コンピュータビジョンと
イメージメディア
128−4
（２００１．７．５）

多視点シルエット画像を用いた手の形状推定

上田 悦子† 松本 吉央† 今井 正和‡ 小笠原 司†

† 奈良先端科学技術大学院大学 情報科学研究科 ロボティックス講座
〒 630-0101 奈良県生駒市高山町 8916-5
Email: etsuko-u@is.aist-nara.ac.jp
‡ 鳥取環境大学 情報システム学科

あらまし 本論文では 3 次元自由形状入力インタフェースとして利用することのできる手形状推定の新しい手法を提案する．物を操作する際の手形状を推定対象とし，手指の各関節の回転・屈曲角度を推定するものである．提案手法では，多視点カメラシステムによって得られた画像から抽出した手領域をシルエット化し，入力として使用する．入力である複数視点のシルエット画像を統合し，手形状は Voxel モデルとして再構成される．この Voxel モデルと手の表面形状モデルとを 3 次元上でフィッティングすることにより関節角度の推定を行う．この手法を用いて手形状シミュレータと実画像を用いた 2 種類の実験を行い，その結果を元にして処理速度を更に上げることによってビジョンベースのインタフェースとして応用できる可能性を示す．

キーワード 手形状推定，シルエット画像，Voxel モデル，モデルフィッティング

Hand Pose Estimation Using Multi-Viewpoint Silhouette Images

Etsuko Ueda†, Yoshio Matsumoto†, Masakazu Imai‡, and Tsukasa Ogasawara†

†Robotics Lab, Graduate School of Information Science, NAIST
8916-5 Takayamacho, Ikoma city, Nara 630-0101 Japan
Email: etusko-u@is.aist-nara.ac.jp
‡Department of Information System, Tottori University of Environmental Studies

Abstract This paper proposes a novel method for hand pose estimation that can be used for 3D free-form input interfaces. The aim of the method is to estimate all joint angles to manipulate an object in the virtual space. In this method, the hand regions are extracted from multiple images obtained by the multi-viewpoint camera system. By integrating these multi-viewpoint silhouette images, a hand pose is reconstructed as a “voxel model”. Then all joint angles are estimated using three dimensional model fitting between hand model and voxel model. Two experiments were performed using the hand-pose simulator and real hand images. The experimentalresults indicate the feasibility of the proposed algorithm for vision-based interfaces, though it requires faster implementation for real-time processing.

keywords Hand Pose Estimation, Silhouette Image, Voxel Model, Model Fitting

1. はじめに

近年のコンピュータシステムの発展に伴ってコンピュータ上での形状表現は 2 次元から 3 次元へと移ってきた．2 次元の形状処理システムでは，操作される 2 次元図形はその次元を落とすことなくディスプレイ上に表示され，デザイナーはポインティングデバイスを用いて直接 2 次元形状を操作（作成・変形など）することができる．しかし，3 次元の形状処理システムでは通常 3 次元形状は 2 次元ディスプレイ上に投影されるため，従来のポインティングデバイスでは 3 次元形状を直接操作することができない．そのため今までの 3 次元形状処理システムを用いて 3 次元自由形状を設計することは非常に困難であり，デザイナーがイメージした形状を直感的に 3 次元形状設計システムに伝えることのできるインタフェースが求められている．

製品設計の初期段階において，意匠デザイナーはクレイモデルを用いてプロトタイプを作成する事が多い．このようなデザイナーの手動作を，自由形状入力インタフェースとして応用することは，直感的なインタフェース構築のための有効なアプローチであるといえる．また長時間の作業を要する設計の実務環境では，このようなインタフェースは非接触であることが望ましい．

本論文では 3 次元自由形状入力インタフェースとして用いることのできる，多視点シルエット画像を用いた新しい手形状推定手法を提案する．

2. 関連研究

これまでに提案されてきたビジョンベースの手形状推定手法は大きく 2 つのカテゴリに分けられる．

- Communicative な手形状の推定[1)~3)]
- Manipulative な手形状の推定[4)~6)]

前者は手話のための手姿勢推定や，VR インタフェースのための手形状認識などを含んでいる．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 種類のコマンドと手の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[2)]．また，Maggioni による Gesture Computer[3)] 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．これらの報告ではリアルタイムでの手形状認識は可能であるが，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，インタフェースとしての手動作はコマンドとしての使用に限られてしまう．

一方，後者は任意の手形状を推定対象としている．島田らは，単眼視動画像を用い緩やかな拘束条件のもとでのした形状推定をおこなっている[4)]．亀田らは単眼視シルエット画像を用い，画像と関節物体モデルとの 2 次元モデルマッチングにより手形状推定をおこなっている[5)]．しかし，これらの単眼視画像では奥行き情報を得ることができないため，単眼カメラシステムを用いて正確な手形状推定をおこなうことは困難である．

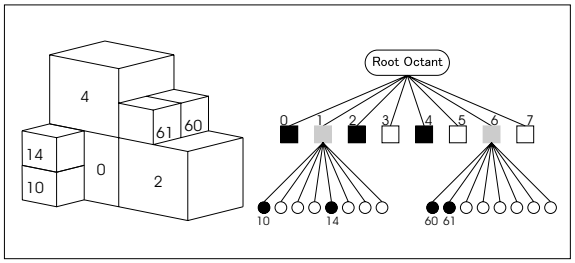

図 1 Octree 表現
Fig. 1 Octree representation

Delamarre らはステレオ画像から再構成した手の表面形状と手のモデルとの間に仮想的な力を発生させ，この力によって手形状を推定する手法を提案している[6)]．ステレオカメラシステムではステレオマッチングによって奥行き情報を得ることができる，しかし避けることのできないミスマッチングにより，推定精度が低下するという欠点を持つ．

複数のカメラを用いた多視点カメラシステムでは，単眼カメラシステムやステレオカメラシステムに比べて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[7),8)] を使えば，ステレオカメラシステムよりさらに安定した奥行き情報を得ることができる．本研究では，この再構成された 3 次元形状を手の観測データとして用いて手形状を推定する．

3. 手の 3 次元計測

3.1 Octree 表現

2 次元画像が Pixel の集合で自然な画像を表現するように，3 次元形状は Voxel と呼ばれる小さな立方体の集合体として表現することができる．しかしこの表現方法では複雑な形状を表現しやすいという長所を持つ反面，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，Octree 表現である．Octree 表現で用いられる各サイズの立方体を Octant と呼ぶ．Octree は**図 1** のように 8 分木による再帰的な構造になっている．それぞれの Octant は対象形状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，3 次元形状の中に完全に内包されている Octant は BLACK，3 次元形状に外側にある Octant は WHITE，3 次元形状の境界上にある Octant は GRAY と定義されている．GRAY と定義された Octant，更に 8 つの Sub-Octant に分割され，それぞれに対して更に属性を持つ．このようにして作成された 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．

3.2 複数シルエット画像からの Voxel モデル生成

本研究で用いた 3 次元形状再構成手法は，Shape from

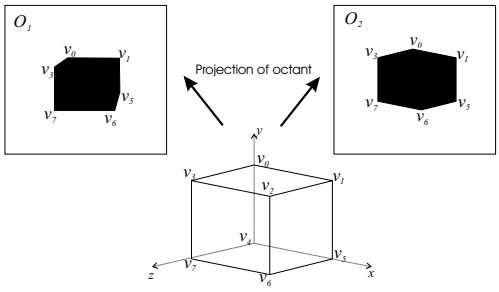

図 2 Octant の投影
Fig. 2 Projection of octant

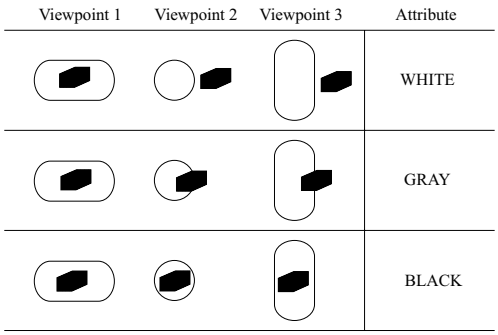

図 3 Octant の属性
Fig. 3 Attribute of octant

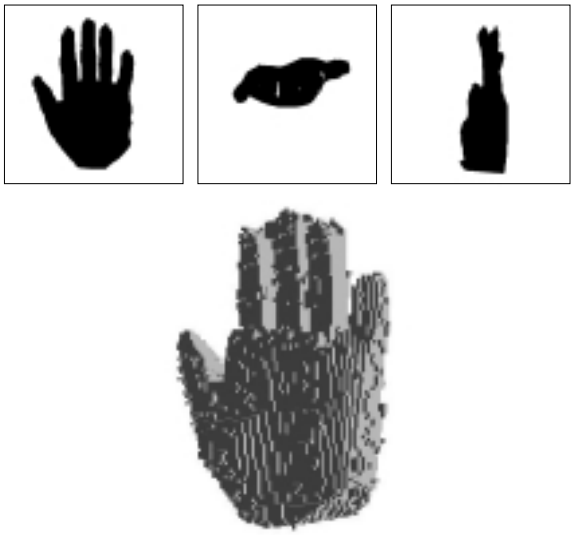

図 4 Voxel モデル生成
(上段：シルエット画像，下段：Voxel モデル)
Fig. 4 Reconstruction of voxel model
(upper : silhouette images, lower : voxel model)

Silhouette の手法である．再構成された 3 次元形状は Voxel モデルと呼ばれ手の観測データとして取り扱われる．この 3 次元形状再構成手法[7),8)] を以下に示す．

N を視点の数とし，視点 i で得られたシルエット画像を $S_i(i = 1, ..., N)$ 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 とする．

複数視点によるシルエット画像が与えられたとき，それぞれの Octant の属性を WHITE，BLACK，GRAY のいずれかに決定するために Octant の投影とシルエット画像間での交差判定を行う．**図 2** は各 Octant のそれぞれの投影面への投影形状を示す．ここでは視点 i での投影された Octant を O_i と定義している．

O_i と S_i の交差判定結果は IN，OUT，ON_BOARDER のいずれかに分類される．**図 3** に示すように，ある Octant についてのすべての交差判定結果が IN であるとき，その Octant の属性は BLACK となる．また，Octant についての交差判定結果が一つでも OUT であるとき，その Octant の属性は WHITE となる．交差判定の結果が両者いずれにもあてはまらない場合は，その Octant の属性は GRAY となる．GRAY となった Octant は更に Sub-Octant に分割され交差判定が繰り返される．Octree が指定したレベルまで達した時点で Octree による復元処理は終了する．このときの BLACK と GRAY の属性を持つ Octant の集合は対象物体を完全に内包し，かつ対象物体の形状を表すようになる．**図 4** に入力シルエット画像とそれらを用いて作成された Voxel モデルを示す．

4. 手のモデル

本研究で用いる 3 次元の手のモデルは 1) 骨格モデル，2) 表面形状モデル から成り立っている．このモデルは安室らが提案した手のモデル構造[9)] を元にしている．

4.1 骨格モデル

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，モデル化している．各指は，リンク（骨）とジョイント（関節）の集合として**図 5** 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．このモデルを骨格モデルと呼ぶ．自由な手動作を表現するために，本研究での各リンクの自由度は図 5 に示すように配置した．骨格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．

4.2 表面形状モデル

各関節位置が決まると，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できなければならない．本研究では手の表面形状を 3 角形パッチの集合で表現し，また各 3 角形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，それぞれの 3 角形パッチの各頂点データに対してどの骨リンクに対応してその位置を決定するのか示す属性を与えている．

5. 手形状推定

5.1 提案手法の概要

骨格モデルの各関節角度の推定は，手の表面形状モデ

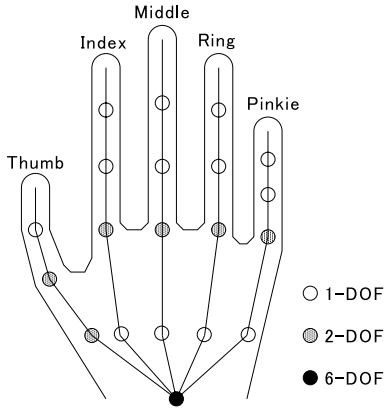

図 5 骨格モデル
Fig. 5 Skeletal hand model

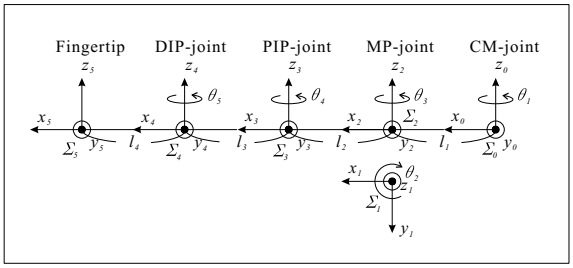

図 6 中指の関節軸
Fig. 6 Joint axis of middle finger

ルを Voxel モデルにフィッティングさせることによって行われる．2 次元の観測データと 3 次元モデルとを用いたモデルフィッティングには 2 種類のアプローチがある．

A. 3 次元モデルを 2 次元平面に投影し，2 次元でのモデルフィッティングを行う方法 (従来手法).

B. 2 次元観測データから 3 次元形状を再構成し，3 次元でのモデルフィッティングを行う方法 (提案手法).

我々の提案する手法は B のアプローチである，この手法は A のアプローチと比べて，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．そのため提案手法によるモデルフィッティングでは従来手法よりもシンプルなアルゴリズムでフィッティングを実装することができる．

Voxel モデルは，空間を立方体に分割した Voxel 空間中において，手が占有する領域を提示している．表面形状モデルもまた手の存在する位置を 3 角形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 Voxel モデルに内包されているとき，骨格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．

今，ある関節位置の関節角度を $\mathbf{a}_i = \{a_i(k)|0 \leq k < 3\}$ ($a_i(k)$ は関節 i の回転軸 k における関節角度) と表現したとき，手の姿勢は $P = \{\mathbf{a}_i|0 < i < r\}$ (r は関節の総数) 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．さらに V を Voxel モデルで占有されている領域と定義すると，手形状推定は $L \subset V$ を満足させる P を決定する問題であると言える．これは，評価関数 $Out = \{\mathbf{p}(m) \not\subset V|0 \leq m < q\}$ の元で $Out = 0$ となるような P を決定するということで実現される．

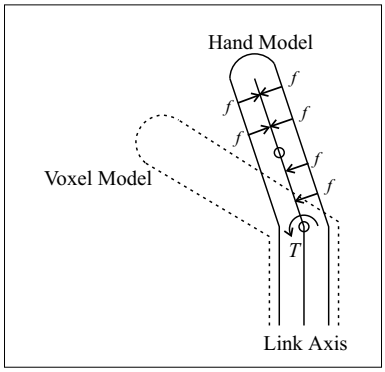

図 7 トルクの発生
Fig. 7 Generation of torque

そのために，Out なる点群に対して Voxel モデルに近づくような引力を発生させ，その引力の方向に従い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．

5.2 推定アルゴリズムの詳細

推定アルゴリズムの詳細は以下のとおりである．

Step1 視点角度の違う複数カメラによってキャプチャされた画像をシルエット化する．

Step2 得られた多視点シルエット画像を用いて Voxel モデルを生成する．

Step3 骨格モデルと観測した手形状を表現する Voxel モデルとを比較する．表面形状モデルの頂点のうち Voxel モデルの外に位置する座標に着目する．

Step4 着目した座標に対して**図 7** のように関節軸方向に力 f を発生させる．発生した 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 トルク方向に関節角度を $\Delta\alpha$ 度変化させる．

Step6 新しい関節角度より関節位置を再計算し，表面形状モデルを構成するそれぞれの 3 角形パッチ頂点位置を更新する．

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．

6. シミュレータを用いた手形状推定

6.1 手形状シミュレータ

手形状シミュレータを用いたて形状推定実験を行った．このシミュレータは，骨格モデルの各関節角度を入力することによって，自由な手形状を生成することができる．**図 9** は手形状シミュレータの画面を示す．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．実験では 3 つの視点 (手の正面，側面，上面) を設定した．

シミュレーション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ものとしているため，シミュレー

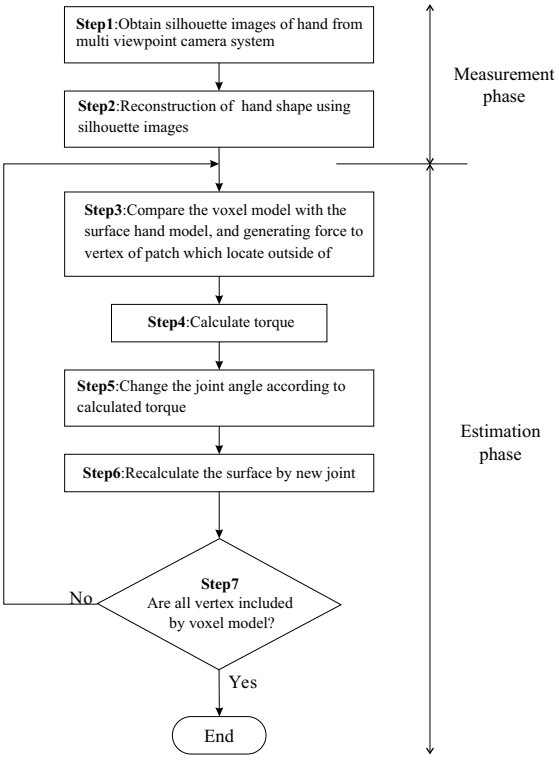

図 8 関節角推定フロー
Fig. 8 Joint angle estimation flow

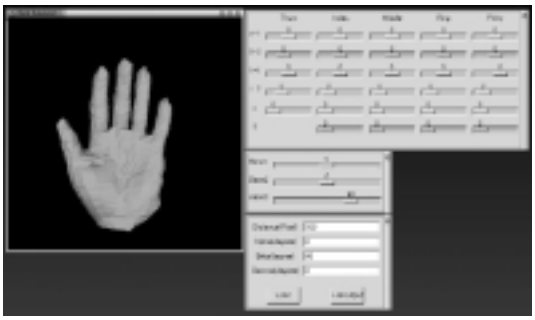

図 9 手形状シミュレータ
Fig. 9 Hand pose simulator

タ上に生成した Voxel モデルを重ねて表示すれば，**図 10** のように，関節の動きによって変化した表面形状部分のみが Voxel モデルの外に出て表示される．

6.2 推定結果

2 種類の手形状を生成し，それらの推定結果を**図 11**, **図 12** に示す．どちらの図も (a) は Voxel モデルを重ねた表面形状モデルの初期形状を，(b1) ∼ (b3) は表面形状モデルが Voxel モデルに収束する過程を，(b3) は Voxel モデルに完全に包含された最終推定表面形状モデルをそれぞれ示している．

6.3 評価

Voxel モデルの Octree レベルは推定精度に影響を与える．Voxel モデルの最小 Octant サイズは 1 辺の長さはそれぞれ，レベル 6 では $8mm$ ，レベル 7 では $4mm$ ，

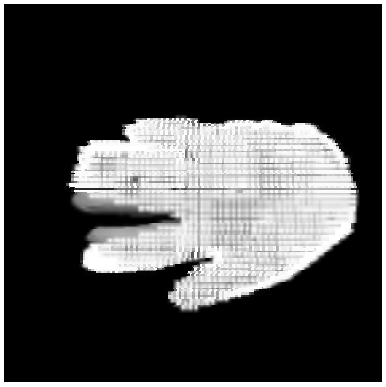

図 10 重ねて表示した骨格モデルと Voxel モデル
Fig. 10 Superimposed image of skeletal hand model and voxel model

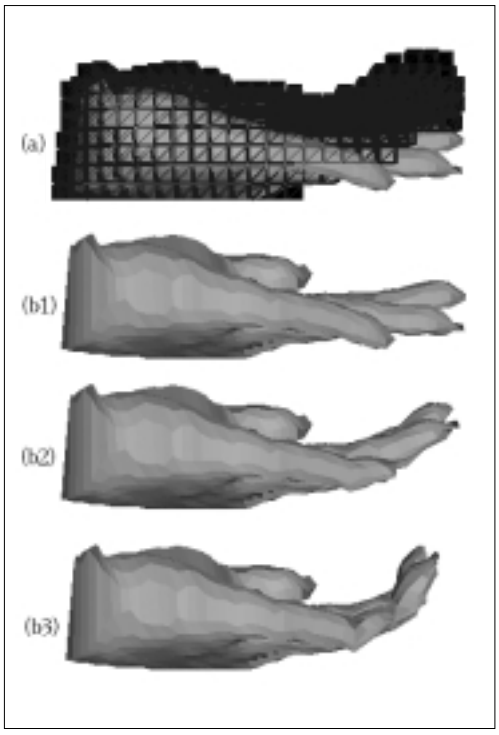

図 11 収束過程 (形状 1)
Octree レベル = 6 (最小 Octant サイズ = 1 辺 8mm)
Fig. 11 Convergence process (pose1)
octree level = 6 (minimum octant = 8mm cube)

レベル 8 では $2mm$ である．Voxel モデルの Octree レベルの違いによる推定精度の比較実験を行った．推定される手形状を以下に示す．

- 示指の MP-屈曲関節を 10 度前屈
- その他の指はまっすぐ伸ばす

また、**図 13** は示指 MP-屈曲関節について Octree レベル 7 と 8 における関節角度推定誤差のグラフを表している．反復回数は 20 回で，各反復毎にプロットしている．推定誤差は以下のように定義している．

$$error = |e\_ang - t\_ang|$$

(e_ang は推定角度，t_ang は設定した関節角度である．)

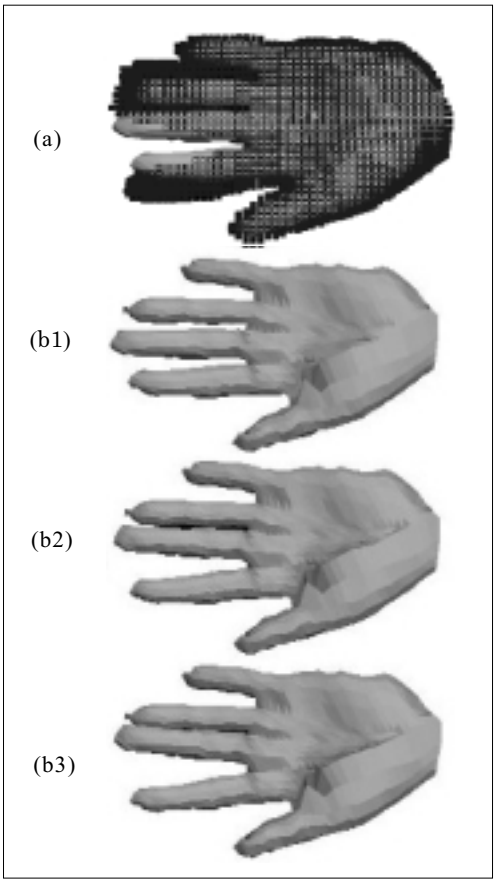

図 12 収束過程 (形状 2)
Octree レベル = 7 (最小 Octant サイズ = 1 辺 4mm)
Fig. 12 Convergence process (pose2)
octree level = 7 (minimum octant = 4mm cube)

図 13 からは，復元レベルが大きければ大きいほど推定精度が上がることが容易に読み取れる．**図 14** は収束率を示している．収束率は以下のように定義される．

$$rate = \frac{in\_vertex}{all\_vertex} \times 100(\%)$$

($rate$ は収束比率，in_vertex は Voxel モデル内部に位置する表面形状モデルの頂点数，all_vertex は表面形状モデルの総頂点数，現状のモデルの総頂点数は 2010 である．)

図 13, 図 14 は Octree レベルが推定精度と推定速度へ影響を与えることを示している．推定精度における比較では Octree レベルが高ければ高いほど推定精度は高いが，収束速度が遅くなってしまう．このように Octree レベルの設定は推定精度を優先するのか推定速度を優先するのかを考慮して行う必要がある．

7. 実画像を用いた手形状推定

7.1 実験システム

実際の手の画像に対して手形状推定を行うために実カメラシステムを構築した．**図 15** 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

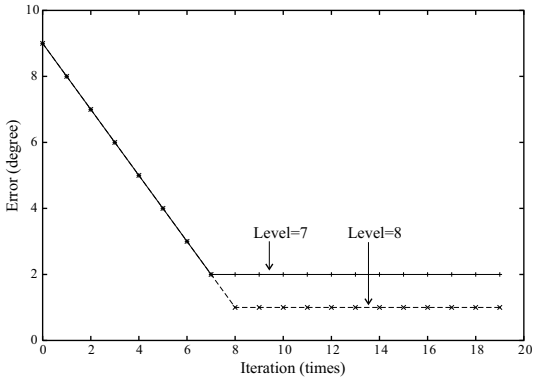

図 13 MP 関節の推定誤差
Fig. 13 Estimation error of MP-joint

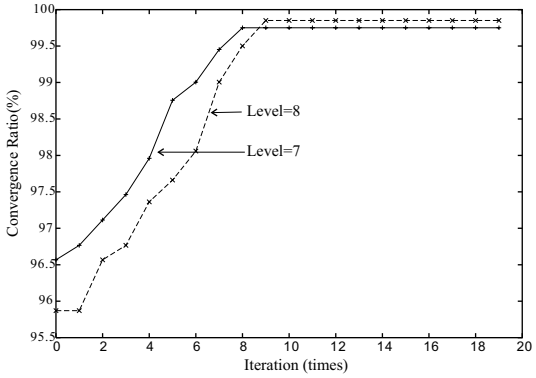

図 14 収束率
Fig. 14 Convergence ratio

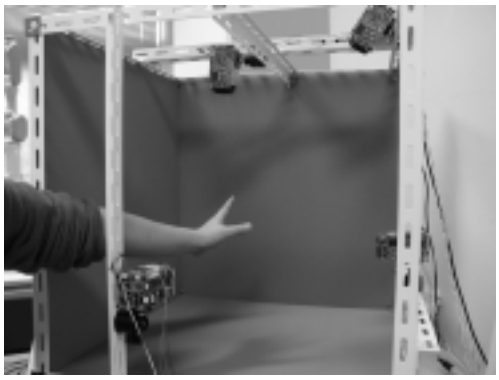

図 15 実験環境
Fig. 15 Experiment environment

している．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．

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．

7.2 実 験 結 果

図 16 は実画像を用いた推定結果を示している．実画像においても提案手法の有効性が確認されたが，リアルタイム推定を行うことを考えると, 推定速度はまだ非常に低

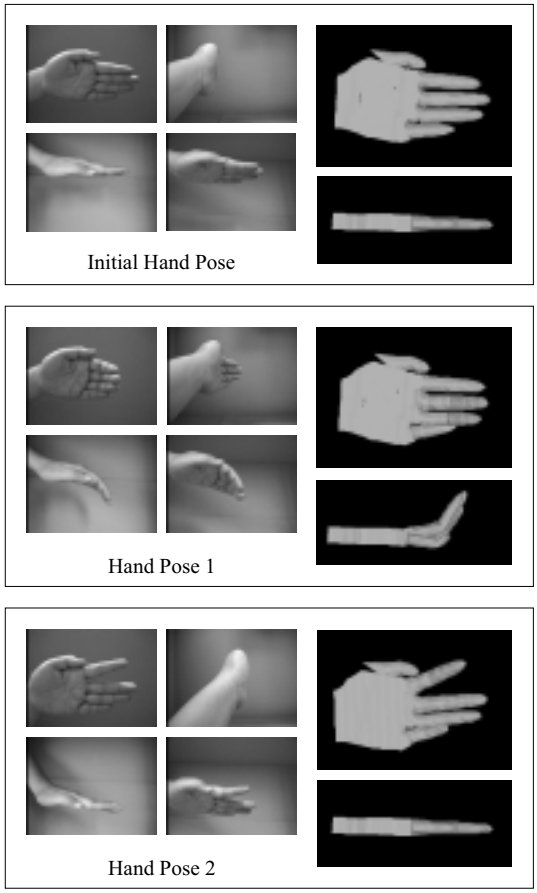

図 16 実画像を用いた手形状推定
Fig. 16 Hand pose estimation using real images

表 1 推定処理時間
Table 1 Processing Time

Image Capture	130
Construction of Voxel Model	70
Pose Estimation	300
Total	500

unit : msec

い．現在の実験システムにおける平均の推定時間を**表 1** に示す．カメラ台数は 4 台，Octree レベルは 7，CPU は PentiumIII 1GHz Dual プロセッサである．

8. おわりに

本研究では 3 次元自由形状入力などのインタフェースとして利用可能なビジョンベースの手形状推定法を提案した．提案手法では手を骨格モデルと表面形状モデルで表現しており，また手の Voxel モデルは多視点カメラシステムによって得られたシルエット画像から再構成される．骨格モデルの関節角度は表面形状モデルと得られた Voxel モデルとの 3 次元モデルフィッティングにより推定される．提案手法の有効性を検証するために，手形状シミュレータを用いて任意の手形状を生成し，その手形状を推定した．最後に実画像を用いた手形状推定実験を行った．

現時点で提案手法には大きく 2 つの問題点が存在する．一つは誤推定の問題である．これは手の骨格モデルと表面形状モデルのモデリングエラーによって引き起こされるものであり，ユーザーの手形状を高精度で再現するモデリング手法が必要である．もう一つは処理速度の問題である．インタフェースとして実用可能な手法となるためには現時点の約 5 倍，少なくとも 10Hz の処理速度が必要と考えている．今後は，現時点での問題点を順次解決し，3 次元形状操作システムのインタフェースとして実用化を目指していく予定である．

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 PAMI*, Vol. 19, No. 7, pp. 677–695, 1997.
2) Akira Utsumi, Jun Ohya, Ryouhei Nakatsu. “Multiple-Hand-Gesture Tracking using Multiple Cameras”. In *Proc. of International Conference on Computer Vison and Pattern Recognition*, pp. 473–478, 1999.
3) C. Maggioni, B. Kämmerer. “Gesture Computer — History, Design and Applications”. In *Computer Vision for Human-Machine Interaction*. Cambridge University Press, 1998.
4) Nobutaka Shimada, Yoshiaki Shirai, and Yoshinori Kuno. “3-D Pose Estimation and Model Refinement of An Articulated Object from A Monocular Image Sequence”. In *Proc. of The 3rd Conf.on Face and Gesture Recognition*, pp. 268–273, 1998.
5) Yoshinari Kameda, Michihiko Minoh, and Katsuo Ikeda. “Three Dimensional Pose Estimation of an Articulated Object from its Silhouette Image”. In *Proc. of Asian Conference on Computer Vision ’93*, pp. 612–615, 1993.
6) 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.
7) Larry Davis, Eugene Borovikov, Ross Cutler, David Harwood and Thanarat Horprasert. “Multi-perspective Analysis of Human Action”. In *In Third Int. Workshop on Cooperative Distributed Vision*, pp. 189–223, 2000.
8) Richard Szeliski. “Rapid Octree Construction from Image Sequences”. *CVGIP:Image Understanding*, Vol. 58, No. 1, pp. 23–32, July 1993.
9) Yoshihiro Yasumuro, Qian Chen, and Kunihiro Chihara. “Three-dimensional modeling of the human hand with motion constraints”. *Image and Vision Computing*, Vol. 17, No. 2, pp. 149–156, 1999.