

# 点配置の一意性を利用した高速なアフィン不変形状認識

堀松 晃<sup>†</sup> 岩村 雅一<sup>†</sup> 黄瀬 浩一<sup>†</sup>

<sup>†</sup> 大阪府立大学大学院工学研究科 〒599-8531 堺市中区学園町 1-1  
E-mail: †horimatsu@m.cs.osakafu-u.ac.jp, ††{masa,kise}@cs.osakafu-u.ac.jp

あらまし 特徴点の配置を利用した物体認識法である Geometric Hashing という手法が提案されている。この手法は、様々な座標系を考慮することで幾何学的変換にロバストである。Geometric Hashing をアフィン変換を受けた文字認識に適用することを考える。しかし、特徴点が多くなると座標系も多くなり、計算量が大きくなってしまふ。そこで、座標系に制約を設けることで計算量を削減する方法を提案する。提案手法を用いた実験を行った結果、アフィン変換を受けた文字を高速に認識出来ることを確認した。

キーワード Geometric Hashing, 点配置の一意性, アフィン不変量, ハッシュ

## Fast Affine Invariant Shape Recognition Using Uniqueness of Arrangement of Points

Akira HORIMATSU<sup>†</sup>, Masakazu IWAMURA<sup>†</sup>, and Koichi KISE<sup>†</sup>

<sup>†</sup> Graduate School of Engineering, Osaka Prefecture University  
1-1 Gakuencho, Naka, Sakai, 599-8531 Japan  
E-mail: †horimatsu@m.cs.osakafu-u.ac.jp, ††{masa,kise}@cs.osakafu-u.ac.jp

**Abstract** Geometric hashing is an object recognition method using arrangements of feature points. This method is a robust retrieval method against geometric transformations by examining all of possible coordinate systems. We use the geometric hashing to recognize characters undergone affine transformation. However, the more feature points are, the more coordinate systems are. Thus computational cost of recognizing characters is large. We propose a method to reduce the computational cost by restricting possible coordinate systems. By experiments, we confirmed that the proposed method can recognize characters faster.

**Key words** Geometric Hashing, Uniqueness of Arrangements of Points, Affine Invariants, Hash

### 1. ま え が き

本稿では、カメラベース文字認識の実時間処理を目指して、高速な文字認識手法について検討する。本研究で想定している文字認識環境は下記の通りである。

- 文字の変形はアフィン歪み

撮像機器にカメラを用いるため、撮像された画像は射影歪みを受ける。しかし、文字領域があまり大きくないと仮定し、射影歪みをアフィン歪みで近似するものとする。

- 文字は連結成分毎に切り出される

照明環境は一様で、ぼけによる劣化は起こらない。文字は欠損なく連結成分毎に抽出される。

上記の条件から、アフィン歪みを受けた連結成分同士を高速に照合できる文字認識手法が求められる。上記の条件はカメラベースの文字認識問題としては比較的容易なものであるが、そ

れでも2次元図形の高速な形状認識を実現するためには解決すべき問題がある。

アフィン変換を受けた連結成分の認識手法として、大きく2種類の方法が考えられる。1つ目の方法は、アフィン変換を受けた図形そのものを正規化し、その後照合する方法である。Leuらは図形を構成する画素の座標値から共分散行列を求め、その逆行列を用いることで拡大・縮小、せん断変形を正規化する方法を提案している[1]。しかし、この方法では回転に任意性を残すため、なんらかの方法で図形の回転を考慮した照合が必要である。この問題に対してHorimatsuらは回転方向の照合を高速化する手法を検討しているが、この手法は十分高速とは言い難い[2]。

もう1つの方は、何らかの方法で図形から安定した特徴点を抽出し、広く知られた特徴点ベースの物体認識手法であるGeometric Hashingを適用する方法である[3]。Geometric

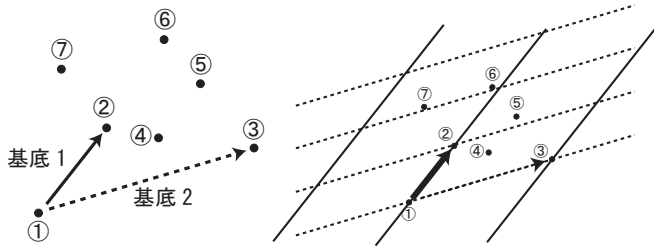


図 1 基底の決定

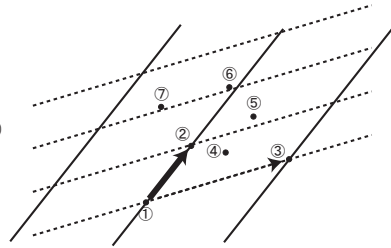


図 2 アフィン不変座標系への射影

Hashing はアフィン歪みを受けた物体の高速な認識を実現可能であるが、(1) アフィン歪みを受けた二値図形から安定な特徴点を算出することは容易でない。(2) 特徴点数を  $P$  としたときの計算量が  $O(P^4)$  と大きいという問題がある。

そこで本稿では、Geometric Hashing の 2 つの問題点を効率良く解決する手法を提案する。提案手法の最大の特長は、図形から幾何学的不変量を計算する原理を逆利用することである。この原理を利用することで、アフィン歪みを受けた二値図形から安定な特徴点を抽出することができる。また、Geometric Hashing の基底の選び方を制限し、最大で計算量を  $O(P^4)$  から  $O(P^2)$  にまで軽減することができる。

まず、Geometric Hashing を用いた形状認識法について述べる。次に、座標系の数を削減した特徴量抽出法を提案し、その方法を用いた認識手法を述べる。そして、関連研究について述べる。提案手法を用いた実験と考察を行い、最後にむすびとする。

## 2. Geometric Hashing [3]

アフィン変換に対応した Geometric Hashing の説明を行う。Geometric Hashing は、特徴点を用いて検索を行う手法である。この手法の処理は、アフィン不変座標系の生成、特徴点の射影、投票処理に分けられる。

最初に、アフィン不変座標系の生成について述べる。例を図 1 に示し、説明を行う。まず、特徴点に対して順番を付ける。次に、この特徴点から順番を考慮して 3 点を選ぶ。ここでは、1 番、2 番、3 番の特徴点を選ぶ。そして、1 番の特徴点を原点、1 番と 2 番の特徴点から基底 1、1 番と 3 番の特徴点から基底 2 を作る。この処理により、1 つのアフィン不変座標系が生成される。次に、特徴点の射影について説明を述べる。

各アフィン不変座標系に対して特徴点の射影を行う。上記の座標系に対して射影する例を図 2 に示す。座標系を生成した 1 番、2 番、3 番以外の全ての特徴点の座標を算出する。括弧は基底 1、基底 2 に対する座標である。各座標系に対して射影する特徴点の数は  $P - 3$  個である。

アフィン不変座標系の生成処理では、特徴点の数を  $P$  とすると、全ての 3 点の組み合わせを試すため、 ${}_P P_3$  通りのアフィン不変座標系が生成される。この処理では、 $O(P^3)$  の計算量がかかる。また、特徴点の射影処理では  $O(P)$  の計算量がかかる。

Geometric Hashing では、識別対象のモデルをハッシュテーブルに登録し、検索の際にハッシュテーブルを参照して投票を行う。登録の処理では、座標系の識別番号とモデルの識別番号

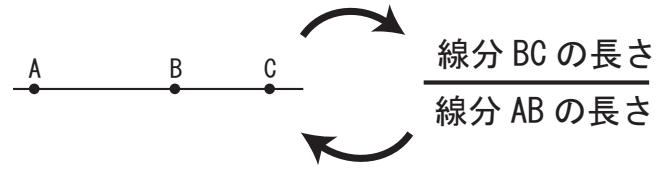


図 3 点配置と不変量

表 1 アフィン変換で保存される性質

性質 1	重心が保存される
性質 2	直線は直線として保存される
性質 3	同一直線上の 3 点を作る線分比は保存される
性質 4	2 つの領域の面積の比は保存される

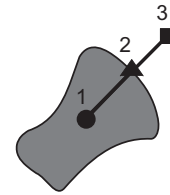


図 4 点配置 1

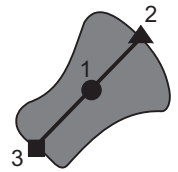


図 5 点配置 2

を組にして特徴点が属するピンに登録する。検索の処理では、特徴点が属するピンを参照して、座標系の識別番号とモデルの識別番号の組に対し投票を行う。最終的に得票数が最も多い座標系の識別番号とモデルの識別番号の組が得られる。この組のモデルの識別番号を検索結果とする。

### 2.1 アフィン変換を受けた形状認識への適用

アフィン変換を受けた形状を認識するために、Geometric Hashing を適用することを考える。ただし、形状は連結成分であるとする。形状の輪郭点を特徴点として適用する。輪郭点の数を  $P$  とすると  $O(P^4)$  の計算量となる。そこで、この計算量を削減する方法を提案する。

### 3. 点配置の一意性を用いた計算量の削減

本稿では、幾何学的不変量を用いて点の配置を決定する方法を提案する。一般的には、点の配置から幾何学的不変量を抽出する。この原理を逆に利用することにより点の配置を一意に決定する。図 3 を用いて説明を行う。同一直線上に 3 点 A, B, C が並んでいるとする。この 3 点の配置によりアフィン不変量である線分比が算出可能である。そこで、アフィン不変量を利用して点の配置を決定する。予め、A, B の点が抽出されているとする。ここで、線分 AB の長さを計算する。そして、AB が通る直線を考え線分比がある値にを決めることにより C の点が一意に決まる。この考えを利用して計算量の削減を行う。

アフィン変換を受けた形状に対し、輪郭点を用いて Geometric Hashing を適用すると認識に時間がかかるという問題がある。この一因は、特徴点から 3 点選んでアフィン不変座標系を生成する数が膨大になるためである。そこで、アフィン変換を受けた形状の性質を用いて座標系の数を削減する。この 3 点の配置を点配置と呼ぶ。ここで用いるアフィン変換で保存される性質を表 1 に示す。

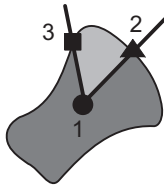


図 6 点配置 3

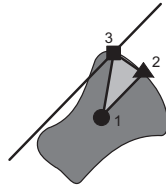


図 7 点配置 4

### 3.1 点配置の一意性 1

表 1 の性質 1, 性質 3 を用いる。性質 1 により, 重心を 1 点目とする。図 4 が示すように, 1, 2 点目の線分に対しての長さを用いて 3 点目を決定する。この方法では, 2 点目を輪郭から抽出すると点配置が一意に決定されるため, 計算量は  $O(P)$  となる。

### 3.2 点配置の一意性 2

表 1 の性質 1, 性質 2, 性質 3 を用いる。性質 1 により, 重心を 1 点目とする。2 点目を輪郭上で任意に選んだ点とする。図 5 が示すように, 性質 2 を用いて 1, 2 点目を結んだ直線と輪郭が交わる点を 3 点目とする。3 点目の候補が複数ある場合は, 性質 3 により最も線分比が大きい (小さい) 点を 3 点目とする。この方法では, 2 点目を輪郭から抽出すると点配置が一意に決定されるため, 3 点を選び不変座標系の生成に必要な計算量は  $O(P)$  となる。

### 3.3 点配置の一意性 3

表 1 の性質 1, 性質 4 を用いる。性質 1 により, 1 点目を決定し, 2, 3 点目を輪郭上とする。図 6 に示すように, 1, 2 点目を通る半直線と 1, 3 点目を通る半直線を考える。性質 4 により, 2 本の半直線で出来た 2 つの領域の面積の比がある値になる点を 3 点目とする。この方法では, 輪郭点を追跡する計算量が  $O(P)$  となるため, 3 点を選ぶ計算量は  $O(P^2)$  となる。

### 3.4 点配置の一意性 4

表 1 の性質 1, 性質 4 を用いる性質 1 により, 1 点目を決定する。2, 3 点目を輪郭上とする。図 7 に示すように, 1, 2 点目を通る線分と 1, 3 点目を通る線分を考える。性質 4 により, 形状全体の面積と 1, 2, 3 点目で出来る三角形の面積比がある値になる点を 3 点目とする。複数候補がある場合には, 2 点目から時計回り (反時計まわり) に回ったときに最初に交差する点とする。この方法では, 直線上の追跡により  $O(T)$  の計算量がかかるため, 3 点を選ぶ計算量は  $O(TP)$  となる。

### 3.5 点配置の一意性 5

表 1 の性質 1, 性質 4 を用いる性質 1 により, 1 点目を決定する。2, 3 点目を輪郭上とする。図 7 と同様の 3 角形を考える。性質 4 により, 形状全体の面積と 1, 2, 3 点目で出来る三角形の面積比が最大になる点を 3 点目とする。この方法では, 輪郭点を追跡する計算量が  $O(P)$  となるため, 全体では  $O(P^2)$  となる。

## 4. 提案手法

3.5 節の点配置の決定方法を用いたアフィン変換に不変な高速形状認識手法を提案する。提案手法の処理の流れを図 8 に示

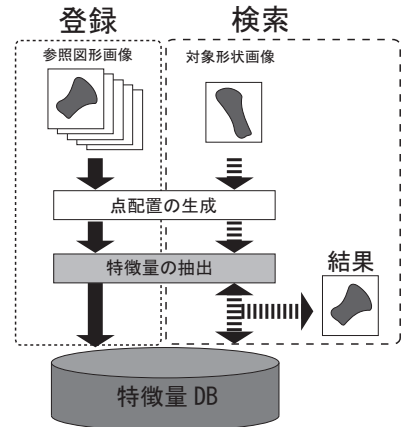


図 8 全体の流れ

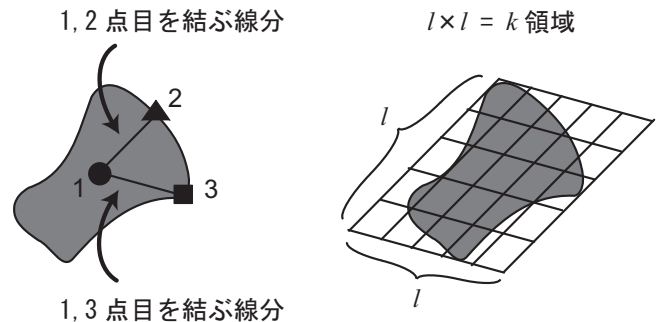


図 9 特徴量の抽出

す。提案する形状認識手法は, 登録処理と検索処理から構成される。登録処理では, 予め用意しておいた参照図形画像それぞれから形状の情報を抽出し特徴量データベースに登録する。検索処理では, 認識対象の形状画像を入力し, 登録処理と同様に形状の情報を抽出する。その情報を特徴量データベースと照合し, その結果を出力する。登録処理と検索処理において, 点配置の生成と特徴量の抽出の処理を共通して行う。以下では, 形状画像において形状が黒, 背景が白の 2 値画像とし, 形状は 1 つの連結成分で構成されるものとする。

### 4.1 点配置の生成・特徴量の抽出

点配置の生成では, 3.5 節で述べた方法を用いる。まず, 2 値の形状画像から重心と輪郭を抽出する。輪郭点の個数を  $P$  とする。点配置の一意性を利用し, アフィン不変座標系が  $P$  通り生成される。

点配置から形状の特徴量を抽出する。図 9(左) のように 1, 2 点目を通る線分と 1, 3 点目を通る線分を考える。これらの線分に平行な線分を用いて対象の形状を平行四辺形で囲む。次に, その平行四辺形の各辺の長さを  $l$  等分するように分割する。図 9(右) では,  $l=5$  の例である。分割される領域数は  $l \times l = k$  となる。形状全体の面積に対する各領域の割合を求める。そして, 各領域の割合を  $n$  段階に離散化し  $r_i$  とする。この処理により,  $k$  次元のベクトルを生成することが出来る。1 つの点配置から 2 点選ぶ組み合わせは  ${}_3P_2 = 3$  個であるので, 各点配置に 3 本の  $k$  次元ベクトルが生成される。この 3 本のベクトルを登録・検索処理に用いる。

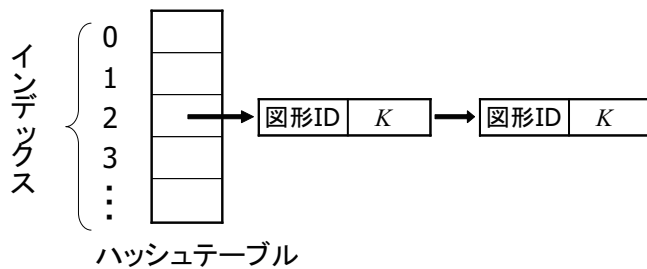


図 10 ハッシュテーブルへの登録

#### 4.2 登録

登録の処理では、予め用意しておいた参照図形画像それぞれの形状情報を特徴量データベースに登録する。特徴量データベースには、高速に検索するためハッシュテーブルを用いる。

参照図形画像から  $P$  個の点配置を生成し、各点配置について 3 本の  $k$  次元ベクトルを得る。各  $k$  次元ベクトルから計算する 1 つのスカラ値  $K$  を算出する。

$$K = 3^m s + \sum_{i=1}^k 3^{i-1} r_i \quad (1)$$

ここで、 $s$  は  $k$  次元ベクトルの種類を表すパラメータであり、 $s = 0, 1, 2$  である。この  $K$  の値を用いてハッシュテーブルのインデックス  $H_{\text{index}}$  を求める。

$$H_{\text{index}} = K \bmod H_{\text{size}} \quad (2)$$

このインデックス  $H_{\text{index}}$  が示すバケットの位置に図形の識別番号である図形 ID と  $K$  の値を組にして登録する。ここで、 $H_{\text{size}}$  はハッシュテーブルのサイズである。また、登録時に衝突が生じた場合は図 10 のようにリスト構造で付け加える。

#### 4.3 検索

対象形状画像から得られた情報を特徴量データベースの情報と照合する。対象形状画像から登録処理と同様に点配置を生成し、各点配置について 3 種類の  $k$  次元ベクトルを得る。各種類の  $k$  次元ベクトルから  $K$  の値を計算し、ハッシュテーブルのインデックス  $H_{\text{index}}$  を求める。 $H_{\text{index}}$  の示すバケットに格納されているリストを得る。そして、リストの各要素について登録されている図形 ID に投票を行なう。ハッシュを用いる場合、 $H_{\text{index}}$  が同じになったからといっても特徴量まで同じとは限らない。そこで、特徴量が一致しない場合の誤投票を防ぐために、 $K$  の値が一致するものだけに投票する。このような処理を各点配置の各種類のベクトルに対して行くと、この対象形状に対する得票数が得られる。

しかし、このようにして得られた得票数をそのまま用いるには問題がある。例えば、アルファベットの「M」や「W」は輪郭点の数が比較的多くなるため、他の文字よりもハッシュテーブルに登録される情報が多くなる。つまり、輪郭点が多い図形に対し不当に多くの投票を行なうことになる。このような投票を補正するために、登録処理における輪郭点の総数  $P$  を用いて以下の式により、投票  $V$  に重み付けを行う。

$$V = \frac{1}{\sqrt{P}} \quad (3)$$

このように重み付けを行った投票を行い、最終的に最大の得票数を得た図形 ID を出力する。

#### 4.4 高速化

検索処理では 2 つのパラメータを変化させることで認識性能と処理時間のバランスを変えることが出来る。対象形状の面積が大きいほど輪郭抽出、特徴量抽出の処理が増加する。そこで、重心と輪郭抽出の処理の前に、対象形状画像から形状の面積を算出する。形状の面積を  $t$  とするように対象形状画像に対し拡大・縮小の変換を行う。この処理により、図形の大きさに依らず高速に認識可能である。また、Geometric Hashing は検索処理を打ち切ることで高速化が可能であることが知られている。提案手法でも、識別に十分な数だけ特徴量の比較を行えば良いため点配置の数を  $u (u \leq P)$  に削減することで高速化が可能である。

### 5. 関連研究

輪郭点を特徴点として利用するアフィン不変な形状認識方法を述べる。まず、Geometric Hashing [3] を用いる方法が考えられる。アフィン変換を含む幾何学的変換に対してロバストに認識可能であるという特徴がある。しかし、輪郭点の数を  $P$  とすると  $O(P^4)$  の計算量がかかる。

Geometric Hashing を一定の条件下で高速化した Locally Likely Arrangement Hashing (LLAH) [4] を適用する方法が考えられる。LLAH は、予め登録した文書画像 10000 枚から射影変換を受けた文書画像を 0.1ms で検索可能な手法である。この手法は、近傍  $m$  個の特徴点から不変量を求め検索に用いる手法である。近傍の特徴点が幾何学的変換を受けても近傍に位置する可能性が高いという条件を用いて Geometric Hashing を高速化している。そこで、輪郭点を特徴点として適用することが考えられる。しかし、近傍の輪郭点の配置に多様性がないため、特徴量に十分な識別性が得られない可能性がある。

また、Rothwell らは射影変換を受けた形状を認識する手法を提案している [5]。この手法は、連結した 5 本の線分や 2 個の円錐曲線を抽出することで、射影変換に不変な特徴点を得る。また、この特徴点には順序が付けられているため、高速に検索が可能である。しかし、アフィン変換を受けた任意の形状から常に線分やコニックを抽出できるとは限らないため適用するのは困難である。

提案手法は Geometric Hashing を連結成分を対象に高速化する手法と捉えることが出来る。提案手法では、アフィン不変座標系の数を削減し高速化を図ることにより、 $O(PQ)$  の計算量で検索可能である。 $Q$  は形状を構成する画素数である。

### 6. 実験

提案手法の有効性を確認するために 3 種類の実験を行った。1 つ目は、アフィン変換を受けた英数字に対しての実験を行った。人工的にアフィン変換を施した英数字画像を認識対象の形状画像として入力し、認識率と認識時間を調べた。2 つ目は、アフィン変換を受けた漢字に対しての実験を行った。アフィン変換を施した漢字画像を認識対象の形状画像とし、認識率と認



図 11 アフィン変換を施した文字の例

表 2 形状が類似している文字

0 O o	6 9	l l	S s	V v
W w	X x	Z z	p d	q b

表 3 区別がつかない文字

7 L	N Z z
-----	-------

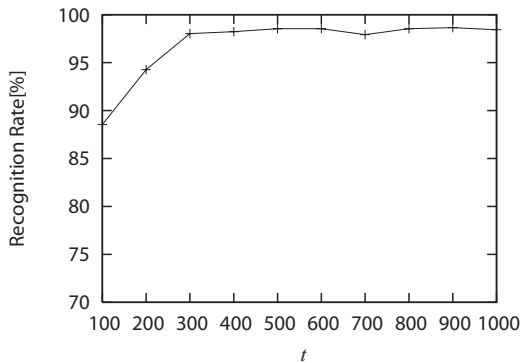


図 12 認識率

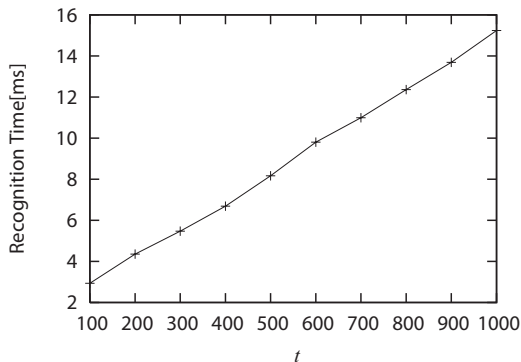


図 13 認識時間

識時間を調べた。3 つ目は、射影変換を受けた英数字に対しての実験を行った。人工的に射影変換を施した英数字画像を認識対象の形状画像とし認識率を調べた。認識率は、対象形状画像数に対して正しい図形 ID を出力した割合とし、認識時間は対象形状画像を入力してから図形 ID を出力するまでの時間とした。計算機は CPU Opteron 2.8Ghz、メモリ 32GByte を用いた。また、 $H_{size} = 2^{18} - 1$  とした。

また、英数字を用いる実験では、表 2 に示す形状が類似している英数字は同一のものとして扱い、複数の輪郭を持つ「i」、 「j」は除いた。

### 6.1 アフィン変換を受けた英数字の認識

アフィン変換を施した英数字に対する認識性能を検証する実験を行った。参照図形画像として、IPA ゴシックフォントの英数字 62 文字の画像を用意した。文字のサイズは「0」が  $51 \times 99$  ピク

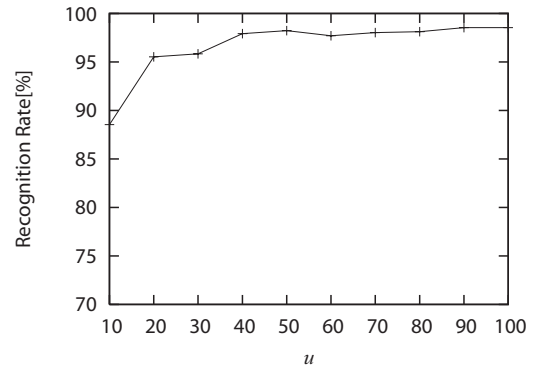


図 14 認識率

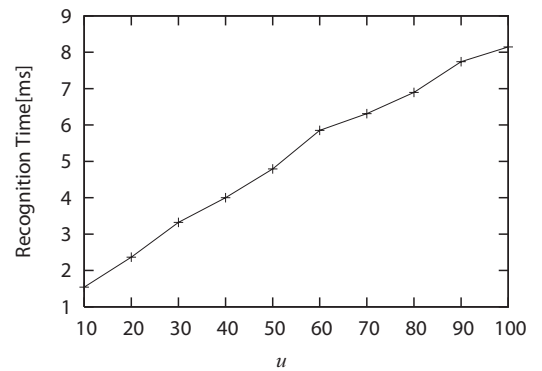


図 15 認識時間

セルとなるポイント数を用いた。対象形状画像として、参照図形画像に対し独立変倍  $\alpha$  ( $\alpha = 1.5, 2, 0$ )、せん断  $\phi$  ( $\phi = -0.5, 0.5$ )、回転  $\theta$  ( $\theta = 2.1, 4.2$ )、拡大・縮小  $\beta$  ( $\beta = -0.8, 1.2$ ) を変化させた  $2^4 = 16$  種類のアフィン変換を施した画像を用いた。アフィン変換行列を  $A$  とすると各パラメータは以下のような対応となっている。

$$A = \begin{bmatrix} \beta & 0 \\ 0 & \beta \end{bmatrix} \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} 1 & \tan(\phi) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha & 0 \\ 0 & 1/\alpha \end{bmatrix} \quad (4)$$

対象形状画像は  $62 \times 16 = 992$  枚である。「A」に対してアフィン変換を施した 4 種類の例を図 11 に示す。 $t, u$  を変化させ認識率と認識時間を調べる実験を行った。その他のパラメータは、 $l = 5$  ( $k = 25$ )、 $n = 2$  とした。

$t$  を 100 から 1000 に 100 ピクセルずつ変化した認識率を図 12、認識時間を図 13 に示す。このとき、 $u = 100$  とした。図 12 が示すように  $t = 100$  まで減少させると認識率が 86.9% に低下した。これは、図形を小さくすると特徴量に 1 画素が与える影響が大きくなり、特徴量が不安定になるためであると考えられる。また、認識率が 98% ほどで停滞するのは、アフィン変換や拡大・縮小による正規化によって劣化が生じるためであると考えられる。 $t = 300$  で 98.0% の認識率、認識時間 5.47ms という結果が得られ、 $t$  を減らすことにより、認識時間を削減出来ることが確認出来た。

検索処理における生成パターン数のパラメータ  $u$  を 10 から 100 に 10 ずつ変化した認識率を図 14、認識時間を図 15 に

# 亜易威委

図 16 単一の連結成分の常用漢字

表 4 漢字を用いた実験結果

認識率 (%)	95.0
認識時間 (ms)	26.0

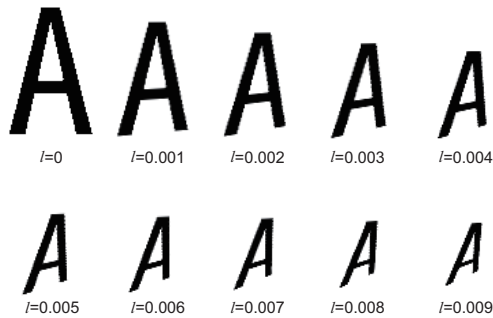


図 17 射影変換を受けた文字の例

示す。このとき、 $t = 500$  とした。図 12 が示すように  $u = 10$  にすると認識率が 81.3% まで低下した。 $u = 40$  で 97.9% の認識率、認識時間 4.00ms という結果が得られ、 $u$  を減らすことにより、認識時間の削減が可能であることが示された。

## 6.2 アフィン変換を受けた漢字の認識

アフィン変換を施した漢字に対する認識性能を検証する実験を行った。参照図形画像として、IPA ゴシックフォントの常用漢字 1945 文字から単一の連結成分で構成される 289 文字の画像を用意した。例を図 17 に示す。フォントとして偏や旁が重なる場合は単一の連結成分として扱った。文字のサイズは、「亜」が  $118 \times 105$  ピクセルとなるポイント数を用いた。対象形状画像として、6.1 と同様に 16 種類のアフィン変換を施した漢字画像を用いた。対象形状画像は  $289 \times 16 = 4624$  枚である。 $t = 500$ 、 $u = 100$  とし認識率と認識時間を調べた。その他のパラメータは、 $l = 7(k = 49)$ 、 $n = 2$ 、 $t = 1000$ 、 $u = 100$  とした。

結果を表 4 に示す。認識時間は 26ms ほどで、95% の認識率であった。登録する参照図形画像が増えても高速に認識可能であることを確認出来た。

## 6.3 射影変換を受けた文字の認識

射影変換を施した英数字に対する認識性能を検証する実験を行った。参照図形画像として、6.1 節で用いた画像を用いた。対象図形画像として、参照図形画像に対し以下の射影変換を施した英数字画像を用いた。透視縮小のパラメータ  $l(l = 0.000 \sim 0.009)$  を 0.001 ずつ変化させた 10 種類の射影変換である。例を図 17 に示す。対象図形画像は  $62 \times 10 = 620$  枚である。 $t = 500$ 、 $u = 100$  とし認識率を調べた。パラメータは、 $l = 4(k = 16)$ 、 $n = 2$  とした。

結果を図 18 に示す。 $l = 0.001$  までは認識率が 100% であったが射影変換の程度が大きくなると大幅に認識率が低下した。形状の劣化とともに特徴量に変化してしまうためであると考え

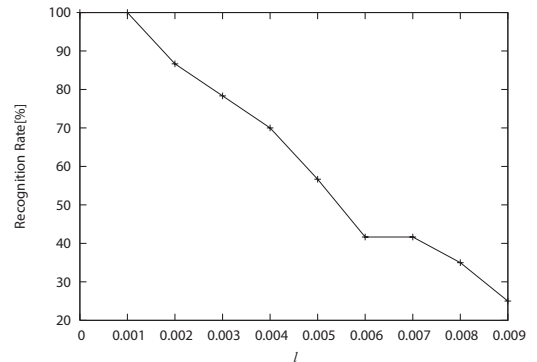


図 18 認識率

られる。

## 7. むすび

Geometric Hashing をアフィン変換を受けた形状認識に適用した場合、座標系を生成する点配置の数が膨大になり計算量が膨らむという問題点があった。そこで、本稿ではアフィン変換を受けた形状の性質を利用し点配置の数を削減する方法を提案した。アフィン変換の前で重心、直線・線分比、面積比が保存される性質により点配置に制約を設けることで数を削減した。また、上述の方法を用いた形状認識法を用いて、アフィン変換への耐性、漢字への適用、射影変換への耐性を調べる実験と考察を行った。

今後の課題としては、特徴量を増やすことが考えられる。3 点の特徴点の配置が決定すれば、4 点目も一意に決定することが出来る。この 4 点目により、格子上に区切る方法が増え特徴量を増加させることが可能である。特徴量の増加により認識率の向上が見込める。

また、射影変換への拡張が考えられる。射影変換に不変な座標系は、4 点の特徴点の配置により生成される。輪郭点数を  $P$  とすると、Geometric Hashing を単純に用いた場合  $O(P^5)$  の計算量がかかる。アフィン変換と同様に、不変量を用いることで点配置の数の削減が期待できる。

謝辞 本研究の一部は、科学研究費補助金 若手研究 (B) 19700177 による。

## 文 献

- [1] J.-G. Leu, "Shape normalization through compacting," Pattern Recognition Letters, vol.10, no.4, pp.243-250, 1989.
- [2] A. Horimatsu, R. Niwa, M. Iwamura, K. Kise, S. Uchida and S. Omachi, "Affine invariant recognition of characters by progressive pruning," Proceedings of the 8th IAPR International Workshop on Document Analysis Systems (DAS2008), pp.237-244, Sept. 2008.
- [3] H. J. Wolfson and I. Rigoutsos, "Geometric hashing: an overview," IEEE Comput. Sci. Eng., vol.4, no.4, pp.10-21, 1997.
- [4] 中居友弘, 黄瀬浩一, 岩村雅一, "特徴点の局所的配置に基づくデジタルカメラを用いた高速文書画像検索," 信学論 D, vol. J89-D, no.9, pp.2045-2054, Sept. 2006.
- [5] C. A. Rothwell, A. Zisserman, D. A. Forsyth and J. L. Mundy, "Using projective invariants for constant time library indexing in model based vision," Proc. British Machine Vision Conference, 1991.