

輪郭線のマッチングに基づく重なりあった物体の認識

RECOGNITION OF OVERLAPPING OBJECTS BASED ON MATCHING THE OBJECTS' BOUNDARY

張鴻賓 美濃導彦 池田克夫
Hongbin ZHANG Michihiko MINOH Katsuo IKEDA

京都大学工学部情報工学科
Department Of Information Science, Kyoto University

あらまし 本報告では、重なっている物体の認識に関して、輪郭線の特徴点のマッチングに基づく手法を述べる。物体の二次元形状を多角形で近似し、曲率の大きい点と円弧の端点の系列で輪郭線を記述する。このようにして、物体の認識問題を特徴点のマッチングと捕らえ、点パターンのマッチング・アルゴリズムを適用する。このマッチング・アルゴリズムでは、マッチングの誤差を計算するために基準線対を定義し、局所的な位置拘束も利用する。また、形状シフト数の概念を導入し、無駄な探索をさせている。これによって、アルゴリズムの速度を高めることができた。実験の結果、本アルゴリズムの有効性を確認した。

ABSTRACT A method for recognition of partially occluded and overlapping objects in composite scenes is presented. This method is based on representing the objects with polygon approximation. After defining base lines by using local constraints, vertices of polygons are sorted by angles to the base line, then a fast algorithm for point matching is applied. In order to reduce the number of possible base lines, a concept of shape shift number is introduced. Experimental results illustrate the effectiveness of the method.

1. はじめに

物体の二次元形状の認識問題は、画像認識における重要な問題として、前から研究されている。特に、重なっている物体の二次元形状の認識問題は、生産の自動化やロボットビジョンなどの分野では、特に重要な問題である。これまでの研究は、参考文献 [1] で詳しく紹介されている。

単一物体の二次元形状の認識においては、大局的な特徴、例えば、領域の重心、面積、境界線の長さ、領域の形状を表わす各種のメジャー (compactness, elongatedness) や、モーメント、チェーン・コード、フーリエ記述子などがよく使われる。しかし、いくつかの物体が重なっていたり、或は互いに接触していたりする場合には、上述の大局的な特徴は使えない。これは主として、次の二つの理由による。

(1) 単一物体の特徴が複合シーンの中では失わ

れてしまうこと。

(2) 偽りの特徴が出てくること。

これらの問題を解決するために、局所的な特徴を用いる手法が、各種報告されている。以下、本報告に関係がある幾つかを紹介する。L.S.Davis と O.D.Faugeras 等は二次元形状を多角形で近似し、弛緩法を形状のマッチングに適用した [2, 3]。また最近、Faugeras 等は、“privileged segment” のマッチングによる仮説の生成と検証に局所的な拘束を利用する方法も提案している [4]。R.L.Kashyap 等は二次元形状を多角形で近似し、マッチングを表現する “association graph” の clique を捜し出すことによって、重なっている物体を識別する手法を提案している [5]。T.L.Turney 等は、物体の輪郭線から subtemplates を抽出した後、Hough 変換の手法を重なりあったの物体の認識に適用した [6]。そのほか、マッチング木の探索によって、重なっている物体の認識を行なう手法も報告されて

いる [7]。

重なりあった物体を認識する問題は、画像処理とマッチング問題とに分けられる。画像処理が完全であると仮定しても、マッチング問題は一般的には計算量が多く、時間かかる。実際には、画像処理段階で、特徴の脱落と挿入の生じるので、実用化のためには、処理速度の大幅な向上が望まれる。本報告では、二次元形状を輪郭線の特徴点の系列によって表現し、順序付けによる点パターンのマッチング・アルゴリズムを適用する。輪郭線の特徴点の順序関係や隣接の点との位置関係等を利用し、平均的に $O(p \cdot \log p)$ の複雑度のアルゴリズムを提案する。但し、 $p = \text{Max}(m, n)$ 、 m 、 n はそれぞれモデル物体と認識対象物体の輪郭線の特徴点の個数である。実験の結果、このアルゴリズムによって、マッチングの速度を大幅に高めることができた。

本報告で述べるアルゴリズムは、次の三つの段階からなる。

(1) 画像の前処理。この段階では、画像から物体の輪郭線を抽出し、それを特徴点の系列に変換する。

(2) 局所的な拘束を利用し、モデルと画像から抽出した輪郭線の間で互いにマッチできる連続した三つの点対を捜し出す。その後で、マッチングを行なう基礎となる基準線対を定義する。さらに、モデルと画像から抽出した輪郭線間の形状シフト数を定義する。これによって、基準線対の数を減らす。

(3) 選択された基準線対に対する位置関係を用いて、モデルと画像から抽出した輪郭線の特徴点を、順序付けする。この順序付けしたリストを用いて、両方の特徴点のマッチングを行ない、最小誤差のマッチングを捜し出す。

以下、第2、3と4節では、上述の三つの段階で用いた手法について述べる。第5節では、実験の結果を示し、第6節では、アルゴリズムの複雑度を分析する。

2. 画像の前処理

物体の二次元形状はその輪郭線によって記述できる。この輪郭線は多角形によって効率的に表現できる。カメラから入力された物体の画像を先ず二値化し、境界線を抽出する。その後、

Pavlidis と Horiwits の split and merge 法を採用し [8,9]、境界線を多角形で近似する。境界線の円弧の部分の円弧抽出アルゴリズム [10] によって抽出する。円弧の両端点と多角形の曲率の大きい頂点を輪郭線の特徴点とする。以上の処理によって、モデル物体 U と複合物体 V (以下、 U パターンと V パターンと呼ぶ) を次の二つのリストで表現することにする。

$$U = \{u_1, u_2, \dots, u_i, \dots, u_m\} \quad 1 \leq i \leq m$$

$$V = \{v_1, v_2, \dots, v_j, \dots, v_n\} \quad 1 \leq j \leq n$$

但し、 u_i と v_j はそれぞれ U パターンと V パターンの特徴点で、 m と n は U パターンと V パターンの特徴点の数である。

以後の処理のために、 U パターンと V パターンの特徴点を輪郭線に沿って、時計回り (或は反時計回り) に順序付けておく。この処理で得られた二つのリストを後の処理で利用する。

3. マッチングの誤差、基準線対と形状シフト数

画像から抽出される特徴点には、ノイズの影響や画像処理手法に起因する各種要因により、誤って抽出された特徴点が含まれたり、あるべき特徴点が出ないことがある。抽出された特徴点の位置にも誤差がある。従って、画像から抽出された U パターンと V パターンの特徴点をマッチさせる時、二つのパターンが全く同一であると判定するのではなく、パターン間の誤差が適当な尺度の下で、誤差値が最小となるものを見つけ出さなければならない。本節では、まず U パターンと V パターンの基準線対の概念を導入し、これによって、 U パターンと V パターン間の誤差の尺度を定義する。

定義1: 図1のように、 U パターンの有向線分 $\overline{u_{i-1}u_i}$ と V パターンの有向線分 $\overline{v_{j-1}v_j}$ が次の二つの条件を満足するとき、それらを U パターンと V パターンの基準線対 (L_i^U, L_j^V) とする。

$$(1) \frac{|\overline{u_{i-1}u_i}|}{|\overline{v_{j-1}v_j}|} = F \text{ とすると、} \\ \frac{(|\overline{u_{i-1}u_i}| - F \cdot |\overline{v_{j-1}v_j}|)}{|\overline{u_{i-1}u_i}|} < L t \\ \frac{(|\overline{u_{i-1}u_i}| - F \cdot |\overline{v_{j-1}v_j}|)}{|\overline{v_{j-1}v_j}|} < L t$$

但し、 $L t$ は長さのしきい値である。

(2) 角 $u_{i-1}u_iu_{i+1}$ と $v_{j-1}v_jv_{j+1}$ との差の絶対値があるしきい値 $A t$ より小さい。

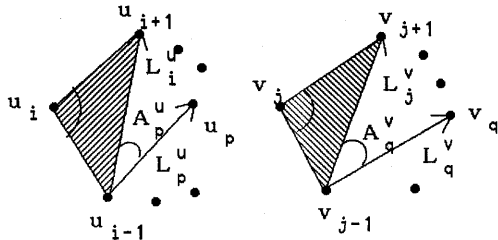


図1 基準線対の定義

$F = \frac{|\overline{u_{i-1}u_{i+1}}|}{|\overline{v_{j-1}v_{j+1}}|}$ をUパターンとVパターン間の比例係数と呼ぶ。また、三角形 $u_{i-1}u_iu_{i+1}$ と $v_{j-1}v_jv_{j+1}$ をUパターンとVパターンの基準三角形対と呼ぶ。

(1) の条件を満足し、かつ $|\text{角 } u_{i-1}u_iu_{i+1} - \pi|$ が十分大きければ、(2) の条件は次の条件 (2') と等価である。

(2') u_i と v_j はそれぞれ $\overline{u_{i-1}u_{i+1}}$ 、 $\overline{v_{j-1}v_{j+1}}$ に対して、同じ側にある。

現実的には、 $|\text{角 } u_{i-1}u_iu_{i+1} - \pi|$ の値が小さな点は画像処理の段階でマージされてしまうので、条件 (2) のテストの代りに、条件 (2') のテストをすれば、十分である。

定義2: (L_i^u , L_j^v) を基準線対とする。Uパターンの点 u_p とVパターンの点 v_q をマッチさせる時、このペアのマッチングによる誤差を次のように計算する (図1参考)。

u_p と u_{i-1} との距離を L_p^u 、線分 $\overline{u_{i-1}u_p}$ と $\overline{u_{i-1}u_{i+1}}$ 間の角度を A_p^u とする。同様に、 v_q と v_{j-1} との長さを L_q^v 、線分 $\overline{v_{j-1}v_q}$ と $\overline{v_{j-1}v_{j+1}}$ 間の角度を A_q^v とする。この時、基準線対 (L_i^u , L_j^v) に対して、 v_q の u_p に対するマッチングの誤差を次式で定義する:

$$\omega(u_p, v_q) = KA \cdot |A_p^u - A_q^v| + KL \cdot |L_p^u - F \cdot L_q^v|$$

但し、KAとKLはそれぞれ角度と長さの荷重因子である。

比例係数Fを定義する理由は点パターンのマッチングを大きさに依存しないようにするためであり、相互角度と点同士間の相互距離を使うのは、マッチングのアルゴリズムを点パターンの回転と移動に対処出来るようにするためである。

以上の定義から分かるように、基準線対を定

義する目的はマッチングの誤差を定義すると同時に、局所的な拘束を利用するためである。基準線対は後のマッチングの開始点となる。

このようにして定義される基準線対は最大 $m \cdot n$ 個である。しかし、の中には、多くの冗長性が含まれている。UパターンとVパターンの一つのマッチ状態を見つけ出すために、ただ一つの基準線対があれば、十分である。そこで、次に、形状シフト数を定義する。

定義3: 基準線対 (L_i^u , L_j^v) の形状シフト数SNを、次のように定義する。

$$SN = i - j;$$

$$IF(SN, L, T, 0) \quad SN = SN + n.$$

但し、nはVパターンの特徴点の個数である。

マッチング時には、Vパターンの輪郭線特徴点リストを固定し、Uパターンの特徴点リストを先頭の要素から一対一対応付けてゆく。これを一つの対応状態と呼ぶ。Uパターンの特徴点リストを循環リストと考えると、対応状態はn個である。この時、一つの対応状態に対して、以上の定義により、可能な基準線対の形状シフト数は同一となる。従って、形状シフト数の等しい基準線対は同一の対応状態から出ているので、一つだけをチェックすれば十分である。この時、チェックしなければならない基準線対は多くてもn個である。その上、基準線対を定義する時には、局所的な拘束を表わした二つの条件を利用するので、現実問題として、平均的な基準線対の数Kはnよりかなり小さく抑えられる。

4. 輪郭線特徴点のマッチング

輪郭線のマッチング手法として、順序付けによる高速な点パターンのマッチング・アルゴリズムを適用した。このマッチングの手法について、簡単に述べる。

まず、UパターンとVパターンの点のある基準線対に対するそれぞれの角度によって順序付けし、二つの順序付けしたリストとする。UパターンとVパターンのマッチングはこの二つの順序付けしたリスト上で行なう。順序付けしたリストは次のように作成する。

(L_i^u , L_j^v) をUパターンとVパターンの一つの基準線対とする。基準三角形以外の各点に対して、基準線との角度を求め、角度の升順に

順序付けしたリストを構成する。ある基準線対に対して、順序付けした二つの点集合をUリストとVリストとする。

$$U \text{ リスト} = (u_4, u_5, \dots, u_p, u_{p+1}, \dots, u_m) \\ 4 \leq p \leq m$$

$$V \text{ リスト} = (v_4, v_5, \dots, v_q, v_{q+1}, \dots, v_n) \\ 4 \leq q \leq n$$

今、 u_p と v_q が対応したとすると、次のマッチング点対候補は、 u_{p+1} と v_{q+1} 、或は v_{q-1} 以降のある範囲内にある。 u_{p+1} と基準線の角度を A_u 、許容角度誤差を At とすると、Vリスト内の点（角度を A_v とする）の中で、 $|A_u - A_v| < At$ を満足する点のみが、 u_{p+1} と対応する可能性がある。従って、探索の範囲がかなり限定出来る。現実には、最大許容角度誤差 At に対して、Uリストの各要素と対応可能なVリストの要素数はある定数以下である。従って、UとVのマッチングはUリストの要素数の線形時間内で完了する。付加的なコストは両リストの順序付けコストだけである。

u_p と v_q がマッチングできるためには、次の三つの条件が必要である（図1参考）。

条件1： u_p の基準線 L_p^u に対する角度を A_p^u 、 v_q の基準線 L_q^v に対する角度を A_q^v とすれば、 $Ae = |A_p^u - A_q^v|$ はある最大許容角度しきい値 At より小さい。

条件2： u_p と基準線 L_p^u の起点との間の長さを L_p^u 、 v_q と基準線 L_q^v の起点との間の長さに比例因子 F を掛けて、その値を L_q^v とすれば、相対長さ誤差 $Le = |L_p^u - L_q^v| / L_p^u$ はある最大許容長さしきい値 Lt より小さい。

条件3： u_p と v_q はそれぞれ隣の両点 u_{p-1} 、 u_{p+1} と v_{q-1} 、 v_{q+1} との位置関係が基準線対を定義する時に同じ条件を満足しなければならない。

前に述べたように、画像から特徴点を抽出する時には、特徴点の誤抽出や脱落が起こる。よって、Uパターンのある点がVパターンの点にマッチできないことがある（Vパターンの場合も同様）。これに対応するために、空の符号 Λ を導入する。即ち、対応する点が存在しないときには、 (u_p, Λ) や (Λ, v_q) などの符号対を導入する。この場合、誤差を $KA \cdot At + KL \cdot Le$ とする。但し、 At は最大許容角度しきい値で、 Le は u_p （或は v_q ）と基準線の起点の間の長さ（ v_q に対しては、 F を掛けた後の長さ）である。

このような処理を行なう目的は、対応点が存

在する方の誤差が対応点の存在しない方の誤差より大きくならないようにするためである。

上述の各段階をまとめて、アルゴリズムの全体の流れ図を図2に示す。

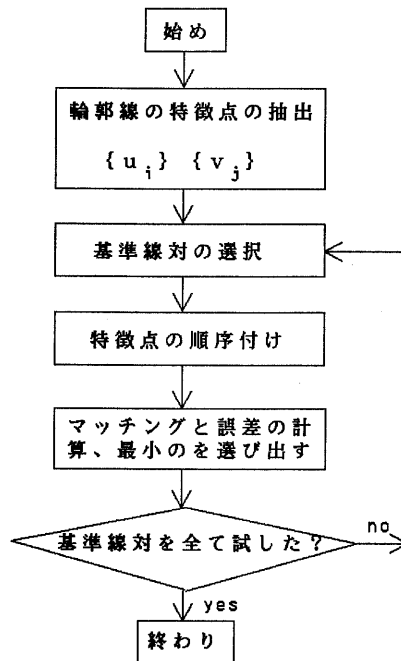


図2 アルゴリズム全体の流れ図

5. 実験結果

NEC製スーパーミニコンピュータMS-190上でいろいろなデータを用いて、本アルゴリズムを実行した。以下、各々の実験及び実験結果について述べる。

実験1：上述のアルゴリズムは輪郭線のマッチングに相当であるか否かをテストするために、次の実験を行なった。二次元乱数発生プログラムによって、ランダムに n 個の点を発生し、これらの点をVパターンの輪郭線特徴点のデータとする。 n 点の内から任意に m ($m \leq n$) 個の点を選び出し、一定角度回転させた後に、各点を原点からの動径を、方位は変化させずに伸縮させて、移動する。ただし、動径の伸縮率に一定のランダムノイズを加える。このようにして求めた m 個の点をUパターンの輪郭線の特徴点

とする。

即ち、乱数発生プログラムによって作り出したVパターンの点 V_p の座標値を (X, Y) とする。次式によりUパターンの対応点 U_p の座標値 (X', Y') を計算する。

$$X' = (X \cdot \cos \beta - Y \cdot \sin \beta) \cdot (f + c)$$

$$Y' = (X \cdot \sin \beta + Y \cdot \cos \beta) \cdot (f + c)$$

ただし、 β は回転角で、 f は伸縮係数、 c は標準偏差値が σ 、平均が0の正規乱数である。

実験では、2次元乱数発生プログラムにより、10組づつの異なる特徴点系列を8種類発生した。実験中で採用した長さのしきい値 L_t は相対長さ誤差15%とし、角度誤差しきい値 A_t は 10° とした。ノイズに当たる正規乱数の標準偏差は0.025とした。これらのデータに対して、マッチングを実行させたところ全て正しい結果を得た。処理に要したCPU時間の平均値と基準線対の数の平均を表1に示す。

この表から分かるように、この程度の点数に対しては、このアルゴリズムはかなり高速である。また、基準線対の個数 k は n より相当小さいことも検証された。

表1 平均CPU時間(ミリ秒)と平均基準線対の数

U図の点数	30	30	50	50
V図の点数	30	40	50	60
CPU時間ms	86	113	318	380
平均基準線対数	2.1	2.1	5.7	3.8
U図の点数	70	70	90	100
V図の点数	70	80	90	100
CPU時間ms	810	835	2335	4042
平均基準線対数	7.0	5.3	12.7	19.4

実験2: 図3の(b)はモデルパターン、(a)は(b)を含むパターンで、輪郭線の特徴点の数はそれぞれ16点と27点である。(b)と(a)の対応付けの結果は図4のようになった。対応付けられた点に丸のマークを、マッチングのできなかった点に三角形のマークを付けている(以下の実験の結果も同様)。マッチングのCPU時間は42ミリ秒で、基準線対の数は2対である。

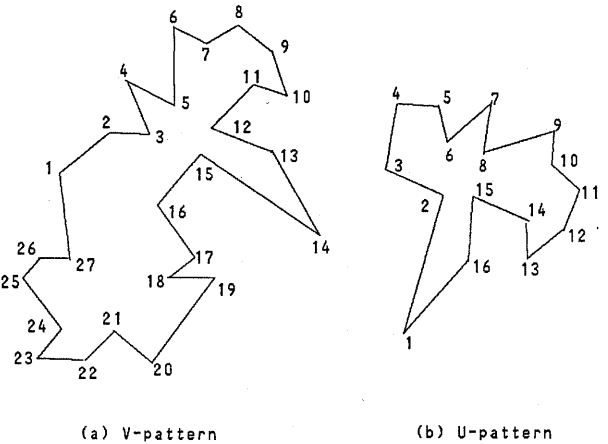


図3 二次元形状のマッチング(実験2)

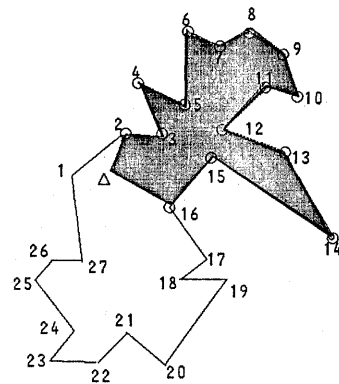


図4 実験2のマッチング結果

実験3: 図5の(c)は(a)と(b)の合成パターンである。ただし、(c)では、(a)のパターンが拡大されている。実験の結果、図6のようにマッチングできた。この実験のデータは文献[3]から引用した。実験環境が異なるので正確には比較できないが、我々の方が効率がよいと考えられる。表2は本アルゴリズムと文献[3]の所用CPU時間及び実験の環境の比較表である。

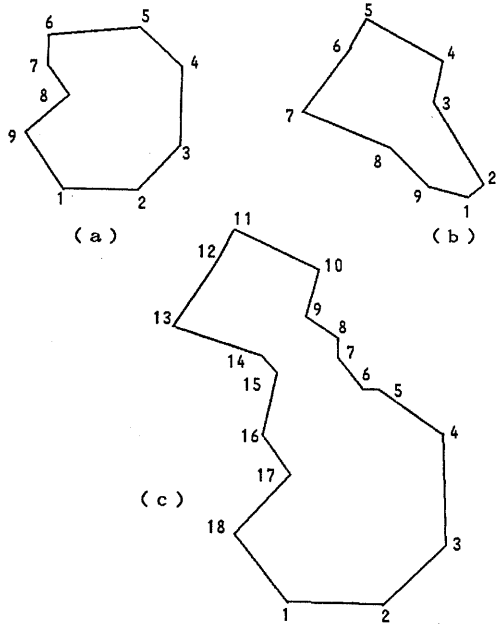
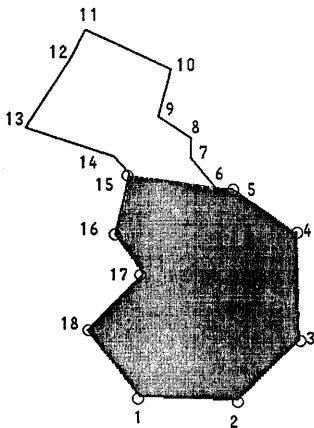


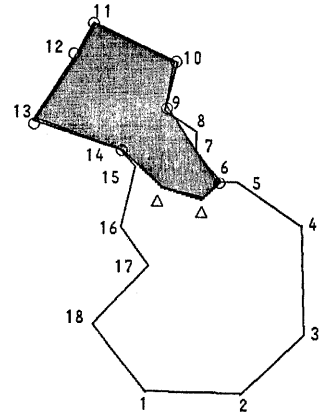
図5 二次元形状のマッチング (実験3)

表2 本アルゴリズムと文献[3]の比較

	マッチングCPU時間		利用コンピュータ
	aとc	bとc	
本手法	25 ms	23 ms	MS-190, 4.0MPS
文献[3]	52.79 s	42.90 s	PDP-10, 1.8MPS



(a) : aとcのマッチング結果



(b) : bとcのマッチング結果

図6 実験3の結果

実験4: 図7は一つのジグソーパズルのような問題である。(a)と(b)と(c)の物体が重なって、(d)になっている。アルゴリズムは、パターン(d)の中から部品のパターン(a)、(b)、(c)を見つけ出した。図8は(a)、(b)、(c)の

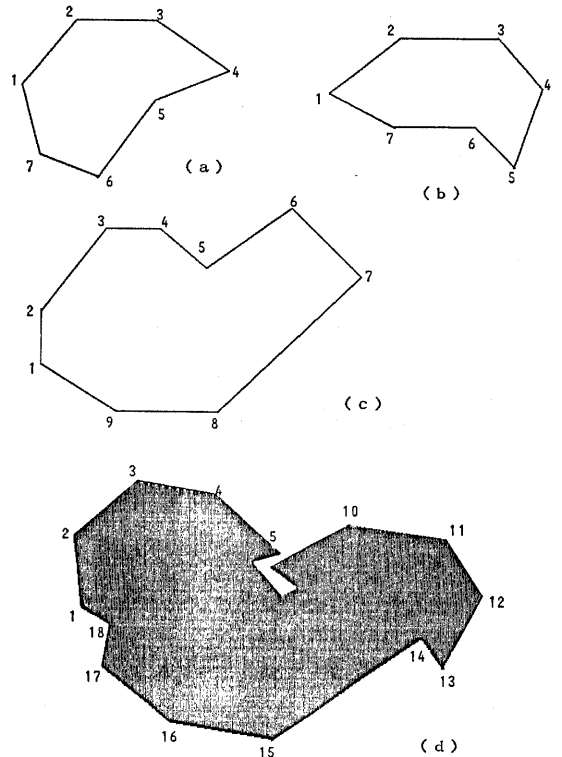


図7 ジグソーパズル問題 (実験4)

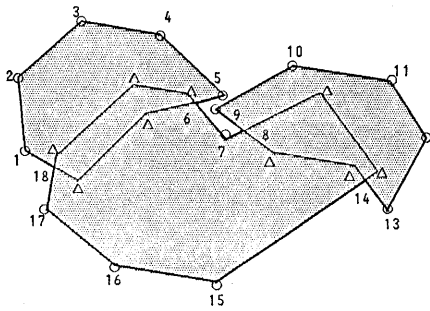


図8 実験4の結果

それぞれと (d) との間の対応付けの結果である。(a) と (d) のマッチング時間は39ミリ秒、(b) と (d) のは13ミリ秒、(c) と (d) のは14ミリ秒で、合計時間は66ミリ秒である。文献[3]のアルゴリズムでは152.7秒かかっている。

実験5: 図9の (a)、(b) はそれぞれ部品の輪郭線の多角形近似で、(c) は (a) と (b) が重なっている複合シーンの輪郭線の多角形近似である。実験の結果、(a) と (c) は図10のようにマッチングされた。(a) の頂点数は21点、(c) の頂点数は23点、CPU時間は62ミリ秒であった。

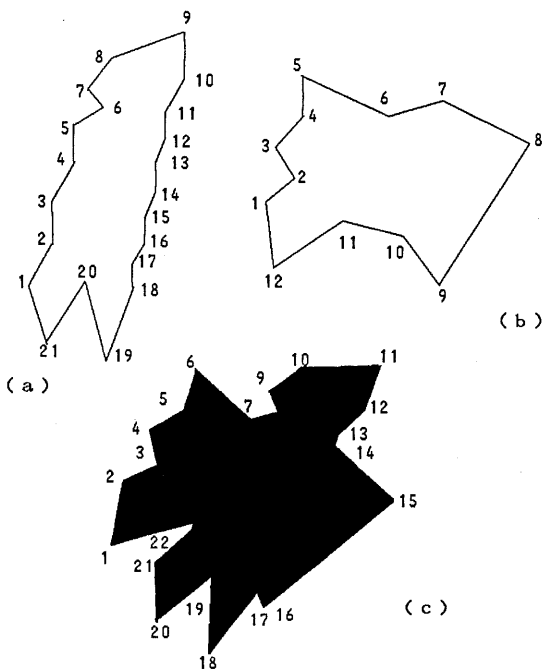


図9 重なっている物体の認識(実験5)

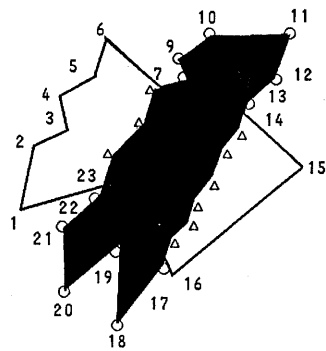


図10 実験5の結果

6. アルゴリズムの複雑度の分析

本節では、アルゴリズムの複雑度を簡単に分析する。本アルゴリズムの複雑度は主として、基準線対の数と一つの基準線対の中で実行される操作の計算量の積で決まる。本アルゴリズムでは、一つの基準線対に対して、ソーティングとマッチングを行なっている。マッチングはUリストの要素数の線形時間内で終了する。ソーティングはU、Vリストそれぞれ $m \cdot \log m$ と $n \cdot \log n$ の時間で出来る。ただし、 m と n はUパターンとVパターンの点の数である。従って、一つの基準線対に対する処理時間は $p \cdot \log p$ (ただし、 $p = \text{Max}(m, n)$) のオーダーとなる。基準線対の数は、前に述べたように、現実の問題に対しては、局所的な拘束条件を利用するので、平均的にある定数 k ($k < n$) 以内である。よって、全体のアルゴリズムの複雑度は $O(p \cdot \log p)$ となる。

7. むすび

重なったり或は互いに接触したりする物体の認識の高速化に対して、本報告では、局所的な拘束条件を利用し、輪郭線の特徴点のマッチングに基づく手法を提案した。モデル物体と認識対象物体の輪郭線を多角形で近似して、特徴点の系列とする。その後、局所的な位置拘束関係を利用し、基準線対を定義した。基準線対を定義する目的は、マッチングの誤差を定義する

ただめだけではなく、後のマッチングの出発点ともなる。さらに、マッチ状態を表現するために、形状シフト数の概念を導入した。これによって、探索される基準線対の数を大幅に減少させた。輪郭線の特徴点のマッチングは点の順序付けによって、大幅に速められたことを実験によって確認した。

今後、各種の画像データに対して、輪郭線の特徴点を抽出する手法に関するアルゴリズムを検討してゆく。

参考文献：

[1] R. T. Chin and C. R. Dyer: Model-based recognition in robot vision, ACM Comput. Surveys, vol. 18-1, pp. 67-108, 1986.

[2] L. S. Davis: Shape matching using relaxation techniques, IEEE PAMI, Vol. PAMI-1, No. 1, pp. 60-72, 1979.

[3] B. Bhanu and O. D. Faugeras: Shape matching of two-dimensional objects, IEEE PAMI, Vol. PAMI-6, No. 2, pp. 137-156, 1984.

[4] N. Ayache and O. D. Faugeras: HYPER: A new approach for the recognition and positioning of two-dimensional objects, IEEE PAMI, Vol. PAMI-8, No. 1, pp. 44-54, 1986.

[5] M. W. Koch and R. L. Kashyap: Using polygons to recognize and locate partially occluded objects, IEEE PAMI, Vol. PAMI-9, No. 4, pp. 483-494, 1987.

[6] J. L. Turney, T. N. Mudge and R. A. Volz: Recognizing partially occluded parts, IEEE PAMI, Vol. PAMI-7, No. 4, pp. 410-421.

[7] W. Eric L. Grimson and Tomas Lozano-perez: Localizing overlapping parts by searching through interpretation tree, IEEE PAMI, Vol. PAMI-9, No. 4, pp. 469-482, 1987.

[8] T. Pavlidis: Piecewise approximation of plane curve, Proc. 1st. IJCP, pp. 396-405, 1973.

[9] T. Pavlidis and S. Horowitz: Segmentation of plane curves, IEEE Comput., Vol. c-23, pp. 860-870, 1974.

[10] T. Pavlidis: Algorithms for graphics and image processing, Computer Science Press, 1982.