{\pi}{2}$	0
3	θ_3	0	0	l_2
4	θ_4	0	0	l_3
5	θ_5	0	0	l_4

中指を例にとって、指の運動学について概観する。図 3.2 に示すように各関節の回転軸を設定し、各関節の相対座標系 Σ をリンクパラメータを用いて決定していく。指の各自由度における相対座標系のリンクパラメータは表 3.3 のようになる。

このようにして各々のローカル座標系を決定した上で、隣り合う回転座標系間の関係を示す同時変換マトリクス A_i は以下のように定義される。

$$A_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

例えば座標系 $\Sigma_1 \rightarrow \Sigma_0$ の変換マトリクス A_1 は以下のようなになる。

$$A_1 = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 \cos \left(-\frac{\pi}{2}\right) & \sin \theta_1 \sin \left(-\frac{\pi}{2}\right) & l_1 \cos \theta_1 \\ \sin \theta_1 & -\sin \theta_1 \cos \left(-\frac{\pi}{2}\right) & -\cos \theta_1 \sin \left(-\frac{\pi}{2}\right) & l_1 \sin \theta_1 \\ 0 & \sin \left(-\frac{\pi}{2}\right) & \cos \left(-\frac{\pi}{2}\right) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos \theta_1 & 0 & -\sin \theta_1 & l_1 \cos \theta_1 \\ \sin \theta_1 & 0 & \cos \theta_1 & l_1 \sin \theta_1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

このマトリクスを順次かけていく事によって、各関節の位置を一意に決定することができる。例えば PIP 関節の位置を求めるための変換マトリクス T は以下の式で与えられる。

$$T = A_1 A_2 A_3$$

$$= \begin{bmatrix} C_1 & 0 & -S_1 & l_1 C_1 \\ S_1 & 0 & C_1 & l_1 S_1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_2 & 0 & S_2 & 0 \\ S_2 & 0 & -C_2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_3 & -S_3 & 0 & l_2 C_3 \\ S_3 & C_3 & 0 & l_2 S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} C_1 C_2 C_3 - S_1 S_3 & -C_1 C_2 C_3 - S_1 C_3 & C_1 S_2 & l_2 (C_1 C_2 C_3 - S_1 S_3) + l_1 C_1 \\ S_1 C_2 C_3 - C_1 S_3 & -S_1 C_2 S_3 - C_1 C_3 & S_1 S_2 & l_2 (S_1 C_2 C_3 - C_1 S_3) + l_1 S_1 \\ -S_2 C_3 & S_2 S_3 & C_2 & -l_2 S_2 C_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Σ_3 座標系の原点座標 $(0, 0, 0, 1)^T$ を T にかけることで Σ_0 座標系での PIP 関節の位置が求まる。この場合は T の第 4 列目が PIP 関節の位置を表している。

3.2.2 表面形状データ

各関節角度が決まると、骨格の姿勢が一意にきまる。そこで手の像をレンダリングするために手の表面形状を表す皮膚の形状データが必要となる。手の表面形状は骨格姿勢の変化に応じて柔軟に変形出来なければならない。本研究では手の表面形状を多数の小さな3角形のポリゴンパッチで表現する。また各ポリゴンパッチの形状が骨格姿勢の変化に対応して変形できるように、ポリゴンパッチの各頂点データがどの骨リンクに対応しているのかを属性分けしている。

本章で用いる表面形状データは、骨格モデルの運動学をあらかじめ考慮に入れて作成した形状推定のための表面形状データである。そのため表面の滑らかさや、リアルさにやや欠けるという欠点がある。具体的には以下のような手順で作成している。

1. 手の実寸大の正面形状を二次元面上にトレースし、骨リンクがちょうど2次元平面上において各指の中心に位置するように配置する。また、骨格構造、皮線位置などを考慮して、各関節位置を指定する。
2. 手掌部の厚みは実測し一様な平板とする。指部の厚みは指幅と同じ直径をもつ円で表現する。また各指は複数の異なった直径を持つ円の連続から構成されているものとする。
3. 手掌上に存在する皮線を目安に、各関節の動きに伴ってどの皮膚領域が動くかを観察し、図3.3に示すように領域分割する。例えば、図3.3で「T-1」と示されている領域は母指(Thumb)の骨リンク1(CM関節からMP関節へのリンク)に関連している領域であることを表している。
4. 手掌部では各領域を等間隔に5ポイントプロットし形状データとする。また円の連続で表現される指部は各円周上を10等分するように等間隔にプロットし形状データとする。

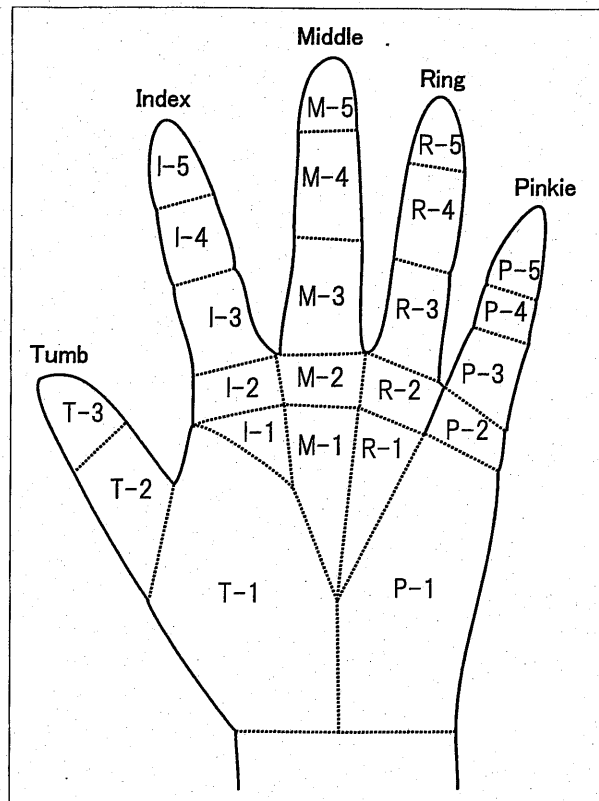


図 3.3 手の領域分割

3.3. 観測データの3次元化

3.3.1 オクトリーによる形状再構成法

本研究では、多視点カメラシステムを用いて得られた複数の2次元画像から、空間分割表現によって再構成された3次元形状を手の観測データとして取り扱う。このような複数画像を用いて形状を再構成する研究はこれまでも様々な研究が行われてきた [33, 34, 35, 36, 37, 38, 39]。この観測データを手のボクセルモデルと呼び、その構成方法 [40] について述べる。

コンピュータビジョンの分野において、複数の画像から3次元形状を復元する手法としては、画像の陰影情報を用いる Shape from shading, 画像の輪郭情報を用いる Shape from countour, 画像の動き情報を用いる Shape from motion, 画

像のシルエット情報を用いる Shape from silhouette, などの手法がある。今回採用した3次元形状再構成手法は、この中でも最も安定な画像の特徴量であるシルエット情報を用いた Shape from silhouette の手法である。

オクトリー表現

2次元画像がピクセルの集合で自然な画像を表現するように、3次元形状はボクセルと呼ぶ小さな立方体の集合体として表現することができる。複雑な形状を表現しやすい反面、大量のメモリと計算量が必要であるという欠点を持っている。この欠点を改良するために3次元形状を大きさの異なる立方体を組み合わせて表現する手法が、オクトリー表現である。オクトリー表現で用いられる各サイズの立方体をオクタントと呼ぶ。オクトリーは図3.4のように8分木による再帰的な構造になっている。オクタント（オクトリーの各ノードにあたる）はそれぞれデータとの関係を表す属性を持っている。この属性には3種類あり、データの中に完全に内包されている立方体は“BLACK”，データの完全に外側にある立方体は“WHITE”，データの一部にかかっている立方体は“GRAY”と定義されている。“GRAY”と定義された立方体は、更に8つの小さな立方体に分割され、それぞれに対して更に属性を持つ。この8分木の階層数によって形状の表現精度を変化させることが出来る。

シルエット画像の統合による形状再構成

n 点の任意視点の任意の一点を $j = 1, 2, \dots, n$ とし、視点 j で得られたシルエット画像を S_j とする。次に、対象物体をすべて内包する大きな“root-octant”を一つ定義し、レベル0とする。次に“root-octant”を8つの“sub-octant”に等分割しレベルを1つ増やす。この8つの“sub-octant”それぞれに対して、オクタントを図3.5のように、それぞれの視点に投影したシルエット画像 O_j と定義する。すべての視点において O_j と S_j のシルエット画像レベルでの交差判定を行う。図3.6に示すように、その結果すべての視点で O_j が完全に S_j に内包されていれば属性“BLACK”を、1つの視点でも完全に外であれば“WHITE”を、それ

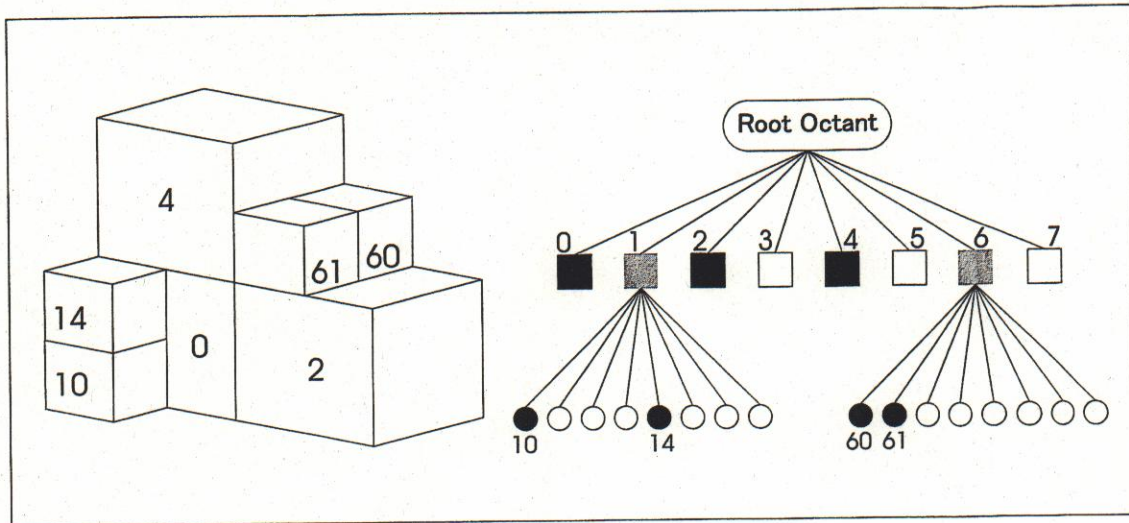


図 3.4 オクトリーモデルとその構造

以外であれば“GRAY”を各オクタントに与える。ここで“GRAY”となったオクタントは更に“sub-octant”に分割され交差判定が繰り返される。すべての視点で交差判定を行い指定レベルまで達した時点で復元処理は終了する。オクトリーのうち“BLACK”のオクタントが対象物体内部のオクタントであることから、このオクタントのみを表示することで対象物体の形状復元が可能となる。復元レベルによっては“GRAY”のままのオクタントがオクトリーの葉の部分で残る可能性がある。このオクタントもあわせて表示すれば対象物体を完全に外包するサイズでの形状復元が可能である。これらの処理フローを図 3.7 に示す。

交差判定の高速化

この方法では、この処理の中で最も重要な位置を占めている“オクタントを各視点に投影した画像 O_j と対象形状のシルエット画像 S_j の交差判定”にかかる処理時間が非常に大きく、リアルタイム処理へのボトルネックとなってしまう。Szeliski は、オクトリー表現を用いる場合において、非常に効果的な交差判定法を提案している [41]。その手法の概要を以下に述べる。

オクタントを 2 次元平面に投影するとその形状は一般的に 6 角形になる。この 6 角形とシルエット画像を交差判定を直接おこなうと、6 角形の各辺とシルエット

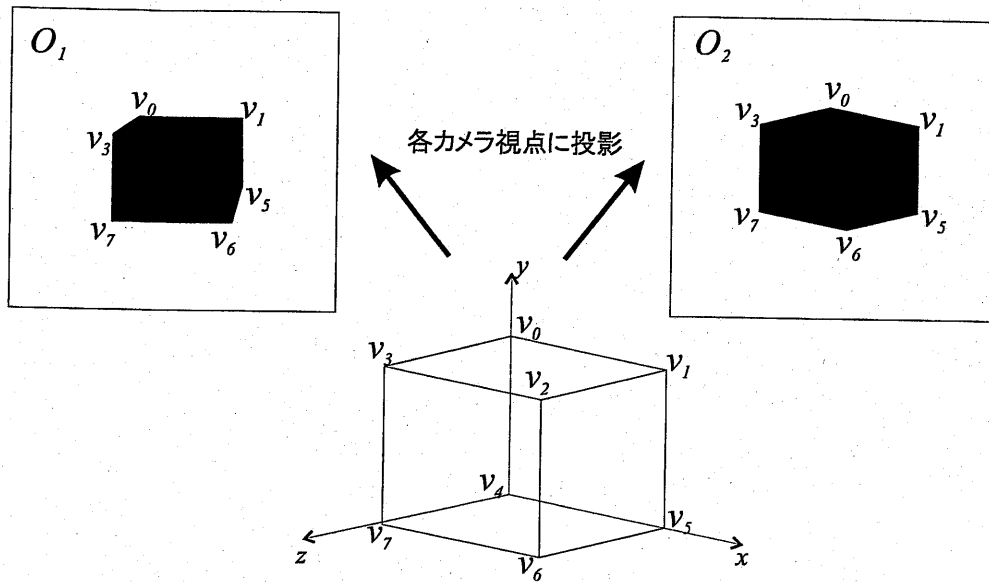


図 3.5 オクタントの投影

正面カメラ	側面カメラ	上面カメラ	属性
			WHITE
			GRAY
			BLACK

図 3.6 オクタントの属性

*Root-Octant*を定義

Level=0

Make_Octree

Level + 1

8個の*Sub-Octant* $C_0 \sim C_7$ を作成

For *Sub-Octant* = C_0 to C_7 **do**

Cross_Check (交差判定)

if *Sub-Octant*が完全内部

then 属性=*BLACK*

if *Sub-Octant*が完全外部

then 属性=*WHITE*

if *Sub-Octant*が交差

then 属性=*GRAY* *Make_Octree*を再帰的に呼び出す

Cross_Check (交差判定)

for すべてのカメラ視点の変換行列 $P = P_1$ to P_n **do**

for *Sub-Octant*の各頂点 $V = V_0$ to V_7 **do**

$v = VP$

8つの v から*Octant*のシルエット画像*O*を作成

if *O*はオブジェクトシルエットの外

then *Sub-Octant*は完全に外と決定

if *O*はすべてのオブジェクトシルエットの中

then *Sub-Octant*は完全に中と決定

else *Sub-Octant*は交差していると決定

図 3.7 オクトリーによる形状再構成のフロー

.
.	.	1	1	1
.	.	1	1	1	1	1	1	.	.
.	1	1	1	1	1	1	1	.	.
.	.	.	1	1	1	1	1	.	.
.	1	1	1	1	1	1	1	.	.
.	1	1	1	1	1	1	1	1	.
.	.	1	1	1	1
.	.	.	.	1	1
.

(A)入力2値画像

0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0
0	0	2	2	1	1	1	1	0	0
0	1	3	2	2	2	2	1	0	0
0	0	0	3	3	3	2	1	0	0
0	1	1	4	4	3	2	1	0	0
0	2	2	5	4	3	2	1	1	0
0	0	3	3	2	1	0	0	0	0
0	0	0	0	2	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0

(B)入力に対する
half distance transform map

図 3.8 片側距離変換マップ

画像の交差の有無をすべて調べることになり、非常に計算量が大きくなる。そこで、六角形の代わりにその六角形を内包する最小の正方形を用いて交差判定を行う。この正方形の左下コーナーの座標を (X_{sq}, Y_{sq}) とし、その一辺の長さを d_{sq} としておく。この交差判定では時に「完全に内部にある」六角形、あるいは「完全に外部にある」六角形に対しても「交差がある」と判定してしまうことがある。しかし、このような「交差がある」と判定したオクタントをすべて“GRAY”の属性にし、さらに次の階層でのチェックを行う。オクタントのサイズが小さくなればなるほど、その投影形状は正方形に近づいていくため、オクトリー全体では、このような正方形を用いた交差判定方法でも最終的には正しい結果を得ることが出来る。

矩形とシルエット画像の交差判定をさらに効果的に行うために、Szeliskiはシルエット画像とその反転画像についての2つの片側距離変換マップを使って行う方法を提案している。この片側距離変換マップはチェスボード距離(8近傍距離)変換画像[42]の一方方向だけを計算した距離変換画像である。図3.8に示す片側距離変換マップの各ピクセルはそのピクセルに左下コーナーを持つ正方形を配置したときに、完全に内包出来る正方形の一辺の長さを示している。この片側距離変換マップとオクタントの投影を内包する前述の矩形の左下のコーナー位置 (X_{sq}, Y_{sq}) とサイズ d_{sq} から、その正方形がシルエット画像に完全に含まれるか、または完

表 3.4 交差判定の高速化

復元レベル	method-1	method-2	method-3	method4
6	112	101	54	36
7	473	366	207	185

(単位 msec)

method-1 : 高速化を行わない交差判定

method-2 : 片側距離変換マップのみ使用

method-3 : オクタント投影テーブルのみ使用

method-4 : method 2,3 を両方とも適用した交差判定

全に外かが判定できる。そのどちらでもない場合は“GRAY”と判定する。図 3.8 において (X_{sq}, Y_{sq}) の位置が○のついているピクセルであるとする、 d_{sq} が 5 以下ならば、その正方形は完全に内包であり、5 より大きければ交差している可能性がある。図 3.8 は入力画像についての片側距離変換マップであるが、入力 2 値画像の反転画像について同様に片側距離変換マップを作成し判定を行うことによって、完全に外にあるか、交差の可能性があるかを判定できる。

root-octant に含まれるすべてのオクタントの投影は、カメラ位置に変更が無い場合は常に同じである。そのため、あらかじめすべてのオクタントの投影形状を計算しすべてのオクタントについて (X_{sq}, Y_{sq}) と d_{sq} を求めておきテーブル化しておくことによって、交差判定に必要な計算量を非常に少なくすることが出来る。

6 角形とシルエット画像の交差判定を直接行った場合、片側距離変換マップを用いた場合、更にテーブルを用いた場合のボクセルモデル作成にかかる時間の比較を行った。表 3.4 に結果を示す。結果に示すように、非常に効果的な手法であることが確認できた。

3.3.2 手のボクセルモデル

図 3.9 に示す手形状生成シミュレータでは視点位置を様々な場所に設定し、その視点から見た手形状を表示することが出来る。

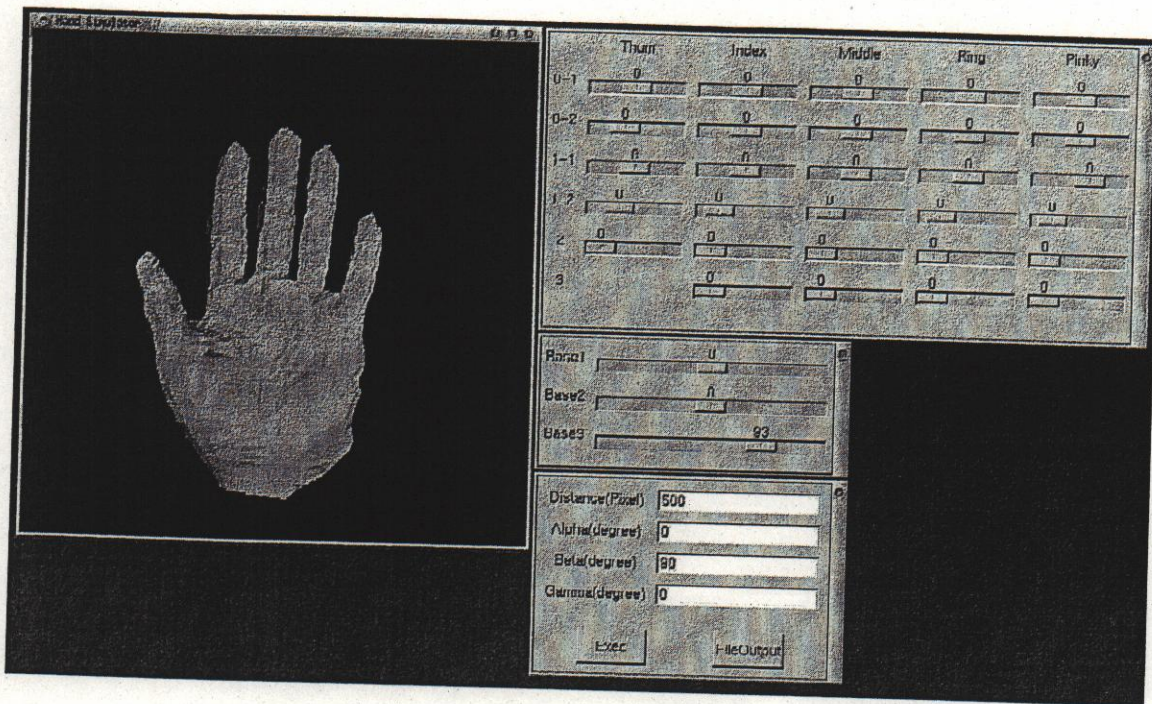


図 3.9 手形状シミュレータ

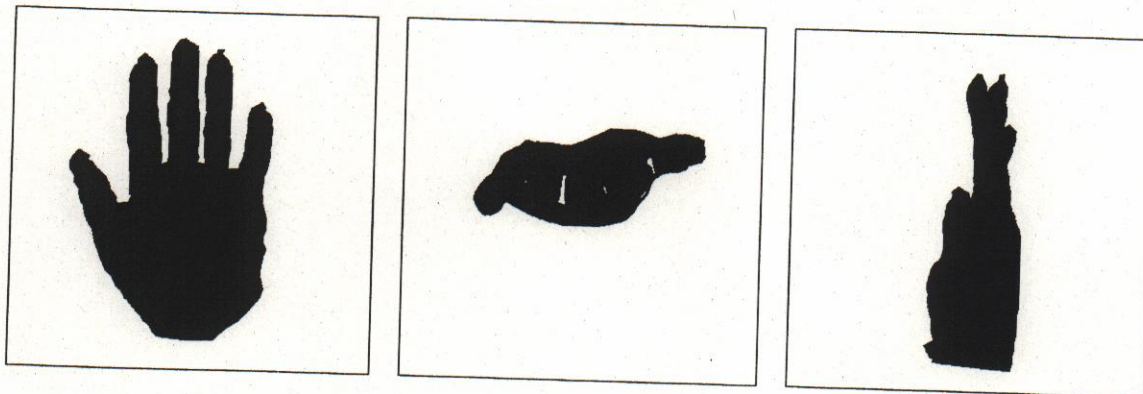


図 3.10 シルエット画像

シミュレータで、基本手形状から各関節角度を数度ずつ変化させて、新しい手形状を作成する。視点を正面、側面、上面の3点に設定し、それぞれの視点からの手形状画像を図 3.10 のようにシルエット化し、3.3.1 節で提案した手法でオクツトリーを用いて再構成する。図 3.10 の3枚のシルエット画像を用いて作成したボクセルモデルを、図 3.11 に示す。

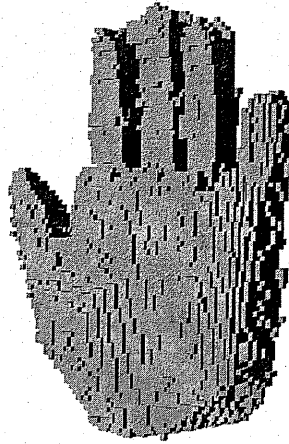


図 3.11 ボクセルモデル

3.4. 手の形状推定

3.4.1 提案手法の原理

3.3節の方法で得られたボクセルモデルと3.2節で述べた骨格モデルの表面形状データのフィッティングによって、骨格データの各関節角度を推定する手法について述べる。

観測データとして再構成したボクセルモデルは、空間を立方体に分割したボクセル空間中において手が占有する領域を提示している。また、表面形状モデルもまた手の存在する位置をポリゴンパッチの頂点座標によって提示している。表面形状モデルが、完全にボクセルモデルに内包されれば手のモデルは観測データにフィットできているものと考えられる。各パッチを構成する頂点座標が骨格モデルのどのリンクに対応しているかは、あらかじめ与えられている。今、ある関節位置の関節角度を $\mathbf{a}_i = \{a_i(k) | 0 \leq k < n_i\}$ ($a_i(k)$ は関節位置 i の自由度 k の角度) と表現したとき、手の姿勢は $P = \{\mathbf{a}_i | 0 < i < r\}$ で表される。この姿勢における表面形状モデルを構成するパッチの頂点群の座標は $L = \{\mathbf{p}(m) | 0 \leq m < q\}$ ($\mathbf{p}(m)$ は頂点座標位置, q は頂点の個数) である。各 $\mathbf{p}(m)$ は P によって一意に決まる。 $L \subset V$ になるように \mathbf{a}_i を決定していくことによって手形状を推定する。 $Out = \{\mathbf{p}(m) \notin V | 0 \leq m < q\}$ を評価関数とし、 $Out = 0$ となるような \mathbf{a}_i を決

定する。

そのために、*Out* なる点群に対してボクセルモデルに近づくようなベクトルを発生させ、その方向によって関節角度を微小に変化させながら評価を繰り返していく。

推定形状が3次元であることから、このように観測形状として3次元形状を用いることによって、微小変化に対する評価関数の計算が簡単になり、従来モデルマッチングの手法としてよく行われている、モデルの形状変化を、2次元画像に反映させ評価を行うことに比べてより直接的なモデルの形状変化を行うことが出来ると考えられる。3次元情報を2次元に落とし評価関数を計算することに比べ、2次元情報を補間し3次元形状を構築すると情報の正確さは同じレベルであるが、補間して得られた情報による収束スピードの向上とアルゴリズムの単純化が期待できる。

3.4.2 アルゴリズムの詳細

図 3.12 に処理フローを示す。処理の流れは以下のとおりである。

Step1: 視点角度の違う複数カメラからの得られた多視点画像をシルエット化する。

Step2: シルエット画像を用いて手のボクセルモデルを作成する。

Step3: 作成した現在の手の形状を表現するボクセルモデルと、初期状態の手の形状を表現している表面形状モデルを比較する。表面形状モデルを構成する3角形パッチの頂点群のうちボクセルモデルの外にある座標に着目する。

Step4: 着目した座標に対して図 3.13 のように関節軸方向に力 f を発生させる。次に f を各関節回りのトルク t に変換する。 t を足し合わせて関節を回転させるためのトルク T を決定する。

Step5: トルク方向に関節を $\Delta\alpha$ 度変化させる。 $\Delta\alpha$ は収束状態に応じて、 $\Delta\alpha = 4[deg] \rightarrow 2[deg] \rightarrow 1[deg]$ と最初は大きく変化させ徐々に変化させる角度を小さくする。

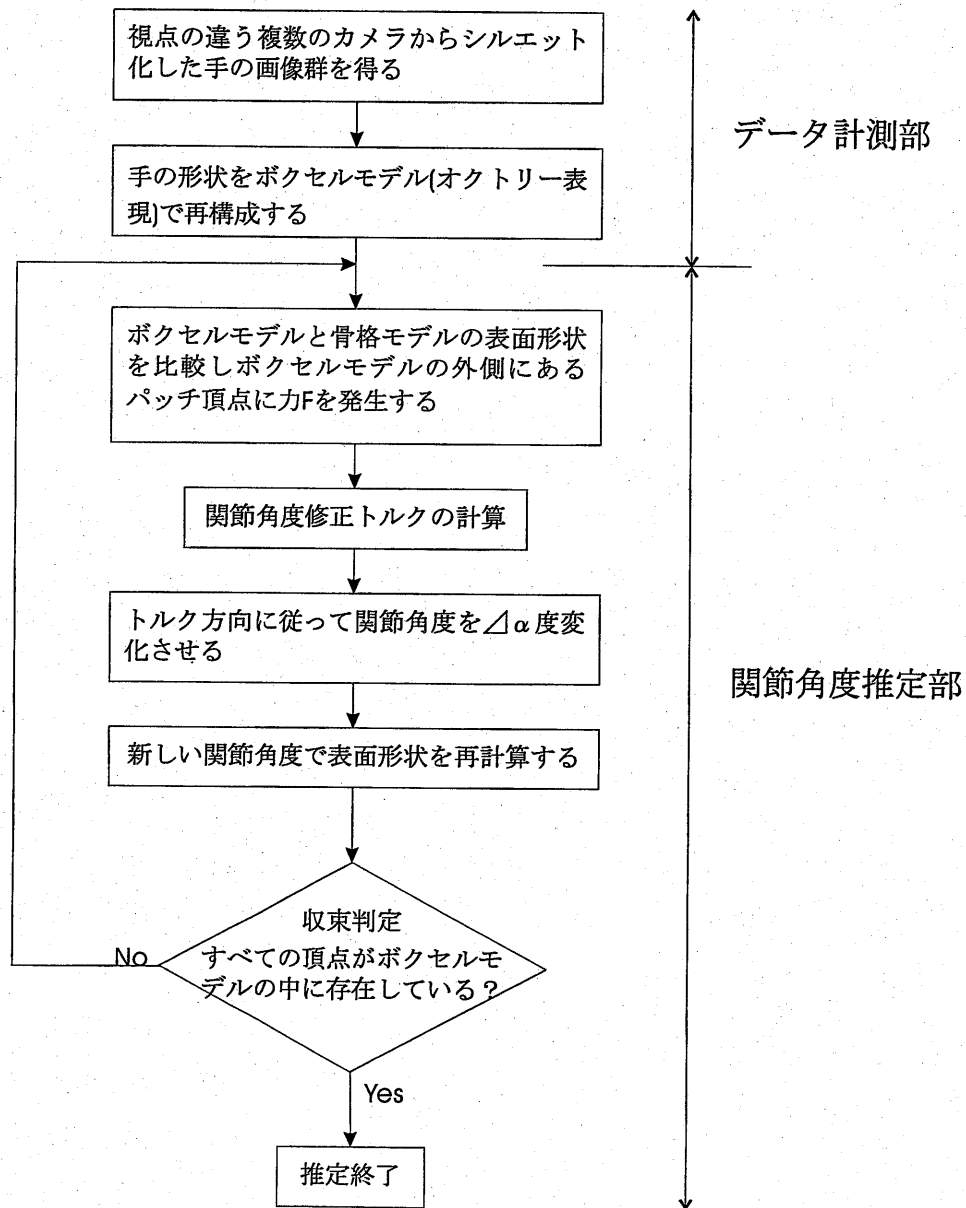


図 3.12 関節角度推定処理フロー

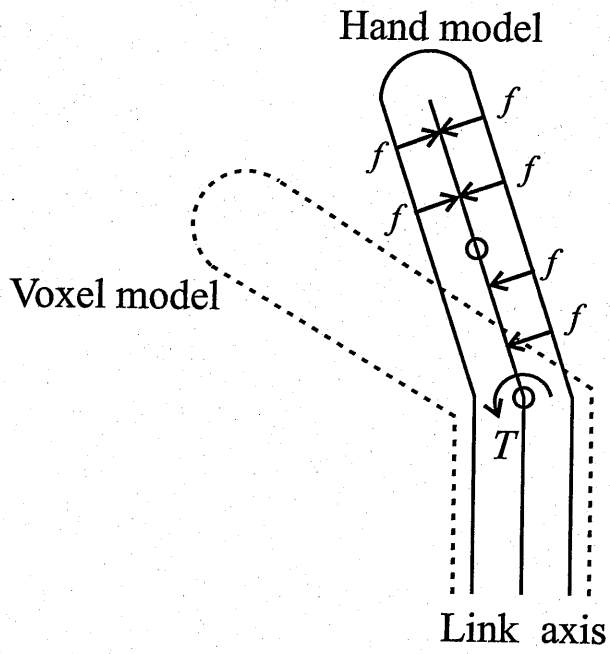


図 3.13 モデルフィッティングの仕組み

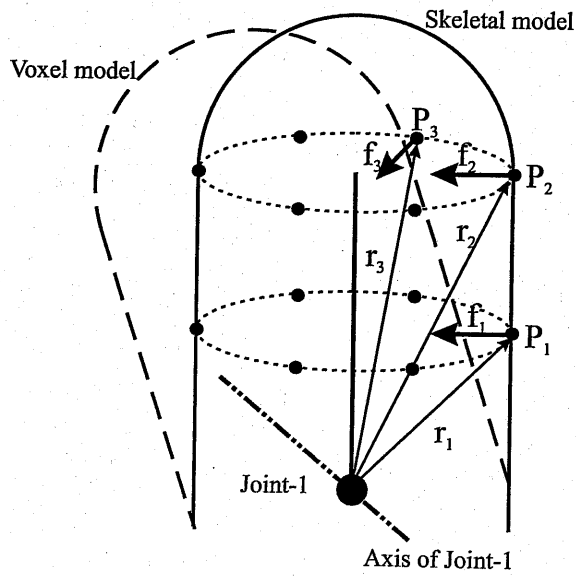


図 3.14 トルクの発生

Step6: 新しい関節角度より関節位置を計算しなおし、関節モデルの形状を更新する。

Step7: 評価関数を計算し、あるしきい値を下回れば推定終了とする。それ以外の場合は Step3 に戻る。

Step4 で述べた各ジョイントの回転方向の決定方法を図 3.14 の場合を例にとって説明する。今、頂点 P_1, P_2 と P_3 はボクセルモデルの外側に位置している。これらの頂点が Joint-1 に関連付けられていることは、あらかじめデータとして与えられている。これらの頂点について、リンク (骨に対応) に対して垂直方向の力 (f_1, f_2, f_3) をそれぞれ発生させる。また、Joint-1 からこれらの頂点に向かう位置ベクトルを r_1, r_2, r_3 とすると、各頂点でのトルクは $t_i = r_i \times f_i$ で与えられる。これらのトルク t_i の総和 $T = \sum_i t_i$ を Joint-1 に関するトルクとする。Joint-1 の回転軸に対する回転方向は、このトルク T の方向を用いて決定する。図 3.14 の場合、トルク T は Link-1 をボクセルモデルの方向に倒すような向きを持つことになり、結果として Link-1 は回転軸に対して反時計方向に $\Delta\alpha$ 度回転する。

Step 7 で述べた評価関数はすべての関節について計算される。評価関数は以下のように定義している、

$$\text{convergence ratio} = \frac{\text{in_vertex}}{\text{all_vertex}} \times 100[\%] \quad (3.1)$$

ここで *convergence ratio* は収束率、*in_vertex* はボクセルモデルの内側に位置する頂点の数、*all_vertex* は着目している関節とリンクに関係しているすべての頂点の数である。着目している関節とリンクに関係しているすべての頂点が、ボクセルモデルの内側に位置しているとき着目している関節の回転角度の推定が出来たと判断する。しかし、各関節の回転角度の範囲を超えたときと、推定を繰り返すあいだにトルクの向きに振動が発生したときは、その関節角度の推定を誤推定のまま終了する。

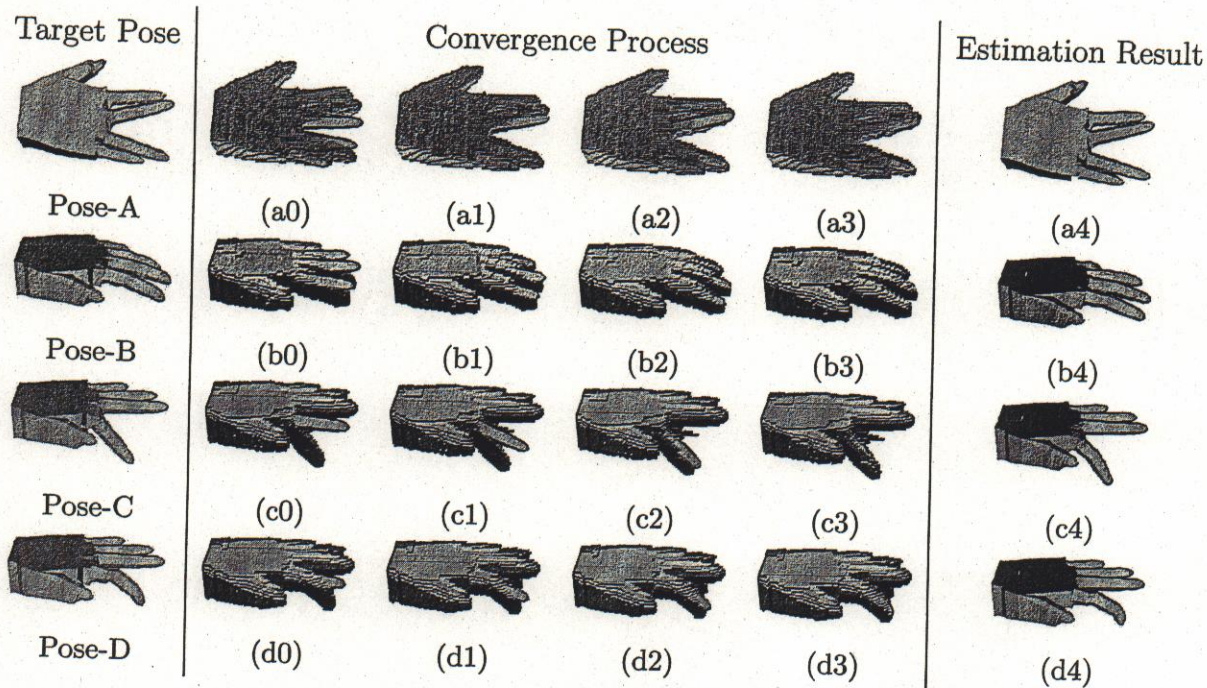


図 3.15 各種手形状における推定過程

3.4.3 推定結果

手形状シミュレータを用いて、4種類の手形状を生成し、その形状を推定した。シミュレーションシステムのCPUは dual Pentium III 1GHzであり、メモリサイズは 1024MBである。またボクセルモデルに用いるオクトリーの階層数は 8、視点は 4つである。4種類の手形状は以下に示すとおりである

Pose-A 示指の MP 関節を 8[deg] 内転，中指の MP 関節を 15[deg] 外転，環指の MP 関節を 10[deg] 内転，小指の MP 関節を 5[deg] 外転

Pose-B 親指以外の 4 指の MP,PIP,DIP 関節をそれぞれ 10[deg] 前屈

Pose-C 示指の MP 関節を 50[deg] 前屈

Pose-D 示指の MP 関節を 10[deg] 外転，示指の PIP と DIP の関節をそれぞれ 30[deg] 前屈

提案している推定手法の目的は連続的で任意の手形状の推定である。そのため、各フレーム間の手形状の変化は小さく、関節角度に関して考えれば約15[deg]以下であると仮定している。このような仮定を前提として考えると、Pose-A, Pose-Bは前提条件の下での手形状である。Pose-C, Pose-Dは仮定に比べて変化の度合いが大きい、このような場合でも推定がロバストに行われるということを示す。

手形状推定のシミュレーション結果を図3.15に示す。図3.15において、(a0), (b0), (c0), (d0)はボクセルモデルに重ねた初期の手形状モデルを示す、そして(a1), (b1), (c1), (d1)～(a3), (b3), (c3), (d3)は骨格モデルがボクセルモデルに収束していく様子を示している。(a4), (b4), (c4), (d4)は手モデルの最終推定形状を示しており、そのときは骨格モデルが完全にボクセルモデルに含まれている。(a4), (b4), (c4), (d4)と生成した手形状(Pose-A～Pose-D)を比べると、どの手形状でも正しく推定できることは明らかである。

3.5. 実画像を用いた手の形状推定

3.5.1 実験システム

実際の手の画像に対して提案手法を適用するために実カメラシステムを構築した。3次元形状入力等のインタフェースをターゲットとしているため、手が動く領域はあまり広くとる必要はない。そこで図3.16のように60cm四方のフレームを構成し、このほぼ中央で動く手の形状を推定することにした。このフレームにCCDカメラを4台取り付けて手の画像を撮影する。カメラ位置は正面、側面、上面、45度上面に設定する。但し、カメラ数は、必要な処理速度・精度によって増減が可能である。またカメラ位置に関しても、ボクセルモデルを安定に作成できる位置であれば、必ずしもこのような位置関係を必要とするものではない。

現在は、手のシルエットを簡単かつ安定に抽出することを目的としてフレーム枠にブルーのカラーボードを設置している。

処理システムの主な構成は以下のとおりである。尚、本研究で用いたハードウェア構成ではスロット数の制限により、カメラの台数は最大4台までしか用いることが出来ない仕様になっている。

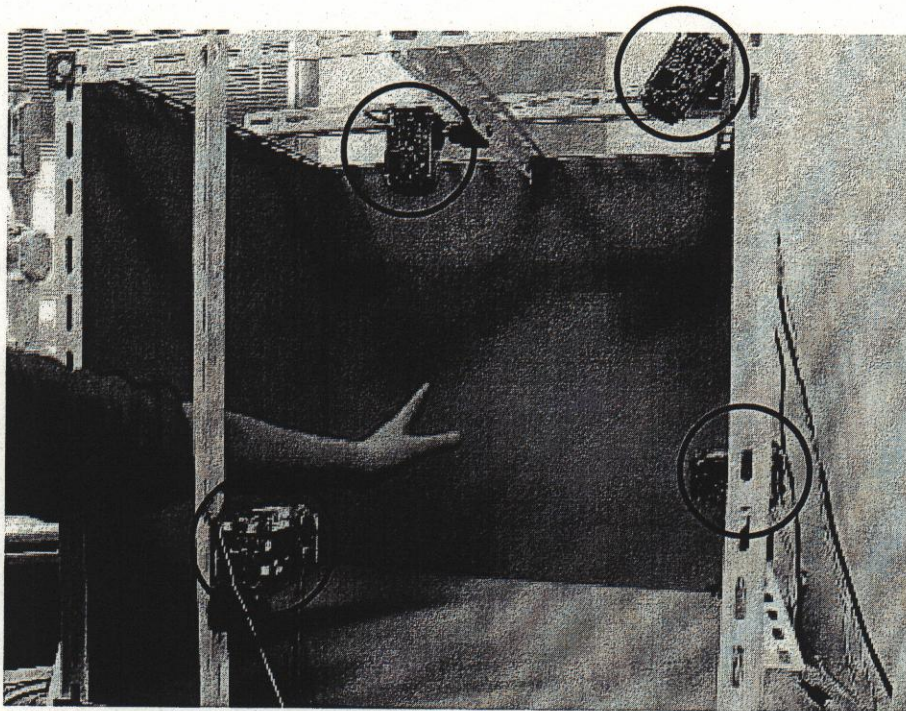


図 3.16 実験環境

CPU	:	PentiumIII 1GHz × 2
Memory	:	1024Mbyte
Graphic Board	:	GeForce4
Capture Board	:	GV-VCP2/PCI(Bt878) × 4
Camera	:	SONY CCB-EX37 × 4

3.5.2 カメラキャリブレーション

手のオクトリー表現によるモデルを作るために、任意視点での3次元空間と2次元画像の対応する点を一つの中心射影行列によって関係付けるキャリブレーションが必要となる。ピンホールカメラモデルにおけるキャリブレーション法には大きく分けて2種類ある。一つは外部変数と内部変数を独立に求める方法である。もう一つは世界座標系から画像座標系までの光学的・幾何学的な関係を一つの中

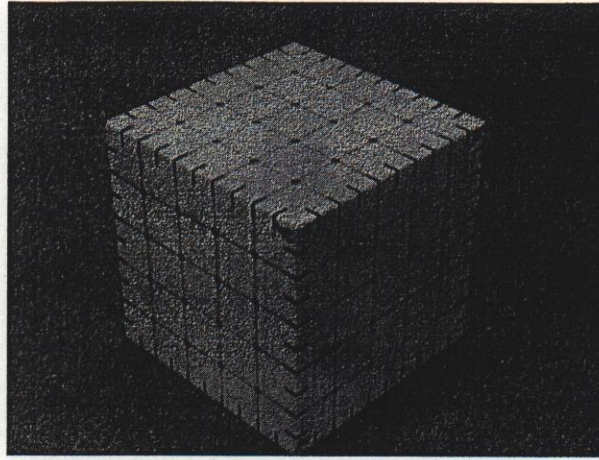


図 3.17 キャリブレーションボックス

心射影行列で表す方法である。本研究ではオクタントの各頂点座標が2次元画面（撮像面）のどの位置に投影されるかを一つの中心射影行列で関係付ける。この中心射影行列は6つの3次元空間上の点と2次元画像上の点との対応付けから求めることが出来る。ここで中心射影行列を P とし、3次元上の点 $M = \{X, T, Z\}^T$ ，画像上の点を $m = \{u, v, 1\}^T$ とすると以下の関係が成り立つ。

$$sm = PM \quad (3.2)$$

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix}$$

である。

図 3.17 に示すキャリブレーションボックスをカメラで撮影し、キャリブレーションボックス上の交点座標がカメラ画像のどの位置に映っているかを指示する。このためのツールとして、図 3.18 に示すようなキャリブレーションツールを作成した。このツールで同一平面上にない6組以上の3次元座標位置と2次元座標の組を指定し、中心射影行列を求める。

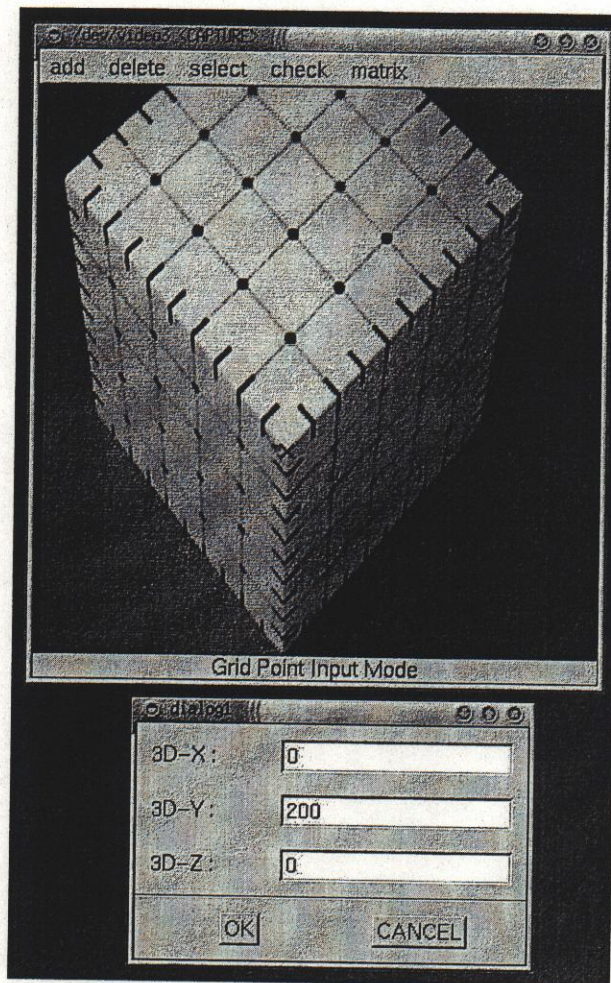


図 3.18 キャリブレーションツール

具体的には、6組以上の基準点から最小2乗法によってパラメータを同定する。
 n 点の基準点の物体座標 (X_i, Y_i, Z_i) とそれに対応するカメラ座標 (X_{ci}, Y_{ci}) が既

知であれば $P_{34} = 1$ と置くことより,

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -X_1 X_{c1} & -Y_1 X_{c1} & -Z_1 X_{c1} \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -X_1 Y_{c1} & -Y_1 Y_{c1} & -Z_1 Y_{c1} \\ & & & & & & \dots & & & & \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -X_n X_{cn} & -Y_n X_{cn} & -Z_n X_{cn} \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -X_n Y_{cn} & -Y_n Y_{cn} & -Z_n Y_{cn} \end{bmatrix} \begin{bmatrix} C_{11} \\ C_{12} \\ \dots \\ C_{32} \\ C_{33} \end{bmatrix} = \begin{bmatrix} X_{c1} \\ Y_{c1} \\ \dots \\ X_{cn} \\ Y_{cn} \end{bmatrix} \quad (3.3)$$

となる. これを,

$$A \cdot C = R \quad (3.4)$$

と表せば, 最小2乗法により

$$P = (A^T A)^{-1} A^T R \quad (3.5)$$

で中心射影行列 P を得ることが出来る.

3.5.3 初期姿勢の推定

手形状シミュレータを用いた手形状推定では手の位置は既知であり, 手形状のベース位置は動かない. また, シミュレータ上に表示された画像より入力シルエット画像群を作成するため, それらより再構成されたボクセルモデルと骨格モデルの表面形状位置は基本的に一致する. そのため手の初期位置・姿勢を推定する必要はなかった. しかし, 実験環境空間内に差し出した手の初期位置は未知である. 推定を行うためには, 初期状態における骨格モデルの表面形状が完全にボクセルモデルに内包されていなければならない. そのために, 本研究では, 手の初期姿勢・形状を以下に示すようなに示すような形とし, それらの知見から初期姿勢・形状をアドホックに推定する. 示した形は図 3.19 のような姿勢・形状である.

形状: 各指を中指方向に, 互いの指側面が沿うように揃える.

母指以外の各指はまっすぐ伸ばす.

母指は示指の橈側面に沿うように関節を曲げる.

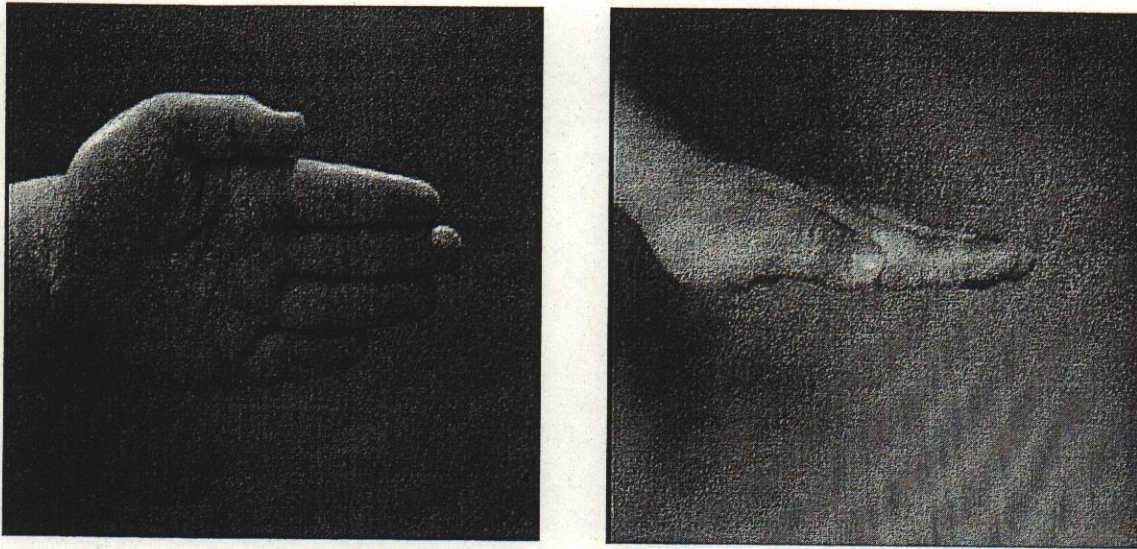


図 3.19 初期手形状

姿勢：中指と実験環境空間の底面がほぼ平行になる。

中指と正面カメラ方向がほぼ直交する。

一般に図 3.19 のような姿勢・形状では、中指の先端位置の X 座標値は指形状の中で最大値を持つ。さらに、その形状でのボクセルモデルは図 3.20 のように分岐を持たずひとかたまりの形状になり、重心位置も幾何的に求めやすい位置となる。これらの状態は個人間の格差も少ない。

また、表面形状モデルの初期状態はシミュレータを用いて図 3.21 のように作成することが出来る。この初期状態から、図 3.20 のボクセルモデルの中に完全に包含されるような、姿勢を推定すればよい。

姿勢推定は以下のように行う。

Step1: 実画像を入力としたボクセルモデルには手首部分も再構成されている。そこで図 3.23 のように、X 座標の最大値から手掌部の長さ分まで、また Z 座標の最大値から手掌の厚み分までを、手部分のボクセルモデルとする。ボクセルモデルの手部分の重心位置を計算する。

Step2: 図 3.22 の左側に示すように、ボクセルモデルの最大の X 座標値を持つオクタントの位置 V_{top} を求める。この位置から重心位置までのベクトルを

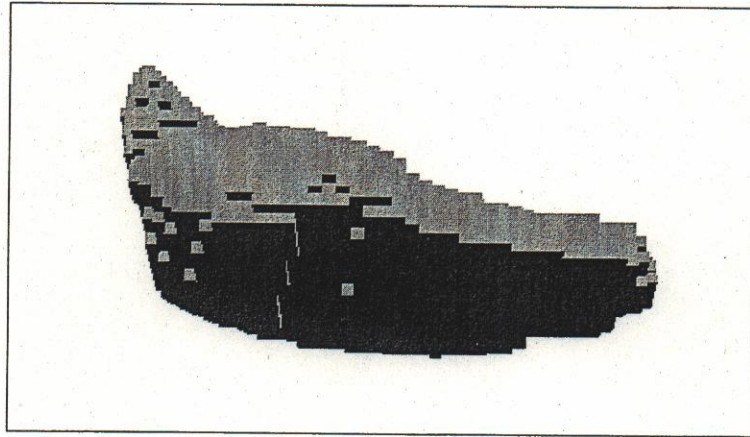


図 3.20 実画像より作成したボクセルモデル

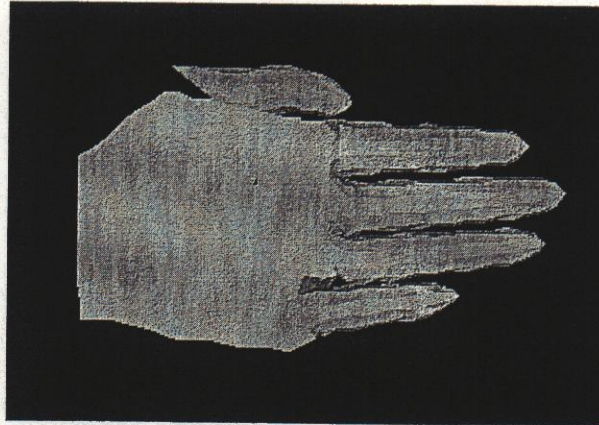


図 3.21 表面形状モデルの初期状態

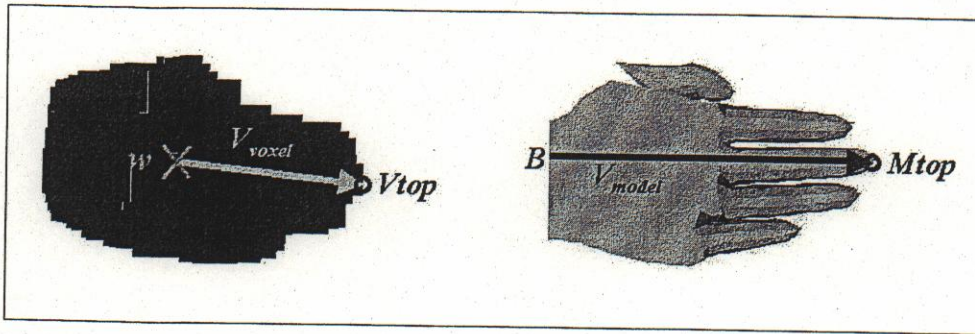


図 3.22 初期姿勢の推定

V_{voxel} とする.

Step3: 図 3.22 の右側に示すように，表面形状モデルの中指の先端位置 M_{top} から，モデルのベース位置へ向かうベクトルを V_{model} とする.

Step4: V_{top} と M_{top} が重なるように骨格モデルを並進させる.

Step5: V_{model} を V_{voxel} に重ねるように骨格モデルを回転させ，初期姿勢を決定する.

3.5.4 連続画像入力による手形状推定

あるフレームの入力画像に対して手形状推定をおこない，その推定結果を次フレームの初期形状として用いることによって手形状推定を連続して実行することが出来る．1フレームの手形状を短い時間間隔で推定することが出来れば，リアルタイムでの手形状推定が可能である．フレーム間で姿勢の回転がないという前提で連続手形状推定を行った．

図 3.24，図 3.25 は実画像を入力とした連続手形状推定の推定結果である．図 3.24 は，示指のみを前屈した手形状の推定結果を，図 3.25 は，環指と小指を順に外転させた手形状の推定結果をそれぞれ示している．現状の処理システムでの，処理時間を表 3.5 に示している．前処理には，2 値化，ノイズ処理，そして片側距離

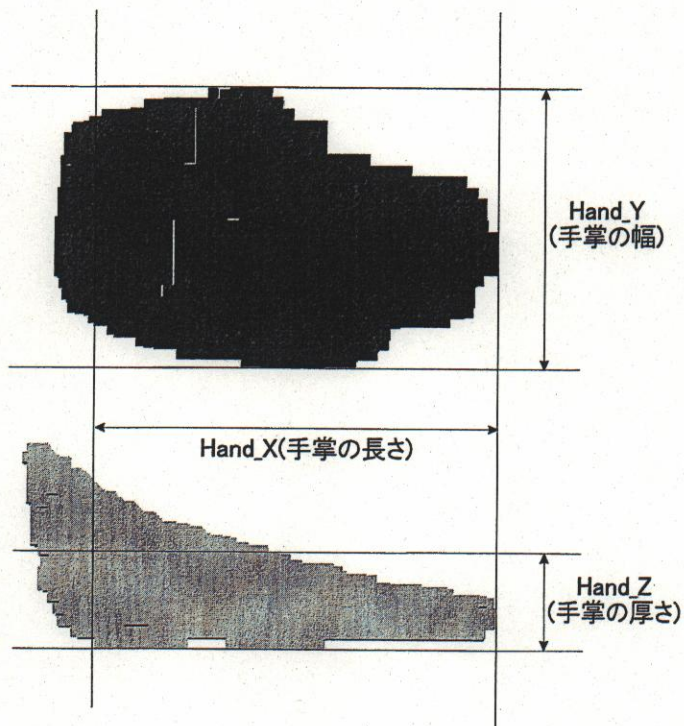


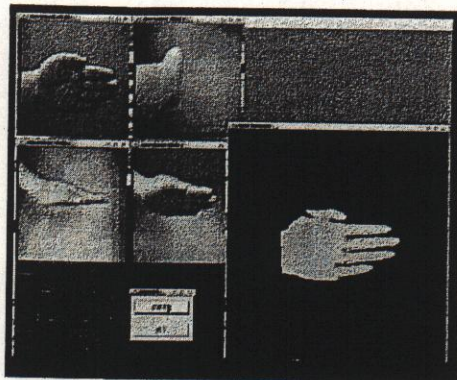
図 3.23 手領域の定義

表 3.5 平均推定時間

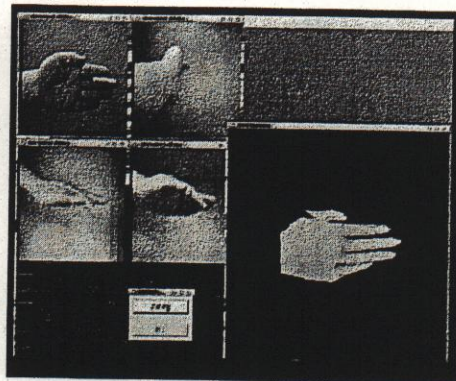
画像キャプチャ+前処理	130 msec
ボクセルモデルの生成	50 msec
手形状推定	160 msec
合計	340 msec

変換マップ作成が含まれている。またカメラの数は4台、オクトリーの階層数は7である。

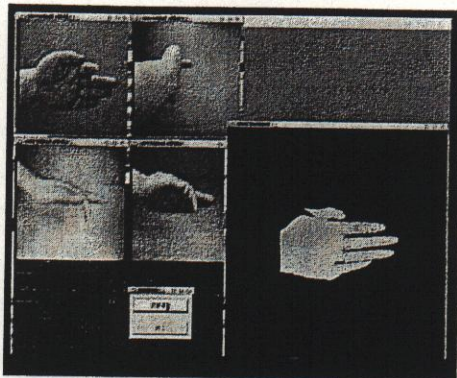
実画像での推定結果と処理速度の表から、提案した推定方法は実画像でもうまく動作するが、推定速度は未だ不十分であることがわかる。



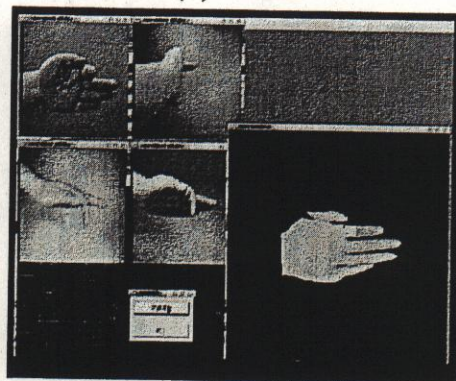
(1) 0.0sec



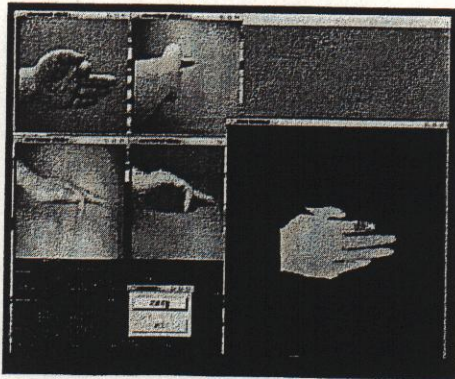
(2) 0.6sec



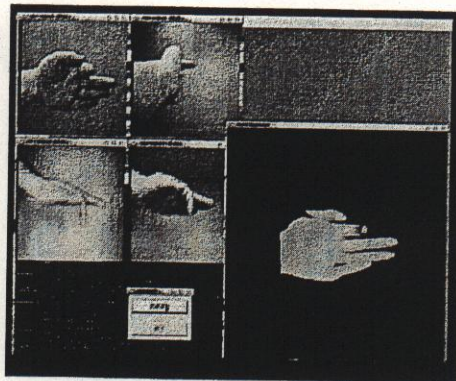
(3) 1.2sec



(4) 1.8sec



(5) 2.4sec



(6) 3.0sec

図 3.24 実画像入力による連続手形状推定の例 (1)

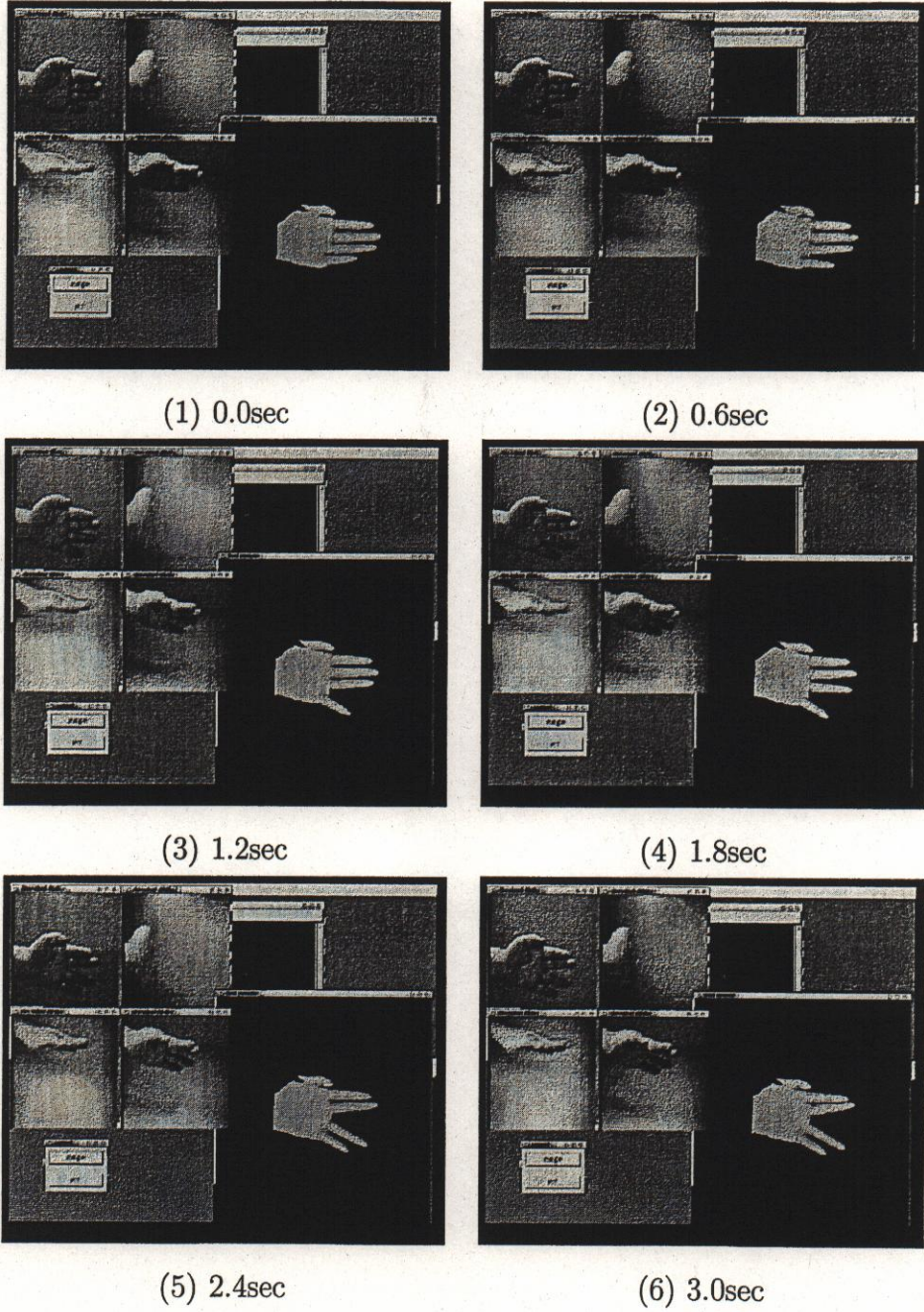


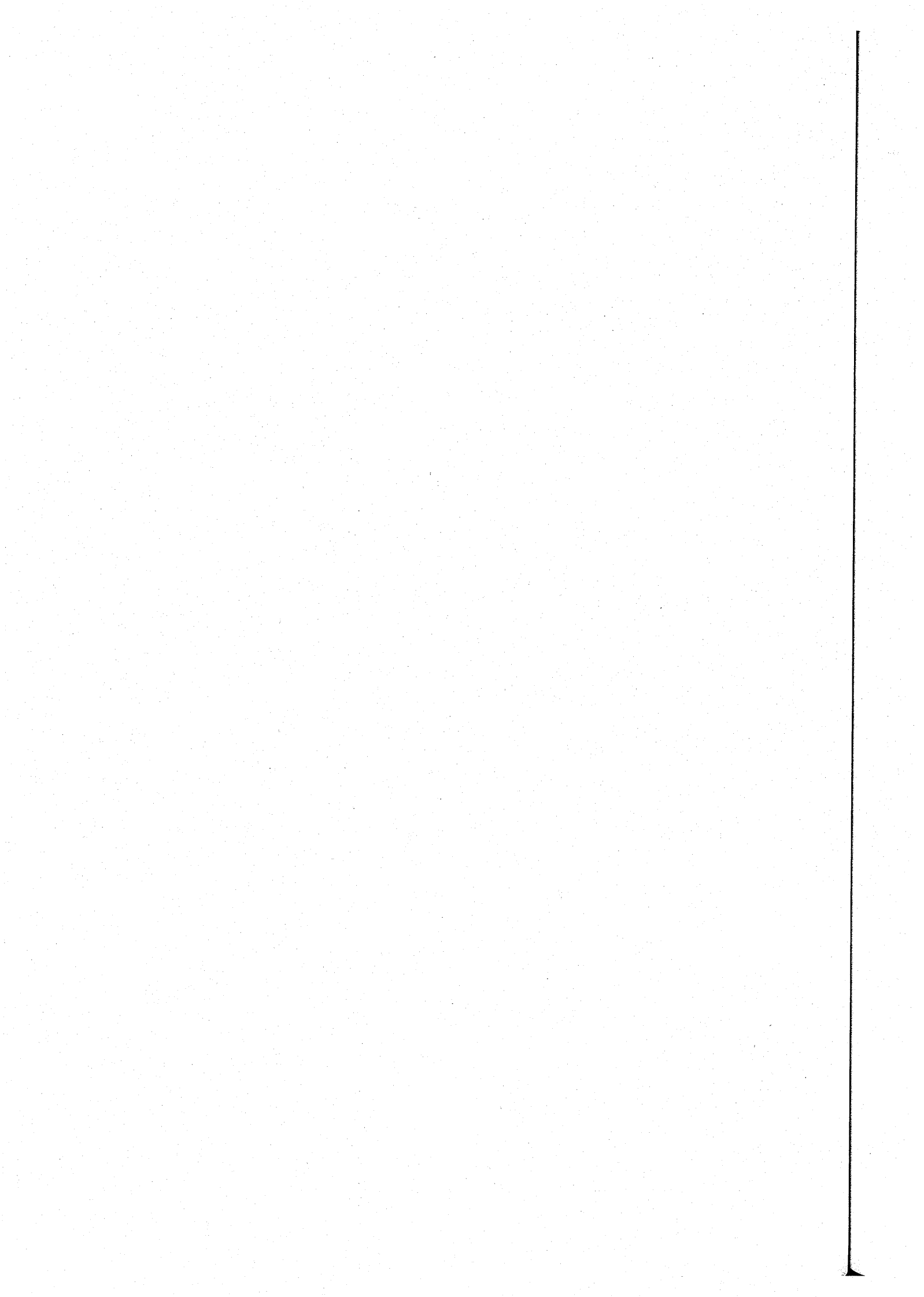
図 3.25 実画像入力による連続手形状推定の例 (2)

3.6. 本章の結論

本章では、あらかじめシステム内に構築した手のモデルと、多視点カメラシステムから得られたシルエット画像を統合して3次元化された手の観測データの間の3次元的なモデルフィッティングを基本とした、手形状推定アルゴリズムを提案した。その上で、実際の手の動作を多視点カメラでキャプチャした複数の動画像に対して実装し、実環境における提案アルゴリズムの検証を行った。

提案した手法は、従来モデルマッチング手法として一般的に行われるモデルの形状変化を2次元画像に反映させ評価を行う手法に比べて、3次元的に行うことによりより直接的にモデルの形状変化を取り扱うことが出来る手法である。この提案手法の有効性を示すため、シミュレーションを行った。手形状シミュレータにより変形度合いの違う4種類の手形状を生成し、推定経過をチェックした。シミュレーションの結果、前提としている関節角度15[deg]以下の小さな形状変化だけでなく、50[deg]近い形状変化でも推定できることが確認できた。

次に実画像を入力として、手動作を観測するために4台のCCDカメラを用いた多視点カメラシステムを構築した。ユーザーにとっては不自然な位置での物体操作になるが、安定してシルエット画像を抽出するために一辺60cmの立方体のフレームに、ブルーの背景を取り付けた観測空間を持つ、カメラ環境であった。次に、実カメラ環境で必要となる初期姿勢の推定について、手の形状の知識を利用してアドホックに行う方法について述べた。指をすべてそろえ、中指方向と手動作空間の底面が平行かつ、中指と正面カメラ方向が直交するようにした形状を初期形状とし、その形状での手の位置・姿勢を簡単に推定する方法を採用した。これらを行ったうえで、提案アルゴリズムによる多視点動画像を入力とする手形状推定を実装し、その動作を検証した。検証した結果、提案した推定手法では実画像を入力とした場合もシミュレーション結果と同様に期待通りの動作をすることを確認した。



第4章 粗密探索法を導入した多視点 シルエット画像を用いた手形 状推定

4.1. 本章の趣旨

本章では、第3章での実画像を用いた手形状推定結果より、精度の良い手形状推定には処理速度の向上が不可欠であることが明確になったことを受けて、処理速度の向上のためにおこなった粗密探索法の導入について述べる。手形状推定における粗密探索法は、最初は粗く推定を行い大まかに関節角度を推定し、その後詳細な推定を行い精度良く関節角度を推定するものであり、推定の多段階化である。

最初に、粗密探索法を導入するために必要な、2種類の解像度を持つ手の表面形状データの表現方法について述べる。次に、第3章で述べた手形状推定手法への粗密探索法の導入について述べる。最後に、シミュレーションと実画像入力による実験により、手形状推定手法改良の効果について検証を行う。

4.2. サブディビジョンサーフェスを用いた手の表面形状データ

粗密探索法を導入するためには、骨格モデルの表面形状データに粗推定時用の粗いデータ (coarse データ) と詳細推定時用の詳細なデータ (fine データ) の2種類の解像度を持つ必要がある。また、人間の手の皮膚用面は骨格姿勢の変化に応じて滑らかに変形する。そのため、骨格モデルの表面形状データも骨格姿勢の変

化に応じて滑らかに変形出来なければならない。これまでの手の表面形状データは皮膚表面を多数の小さな3角形のポリゴンで表現していた。また各ポリゴンの形状が骨格姿勢の変化に対応して変形できるように、ポリゴンの各頂点がどのリンクに対応してその位置を変更するかをあらかじめデータとして与えていた。この頂点データの数は、手全体を滑らかに表現するためには約2000点必要である。しかし頂点間の位置関係を考慮していないため、関節を大きく曲げるなどの姿勢をとったときには皮膚表面の滑らかな変形が保持できず、実際の手形状と大きく違う表面形状になってしまう欠点を持っていた。

これら二つの要求を満たすような手の表面形状を得るために、サブディビジョンサーフェス(細分割曲面)を導入した。以下にサブディビジョンサーフェスの詳細と、サブディビジョンサーフェスを用いた表面形状データの作成方法について述べる。

4.2.1 サブディビジョンサーフェスを用いた3次元形状表現

サブディビジョンサーフェスは、3次元オブジェクトのパラメトリックな表現の一つで、ポリゴンからなる3次元オブジェクトを、その各頂点にあるルールに従って移動させ、新しく稜線を生成することを何度も繰り返しながら、ポリゴンを分割して得られる滑らかな曲面のことである。他のパラメトリック表現と比べて、サブディビジョンサーフェスには二つの利点がある。一つは一枚の面で多面体形状の滑らかな表面を定義できるという点である。B-splineやNURBSなど従来のパラメトリック曲面を用いて多面体の表面形状を表現する場合、オブジェクト全体を一つの曲面で表すことが出来ないため、複数の自由曲面を貼りあわせて表現する必要がある。そのため、複数の曲面の滑らかな接続が必要となる。また複数の曲面が、既に滑らかに接続されている場合でも、一部の曲面形状が変形すると互い面の滑らかな接続性が失われるため、曲面の連続性を保つプロセスが必要となる。そのため、リアルタイムでの滑らかな多面体形状の変形は困難なものとなっている。もう一つの利点はサブディビジョンサーフェスは細分割の回数によって曲面の滑らかさやポリゴン面積を変えるLOD(Level of Detail)の特徴を持っていることである。細分割の回数で、表面形状を構成するポリゴン数やポリゴンの

大きさを調整することが出来る。同じ形状を現すのに、複数の解像度を与えることが出来るのである。

ポリゴンの分割方法も、Catmull-Clark サブディビジョンサーフェス、Loop サブディビジョンサーフェス、Doo-sabin サーフェスなど様々な種類が存在する。本研究ではそれらの中でも、細分割を無限大回数繰り返した極限の曲面が4階のB-spline に収束する性質をもつ Catmull-Clark サブディビジョンサーフェスを手の表面形状データの表現方法として採用している。Catmull-Clark サブディビジョンサーフェスを用いた3次元形状の多面体表現方法 [43] について以下の節で述べる。

4.2.2 Catmull-Clark サブディビジョンサーフェス

本研究での Catmull-Clark サブディビジョンサーフェスは、“face” と呼ばれる四角形のポリゴンで多面体を表現している。また多面体の稜線辺を“edge” と呼び、また多面体の各頂点を“vertex” と呼ぶ。多面体表現の初期状態において、オブジェクトの概形を粗い face による多面体で表現しておく。この初期状態の多面体を“初期多面体”と呼ぶ。また初期多面体のワイヤースケルト表現を初期格子と呼ぶ。

細分割のプロセスは大きく2つのフェーズに分けられる。一つは新しい vertex の生成プロセス、もう一つは多面体の再構成である。以下に、それぞれのプロセスの流れを示す。

[vertex の生成] 1回のサブディビジョンでは初期多面体から以下のような3種類の頂点が生成される。

- Face points(f): 各 face に関して、face point は各々の中心点とする (図 4.1(a)).
- Edge points(e): 各 edge に関して、edge point は式 (4.1) のように face と edge の重み付き平均で与えられる (図 4.1(b)).

$$e = \frac{a_{avg} + m}{2} \quad (4.1)$$

ここで、 a_{avg} は対象としている edge を含む face(2面存在する)それぞれの中心位置の平均、 m は対象としている edge の中点である。

- Vertex points(v): 各 vertex p に関して、vertex point は式 (4.2) のように face と edge と vertex の重み付き平均で与えられる (図 4.1(c)).

$$v = \frac{a_{avg} + 4m_{avg} + p}{6} \quad (4.2)$$

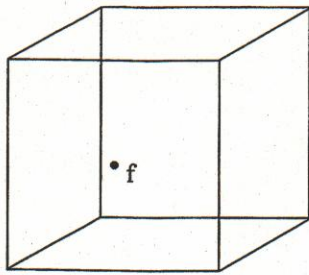
ここで、 a_{avg} は対象としている vertex p を含む face(3面ないしは4面存在する)それぞれの中心位置の平均、 m_{avg} は対象としている vertex p を含む edge(face と同数存在)それぞれの中心位置の平均、 p は対象としている vertex の位置である。

[多面体の再構成] 細分割された多面体は以下のように再構成される。face point とその面を構成する edge の edge point 間のそれぞれに対して新しい edge を生成する。edge point とその edge を構成する2つの vertex の vertex point 間それぞれに対して、さらに新しい edge を生成する (図 4.1(d))。一つの face は一度の細分割過程によって4つの face に分割される。この細分割過程は必要な滑らかさに達するまで再帰的に繰り返すことが出来る。図 4.2 に細分割の回数による多面体の滑らかさの違いを示す。

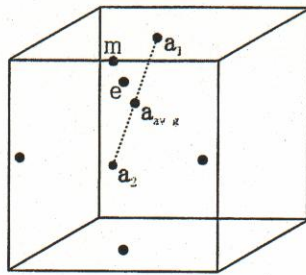
上記のように説明した細分割過程は、以下のようにベクトル行列の積で表現することが出来る。

$$d = Ap \quad (4.3)$$

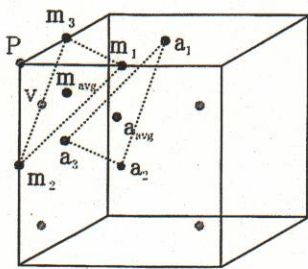
$p = [p_1, p_2, \dots, p_n]^T$ は初期多面体の vertex 位置の集合、 $d = [d_1, d_2, \dots, d_m]^T$ は細分割された多面体の vertex 位置の集合、常に $n < m$ である。 p は“コントロールポイント”， d は“データポイント”と呼ばれる。 A は $m \times n$ の重み行列であり、細分割のルールを表している。



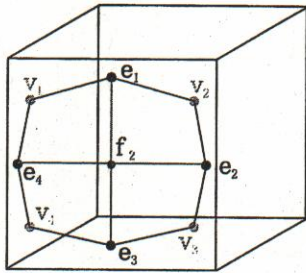
(a) face point の生成



(b) edge point の生成

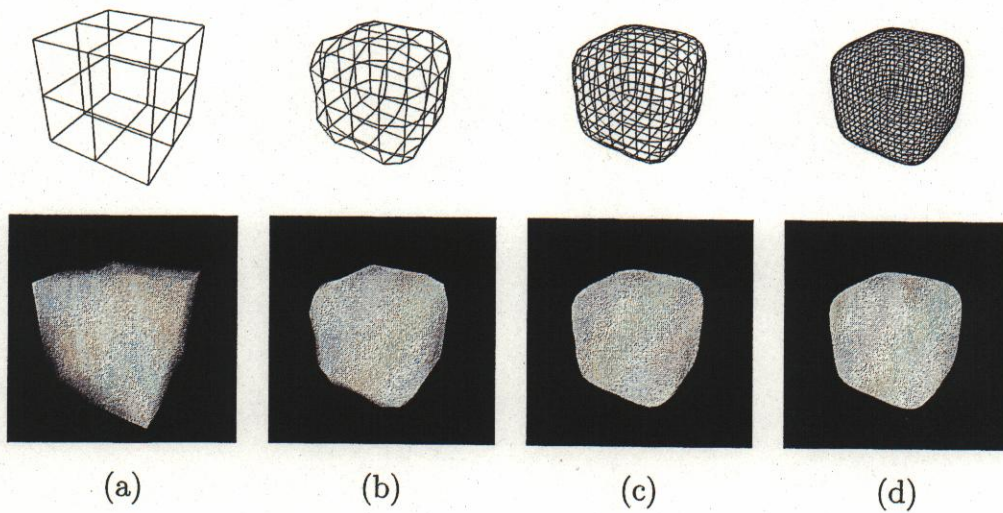


(c) vertex point の生成



(d) 新しい edge の生成による細分割面の再構成

図 4.1 サブディビジョンサーフェスの過程



(a)

(b)

(c)

(d)

図 4.2 サブディビジョンサーフェス: (a) 初期状態, (b) 1回細分割, (c) 2回細分割, (d) 3回細分割

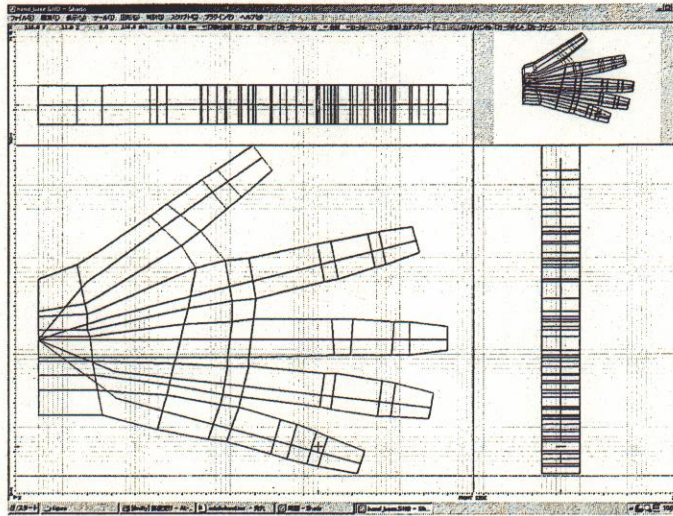


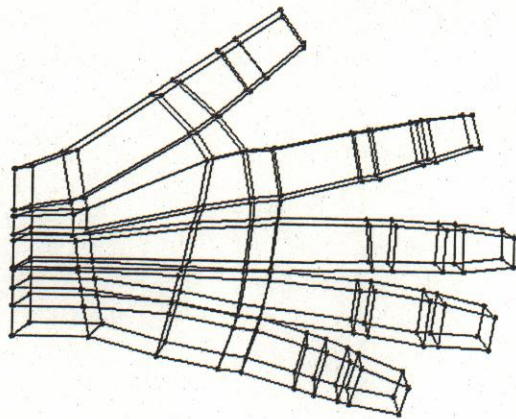
図 4.3 Shade 6 を用いて作成した手の初期格子

4.2.3 手の表面形状データの設計手順

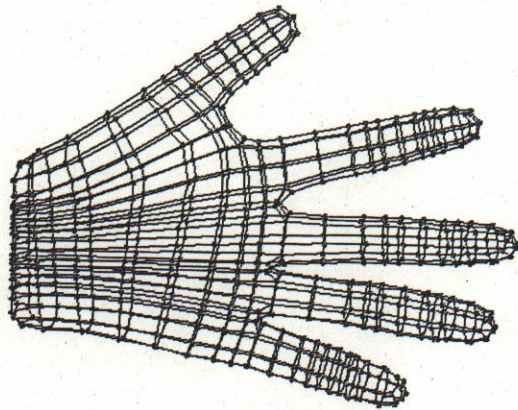
手の表面形状データを生成するためには、まず最初に初期格子を設定しなければならない。人間の手の表面形状をモデル化するためには、細分割後の形状を考慮して初期格子を生成する必要がある。また Catmull-Clark サブディビジョンサーフェスでは、各ポリゴンを四辺形で表現しておかなければならない。そのため、初期格子を 3次元のモデリングツールを用いずに設定することは非常に困難である。そこで、本研究では市販の 3次元モデリングシステムである、“LightWave 3D Ver.7.5” と “Shade 6” のいずれかを用いて初期格子を設定できるように、データコンバータを作成した。これによって、モデリングシステムの多様な機能を有効に使った初期格子の設定が可能になる。

以下にモデルの作成手順を示す。

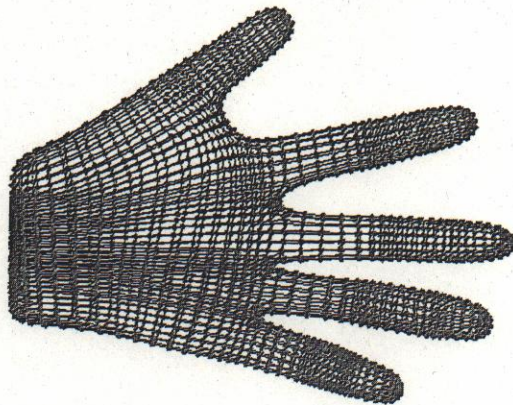
1. 図 4.3 のように 3D モデリングシステム (LightWave 3D もしくは Shade 6) を用いて細分割曲面を構成するための初期格子を設定する。
2. 図 4.4 のように細分割曲面生成プログラムによって手表面形状データを生成する。図 4.4 は上から、初期格子、1 回細分割、2 回細分割の結果をそれぞれ示している。



初期格子 (ポリゴン数=120)



1回細分割後の表面形状データ (ポリゴン数=480)



2回細分割後の表面形状データ (ポリゴン数=1920)

図 4.4 細分割された手形状データ

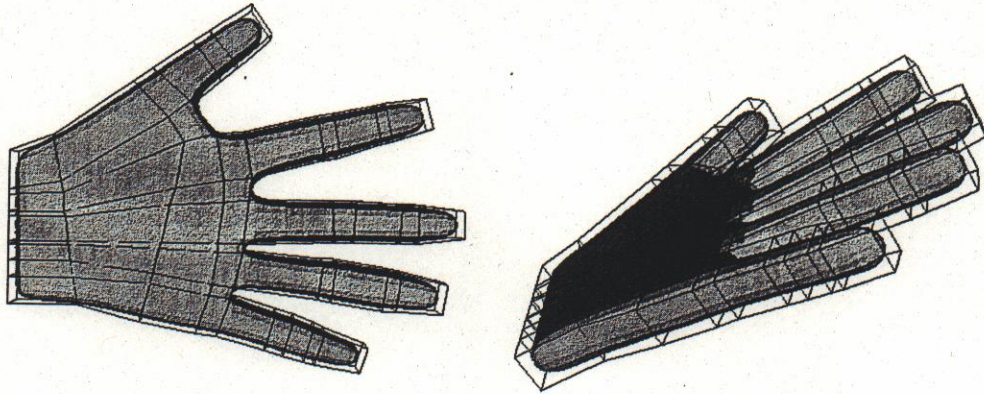


図 4.5 新しい手の表面形状データ

3. 手のリンクと初期格子点の接続関係を設定ツールを用いて設定する.
4. 手形状シミュレータを用い, 関節角の変化に対して表面形状が滑らかに変形するのを確認する.

このようにして作成された手の表面形状データの例を図 4.5 に示す. ラティスの中に, 手の表面形状データが納まっていることが分かる.

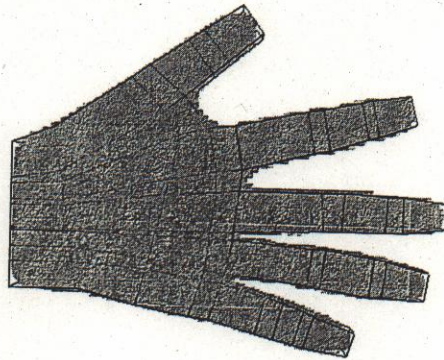
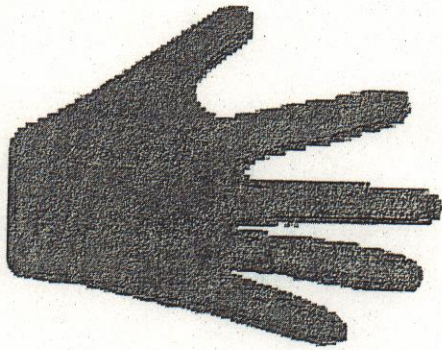
4.3. 自然な手動作における拘束条件の導入

これまでに手指の動きの拘束条件として, 指の可動範囲のみを考慮に入れていた. 人間の自然な手動作を推定の対象とするとき, 各指の中での運動拘束と, 各指の間の運動拘束を導入すると自由度や計算量の減少が期待できるようになる [44].

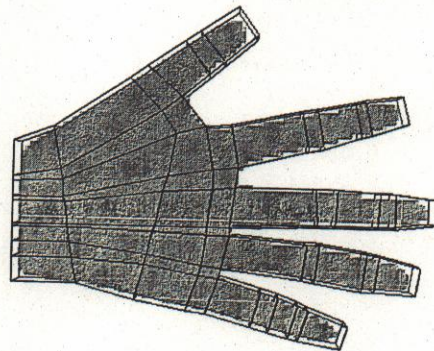
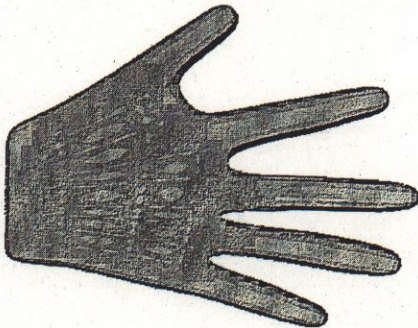
人間は示指, 中指, 環指, 小指の DIP 関節だけを単独で曲げることは出来ず, DIP 関節を曲げようとするとき, 同じ指の PIP 関節も同時に曲がる. これが各指の中での拘束である. このときの関係は以下のように近似できる.

$$\theta_{DIP} = \frac{2}{3}\theta_{PIP}$$

本研究においても, この拘束を導入する. 拘束条件の導入によって, 骨格モデルの自由度は 31 から 27 に低減する. DIP 関節については推定のための力を発生しなくても良くなり, 計算量を減らすことが出来る.



(a) 外ボクセルモデルと表面形状の関係 (b) 外ボクセルモデルと初期格子の関係



(c) 内ボクセルモデルと表面形状の関係 (d) 内ボクセルモデルと初期格子の関係

図 4.6 ボクセルモデルの違いと骨格モデルとの位置関係

これ以外に各指の間の運動拘束も存在する。例えば、示指の MP 関節を前屈すると中指の MP 関節も前に曲がろうとする力が生じ少しだけ前屈する。しかし、これは各指の DIP 関節と PIP 関節との関係のように数式化できていないこと、そして本研究で提案する推定手法は各指の関節推定を独立して行うことから、本改良ではこの拘束関係の導入はしない。

4.4. 粗密探索法による手形状推定の高速化

3.3.1 節に述べたとおり、ボクセルモデルを構成するオクタントのうち“BLACK”と“GRAY”の属性を持つものを表示すると、図 4.6(a)のようにオブジェクトを

完全に外包するサイズでボクセルモデルを生成することが出来る。この状態で生成したボクセルモデルを“外ボクセルモデル”と呼ぶことにする。また、外ボクセルモデルと手の初期格子の大きさの関係は図 4.6(b) のようになり、初期格子からはみ出している部分も一部存在する。一方、“BLACK”の属性を持つオクタンクトのみを表示すると、図 4.6(c) のようなオブジェクトに完全に含まれるサイズのボクセルモデルを生成することが出来る。この状態で生成したボクセルモデルを“内ボクセルモデル”と呼ぶことにする。また、内ボクセルモデルと手の初期格子の大きさの関係は図 4.6(d) のようになり、初期格子内に完全に存在するようになる。このような位置関係をまとめると、手形状が正しく推定されているならば、(1) 内ボクセルモデルは完全に手の表面形状データと初期格子データの内部に存在する、(2) 外ボクセルモデルは手の表面形状データを完全に覆う、の 2 つの条件が成立していることになる。

この条件を用いて、ポリゴン数の少ない初期格子データが内ボクセルモデルの外側になるように力を発生させて収束させるという初期の粗推定を行うことが出来る。観測データとの位置関係による粗推定時のフィッティングのための力の発生方法は図 4.7 のように、内ボクセルモデルの内側に存在する coarse データのエッジの中心に対して、リンクと逆向きの力を発生させる。図 4.7 では、矢印の付いているエッジ中心が内ボクセルモデルの内側に存在していることがわかる。そのため、その位置に初期格子の外向きの力を発生させている。

以下に推定の流れを示す。

Step1: 粗推定のために多視点画像から得られた画像を統合し内ボクセルモデルを生成する。

Step2: 内ボクセルモデルの内側に存在する初期格子の各辺の中心に対して、図 3.13 とは反対向きの力を発生させトルクを計算し、すべての辺の中心が内ボクセルモデルの外側になるように収束計算する。

Step3: 詳細推定のために、外ボクセルモデルを作成する。

Step4: 外ボクセルモデル外側に存在する細分割されたポリゴンに対して、図 3.13 のように力を発生させトルクを計算し、すべてのポリゴンが外ボクセルモ

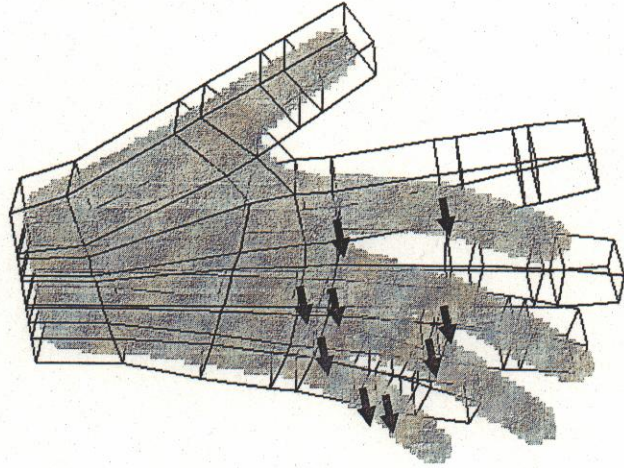


図 4.7 粗推定時フィッティングのための力の発生

デルの内側になるように収束計算する。

Step5: Step4で得られた関節角度を推定結果とする。

3章で提案した手法に比べて、初期格子を用いた荒い推定処理である Step1 と 2 の処理が加えられている。

4.5. 実画像を入力とした手形状推定改良の検証

4.5.1 手の初期位置の決定

第3章の実画像入力による手形状推定では、手の初期位置・姿勢推定を3.5.3節のようにアドホックな方法で行っていた。キャプチャした画像のノイズや、手首部分のデータによってボクセルモデルが予想した形状にならない場合もある。特に掌表面がフラットにならない場合は、必ず初期姿勢の推定に失敗する。このよ

うに初期姿勢の姿勢に失敗した状態で、手形状推定を始めると誤推定が生じると共に推定時間も大きくなる。より安定した手形状推定のためには、3.5.3節で採用した方法で姿勢推定を行う以上に精度良く、骨格モデルの手形状・姿勢・位置を実空間の手の形状・姿勢・位置に一致させる必要がある。

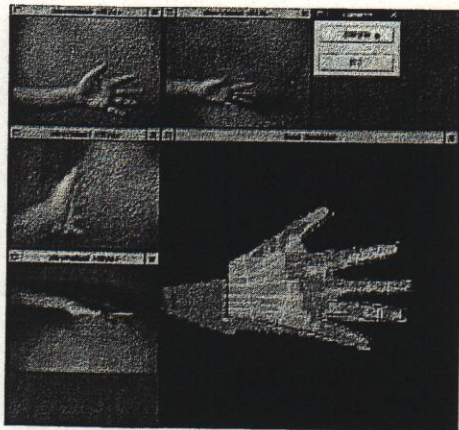
そこで、本改良での実画像入力による手形状推定では、システムが用意した骨格モデルの形状・姿勢・位置にユーザ自身の手の形状・姿勢・位置を合わせる方法を採用した。ユーザは推定初期状態において、リアルタイムにユーザの手形状をあらわすボクセルモデルを見ながら、自分自身の手の形状・姿勢・位置を骨格モデルの中に納まるように調整する。目視によって、骨格モデルとボクセルモデルの一致が得られたと判断されたときを初期状態とし、推定を開始する。

ユーザー側で手の初期位置を合わせることによって、手を開いた状態を初期状態とすることが出来る。これはシミュレーションと同じ初期状態であり、シミュレーションの結果を反映しやすい状態であるとも言える。

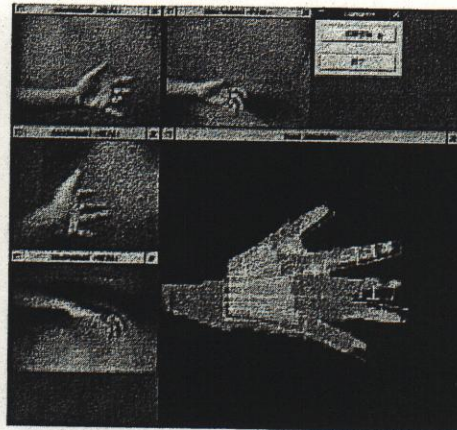
4.5.2 手形状推定の結果

改良をした手形状推定アルゴリズムを用いて、実画像入力による手形状推定を行った。毎回の表面形状計算を行う必要のない粗推定だけを用いて、どの程度実画像の変形に追従できるかを図4.8, 図4.9に示した。

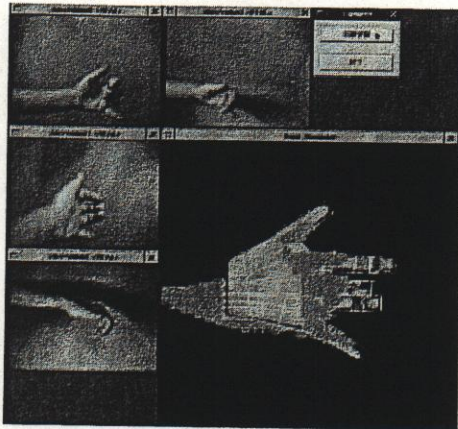
画像において、灰色で表示されている部分が4つのカメラから得られた手の画像空再構成された3次元の手の観測データである。また白い格子は手形状を表現するサブディビジョンサーフェスのコントロールポイントから構成される初期格子を表している。粗推定はこの初期格子のみを用いて推定している。手の形状変化は非常に低速に行った。これはボクセル表現している観測データを推定結果に重畳表示するための描画時間がかかるためである。3章での動画像入力による手形状推定の結果と比べて、大きな動きにも追従出来ていることがわかる。また、推定過程を動画像で確認すると、誤推定からの復帰も3章での結果に比べて高速に行えることが確認できる。



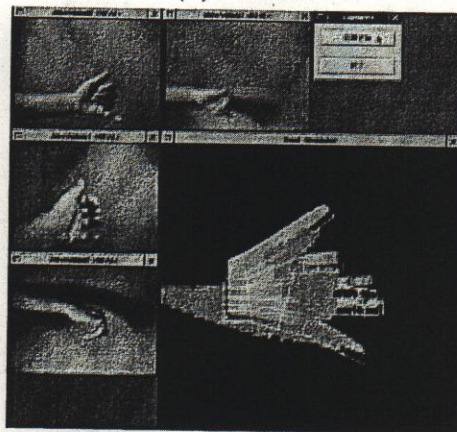
(1) 0.0sec



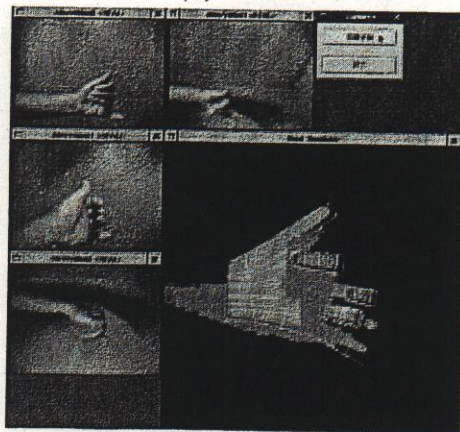
(2) 2.0sec



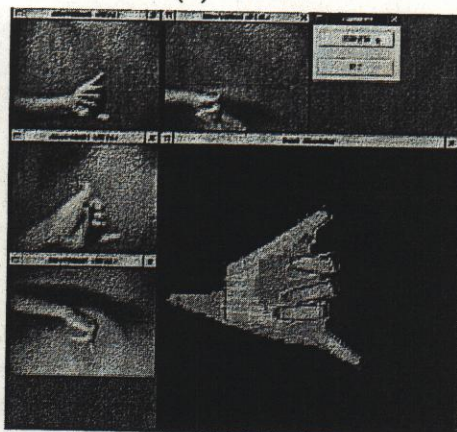
(3) 4.0sec



(4) 6.0sec

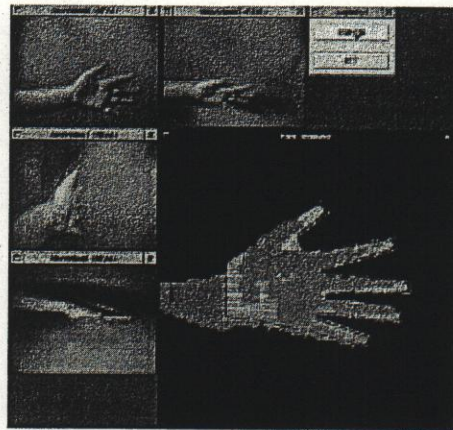


(5) 8.0sec

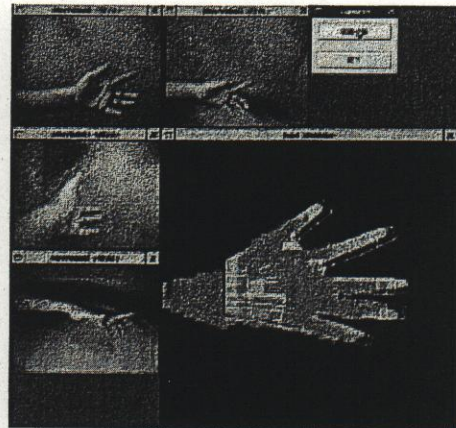


(6) 10.0sec

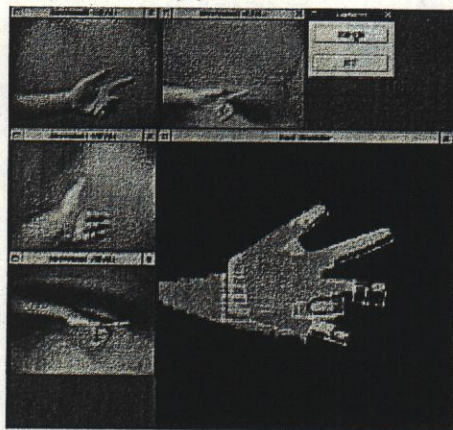
図 4.8 粗推定を用いた実画像手形状推定 (1)



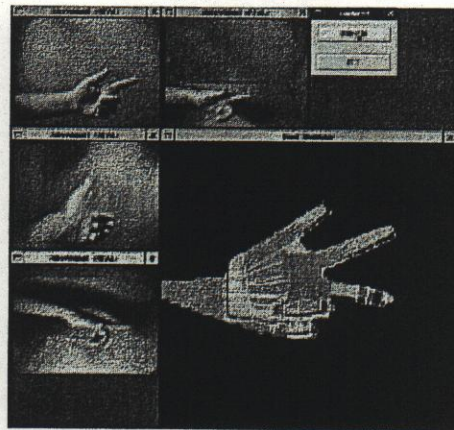
(1) 0.0sec



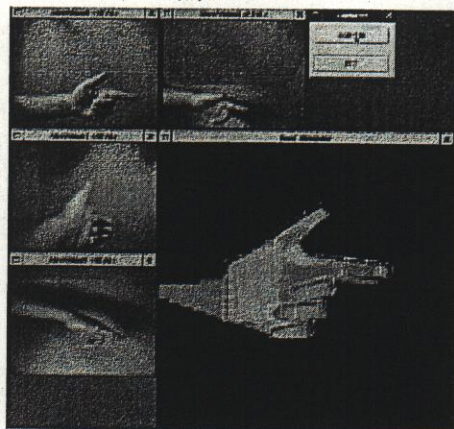
(2) 4.0sec



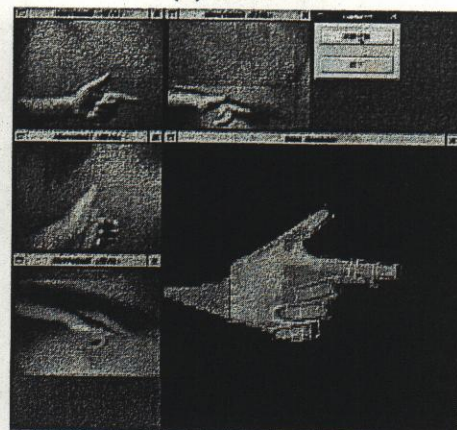
(3) 8.0sec



(4) 12.0sec



(5) 16.0sec



(6) 20.0sec

図 4.9 粗推定を用いた実画像手形状推定 (2)

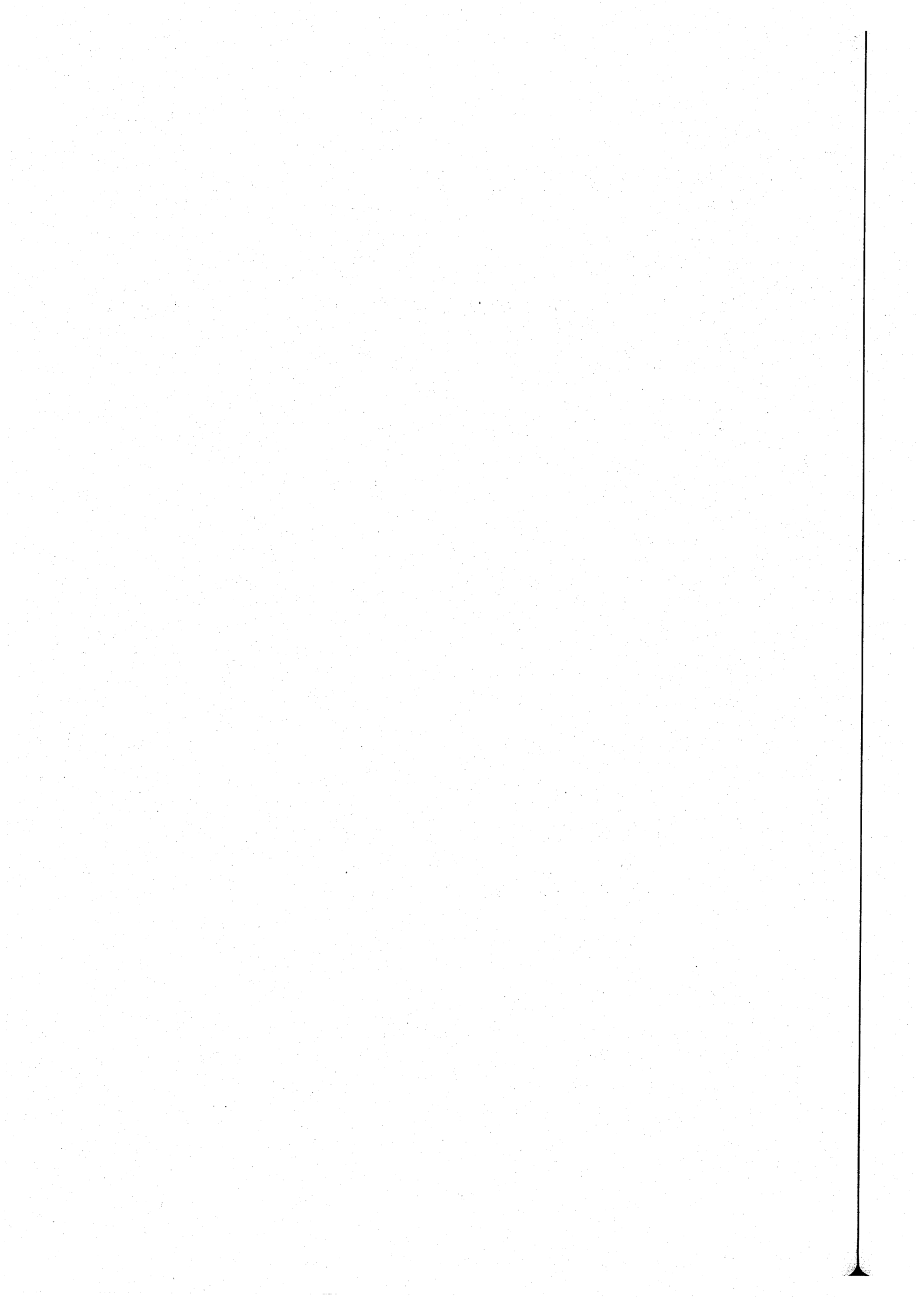
4.6. 本章の結論

本章では，提案手法の処理速度の向上のために粗密探索法の導入を行い，効果の検証を行った。

最初に，粗密探索法を導入するために必要な，2種類の解像度を持つ手の表面形状データの表現にサブディビジョンサーフェスを導入した。特に本研究では，Catmull-Clark サブディビジョンサーフェスを採用している。サブディビジョンサーフェスの導入によって，手の表面形状データを荒さの違いによって複数個持つことが出来るようになった。既存の3次元モデリングシステム (“shade6” or “Lightwave3D 7.5”) を用いて作成した初期手形状は，120個のポリゴンで荒く表現しているが，1回細分割で480個のポリゴン，2回細分割で1920個のポリゴンになり，滑らかな表面形状をデータとして持つことが出来た。また，サブディビジョンサーフェスを導入することで，骨格姿勢の変化に滑らかに追従できる手の表面形状を得ることが出来た。また，自然な手動作における，各指の中での運動拘束を導入した。この拘束条件によって，手の自由度は31から27に減少し，計算量を減らすことが可能になった。

次に，第3.10章で提案した推定手法に粗密探索法を導入する方法について詳細に述べた。追加された処理は粗推定であり，初期格子の内側に内ボクセルモデルが存在するように，内ボクセルモデルの内側に存在する初期格子に対して外向きの力を発生させることによって，関節角度を推定している。

実画像入力による連続手形状推定では，粗推定のみを用いて手形状推定を行い，高速化と実画像への追従性を確認することが出来た。



第5章 手形状推定手法の性能評価

5.1. 本章の趣旨

本章では，第3章，第4章で提案した多視点シルエット画像を用いた手形状推定手法の評価を行う。

最初に，手形状シミュレータを用いて提案手法の様々な設定条件による推定精度の比較を行い，提案手法のもつ性能を評価する．設定条件は，(1) ボクセルモデルの復元レベルの違いと (2) カメラの数とその位置の違いによる推定精度について比較評価を行う．次に，実カメラシステムを用いた手形状推定において，4台目の適切なカメラ位置を評価する．最後に，第4章で提案した粗密探索法の導入による推定精度の向上の効果について，評価を行う．

5.2. ボクセルモデルの復元レベルの違いによる推定精度の比較

本論文で提案する手形状手法は観測データをオクトリーを用いて3次元化しているそのため，3次元化された手の観測データの再構成精度は，手形状の推定精度に影響を及ぼす．本節ではこのボクセルモデルの再構成精度を示すオクトリーの階層数の違いによる，推定精度の比較を行う．本論文では，ボクセルモデルのオクトリーの階層の深さを“オクトリーレベル”と呼ぶ．ボクセルモデルの最小オクタントサイズは1辺の長さはそれぞれ，オクトリーレベル6では $8mm$ ，オクトリーレベル7では $4mm$ である，オクトリーレベル8では $2mm$ ，オクトリーレベル9では $1mm$ である．前述した通りボクセルモデルは実際の手に外接するように作られる．

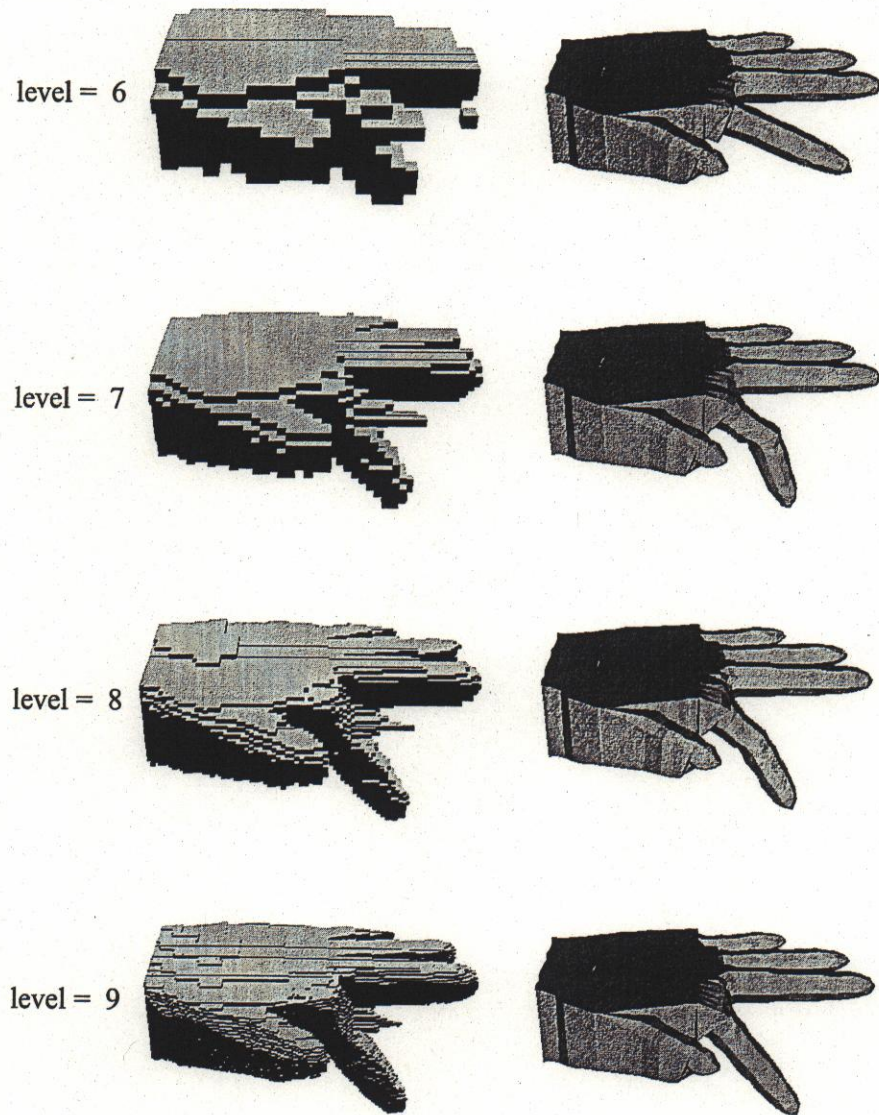


図 5.1 オクトリーレベルが推定結果に与える影響 (左側: ボクセルモデル, 右側: 推定された手形状)

まず最初に、示指の MP 関節を 50[deg] 前屈した手形状に対して作成した各オクトリーレベルのボクセルモデルと、そのボクセルモデルを用いて推定した結果を図 5.1 に示す。オクトリーレベルが大きくなるに従ってボクセルモデルが観測対象の手に近くなるように復元されているのが分かる、この推定対象の手形状ではオクトリーレベルが大きくなるに従って推定精度も上がっていく様子が示されている。

次に複数の目標手形状を用いて、推定精度に対するオクトリーレベルの影響を確認するために実験を行った推定実験では基本のアルゴリズムの性能を評価するために、粗密推定は行わず、詳細推定のみを行っている。手形状はサブディビジョンサーフェスを用いて表現している。手形状シミュレータを用いて作成した評価のための 4 種類の手形状は、以下の通りである。

Pose-E 示指の MP 関節を 8[deg] 内転、中指の MP 関節を 15[deg] 外転、環指の MP 関節を 10[deg] 内転、小指の MP 関節を 5[deg] 外転

Pose-F 親指以外の 4 指の MP,PIP 関節をそれぞれ 10[deg] 前屈, DIP 関節を 6.7[deg] 前屈

Pose-G 示指の MP 関節を 10[deg] 外転し更に 60[deg] 前屈, 小指の MP 関節を 20[deg] 外転し更に 25[deg] 前屈

Pose-H 示指の MP 関節を 30[deg]・PIP 関節を 20[deg]・DIP 関節を 13.3[deg] それぞれ前屈, 中指の MP 関節を 30[deg]・PIP 関節を 25[deg]・DIP 関節を 16.7[deg] それぞれ前屈, 環指の MP 関節を 30[deg]・PIP 関節を 30[deg]・DIP 関節を 20[deg] それぞれ前屈, 小指の MP 関節を 30[deg]・PIP 関節を 35[deg]・DIP 関節を 23.3[deg] それぞれ前屈

図 5.2 ~ 図 5.5 は推定過程における、各繰り返し毎の収束率の推移を示している。

オクトリーレベルが低いときには、生成されるボクセルモデルが実際の手形状より大きく作られるため、微小な変形の Pose-E, Pose-F では早く収束する。しかし、大きな変形である Pose-G, Pose-H では、誤った推定や推定の停止の原因とも

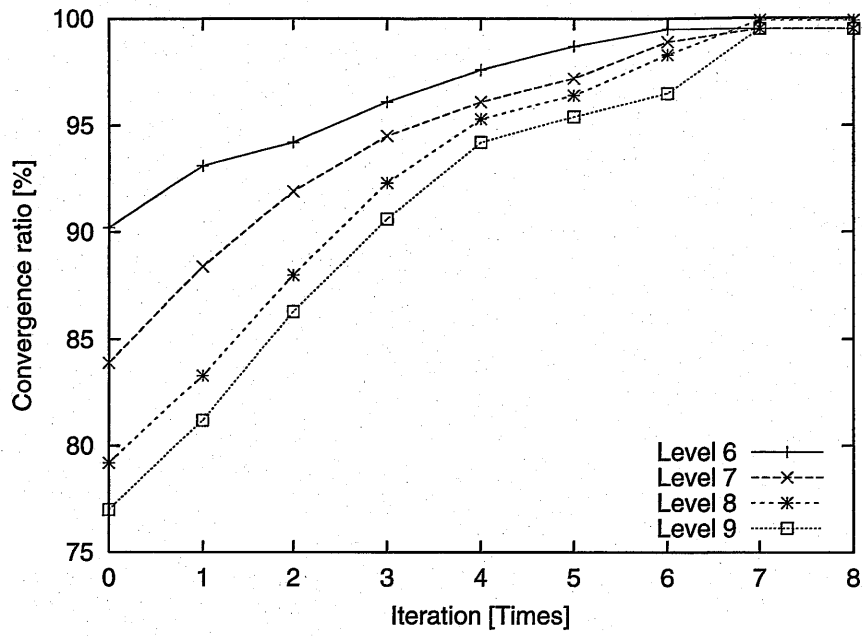


図 5.2 繰り返しごとの収束率の推移 (Pose-E)

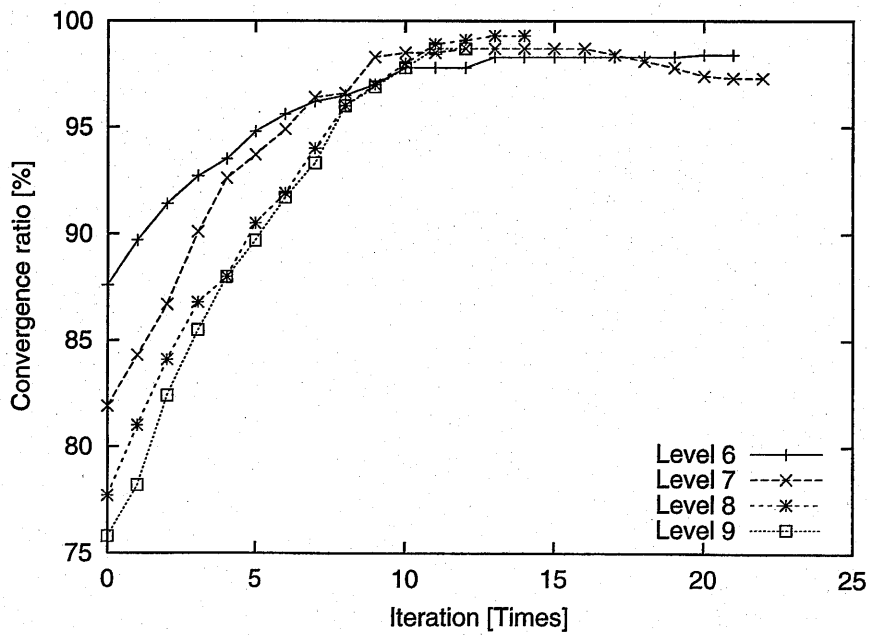


図 5.3 繰り返しごとの収束率の推移 (Pose-F)

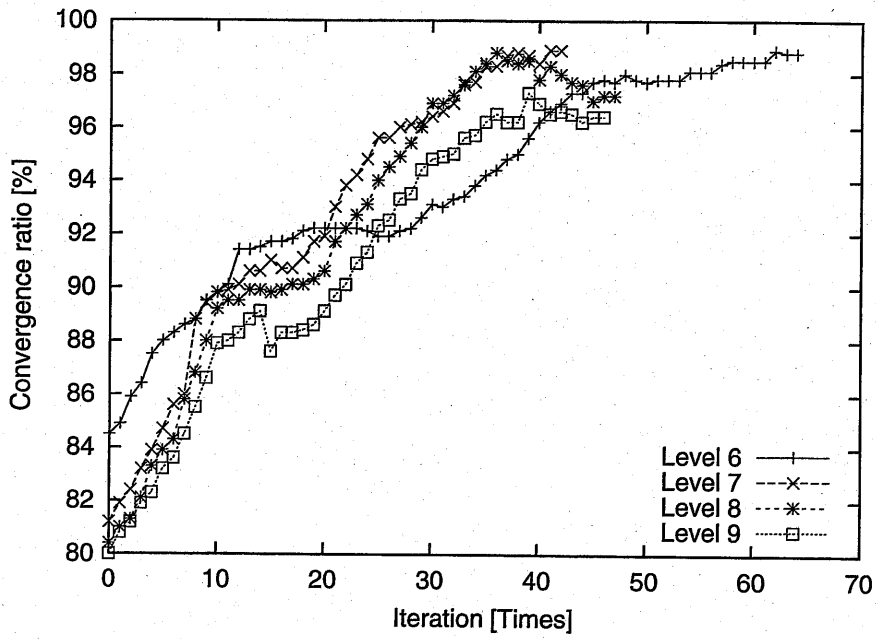


図 5.4 繰り返しごとの収束率の推移 (Pose-G)

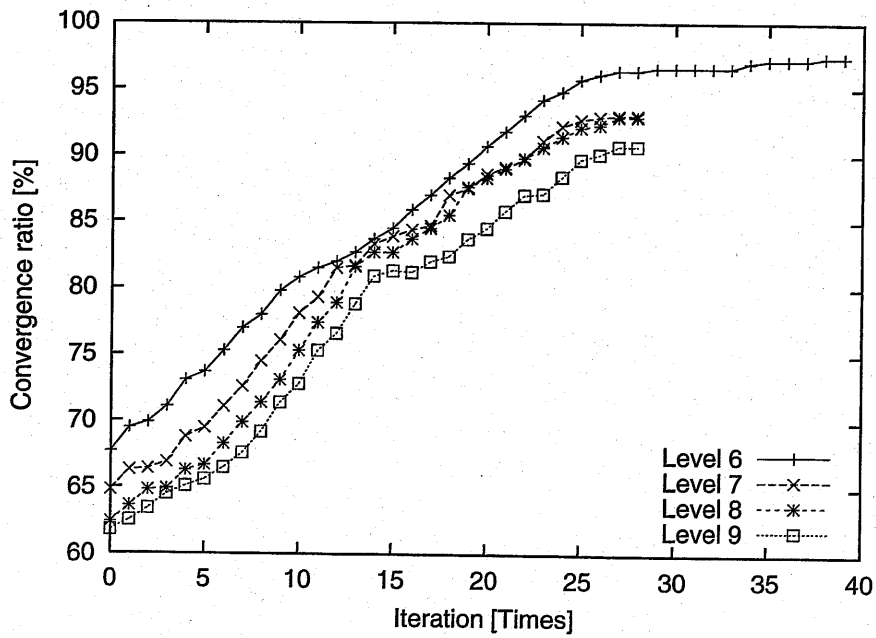


図 5.5 繰り返しごとの収束率の推移 (Pose-H)

なる。しかしこの収束率は収束判定の指標を与えるものに過ぎず、実際の推定精度を現しているものではない。図 5.1 のボクセルモデルをみてもわかるように、オクトリーレベルが低くなると指の太さは実際の手形状より太く再構成され、そのため収束率が 100% になったとしても推定結果が正しいとはいえないからである。一方、オクトリーレベルが高くなるとボクセルモデルは高精度に実際の手形状を再構成することが出来る。そのため簡単に収束率が 100% になることはめったにない、しかし完全に 100% にならずとも推定結果は実際の手形状に近いものと成りうる。

推定精度を定量的に評価するために、以下のように推定誤差を定義する。

$$error = \sum_i dist(epos_i, tpos_i) \quad (5.1)$$

ここで、 $epos_i$ は i 番目の関節位置、 $tpos_i$ は与えられた手形状における i 番目の関節位置、 $dist(epos_i, tpos_i)$ は $epos_i$ と $tpos_i$ 間のユークリッド距離である。

図 5.6 ~ 図 5.9 はオクトリーレベル 6~9 における各繰り返し毎の推定誤差の推移を示している。どの手形状においても、オクトリーレベルが高くなると推定精度が良くなることがわかる。

図 5.10 は各オクトリーレベルにおけるボクセルモデル生成の時間を示している。レベルが 6 から 9 に上がるに従って、処理時間も増えていく。オクトリーレベルが 9 の時に、処理時間が急激に増大している。本研究では、各オクタントをカメラ平面に射影する際、高速化のためにあらかじめ計算しておいたルックアップテーブルを用いている。しかしメモリサイズの制約からオクトリーレベル 9 ではすべてのルックアップテーブルを作成することが出来ないため、このような処理時間が増大してしまう。

これらのグラフ (図 5.2 ~ 図 5.10) は観測データの復元精度を示すオクトリーレベルが推定精度と推定速度へ大きな影響を及ぼすことを示している。手形状推定をどのような目的で応用するかにより、精度優先か速度優先かが異なり、オクトリーレベルの設定も考慮する必要がある。手形状は 1[deg] の関節角度の違いによっては大きく左右されないことより、どの程度の曖昧さを手形状推定に与えることができるかによって、推定速度やロバスト性の向上を図ることができると考えられる。

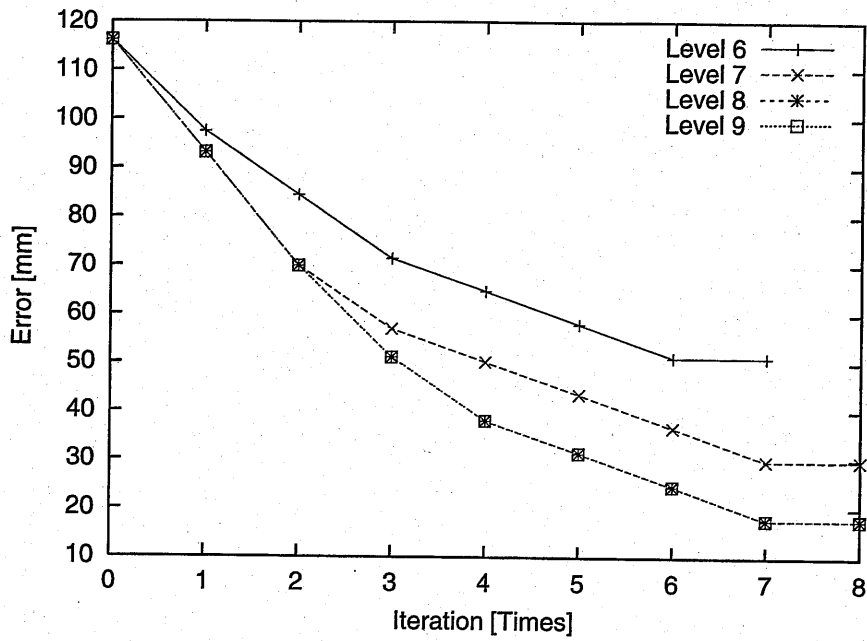


図 5.6 繰り返しごとの推定誤差の推移 (Pose-E)

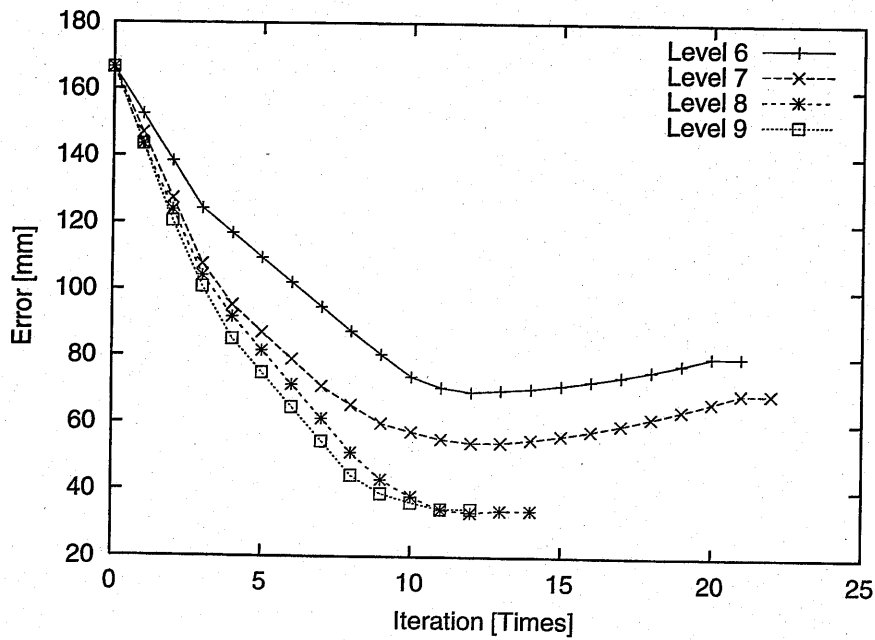


図 5.7 繰り返しごとの推定誤差の推移 (Pose-F)

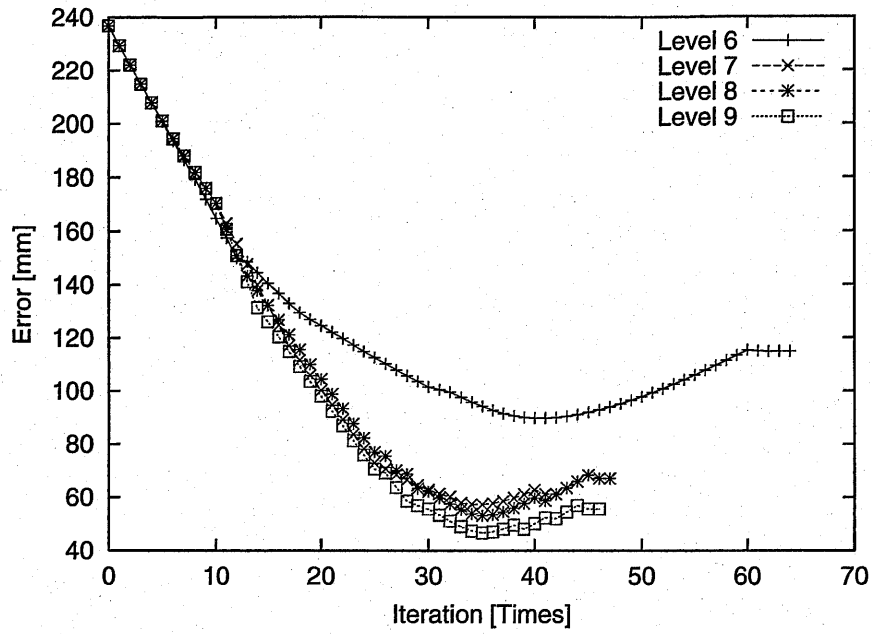


図 5.8 繰り返しごとの推定誤差の推移 (Pose-G)

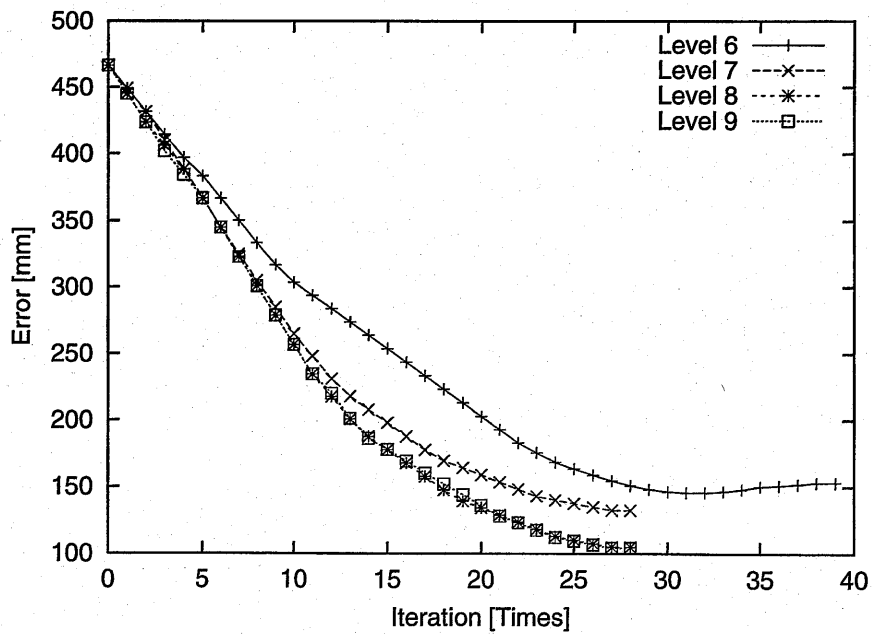


図 5.9 繰り返しごとの推定誤差の推移 (Pose-H)

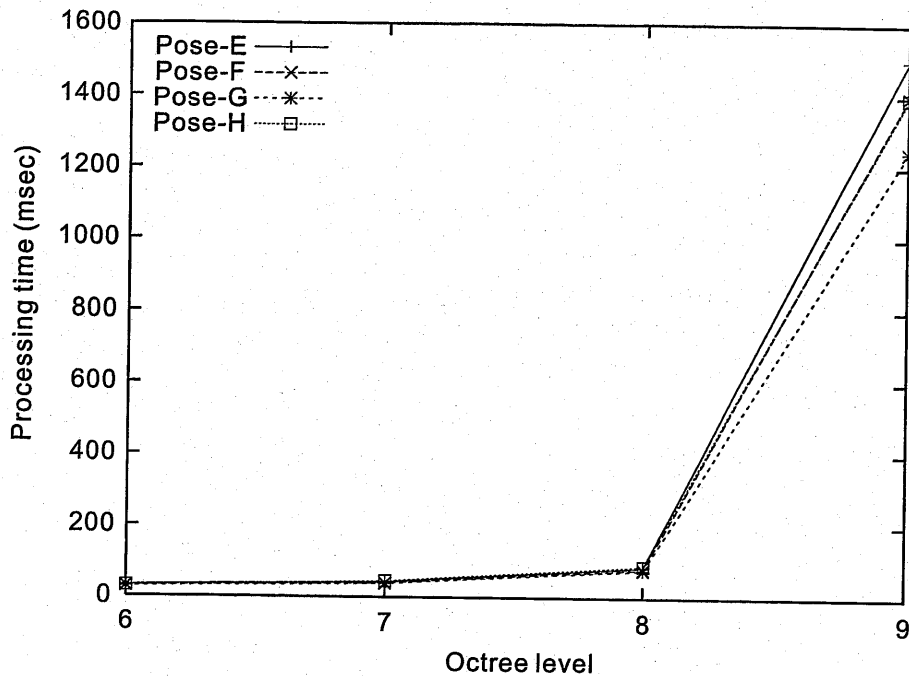


図 5.10 オクトリーレベルの違いによるボクセルモデル作成に要する時間

5.3. カメラの台数の違いによる推定精度の比較

5.3.1 偽の指の発生による誤推定

カメラの位置と数は、生成されるボクセルモデルの精度に大きな影響を与える。そこでカメラの位置を確認するためのシミュレーションを行った。

図 3.15 の Pose-C の手形状を、図 5.11 の左側に示すカメラ位置でキャプチャしたシルエット画像を図 5.11 の右側に示している。

(1),(2),(3) の 3 つのカメラを用いてボクセルモデルを生成した結果を図 5.12(a) に示す。自己オクルージョンが原因で偽の示指が生成されているため、示指の推定結果も間違っただけのものになっている。(4) の位置のカメラを追加すると、図 5.12(b) に示すようにボクセルモデルの偽の示指は生成されず、推定結果も正しいものとなる。この実験はカメラの位置と数は正確なボクセルモデルを生成するときに必要なファクタとなることを示している。

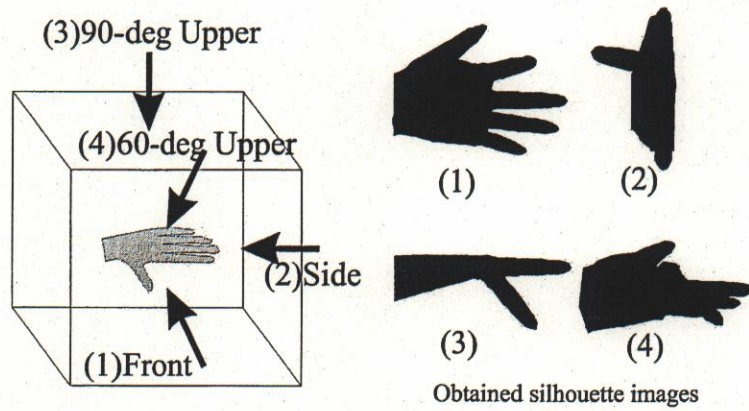


図 5.11 カメラの視点とそれぞれのカメラで得られるシルエット画像

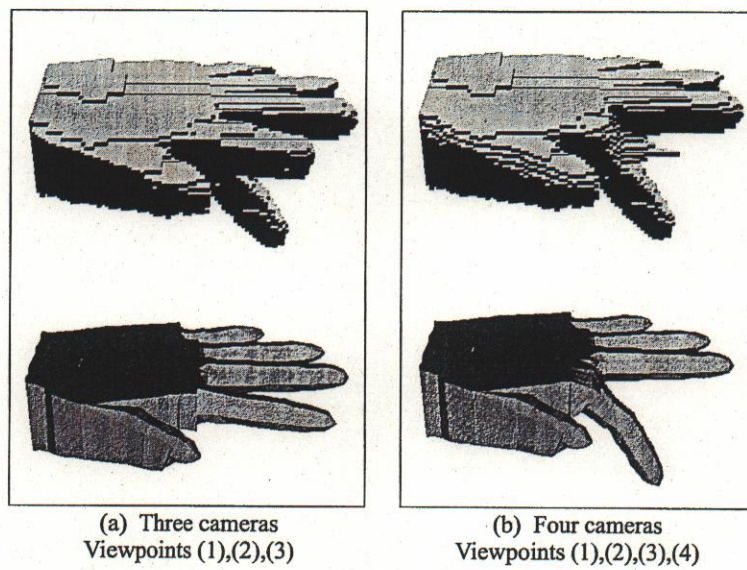


図 5.12 カメラの位置と台数が推定結果に与える影響
 (上側: 再構成されたボクセルモデル [level=8], 下側: 推定された手形状)

5.3.2 一般的なカメラ台数による推定精度の比較

5.3.1節でカメラの位置と数が重要なファクタとなることを示した。そこで、本節ではカメラの台数が、手形状推定に及ぼす影響をシミュレーションによって確認する。

カメラの台数のみを評価するために、それぞれの台数でのカメラの配置を正多面体の頂点となるようにし、カメラの台数が4, 6, 8, 12, 20の場合について手形状推定結果を評価した。多視点カメラの配置として最も基本となる正面・側面・上面にカメラを配置したカメラ台数=3の場合を加えて、カメラ台数を6通りに変化させ、それぞれの場合について評価を行った。推定に用いた手形状は Pose-E~Pose-Hを用いた。図5.13に Pose-Hの手形状を推定したときの、カメラ台数の違いによる推定結果を示す。また図5.14には4種類の手形状について、各カメラ配置での推定誤差のグラフを示す。

図5.13を見ると、カメラ台数が3台または4台の場合は、ボクセルモデルにおいて偽の指が発生しており、またカメラ台数が6台の場合には掌部分に不要な厚みが残っている。図5.14の Pose-Hの推定誤差をみると、カメラ台数が8台以上になると急激に推定誤差が減少している。しかし、Pose-Fではカメラ台数が8台のときに推定誤差が最も大きくなっているため、カメラ台数が8台で十分であるとは言い切れない。図5.14からは、どの手形状を推定する場合においてもカメラ台数が12台以上になると推定誤差が小さくなり安定していることが分かる。このように、正多面体の頂点にカメラを配置する条件においては、カメラの台数が12台以上で推定が十分安定すると言える。しかし、一般的にこの実験結果から必要なカメラ台数の閾値を決定することは難しい。この理由は、人間の手形状が凸多面体では表すことが出来ないこと、また指を自由に動かす場合自己オクルージョンの発生がボクセルモデルの生成に大きく影響するため、カメラ台数よりも位置のほうが推定誤差に強い影響を与える為である。



図 5.13 カメラ台数の違いによる手形状推定の結果
 (1 段目: ボクセルモデル, 2 段目: ボクセルモデルと推定結果の重畳表示, 3 段目: 推定結果)

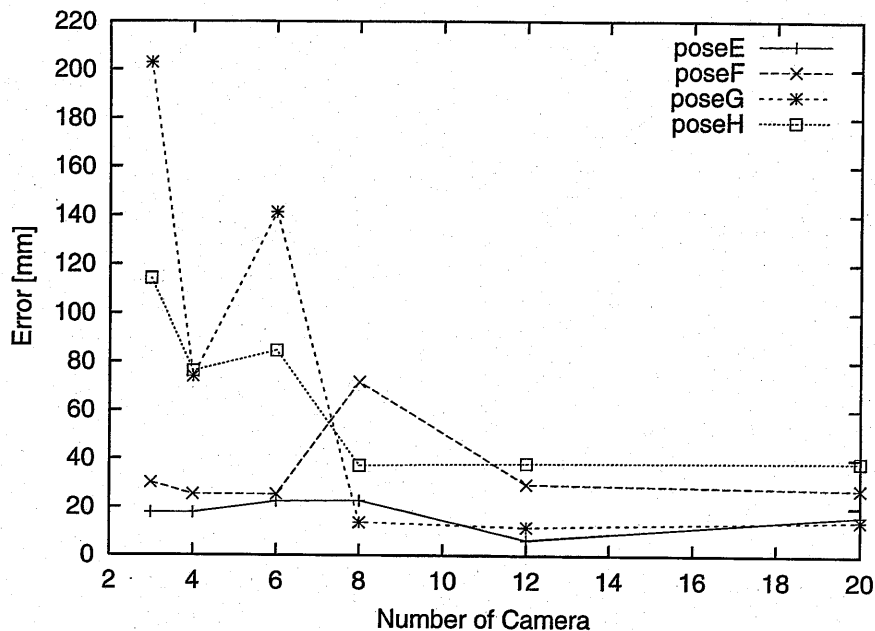


図 5.14 カメラの台数の違いによる推定誤差

5.4. カメラ位置の違いによる推定精度の比較

本研究で構築した実カメラシステムでは、PCにビデオキャプチャカードを4枚挿して、4台のカメラからの画像をキャプチャしている。このカメラ4台という台数制限は、3.5.1節で述べたとおり、現在の本システムのハードウェア構成上の制限によるものである。4台のカメラを用いる実カメラシステムにおける、最適なカメラ配置についてシミュレーションにより評価を行った。

基本のカメラ配置を三面図を得るため正面、側面、上面の3つに決定し、もう一台の最適なカメラ配置を評価するという方法で行った。図5.15に示すように、極座標表現を用い、手形状が原点にあるとして4台目のカメラ位置を1~12の12種類設定する。設定したカメラ位置は、正面カメラを基準とし、 $x-z$ 平面に $90[\text{deg}]$, $45[\text{deg}]$, $-45[\text{deg}]$, $-90[\text{deg}]$ とその位置から y 軸方向へ $45[\text{deg}]$, $-45[\text{deg}]$ それぞれ移動したカメラ位置である。手形状の中心からの距離は $400[\text{mm}]$ とし、カメラは原点にある手形状中心を向いているとする。それぞれのカメラ位置から見た、Pose-Hの手形状を図5.16に示す。

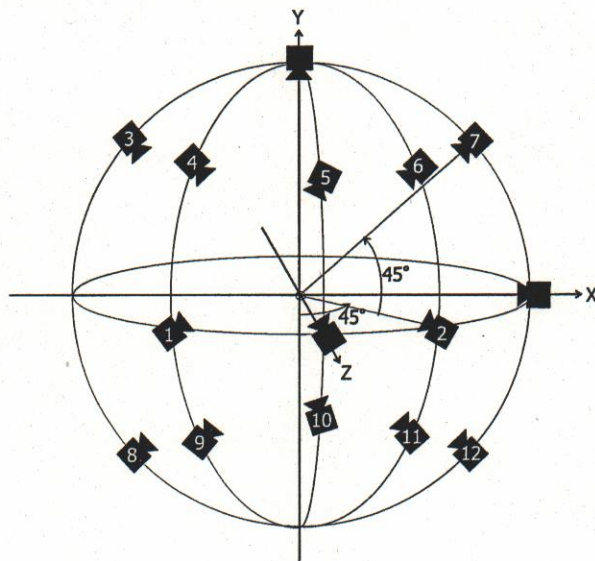


図 5.15 カメラの位置

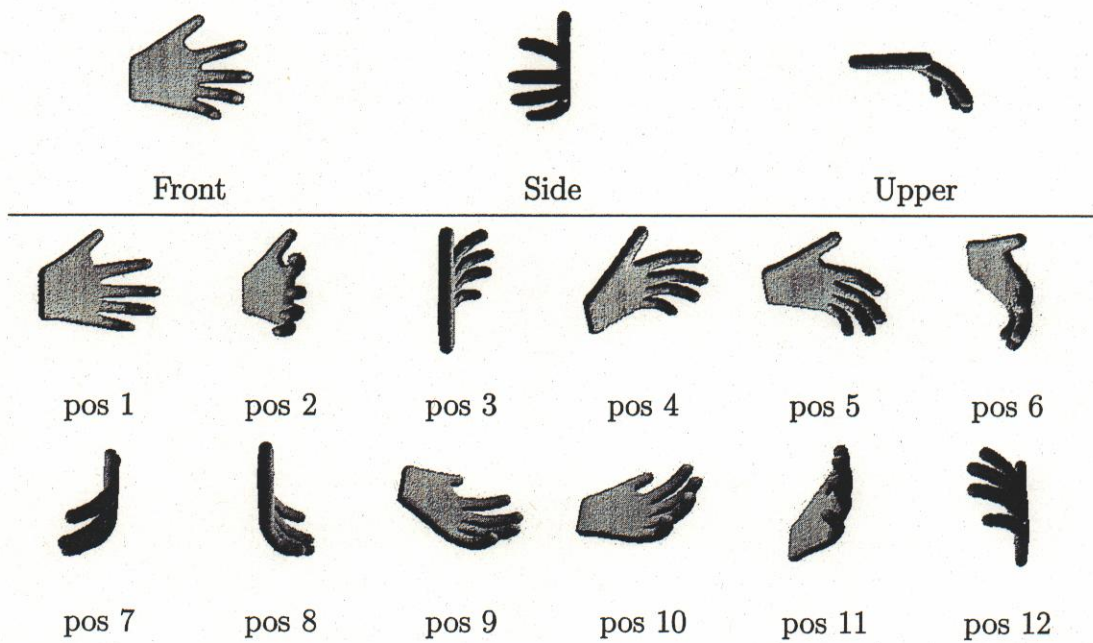


図 5.16 シミュレータで作成した手形状

基本のカメラ位置のシルエット画像(3枚) + それぞれのカメラ位置のシルエット画像(1枚)の合計4つのシルエット画像を用いて, Pose-E~Pose-Hの手形状のボクセルモデルを作成し手形状推定を行った結果を, 図5.17に示す. それぞれの手形状において, 推定誤差が最小になるカメラ位置が異なるため, この4つのグラフからは, 最適なカメラ位置がどれであることを示すことは出来ない. そこで, Pose-E~Pose-Hそれぞれの各カメラ位置での推定誤差を加えた結果を, 図5.18に示す. このグラフから, 実験に使用した4種類の手形状においてはカメラ位置=5(正面カメラと上面カメラの間45[deg]にカメラを配置)した場合に, 最も推定誤差が少なくなるという結果が得られた. この結果は, 第3章での, 実画像カメラシステムにおけるカメラ配置が, 良好な推定結果を得ることが出来るカメラ配置である可能性が高いことを示している.

これらの評価は, どのような手形状に対しても最適な推定結果を与えるカメラ配置を決定することは不可能であり, 複数台のカメラを配置した上で, 自己オクルージョンがなるべく少ないカメラを適応的に選択しながら推定を行う必要があることを示唆している.

5.5. シミュレーションによる手形状推定改良の検証

本節では, 第4章で提案した粗密探索法の導入による推定精度の向上の効果について, 評価を行う. 推定させた手形状は, 5.2節で作成したPose-Fを用いた.

図5.19に粗密探索法の導入有りと無しの場合それぞれにおける各繰り返し毎の推定誤差を示す.

従来手法と処理時間は同程度であるが, 推定精度が改善されていることがわかる. また, 粗密推定の19[msec]迄が粗推定部である. グラフより粗推定部分では, 推定誤差は推定が始まると急激に減少することがわかる. また, 推定精度も粗推定だけでも改善前の手法に比べて良いことがわかる. シミュレーションによる評価では, 推定精度の向上が確認できたが, 処理速度の向上はほとんど図られなかった. これは, 手の表面形状の包含判定の際に, 大きなサイズの行列計算を推定の繰り返し毎に行う必要があり, その部分に処理時間の多くを費やしている

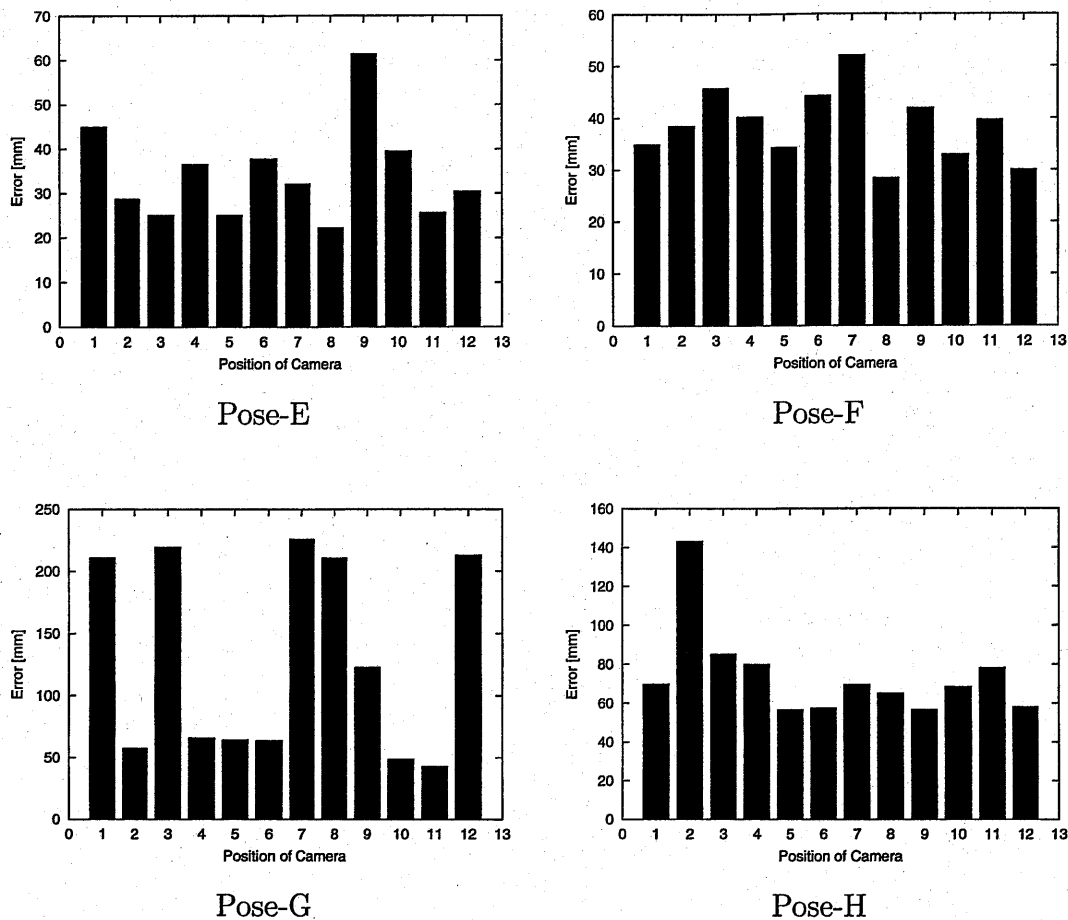


図 5.17 それぞれのカメラ位置を用いた場合の推定誤差

ためであることが理由である。

5.6. 本章の結論

本章では、提案した多視点シルエット画像を用いた手形状推定手法の評価を行った。

最初にボクセルモデルの復元レベルの違いにより、推定精度と処理時間にどのような差が見られるかを検証した。その結果、手形状推定をどのような目的で応用するかにより、精度優先か速度優先かが異なり、オクトリーレベルの設定も考

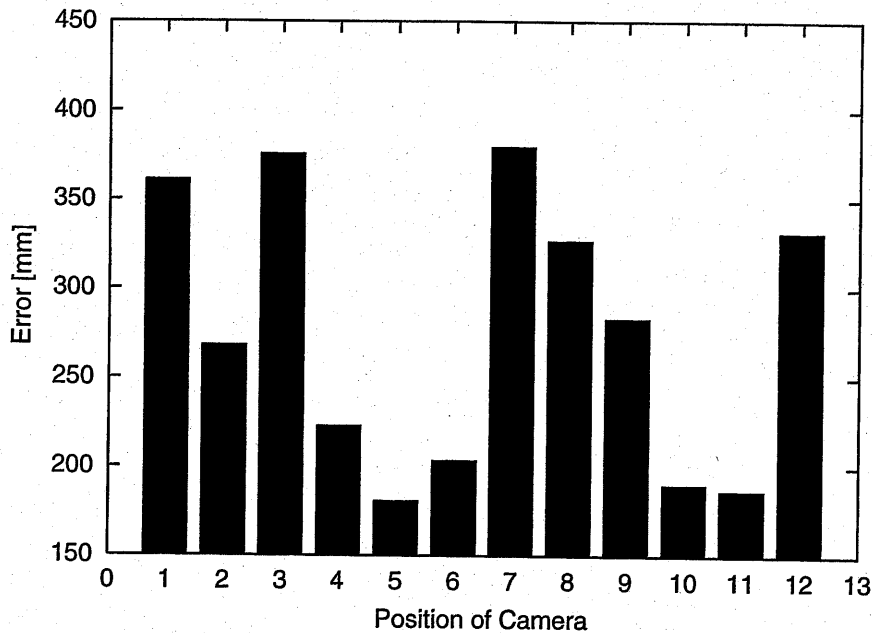


図 5.18 カメラの位置の違いによる手形状推定結果

慮する必要があることが示された。

次に、カメラの数の違いによる推定精度の比較を行った。カメラ配置を5種類の正多面体の頂点にあると設定し、それぞれにおける推定精度の評価を行った。推定した4種類の手形状においてはカメラ台数が12台を越えると、推定誤差が小さくなり安定すると言う結果が得られた。しかしそれ以上に、カメラの位置が推定精度に大きな影響を及ぼしていることも示された。

次に、実カメラシステムのハードウェアの制約を考慮した4台カメラの、適切な配置を評価した。評価の結果からは、現状の実カメラシステムの配置が適切であることが示されたが、同時に、どのような手形状に対しても最適な推定結果を与えるカメラ配置を決定することは不可能であり、複数のカメラからオクルージョンが少ないカメラを選択しながら推定を行う必要性を示した。

最後に、第4章で提案した粗密探索法の導入による推定精度の向上の効果について、評価を行った。シミュレータを用いて評価を行った結果、推定精度の向上が確認できたが、処理速度の向上は図られなかった。これは、手形状表面を求め

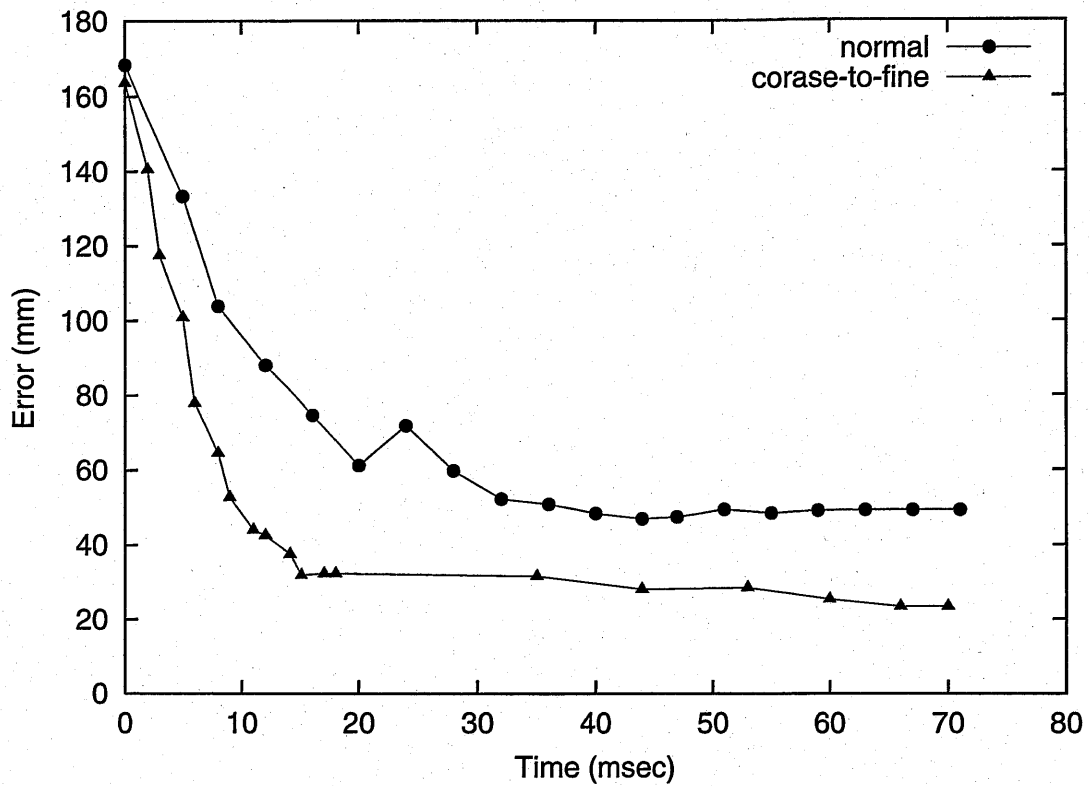


図 5.19 推定誤差の比較

るために、大きなサイズの行列計算を推定の繰り返し毎に行う必要があり、その部分に処理時間の多くを費やしているためであることが理由であった。しかし、粗推定のみでも改良前に比べて推定精度は良く、処理時間も約 1/4 になった。

第6章 コンピュータインタフェース としての手形状推定の応用

6.1. 本章の趣旨

本章では、第3章、第4章で提案した手形状推定方法のインタフェースへの応用の可能性を検証するために、手形状推定の結果をコンピュータインタフェースの一つバーチャルクレイモデリングシステムの入力として取り扱う。

最初に、検証の為のアプリケーションとして、提案手法により推定された操作的な手動作によってバーチャルクレイの変形を行うバーチャルクレイモデリングシステムのプロトタイプを実装する。このプロトタイプシステムにおいて操作対象となるバーチャルクレイは、改良推定手法で手の表面形状データに用いた、Catmull-Clark サブディビジョンサーフェスによって表現される。また、バーチャルクレイは FFD(Free Form Deformation) を応用した変形方法によって変形される。

次に変形シミュレーションを用いて、実装したバーチャルクレイモデルが、どの程度滑らかにその形状を変形することが出来るかを示す。

最後に、実空間の手動作を入力としてバーチャルクレイの変形を行い、その操作性の検証を行う。

6.2. バーチャルクレイモデリングインタフェースプロトタイプ

6.2.1 仕様

本研究では、提案した手形状推定のインタフェース入力への応用の一つとして、仮想空間中の塑性体を実空間の手の動きで変形する、バーチャルクレイモデリングインタフェースを実装する。実装するシステムの仕様は以下の通り、
入力

- 4台の CCD カメラによる多視点画像入力 (手形状推定システムと同じ)
- 観測空間での手動作の際の手形状を直接入力

クレイモデル

- Catmull-Clark サブディビジョンサーフェスによるバーチャルクレイの表現
- direct FFD によるバーチャルクレイの変形

変形を含むバーチャルクレイのモデル化については、6.3 節で詳細に述べる。

6.2.2 処理の流れ

バーチャルクレイモデリングシステムでのバーチャルクレイの変形処理の流れは以下の通りである。図 6.1 にフローチャートを示す。

Step1: 多視点カメラシステムにより手をキャプチャする。

Step2: 手形状を推定する。

Step3: 前フレームで推定した手形状と、現フレームで推定した手形状の位置・形状の違いを基にして、手とバーチャルクレイとの干渉チェックを行う。

Step4: バーチャルクレイ表面位置を変更する。

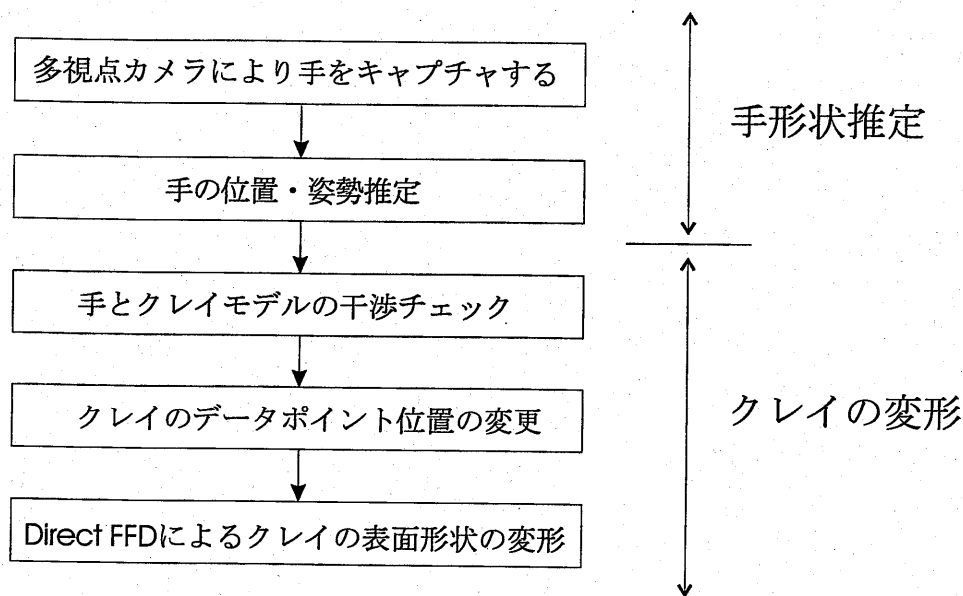


図 6.1 バーチャルクレイ変形のフローチャート

Step5: directFFDにより、バーチャルクレイの表面形状全体を滑らかに変形する

このように、システムは、Step2までの手形状推定部分と残りのクレイの変形部分のように、大きく2つの処理に分かれている。この処理サイクルを繰り返すことにより、手動作によるバーチャルクレイの変形を実現している。

6.3. バーチャルクレイのモデル化

6.3.1 サブディビジョンサーフェスによるバーチャルクレイの表現

実空間中の3次元物体をモデリングする方法は、ワイヤーフレームモデル、サーフェスモデル、そしてソリッドモデルと大きく3つに分類することが出来る。まず最初にそれぞれの特徴について概観する [45]。

ワイヤーフレームモデル ワイヤーフレームモデルは3次元形状を稜線(直線でも曲線でも可)と頂点だけで表現する。また、稜線は始点と終点の座標を持

つ. このような構造はコンピュータの中で表現するのも簡単で, 必要とされるメモリも少なくてもよい. このように少ない計算処理で操作が可能なことから, コンピュータの計算能力が低かった昔にはよく使われるモデルであった. 処理のオーバーヘッドが小さいことから, 現在でも3Dグラフィックスアプリケーションなどにおいて, 編集時の表示方法などとして利用されている. しかし, このモデルでは立体を一意に解釈することが出来ないという欠点がある. また, 面の形状を持たないため, 立体としては不完全である. このため, 立体の体積・表面積・重心などを求められないことや, 他の物体との干渉などをシミュレーションできないという問題点がある.

サーフェスモデル このモデルでは, ワイヤフレームモデルのデータに加え, 面のデータを持つ. 立体の表面のデータはあるが, 中は空になっている. 位相的な構造として, 面はそれを構成する稜線データも持つ. 面の形状データをどのような数学的表現で持つかはシステムにより異なる. 例えば, バーチャルクレイを表現するモデルとしては, 物体の面を小さなポリゴンに分割して表現するポリゴンモデルや円柱面・球面などの2次曲面のほか, Bézier 曲面・NURBS などのようなパラメトリック表現による自由曲面や, 陰関数表現による形状境界面を定義する方法による表現などが用いられる. Debunne[46] や Hsu[47], Wyvill[48] らは物体表面を多数のパラメトリックなパッチから構築し, その制御点をコントロールすることで形状表面を滑らかに変形するモデルを採用している. 松宮らによる陰関数を用いた粘土のモデルでは [49], データグローブを用いて推定した手の表面による粘土表面形状の変形は, 表面をあらわしている陰関数を用いて数学的に計算することが出来る. しかし, サーフェスモデルは立体を面の集まりとして表現しているだけであり, 立体の表現方法としては厳密ではない. 例えば曲面を切断した場合には, その切断面を再構成するなどの手間が必要になる. 位相の変化に対応するには, 別処理が必要となり計算コストは高くなる. 他の物体との干渉は, 干渉部分に関してはシミュレーションすることが可能である, しかし体積一定変形をシミュレーションすることは不可能である.

ソリッドモデル 空間中の3次元物体を完全に表現できるモデルがソリッドモデルである。サーフェスモデルは中空であったが、ソリッドモデルは中身も詰まっている。このため、解析・加工などの様々な場面で厳密な物理シミュレーションが可能になる。厳密な物理モデルに基づく形状モデリングに関する研究は、Pentlandらによる有限要素法に基づく弾性体の変形を扱うもの [50] や、Terzopoulosらによる弾性理論に基づいて物体の変形を扱うもの [51]、また小田らによる粒子ベースでそれらの力学関係を計算することによる変形 [52] などがある。

ソリッドモデルのコンピュータ内部での表現方法には、多くの方法が提案されてきている。例えばバーチャルクレイを表現する場合は、物体内部に質点を配置する方法やボクセルを用いて表現する方法などがある。ボクセルを用いて、厳密な物理モデルを用いずに体積一定変形を実現する研究もこれまでに数多く報告されている [53, 54, 55]。そのうち、荒田らはセル構造オートマトンをボクセル空間に応用した仮想粘土モデリングの提案を行っている。物理モデルに対して厳密に忠実ではないが、実時間で計算できる手法を報告している。しかしまだ、モデルの提案のみでインタフェースとして取り扱えるまでには至っていない。一方、McDonnellらは物体内部に質点を配置するモデルで粘土を表現しており、それらの質点の初期位置はサブディビジョンソリッドで作成している [56, 57]。また質点の位置は、質点間にバネとダンパを仮想的に配置しそれら要素の運動によって変更しており、表面形状はリアルタイムにかつ滑らかに変形できる。

他の方法と比較して、ソリッドモデルではデータ構造が複雑になるため、形状の変形や表示に要する時間も大きくなるという欠点を持っている。

これらをバーチャルクレイを表現するという観点から考えると、ワイヤーフレームモデルでは形状表面と操作をする手との接触判定が正しく出来ないことから、不適切なモデルであると言うことは明らかである。そこで、サーフェスモデルとソリッドモデルのうち、従来のバーチャルクレイモデリングシステムにおける粘土のモデル化方法を、変形を行うという観点から比較を行い表6.1にまとめた。

表 6.1 3次元オブジェクトのモデリング方法の変形における比較

		体積一定変形	変形速度	滑らかさ
サーフェスモデル	陰関数表現	×	○	○
	ポリゴンパッチ	×	○	×
	パラメトリック表現	×	○	○
ソリッドモデル	質点モデル	○	△	○
	ボクセルモデル	○	×	×

表 6.1 に示すように、サーフェスモデルでは体積一定変形を行うことは不可能である。一方、ソリッドモデルではサーフェスモデルに比べて、形状変形に要する時間が大きくなる。本章で評価のために作成したプロトタイプシステムは、製品設計のための 3次元自由形状の入力としても応用可能なバーチャルクレイモデリングを目指しており、実空間の粘土の振る舞いを完全にシミュレーションすることを目的とはしていない。そのため、体積一定変形は不必要であり、デザイナーが操作した部分の周辺を滑らかに変形することが出来れば良い。このような理由から、プロトタイプシステムではサーフェスモデルを採用しオブジェクトの表面を、コントロールポイントを基本としたパラメトリック表現によって形状表面をモデル化する。特に、第 4 章の手の表面形状データの表現に用いた、パラメトリック表現の一種である Catmull-Clark サブディビジョンサーフェスによるサーフェスモデルによってモデル化する。

6.3.2 FFD を応用したバーチャルクレイの変形

デザイナーのオペレーションによって、バーチャルクレイの表面形状を表すデータポイントの位置の変更が生じた場合、オブジェクトの表面形状全体を再計算する必要が生じる。従来、式 (4.3) のようにパラメトリック表現されたモデル形状を変形する場合、コントロールポイントを定義しその位置の変更によって、形状変形を行なう FFD という手法がしばしば用いられる。FFD の基本手順を以下に

示す。

Step1: コントロールポイントの位置を変更する。

Step2: 定義済みの重み行列を各コントロールポイントベクトルに掛け、データポイントの位置を再計算する。

Step3: 再計算されたデータポイントを用いて、新しい表面形状を得る

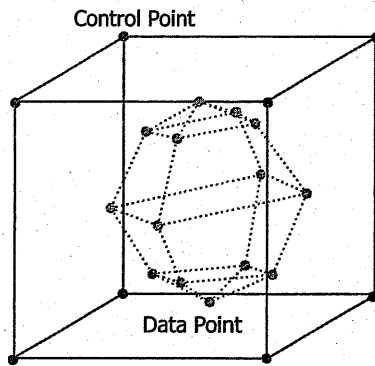
FFDによって、各ポリゴンパッチ接続部分での連続性が保たれ、滑らかな表面形状の維持が可能である。しかしこのようなコントロールポイントを介しての形状変形には以下のような欠点がある。

1. 精密な形状を得ることは困難である。
2. オブジェクト上のデータポイントの正確な位置を決定することは困難である。
3. コントロールポイントとデータポイントを対応付けるメカニズムを理解していないユーザーではコントロールポイントの位置変更がどのように形状変形に影響を及ぼすかを想像できない。
4. コントロールポイントはオブジェクトから離れて位置しているため、オブジェクト自身によって隠れているときや表示領域の外側に存在するときは、操作することが難しくなる。

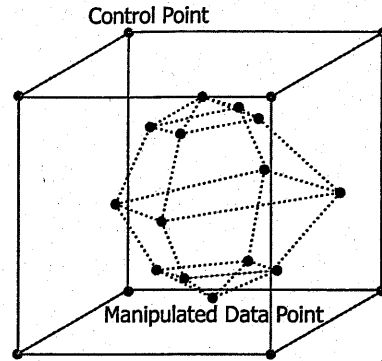
このような欠点から、コントロールポイントによる形状表面の制御は熟練を必要とする。そのため直感的なインタフェースを構築するために、オブジェクトの変形手法としてFFDを採用することは好ましくない。そこで、本研究ではHsuらが提案した形状表面を直接操作するFFDによる滑らかな形状変形の手法“direct FFD” [47]を採用する。

directFFDは、最初にユーザーがオブジェクト表面のデータポイントの位置を直接操作によって変更し、その操作によって得られた変更後のデータポイント群を生成するような、コントロールポイント位置を推定する手法である。本研究で実装したdirect FFDの手順を以下に述べる。

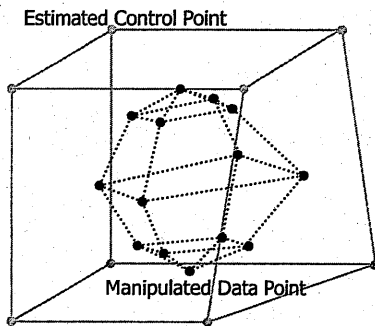
Step1: データポイントの位置ベクトル d をユーザーの手動作によって d_{new} に変化させる (図 6.2(b)).



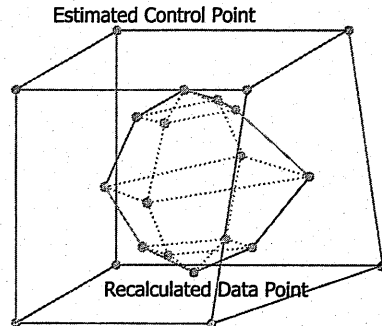
(a) 初期状態



(b) 手動作によるデータポイントの位置変更



(c) コントロールポイントの再計算



(d) 新しい表面形状の再計算

図 6.2 direct FFD の過程

Step2: 新しいデータポイントの位置ベクトル d_{new} を生成することが出来るコントロールポイントの位置ベクトル p_{new} を式(6.1)により推定する(図 6.2(c)).

$$p_{new} = A^+ d_{new} \quad (6.1)$$

ここで、 A^+ は、式(6.2)により最小二乗法を用いて計算される Catmull-Clark サブディビジョンによる重み行列 A の擬似逆行列である。

$$A^+ = A^T (AA^T)^{-1} \quad (6.2)$$

Step3: 推定されたコントロールポイントの位置ベクトル p_{new} を用いて、オブジェクトの表面形状を与えるデータポイントの位置ベクトル d_{mod} を式(6.3)

によって再計算する (図 6.2(d)).

$$d_{mod} = Ap_{new} \quad (6.3)$$

ユーザーのオペレーションによって得られた d_{new} に比べて, 式 (4.3) で得られた d_{mod} は操作したデータポイントの周辺の変形まで考慮に入れるため, より滑らかなオブジェクトの表面形状を与えることが出来る.

6.3.3 提案したモデル化の評価

提案したサブディビジョンサーフェスによるバーチャルクレイのモデル化について, 変形の滑らかさについての評価を行った. バーチャルクレイモデルを実装し, それぞれの細分割レベルについて, 仮想空間中の球状剛体による操作物体によって変形を施した. 変形シミュレータでは, 操作ツールである球状剛体の位置と大きさをスライダを用いて制御する. 時間に関しては, シミュレータのスライダのサンプリング間隔を実時間における剛体運動のサンプリング間隔と考える. 現時刻 (t) の操作ツールの位置 Pos_t と 1 時刻前 ($t-1$) の操作ツールの位置 Pos_{t-1} が成す円筒形内にあるデータポイントを, 現時刻のツールの球表面に平行移動させることにより, データポイントのユーザによる変形を実現する.

以下に評価に用いた 3 つの初期状態とそれぞれについて 1~3 回細分割した結果のサブディビジョンサーフェスと, 変形ツールによって変形されたサブディビジョンサーフェスの様子を示す.

1. 図 6.3 は, 初期状態を単純な正 6 面体とし, その初期状態を細分割したサブディビジョンサーフェスである. この初期状態から得られるサブディビジョンサーフェスは, 細分割の回数が増えるに従って球形状に近づいていくのがわかる.
2. 図 6.4 は, 1 の正 6 面体の各面を 4 分割した初期状態から生成したサブディビジョンサーフェスである.
3. 図 6.5 は, 1 の正 6 面体の各面を 16 分割した初期状態から生成したサブディビジョンサーフェスである.

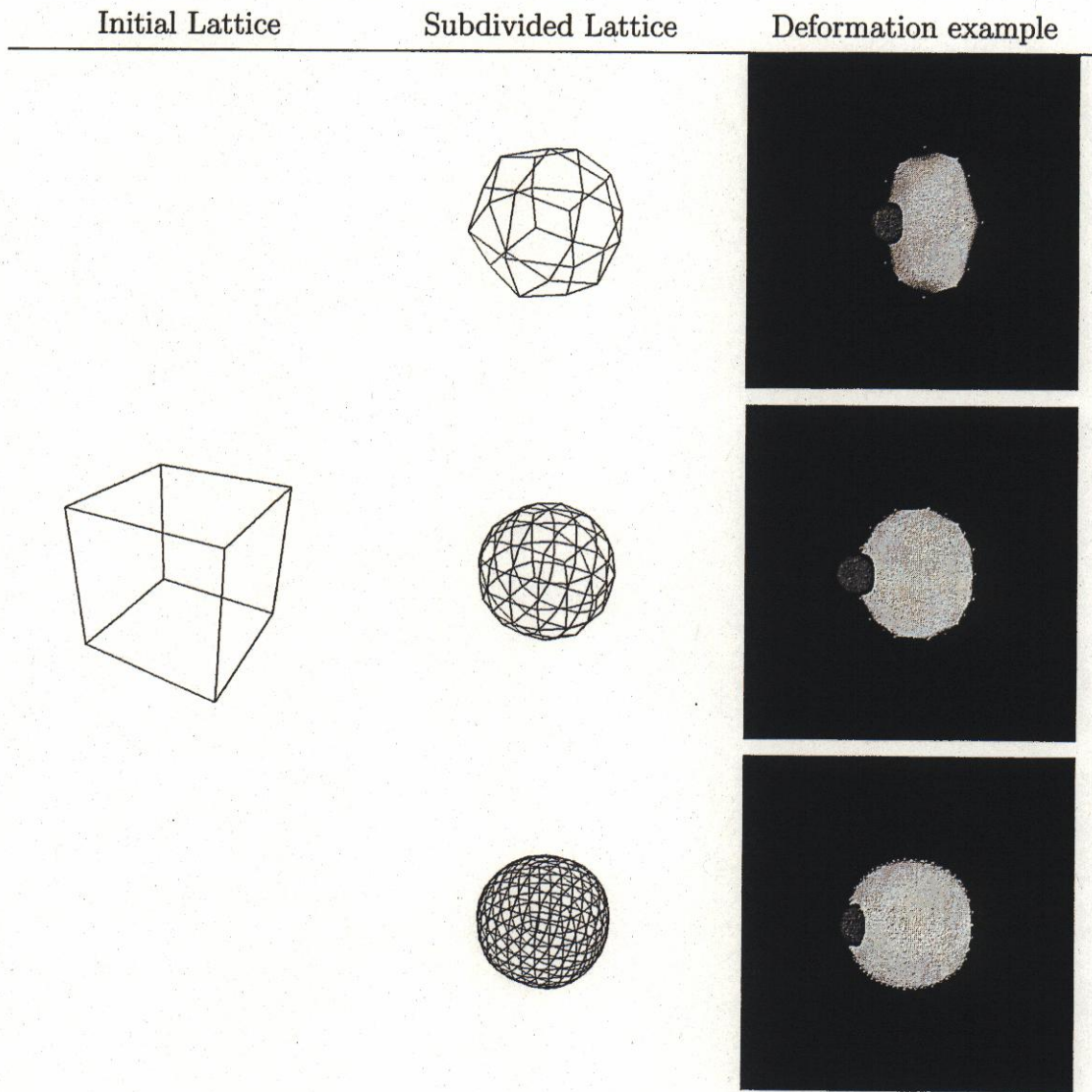


図 6.3 正六面体に対するサブディビジョンサーフェス

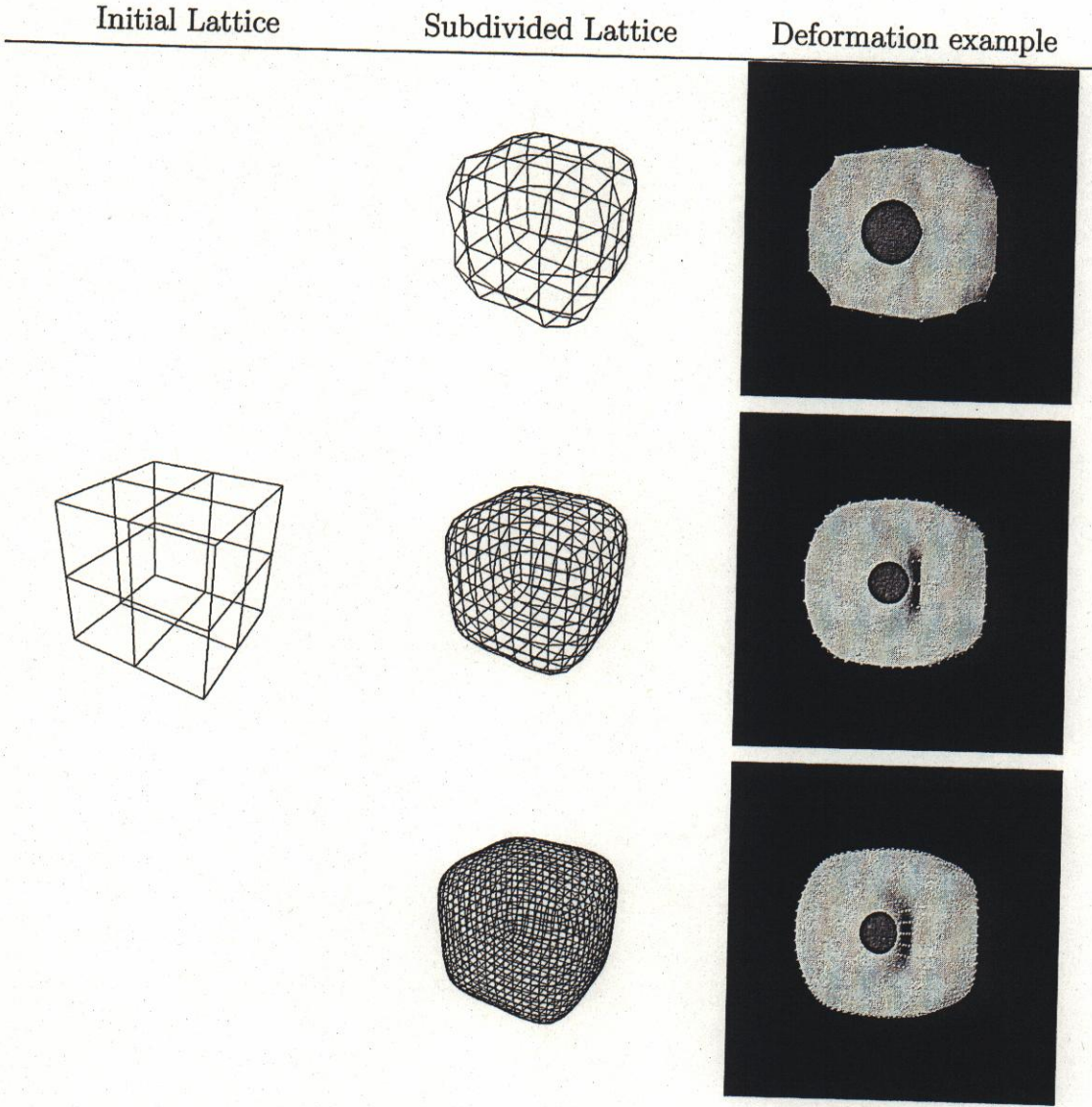


図 6.4 正六面体の各面を 4 分割した初期格子に対するサブディビジョンサーフェス

Initial Lattice

Subdivided Lattice

Deformation example

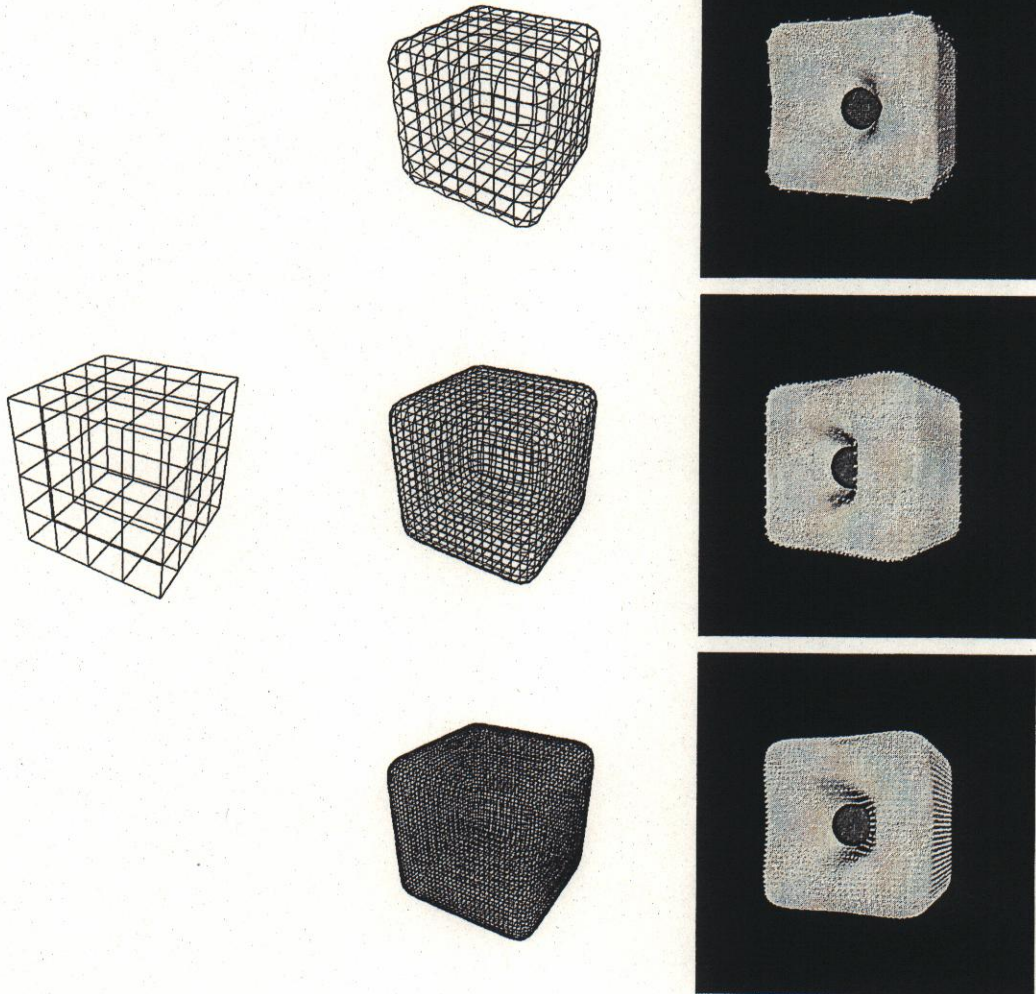


図 6.5 正六面体の各面を 16 分割した初期格子に対するサブディビジョンサーフェス

図 6.3 に示す初期格子が単純な正 6 面体である場合の変形例では、いずれの細分割回数の場合も操作ツールとバーチャルクレイの干渉している部分の場所に関係なくバーチャルクレイの形状全体が変形する。一方、図 6.4, 図 6.5 のように、初期格子の分割数が大きくなればなるほど、操作ツールと干渉しているバーチャルクレイの近傍だけの変形が得られる。これは、バーチャルクレイの形状表面を構成しているデータポイントは近傍のコントロールポイントに重みをかけて作成されているため、サブディビジョンサーフェスが表現できる形状の自由度はコントロールポイントによって決定されるからである。

これらの評価結果は、粘土の変形をよりの確にモデリングするためには細分割の回数を増やすことより、初期格子の分割数を大きくしておいたほうが良いことを示している。これらの議論より、本研究ではバーチャルクレイモデルとして初期格子を単純な正 6 面体の表面を 8 分割したものを用い、その初期格子に対して 2 回の細分割を実行して得られるサブディビジョンサーフェスを用いる。

6.4. 実空間の手動作によるバーチャルクレイ変形

6.4.1 手の動作可能領域の検証

構築した実カメラシステムにおける、手の動作可能領域は図 6.6 のようになる。図 6.6 において、オクトリーで表現されている部分が手のボクセルモデルを作成できる領域となる。この領域外ではボクセルを生成することが出来ず、手のボクセルモデルは一部分を欠いたモデルとなり、正確な推定の為の観測データとなることが出来ない。この理由は、ボクセルモデルを生成する際にカメラに写っている領域には手が存在し、写っていない部分には手が存在しないという前提を用いているため、4つのカメラすべてに手が写る場合のみ欠落の無いボクセルモデルの生成が可能であることにある。

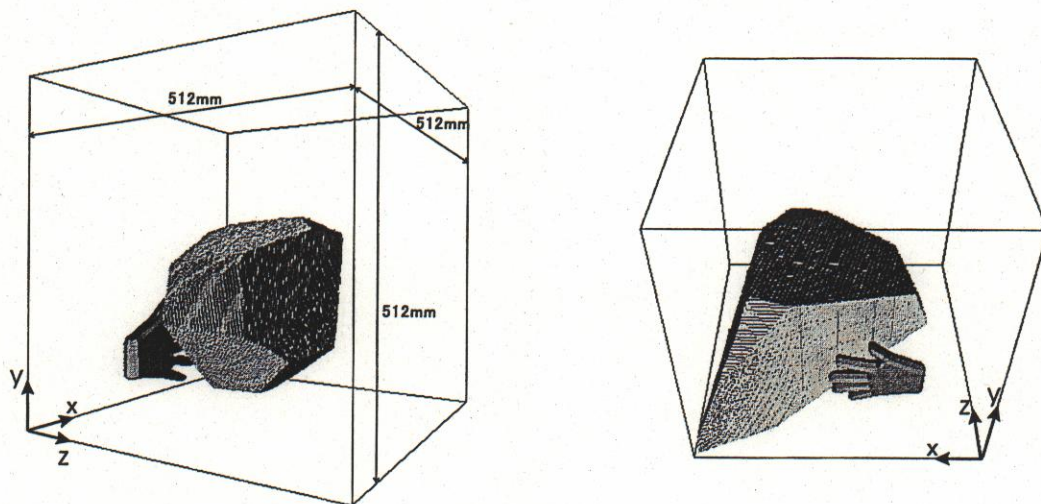


図 6.6 実カメラシステムにおける手の動作可能領域

6.4.2 バーチャルろくろシステムプロトタイプ

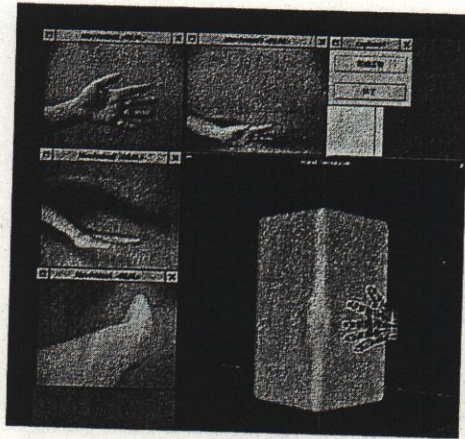
6.4.1 節で示したとおり、現状の実カメラシステムにおける、手の可動領域は非常に小さい。そのため、操作対象のバーチャルクレイの一部分のみしか変形することが出来ず、全体形状の変形は不可能である。手の可動領域が小さくても、バーチャルクレイの全体形状を変形できるアプリケーションとして、“ろくろによる形状変形”が考えられる。これは、“バーチャルろくろシステム”のプロトタイプでもある。

モデリングした仮想粘土を一定の速度で y 軸回りに時計方向に回転させ、回転ごとに各時刻における推定手形状との接触判定を行い、バーチャルクレイのデータポイントの位置を変更させる。

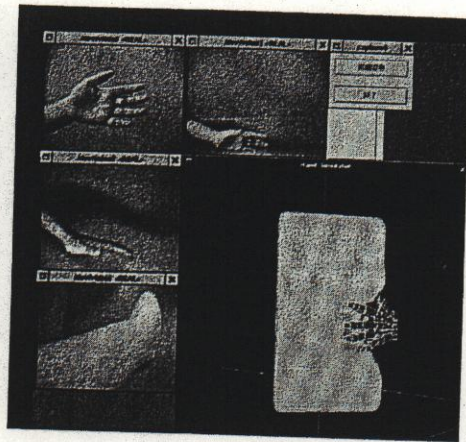
バーチャルろくろシステムのプロトタイプによる変形例を図 6.7 に示す。

200[msec] 毎に 2[deg] の速さでバーチャルクレイを回転させた場合、手形状の推定と接触判定をバーチャルクレイの回転が約 300[msec] 必要となるため、90[deg] の回転に伴うバーチャルクレイの変形は約 25[sec] 必要であった。

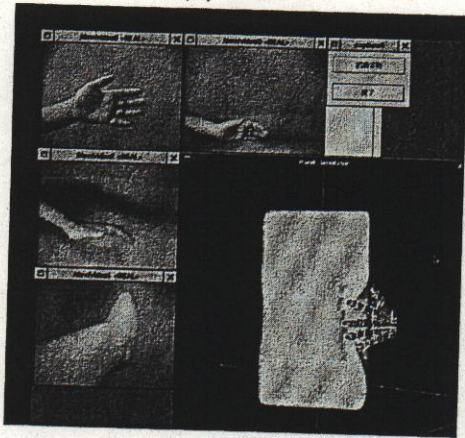
6.4.1 節で示した、現状の実カメラシステムにおける手の可動領域は、カメラの台数を増やし、「得られた多視点画像における手領域以外の領域のボクセル空間に



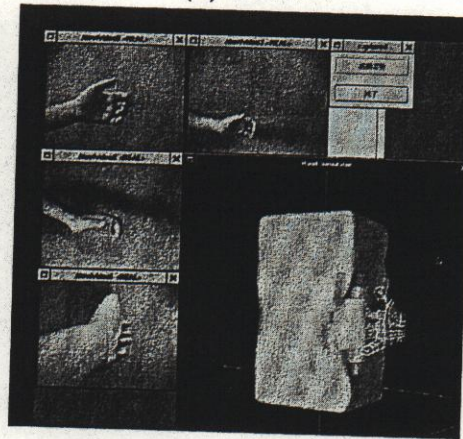
(1) 0.0sec



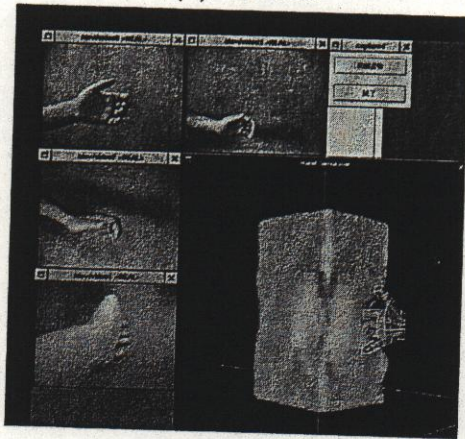
(2) 5.0sec



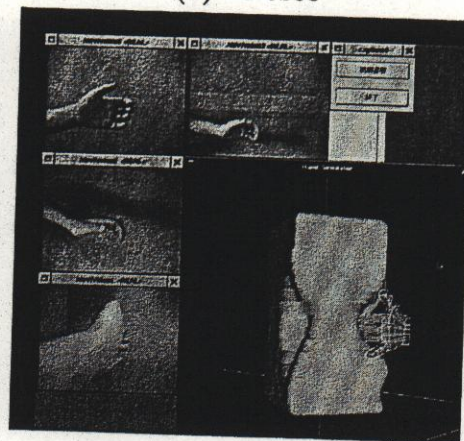
(3) 10.0sec



(4) 15.0sec



(5) 20.0sec



(6) 25.0sec

図 6.7 ろくろによるバーチャルクレイの変形

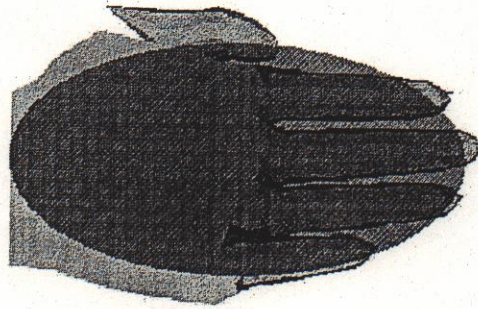


図 6.8 手領域を厚みのないへらとして取り扱う

は手は存在しない、しかし得られた画像領域外のボクセル空間は不定である」としボクセル空間を削ることで、拡大することが可能であると考えられる。この考え方を実装すれば、操作対象のバーチャルクレイを動かすことなく通常の粘土細工をシミュレーションすることが出来る。ワークスペースの拡大を実装できた場合、バーチャルクレイの全体形状を変形できるアプリケーションとして、以下の2種類のアプリケーションが実現できる。

- 手の位置と姿勢を用いたバーチャルクレイの変形
- 指形状を用いたバーチャルクレイの変形

次節以降の2つの節で、これら2種類のアプリケーションの詳細について述べる。

6.4.3 手の位置と姿勢を用いたバーチャルクレイモデリングインタフェース

すべての指をそろえてまっすぐに伸ばしたままで変わらない(実画像入力による初期姿勢の推定時の手形状と同じ)という条件で提案手法を応用し手の位置と姿勢のみの推定を行い、図6.8のように手を覆う楕円状のへらがあると考え、その上で推定した手動作を「へら」の動作として取り扱うことにより、バーチャルクレイモデリングインタフェースのプロトタイプとして実装した。

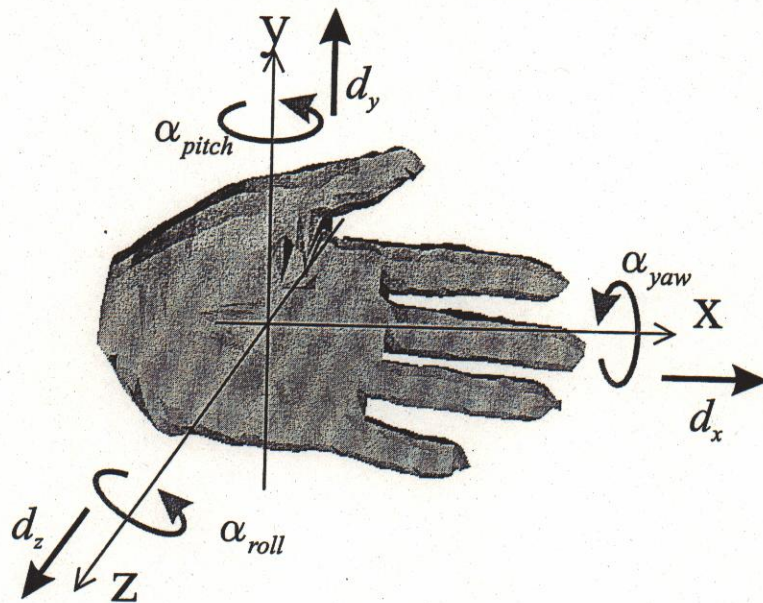


図 6.9 推定に用いる座標系

手の位置・姿勢推定アルゴリズム

3次元モデルフィッティングによる、手の位置・姿勢推定アルゴリズムについて以下に説明する。図 6.9 に示すように、フィッティング過程における座標系の原点は骨格モデルの重心位置であり、各軸の方向はワールド座標系のそれに等しい。

手の位置・姿勢推定処理の流れを、図 6.10 に示し、以下に各処理について説明する。

まず最初に骨格モデルの重心がボクセルモデルの重心に重なるように、骨格モデルを平行移動させる。この状態で、骨格モデルの各パッチのうち、ボクセルモデルの外側に位置しているパッチを“o-patch”と定義する。すべての o-patch に対して、図 6.11 中の f_1, f_2, f_3 のように、手の内側向きでそれぞれのパッチに垂直な力ベクトルを発生させる。次に、この力ベクトルとパッチから重心への距離ベクトル (図 6.11 の r_1, r_2, r_3) を用いてモーメントを計算する。

手全体に対して、これらの力とモーメントの総和を計算し、各軸方向の加速度

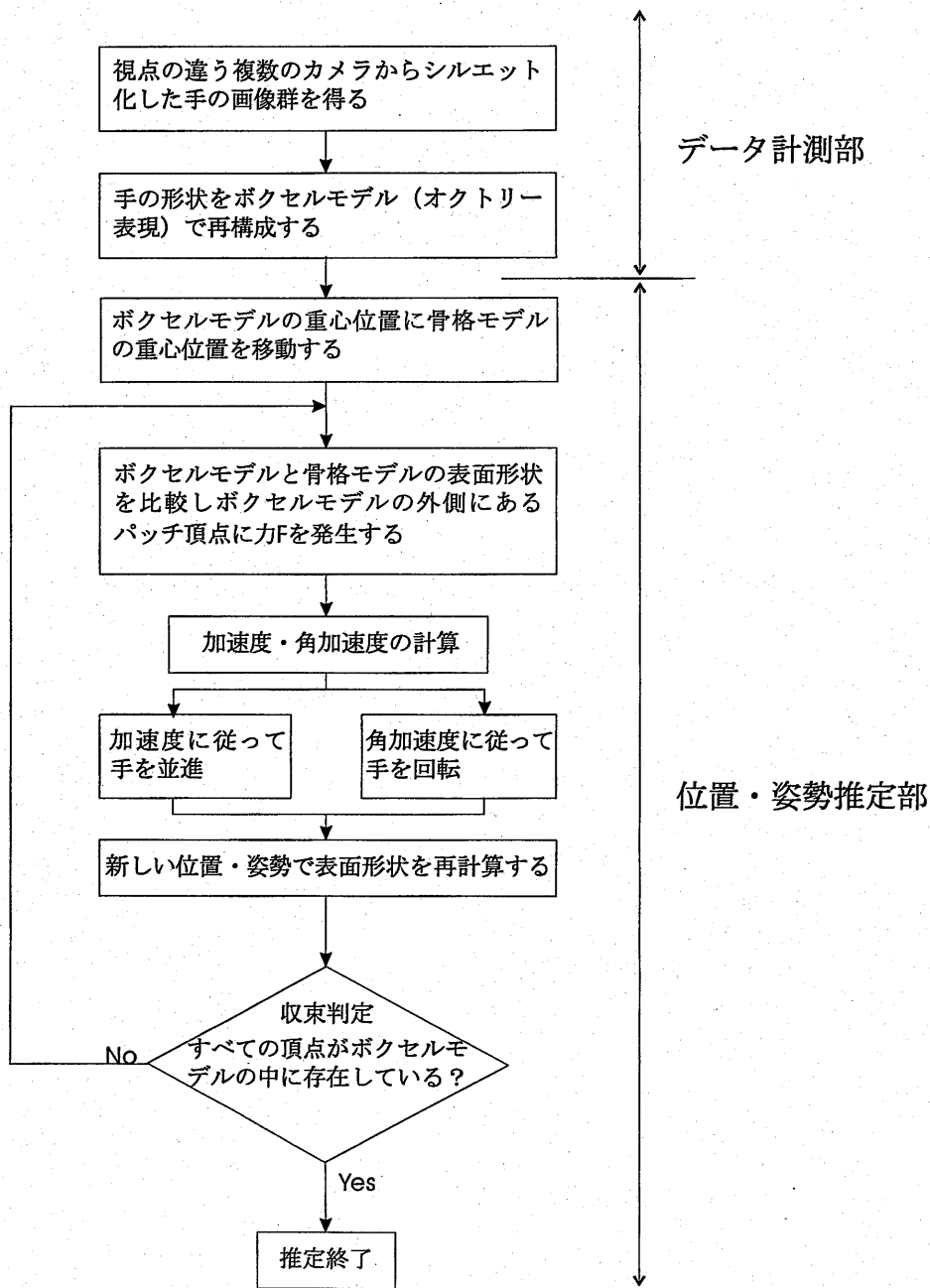


図 6.10 位置・姿勢の処理の流れ

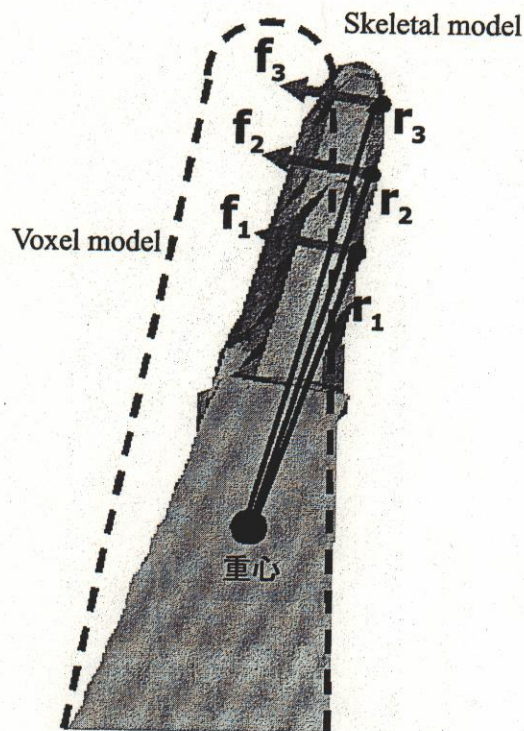


図 6.11 α -patch に発生する力とモーメント

と各軸回りの角加速度を式 (6.4), (6.5) によって計算する.

$$m\ddot{x} = F \quad (6.4)$$

$$I\ddot{\theta} = N \quad (6.5)$$

\ddot{x} は加速度, F は力の総和, m は骨格モデルの質量, $\ddot{\theta}$ は角加速度, N はモーメントの総和, I は慣性モーメント行列である.

手の並進・回転は \ddot{x} と $\ddot{\theta}$ の大きさによって同時に行われる. 図 6.9 において, α は各軸回りの回転角度, d は各軸方向の並進距離である. 微小量の並進・回転によって, 骨格モデルの位置と姿勢は変更される. この過程を繰り返す, 骨格モデルがボクセルモデルに完全に含まれたとき手の位置と姿勢が完全に推定できたとみなし, 処理を終える.

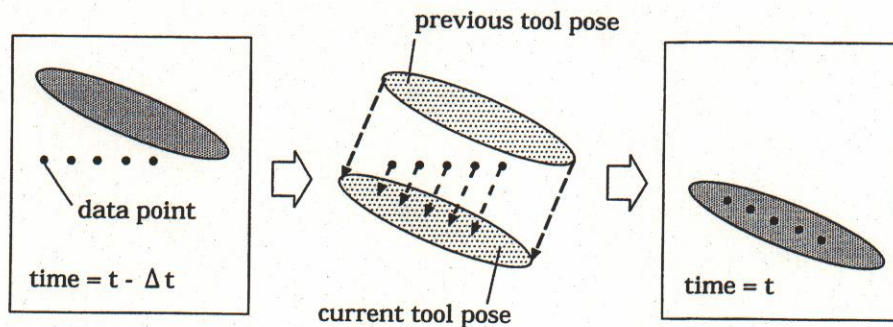


図 6.12 クレイとのインタラクション

手とバーチャルクレイとの干渉チェック

推定した手の位置・姿勢による，クレイ表面のデータポイントの位置変更方法について説明する．図 6.12 のように，前フレームの掌と現フレームの掌が成す領域に含まれるクレイモデルのデータポイントの位置を，現フレームのツール位置へ射影することによって位置を変更している．このように，本節で実装したバーチャルクレイモデリングインタフェースのプロトタイプでは，クレイを押すという動作のみを行う．

手の位置・姿勢によるバーチャルクレイ変形結果

図 6.13 は手の上下動作によるバーチャルクレイの連続的な形状変形の例を示している．左上ウィンドウの画像 (正面方向カメラからの画像) の手の位置の上下に伴い，右下のウィンドウ (仮想空間を表示) 内の手がバーチャルクレイの角を押し込み，滑らかに角がつぶれていく様子が表現されている．

現在のバーチャルクレイの変形の処理速度は，約 3[Hz] である．

6.4.4 指形状を用いたバーチャルクレイ変形

これまで提案してきた手形状推定により推定した指先の形状によってバーチャルクレイの変形を行う．変形は手指の表面形状とクレイモデルの正確な干渉を

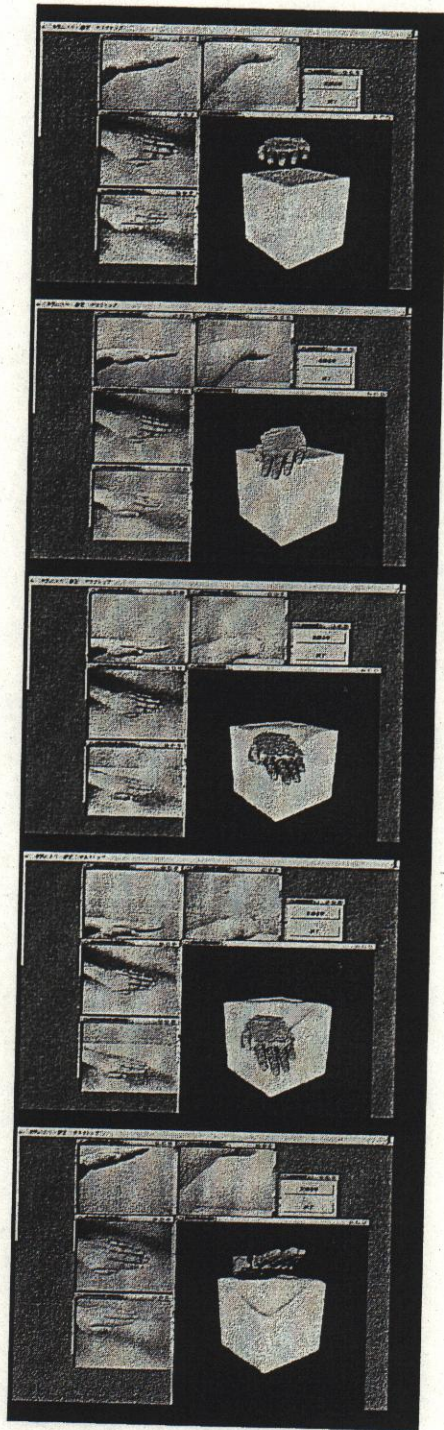
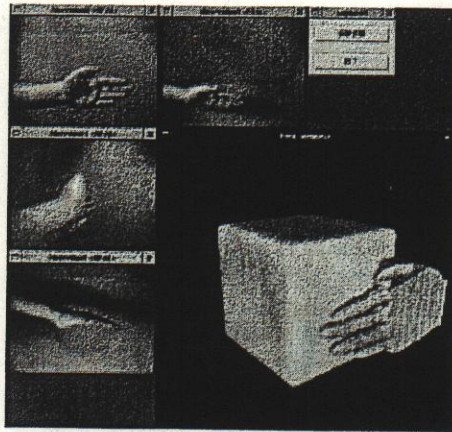
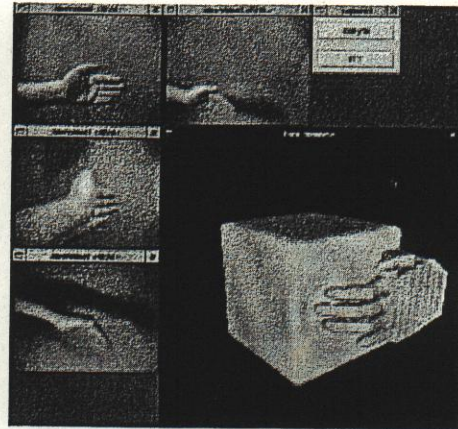


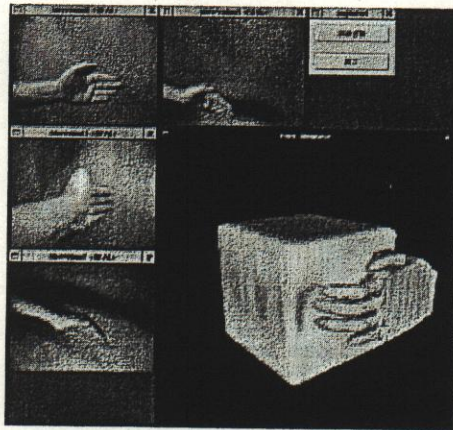
図 6.13 手の位置・姿勢によるバーチャルクレイの変形



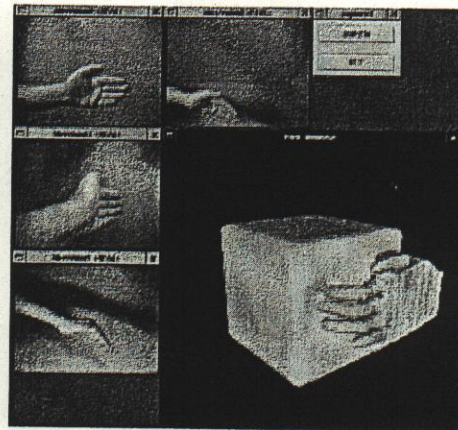
(1) 0.0sec



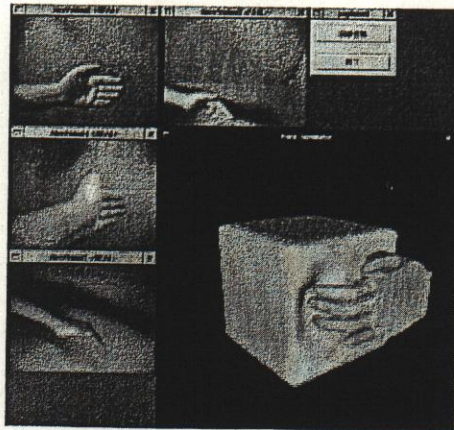
(2) 1.5sec



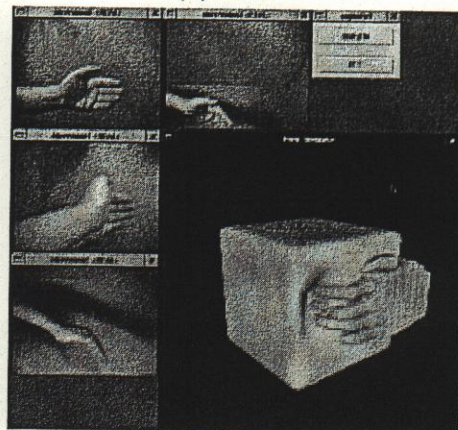
(3) 3.0sec



(4) 4.5sec

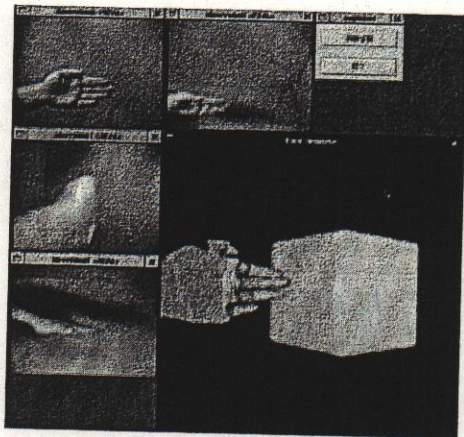


(5) 6.0sec

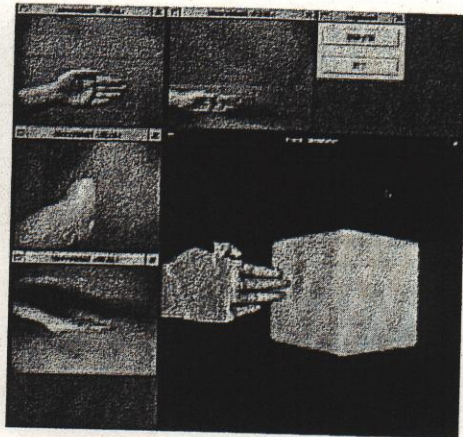


(6) 7.5sec

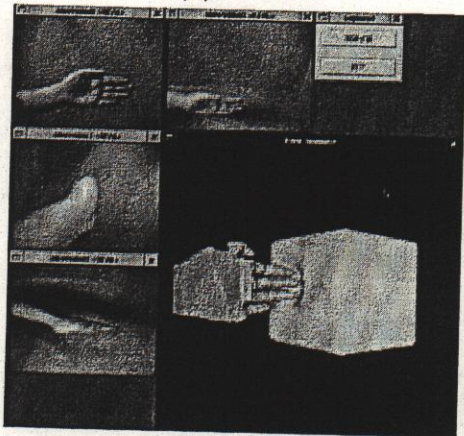
図 6.14 指先によるバーチャルクレイの変形 (1)



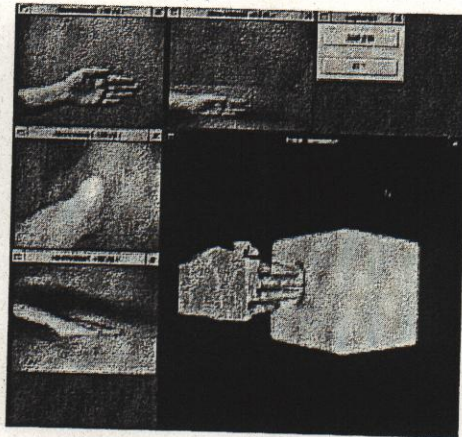
(1) 0.0sec



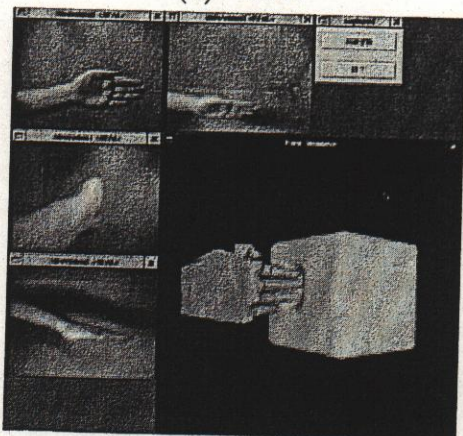
(2) 1.0sec



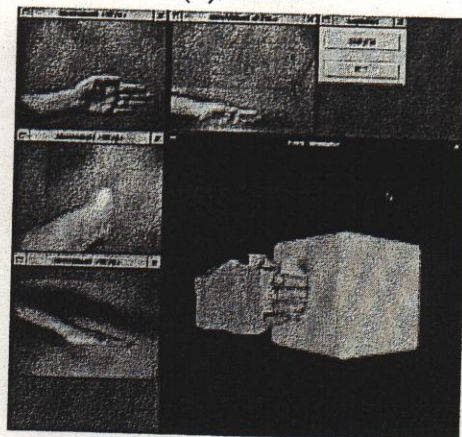
(3) 2.0sec



(4) 3.0sec



(5) 4.0sec



(6) 5.0sec

図 6.15 指先によるバーチャルクレイの変形 (2)

チェックするのではなく、推定した手形状の指先に球が、各関節間に関節間の長さを持つシリンダが存在すると考えて、6.4.3節で述べたようにバーチャルクレイのデータポイントの位置を変更することによって実行される。指先によるバーチャルクレイの変形例を図6.14、図6.15にそれぞれ示す。

6.5. 本章の結論

本章では、提案してきた手形状推定アルゴリズムをバーチャルクレイモデリングインタフェースの入力として用いることの有効性の検証を行った。

検証の為のアプリケーションとして、バーチャルクレイモデリングシステムを実装した。このシステムにおいては、バーチャルクレイは Catmull-Clark サブディビジョンサーフェスによって表現した。その理由は、評価システムでは体積一定変化よりもリアルタイム変形をより必要とするためソリッドモデルではなくサーフェスモデルで十分なためである。また、デザイナーが操作した部分の周辺を滑らかに変形する必要があることから、複数面の滑らかな接続を考慮する必要がなく、細分割を無限大回繰り返して得られる極限曲面が4階の B-spline に収束することが知られている Catmull-Clark サブディビジョンサーフェスを採用した。

続いて、手とバーチャルクレイの干渉チェックと、データポイントの位置変更方法について詳しく述べ、更にバーチャルクレイ全体を滑らかに変形させる direct FFD について詳細に述べた。

その上で、バーチャルクレイモデルを実装し、変形シミュレーションによってどの程度滑らかにその形状変形が出来るかを確認し、モデルの初期格子数と細分割回数を決定した。

最後に推定した手形状によるバーチャルクレイモデリングインタフェースのプロトタイプを実装し、その動作の検証を行った。

第7章 結論

7.1. まとめ

本論文では、3次元システムにおいて、従来の入力方法に比べてより自然な入力方法の構築のために、人間の操作的な手動作を直接入力として使用することを目的とした、多視点画像に基づく手形状推定の新しい手法の提案とその検証を行った。

産業界においては、工業製品設計に3次元CAD/CAMシステムや3次元モデリングシステムを用いることが当たり前になっている。このような3次元システムにおいては、そのシステムの持つ機能が重要であることは当然であるが、マンマシン・インタフェースの重要性も同様に高く、より使いやすいシステムへの様々なアプローチがこれまでに研究されてきた。しかしそれらの研究成果にも係わらず、3次元自由形状を自由に入力・変形などの操作をする作業は未だ困難で、ユーザーにとって大きな負担となっているのが現状であり、3次元システムを用いるユーザーの中には「入力インタフェースをどうにかして欲しい」という不満を持つものも依然少なくないのである。また、近年急速に発達してきたバーチャルリアリティにおいては、実空間と同様の操作方法で仮想空間の物体を操作出来るようなインタフェースが求められている。このような問題の解決のために、本研究に取り組み始めたのである。

これを解決する一つの方法として、「人間が物を作るときには手を使う」ことから、本研究は実空間での手動作を認識しインタフェースとして用いることに着目した。人間の手動作は、言葉、感情の表現、形状の表現などのコミュニケーション手段として、また道具や形状の操作手段として多くの機能を有している。この大きな表現力を持つ人間の手動作を、コンピュータと人間の自然なインタフェー

手段として応用することは、有効なアプローチである。また工業製品設計の初期段階における大まかな形状設計時、デザイナーはクレイモデルを用いる事が多い。このような自由に変形できる柔らかな塑性体（たとえば粘土）を仮想空間上でモデル化し、このモデルを人間の手動作によって変形させることが出来れば、デザイナーの自由な発想を阻害することなく、簡単に3次元モデリングシステムでの自由形状の入力や変形が出来るようになると思う。さらに製品設計等の実務環境では、このような3次元モデリングシステムを長時間にわたって使用することが多い。そのような使用環境でのインタフェースとしては、データグローブや力覚提示デバイスなどの接触型のインタフェースよりも、何も装着する必要のない非接触型のインタフェースのほうがユーザーの拘束感が少ないことから好ましいといえる。

そこで本研究ではバーチャルクレイモデリングインタフェースの構築に最終目標を置き、その枠組みの中に位置する本論文では、バーチャルクレイモデリングインタフェース構築の基本でありキーアイデアとなるビジョンベースリアルタイム手形状推定手法の提案をおこなった。

第2章では、まず最初に推定対象である人間の手の構造について、本論文で必要となる各部の名称と動作の説明を行った。更に、本研究で推定対象とする人間の操作的な手動作の際の手形状がどのように変化するかを画像を用いて解析した。

第3章では、本論文の最も基本となる多視点シルエット画像を用いた手形状推定のアルゴリズムの提案と検証を行った。あらかじめシステム内に構築した手のモデルと、多視点カメラシステムから得られたシルエット画像を統合して3次元化された手の観測データの間との3次元状での直接的なフィッティングを基本とした、手形状推定アルゴリズムを提案した。提案した手法は、従来モデルマッチング手法として一般的に行われる、モデルの形状変化を2次元画像に反映させ評価を行う手法に比べて、3次元的に行うことによりより直接的にモデルの形状変化を取り扱うことが出来る手法である。この提案手法の有効性を示すため、シミュレーションを行った。手形状シミュレータにより変形度合いの違う4種類の手形状を生成し、推定経過をチェックした。シミュレーションの結果、前提としている関節角度15[deg]以下の小さな形状変化だけでなく、50[deg]近い形状変化

でも推定できることが確認できた。その上で、提案した手形状推定アルゴリズムを、実際の手の動作を多視点カメラでキャプチャした複数の動画像に対して実装し、実環境における提案アルゴリズムの検証を行った。手動作を観測するために4台の CCD カメラを用いた多視点カメラシステムを構築し、実カメラ環境で必要となる初期姿勢の推定について、手の形状の知識を利用してアドホックに行う方法について述べた。続いて、提案アルゴリズムによる複数の動画像を入力とする手形状推定を実装し、その動作を検証した。検証の結果、提案した推定手法では実画像を入力とした場合もシミュレーションと同様に期待通りの動作をすることが示された。

第4章では、提案手法の処理速度の向上のために最初に粗く関節角度の推定を行い探索範囲を狭めてから、細かく関節角度推定を行う粗密探索法の導入を行い、効果の検証を行った。最初に、粗密探索法を導入するために必要な、2種類の解像度を持つ手の表面形状データの表現に Catmull-Clark サブディビジョンサーフェスを導入した。サブディビジョンサーフェスの導入によって、手の表面形状データを荒さの違いによって複数個持つことが出来るようになった。既存の3次元モデリングシステムを用いて作成した粗い手の表面形状データは120個のポリゴンから成っているが、細分割を1回行った細かい手の表面形状データは480個のポリゴンから成る。これは必要に応じて更に細分割を行い1920個のポリゴンから成る表面形状データとして持つことも可能である。また、サブディビジョンサーフェスを導入することで、骨格姿勢の変化に滑らかに追従できる手の表面形状を得ることが出来た。更に、自然な手動作における、各指の中での運動拘束を導入した。この拘束条件によって、手の自由度は31から27に減少し、計算量を減らすことが可能になった。これらのモデルの改良を行った上で、基本手形状推定手法に粗密探索法を導入した。粗密探索法の導入により追加された処理は、粗推定であり、初期格子の内側に内ボクセルモデルが存在するように、内ボクセルモデルの内側に存在する初期格子に対して外向きの力を発生させることによって、初期格子を内ボクセルモデルの外側に位置するような関節角度を推定する処理である。実画像入力による連続手形状推定では、粗推定のみを用いて手形状推定を行い、実画像への追従性を確認することが出来た。

第5章では、提案した多視点シルエット画像を用いた手形状推定手法の評価を行った。最初にボクセルモデルの復元レベルの違いにより、推定精度と処理時間にどのような差が見られるかを検証した。その結果、手形状推定をどのような目的で応用するかにより、精度優先か速度優先かが異なり、復元レベルの設定も考慮する必要があることが示された。次に、カメラの数の違いによる推定精度の比較を行った。カメラ配置を5種類の正多面体の頂点にあると設定し、それぞれにおける推定精度の評価を行った。推定した4種類の手形状においてはカメラ台数が12台を越えると、推定誤差が小さくなり安定すると言う結果が得られた。しかしそれ以上に、カメラの位置が推定精度に大きな影響を及ぼすことも示された。その上で、実カメラシステムのハードウェアの制約を考慮した4台カメラの、適切な配置を評価した。評価の結果からは、現状の実カメラシステムの配置が適切であることが示されたが、同時に、どのような手形状に対しても最適な推定結果を与えるカメラ配置を決定することは不可能であり、複数のカメラからオクルージョンが少ないカメラを選択しながら推定を行う必要性を示した。最後に、粗密探索法の導入による推定精度の向上の効果について、評価を行った。シミュレータを用いて評価を行った結果、推定精度の向上が確認できたが、処理速度の向上は図られなかったが、粗推定のみでも改良前に比べて推定精度は良く、処理時間も約1/4になっていることが確認できた。

第6章では、提案した手形状推定アルゴリズムをバーチャルクレイモデリングインタフェースの入力として用いることの有効性の検証を行った。検証の為のアプリケーションとして、バーチャルクレイモデリングシステムのプロトタイプを実装した。このシステムにおいては、バーチャルクレイはCatmull-Clarkサブディビジョンサーフェスによって表現した。その理由は、評価システムでは体積一定変化よりもリアルタイム変形を必要とするためソリッドモデルではなくサーフェスモデルで充分であると判断したためである。また、デザイナーが操作した部分の周辺を滑らかに変形する必要があることから、複数面の滑らかな接続を考慮する必要がなく、細分割を無限大回繰り返して得られる極限曲面が4階のB-splineに収束することが知られているCatmull-Clarkサブディビジョンサーフェスの採用を決定した。バーチャルクレイの変形は(1)手とバーチャルクレイの干渉チェッ

ク、(2) クレイのデータポイントの変更、(3) directFFD による表面形状全体の再計算、の3つのステップで行った。手とバーチャルクレイの干渉チェックは、一つ前のフレームで推定された手の位置・形状と方法と現フレームで推定された手の位置・形状を用い、それらが成す体積空間とクレイの位置関係によって行った。体積空間内に存在するクレイのデータポイントが位置変更の対象となり、データポイントを現フレームの手形状の上に投影することで、位置の変更を実現した。その後、direct FFDによってバーチャルクレイ全体を滑らかに変形させる方法をとった。変形シミュレーションによってどの程度滑らかにその形状変形が出来るかを確認し、モデルの初期格子数と細分割回数を決定した。最後に推定した手形状によるバーチャルクレイモデリングインタフェースのプロトタイプを3種類(1)バーチャルクレイを回転させることによるろくろシミュレーションシステム、(2)手の位置と姿勢のみを用いたバーチャルクレイの変形、(3)推定した手形状の指部分を用いたバーチャルクレイの変形)を実装し、その動作の検証を行った。

本論文全体を通しての実験と検証結果から、提案した多視点画像に基づく手形状推定手法はシミュレーションにおいてはそのアルゴリズムの有効性・正当性を示すことが出来た。しかし実画像入力実験の結果は、実装したハードウェア環境においては、その処理速度を少なくともツールを用いた粘土変形の処理時間までに向上させる必要があることを示すものとなった。

7.2. 課題と今後の展開

提案手法の高速化が今後の最重要課題であることは、本論文を通して明らかになったことである。提案手法はこの先以下に示す2つの方向を目指すことで、ビジョンベース手形状推定の新しい手法としての効果を明確に与えることが出来ると筆者は考えている。

1. PC クラスタや基本のソフトウェアのハード化などによる実装ハードウェアの最大限の高速化によりリアルタイム性と高精度化の両方を実現し、完全な任意の手形状推定を目標とする。
2. 通常のハードウェア環境で3次元モデリングシステムの入力インタフェース

としての手形状の曖昧性を許容することによりリアルタイムに近い処理時間での手形状推定を一般的なコストで実現し、市販の3次元システム（例えば3次元モデリングシステム）のインタフェースとなることを目標とする。

また、本研究の最終目的は、工業製品設計の実務環境で使用している3次元CAD/CAMシステムや、3次元モデリングシステムにおける、3次元形状操作インタフェースの一つとしてのバーチャルクレイモデリングインタフェースの構築である。この観点から言えば、本論文で提案し検証したことは、このバーチャルクレイモデリングインタフェースの入力手法の確立に過ぎない。今後は最終目標である「実務環境で使用可能な3次元設計システムの一つとなりうるバーチャルクレイモデリングインタフェースの構築」を目指して以下のような様々な試みを行っていく予定である。

- CAD空間上での手とクレイモデルの干渉によるクレイモデルの変形手法の構築
- 両手や複数人での協調作業への拡張
- カメラシステムの最適化
- 高速なハードウェア化
- 市販CADシステムとのハンドリング

最後に、このシステムが設計の現場において、多くの設計者やデザイナーのイメージーションをダイレクトにコンピュータに伝えることの出来るインタフェースとなり、産業界に貢献することが、私の喜びでもあり誇りとも成りうるであろう。

謝辞

本論文に書き記した一連の研究は、筆者が奈良先端科学技術大学院大学 情報科学研究科在学中に、ロボティクス講座 (小笠原研究室) において執り行われたものです。この研究は多くの方々のお力添えにより実現することができました。ここでお礼を申し上げたいと思います。

本学 情報科学研究科 小笠原 司 教授は、エンジニアとしての社会経験を経て再度研究者としての道を目指し始めた筆者を博士前期課程の学生として快く受け入れて下さいました。そして、筆者のアイデアを最大限尊重して頂き、必要な研究環境を滞りなく整えて頂いたため、研究をスムーズに行うことが出来ました。今までの自分の経験からの思い入れと直感的な着想から研究を立ち上げ、何とか本論文を執筆することができたのも、ひとえに小笠原教授の忍耐強いお力添えがあったからだと思います。ありがとうございました。

本学 情報科学研究科 横矢 直和 教授には、博士前期課程、後期課程を通じてこの研究テーマに対してご教示を頂きました。また、学外の研究会発表の場においても、多くの御質問をしていただきました。バーチャルクレイモデリングインタフェースは横矢教授の研究室において既に研究がされており、筆者の手法での問題点などを的確に指摘していただきました。先生から頂いたコメントに対して常に答えを出し続けていくことを意識して研究を進めることで、本研究の有効性を強く裏付けることが出来ました。ありがとうございました。

本学 情報科学研究科 木戸出 正繼 教授には、本研究をまとめるにあたり、数々のご助言、ご指摘を頂きました。また、学外において産学共同研究開発テーマの立案にも参加させていただきました。その経験は、本研究の将来展開の具体化・実現化の大きな道標となりました。ありがとうございました。

本学 情報科学研究科 松本 吉央 助教授は、「ビジョンを用いて手形状を認識し3次元形状を操作するインタフェースを作りたい」との目標だけを持っていた筆者に対して、具体的な実現方法のアイデアを示してくださいました。松本助教授のご助言があったからこそ、この研究が、大きな目標に向かってのスタートを切ることが出来たのだと思っております。博士課程の4年半の間、研究に対して常にご相談に乗っていただき、強力なサポートをしていただき、心強く研究を進めることが出来ました。また、英語論文執筆の際には何度も何度も添削をしていただきました、そのうえ学会発表前にも懇切丁寧なご指導、ご助言を頂きました。ありがとうございました。

本学 情報科学研究科 上田 淳 助手には、研究会で様々な質問を頂きました。また制御の知識がほとんどなかった筆者に対して、分かりやすく丁寧に制御の観点から、推定手法に対する問題点や改良点などのご指摘を頂きました。特にフィッティングの際の力の発生方法の曖昧な表現を修正することが出来たのも上田助手のおかげだと思います。ありがとうございました。

鳥取環境大学 今井 正和 教授 には、筆者が博士前期課程の在学中には直接研究室でご指導を頂き、研究会などで多くの鋭いご質問を頂きました。本研究のキーとなる手形状推定手法の確立において論理的裏付けを明確にすることができたのも、先生の質問によるところが大きかったと思います。また、御異動されてからもお会いするたびに励ましのお言葉を頂きました。ありがとうございました。

和歌山大学 中村 恭之 助教授 には、本研究を遂行するに当たって様々な知識や手法を御教授いただきました。これらの知識は今後研究を進めていく上でなくてはならないものであると感じております。ありがとうございました。

本学 情報科学研究科 安室 喜弘 助手には、本研究を立ち上げるにあたり手形状シミュレータや手形状データなどのデータを快く提供していただきました。ソフトウェアやデータに関する細かな質問に対してもお忙しい中ご回答やご助言を下さいました。これらがあったからこそ、スムーズに研究を立ち上げる事が出来ました。ありがとうございました。

本学 情報科学研究科 児玉 浩卓 君 (現 シャープ) には、バーチャルクレイモデルの評価のためにインバースキネマティクスを用いた手形状推定手法の実装を

行ってもらいました。ありがとうございました。

本学 情報科学研究科 竹村 裕 君 には、博士前期課程のときから同期生として長い間様々な点で助けてもらいました。研究生活においても良き相談相手となってくれました、本当に心強かったです。ありがとうございました。

本学 情報科学研究科 怡土 順一 君、小枝 正直 君 には、毎日の研究遂行にあたって多くの場面で力を貸してもらいました。特に、PCのシステム構築の知識がほとんどない筆者にとって、システム設定の場面で毎日のように助けてもらいました。また、研究グループにおいても大きな力となってくれました。ありがとうございました。

また、筆者が在学中、研究生活を様々な面で支えて頂いた本学 情報科学研究科 ロボティクス講座の卒業生、在学生の皆様には感謝致します。また、研究室の運営にこまやかな心配りをしてくださり、また励ましの言葉をかけてくださった歴代秘書の金岡 恵さん、田中 舞さん、中谷 愛子さんに心よりありがとうの言葉を贈ります。

最後に、私の研究活動を理解し温かく見守り続けてくれた夫・哲之と二人の子どもたち・碧と周、そして私をずっと支え続けてくれた母親と今は亡き父親に深く感謝し、論文の結びとしたいと思います。



著者研究業績

論文誌

1. Etsuko Ueda, Yoshio Matsumoto, Masakazu Imai and Tsukasa Ogasawara.
“Hand Pose Estimation for Vision-based Human Interface”, IEEE Transactions on Industrial Electronics. Vol.50, No.4, pp.676–684, August, 2003

国際会議発表

1. Etsuko Ueda, Hiroataka Kodama, Yoshio Matsumoto and Tsukasa Ogasawara.
“Non-contact Virtual Clay Modeling Interface using Multi-viewpoint Images”, In Proceedings of IAPR Workshop on Machine Vision Applications (MVA2002), pp.56–59, 2002.
2. Etsuko Ueda, Yoshio Matsumoto, Masakazu Imai and Tsukasa Ogasawara,
“Hand Pose Estimation Using Multi-Viewpoint Silhouette Images”, In Proceedings of 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2001), pp.1989–1996, 2001.
3. Etsuko Ueda, Yoshio Matsumoto, Masakazu Imai and Tsukasa Ogasawara,
“Hand Pose Estimation for Vision Based Human Interface”, In Proceedings of 10th IEEE Int. Workshop on Robot and Human Communication (RO-MAN 2001), pp. 473–478, 2001.

国内発表

1. 上田 悦子, 児玉 浩卓, 松本 吉央, 小笠原 司. “マーカーを用いた手形状入力による非接触仮想粘土細工システム”, ロボティクス・メカトロニクス講演会 2003 講演論文集, 2P2-2F-E1, 2003.5.
2. 上田 悦子, 児玉 浩卓, 松本 吉央, 小笠原 司. “多視点カメラシステムを用いたバーチャルクレイモデリングインタフェース”, 電子情報通信学会・パターン認識・メディア理解研究会 (PRMU) 2002.11.
3. 上田 悦子, 児玉 浩卓, 松本 吉央, 小笠原 司. “多視点画像を用いたバーチャルクレイモデリングインタフェース”, 第 20 回ロボット学会学術講演会予稿集, 2002.10.
4. 上田 悦子, 松本 吉央, 今井 正和, 小笠原 司. “多視点シルエット画像を用いた手の形状推定”, 情報処理学会研究報告, 2001-CVIM-128, pp.25-31, 2001.
5. 上田 悦子, 松本 吉央, 今井 正和, 小笠原 司. “多視点シルエット画像を用いた手の形状推定 -実カメラシステムを用いた実装-”, ロボティクス・メカトロニクス講演会 2001 講演論文集, 2P2-N9, 2001.6.
6. 上田 悦子, 松本 吉央, 今井 正和, 小笠原 司. “多視点シルエット画像を用いた手の形状推定”, 第 18 回ロボット学会学術講演会予稿集, pp.393-394, 2000.9.

参考文献

- [1] Vladimir I. Pavlovic, Rajeev Sharma, Thomas S Huang. "Visual Interpretation of Hand Gestures for Human-Computer Interaction : A Review". *IEEE Pattenrn Analysis and Machine Intelligence*, Vol. 19, No. 7, pp. 677-695, July 1997.
- [2] John Weng, Yuntao Cui. "Recognition of Hand Signs from Complex Backgrounds". In *Computer Vision for Human-Machine Interaction*. Cambridge University Press, 1998.
- [3] 今川 和幸, 呂 山, 猪木 誠二, 松尾 英明. "肌色領域により隠れて見える場合を考慮した手話動画像からの手の実時間追跡". 電子情報通信学会論文誌, Vol. J81-D-II, No. 8, pp. 1787-1795, August 1998.
- [4] Chris Talor, Andreas Lanitis, Tim Cootes, Gareth Edwards, Tariq Ahmad. "Model-Based Interpretation of Faces and Hand Gestures". In *Computer Vision for Human-Machine Interaction*. Cambridge University Press, 1998.
- [5] Masahiro Yachida, Yoshio Iwai. "Looking at Human Gestures". In *Computer Vision for Human-Machine Interaction*. Cambridge University Press, 1998.
- [6] 宮里 勉, 岸野 文郎. "臨場感通信会議における仮想空間を会した指差し指示動作知覚の評価". 電子情報通信学会論文誌, Vol. J80-D-II, No. 5, pp. 1221-1230, May 1997.
- [7] 石淵 耕一, 岩崎 圭介, 竹村 治雄, 岸野 文郎. "画像処理を用いた実時間手振り推定とヒューマンインタフェースへの応用". 電子情報通信学会論文誌,

Vol. J79-D-II, No. 7, pp. 1218–1229, July 1996.

- [8] 内海 章, 大谷 淳, 中津 良平. “多数カメラを用いた両手手振りの検出”. 情報処理学会論文誌, Vol. 40, No. 8, pp. 3143–3154, 1999.
- [9] Christoph Maggioni, Bernhard Kämmerer. “Gesture Computer — History, Design and Applications”. In *Computer Vision for Human-Machine Interaction*. Cambridge University Press, 1998.
- [10] 舟橋 健司, 安田 孝美, 横井 茂樹, 鳥脇 純一郎. “3次元仮想空間における仮想手による物体操作モデルと一実現法”. 電子情報通信学会論文誌, Vol. J81-D-II, No. 5, pp. 822–831, 1999.
- [11] Thomas Massie. “A Tangible Goal for 3D Modeling”. *IEEE Computer Graphics and Applications*, pp. 62–65, May/June 1995.
- [12] 北村 喜文. “仮想世界の物理法則”. 日本機械学会誌, Vol. 102, No. 971, 1999.
- [13] Ying Win, Thomas S. Huang. “Hand Modeling Analysis, and Recognition For Vision-Based Human Computer Interaction”. In *IEEE Signal Processing Magazine*. May 2001.
- [14] 岡 兼司, 陳 欣蕾, 中西 泰人, 佐藤 洋一, 小池 英樹. “拡張机型インタフェースのための複数指先の追跡とその応用”. 情報処理学会論文誌 コンピュータビジョンとイメージメディア, Vol. 44, No. SIG 5 (CVIM 6), pp. 74–84, April 2003.
- [15] 亀田 能成, 美濃 導彦, 池田 克夫. “シルエット画像からの関節物体の姿勢推定法”. 電子情報通信学会論文誌, Vol. J79-D-II, No. 1, pp. 26–35, 1996.
- [16] 島田 伸敬, 白井 良明, 久野 義徳. “確率に基づく探索と照合を用いた画像からの手指の3次元姿勢推定”. 電子情報通信学会論文誌, Vol. J79-D-II, No. 7, pp. 1210–1217, 1996.

- [17] 島田 伸敬, 白井 良明, 久野 義徳, 三浦 純. “緩やかな制約知識を利用した単眼視動画像からの関節物体の形状と姿勢の同時推定”. 電子情報通信学会論文誌, Vol. J81-D-II, No. 1, pp. 45-53, 1998.
- [18] N. Shimada, K. Kimura, Y. Shirai. “Real-time 3-D Hand Posture Estimation based on 2-D Appearance Retrieval Using Monocular Camera”. In *Proc. Int. WS. on RATFG-RTS (satellite WS of ICCV2001)*, pp. 23-30, 2001.
- [19] 岩井 儀雄, 八木 康史, 谷内田 正彦. “単眼動画像からの手の3次元運動と位置の推定”. 電子情報通信学会論文誌, Vol. J80-D-II, No. 1, pp. 44-55, 1997.
- [20] 遊佐 博幸, 兵庫 明, 関根 慶太郎. “基準抽出による2次元平面上における手形状認識”. 電子情報通信学会論文誌, Vol. J80-D-II, No. 5, pp. 1209-1220, 1997.
- [21] 中嶋 正之, 柴 広有. “仮想現実世界構築のための指の動きの検出法”. 電子情報通信学会論文誌, Vol. J77-D-II, No. 8, pp. 1562-1570, 1994.
- [22] 渡辺 賢, 岩井 儀雄, 八木 康史, 谷内田 正彦. “カラーグローブを用いた指文字の認識”. 電子情報通信学会論文誌, Vol. J80-D-II, No. 10, pp. 2713-2722, October 1997.
- [23] James M. Rehg, Takeo Kanade. Visual Tracking of High DOF Articulated Structures: an Application to Human Hand Tracking. In *Proc. of European Conference on Computer Vision '94*, 2, pp. 35-46, 1994.
- [24] Quentin Delamarre, Olivier Faugeras. “Finding pose of hand in video images : a stereo-based approach”. In *Proc. of The 3rd Conf. on Face and Gesture Recognition*, pp. 585-590, 1998.
- [25] Roberto Cipolla, Nicholas Hollinghurst. “A Human-Robot Interface using Pointing with Uncalibrated Stereo Vision”. In *Computer Vision for Human-Machine Interaction*. Cambridge University Press, 1998.

- [26] 高橋 和彦, 坂口 竜己, 大谷 淳. “三眼視による実時間非接触非装着型三次元人物姿勢推定法”. 電子情報通信学会技術報告, Vol. PRMU99-94, No. 10, pp. 47-54, 1999.
- [27] 大野 一, 池内 克史. “多視点ステレオカメラによる手のトラッキング”. 電子情報通信学会技術報告, Vol. PRMU99-7, No. 7, pp. 45-52, 1999.
- [28] 鎌倉 矩子. 手のかたち 手の動き. 医歯薬出版株式会社, 1989.
- [29] Adalbert I. Kapandji. カパンディ 関節の生理学 (I) 上肢. 医歯薬出版株式会社, 1986.
- [30] 上羽 康夫. 手 その機能と解剖. 金芳堂, 1970.
- [31] 池原 和子, 上久保 恵美子, 比企 静雄. “手の形の操作の困難さと手話で使われる可能性との関係”. 電子情報通信学会論文誌, Vol. J79-D-II, No. 9, pp. 1562-1574, 1996.
- [32] 安室 喜弘. “手の3次元形状モデリング”. Master's thesis, 奈良先端科学技術大学院大学, 1997.
- [33] Shougo Tokai, Toshikazu Wada, Takashi Matsuyama. “Real Time 3D Shape Reconstruction using PC Cluster System”. In *Third Int. Workshop on Cooperative Distributed Vision*, pp. 171-187, 2000.
- [34] Larry Davis, Eugene Borovikov, Ross Cutler, David Harwood, Thanarat Horprasert. “Multi-perspective Analysis of Human Action”. In *Third Int. Workshop on Cooperative Distributed Vision*, pp. 189-223, 2000.
- [35] Robert T. Collins. “A Space-Sweep Approach to True Multi-Image Matching”. In *IEEE Computer Vision and Pattern Recognition*, pp. 358-363, 1996.
- [36] Hideo Saito, Takeo Kanade. Shape Reconstruction in Projective Grid Space from Large of Images. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 49-54, 1999.

- [37] Charles R. Dyer. Volumetric Scene Reconstruction from Multiple Views. In *Foundation of Image Analysis*. Kluwer, Boston, 2001.
- [38] Dan Snow, Paul Viola, Ramin Zabih. Exact Voxel Occupancy with Graph Cuts. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 345–352, 2000.
- [39] Andrew C. Prock, Charles R. Dyer. “Towards Real-Time Voxel Coloring”. In *Proceedings of Image Understanding Workshop*, pp. 315–321, 1998.
- [40] 澤 弘義, 坂本 尚久, 岩成 英一, 岡田 至弘, 金出 武雄. “複数任意視点からの Octree 表現による 3 次元形状復元”. 情報処理学会研究会報告 コンピュータビジョンとイメージメディア, Vol. 115, , 3 1999.
- [41] Richard Szeliski. “Rapid Octree Construction from Image Sequences”. *CVGIP:Image Understanding*, Vol. 58, No. 1, pp. 23–32, July 1993.
- [42] 昌達 K'z. “アルゴリズムラボ 第 12 回 骨格化と細線化処理”. In *C MAGAZINE*. September 2000.
- [43] Edward Catmull, James Clark. “Recursively Generated B-spline Surfaces on Arbitrary Topological Meshes”. *Computer Aided Design*, Vol. 10, No. 6, pp. 350–355, 1978.
- [44] John Lin, Ying Wu, Thomas S. Huang. “Modeling the Constraints of Human Hand Motion”. In *Proc. of Workshop on Human Motion 2000 (HUMO2000)*, pp. 121–126, 2000.
- [45] 鳥谷 浩志, 千代倉 弘明. 3次元CADの基礎と応用. 共立出版, 1991.
- [46] Gilles Debunne, Mathieu Desbru, Marie-Paule Cani, Alan H. Barr. “Dynamic Real-Time Deformations using Space & Time Adaptive Sampling”. In *Proc. of Intn' Conf. Siggraph 2001*, pp. 31–36, 2001.

- [47] William M Hsu, John F. Hughes, Henry Kaufman. "Direct Manipulation of Free-Form Deformations". In *Proc. of Intn' Conf. Siggraph 1992*, pp. 177-184, 1992.
- [48] Geoff Wyvill, Dean McRobie. "Modeling with Features". *IEEE Computer Graphics and Applications*, pp. 40-46, September/October 1997.
- [49] 松宮 雅俊. "陰関数曲面とパーティクルシステムを用いた仮想粘土細工による自由形状モデリングに関する研究". PhD thesis, 奈良先端科学技術大学院大学, 2002.
- [50] A. Pentland, J. Williams. "Good vibrations: Modal dynamics for graphics and animation". *Computer Graphics*, Vol. 23, No. 3, pp. 215-222, 1989.
- [51] D. Terzoulos, K. Fleischer. "Deformable models". *The visual computer*, Vol. 4, No. 6, pp. 306-331, 1988.
- [52] 小田 泰行, 千葉 則茂. "粒子ベースモデルによる粘土のビジュアルシミュレーション". 情報処理学会研究会報告 グラフィクスと CAD, No. 88-5, pp. 25-30, 1997.
- [53] Vijay Chandru, Swami Manohar, C. Edmond Prakash. "Voxel-Based Modeling for Layerd Manufacturing". *IEEE Computer Graphics and Applications*, pp. 42-47, November 1995.
- [54] 荒田 秀樹, 高井 昌彰, 高井 那美, 山本 強. "能動的ボクセル空間における仮想粘土モデリング —モデルの基本理念—". 電子情報通信学会論文誌, Vol. J82-D-II, No. 11, pp. 2008-2016, 1999.
- [55] Tinsley A. Galyean, John F. Hughes. "Sculpting: An Interactive Volumetric Modeling Technique". *ACM Computer Graphics*, Vol. 25, No. 4, pp. 267-274, July 1991.

- [56] Kevin T. McDonnell, Hong Qin. “Dynamic Sculpting and Animation of Free-form Subdivision Solids”. In *Proc. of Intn’ Conf. Computer Animation 2000*, pp. 126–133, 2000.
- [57] Kevin T. McDonnell, Hong Qin, Robert A. Wlodarczyk. “Virtual Clay: A Real-time Sculpting System with Haptic Toolkits”. In *Proc. of the 2001 ACM Symposium on Interactive 3D Graphics*, pp. 179–190, 2001.