

## 相似変換により不変なコードと Shape Matching への応用について

吉田英司 三田誠一

豊田工業大学 〒468-8511 名古屋市天白区久方 2-12-1

E-mail: eyoshida@toyota-ti.ac.jp, smita@toyota-ti.ac.jp

**あらまし** 本稿では二次元において相似変換(平行移動, 回転, 拡大縮小)により不変な性質を持つコードを利用して, お互い相似な関係にあるものや, 大部分の形状が類似している(部分的な欠損, 変形がある場合)二物体の類似度を比較する方法を説明する. 提案手法では物体の輪郭の曲率を基にして物体のコード化を行い, 類似度を比較可能なものにする. また, 曲率を利用する方法としてよく知られているCSS (curvature scale space) との比較を行う. その中で, 回転, 拡大縮小に対する評価実験を行い, 本提案が認識率を維持しつつ計算時間を短縮できることを明らかにした. つづいて部分的な形状の欠損, 変形がある場合にはShape context 法と性能比較を行い, より高い認識率が得られることを示した.

**キーワード** shape matching, コード化, 曲率, shape classification,

### Invariant codes for similar transformations and its application to Shape Matching

Eiji Yoshida, Seiichi Mita

Toyota Technological Institute 2-12-1 Hisakata, Tempaku, Nagoya, 468-8511, Japan

E-mail: eyoshida@toyota-ti.ac.jp, smita@toyota-ti.ac.jp

**Abstract** In this paper, we propose a new method of shape matching. Our proposed method encodes the contour of an object by using the curvature of an object. If two objects are similar (translation, rotation and scaling) in shape, these codes themselves or their cyclic shift have the same values. We have compared our method with other methods such as Fourier descriptor, CSS(curvature scale space) and shape context. We have shown that the computational cost of our method is lower than that of CSS and moreover the recognition rate of our method is nearly the same as CSS.

**Keyword** shape matching, invariant code, curvature, shape classification

#### 1. はじめに

二次元の物体の Shape Matching については, 現在に至るまで様々な研究が行われている. 特に相似な2物体のマッチング(図1), 部分的な欠損, 変形, ノイズはあるが輪郭形状の大部分が一致している場合(図2)のマッチング手法は多数ある. その中でも代表的なものとしてフーリエ記述子([7]), モーメント不変量([6]), CSS(curvature scale space)([1], [2]), shape context([5]) 等がある. どの手法も図形認識において信頼性の高い手法である. しかしながら, 各手法とも問題点が無い訳ではない. 例えば, フーリエ記述子, モーメント不変量に関しては, 形状が全く異なる場合でも同一の形状であるとみなす場合がある. CSS に関しては膨大な計算量が必要となる.

そこで今回はこれらの問題点を補うことに着目し, 特に CSS との比較を中心に考えた. そのために物体の

曲率の情報を利用して物体をコード化する手法を提案する. このコードは相似変換(平行移動, 回転, 拡大縮小)について不変(循環した関係)であり, 2物体のコードが循環した関係にあるならば, それらの形状も類似なものとなる. 又, CSS と比較した場合計算時間は大幅に削減できる. 更に認識率の比較のために MPEG 7\_CE-Shape1([2], [4]) を利用して回転, 拡大縮小についての実験を行った. その中で, CSS と比較してもほぼ同等の認識率を得た. つづいて部分的な形状の欠損, 変形がある場合についての評価実験を行った. この実験では shape context 法との比較を行った. 以後これらについての説明を行う.

第2章で比較対象であるCSS法の概要について述べる. 第3章では提案手法であるコード化手法について説明する. 第4章では比較実験を行い, 第5章で結論を説明する.



図1. 互いに相似 図2. 形状の大部分が一致

## 2. CSS法の概要

本章では比較対象であるCSSの概要について、提案手法との違いを明確にするために簡単に説明する。CSSとはFarzin Mokhtarian等([1], [2])によるShape Matchingにおける手法の一つで、曲率尺度空間とも呼ばれる。Shape Matchingについての手法は様々なものが存在するが、CSSはその中でも効果の高い手法である。特にノイズのある画像への対応に効果を発揮する。CSS法では大まかには以下のことを行う。

- ① パラメータ表示された物体の輪郭を表す座標全体

$$\{(x(t), y(t)) | t \in [0, 1]\}$$

について、次のようにガウスフィルタを使って平滑化する。すなわち

$$X(t, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \int_0^1 e^{-\frac{(u-t)^2}{2\sigma^2}} x(u) du \quad (1)$$

$$Y(t, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \int_0^1 e^{-\frac{(u-t)^2}{2\sigma^2}} y(u) du \quad (2)$$

$X(t, \sigma), Y(t, \sigma)$  はそれぞれ  $x(t), y(t)$  をガウス関数の線形結合により近似したものとなる。これらは分散値  $\sigma$  の大きさにより近似の精度が決まる。特に分散値を大きくしていくと、輪郭の近似座標

$$\{(X(t, \sigma), Y(t, \sigma)) | t \in [0, 1]\}$$

による物体の概形はいずれ凸型へと変化していく(図3)。

- ② ①で計算した  $X(t, \sigma), Y(t, \sigma)$  から曲率を計算する。曲率は以下の式で表される([1], [2])。

$$\kappa(t, \sigma) = \frac{\dot{X}(t, \sigma)\ddot{Y}(t, \sigma) - \ddot{X}(t, \sigma)\dot{Y}(t, \sigma)}{\sqrt{(\dot{X}(t, \sigma)^2 + \dot{Y}(t, \sigma)^2)^3}} \quad (3)$$

ここで  $\dot{X}, \dot{Y}$  はそれぞれ  $X, Y$  のパラメータ  $t$  による第一回導関数を表し、 $\ddot{X}, \ddot{Y}$  は第二回導関数を表す。 $\dot{X}, \dot{Y}, \ddot{X}, \ddot{Y}$  は(1),(2)を変形して次で表すことが出来る。ガウスフィルタを利用する理由は曲率計算の際に導関数を計算する必要があることもある。

$$\dot{X}(t, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \int_0^1 \frac{u-t}{\sigma^2} e^{-\frac{(u-t)^2}{2\sigma^2}} x(u) du \quad (4)$$

$$\dot{Y}(t, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \int_0^1 \frac{u-t}{\sigma^2} e^{-\frac{(u-t)^2}{2\sigma^2}} y(u) du \quad (5)$$

$$\ddot{X}(t, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \int_0^1 \frac{1}{\sigma^2} \left(-1 + \frac{(u-t)^2}{\sigma^2}\right) e^{-\frac{(u-t)^2}{2\sigma^2}} x(u) du \quad (6)$$

$$\ddot{Y}(t, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \int_0^1 \frac{1}{\sigma^2} \left(-1 + \frac{(u-t)^2}{\sigma^2}\right) e^{-\frac{(u-t)^2}{2\sigma^2}} y(u) du \quad (7)$$

- ③ (3)式の

$$\kappa(t, \sigma) = 0$$

となる点  $(t, \sigma)$  を抽出する。

- ④ ①~③の操作を各分散値について行う。この操作は輪郭の近似座標

$$\{(X(t, \sigma), Y(t, \sigma)) | t \in [0, 1]\}$$

による物体の概形が凸型(曲率がゼロにならない)になることにより、有限回で終了する。

- ⑤ ①~④の操作から  $\kappa(t, \sigma) = 0$  のグラフを作成する(図4)。2物体を比較する際にはこのグラフの極値を検出して比較する。

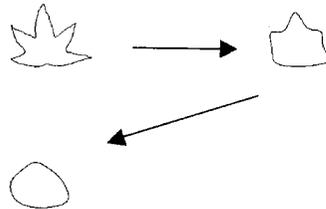


図3. 分散値の変化による物体の変化

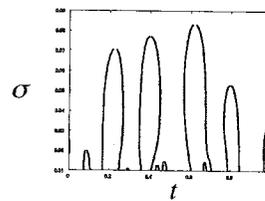


図4. CSSのグラフ

CSSの特長は図4のようなグラフが相似変換により不変な性質を持つ(回転の場合グラフが平行移動する)のと、ノイズのある物体でも上のグラフはほぼ変化しないことである。ただしこのグラフを作成するためには④の操作により膨大な時間が必要となることが分かる(形状次第だが、大抵の場合グラフ作成の為に100回の繰り返しは必要)。

### 3. 提案手法について

本章では提案手法について述べる。第2章で説明したように、CSSは信頼性の高い方法であるが、多くの計算時間を要する。そこでわれわれはCSSのメリットを出来る限り残しつつ、計算時間の短縮を行うことを目的とした。その一つの方法としてコード化手法を提案する。まずはわれわれの手法でも曲率を利用するので、本題に入る前に曲率について少し説明する。

#### 3-1. 曲率について

曲率は第2章の(3)式で表されるものであるが、数学的性質として座標の回転変換、平行移動に関して不変な性質を持つ。又、拡大縮小に関しても定数倍変化するのみである。しかしながら実際の画像に関しては量子化誤差、ノイズによる誤差から完全に不変になるというわけではなく、曲率のグラフの値が変化してしまう場合は大いに考えられる。よって曲率の値をマッチングにそのまま利用するのは得策ではない。ただ、グラフの大まかな概形、特に極値の位置には大きな変動はない。図5がお互い相似な関係にある2物体の曲率のグラフである。

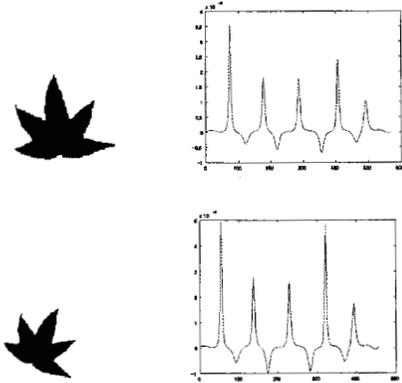


図5. 互いに相似な関係にある2物体とそれらの曲率のグラフ

このように各点での曲率の値は異なるが、グラフの大まかな概形はほぼ同じ形になる。この性質を利用するのが提案手法で、これが計算時間を削減するためのポイントとなる。

#### 3-2. 提案手法について

提案手法では以下の手順で物体のコード化を行う。大まかに説明すれば、曲率の極値を抽出してそれらの点を基にして三角形をつくり、物体を三角形の枠で近似するようなことを行う(図6)。

- I. パラメータ表示された物体の曲率を計算する。つまり第2章の操作①、②を行う。分散値は大体0.01~0.025をとる。
- II. 曲率のグラフから極値を検出し、その極値に対

応する輪郭上の点を検出する(今後これらの点を輪郭上の特徴点と呼ぶことにする)。

- III.  $\{p_1, p_2, p_3, \dots, p_n\}$  を  $\Pi$  で得られた輪郭上の点とする。このとき、下の行列を物体のコードとする。

$$\begin{pmatrix} c_{11} & c_{21} & c_{31} & \dots & c_{n1} \\ c_{12} & c_{22} & c_{32} & \dots & c_{n2} \\ c_{13} & c_{23} & c_{33} & \dots & c_{n3} \\ c_{14} & c_{24} & c_{34} & \dots & c_{n4} \end{pmatrix}$$

ここで  $p_i p_{i+1}$  を線分  $p_i p_{i+1}$  ( $i=1, 2, 3, \dots, n$ ) の長さとし、

$$c_{i1} = \frac{p_{i-1} p_i}{p_{i-1} p_i + p_i p_{i+1} + p_{i-1} p_{i+1}}, \quad (8)$$

$$c_{i2} = \frac{p_i p_{i+1}}{p_{i-1} p_i + p_i p_{i+1} + p_{i-1} p_{i+1}}, \quad (9)$$

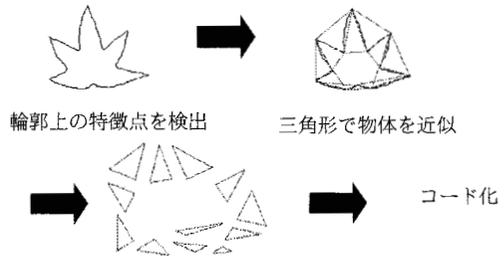
で表す。

又、 $l_i$  を三角形  $p_{i-1} p_i p_{i+1}$  ( $i=1$  の時は  $p_n p_1 p_2$ ,  $i=n$  の時は  $p_{n-1} p_n p_1$ ) の面積、 $l = \sum_{i=1}^n l_i$

として、

$$c_{i3} = \frac{l_i}{l} \quad (10)$$

とする。 $c_{i4}$  は  $p_i$  に対応する曲率の符号がプラスならば0、それ以外は1をとるものとする。



半時計回りに三角形を並べ、順番に辺、面積、凹凸の情報を数値で表す。

図6. コード化の流れのイメージ図

このコードは各列が三角形の情報を表している。我々の手法ではこのコードを利用して2つの物体の類似度を測る。

このコードの特長として、お互い相似(平行移動、回転、拡大縮小)な関係を持つ2物体については、これらのコードはお互いの列が循環した関係になっている。つまり一方のコードの列を巡回シフトすれば、他方の

コードに一致する。この性質は3-1で説明したように、曲率の極値の位置関係は同じになるので、極値と対応する輪郭上の点の位置関係も同じものになることに起因する。そして、これらの点から三角形を作成した場合、各三角形の形状とそれらの順番は、スタート地点を入れ替えれば一致することが分かる。

更にこのコードは2物体がもし同じコードを持つならば、それら2物体の形状も類似なものになる。これによって本提案はフーリエ記述子やモーメント不変量の欠点である形状が全く異なるものを同じものであるとみなすことはない。

### 3-3. 提案手法による物体の比較方法

本章では前章で定義したコードを利用して、どのように物体の類似度を比較するのかについて説明する。

コードを利用した物体の比較方法については以下の方法で行う。

$A = \{p_1, p_2, \dots, p_m\}, B = \{q_1, q_2, \dots, q_n\}$  をある2物体の輪郭上の特徴点全体の集合とする。

- ①  $p_1, p_2, \dots, p_m, q_1, q_2, \dots, q_n$  の対応する曲率の極値の絶対値の大きさを調べる。
- ② もし  $m > n$  ならば、曲率の絶対値の大きさが  $n+1$  番目から  $m$  番目までの値に対応する特徴点を取り除く。
- ③ ②で特徴点の個数を揃えてから  $A, B$  それぞれのコードを作成する。

④

$$CA = \begin{pmatrix} ca_{11} & ca_{21} & ca_{31} & \dots & ca_{n1} \\ ca_{12} & ca_{22} & ca_{32} & \dots & ca_{n2} \\ ca_{13} & ca_{23} & ca_{33} & \dots & ca_{n3} \\ ca_{14} & ca_{24} & ca_{34} & \dots & ca_{n4} \end{pmatrix}$$

$$CB = \begin{pmatrix} cb_{11} & cb_{21} & cb_{31} & \dots & cb_{n1} \\ cb_{12} & cb_{22} & cb_{32} & \dots & cb_{n2} \\ cb_{13} & cb_{23} & cb_{33} & \dots & cb_{n3} \\ cb_{14} & cb_{24} & cb_{34} & \dots & cb_{n4} \end{pmatrix}$$

をそれぞれ  $A, B$  のコードとする(各成分の意味は3-2の定義と同様のものとする)。このとき次が成り立てば、 $CA$  の  $i$  列目と  $CB$  の  $j$  列目が重なったと判断する。

- a.  $|ca_{i1} - cb_{j1}| < t_1, |ca_{i2} - cb_{j2}| < t_1$   
( $t_1$  は 0.1 程度)
- b.  $\begin{cases} 1/t_2 < ca_{i3}/cb_{j3} < t_2 & (cb_{j3} \neq 0) \\ ca_{i3} - cb_{j3} = 0 & (else) \end{cases}$   
( $t_2$  は 2~2.5 程度)

$$c. cb_{i4} - cb_{j4} = 0$$

そして、上の  $a, b, c$  を基にして次のような成分を持つ  $n$  次正方行列  $COM = (com_{ij})$  を作成する。

$$com_{ij} = \begin{cases} 1 & CA \text{ の } i \text{ 列と } CB \text{ の } j \text{ 列が重なる。} \\ 0 & \text{その他} \end{cases}$$

⑤ 次の数を類似度を表す指標とし、計算する。

$$S = S_1 + S_2 + S_3 + S_4$$

ここで

$$s_1 = 1 - \frac{1}{n} \# \{i | \exists j = 1, 2, \dots, n \text{ s.t. } com_{ij} = 1\}$$

$$s_2 = 1 - \frac{1}{n} \max \left\{ \sum_{i=1}^n com_{i u_i} \mid (u_1, u_2, \dots, u_n) \in U \right\}$$

$$(U = \{(i, i+1, i+2, \dots, n, \dots, i-1) \mid i = 2, \dots, n\} \cup \{(1, 2, 3, \dots, n)\})$$

$$s_3 = \sum_{i=1}^n s_3^{(i)}$$

$$s_3^{(i)} = \begin{cases} \min \{ |ca_{i3} - cb_{j3}| \mid com_{ij} = 1 \} & (\text{if } \sum_{i=1}^n com_{ij} \neq 0) \\ |ca_{i3} - cb_{u_i3}| & (else) \end{cases}$$

$$\left( \sum_{i=1}^n com_{i u_i} = \max \left\{ \sum_{i=1}^n com_{i u_i} \mid (u_1, u_2, \dots, u_n) \in U \right\} \right)$$

$$s_4 = \frac{m-n}{m+n}$$

- ⑥  $1 \leq i \leq t_3 - 1$  なるすべての  $i$  について、曲率の絶対値の大きさが  $n-i$  番目より小さい点を  $A, B$  から取り除き、③~⑤の操作を行う。 $t_3$  は大体 0.7n から 0.8n 位をとる。
- ⑦ ⑥までで求めた類似度  $S$  の平均値を計算し、その値を物体  $A, B$  の類似度とする。

上で述べた類似度を測る尺度である  $s, s_1, s_2, s_3, s_4$  の意味について少し説明を加えておく。まず  $s$  は値が小さいほど、類似度が大きいという性質を持つ。 $s_1$  はどのくらい列が重なっているかを調べる。言い換えれば、2物体の中でどのくらい類似な形状を持つ三角形が含まれているのを見る。但しこれだけでは形状が異なる場合でも  $s_1$  の値は小さくなる可能性がある。次に  $s_2$  はどのくらい連続して列が重なっているかを調べる。この値は2物体の形状が類似していない限り、小さな値を示さない。 $s_3$  は重なった部分の面積の違いをみるものである。これによりたとえ  $s_1$  が小さくても、

ある場所で形状が大きく異なる部分があれば、 $s_3$ の値は大きくなる。 $s_4$ は  $m$  と  $n$  が大きく離れていれば、値は大きくなる。これによって形状が大きく異なる場合は類似度が小さくなる。尚、提案手法では他にも類似度をはかるパラメータを利用しているが、紙面の都合上割愛する。次章で行う実験ではそのパラメータを利用するが、パラメータを追加する場合と追加しない場合では、認識率の差はせいぜい1%弱である。

操作⑥の意味は特徴点の個数を減らしてそれぞれ比較することにより、形状の近いものは類似度が安定して高い値を示し、大きく異なるものは類似度が小さくなる仕組みからくる。

#### 4. 性能実験について

本章では回転、拡大縮小の認識実験と部分的な欠損、変形がある場合の実験を行い、それぞれフーリエ記述子、CSS との比較、shape context との比較を行う。

##### 4-1. 回転、拡大縮小の認識実験

実験データには標準的なデータベースである MPEG7\_CE-Shape1([2],[4])を利用した。このデータベースは各クラスが20種類の画像を含む、計70クラス1400枚の画像から成る。以下がその画像の例である。



図7. データベースの画像の例

今回は各クラスから1枚ずつ合計70枚の画像を選び、回転と拡大縮小に関する評価実験を行った。

##### (回転実験)

各クラスの一枚選んだ画像について、 $0^\circ$ ,  $9^\circ$ ,  $36^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $150^\circ$  回転した画像を作成し、1クラス6枚の画像からなる計420枚のデータを作成する。そして、各画像について420枚全画像との類似度を比較し、類似度の順位を調べる。そして1番目から6番目の中に何枚同一クラスの画像が含まれているか調べる。この場合  $420 \times 6 = 2520$  が全体の数であり、認識率は各画像における正解数の総和を2520で割ったものにより得られる。



図8. 回転変換された画像の例

##### (拡大縮小実験)

各クラスの一枚選んだ画像について、縮尺が0.1

倍、0.2倍、0.25倍、0.3倍、1倍、2倍の計6枚の画像を用意し、全体で420枚のデータを作成する。そして、各画像について、420枚全画像との類似度を比較し、類似度の順位を調べる。そして1番目から6番目の中に何枚同一クラスの画像が含まれているか調べる。認識率に関しては回転実験と同様である。

回転、拡大縮小の認識実験の実験結果は以下の様になる。

	フーリエ	CSS([4])	提案手法
回転	100%	100%	100%
拡大縮小	86.35%	89.76%	90.40%

表1. 回転、拡大縮小の実験結果

拡大縮小の結果が全体的に低いのは、縮尺を0.1にした場合、原画像のサイズが小さいと画像の情報がほとんど無くなってしまいうためである。原画像のサイズが  $400 \times 400$  以下だと0.1に縮小すると原型をとどめていない場合が多い(図9)。我々の実験では分散値を0.22で与えた。拡大縮小実験に関しては分散値や他の閾値の影響で認識率に1~2%の変動はあるが、それでもCSSと結果に違いはあまりないことが表から分かる。



図9. 原型が無くなってしまいう例

##### 4-2. 部分的な欠損、変形に関する実験

実験データ([5])は各クラス11枚の画像を含む9クラス計99枚の画像から成る。以下はデータに含まれる画像の一例である。



図10. データベースに含まれる画像の一例

図10から分かるようにこのデータ内の画像は部分的な欠損や変形がある画像からなる。

実験では各画像について 99 枚全画像と比較し、類似度を計算して、上位 11 位以内に何枚同一クラスの画像が含まれているかを調べる。この場合可能な数は  $11 \times 99 = 1089$  なので、各クラスの正解数の総和を 1089 で割ったものが認識率となる。次の表が実験結果である。

	Shape context	提案手法
認識率	76. 86%	81. 82%

表 2. 実験結果

この実験ではガウスフィルタをかける際の分散値は 0.01 とした。これも閾値の調整で 2~3%の変動はある。認識率が 80%台になる理由として、次のような画像が含まれていることが挙げられる。上下の画像は共に同一クラスの画像である。

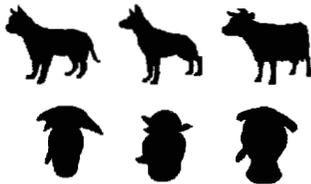


図 1 1. 同一クラスの画像だが認識の困難な例。

これらの画像は形状が類似であるとは言い難い。図 1 0 のような場合は高い類似度を示すので、認識できる可能性が高い。

#### 4-3. 計算量について

計算量については CSS では分散値を変更しながら平滑化を行い、図 4 のようなグラフを作る。それに対し、提案手法では平滑化を行うのは一回のみである。このことから計算時間の大幅な短縮が実現可能である。実際サイズが  $148 \times 224$  の図形(図 1~6 で利用されている葉っぱの画像)について、matlab で作成した提案手法で行う曲率計算と、CSS のグラフ作成のプログラムの平均実行時間(CPU は pentium(R) D 3.2GHz)を比較すると、CSS では大体 150~160 秒かかるのに対し、提案手法ではその 100 分の 1(1.4~1.5 秒)で済む。

2 物体の特徴点を抽出した後のコードの比較にかかる計算オーダーは  $O(n^3)$  ( $n$ : 2 物体の特徴点の個数の少ない方の個数)である。matlab (CPU は上と同様)のプログラムによる平均所要時間は約 0.2~0.25 秒なので負担は少ない。フーリエ記述子と提案手法を比較した場合、高速フーリエ変換を利用すれば提案手法より計算時間は短くて済む。但し、拡大縮小実験での認識率では提案手法の方が上回る。

#### 4-4. ノイズのある画像についての検討

ノイズのある画像への対応に関しては、まだ明確な答えが得られているわけではないが、予め分散値を大きめに取って、曲率のグラフをノイズのある状態でも滑らかな形状にすることは可能である。また B-スプ

ライン法などを利用して物体を近似する方法も考えられる。

## 5. まとめ

本稿では物体の輪郭形状を曲率の極値を基にしてコード化し、相似な関係を持つ 2 物体や、大部分の形状が一致している場合の類似度の比較手法を提案した。この手法は、相似な形状に関するマッチング、部分的な欠損、変形がある場合のマッチングに高い効果を示すことが比較実験により得られた。特に CSS との比較では認識率を維持しつつ、計算時間を短縮できることが示された。今後は部分的な欠損、変形がある場合についてもフーリエ記述子、CSS との比較実験を行っていきたい。

### 謝辞

本研究は、豊田工業大学先端知能システム・デバイス統合センター(私立大学学術研究高度化推進事業)の援助のもとで行われた。

### 文 献

- [1] F. Mokhtarian and A. K. Mackworth, A theory of multi-scale, curvature based shape representation for planar curves, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 14, no. 8, pp. 789-805, 1992.
- [2] F. Mokhtarian and M. Bober, Curvature scale space Representation: Theory, Applications, and MPEG-7 Standardization, Computational Imaging and Vision, Vol. 25, Kluwer Academic Publishers, 2003.
- [3] L. J. Latecki and R. Lakamper, Shape similarity measure based on correspondence of visual parts, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 22, no. 10, pp. 1185-1190, 2000.
- [4] L.J.Latecki, R.Lakamper and U.Eckhardt, Shape Descriptors for Non-rigid Shapes with a Single Closed Contour, CVPR, 2000.
- [5] S. Belongie, J. Malik, and J. Puzicha, Shape matching and object recognition using shape context, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 24, pp. 509-522, 2002.
- [6] A. Khotanzad and Y. H. Hong, Invariant image recognition by Zernike moments, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 12, no. 5, pp. 489-497, 1990.
- [7] C. T. Zahn and R. Z. Roskies, Fourier descriptors for plane closed curves, IEEE Trans. on Computers, C-21, pp. 269-281, 1972.