

## 論 文

アフィン変換に不変な任意図形検出法として拡張された  
一般化ハフ変換木村 彰男<sup>†</sup> 渡辺 孝志<sup>†</sup>Generalized Hough Transform to be Extended as an Affine-Invariant  
Detector of Arbitrary ShapesAkio KIMURA<sup>†</sup> and Takashi WATANABE<sup>†</sup>

あらまし 画像から任意の曲線図形をアフィン変換に不変な形で検出する問題は、画像認識において基本的で重要な研究課題である。従来、そのような検出を意図した手法がいくつか提案されているが、それらはいずれも (1) 他物体との重なりによって検出対象図形の輪郭線の一部が隠ぺいされている, (2) 雑音によって輪郭線に途切れが生じている, などの状況にはうまく対処できない。そこで本論文では, このような場合にも対処可能な新しい図形検出法を提案する。まずはじめに, 任意曲線を含んだ不連続輪郭線画像から安定して接線情報を抽出するために, ハフ変換を用いた円弧当てはめによる接線情報抽出法を新たに提案する。続いて, “アフィン変換下においては直線の平行性が保存される” という性質に着目し, 図形の各輪郭点位置での接線情報を積極的に利用した形で Ballard の一般化ハフ変換 [1] をアフィン変換に不変な図形検出法として拡張する。提案手法の有効性を検証するために行った評価実験では, 良好な検出結果が得られた。

キーワード 図形認識, 一般化ハフ変換, アフィン変換, 接線情報, コンピュータビジョン

## 1. ま え が き

入力画像から任意形状の 2 次元図形をアフィン変換に不変な形で検出する問題は, 画像認識における基本的で重要な研究課題である。例えば, 擬似中心射影 (paraperspective projection) においては, あるシーンを異なったカメラ視点で見たときの各々の投影画像面は互いにアフィン変換の関係にあるので [2], アフィン変換に不変な 2 次元図形認識法を確立することは, コンピュータビジョンのような 3 次元画像解析を行う上でも極めて重要な研究課題であるといえる。

従来, このようなアフィン変換不変な形での図形検出を意図した手法としては, フーリエ記述子を利用したもの [3], 周波数領域における不変特徴を利用したもの [4], ニューラルネットワークを利用したもの [5], 曲線モーメントに基づくもの [6], 幾何学的ハッシングを用いるもの [7]~[9] など, いくつか報告されており,

ある限られた条件下においては効率的な検出が可能であることも示されている。しかしながら, (1) 他物体との重なりによって検出対象図形の輪郭線の一部が隠ぺいされている, (2) 雑音等によって輪郭線に途切れが多数生じている, (3) 検出対象図形が一般的な曲線図形である, などのより一般的な状況においても効果的に対処できる手法はまだない。例えば, 文献 [3], [6] では, 検出対象図形に対して輪郭線の連続性を要求しているのだから, (2) に対処できない (したがって (1) にもうまく対処できない)。文献 [4] では, 単純でない図形の切出しが難しく, (1) にうまく対処できない。文献 [5] では, 変換前の図形と変換後の図形に対し, 対応する点を手動で指定する必要がある。文献 [7]~[9] の手法は雑音には比較的強いものの, そのままでは処理時間が膨大にかかるため, 実際には検出対象を有限個の特徴点や特徴線分でうまく表現できる図形に限定する必要がある。したがって, 真に実利用を考えるならば, そのような一般的な状況にも対処できる新たな図形検出法の確立が望まれる。

一方, 認識問題を一段下の相似変換不変な任意図形

<sup>†</sup> 岩手大学工学部情報システム工学科, 盛岡市  
Department of Computer and Information Science, Iwate  
University, 4-3-5 Ueda, Morioka-shi, 020-8551 Japan

検出に限定すれば、上に述べたような場合においても対処可能な図形検出法として一般化ハフ変換 (GHT [1]) が提案されており、最近その実用化手法として高速 GHT [14] が開発されている。そこで本論文では、この GHT のロバスト性に加え、“直線の平行性がアフィン変換下においても保存される”という性質に着目し、各輪郭点 (エッジ点) 位置での接線情報を積極的に利用するような形で GHT を拡張した、アフィン変換に不変な任意図形検出法を新たに提案する。これにより、提案手法は、従来法で検出が困難と考えられる状況下においてもロバストな図形検出が可能となっている。ただしそのためには、各輪郭点位置で接線情報を精度良く安定に求めることが必要となる。

そこで本論文では、まず 2 値の輪郭線画像から接線情報を安定に抽出する手法として、ハフ変換を用いた円弧当てはめによる接線情報抽出法を新たに提案する。この手法では、比較的曲率の大きい輪郭点においても安定して接線情報を抽出することが可能であり、更に、輪郭線が不連続となっている場合にも適用可能となっている。続いて、これらの接線情報を積極的に利用する形で GHT を拡張して、アフィン変換に不変な図形検出法を構成する。そして、評価実験によってその有効性を示す。

## 2. 輪郭線画像からの接線情報の抽出

本論文で提案するアフィン変換不変な図形検出法では、検出対象物体の各輪郭点位置での接線情報を必要とする。そこで本節では、輪郭線画像からの接線情報抽出問題について論じる。

### 2.1 従来の接線情報抽出法とその問題点

入力画像が濃淡画像である場合には、例えば微分処理を施してこう配情報を取得し、それに直交する方向として接線情報を得ることが可能である。しかしこの方法では、必ずしも信頼性の高い接線情報が得られない。なぜなら、微分処理自体が局所的処理であり、多分に雑音等の影響を受けやすいからである。これに対し、通常の微分処理よりも広い範囲の画素が考慮される手法として、Canny の最適フィルタ [10] や ISEF [11] などの優れた手法が知られているが、これらは主に濃淡画像から安定して輪郭点画素 (エッジ点) を抽出するために開発されており、得られるこう配情報の精度という点では十分でない。したがって、輪郭線画像のみから接線情報を高精度に抽出する手法を開発することは、研究上の意義が大きいと考えられる。

このような立場から、渡辺ら [12] はハフ変換と共分散行列を組み合わせた接線情報抽出法を提案している。筆者らの知る限りでは、輪郭線画像のみから接線情報を抽出する手法は彼らのものが唯一であるが、この手法においても「曲率の小さくないエッジ点位置において得られる接線情報は十分でない」という問題点が残されている。彼らの手法では、曲率の大きいエッジ点位置でも強制的に直線を当てはめようとするために、ハフ変換を施した際の投票値が集積せず、結果として接線情報を得ること自体が困難、あるいは得られる接線情報が信頼性の低いものになってしまう (このことは後に実験結果で示す)。一般に、任意形状の図形は曲率が小さくない点を多数含んでおり、しかも、そのような点は図形自身を表現する特徴的な輪郭点としてもとらえることができるため、任意図形検出という立場から考えた場合には、そのような点でも安定して接線情報を抽出できる手法が強く求められる。

そこで本論文では、この問題に対処するために、注目画素周辺のエッジ点列に近似的に円弧を当てはめ、その円弧から注目画素位置での接線情報を求める、という手法を新たに提案する。以下、提案手法の詳細について説明するが、ここでの処理対象画像は適当な方法 (例えば、Canny フィルタ) で得られた 2 値の細線化された輪郭線画像とし、そのエッジ点は必ずしも連結している必要はないものとする。

### 2.2 ハフ変換を用いた円弧当てはめによる接線情報抽出法

今、注目画素の中心を原点  $(0, 0)$  と考えると、これを通る円の方程式は

$$x^2 + y^2 - 2gx - 2fy = 0 \quad (1)$$

で表せる。円の中心は  $(g, f)$  である。このとき、注目画素における接線のこう配  $\theta$  は、式 (1) の微分から

$$\tan \theta = \left. \frac{dy}{dx} \right|_{(x,y)=(0,0)} = -g/f \quad (2)$$

と計算できる。したがって、この円が注目画素のほかにも 2 点  $(x_1, y_1)$ ,  $(x_2, y_2)$  を通るとすると、式 (1) を連立させて式 (2) に代入することで

$$\theta = \arctan \frac{(x_2^2 + y_2^2)y_1 - (x_1^2 + y_1^2)y_2}{(x_2^2 + y_2^2)x_1 - (x_1^2 + y_1^2)x_2} \quad (3)$$

を得る。これを用いると、以下に示したハフ変換的な手順によって、注目画素位置での接線方向を安定に求めることが可能となる。ただし、この接線情報の抽出

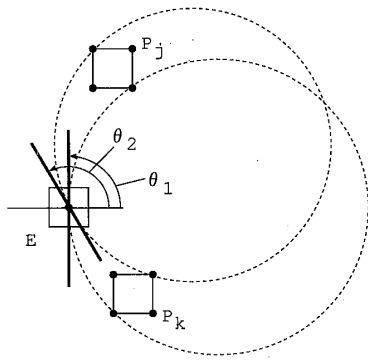


図1 画素の広がり考慮した投票  
Fig. 1 Voting pattern considering pixel's extent.

処理は局所的な処理となるので、本手法では標本化誤差の影響も考慮して処理アルゴリズムを構成する。

[処理手順]

(手順1) 注目エッジ点  $E = (0, 0)$  を中心とする適当なサイズ (例えば,  $13 \times 13$  画素) の正方形窓を考える。窓内に含まれる  $E$  を除いたエッジ点の集合を  $\{P_i(x_i, y_i)\}$  とする。また、投票空間として、角度  $\theta$  の区間  $[0^\circ, 180^\circ)$  を適当な幅  $\Delta\theta$  で量子化した1次元配列を用意する。

(手順2)  $\{P_i\}$  から2点  $P_j, P_k$  を選び、画素  $E, P_j, P_k$  をすべて通る円を考える。ただしここでは、線分  $\overline{EP_j}$  と  $\overline{EP_k}$  のなす角度が指定角度 (例えば,  $60^\circ$ ) 以下となるような2点は選ばない<sup>(注1)</sup> ものとする。この円の  $E$  における接線方向は式 (3) から求められるが、各画素は広がりをもっているため、ここでは3点  $E, P_j, P_k$  を通る可能性のある円として、図1に示したものを考えることにする。すなわち、画素  $P_j, P_k$  の四隅の点をそれぞれ  $P_{jm}, P_{kn}$  ( $m, n = 1, 2, 3, 4$ ) とし、3点  $E, P_{jm}, P_{kn}$  を通る円 (計16通り) に対し、式 (3) によって接線方向  $\theta_{mn}$  を求める。それらの16個の  $\theta_{mn}$  の最小値を  $\theta_1$ 、最大値を  $\theta_2$  とするとき、 $\theta_1 \sim \theta_2$  の範囲に対応する配列セルすべてに対して値1を加える (これを投票と呼ぶ)。

(手順3)  $(P_j, P_k)$  の可能なすべての組合せに対して投票を行い、角度  $\theta$  の投票空間において、適当なしきい値を超えた最大投票値をもつセルに対応する角度  $\theta$  を注目エッジ点  $E$  での接線情報とする。なお、投票空間で最大投票値が連続して並ぶ場合は、それらの中央に位置するセルを  $\theta$  として選ぶものとする。 □

実際に投票を行うのは角度  $\theta$  に対してのみであるので、処理を高速に実行することができる。更に、 $\theta$  は

式 (3) によって計算されるので、仮に窓内のエッジ点群が直線的に並んでいたとしても何ら問題なく計算が可能である。

### 2.3 評価実験

提案手法と渡辺らの方法 [12] に対し、それらの接線情報抽出性能を比較するための評価実験を試みた。本実験では、異なった曲率をもつ輪郭点が多く含まれる図形として楕円を選び、その短軸半径  $\beta$  を10 (画素) に固定したまま長軸半径  $\alpha$  を10 ~ 30 (画素) の間で変化させて種々の楕円を発生させ、それらに対して両手法を適用して得られる接線情報の精度を比較した。楕円の中心座標は  $(128, 128)$  であり、回転は加えていない。渡辺らの方法においては、窓サイズを  $9 \times 9$  画素に固定し、検出しきい値が6と3のそれぞれの場合について行った (理由は後述する)。また、提案手法は円弧当てはめであり、その窓は直線当てはめの場合よりも広げて曲線の曲がり部分も含め得るようになる必要がある。そのため、ここでは提案手法の窓サイズは  $13 \times 13$  画素とした。検出しきい値は25とした。なお、どちらの手法においても投票空間の量子化幅は  $\Delta\theta = 1^\circ$  とした。

図2は、それぞれの手法において接線情報を抽出できた割合 (接線こう配を計算できた輪郭点数 ÷ 対象楕円の総輪郭点数  $\times 100[\%]$ ) を表し、図3及び図4は、計算できた接線こう配 (角度) の平均検出誤差 (絶対値) と最大検出誤差をそれぞれ表している。これらの結果から、渡辺らの手法では、検出しきい値がある程度大きい場合 (実験では6を使用) には精度の高い抽出が可能であることがわかる。特に、 $\alpha$  が大きくなってくると、局所的には直線とみなせる場所が増えてくるため、平均誤差が小さくなっている。しかしその反面、図2に示すように、接線情報を抽出できる輪郭点の割合は50%前後、とかなり限定されてしまう問題がある。この原因は、既に指摘したように、曲率の小さな点の周辺にも直線を当てはめようとしているためである。検出しきい値を下げれば計算可能点を増やすことも可能であるが (実験ではしきい値3の場合を示した)、その場合には、得られる接線情報の誤差が大きくなるという別の問題が発生してしまう。

これに対して提案手法は、いずれの楕円に適用した場合でもほぼ100%に近い割合、すなわち、ほぼすべ

(注1): これは、 $P_j$  と  $P_k$  を、注目画素  $E$  を挟んだ両側から1点ずつ選択するための措置である。これにより、標本化誤差の影響が軽減され、誤った角度投票が抑えられる。

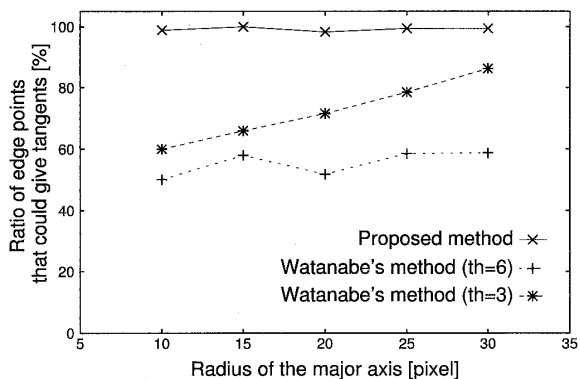


図 2 楕円の接線情報が抽出できた輪郭点数の割合  
Fig. 2 Ratio of edge points that could give tangents for ellipses.

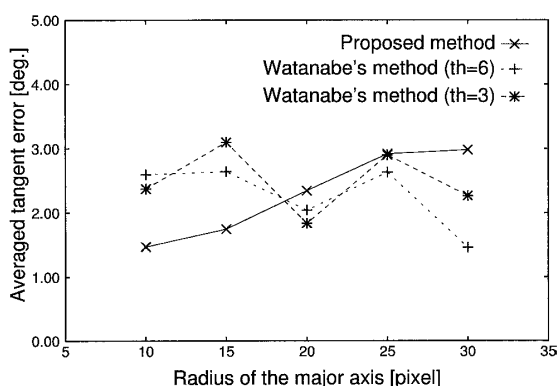


図 3 楕円の接線情報の検出精度 (平均誤差)  
Fig. 3 Accuracy of extracted tangents for ellipses (averaged tangent error).

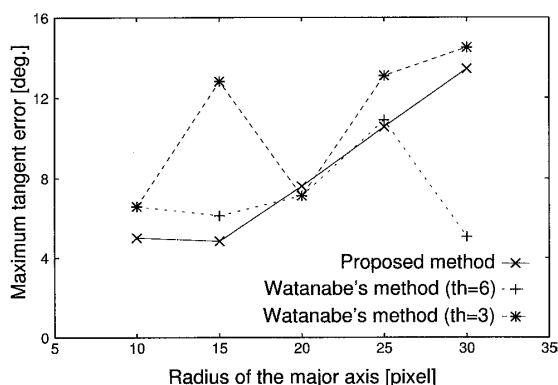


図 4 楕円の接線情報の検出精度 (最大誤差)  
Fig. 4 Accuracy of extracted tangents for ellipses (maximum tangent error).

での輪郭点において接線情報が抽出できている。平均検出誤差も、 $\alpha$  の大小にかかわらず  $3^\circ$  以内に抑えられており、更に、 $\alpha$  が 10 ~ 15 画素と小さい楕円に対しては渡辺らの手法を用いた場合よりも誤差が減少し

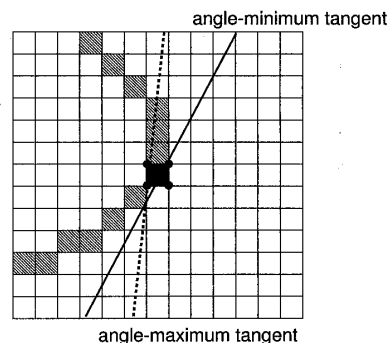


図 5 誤差が大きくなる画素パターン例  
Fig. 5 An example of edge pixel pattern with large tangent error.

ている。また、最大誤差に関しても、 $\alpha = 30$  の場合を除いて渡辺らの手法と同等、あるいはより減少している、という良好な結果が得られた。すなわち、提案手法は接線情報が抽出できる輪郭点の割合を大幅に高めたにもかかわらず、平均誤差と最大誤差に関しては渡辺らの方法と同等なレベルに抑えることができた、といえる。

図 4 における  $\alpha = 30$  の楕円で提案手法の検出誤差が最大となる画素パターン例を図 5 に示した。この図のようにもととのパターン上で曲率が急激に変化する点においては、標準化誤差の影響がかなり大きい。得られる接線情報は非常に不安定となる。例えば、注目画素 (黒い部分) の四隅の点をそれぞれ原点と考えて提案手法を適用すると、求められた四つの角度間には最大  $14^\circ$  の違いが生じてしまう (図中では、算出角度が最小・最大となる場合を示した)。したがって、最大誤差をより減少させ、安定した接線情報だけ抽出することを考えるならば、注目画素の四隅も考えて提案手法を適用し、その結果算出された五つの角度の変動範囲がしきい値  $\theta_e$  以上になったときに得られた解を棄却する、というように手順を変更するとよい。しかしながら、このような変更を加えた場合には、接線情報を抽出できる輪郭点数が減少し、処理時間も増大してしまう<sup>(注2)</sup>。そこで今回は、提案手法をそのままの形で使用するものとした。実際、図 5 のような輪郭点は、全体の輪郭点数に比べればごく少数と考えられ、後述するアフィン変換不変な図形検出法自体もハフ変換を基本として構成されているので、このような

(注2) : 例えば、 $\theta_e = 12^\circ$  として  $\alpha = 30$  の楕円にこの方法を適用すると、最大誤差は  $13.5^\circ \rightarrow 9.9^\circ$  に減少し、接線情報が抽出できた輪郭点数の割合は  $100\% \rightarrow 87.5\%$  に減少する。

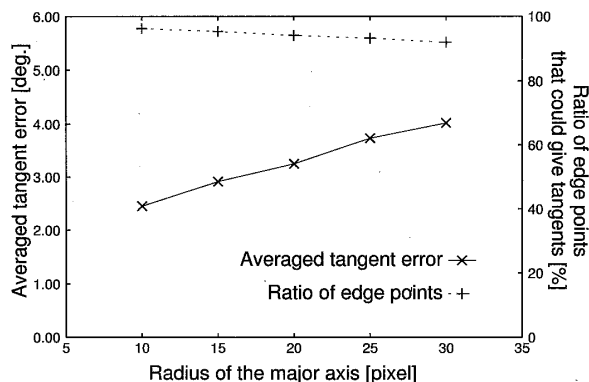


図6 楕円の接線情報の検出精度 (雑音がある場合)  
Fig.6 Accuracy of extracted tangents for ellipses with noises.

誤差は他の (接線情報が安定に得られる) 点からの投票によって全く相殺されてしまうと考えられる。

また、雑音環境下における提案手法の接線情報抽出性能を評価するために、先の楕円に一樣乱数で窓サイズの約5%の雑音画素を論理的に加え、各輪郭点位置でそれぞれ20回分の平均検出誤差 (絶対値) を算出した上で楕円全体における総平均誤差を求めた。その結果を図6に示す。なお同図には、接線情報が抽出できた輪郭点の割合 (20回分の平均値) も重ね書きしてある。雑音のない場合と比べると、接線情報を抽出できた輪郭点の割合が若干低下しているものの、いずれの楕円に対しても誤差は  $+1^\circ$  以内の増加に抑えられており、安定した検出力を保持していることがわかる。また、最大誤差については、付加した雑音画素が (例えば別の円弧が構成されるような) 特殊な配置になってしまうような場合を除けば、 $\alpha = 20$  以下の楕円では  $9^\circ$  以内、 $\alpha = 30$  の楕円であっても  $14^\circ$  以内に抑えられており、雑音のない場合とそれほど変わらない性能を保持できていることが確認された。

以上の結果より、提案手法は、より広範囲の画像に対処できる接線情報抽出法として構成されているといえる。

### 3. アフィン変換に不変な任意図形検出法

2次元のアフィン変換は六つのパラメータで記述される (後述) ため、直接的にハフ変換的手法を適用しようとするとその探索空間は6次元となり、現実的には実行不可能である。そこで、本論文では、「直線の平行性保存」というアフィン変換の性質に着目し、まず最初に検出対象物体の位置座標を求め、その後で残り

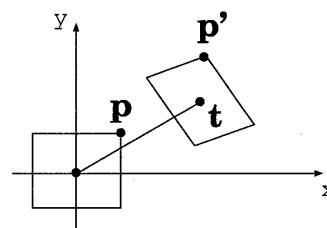


図7 アフィン変換  
Fig.7 Affine transformation.

の四つのパラメータを求める、というように段階的にハフ変換を適用することを考える。それでも第2段階のハフ変換では4次元空間の探索が必要となるため、本手法では処理の効率化を図るために、パラメータ空間の分解能を適応的に変化させて投票を行う適応的ハフ変換 (AHT [13]) を使用する。

以下、提案手法であるアフィン変換に不変な図形検出法について説明するが、処理対象とする画像は2.と同様な2値の輪郭線画像とし、各エッジ点での接線情報は2.で述べた方法によって既に求められているものとする。

#### 3.1 アフィン変換の定義

二つの2次元直交座標系  $\Sigma, \Sigma'$  を考える。今、1点  $P$  の  $\Sigma, \Sigma'$  に関する座標をそれぞれ  $\mathbf{p} = (x, y)^t$ ,  $\mathbf{p}' = (x', y')^t$  としたとき、

$$\mathbf{p}' = A\mathbf{p} + \mathbf{t} \quad (4)$$

で表される  $\mathbf{p}$  から  $\mathbf{p}'$  への変換をアフィン変換という。ここで、 $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ ,  $\mathbf{t} = (t_x, t_y)^t$  であり、 $ad - bc \neq 0$  とする (図7)。

式(4)より、アフィン変換はパラメータ  $\{a, b, c, d\}$  による線形変換とパラメータ  $\{t_x, t_y\}$  による平行移動を組み合わせた計六つのパラメータによる変換となっている。したがって、変換前と変換後の対応点の対が3個 ( $\mathbf{p}_i \leftrightarrow \mathbf{p}'_i, i = 1, 2, 3$ ) 定まれば、アフィン変換パラメータ  $\{a, b, c, d, t_x, t_y\}$  を唯一に定めることができる。

アフィン変換には次の重要な性質

- 平行な直線は平行な直線に移る (平行性保存)
- 図形の重心は変換後の図形の重心に移る

がある。以下では、これらの性質を積極的に利用した図形検出法を構成する。

#### 3.2 アフィン変換不変な図形検出手順

今、検出対象となる2次元図形をテンプレート

表 1 R 表の構成  
Table 1 Contents of R-table.

| 接線方向       | $\mathbf{p}_1$ の座標 | $\mathbf{p}_2$ の座標 | $\mathbf{p}_3, \mathbf{p}_3', \dots$ の座標 | $\mathbf{p}_3, \mathbf{p}_3', \dots$ での接線方向 |
|------------|--------------------|--------------------|--|---|
| $\Theta_i$ | $(X_1, Y_1)_i$     | $(X_2, Y_2)_i$     | $(X_3, Y_3)_i, (X_3', Y_3')_i, \dots$    | $\Gamma_i, \Gamma'_i, \dots$                |
| ...        | ...                | ...                | ...                                      | ...   |

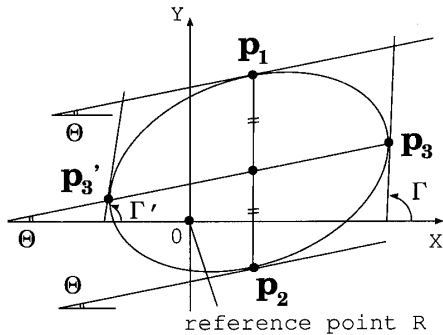


図 8 テンプレート T の記述  
Fig. 8 Description of template T.

$T$  と呼び、 $T$  を構成しているエッジ点を  $\{(X_i, Y_i), i = 1, 2, \dots\}$  で表すものとする。ただし、各  $(X_i, Y_i)$  は、 $T$  の参照点  $R$  (任意にとってよい) を原点とした座標値で表されているとする。

3.2.1 R 表, 及び E 表の生成

GHT [1] と同様の考え方で、まずはテンプレート  $T$  を表す参照表 (R 表) (表 1) を作成する。ただし、ここではエッジ点の接線情報を利用して次の手順で作成するものとする。

[R 表作成手順] (図 8 参照)

(手順 1)  $T$  を構成するエッジ点群を、その接線方向  $\Theta$  でソーティングし、同じ  $\Theta$  をもつエッジ点同士  $\mathbf{p}_1 = (X_1, Y_1)^t, \mathbf{p}_2 = (X_2, Y_2)^t$  を組み合わせる。

(手順 2) 線分  $\overline{\mathbf{p}_1\mathbf{p}_2}$  の中点から、 $\Theta$  の方向に直線を引き、それが  $T$  と交差する点を求め、それぞれ  $\mathbf{p}_3 = (X_3, Y_3)^t, \mathbf{p}_3' = (X_3', Y_3')^t, \dots$  とする。点  $\mathbf{p}_3, \mathbf{p}_3', \dots$  での接線方向をそれぞれ  $\Gamma, \Gamma', \dots$  とする。

(手順 3)  $\Theta$  の大小順に、表 1 のような形式で  $\Theta, (X_1, Y_1), (X_2, Y_2), (X_3, Y_3), (X_3', Y_3'), \dots, \Gamma, \Gamma', \dots$  の各値を登録する。 □

次に、同様の手順によって入力画像  $I$  からエッジ点表 (E 表) を作成する。

[E 表作成手順] (図 9 参照)

(手順 1)  $I$  に含まれるエッジ点を、その各々の接線方向  $\theta$  でソーティングし、同じ  $\theta$  をもつエッジ点をまとめて表 2 のような形式で登録する。 □

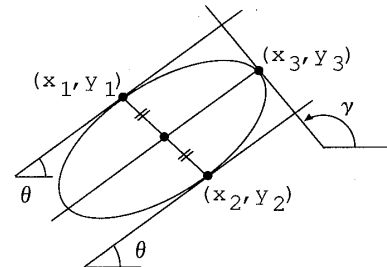


図 9 入力画像からの E 表の作成と利用  
Fig. 9 Generation and utilization of E-table from an input image.

表 2 E 表の構成  
Table 2 Contents of E-table.

| 接線方向       | エッジ点 $\mathbf{q}$ の座標               |
|------------|-------------------------------------|
| $\theta_j$ | $(x_1, y_1)_j, (x_2, y_2)_j, \dots$ |
| ...        | ...                                 |

以下、ここで作成した R 表, E 表に基づいて図形検出を行う。

3.2.2 移動量  $t$  の検出

はじめに、次の手順で参照点の候補位置を算出する (図 9 参照)。

(手順 1)  $\theta$  を  $[0, \pi)$  の間で増分  $\delta\theta$  で変化させ、各  $\theta$  に対して以下の手順 2~手順 5 を繰り返す。

(手順 2) E 表において、 $\theta_j = \theta$  となるエッジ点を二つ選び、それを  $\mathbf{q}_1 = (x_1, y_1)^t, \mathbf{q}_2 = (x_2, y_2)^t$  とする。

(手順 3) 線分  $\overline{\mathbf{q}_1\mathbf{q}_2}$  の中点から  $\theta$  の方向へ直線を引き、その直線上に存在するエッジ点  $\mathbf{q}_3 = (x_3, y_3)^t$  とその接線方向  $\gamma$  を求める。

(手順 4) R 表に登録されているすべてのレコードについて、以下の処理を行う。

(手順 4-1) 一つのレコード  $\Theta_i$  について、 $\theta = \Theta_i, \mathbf{p}_1 = (X_1, Y_1)_i^t, \mathbf{p}_2 = (X_2, Y_2)_i^t, \mathbf{p}_3 = (X_3, Y_3)_i^t, \Gamma = \Gamma_i$  とおく。

(手順 4-2) 三つの対応点 ( $\mathbf{p}_i \leftrightarrow \mathbf{q}_i, i = 1, 2, 3$ ) と、式 (4) に対応する関係式

$$\mathbf{q}_i = A\mathbf{p}_i + \mathbf{t} \tag{5}$$

を用いて、アフィン変換パラメータ  $\{a, b, c, d, t_x, t_y\}$  を算出する。

(手順 4-3) 次式で  $(x_\theta, y_\theta)$ ,  $(x_\gamma, y_\gamma)$  を求める.

$$\begin{pmatrix} x_\theta \\ y_\theta \end{pmatrix} = A \begin{pmatrix} \cos \Theta \\ \sin \Theta \end{pmatrix}, \quad (6)$$

$$\begin{pmatrix} x_\gamma \\ y_\gamma \end{pmatrix} = A \begin{pmatrix} \cos \Gamma \\ \sin \Gamma \end{pmatrix} \quad (7)$$

(手順 4-4) 角度の許容誤差  $\Delta_1$ ,  $\Delta_2$  に対して

$$|\theta - \arctan(y_\theta/x_\theta)| \leq \Delta_1 \quad (8)$$

$$|\gamma - \arctan(y_\gamma/x_\gamma)| \leq \Delta_2 \quad (9)$$

がともに成立するならば, 入力画像と同サイズの 2 次元投票配列  $H_{xy}$  における要素  $H_{xy}(t_x, t_y)$  に 1 を加える (投票).

(手順 4-5)  $\mathbf{p}_3 = (X_3', Y_3')_i^t$ ,  $\Gamma = \Gamma_i'$  として, 手順 4-2~手順 4-4 を繰り返す. (更に  $(X_3'', Y_3'')^t$ ,  $\Gamma''$ , ... も登録されている場合には, それらについても同様の手順を繰り返す.)

(手順 5) 手順 3 において, 他のエッジ点  $(x_3, y_3)$  があれば同じ処理を繰り返し, なければ手順 2 に戻って別のエッジ点を二つ選んで手順 3 以下を繰り返す.

(手順 6) すべての投票が終了したら配列  $H_{xy}$  を走査し, 投票値がしきい値  $TH_t$  以上の要素  $H_{xy}(t_{kx}, t_{ky})$  を求めて,  $\mathbf{t}_k = (t_{kx}, t_{ky})^t$  を検出対象図形  $T$  の参照点候補位置とする.

### 3.2.3 線形変換 $A$ の検出

上記の手順で得られた参照点候補位置の集合  $\{\mathbf{t}_k = (t_{kx}, t_{ky})^t\}$  に対して以下に示す処理を適用し, 各  $\mathbf{t}_k$  に対するパラメータ  $A_k = \{a_k, b_k, c_k, d_k\}$  をそれぞれ求める. ここで,  $\{a_k, b_k, c_k, d_k\}$  はすべて実数であり, それぞれがかなり異なった範囲を動くので, 本手法では投票空間の所要メモリ削減と処理の効率化のために, AHT [13] の考え方を採用する.

(手順 7) 前項で述べた手順 1~手順 6 を再度繰り返して, 各  $\mathbf{t}_k$  に対応する配列要素  $H_{xy}(t_{kx}, t_{ky})$  に投票がなされる際のパラメータ  $\{a_{ki}, b_{ki}, c_{ki}, d_{ki}\}$  をすべて各リスト  $L_k$  に登録する.

(手順 8) パラメータ  $\{a, b, c, d\}$  に対応する  $9 \times 9 \times 9 \times 9$  個のセルで構成された 4 次元投票配列  $H_{abcd}$  を用意し, リスト  $L_k$  に登録された各々のレコードに対して, 以下の手順で AHT を適用する.

(手順 8-1) リスト  $L_k$  を一度走査してパラメータ  $\{a_{ki}, b_{ki}, c_{ki}, d_{ki}\}$  のそれぞれの最大変動範囲を調べ, その範囲を探索すべき  $H_{abcd}$  空間全体に対応させ

る (注3).

(手順 8-2) リスト  $L_k$  に登録されている各レコードに対応する配列要素  $H_{abcd}(a_{ki}, b_{ki}, c_{ki}, d_{ki})$  に 1 を加える (投票). すべてのレコードに対する投票を終えたら  $H_{abcd}$  を走査し, しきい値  $TH_A$  以上の要素を抽出してそれらすべてをスタックに積む.

(手順 8-3) スタックから要素の一つ取り出す. スタックが空のときは手順 9 に進む.

(手順 8-4) 取り出した要素の各パラメータ軸におけるそれぞれの分解能が所望の数値 (例えば, 相対値で 0.01 以下) となったらそれを解候補としてリストに登録し, 手順 8-3 に戻る. そうでなければ, 次に進む.

(手順 8-5) 取り出した要素を中心とした  $3 \times 3 \times 3 \times 3$  の範囲を改めて  $H_{abcd}$  空間全体と考え, それを  $9 \times 9 \times 9 \times 9$  個のセルで再分割した上で手順 8-2 に戻る. ただし, ここでは分解能が既に所望の数値に達しているパラメータ軸に関しては再分割は行わないものとする.

(手順 9) すべてのリスト  $\{L_k\}$  に対して手順 8 を繰り返し, 解候補のリストに登録された要素を解パラメータとする.  $\square$

以上の手順により, たとえ同一参照点にアフィン変換パラメータの異なる図形が複数個存在していたとしても, 個々に分離検出が可能である.

## 4. 評価実験

提案手法の有効性を検証するために評価実験を行った. 計算には, 動作クロック 400 MHz の PentiumII プロセッサを使用した. 実験に使用したパラメータは,  $\delta\theta = 1^\circ$ ,  $\Delta_1 = \Delta_2 = 5^\circ$ , 第 1 段階投票 ( $t$ ) の検出しきい値  $TH_t = 500$ , 第 2 段階投票 ( $A$ ) の検出しきい値  $TH_A = 30$  である.

### 4.1 円テンプレートによる楕円検出

まず, 楕円が円のアフィン変換になっていることを利用して, 円テンプレートを用いて複数楕円の検出を試みた. 実験では, 半径 20 画素の円テンプレート (図 10) を用い, 図 11 に示した複数楕円の画像に対して提案手法を適用した. この入力画像では, 楕円の輪郭線の一部が重なりによる隠ぺいで大幅に欠落している点に注目されたい. 検出結果を図 12 に示す. 検出された楕円は太線で表示し, 入力画像に重ね書きし

(注3): すなわち, 開始時には, 配列  $H_{abcd}$  の要素の分解能は各パラメータ (軸) の最大変動幅の  $1/9$  に対応する.

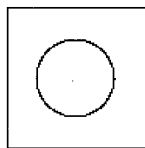


図 10 円テンプレート  
Fig. 10 Circle template.

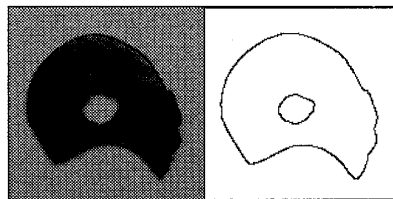


図 13 テンプレート (実画像)  
Fig. 13 Real image of the template of an industrial part.

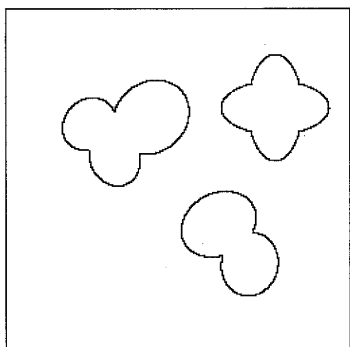


図 11 入力楕円画像  
Fig. 11 The tested image of ellipses.

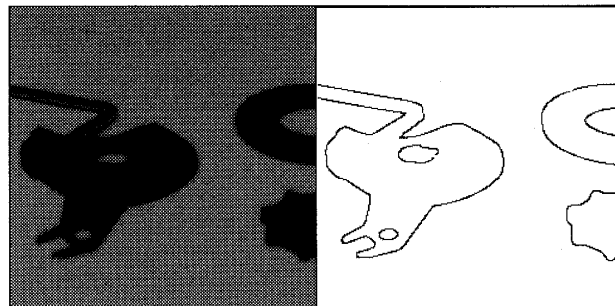


図 14 入力部品画像  
Fig. 14 The tested image of industrial parts.

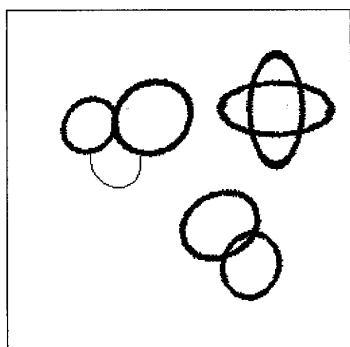


図 12 検出結果  
Fig. 12 The detected result.

である。検出に要した時間は 12.3 秒であった。これより、複数の欠損楕円がうまく分離・検出されていることがわかる。特に、同心楕円 ( $\{t_x, t_y\}$  は同じで  $\{a, b, c, d\}$  が異なる楕円) もうまく検出できている<sup>(注4)</sup>。

なお、図 11 左上部にある、三つの重なり合った楕円の中の一つについては検出できなかった。これは、この欠損楕円からは接線が平行となる輪郭点対が得られなかったためである。この例のように、円や楕円に類似した単純な図形では輪郭線が連続して 50% 以上欠落すると平行接線が得られなくなるので注意が必要である (一般的な曲線図形については一概にいうことはできない)。

#### 4.2 実画像を用いた場合の検証実験

続いて、より一般的な画像 (工業製品の部品画像) で実験を試みた。実験に用いたテンプレート画像と入力画像をそれぞれ図 13 と図 14 に示す。左側が濃淡画像、右側が Canny フィルタ [10] によって得られた輪郭線抽出画像を表しており、それらの画像サイズは、それぞれ  $158 \times 158$  画素、 $256 \times 256$  画素である。これらは、平面上に置いた薄板状の各部品をデジタルカメラを使って入力した実画像であるが、それぞれの視点が異なっているため、検出対象物体はアフィン変換的に変形していることに注目されたい。更に、検出対象物には他物体との重なりによる隠ぺいも存在している。検出結果を図 15 に示す。図中では、検出されたアフィン変換パラメータでテンプレートを変換した図形を重ね書きしてある。これより、対象物がうまく検出されていることが確認できる。検出に要した処理時間は 11.2 秒であった。

以上の結果から、提案手法の妥当性と有効性はその基本部分において確認できたものとする。

(注4)：この実験で楕円は一般的な曲線図形の一つの例として選ばれたにすぎない。楕円検出だけを目的にするのであれば、楕円の代数的・幾何的性質を積極的に利用したより効率的な検出法が多数提案されている (例えば [12])。



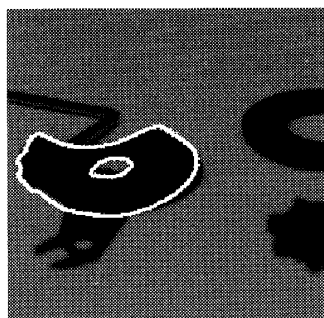


図 15 検出結果  
Fig. 15 The detected result.

### 4.3 検 討

従来手法の多く [3]~[6] は、他物体との重なりによって輪郭線の一部が隠れさされていたり、雑音によって輪郭線に途切れが生じているような場合にはうまく対処できない。幾何学的ハッシング [7] ではこのような問題はないが、テンプレートと入力画像の間で対応する 3 点を組合せ的に選んで投票することが必要であり、そのために膨大な処理時間がかかってしまい、実際には検出対象図形を有限個の特徴点や特徴線分できうまく表現できるような図形に限定している [8], [9]。

これに対して、提案手法ではこのような制約がなく、投票処理も同一の接線方向をもつ 2 点の組合せを基本としているので、比較的高速である。また、投票空間も画像サイズの 2 次元配列と AHT を併用した  $9 \times 9 \times 9 \times 9$  サイズの 4 次元配列から構成されており、所要メモリも少なく、効率的な手法となっている。

また、提案手法は、Ballard の GHT がアフィン変換に不変な図形検出法として拡張可能なことを初めて示したものであり、この意味でも重要であると考えられる。特に、もとの GHT が相似変換の二つのパラメータ { 回転角, 伸縮率 } に関してしらみつぶしの探索を行っていたのに対して、提案手法はアフィン変換に拡張されているにもかかわらずその四つのパラメータ  $\{a, b, c, d\}$  に関してしらみつぶしの探索は行っておらず、効率的な手法として実現されていることに注意されたい。

提案手法の課題は、更なる処理効率の改善と検出性能のロバスト化にあるといえる。例えば、検出対象図形中で接線が互いに平行となるエッジ点対が数多くある場合 (長方形など) には、R 表自体が大きくなってしまいうため、処理時間が増大する。このような場合には、線分近似手法などを併用した形でのアルゴリズム

の拡張が必要であろう。これとは逆に、接線が平行になる輪郭点対があまり得られなかった場合には、(実験結果でも示したように) 検出自体が不可能となる場合もあり、工夫が必要である。例えば、二つの (接線が平行でない) 輪郭点を結ぶ線の傾きと、それぞれの輪郭点における接線の傾き等をもとにした検出法を構成すること等が考えられるであろう。また、本手法における図形検出精度は、そのアルゴリズムの構成上、各エッジ点における接線情報の抽出性能に大きく左右されているといえる。したがって、検出精度向上のためには、より精度の高い接線情報抽出法の開発が必要である。これらの問題については、今後の検討課題としたい。

### 5. む す び

本論文では、任意の曲線図形をアフィン変換に不変な形で検出する手法について検討した。具体的には、まず、任意曲線を含んだ不連続輪郭線画像から安定に接線情報を抽出する手法として、ハフ変換を用いた円弧あてはめによる接線情報抽出法を新たに提案した。続いて、相似変換に不変な図形検出法である Ballard の一般化ハフ変換 (GHT) をアフィン変換に不変な図形検出法として拡張するために、アフィン変換における平行性保存の性質に着目し、接線情報を積極的に利用した形の図形検出アルゴリズムを提案した。これらによって、従来手法では対処が困難であった、他物体との重なりによって輪郭線の一部が隠れさされていたり、雑音によって輪郭線が途切れているような場合にも、初めて実用的に対処できるようになった。提案手法は主として 2 次元図形の検出を意図したものであるが、冒頭で述べたように 3 次元画像解析等にも応用可能と考えられる。今後の課題としては、図形検出のロバスト化と高速化のための改良、より高精度な接線情報抽出法の確立、などが挙げられる。

謝辞 本研究の一部は文部省科学研究費奨励研究 A (No.12780256) の助成による。

### 文 献

- [1] D.H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recognit.*, vol.13, no.2, pp.111-122, 1981.
- [2] 徐 剛, 辻 三郎, 3 次元ビジョン, 共立出版, 1998.
- [3] 宮武孝文, 松山隆司, 長尾 真, "フーリエ記述子を用いたアフィン変換に不変な曲線の認識について," *情報学論*, vol.24, no.1, pp.64-71, Jan. 1983.
- [4] J. Ben-Arie and Z. Wang, "Pictorial recognition of

objects employing affine invariance in the frequency domain," *IEEE Trans. Pattern Anal. & Mach. Intell.*, vol.20, no.6, pp.604-618, 1998.

- [5] G. Bebis, M. Georgiopoulos, N.V. Lobo, and M. Shah, "Learning affine transformations," *Pattern Recognit.*, vol.32, no.10, pp.1783-1799, 1999.
- [6] D. Zhao and J. Chen, "Affine curve moment invariants for shape recognition," *Pattern Recognit.*, vol.30, no.6, pp.895-901, 1997.
- [7] Y. Lamdan, J.T. Schwartz, and H.J. Wolfson, "Affine invariant model-based object recognition," *IEEE Trans. Rob. Aut.*, vol.6, no.5, pp.578-589, 1990.
- [8] M.S. Costa, R.M. Haralick, and L.G. Shapiro, "Optimal affine-invariant matching: Performance characterization," *SPIE*, vol.1662, pp.21-34, 1992.
- [9] F.C.D. Tsai, "Geometric hashing with line features," *Pattern Recognit.*, vol.27, no.3, pp.377-389, 1994.
- [10] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. & Mach. Intell.*, vol.PAMI-8, no.6, pp.679-698, 1986.
- [11] J. Shen and S. Castan, "An optimal linear operator for step edge detection," *Computer Vision, Graphics and Image Processing*, vol.54, no.2, pp.112-133, 1992
- [12] 渡辺孝志, 畠山雅充, 木村彰男, "ハフ変換を用いた接線情報の抽出と欠損楕円の検出," *信学論 (D-II)*, vol. J82-D-II, no.12, pp.2221-2229, Dec. 1999.
- [13] J. Illingworth and J. Kittler, "The adaptive Hough transform," *IEEE Trans. Pattern Anal. & Mach. Intell.*, vol.PAMI-9, no.5, pp.690-698, 1987.
- [14] 木村彰男, 渡辺孝志, "図形検出力を向上させた高速一般化ハフ変換," *信学論 (D-II)*, vol.J83-D-II, no.5, pp.1256-1265, May 2000.

(平成 12 年 9 月 8 日受付, 1 月 9 日再受付)



渡辺 孝志 (正員)

昭 46 東北大学院工学研究科修士課程了。同年(株)日立製作所入社。昭 55 東北大学院工学研究科博士課程了。工博。同年岩手大学工学部助手。現在同教授。画像処理, パターン認識, セルオートマトン, 地理情報システムなどの研究に従事。情報処理学会, 日本リモートセンシング学会, 地理情報システム学会各会員。



木村 彰男 (正員)

平 5 岩手大学院工学研究科情報工学専攻修士課程了。同年ソニー(株)入社。在社中は磁気記録関係の研究開発に従事。平 7 岩手大学工学部助手, 現在同講師。博士(工学)。画像処理, パターン認識に関する研究に従事。情報処理学会, 画像電子学会

各会員。