ディジタル図形の相似及びアフィン不変認識に関する研究

相良, 哲生

https://doi.org/10.11501/3181887

出版情報：九州芸術工科大学, 2000, 博士（工学）, 課程博士

第5章

多値画像における相似不変認識

本章では，自己距離関数を改良し，これを多値画像に適用した特徴抽出法を用いてディジタル図形を認識する方法を提案する。本章で提案する特徴は，ディジタル図形の“中心距離関数”という概念にもとづいており，単連結でないディジタル図形にも適用できる。この中心距離関数の違いを測ることによって，相似不変（シフト・回転・スケール不変）な図形の認識を行うことができる。この中心距離関数を用いた認識実験結果も示す。

5.1 まえがき

第3章において，自己距離関数という新しい概念を用いた，シフト・スケール・回転に不変（相似不変）であり，非単連結なディジタル2値図形に適用可能な認識方法を述べた。自己距離関数は，図形を構成する点の全ての2点の組み合わせで得られた距離の分布関数の一種と考えられる。したがって，n 点から成る図形の自己距離関数を求めるためには，n の二乗のオーダーの計算が必要である。そこで，計算量の削減を目的とし，中心距離関数という自己距離関数の改良版を考案し，この改良した関数を用いた方法を提案した[74]。中心距離関数法では，図形を構成する全ての点

と，図形の重心点との間の距離を用いる。このことにより，n 点から成る図形の中心距離関数を求めるのに必要な計算量は n の一乗のオーダーで済む。本章では，これまで 2 値図形に用いていた中心距離関数法を多階調の図形（多値図形）に対しても適用できるように拡張する。

5.2 ディジタル多値図形

ディジタル多値図形は，有限個の 3 次元空間上の点の集合として示される。この 3 次元空間は，通常の 2 次元平面に対して，点の階調値を示す第 3 の座標軸を新たに加えた 3 次元空間である。

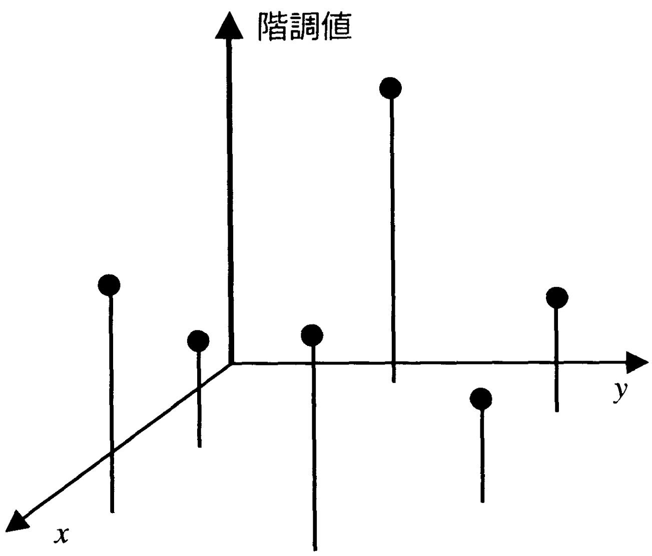

図 5.1　3 次元空間上でのディジタル多値図形

本論文では，ディジタル多値図形を次のように定義する。

［定義 1］　ディジタル多値図形とは，有限個の 3 次元の点の集合である。ただし，点の座標値は任意の実数値をとるものとする。

ディジタル多値図形を表す点の集合を D とすれば，

$$D = \{(x_i, y_i, z_i) \mid i = 1, \ldots, n\},\ x_i, y_i, z_i \in R \quad (式\ 5.1)$$

である。ここで，R は実数の集合を表す。また，z_i は，点(x_i, y_i) における階調値を表す。

通常の画素で表現されるディジタル多値図形は式 5.1 において，

$$x_i, y_i, z_i \in N \quad (式\ 5.2)$$

であり，式 5.1 の特別な場合である。ただし N は整数の集合を表す。
このように，図形を点の集合と定義したので，点の "順序" や "つながり" に関する情報は一切捨象される。すなわち，

$$D = \{(x_i, y_i, z_i) \mid i = 1, \ldots, n\}$$
$$\equiv \{(x_{p_i}, y_{p_i}, z_{p_i}) \mid i = 1, \ldots, n\} \quad (式\ 5.3)$$

ただし，$(p_1, p_2, \cdots, p_n)$ は$(1, 2, \cdots, n)$の任意の順列である。

3 次元空間上における，中心距離関数を定義する。反転した多値図形どうしや，階調値がシフトした多値図形どうしを識別するために，図形の重心を次のように定義する。

［定義 2］ ディジタル多値図形の重心は，xy 平面において平均となる点とする。

g をディジタル多値図形における重心としよう。

$$g = (\bar{x}, \bar{y}, 0)$$
$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \quad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \quad (式\ 5.4)$$

次に，階調値を示す軸を，重心 g と各点$(x_i, y_i, 0)$との間のユークリッド距離の平均値にあわせスケーリングする。これは，ディジタル多値図形を三次元空間上へマッピングする際の正規化に相当する。

［定義 3］　ディジタル多値図形を構成するすべての点と重心との間のユークリッド距離を測り，そのヒストグラムを求める，このヒストグラムにおける距離の最大値を 1 に正規化したものを中心距離密度と呼ぶ。

［定義 4］　中心距離密度を累積表現し，その最大値（距離 1 のところに現れる）を 1 としたものを中心距離関数（累積分布）と呼ぶ。

もし，ディジタル多値画像における重心を$(\bar{x}, \bar{y}, \bar{z})$と定義した場合，階調値が反転した図形どうしや，階調値がシフトした図形どうしはそれぞれ同じ中心距離関数を持つこととなり，これらの図形は同一視される。

ディジタル多値図形には次の性質がある。

［性質 1］　互いに相似なディジタル多値図形（鏡像も含む）の中心距離関数は等しい。

実際には，測定誤差の吸収と情報圧縮のために，定義 3 における区間[0, 1]を M 等分して，各部分区間に落ちる中心距離密度の累積分布を用いる。この場合は M 次元ベクトルが定まるが，これを中心距離ベクトルと呼ぶ。この M の値が大きくなるほど元の多値図形との対応が精密になるが，この M の値をどうすればよいかは，情報圧縮との兼ね合いである。

一例として，図 5.2のディジタル多値図形の M =100 とした場合の中心距離関数を図 5.3に示す。

図 5.2 ディジタル多値図形

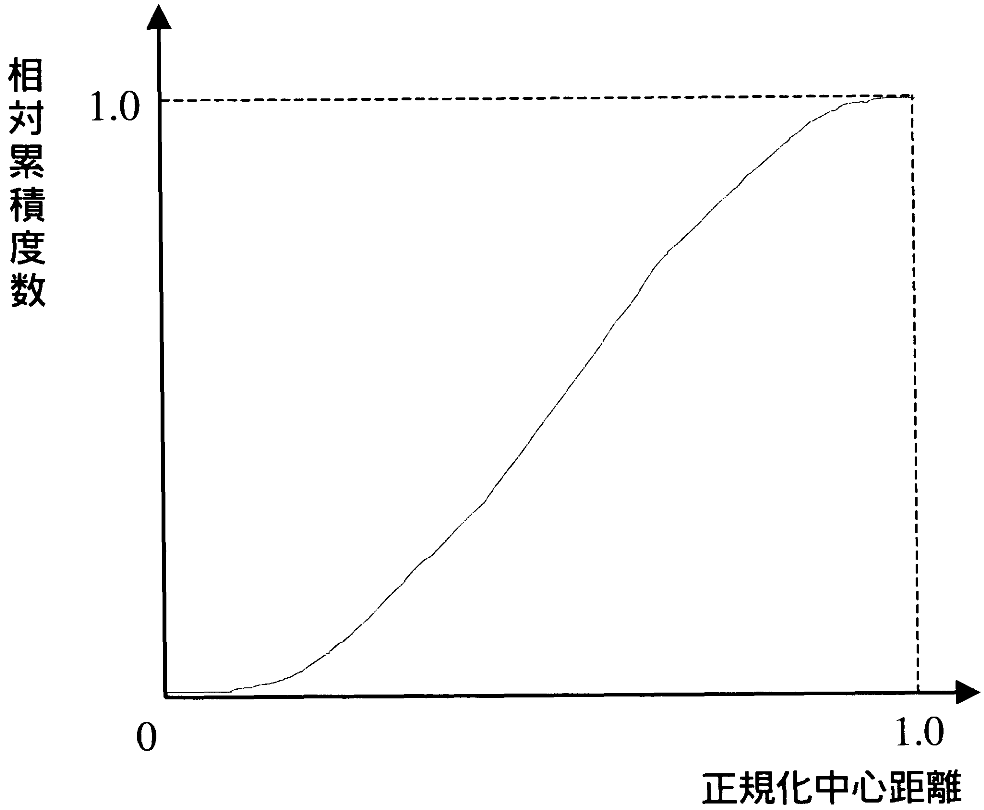

図 5.3 中心距離関数

理解を容易にするために，以上の定義等に関することを若干数学的に表現しておこう。中心距離密度を関数と考えて $f(x)$とすると，

$$f(x)=\sum_{i=1}^{P} a_i\delta(x-d_i),\quad 0\leqq x, d_i \leqq 1 \qquad (式\ 5.5)$$

ただし，P は距離の種類の数，a_i は具体的な距離 d_i の度数である。なお，ディジタル図形を構成する点の数を n とすると，

$$\sum_{i=1}^{P} a_i = n \qquad \text{(式 5.6)}$$

という関係にある。式 5.5 中の $\delta(x-d_i)$ はデルタ関数であり，次の性質を持つ。

$$\delta(x-d_i)=0, \quad x \neq d_i \qquad \text{(式 5.7)}$$

$$\int_0^x \delta(y-d_i)dy = \begin{cases} 1, & x \geqq d_i \\ 0, & \text{その他} \end{cases} \qquad \text{(式 5.8)}$$

中心距離関数を $F(x)$とすると，

$$F(x) = \frac{\int_0^x f(y)dy}{\int_0^1 f(y)dy} \qquad \text{(式 5.9)}$$

ここで，

$$\begin{aligned} \int_0^1 f(y)dy &= \int_0^1 \sum_{i=1}^{P} a_i \delta(y-d_i)dy \\ &= \sum_{i=1}^{P} a_i \\ &\equiv n \end{aligned} \qquad \text{(式 5.10)}$$

$$\begin{aligned} \int_0^x f(y)dy &= \sum_i \int_0^x a_i \delta(y-d_i)dy \\ &= \sum_{i \in \Omega_x} a_i \end{aligned} \qquad \text{(式 5.11)}$$

ただし，

$$\Omega_x = \{\, i \mid x \geqq d_i \} \qquad \text{(式 5.12)}$$

また，中心距離ベクトルを

$$\boldsymbol{x} = (x_1, x_2,\cdots, x_{\mathrm{M}}) \qquad \text{(式 5.13)}$$

とすると，

$$x_i = F\left(\frac{i}{M}\right), \quad i = 1, 2, \ldots, M \qquad \text{(式 5.14)}$$

であり，$F(x)$の性質から常に

$$x_M = F(1) = 1 \qquad \text{(式 5.15)}$$

となる。したがって $\boldsymbol{x}$ は本質的には$(M-1)$次元である。

5.3 中心距離ベクトルを用いた識別

中心距離ベクトルにより，ディジタル多値図形を特徴付けることができる。前節で述べた性質によると，定義 4 で示した中心距離関数が等しければ，元の多値図形は互いに相似である。また，中心距離関数の違いは，図形の違いを反映している。そこで，中心距離関数を用いて，二つの図形の違いを測ることにしよう。実際には中心距離ベクトルを用いる。

2 つのディジタル多値図形 X と，Y を考える。それぞれの M 次元中心距離ベクトル $\boldsymbol{x}, \boldsymbol{y}$ は，すなわち，

$$\boldsymbol{x} = (x_1, x_2,\cdots, x_M) \qquad \text{(式 5.16)}$$

$$\boldsymbol{y} = (y_1, y_2,\cdots, y_M) \qquad \text{(式 5.17)}$$

とすると，これら 2 つのベクトル間の距離 d を次式のように定義する。

$$d(X,Y) = \sum_{i=1}^{M} | x_i - y_i | \qquad \text{(式 5.18)}$$

この距離 d は，2 つの多値図形の間の違いを示す。M が適切に選ばれていて，X と Y が互いに相似ならば，$d(X,Y) \cong 0$ となるはずである。

逆に，X と Y が大きく異なっていれば，$d(X,Y)$は大きな値となる。したがって，中心距離ベクトルの距離 d を測ることにより，多値図形のクラス化に用いたり，識別に使用したりすることができる。

この方法では，ディジタル図形を点の座標の集合と考え，それらの点の間の順序やつながりに関する情報を一切捨象しているので，図形が単連結(SC)である場合も，そうでない場合(NSC)も，全く同様に扱うことができる。

一例として，図 5.4に示す 2 つのディジタル多値図形の中心距離関数を図 5.5に示す。二つの図形 A, B の形状の違いは，図 5.5の様な中心距離関数の違いとなって現れる。ちなみに，この場合の 2 つの図形の間の中心距離ベクトル間の距離 d は 5.4 である。

図 5.4 二つの多値図形

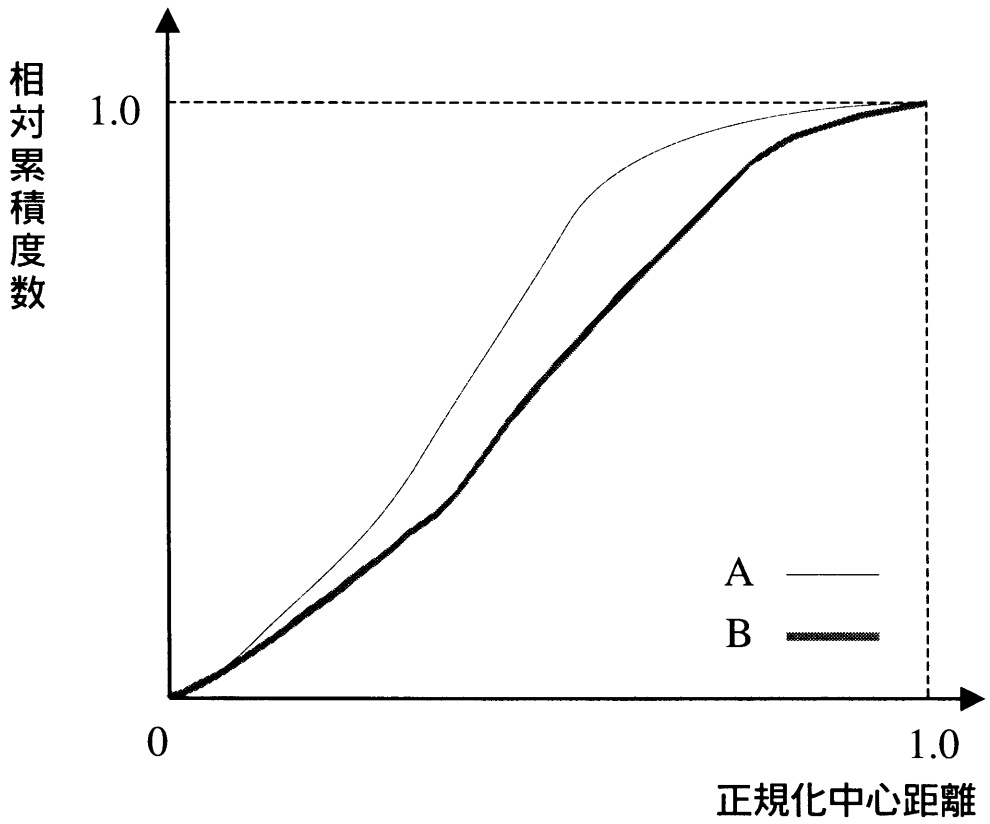

図 5.5 中心距離関数の違い

5.4　ディジタル多値図形認識の実験

本節では，前述の中心距離ベクトルを使用して，ディジタル多値図形の認識の実験を行った結果を述べる。本節の実験では，図 5.6に示すような(C)～(F)の 4 種類の幾何学多値図形を原図形（標準パターン）として用いた。

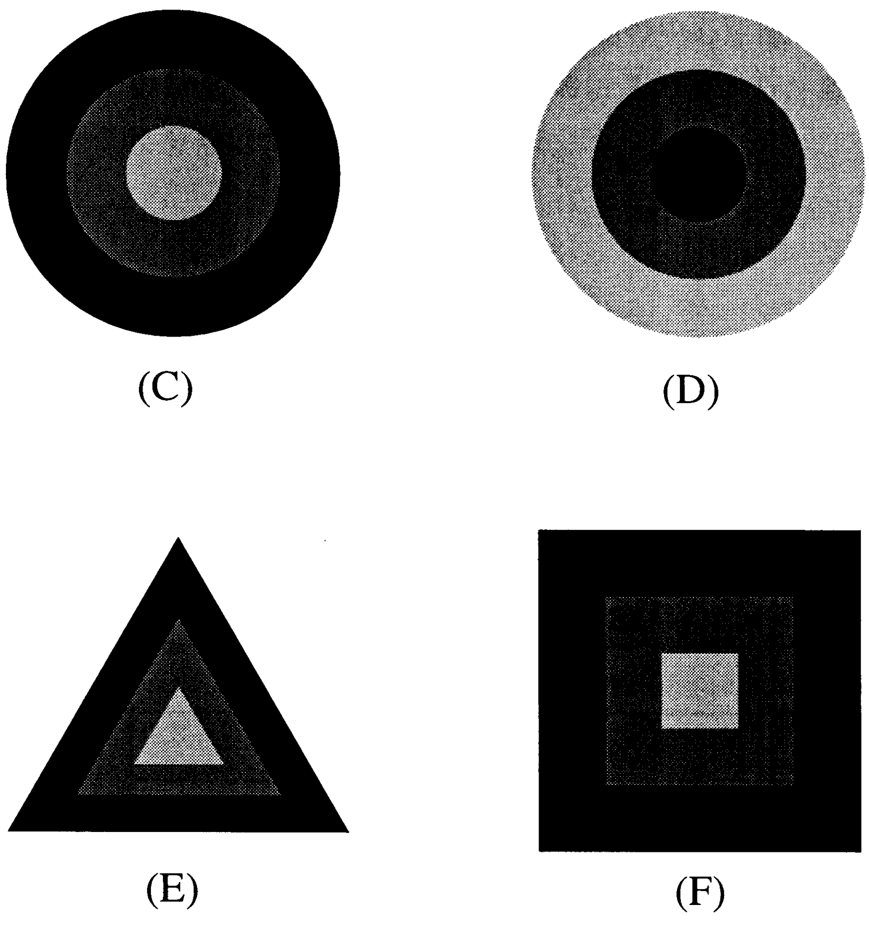

図 5.6 幾何学多値図形

この実験では，ある原図形を元に生成した相似図形を用いた。スケールは0.5～2.0倍までを0.1刻みで16種類，回転は0～$\pi/2$を$\pi/30$刻みで16種類，合計で256種類の相似図形を生成した。4種類の原図形に対してそれぞれ256種類の相似図形のクラスC_C～C_Fを生成し，これら1024種類の相似図形をランダムに入力した。そして，4種類の標準図形との距離を測り，たとえば，あるクラスC_Dに属する相似図形が原図形Dに最も近い場合，正しく識別できたものとした。その結果，1024種類の全ての相似図形から，原図形を正しく識別することができた。つまり認識率は100%であった。

また，これら(C)から(F)までの4種の原図形間の中心距離ベクトルの距離を測った。その結果は表5.1のとおりである。この表から，各クラスは他のクラスとよく分離していることがわかる。

表 5.1 幾何学図形間の距離

	C	D	E	F
C	0.0	5.8	19.5	11.8
D		0.0	19.6	12.3
E			0.0	7.7
F				0.0

5.5　まとめ

本章で提案したディジタル多値図形の認識法は，“中心距離関数”と呼ばれるものにもとづいており，自己距離関数同様に，図形を構成する点の順序に関する情報を一切必要としない点に著しい特徴がある。このため，ディジタル図形が単連結，非単連結にかかわらず適用することができる。さらに，中心距離関数が多値図形に適用できることを示した。本方法によれば，多値図形の相似変換に不変な認識をきわめてシンプルな方法で行うことができる。自己距離関数から中心距離関数への改良では，図形の重心を用いることにより，計算量を図形を構成する点の数 n の二乗のオーダーから一乗のオーダーへと減らすことができた。同様の改良が自己三角形関数にも適用できると考えられ，今後の課題として検討していきたい。