

パイプライン型画像処理装置を用いた 実時間手形状認識

石淵 耕一・竹村 治雄・岸野 文郎 (ATR通信システム研究所)

This paper describes a new hand shape recognition method for man-machine interfaces using pipeline image processor. It achieves high processing speed and high positional accuracy. They are obtained by using a simple hand's feature extraction method running on a pipeline image processor. We have developed an algorithm for noncontact recognition of hand shape, and we have incorporated this into a virtual environment to enhance the operator's sensation of reality.

Keywords: real-time; recognition; image processing;

1. はじめに

人とコンピュータ間の自然なコミュニケーションを目指したマンマシンインタフェースの研究が盛んである。特に近年のコンピュータ技術の進歩は仮想の3次元環境を人間の周囲に提示し、その中に置かれた仮想物体を人間が直接操作することを可能とした。このような仮想環境を用いてこれまでいくつかの応用システムが提案されている[1][2]。筆者らも「遠隔地にいる人々があたかも一堂に会する感覚で会議を開くことのできるシステム」を臨場感通信会議システムと呼び、その実現へ向けての研究を進めている[3]。

一般にこのような仮想環境を利用したシステムでは、表示装置としてHMDあるいは両眼立体視用眼鏡(固定表示面との併用)といった眼鏡型装置、入力装置としてデータグローブあるいは同機能をもつグローブ型装置が用いられている。しかし、このような接触型装置の装着は会議システムのような応用を考える

時、参加者の臨場感を損なうことにもなりあまり好ましくない。表示装置の非接触化に関してはレンチキュラー・スクリーンを用いた方式が検討されているが[4]、入力装置の非接触化についての議論は少ない。その理由としては、どのようなセンサーを用いるにせよ、実時間処理が非常に困難であること、実時間性を考慮せずとも、変形が著しく特徴抽出が困難な手形状を正確に認識するのは難しいこと等が挙げられる。画像処理を用いて形状を認識しようとする試みはコンピュータ・ビジョンの分野で多くの研究がなされ、優れたアルゴリズムも数多く提案されているが、実時間処理を念頭に置いた非剛体形状認識についての報告は少ない。

本稿では、これら問題点を考慮の上、非接触型手形状入力装置を実現するための画像処理を用いた実時間手形状認識手法の概要を述べた後、高速コマンド生成

Real-Time Hand Shape Recognition Using Pipeline Image Processor
ISHIBUCHI Koichi, TAKEMURA Haruo, KISHINO Fumio
(ATR Communication Systems Research Laboratories)

アルゴリズムを提案する。また、このアルゴリズムを用い構築したマンマシンインタフェース環境を紹介するとともに、その環境にて行ったポインタ精度実験、コマンド認識実験及び特徴点検出速度実験結果について述べる。

2. 実時間手形状認識手法の概要

仮想環境下で使用する非接触手形状検出装置を実現するため、画像処理を用いた手形状認識手法の概要を述べる。手形状認識手法を決定する際考慮すべき点は、処理速度、ポインタ精度及びコマンド認識率である。ポインタ精度とは仮想環境下のどの仮想物体に操作を加えるかを決定する際用いられる空間位置の精度を指すこととし、グローブ型装置の場合は併用して用いられる磁気センサの空間位置精度がこれにあたる。コマンド認識率は手形状に付与したコマンドがどの程度認識できるかを表すもので、本稿では手の空間位置と独立であるとする。これら3点と操作性との係わりに関する議論は別の機会に譲ることとし、本稿ではサンプリング周波数が10 Hz以上の処理を実時間処理と呼び、実時間で高いポインタ精度及びコマンド認識率を実現する手法を提案する。

実時間処理を可能にするため図1に示す手の記述法を用いる。

rule1: 手の形状を手の重心と指先の3次元座標で記述する

手の指間接の曲げ角度の制約等を考慮すると、この仮定を用いて多くの手形状の識別が可能であることが報告されている[5]。手が非剛体であることを考慮すると、3次元座標を検出するため最低2台のカメラが必要となり、手形状認識は図2のような2段階で達成される。また、ポインタ座標はいずれかの指

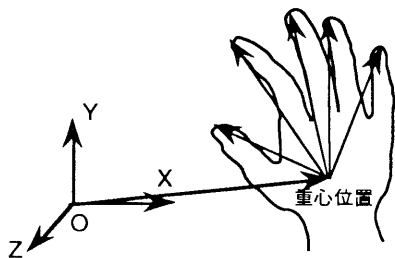


図1 手の記述法

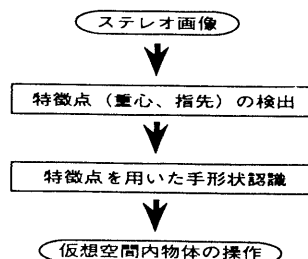


図2 手形状認識のフロー

先あるいは重心位置とすれば良い。

画像から指先を検出する際問題となるのは、指が手の平や他の指に遮蔽されるいわゆるオクルージョン問題と、ノイズ問題である。オクルージョン問題はコンピュータ・ビジョン分野でも明確な解法はなく、その推定ができるのみである。本稿ではオクルージョンの推定問題は取り扱わないが、システムの連続動作を保証するためノイズ問題と同様の扱いをし、以下の簡単なルールを適用する。

rule2: 指先が検出されない指は曲げられているか、伸ばしているがオクルージョンが生じているかのいずれかとする

rule3: 伸ばした指のオクルージョンはステレオ画像上で同時に任意の $(2n+1)$ フレーム間最大 n フレームしか生じない

伸ばした指の本数のみに意味を持たせることにすると、上記ルールは指が伸びているかどうかの判断を $(2n+1)$ フレーム間で行なうことを示している。指の本数認識には $(2n+1)$ フレームの遅れを生じることになるが、その認識率を上げることができる。図3には $n=1$ の例を示す。ポインティング位置も $(2n+1)$ フレームのデータを用い補正することで、精度を上げることができる。但しこれらは実時間処理の達成及びrule3に従

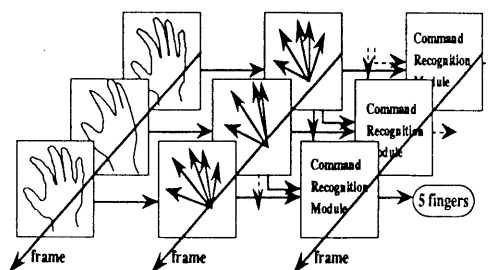


図3 指本数認識のフロー

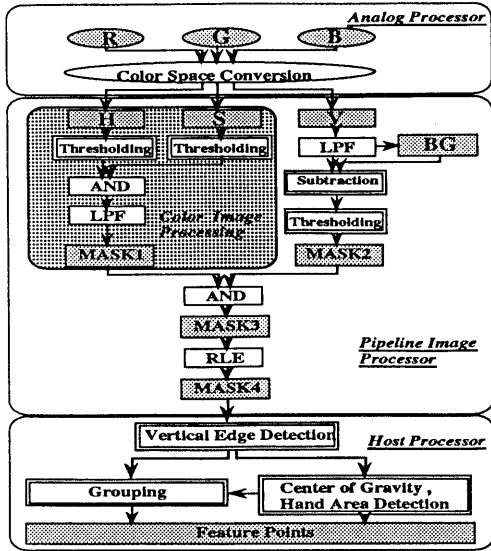


図4 2次元特徴点検出装置
うオペレータの操作が不可欠である。第3章では、実時間処理を達成するための具体的なコマンド生成アルゴリズムについて述べる。

3. コマンド生成アルゴリズム

3.1 2次元特徴点検出

手形状の知識が必要でない前処理をパイプライン化することで、2次元特徴点検出を実時間でこなす。以下、この特徴点検出装置(図4)を3つのユニットに大別し示す。

(1) アナログ処理ユニット

RGBアナログ信号を入力し、HSVデジタル画像を出力する。

(2) パイプライン画像処理ユニット

HSV画像を入力し、ラン・レンジ・コード化画像MASK4を出力する。表1に中間生成画像の説明、図5に入力された明度画像V及び最終検出結果

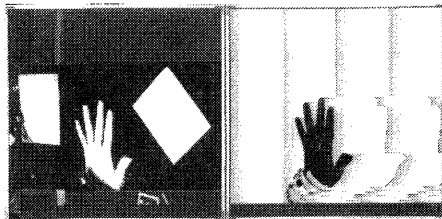


図5 入力明度画像、最終検出結果

MASK1	手に固有な色相、彩度それぞれのしきい値を用いて手領域を抽出した2値画像
BG	明度の背景画像
MASK2	背景画像BGと入力明度画像Vとの差分をとり2値化した画像
MASK3	MASK1とMASK2との共通領域画像
MASK4	MASK3をRun Length Encoder (RLE) チップを用いコード化した画像

表1 中間生成画像

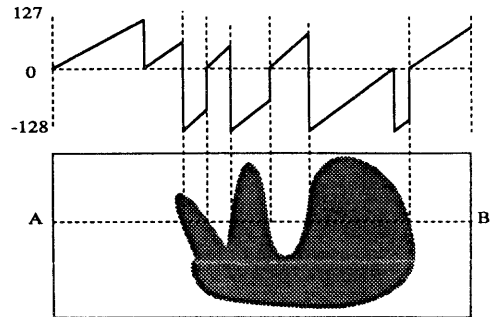


図6 ラン・レンジ・コード化処理

(MASK4と検出された特徴点)を示す。但し、各画像は512×480の解像度及び8bits/pixelのデータを持つ。

図6には線分AB上のピクセルに対してラン・レンジ・コード化した例を示す。AB上の各ピクセルは領域内、領域外で2値化されている。AからBに向かってコード化すると、領域内は正(初期値:0)、領域外は負(初期値:-128)の数字が割り当てられ、それらの数字は順にインクリメントされる。但し、128回インクリメントされた場合は初期値に戻る。この画像を作成しておけば、画像すべてを検索しなくても領域変化点のアドレスを高速に検出することができる。但し、サンプリング周波数はエッジが垂直に近いほど高く、完全な水平エッジは検出されない。

なお、これらの処理を安定に行なうため以下のルールに従うものとする。

rule4: 背景は一様、光源は白色光を用い、背景・光源の変化はほとんどない

rule5: オペレータは手と色相の違う長袖の服を着用
rule4に関しては、MASK3をMASK1で代用できれば

(肌の色に依存) ある程度の背景・光源変化が許される。*rule5*に関しては、オペレータが半袖等の場合は領域抽出が不安定になるため設けられた。

(2) ホスト処理ユニット

MASK4を入力し、手の重心および指先位置を出力する。このユニットでは手形状の大きな知識を用いて特徴点を検出する。

rule6: 伸ばした指はほぼ同一方向を向いている

rule7: オペレータは画像に垂直に手が写るよう操作する

*rule7*は一見厳しいルールのようにであるが、図8に示すような大型表示面(両眼立体視用眼鏡を併用)のシステムを用いれば、オペレータはあまり意識せずにこのルール満たす操作ができる。以下このユニットの3つのモジュールについて述べる。

1) 垂直エッジ検出モジュール

ラン・レングス・コード化画像MASK4からエッジを検出する。

2) 重心・領域検出モジュール

エッジ情報から手の重心位置及び領域の最大・最小を計算し、グルーピング時の処理領域(図6)を作成する。

3) グルーピング・モジュール

処理領域内エッジに対して、 y の最小値から最大値に向け各行ごとに以下の2つのステップを繰り返し、指先と見かけの指の長さを検出する。

STEP1: 第 n 行のエッジ (x,y) が存在する i 個のどのグループにも属さなければ、そのエッジを $i+1$ グループの指先とする。

STEP2: i 番目のグループに属する場合はその点と i 番目の指先から直線 $L(i)$ 及び $L(i)$ を求める。

但し、グループに属するかどうかの判断は基本的には $L(i)$ 上にあるかどうかで決まるが、その許容範囲は $L(i)$ 及び $L(i)$ と (x,y) との距離の関数とする。

指先と見かけの指の長さが得られた後、ノイズ処理

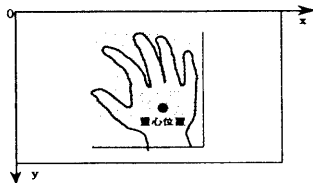


図6 処理領域

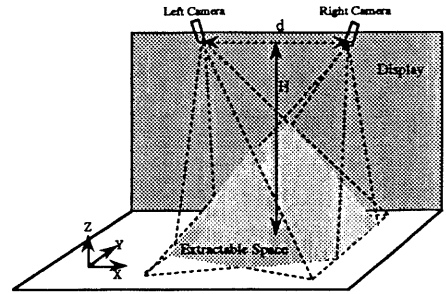


図7 検出可能空間

として、見かけの長さ及び指先と重心との距離にそれぞれしきい値を設け、最終的な指先を決定する。

3.2 ポインタ計測

両画像上にポインタとして用いる特徴点対が見つければ、その3次元座標は三角測量で容易に計測することができる。ポインタは重心位置の対あるいは最もスクリーンに近い特徴点の対を選ぶことができる。

三角測量を用いた空間位置の検出精度は画像上の対応づけが既知であるとする、主に、2つのカメラ間距離 d 、輻輳角 α 、カメラの特性(解像度 512×480 で固定、視野角 θ 等)及び計測点とカメラとの距離に依存する。システムを構築する際には、図7に示すような計測可能空間をできるだけ大きくし、 X 、 Y 、 Z 方向でなるべく均等の解像度を得たい。 $H = 138.0\text{cm}$ 、 $\theta = 34.2^\circ$ で固定すれば、検出可能空間内 X 、 Y 、 Z 方向の最小解像度と検出可能空間体積の d 、 α に対する関係はおおよそ図8のようになる。本稿では対応付けの容易さをも考慮して $d = 120.0\text{cm}$ 、 α

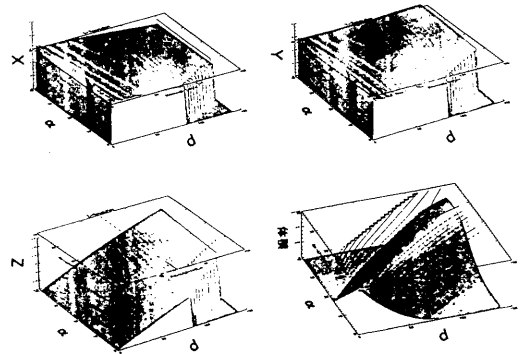


図8 解像度、検出可能空間

=30.0° と設定した。

なおノイズや計測誤差等から生じる高周波成分を取り除くため、三角測量で計測されたポイント座標は3次の移動平均フィルタを用いて補正される。

3. 3 コマンド認識

表2のように検出された指の本数にコマンドを割り当てる。

コマンド	指の本数
掴む	0以上4未満
離す	4以上5以下

表2 コマンド

但し、nフレーム目の左右画像から検出された指の本数をそれぞれLEFT(n)、RIGHT(n)とすれば、認識に用いた指の本数Finger(n)は、

$$Finger(n) = \sum_{t=n-2}^n (\text{MAX}(\text{LEFT}(t), \text{RIGHT}(t))) / 3$$

とした(図3)。

4 実験

4. 1 ハードウェア構成

実験で用いたハードウェア構成を図9に示す。オペレータがカメラの検出可能空間に手を入れると、手形状認識ユニットで認識されたコマンド及びポイント座標が、表示ユニットに渡され、表示ユニットに構築された仮想環境内の仮想物体をコマンドに従い操作できる(図10)。以下では、この環境下で行なわれたポイント精度実験、コマンドの認識実験及び特徴点検出

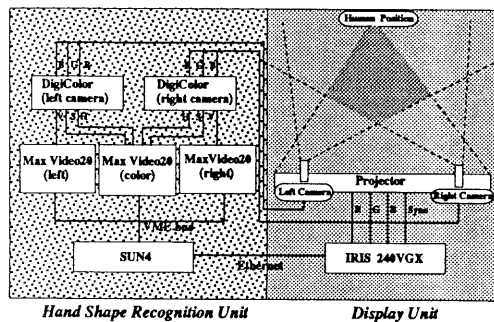


図9 ハードウェア構成

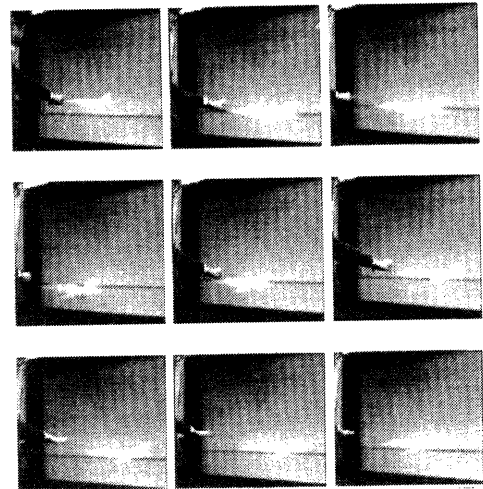


図10 仮想物体の操作
速度実験について述べる。但し、これらの実験は全て本稿に示したルールに従って行なった。

4. 2 ポインタ精度検出実験

(1) 実験方法

指先と同じ程度の太さの棒に磁気センサを付け、検出可能空間内の45cm立方でポインタデータと磁気センサデータとを計測した(図11)。ポインタ位置は磁気センサの先端を用いるとし、磁気センサを空間のXYZ軸(ワールド座標系と呼ぶ)に沿い88点、0.5cm間隔で移動し計測されたデータとのズレを測る。但し本稿ではX軸に沿い計測したデータのみを示す。磁気センサ及び特徴点検出装置の座標系は、予めワールド座標系と一致するようにキャリブレーションしておく。磁気センサーのソースは、ワールド座標系で(0,0,-5)にセットし、単位はcmを用いる。

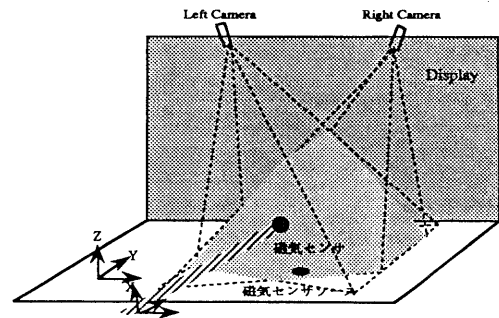


図11 ポインタ精度実験

(2) 実験結果

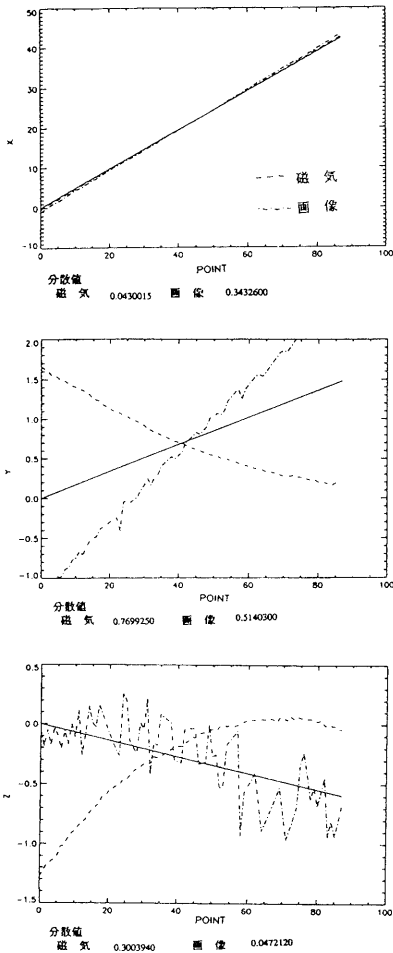


図 12 ポインタ精度

磁気センサと特徴点検出装置の精度が両者の分散値を比較するとほとんど変わらない結果がでた。キャリブレーションが正確に行われていないといった理由もあるが、特徴点検出装置ではたかだか1.2cm程度のずれしか生じないということは重要なことである。

4.3 コマンド認識・検出速度実験

(1) 実験方法

3人の被験者に仮想物体を掴む、離す、この2種の動作を20回行ってもらい、誤動作の回数を数えた(図10)。但し実験前に、被験者には本稿に書かれ

たルールを説明し、5分間の練習を行なった。また同時に特徴点検出速度を計測した。

(2) 実験結果

	誤認識回数 (回)	誤認識率 (%)	平均検出速度 (frames/sec)
被験者1	1	0.05	12.2
被験者2	0	0.00	13.8
被験者3	3	0.15	10.9
平均	1.33	0.67	12.3

表3 認識精度・検出速度実験

誤認識率は平均0.67%であったが、熟練すれば認識率はさらに上がるものと思われる。

検出速度はいずれも10frames/sec以上で、目標としていた実時間処理を達成できた。なお、平均検出速度の違いは、画像上の処理領域の大きさに依存するものである。

5 まとめ

バイブライン型画像処理装置を用いた実時間手形状認識手法と、その手法を用い作成した手形状認識装置について述べた。またその装置を用いマンマシンインタフェース環境を構築し、その環境下で行なったポイント精度、認識精度、検出速度実験により手法の有用性を示した。他にも実時間で手形状を認識する手法が提案されているが[6]、撮像された手の大きさに依存しない本手法はより有効であろう。今後はオクルージョン対策として予測フィルタ等の検討を進めていく予定である。

参考文献

- [1] S.Fisher et al: Virtual Environment Display System; ACM 1986 Workshop on Interactive 3D Graphics (1986)
- [2] 広瀬: 仮想環境: 第6回ヒューマン・インタフェース・シンポジウム論文集(1990)
- [3] 岸野、山下: 臨場感通信のテレコンファレンスへの適用: 信学会研究会IE89-35(1989)
- [4] N. Tetsutani et al: Stereoscopic Display Method Employing Eye-Position Tracking: The International Symposium on 3D Image Technology and arts (1992)
- [5] 大西、竹村、岸野: 手形状記述の一提案: 信学春全大分冊pp252-252(1990)
- [6] 長谷川他: VASのための入力同画像に反応する実時間人物表情の合成: 第7回ヒューマン・インタフェース・シンポジウム(1991)