

## 線分近似による一般化ハフ変換の高速化と任意図形検出

正 員 渡辺 孝志<sup>†</sup> 非会員 石戸橋 真<sup>†</sup>

## Acceleration of Generalized Hough Transforms by Polygonal Approximation and Detection of Arbitrary Shapes

Takashi WATANABE<sup>†</sup>, *Member* and Makoto ISHITOBASHI<sup>†</sup>, *Nonmember*

あらまし 本論文では、相似変換不変な任意図形検出法として、一般化ハフ(Hough)変換(GHT)を高速化した線分近似GHTに基づく方法を提案している。具体的には、輪郭線が区分的に線分近似されている図形パターンに対してGHTの高速化が可能であることを示すと共に、これに分解能の多重化処理と非線形最小2乗法による最適パラメータ推定を組み合わせたパターン検出法を述べている。本手法は、(1)投票操作が画素単位から線分単位となり、投票処理が高速化される、(2)精度の高いパラメータ推定が可能である、(3)重なりなどで輪郭線の一部が欠損した画像にも対処できる、などの特徴を有している。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。

## 1. ま え が き

画像認識や画像計測などの分野において、相似変換に関して不変なパターン検出法を確立することは重要な研究課題である。相似変換不変な形でのパターン検出を意図した代表的な手法としては、フーリエ・メルン変換<sup>(1)</sup>、複素対数変換<sup>(2)</sup>、フーリエ記述子<sup>(3)</sup>、一般化ハフ変換<sup>(4)</sup>(以下、GHTと略称する)などがあるが、それぞれ問題を抱えており満足できる状況にはない。例えば、フーリエ・メルン変換はフーリエ変換画像の位相成分を無視しているために、適用できる画像がかなり限られる。複素対数変換は回転・伸縮に関して不変であるが、移動不変ではない。フーリエ記述子は一筆書き可能で孤立したパターンについてのみ適用可能である。また、GHTは所要メモリが大きい、処理時間がかかる、などの問題がある。

その中で、GHTは他の方法に比べて、(1)雑音に強い、(2)認識対象物同士が重なり合っただけでパターンの一部が陰べいされるような場合にも対処できる、などの優れた特徴を有しており、改良によってその適用範囲を更に広げ得るものと考えられる。筆者らは、先に、こ

のような立場からGHTを用いた任意だ円パターンの検出問題を論じた<sup>(5)</sup>。任意だ円の検出問題は更に1段上のアフィン変換不変な検出問題であり、しかも筆者らの提案しただ円検出法はだ円の幾何学的性質に強く依存しない手法となっているので、この手法を発展させることで一般形状パターンの検出法が確立できるものと考ええる。

本論文では、より優れた相似変換不変なパターン検出法の確立を目的としてGHTの改良を試みた。具体的には、輪郭線が区分的に線分近似されている図形パターンに対してGHTの高速化が可能であることを示すと共に、これに分解能の多重化処理と非線形最小2乗法による最適パラメータ推定を組み合わせたパターン検出法を提案している。これにより、GHTにおけるパターン検出処理の精度と速度に関して一層の向上を図ることができる。以下、2.ではGHTの問題点について論じ、3.ではGHTの高速化手法としての線分近似GHTの原理を述べる。4.では更に検出処理速度と検出精度の改善を図るための諸工夫を述べる。5.では具体的な処理手順の詳細を説明し、6.では提案手法の有効性を検証した評価実験の結果について述べる。

## 2. 一般化ハフ変換の問題点

ハフ変換は、当初、直線検出のための手法として提

<sup>†</sup> 岩手大学工学部情報工学科, 盛岡市  
Faculty of Engineering, Iwate University, Morioka-shi, 020 Japan

案されたが<sup>(6),(7)</sup>, Merlin と Farber により任意曲線図形に適用可能な手法 (以下, MF 法と言う) として拡張された<sup>(8)</sup>. MF 法はパターンマッチングの一種であるが<sup>(5)</sup>, これは次のような手順で構成される.

〈MF 法の原理〉

処理対象画像は 2 値化された輪郭線画像  $I$  とする.

(手順 1) テンプレート画像  $T$  上に参照点  $R$  を一つ適当に選び,  $R$  を中心にして  $T$  を 180 度回転したパターン  $\underline{T}$  を作成する. すなわち,

$$\underline{T}(x, y) = T(-x, -y) \quad (1)$$

(手順 2) 入力画像  $I$  と同じ大きさの配列  $A$  を用意して, その初期値を 0 にセットする.

(手順 3)  $I(x, y) = 1$  となる各エッジ点  $(x, y)$  に対し, 配列  $A$  上で点  $(x, y)$  に参照点を置いてパターン  $\underline{T}$  を描き, その輪郭線が通過する画素位置  $(x', y')$  の配列要素  $A(x', y')$  に 1 を加算する (これを投票と言う).

(手順 4) 全エッジ点について投票を終えたならば, 配列  $A$  を走査し, 投票数がしきい値以上となる配列要素  $A(x, y)$  のインデックス値  $(x, y)$  を検出対象パターンの存在する画素位置候補とする. (手順完)

MF 法ではテンプレートの平行移動パターンのみが検出可能であり, 伸縮と回転の変換を受けたパターンについては考慮されていなかった. そこで, Ballard は MF 法を拡張し, 投票空間に 4 次元配列 (位置座標  $x, y$ , 伸縮率  $\alpha$ , 回転角  $\theta$ ) を用いる相似変換不変なパターン検出法を提案した<sup>(4)</sup>. この手法を一般化ハフ変換 (GHT) と呼ぶ習わしであるが, その問題点としては次が挙げられる.

(1) 膨大な作業用メモリが必要である 投票空間として 4 次元配列が必要であり, その大きさは検出精度の 4 乗に比例したものとなる.

(2) 膨大な処理時間がかかる 上記投票操作を  $\alpha$  と  $\theta$  の組合せ数だけ繰り返す必要があり, これが処理時間を増大させる最大の要因となっている. このため, GHT では,  $R$  表と呼ばれる変換表と各エッジ点におけるこの配情報を利用して投票処理の効率化を図っているが, まだ十分ではない.

(3) パターンの検出精度が低い 投票空間では高い投票数のピーク値が孤立して存在することは少なく, 同程度のピーク値が連続して広く分布し, ピーク値が顕著でない場合が多い. このような場合における適切なパラメータ決定法が知られていない.

以上のような問題は画像が複雑になるほど顕著である. 従って, GHT に基づくパターン検出法を実用レベ

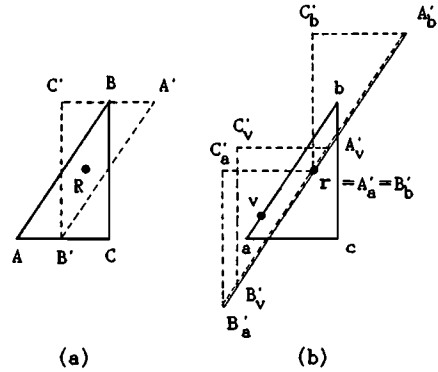


図1 線分近似 GHT の説明

Fig. 1 Illustration of the MF method using polygonal approximation. (a) shows a template pattern and (b) does the voting configuration.

ルのものとするためには, まだ検討すべき課題が多く残っていることがわかる. 以下では, GHT の原型となった MF 法に立ち返って, これらの問題点の解決法を模索していく.

### 3. 線分近似一般化ハフ変換

#### 3.1 基本的な考え

GHT を高速化するためには, 投票回数を削減すると共に, 1 回当りの投票処理を効率化することが必要である. MF 法では画像の各エッジ点ごとに投票パターン  $\underline{T}$  全体を投票した. GHT では, 各エッジ点でのこの配情報を利用して,  $\underline{T}$  の一部だけを投票すれば良いように改良された. これによって投票処理はかなり効率化されたが, それでも各エッジ点ごとに投票操作が必要である. 多くのエッジ点ではその座標値と濃淡この配値が隣接エッジ点間で緩やかに変化しており, これを利用すれば投票処理を更に効率化できるものと考えられる. 実際, 入力画像とテンプレート画像における輪郭線を区分的に線分近似することで, MF 法を効率良く実行できることが示せる.

以下, これを例によって説明しよう.

〈線分近似 GHT の原理〉

今, 図 1 (a) に示すようにテンプレート  $T$  を三角形  $\triangle ABC$  とする. 参照点を  $R$  とすれば, MF 法での投票パターン  $\underline{T}$  は 180 度回転した  $\triangle A'B'C'$  となる. 図 1 (b) の入力画像  $I$  中にパターン  $\triangle abc$  があり, テンプレート  $T$  に平行移動で一致すると仮定する. このとき, 線分  $ab$  上のエッジ点  $v$  による投票結果は  $\triangle A'_v B'_v C'_v$  であり, 線分  $ab$  上の全エッジ点による投

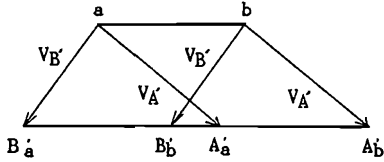


図2 投票処理の説明  
Fig. 2 Illustration of voting processing.

票結果は  $\triangle A_v B_v C_v$  を線分  $ab$  上に沿って1画素ずつ移動したものとなる。従って、線分  $B_a A_b$  上の投票値が際立って大きくなる。特に、参照点  $R$  に対応する位置  $r$  の周辺部分で顕著に大きい。  $r$  が検出すべき参照点の位置であるから、これにより MF 法の投票操作は次のように効率化できることがわかる。

(1) 入力画像中で投票操作が必要なエッジ点  $(v)$  は、その所属する線分  $(ab)$  がテンプレート画像中のいずれかの線分  $(AB)$  と平行なものに制限できる。

(2) 投票では、投票パターン  $T (= \triangle A' B' C')$  全体を投票する必要はなく、投票操作を行うエッジ点  $(v)$  の所属線分  $(ab)$  に平行な線分  $(A' B')$  のみを投票すればよい。

(3) しかも、同一線分  $(ab)$  上にある各エッジ点による投票結果をまとめたパターンは、また一つの線分  $(B_a A_b)$  である。この様子を、対応する線分  $ab$  と線分  $A' B'$  の長さが異なる一般の場合について示すと図2のようになる。ここで、  $V_A, V_B$  は  $T$  における点  $A', B'$  の位置ベクトルである。投票結果の線分を  $s (= B_a A_b)$  とすると、線分  $s$  の長さ  $l(s)$  は

$$l(s) = l(ab) + l(A' B') - 1 \quad (2)$$

である。今、

$$l_{\max} = \max\{l(ab), l(A' B')\} \quad (3)$$

$$l_{\min} = \min\{l(ab), l(A' B')\} \quad (4)$$

とおけば、  $s$  上の各点での投票値は  $s$  の両端点で1、内側に1画素ずつ入るごとに1増加し、両端から  $l_{\min} - 1$  以上離れた点では一定値  $l_{\min}$  となっている。よって、実際の投票値加算は線分  $s$  全体ではなく、この一定値となっている部分線分  $s' (= B_b A_a)$  の近傍のみに制限してよいこととなる。ここで、  $s'$  の長さは

$$l(s') = l_{\max} - l_{\min} + 1 \quad (5)$$

である。

従って、線分近似された画像では線分単位に投票処理を行うことができ、エッジ点単位の投票処理を行う従来の GHT に比べて、かなりの高速化が期待できる。以下、この手法を線分近似 GHT と呼ぶことにする。

線分近似 GHT を適用するためには、輪郭線を線分近

表1 線分表 (S表) の形式

傾斜角 $\phi$	線分数 $n$	線分データ
0	$n_0$	$\{ \langle l_i, v_{i1}, v_{i2} \rangle, (i=1, 2, \dots, n_0) \}$
$\Delta\phi$	$n_1$	$\{ \langle l_i, v_{i1}, v_{i2} \rangle, (i=1, 2, \dots, n_1) \}$
$2\Delta\phi$	$n_2$	$\{ \langle l_i, v_{i1}, v_{i2} \rangle, (i=1, 2, \dots, n_2) \}$
...	...	...

似する手法が必要である。本論文では輪郭線の線分近似手法については特に議論はしないが、輪郭線が途切れなく連続している場合には多くの方法が知られている<sup>(9)</sup>。

### 3.2 線分データの管理

線分近似 GHT では、線分近似された入力画像とテンプレート画像の間で平行な線分対を見出す必要がある。そこで、輪郭線の近似線分を線分の傾斜角ごとにまとめた線分表 (以下、S表と言う) を用いることで、平行な線分対の探索を高速化する。

S表の形式を表1に示す。ここで、  $\Delta\phi$  は線分傾斜角の量子化幅であり、傾斜角  $\phi (= m\Delta\phi)$  の欄には

$$|\theta_i - \phi| \leq \Delta\phi/2 \quad (6)$$

なる傾斜角  $\theta_i$  をもつ線分  $s_i$  が収容される。登録される線分データは線分  $s_i$  の長さ  $l_i$  (画素数換算値で)、両端点座標  $v_{i1} = (x_{i1}, y_{i1}), v_{i2} = (x_{i2}, y_{i2})$  である。

### 3.3 線分近似 GHT の検討課題

MF法は移動に関して不変な検出法であるが、回転や伸縮に関しては不変でない。従って、線分近似 GHT で任意図形を検出するためには、伸縮率  $\alpha$  と回転角  $\theta$  の異なった組  $(\alpha, \theta)$  について投票パターン  $T$  を生成し、これらについて3.1で述べた投票処理を繰り返し適用して、最大投票数を与える変換パラメータ  $(x, y, \alpha, \theta)$  を見出す必要がある。その際、最大投票数を与える位置座標  $(x, y)$  を  $(\alpha, \theta)$  ごとに更新して求めていくことにすれば、投票空間は4次元配列でなく、  $(x, y)$  だけの2次元配列で済む。

しかし、線分近似 GHT を実用化する上ではまだ次のような問題に対処する必要がある。

#### 〈線分近似 GHT の検討課題〉

(1) 必要な  $(\alpha, \theta)$  の組合せ数は膨大であり、そのために処理時間がかかる。

(2) 1組の  $(\alpha, \theta)$  ごとに投票用2次元配列  $A$  全体を2回走査 (配列クリアと最大値検出) するので、画像が大きいと処理時間がかかる。

(3) 複数パターンが存在する場合、パターンを一つずつ検出するたびに線分近似 GHT を適用し直すのでは、

処理時間がかかる。

(4) パターンの検出精度を高めるための工夫がまだ与えられていない。

なお、線分を投票単位とする GHT という考え方は Davis<sup>(12)</sup> にも見られるが、そこでは 3.1 で述べたような効率的な投票形式とはなっておらず、上記の問題に関する検討もほとんど行われていない。

#### 4. パターン検出手順の工夫

以下では、前章に述べた線分近似 GHT の検討課題を解決するための対応策について述べる。

##### 4.1 パラメータ分解能の多重化

分解能の多重化は GHT を高速化する上で有効である<sup>(13)</sup>。そこで、本研究でも前章の課題(1)と(2)に対処するため、変換パラメータ(位置座標、回転角、伸縮率)の分解能を多重化して考える。まず、位置座標については、入力画像の分解能レベルを第1レベル(最下位層)として、第 $(k-1)$ レベルの $2 \times 2$ 画素を第 $k$ レベルの1画素に対応させ、全部で $k$ 階層からなる多重化構成をとる。これに対応し、第 $k$ レベルにおける伸縮率と回転角の分解能も第 $(k-1)$ レベルにおける分解能の $1/2$ で考える。

パターン検出手順としては、次の 4.2 に述べるように、上位層で粗く変換パラメータの候補を選び、下位層でより詳細にパラメータを決定していくというプロセスを踏むものとする。これにより探索に必要な $(\alpha, \theta)$ の組合せ数と投票用配列が縮小されるので、全体としてかなりの高速化が期待できる。

##### 4.2 投票処理の制限と線分マッチングの採用

線分近似 GHT は GHT よりも高速であるが、それでも1回当りの処理時間はかなりかかり、多重化された分解能の全レベルにわたって線分近似 GHT を適用するのは得策ではない。そこで、線分近似 GHT は最上位層にのみ適用し、中間層での詳細なパラメータ探索は線分マッチングによるものとする。

すなわち、線分近似された入力画像とテンプレート画像に対して最上位層の分解能レベルで $\alpha$ と $\theta$ を変えて線分近似 GHT を適用し、しきい値以上の投票数を得た相似変換パラメータを候補リストに登録する。登録されるパラメータは $\alpha, \theta$ , 位置座標 $x, y$ , 投票数 $n$ である。次に、投票数の大きい順に候補リストからパラメータ $(x, y, \alpha, \theta)$ を選ぶ。レベル $k$ の層 $(2 \leq k \leq k)$ における一つのパラメータに対してレベル $(k-1)$ の層では複数個の詳細パラメータが対応するが、そのなか

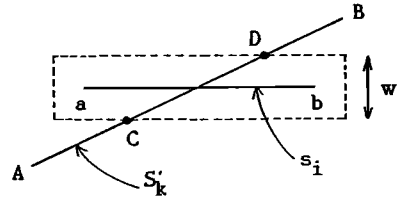


図3 線分マッチング度

Fig. 3 Matching measure of line segments.

ら適切なものを線分マッチング度に基づいて選択していく。

線分マッチング度としては次に示すようなもの考える(図3参照)。まず、入力画像 $I$ 中の各線分 $s_i$ の幅を $w$ として考える。次に、該当する詳細パラメータで原テンプレート $T$ を相似変換して得られるテンプレートを $T'$ とし、 $T'$ 中の線分 $S_k'$ が $I$ 中の線分 $s_i$ に交差する度合(線分マッチング度)を

$$m(s_i, S_k') = \{\text{線分 } CD \text{ の長さ}\} \quad (7)$$

で定義する。これをテンプレート $T'$ の全線分について求め、その和 $M$ が最大となる詳細パラメータを最適値として選択する。

候補リストから選ばれた最上位層のパラメータに対して、第2レベルの層まで、上記手順でその詳細パラメータを決定していく。最終的なパラメータは、その結果に基づいて、第1レベルの分解能で次の 4.3 に述べる方法により決定する。

線分マッチング度の計算は簡単であり、探索範囲が限定されている場合には線分近似 GHT を繰り返して使用するよりも処理が速くなる。この場合、線分近似 GHT の適用が最上位層のみに限定されており、複数パターンを検出は候補リスト中の残りのパラメータについて上記探索処理を繰り返せばよいので、前章の課題(3)に対処できる。その詳細は 5. の具体的アルゴリズムに述べる。

##### 4.3 非線形最小2乗法と最適パラメータ決定

前章の課題(4)に対処するため、ここでは非線形最小2乗法を用いた最適な相似変換パラメータの決定法について述べる。

大まかな変換パラメータの値は第2レベル層までの線分マッチングにより得られており、ここでの処理分解能レベルは最下位層の第1レベルである。

今、入力画像 $I$ は $M$ 本の線分 $\{s_i\}$ 、テンプレート画像 $T$ は $N$ 本の線分 $\{S_k\}$ で近似されているとする。ここで、 $T$ の線分 $S_k$ の傾斜角を $\phi_k$ 、その中点座標を

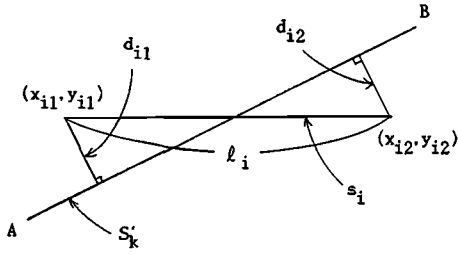


図4 線分間距離  $D_{ik}$   
Fig. 4 Distance between line segments  $D_{ik}$ .

$(X_k, Y_k)$ とする。このとき、線分  $S_k$ の直線方程式は  

$$(y - Y_k)\cos\phi_k - (x - X_k)\sin\phi_k = 0 \quad (8)$$
 で与えられる。伸縮率  $\alpha$ 、回転角  $\theta$ 、移動量  $(x_0, y_0)$ の相似変換により、線分  $S_k$ は線分  $S_{k'}$ に移り、その方程式は次式となる。

$$(y - Y_{k'})\cos\phi_{k'} - (x - X_{k'})\sin\phi_{k'} = 0 \quad (9)$$

但し、

$$\phi_{k'} = \phi_k + \theta \quad (10)$$

$$\begin{pmatrix} X_{k'} \\ Y_{k'} \end{pmatrix} = \alpha \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} X_k \\ Y_k \end{pmatrix} + \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \quad (11)$$

である。入力画像  $I$ 中の線分  $s_i$ の両端点座標を  $(x_{i1}, y_{i1})$ ,  $(x_{i2}, y_{i2})$ とする。線分  $s_i$ に線分  $S_{k'}$ が対応することを  $\varepsilon_i(k)=1$ 、そうでないことを  $\varepsilon_i(k)=0$ で表す。ここで、 $\varepsilon_i(k)$ は第2レベルでの線分マッチング結果を用いて計算され、 $\varepsilon_i(k)=1$ は

$$m(s_i, S_{k'}) = \max_j \{m(s_i, S_{j'})\} > 0 \quad (12)$$

であることを示す。このとき、線分  $s_i$ の線分  $S_{k'}$ に対する距離尺度  $D_{ik}$ を

$$D_{ik} = l_i \sqrt{(d_{i1}^2 + d_{i2}^2)/2} \quad (13)$$

で定義する(図4)。ここで、

$$l_i = \sqrt{(x_{i1} - x_{i2})^2 + (y_{i1} - y_{i2})^2} \quad (14)$$

$$d_{ij} = |(y_{ij} - Y_{k'})\cos\phi_{k'} - (x_{ij} - X_{k'})\sin\phi_{k'}| \quad (j=1, 2) \quad (15)$$

であり、 $D_{ik}$ は2線分  $(s_i, S_{k'})$ 間の面積的な距離尺度を表す。例えば、 $s_i$ が  $S_{k'}$ に重なれば  $D_{ik}=0$ で、 $d_{i1}$ または  $d_{i2}$ が大きくなるほど  $D_{ik}$ も大きくなる。

相似変換パラメータ  $\Theta = (\alpha, \theta, x_0, y_0)^T$ の最適値  $\Theta^*$ は、次の評価関数

$$J = \sum_{i=1}^M \sum_{k=1}^N \varepsilon_i(k) D_{ik}^2 \quad (16)$$

を最小とする点として定める。この最小化問題は非線形最小2乗問題となるので、ここではニュートン・ラフソン法を用いて反復的手法で解くことを考える<sup>(11)</sup>。

評価関数  $J$ のパラメータ  $\Theta$ に対するヘシアン行列を  $H$ とすれば、 $n$ 番目の推定値  $\Theta^{(n)}$ に対して

$$\begin{pmatrix} \Delta\alpha \\ \Delta\theta \\ \Delta x_0 \\ \Delta y_0 \end{pmatrix}^T = H^{-1} (\partial J / \partial \alpha, \partial J / \partial \theta, \partial J / \partial x_0, \partial J / \partial y_0)^T |_{\Theta = \Theta^{(n)}} \quad (17)$$

を計算し、反復式として

$$\alpha^{(n+1)} = \alpha^{(n)} - \Delta\alpha \quad (18)$$

$$\theta^{(n+1)} = \theta^{(n)} - \Delta\theta \quad (19)$$

$$x_0^{(n+1)} = x_0^{(n)} - \Delta x_0 \quad (20)$$

$$y_0^{(n+1)} = y_0^{(n)} - \Delta y_0 \quad (21)$$

を得る。停止規則は  $|\Delta\alpha|$ ,  $|\Delta\theta|$ ,  $|\Delta x_0|$ ,  $|\Delta y_0|$ がすべて十分小さくなった時点で打ち切るものとする。

この反復計算で得られるのは  $J$ の極小点であるが、初期値  $\Theta^{(0)}$ としては第2レベル層までの線分マッチングでかなり良好なものが得られているので、上記反復処理で得られる極小点は  $J$ の最小点  $\Theta^*$ とみなすことができる。

## 5. パターン検出アルゴリズム

以下に、線分近似 GHT にパラメータ分解能の多重化、線分マッチング、非線形最小2乗法による最適パラメータ推定などの処理を併用したパターン検出手順を述べる。説明を簡単にするため、ここでは分解能の多重化レベル数が  $p=3$ の場合について説明する。

〈パターン検出アルゴリズム〉

(手順1) テンプレート画像  $T$ と処理対象の入力画像  $I$ を2値の輪郭線画像に変換し、更に、これらを適当な方法で線分近似する(以下、線分近似された画像も同じく  $T$ と  $I$ で表す)。

(手順2) テンプレート  $T$ の座標原点を参照点  $R$ をおき、 $T$ を点  $R$ について、180度回転させたものを  $\underline{T}$ とする。また、画像  $I, T, \underline{T}$ に対する線分表を作成し、それぞれ  $S(I)$ 表、 $S(T)$ 表、 $S(\underline{T})$ 表と呼ぶ(以下、 $S(I)$ 表に登録される線分を小文字  $s_i$ 、 $S(T)$ 表や  $S(\underline{T})$ 表に登録される線分を大文字  $S_k$ で表す)。

(手順3) 分解能の最上位レベル(第3層)で、線分近似 GHT を以下により適用する。

(3.1) 伸縮率  $\alpha_3$ 、回転角  $\theta_3$ を増分  $\Delta\alpha_3, \Delta\theta_3$ で

$$\alpha_{3\min} \leq \alpha_3 \leq \alpha_{3\max}, \quad \theta_{3\min} \leq \theta_3 \leq \theta_{3\max}$$

の範囲で変え、(3.2)から(3.6)の処理を繰り返す。

(3.2) 入力画像  $I$ の縦横ともに  $1/2^{p-1}$ (= $1/4$ )サイズの投票配列  $A$ を用意し、その内容を0に初期化する。

(3.3)  $S(\underline{T})$ 表の線分データを  $(\alpha_3, \theta_3)$ 分だけ相似変換させて  $S(\underline{T}')$ 表を作成する。すなわち、線分の長さは  $\alpha_3$ 倍し、座標値は  $\alpha_3$ 倍して  $\theta_3$ だけ回転し、線分の

角度  $\phi$  は

$$\phi' \equiv \phi + \theta_3 \pmod{\pi}$$

で置き換える。

$$(3.4) \quad S(T') \text{ 表の各線分 } S_k' \text{ を角度 } \phi' \text{ 順に探索し,} \\ |\phi - \phi'| \leq \eta_\phi \quad (\eta_\phi: \text{しきい値})$$

なる角度  $\phi$  をもつ線分  $s_i$  を  $S(I)$  表から求める。

(3.5) この線分対  $(s_i, S_k')$  に対し,  $s_i$  の始点と終点を  $a, b$  で,  $S_k'$  の始点ベクトルと終点ベクトルを  $V_{A'}, V_{B'}$  で表わす。図 2 に示すように, 共線関係にある 4 点を  $A_a' = a + V_{A'}$ ,  $B_a' = a + V_{B'}$ ,  $A_b' = b + V_{A'}$ ,  $B_b' = b + V_{B'}$  とおくと, 線分  $B_b'A_a'$  に相当する配列  $A$  の要素に投票値として  $l_{\min} = \min\{l(s_i), l(S_k')\}$  を加算する。

(3.6)  $S(T')$  表の全線分について上記探索が終了したならば, 配列  $A$  を走査し, しきい値  $VTH$  を超す投票値  $n$  をもつ画素位置  $(x_3, y_3)$  と  $\alpha_3, \theta_3, n$  を候補リストに登録する。ここで,  $VTH$  は

$$VTH = l(T') \cdot r_c, \quad (0 < r_c < 1)$$

である。ここで,  $l(T')$  は  $T'$  の総線分長 (画素数換算) である。

(手順 4) 分解能の中間レベル (第 2 層) で, 線分マッチング度による詳細パラメータの探索を以下により行う。

(4.1) 最上位レベルでの候補リストから投票数最大のパラメータ  $\Theta_3 = (\alpha_3, \theta_3, x_3, y_3)$  を選ぶ。

(4.2) 伸縮率  $\alpha_2$ , 回転角  $\theta_2$ , 座標位置  $(x_2, y_2)$  を増分  $\Delta\alpha_2, \Delta\theta_2, \Delta x_2, \Delta y_2$  で

$$\alpha_3 - \Delta\alpha_2 \leq \alpha_2 \leq \alpha_3 + \Delta\alpha_2$$

$$\theta_3 - \Delta\theta_2 \leq \theta_2 \leq \theta_3 + \Delta\theta_2$$

$$x_3 - \Delta x_2 \leq x_2 \leq x_3 + \Delta x_2$$

$$y_3 - \Delta y_2 \leq y_2 \leq y_3 + \Delta y_2$$

の範囲で変え, 次の(4.3)の処理を繰り返す。ここで, 各増分は  $\Delta\alpha_2 = \Delta\alpha_3/2$ ,  $\Delta\theta_2 = \Delta\theta_3/2$ ,  $\Delta x_2 = \Delta y_2 = 2^{p-2}$  である。

(4.3) 各パラメータ  $\Theta_2 = (\alpha_2, \theta_2, x_2, y_2)$  について  $S(T)$  表の線分データを相似変換して  $S(T')$  表を作成し,  $S(T')$  表と  $S(I)$  表の各線分間のマッチング度を求め, その総和を  $M(\Theta_2)$  とする。

(4.4) 得られた  $\{M(\Theta_2)\}$  の中で最大値を与えるパラメータを  $\Theta_2^* = (\alpha_2^*, \theta_2^*, x_2^*, y_2^*)$  とし,

$$M(\Theta_2^*)/l(T') \geq MTH \quad (0 < MTH < 1)$$

ならば手順 5 に行き, そうでなければ候補リストから  $\Theta_3$  を削除して手順 7 にいく。

(手順 5) 分解能の最下位レベル (第 1 層) で, 非線形

最小 2 乗法を用いて最終的なパラメータ決定を以下により行う。

(5.1)  $S(I)$  表の各線分  $s_i$  に対し,  $S(T)$  表に変換パラメータ  $\Theta_2^*$  を施して得られる  $S(T')$  表の各線分  $S_k'$  との線分マッチング度  $m(s_i, S_k')$  に基づき, 式(12)により対応付け関数  $\varepsilon_i(k)$  を定める。

(5.2) 初期パラメータ値を  $\Theta^{(0)} = \Theta_2^*$  とし, 4.3 節に述べたニュートン・ラフソン法を用いて, 評価関数  $J$  の最小値  $\Theta^* = (\alpha^*, \theta^*, x^*, y^*)$  を求め, これを検出パターン of 相似変換パラメータ値とする。

(手順 6)  $S(I)$  表から, 検出パターンを削除する。すなわち,  $\varepsilon_i(k) = 1$  となっている線分  $s_i$  の登録データを削除する。また, 候補リストから  $\Theta_3$  の近傍パラメータ  $\Theta_3'$  を削除する。すなわち,

$$|\alpha_3' - \alpha_3| \leq \eta_\alpha, \quad |\theta_3' - \theta_3| \leq \eta_\theta,$$

$$|x_3' - x_3| \leq \eta_x, \quad |y_3' - y_3| \leq \eta_y,$$

となっているパラメータ  $\Theta_3' = (\alpha_3', \theta_3', x_3', y_3')$  を削除する。

(手順 7) 候補リストが空になったら終了し, そうでないときは(4.1)に戻る。 (手順完)

## 6. 実験と検討

提案手法の有効性を検証するために, 簡単な評価実験を行った。使用計算機は三洋 MPS-020-3 である。

### 6.1 処理時間の評価実験

最初に, 単純なテンプレートパターンを用いて, MF 法 (2.), オリジナルな線分近似 GHT (3.) および改良された線分近似 GHT (4., 5.) のそれぞれにおける処理時間を測定した。実験に用いたテンプレート画像は 6 線分で構成される簡単な英字であり, その総輪郭点数は 22 画素である。入力画像は  $128 \times 128$  画素の大きさで, テンプレートを適当な  $\alpha$  と  $\theta$  で相似変換して配置し, その総輪郭点数は 80 画素である。パラメータの探索範囲は, 文字の大きさが 5 画素から 64 画素まで (精度は 1 画素), 角度が 0 度から 359 度まで (精度は 1 度) である。実験に用いた各種定数は次のとおりである:  $\Delta\alpha_3 = 4$  画素相当,  $\Delta\theta_3 = 4^\circ$ ,  $\Delta\phi = 2^\circ$ ,  $r_c = MTH = 0.5$ ,  $w = 6$  画素,  $\eta_\alpha = 8$  画素相当,  $\eta_\theta = 4^\circ$ ,  $\eta_x = \eta_y = 6$  画素,  $\eta_\phi = 4^\circ$ 。実験の結果, MF 法が約 30 時間, オリジナルな線分近似 GHT が約 1 時間, 改良された線分近似 GHT が約 60 秒という処理時間 (実行時間) を要した。但し, 検出パラメータの誤差はすべて 1 画素以下, 1 度以下であった。これにより, 提案手法は MF 法に比べて約 1,800 倍の高速化を達成したことになる。

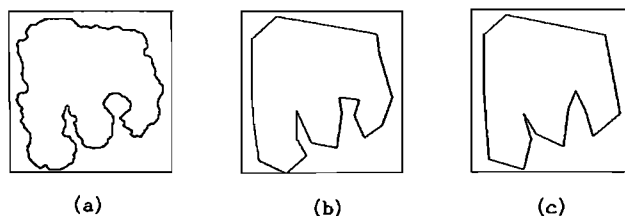


図5 実験に用いたテンプレートパターン

Fig. 5 Template patterns used in the experiment. (a) shows the original pattern, (b) and (c) do the polygonally approximated patterns.

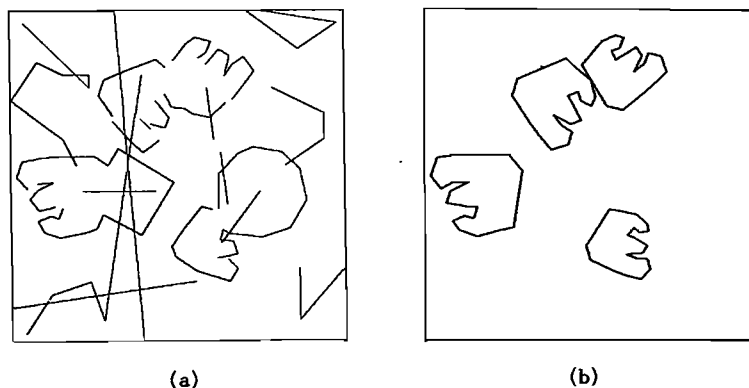


図6 パターン検出例

Fig. 6 Example of patterns detected by the proposed method. (a) shows the input image and (b) does the result of pattern detection.

表2 検出パターンの精度

入力データ			検出データ		
伸縮率 $\alpha$ (画素数)	回転角 $\theta^\circ$	位置座標 ( $x, y$ )	伸縮率 $\alpha$ (画素数)	回転角 $\theta^\circ$	位置座標 ( $x, y$ )
2.333 (70)	90	(40, 130)	2.346 (70)	90	(42, 131)
2.167 (65)	330	(100, 70)	2.118 (64)	327	(100, 71)
1.833 (55)	300	(150, 180)	1.747 (52)	294	(151, 182)
2.000 (60)	230	(150, 50)	1.900 (57)	231	(152, 48)

## 6.2 重なり合う欠損パターンの検出実験

次に、より複雑な場合を想定して、テンプレート画像としては図5(a)に示すような「十和田湖」の形状を用いた。線分近似としては、輪郭線を追跡しながら誤差が $\epsilon$ 以内に収まるように区分的に近似していく手法を用いた(本実験では $\epsilon=3$ 画素)。その近似結果例を図5(b), (c)に示す。図5(a)の総輪郭点数は258画素、図5(b)の線分数は20、大きさは $30 \times 30$ 画素である。入力画像としては、図6(a)に示すように、重なりで輪郭線の一部が欠損し、雑音としての線分パターン

が加わった線分画像を用いた。この画像のサイズは $256 \times 256$ 画素、総線分数は100、総輪郭点数は2461画素である。6.1の実験と異なる定数としては、 $\alpha_{\min}=15$ 画素、 $\alpha_{\max}=74$ 画素、 $r_c=0.25$ である。図5(b)をテンプレート画像として検出処理を行った結果を図6(b)に示す。入力画像中の四つの「十和田湖」パターンが検出でき、その精度は表2に示すようにほぼ満足すべきものとなった。検出に要した時間は6分56秒であり、このうち、最上位層の線分近似GHTの処理に2分14秒を要した。

### 6.3 検 討

#### (1) 線分近似 GHT の適用範囲について

提案手法は輪郭線の線分近似に基づいており、その線分近似精度はパターンの検出精度に影響を与える。特に、線分の傾斜角を正しく推定することが必要である。これは、線分近似 GHT で高速化を達成できた最大の要因が、投票対象線分をテンプレート画像と入力画像中の平行線分対だけに限定し得たことにあるからである。例えば、平行でない 2 線分を投票対象とした場合には、その投票域は図 2 のような線分  $B_0'A_0'$  とはならず平行四辺形で囲まれる領域となってしまう。提案手法において線分近似 GHT の適用を分解能の荒い最上位層に限定したもう一つの理由は、このような状況に対処するためである。実際には線分傾斜角の精度が十分でない場合が多く、平行性からずれた線分対に対する投票処理の高速化を図ることが実用化の上で必要となるが、この問題は今後の検討課題である。

#### (2) 評価画像について

本実験では提案手法の原理的な正当性を検証することが主目的であり、そのため、処理対象画像も比較的簡単なものを用いた。原理的な面では線分近似 GHT の有効性を立証できたものと考えが、本手法をより正当に評価するためには、実際の画像で評価実験を行っていくことが不可欠である。その際には、雑音環境下で連続性が全く欠如してしまうような輪郭線にも対処できる線分近似手法が必要である。本論文ではこの問題については議論しなかったが、現在この立場での線分近似手法の検討を進めており<sup>(10)</sup>、これを踏まえた評価実験については今後の課題としたい。

## 7. む す び

本論文では、相似変換不変なパターン検出法として、一般化ハフ変換 (GHT) を高速化した線分近似 GHT に基づく任意図形検出法を提案した。線分近似 GHT では、パターンの輪郭線を区分的に線分近似することで GHT における投票操作を画素単位から線分単位に変更でき、これによって GHT をより高速化できることを示した。更に、線分近似 GHT に分解能の多重化処理と非線形最小 2 乗法による最適パラメータ推定を組み合わせることで、(1)所要メモリが少ない、(2)投票処理が高速である、(3)精度の高いパラメータ推定が可能である、などの特徴を有する任意図形検出法が得られた。今後の検討課題としては、線分近似精度が十分でない場合にも対処できる高速な投票処理方式を開発するこ

と、などがある。

謝辞 日ごろ御指導を頂く本学工学部情報工学科横山隆三教授、ならびに実験に御協力を頂いた同学科相馬孝志技官に感謝する。なお、本研究の一部は文部省科学研究費(一般研究(C) No. 03650286)による。

### 文 献

- (1) Casasent D. and Psaltis D.: "Position, rotation, and scale invariant optical correlation", *Applied Optics*, **15**, 7, pp. 1795-1799 (1976).
- (2) Messner R. A. and Szu H. H.: "An image processing architecture for real time generation of scale and rotation invariant patterns", *Comput. Vision, Graphics & Image process.*, **31**, pp. 50-66 (1985).
- (3) 上坂吉則: "開曲線にも適用できる新しいフーリエ記述子", *信学論(A)*, **J67-A**, 3, pp. 166-173 (1984-03).
- (4) Ballard D. H.: "Generalizing the Hough transform to detect arbitrary shapes", *Pattern Recognition*, **13**, 2, pp. 111-122 (1981).
- (5) 渡辺孝志, 柴田俊浩: "Hough 変換と階層化画像を用いた欠損楕円の検出", *信学論(D-II)*, **J73-D-II**, 2, pp. 159-166 (1990-02).
- (6) 與水大和: "Hough 変換に関する最近の研究動向", *情報処理学会研資*, CV51-1, pp. 1-8 (1987-11).
- (7) Illingworth J. and Kittler J.: "A survey of the Hough transform", *Comput. Vision, Graphics & Image Process.*, **44**, pp. 87-116 (1988).
- (8) Merlin P. M. and Farber D. J.: "A parallel mechanism for detecting curves in pictures", *IEEE Trans. Comput.*, **24**, 1, pp. 96-98 (1975).
- (9) 高木幹雄, 鳥脇純一郎, 田村秀行編: "画像処理アルゴリズムの最新動向", *新技術コミュニケーションズ* (1986).
- (10) 渡辺孝志, 鈴木一也: "図形認識のための不連続な線画像の線分近似手法", 第 21 回画像工学コンファレンス論文集, pp. 267-270 (1990).
- (11) 田辺国土: "非線形最小 2 乗法のアルゴリズム", *応用統計学*, **9**, 3, pp. 119-140 (1981).
- (12) Davis L. S.: "Hierarchical generalized Hough transforms and line segment based generalized Hough transforms", *Pattern Recognition*, **15**, 4, pp. 277-285 (1982).
- (13) Neveu C. F., Dyer C. R. and Chin R. J.: "Two-dimensional object recognition using multiresolution models", *Comput. Vision, Graphics & Image Process.*, **34**, pp. 52-65 (1986).

(平成 2 年 10 月 9 日受付, 3 年 2 月 18 日再受付)





**渡辺 孝志**

昭44 東北大・工・通信卒。昭46 同大大学院修士課程了。昭47(株)日立製作所入社。昭55 東北大・大学院博士課程了。工博。同年岩手大・工・情報助手。現在、同助教授。この間、パターンの学習認識、集積回路のCADシステム、セル構造オートマトン、画像処理の研究に従事。情報処理学会、計測自動制御学会、日本リモートセンシング学会各会員。



**石戸橋 真**

昭63 岩手大・工・情報卒。平2 同大大学院修士課程了。同年(株)富士通青森システムエンジニアリング入社。在学中は画像処理の研究に従事。情報処理学会会員。