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
バージョン：
権利関係：

第3章

自己距離関数を用いた相似不変認識

本章では，ディジタル図形の新しい特徴抽出法を考案し，これ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“自己距離関数”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。この自己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）な図形認識を行うことができる。この自己距離関数を用いた認識実験結果も示す。

3.1　まえがき

2次元図形の輪郭線には，その認識のための重要な情報が含まれている。たとえば工業部品の画像などでは，単純な背景上での画像の輪郭線が認識のための手がかりを与える。また，魚種識別[18],[39]-[42]，個人識別[44]-[61]においても，魚の輪郭線，人の横顔の輪郭線は識別のための有効な手がかりとなる。このような輪郭線にとどまらず，工学技術において曲線（線図形）の処理・認識が必要な場合はきわめて多い。

本章では，有限個の点の集合をディジタル図形と呼ぶことにし，この図形のシフト・回転・スケール変換に不変な認識を行うための新しい方法

を提案する。上述の点の座標値は任意の実数値と考え，通常の画素で表示された線図形は，座標値が整数値であるという特別な場合と考えることにする。

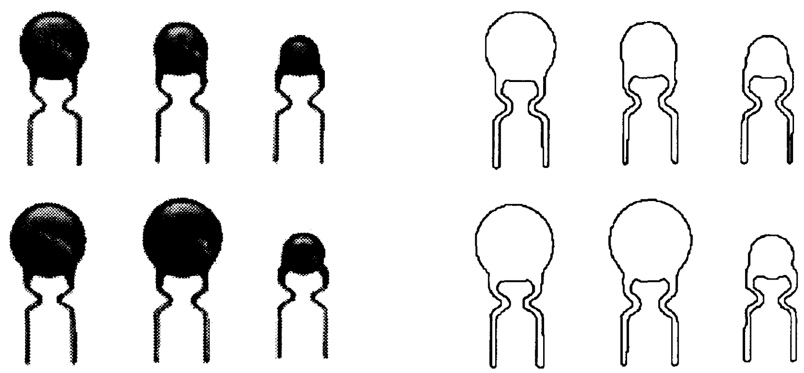


図 3.1 工業製品とその輪郭線

線図形は大きく分けて“単連結(simply connected; SC)”の場合と“非単連結(non-simply connected; NSC)”の場合とがある。これら線図形を計算機によって認識する立場からいえば，SC の認識よりも NSC の認識が極端に難しい。なぜなら、SC を構成する点の順序付けは簡単であるが、NSC の点の順序付けはいく通りも可能であり、任意に与えられた互いに相似な NSC に対して、点の順序付けを一意に決めることが難しいからである。

従来，ディジタル図形の認識を行う場合，フーリエ記述子のパワースペクトル[19]-[21]やモーメント不変量[25],[26]などを特徴量とする方法が用いられてきた。さらには，CSS(Curvature Scale Space)を用いる方法[27]，複素自己回帰モデルを用いる方法[28]，ディジタル曲線を多角形近似する方法[29],[30]がある。しかしながら，これらの方法では，互いに相似な図形を同一のものと認識するためには，曲線を構成する点の順序付けに関する先験的な情報が必要であり，この情報がない場合，非単連結な(NSC)図形への適用はほとんど不可能である。

本章では，NSC にも適用可能で相似不変な特徴抽出法，自己距離関数

方を提案する。本方法は文献[62],[63]における chord length 法の一つの変形であるとも考えられる。しかしながら[62],[63]においては，閉曲線の処理が前提とされており，しかも図形を構成する点の異なるもの同士比較の方法や情報の圧縮法については陽に言及していない。もちろん[62],[63]にも閉曲線を対象として，計算量の削減の方法も示されているが，本章で扱うような非閉曲線に対しては使うことはできない。

一方，本章で提案する方法によれば，曲線を構成する点の数が異なっていても，それらの形の違いを計算することができ，情報の圧縮法も明確である。また後述するように，曲線の近傍のノイズの吸収も考慮し得る。以下，本方法に基づいてディジタル図形の相似変換（シフト・回転・スケール変換）に不変な認識を行う方法及び実験例を示す。

なお，厳密にいえば，有限個の点集合として定義されたディジタル図形に対して，単連結，非単連結という概念は曖昧ではあるが，ここではディジタル曲線を“素直に”つないでいくと視覚的に単連結に見えるものを単連結ということにする。しかし，本手法では，SC も NSC も同様に取り扱うから，以下の議論ではこの区別をする必要は実はない。

3.2 ディジタル図形と自己距離関数

本論文では，ディジタル図形を次のように定義する。

［定義 1］ ディジタル図形とは，有限個の 2 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。

ディジタル図形を表す点の集合を D とすれば，

$$D=\{(x_i,y_i)\mid i=1,\ldots,n\},\quad x_i,y_i\in R \qquad (式\ 3.1)$$

である。R は実数の集合を表す。

通常の画素で表現されるディジタル図形は式 3.1 において，

$$x_i, y_i \in N \qquad \text{(式 3.2)}$$

であり，式 3.1 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，

$$D = \{(x_i, y_i) \mid i = 1, \ldots, n\}$$

$$\equiv \{(x_{p_i}, y_{p_i}) \mid i = 1, \ldots, n\} \qquad \text{(式 3.3)}$$

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

［定義 2］ ディジタル図形を構成するすべての点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自己距離密度と呼ぶ。

［定義 3］ 自己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 としたものを自己距離関数（累積分布）と呼ぶ。

ディジタル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図形（鏡像も含む）の自己距離関数は等しい。

実際には，測定誤差の吸収と情報圧縮のために，定義 2 における区間 [0, 1] を M 等分して，各部分区間に落ちる自己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，これを自己距離ベクトルと呼ぶ。この M の値が大きくなるほど元の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。

以上のように本方法では，点の間の距離の最大値を 1 に正規化し，累

積全度数も 1 とし，さらにこの自己距離関数（累積分布）を M 等分して表現するので，点の数の異なる図形の比較が容易になる。また，図形の近傍に現れるノイズの影響も，M 個の標本化によって吸収される。

一例として，図 3.2のディジタル曲線の M =100 とした場合の自己距離関数を図 3.3に示す。

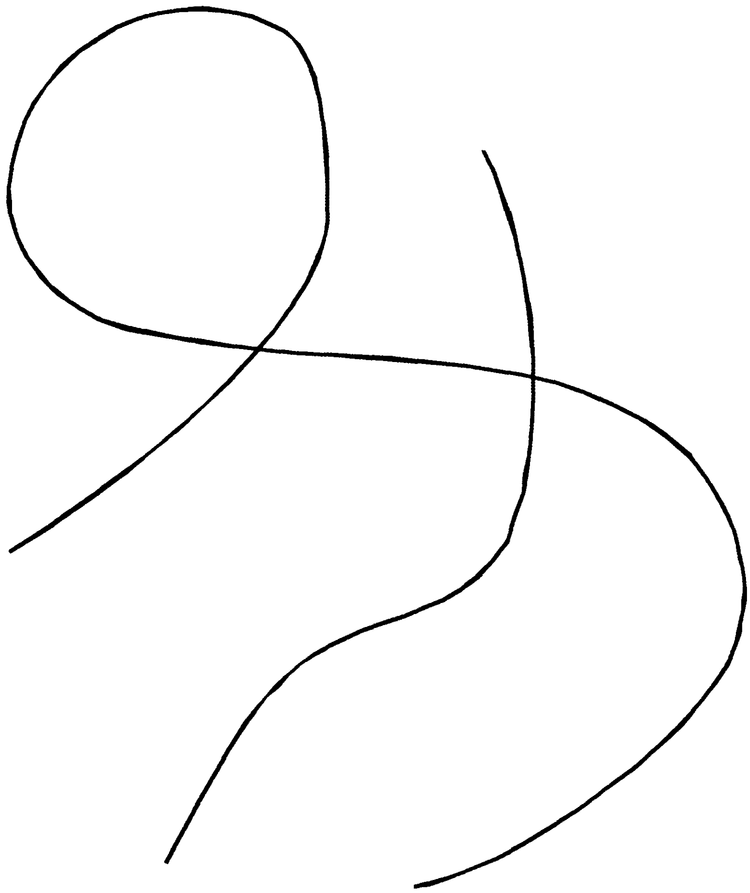


図 3.2 ディジタル図形

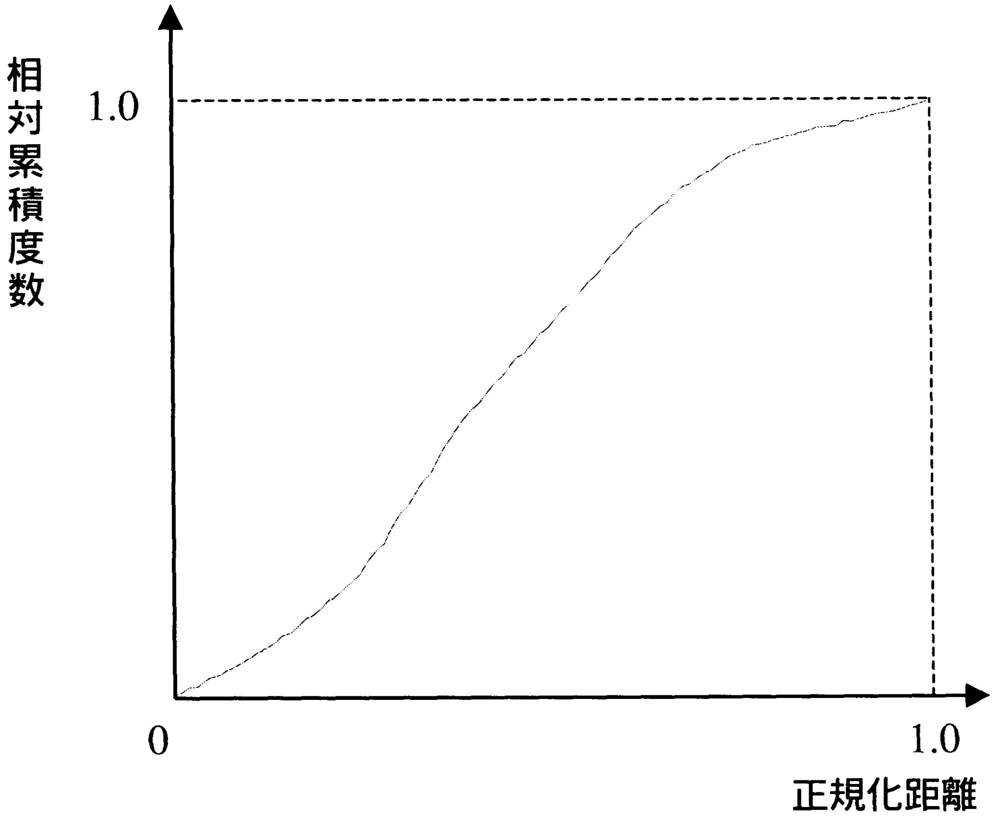


図 3.3 自己距離関数

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。自己距離密度を関数と考えて$f(x)$とすると，

$$f(x)=\sum_{i=1}^{P}a_i\delta(x-d_i),\quad 0\leqq x,d_i\leqq 1 \qquad \text{(式 3.4)}$$

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，

$$\sum_{i=1}^{P} a_i = \frac{1}{2} n(n-1) \qquad \text{(式 3.5)}$$

という関係にある。式3.4中の$\delta(x-d_i)$ はデルタ関数であり，次の性質を持つ。

$$\delta(x-d_i) = 0, \quad x \neq d_i \qquad \text{(式 3.6)}$$

$$\int_0^x \delta(y-d_i)dy = \begin{cases} 1, & x \geqq d_i \\ 0, & \text{その他} \end{cases} \qquad \text{(式 3.7)}$$

自己距離関数を $F(x)$とすると，

$$F(x) = \frac{\int_0^x f(y)dy}{\int_0^1 f(y)dy} \qquad \text{(式 3.8)}$$

ここで，

$$\begin{aligned} \int_0^1 f(y)dy &= \int_0^1 \sum_{i=1}^{P} a_i \delta(y-d_i)dy \\ &= \sum_{i=1}^{P} a_i \\ &\equiv \frac{1}{2} n(n-1) \end{aligned} \qquad \text{(式 3.9)}$$

$$\begin{aligned} \int_0^x f(y)dy &= \sum_i \int_0^x a_i \delta(y-d_i)dy \\ &= \sum_{i \in \Omega_x} a_i \end{aligned} \qquad \text{(式 3.10)}$$

ただし，

$$\Omega_x = \{\, i \mid x \geqq d_i \} \qquad \text{(式 3.11)}$$

また，自己距離ベクトルを

$$\boldsymbol{x} = (x_1, x_2,\cdots, x_{\mathrm{M}}) \qquad \text{(式 3.12)}$$

とすると，

$$x_i = F\left(\frac{i}{M}\right),\quad i=1,2,\ldots,M \qquad \text{(式 3.13)}$$

であり，$F(x)$の性質から常に

$$x_M = F(1) = 1 \qquad \text{(式 3.14)}$$

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

ここで，簡単な図形を例にとって，自己距離ベクトルを求める手順を説明しよう。図 3.4に 6 つの点からなる図形を示す。この図形から得られる距離の総数は式 3.5 により，

$$\frac{1}{2}6(6-1)=15$$

となる。隣り合う各点間の距離を 1 とし，これらの距離の度数分布を求めると，表 3.1の様になる。自己距離密度は，距離の最大値を 1 に正規化したものであるから，図 3.5のようになる。

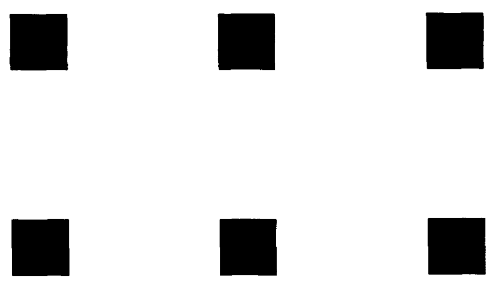
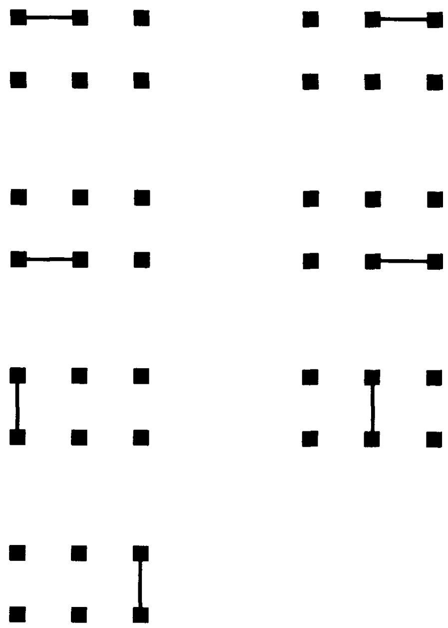
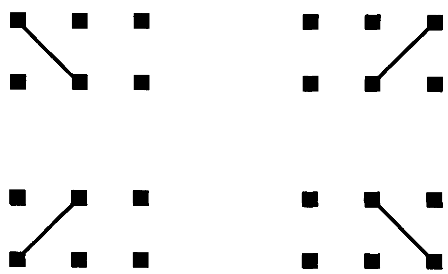


図 3.4　6 つの点からなる図形

表 3.1　6 点から構成される距離と度数

距離値	2 点の組み合わせ	度数
1		7
$\sqrt{2}$		4
2		2
$\sqrt{5}$		2

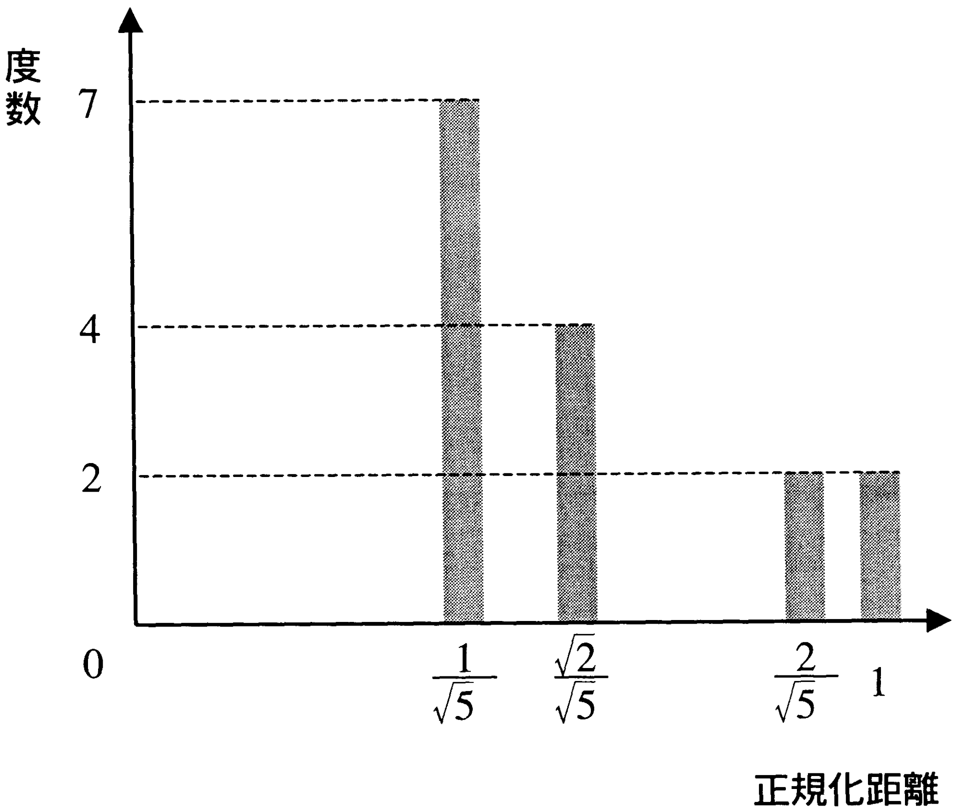


図 3.5　6 点からなる図形の自己距離密度

図 3.4の 6 点からなる図形の自己距離関数を図 3.6に示す。自己距離密度（図 3.5）を累積表現し，度数を 1 に正規化している。また，このとき，ベクトルの次元 $M = 3$ としたときの自己距離ベクトルは図 3.7の様になる。3 つの部分区間におちる自己距離密度の累積が，ベクトルの各次元の値となる。また第 3 番目のベクトルの成分が 1 になるように正規化されたものとなっている。

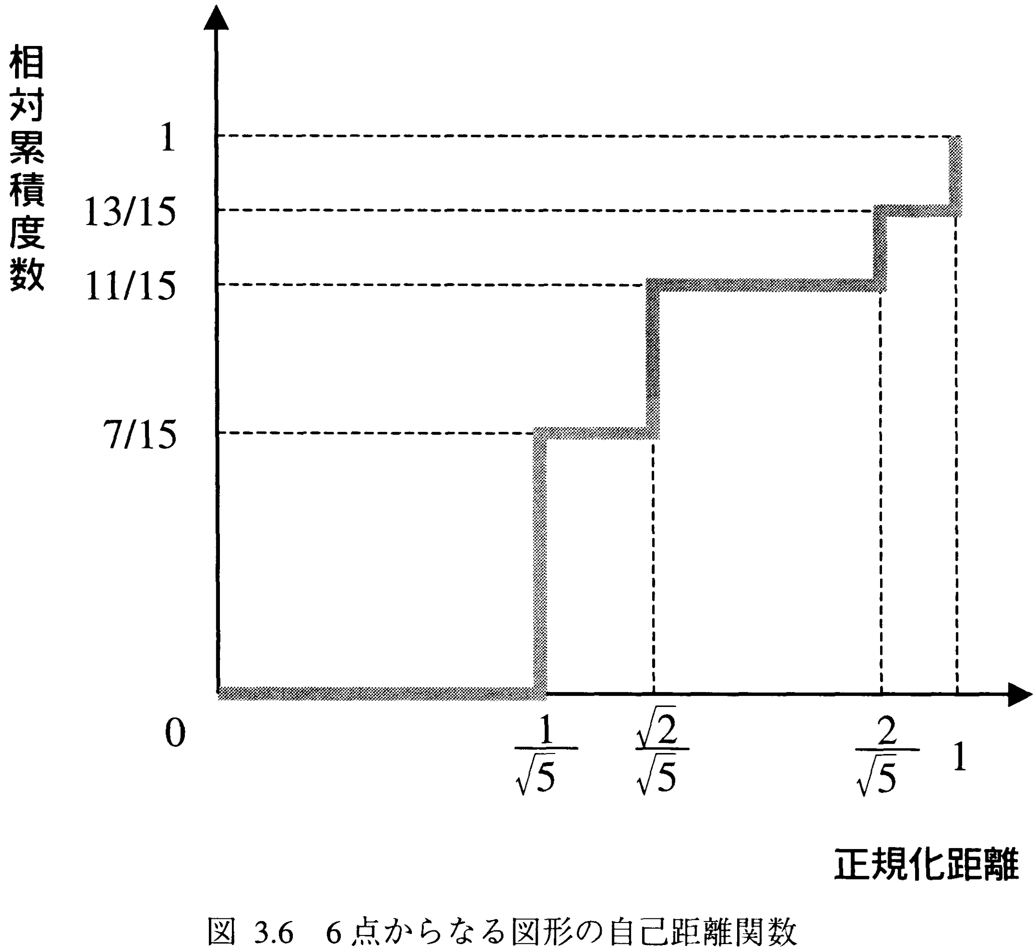


図 3.6　6 点からなる図形の自己距離関数

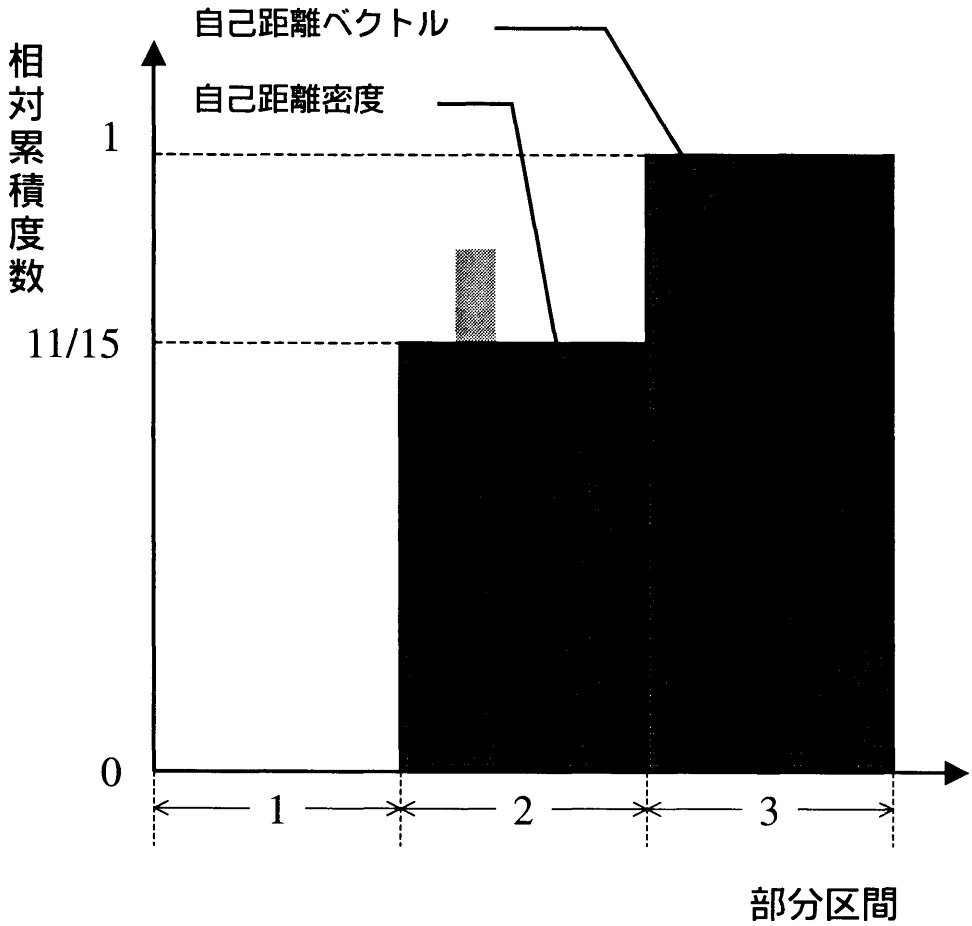


図 3.7　6 点からなる図形の自己距離ベクトル

3.3 ディジタル図形間の距離

前節で定義した自己距離関数は，その図形の形状を表したものであり，互いに相似な図形は同一の自己距離関数を持つ。したがって，この自己距離関数を用いれば，ディジタル図形のシフト・回転・スケール変換に不変な認識が行える。

前節で述べた性質 1 の逆は成り立たない。つまり，定義 2 で示した自己距離関数が等しいということは，もとの図形が互いに相似であることの必要条件にすぎない。しかしながら，実用上は，自己距離関数が等しければ元の図形が互いに相似であり，自己距離関数の違いは図形の違いを反映していることを実験的に確認できる。情報圧縮の観点から，自己距離関数を用いて，二つの図形の違いを測ることにしよう。具体的には自己距離ベクトルを用いる。

2 つの図形 X と，Y を考える。それぞれの M 次元自己距離ベクトル $\boldsymbol{x}, \boldsymbol{y}$ は，すなわち，

$$\boldsymbol{x} = (x_1, x_2, \cdots, x_M) \qquad \text{(式 3.15)}$$

$$\boldsymbol{y} = (y_1, y_2, \cdots, y_M) \qquad \text{(式 3.16)}$$

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。

$$\begin{aligned} d(X,Y) &= \| \boldsymbol{x} - \boldsymbol{y} \| \\ &= \left(\sum_{i=1}^{M} (x_i - y_i)^2 \right)^{\frac{1}{2}} \end{aligned} \qquad \text{(式 3.17)}$$

すなわち，2 つのベクトル間のユークリッド距離である。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。式 3.17 で示したベクトル間の距離 d は，

$$d'(X,Y) = \sum_{i=1}^{M} | x_i - y_i | \qquad (式\ 3.18)$$

としてもよい。

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 3.8に示す 2 つのディジタル図形の自己距離関数を図 3.9に示す。これは実際には M=100 とした自己距離ベクトルであるが，M=100 としたこの程度の大きさの図では連続曲線のように見える。図 3.8の二つの図形 A, B の形状の違いは，図 3.9の様な自己距離関数の違いとなって現れる。ちなみに，この場合の 2 つの図形の間の距離(式 3.18)は 3.5 である。

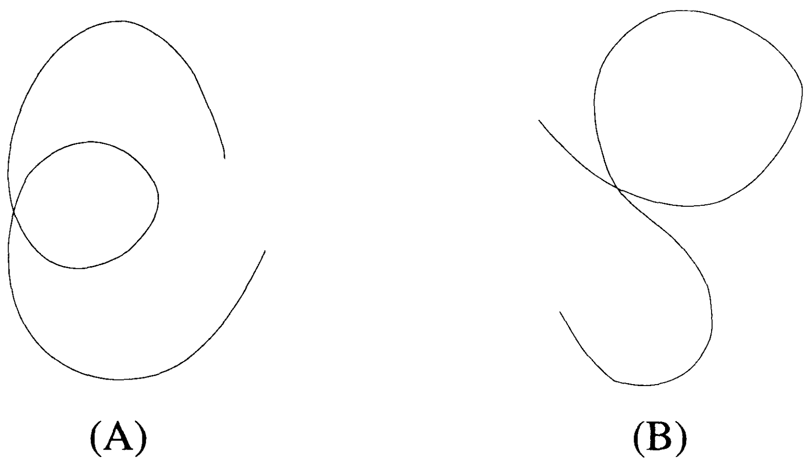


図 3.8 二つのディジタル図形

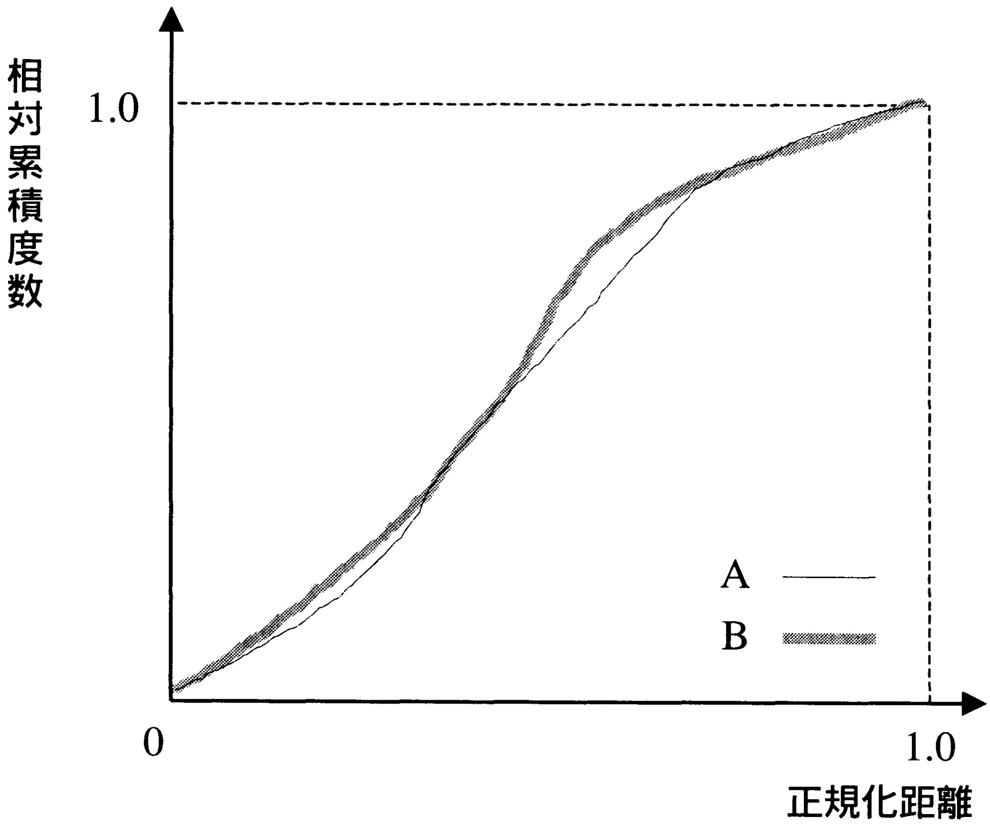


図 3.9 自己距離関数の違い

3.4 自己距離関数を用いたディジタル図形認識の実験

本節では，前述の自己距離ベクトルを使用して，ディジタル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，自己距離ベクトルの次元数（部分区間数）M=100 とした時の距離(式 3.18)を用いた。

3.4.1 原図形とその相似図形との距離

本実験では，図 3.2を原図形として，その相似図形との距離を測る。スケールは，0.5～2.0 倍を 0.1 刻みで 16 種類，回転は $0 \sim \pi/2$ を $\pi/30$ 刻みで 16 種類，合計 256 個の相似図形を作り，原図形との距離を測った。その結果を表 3.2に示す。この表から，原図形とその相似図形との間の距離は 1.3 未満であることがわかる。

表 3.2 標準図形との間の距離

スケール	回転角度(×π/30) 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0.5	0.667	0.511	0.360	0.452	0.616	0.802	0.516	0.978	0.691	0.627	0.433	0.335	0.609	0.420	0.337	0.724
0.6	0.296	0.178	0.179	1.275	0.381	0.530	0.635	0.749	0.613	0.306	0.236	0.221	0.453	0.461	0.379	0.258
0.7	0.474	0.302	0.283	0.466	0.734	0.556	0.726	0.607	0.515	0.574	0.262	0.357	0.226	0.271	0.490	0.442
0.8	0.198	0.171	0.283	0.464	0.557	0.477	0.939	0.859	0.692	0.592	0.232	0.186	0.238	0.263	0.267	0.282
0.9	0.260	0.224	0.246	0.763	0.884	0.726	0.658	0.747	0.549	0.396	0.264	0.219	0.384	0.301	0.304	0.348
1.0	0.000	0.230	0.265	0.408	0.678	0.663	0.678	0.683	0.586	0.420	0.221	0.085	0.090	0.167	0.387	0.254
1.1	0.259	0.197	0.236	0.368	0.465	0.711	0.919	0.839	0.704	0.523	0.301	0.199	0.300	0.294	0.323	0.360
1.2	0.215	0.183	0.164	0.310	0.391	0.684	1.042	0.759	0.609	0.381	0.216	0.369	0.205	0.358	0.315	0.239
1.3	0.321	0.185	0.250	0.391	0.422	0.629	0.944	0.691	0.802	0.486	0.375	0.208	0.219	0.407	0.235	0.339
1.4	0.148	0.153	0.193	0.494	0.553	0.722	0.900	0.825	0.666	0.453	0.261	0.264	0.222	0.292	0.193	0.256
1.5	0.249	0.282	0.186	0.521	0.406	0.568	0.724	0.529	0.372	0.326	0.177	0.240	0.178	0.346	0.250	0.193
1.6	0.088	0.083	0.250	0.270	0.451	0.427	0.676	0.594	0.318	0.125	0.417	0.099	0.140	0.238	0.198	0.216
1.7	0.224	0.174	0.204	0.212	0.211	0.316	0.374	0.342	0.202	0.204	0.285	0.252	0.205	0.243	0.218	0.209
1.8	0.179	0.151	0.157	0.500	0.202	0.757	0.395	0.748	0.337	0.302	0.168	0.198	0.866	0.242	0.291	0.203
1.9	0.172	0.170	0.152	0.202	0.210	0.213	0.435	0.333	0.827	0.319	0.258	0.765	0.415	0.207	0.286	0.167
2.0	0.164	0.220	0.174	0.193	0.148	0.249	0.152	0.466	0.592	0.464	0.250	0.139	0.641	0.429	0.186	0.164

3.4.2 幾何学図形の認識

ここでは，単純な幾何学図形の識別に関する実験を行った。図 3.10に示す様な 4 種の幾何学図形，真円，正三角形，正方形，十字形を用いた。それぞれの図形の自己距離関数を図 3.11に示す。図形の形状の違いが自己距離関数の違いとなって現れることがわかる。これら 4 種の図形の間の距離は表 3.3のとおりである。

さらに，図 3.10の 4 種の図形を原図形として，それぞれの図形に対して3.4.1の実験と同様の方法によって 256 種類の相似図形を作り，各相似図形を識別する実験を行った。相似図形と 4 種の原図形との距離を式 3.18を用いて比較し，その最小値を与える原図形と相似であると決定した。その結果，すべての相似図形を正しく識別することができた。

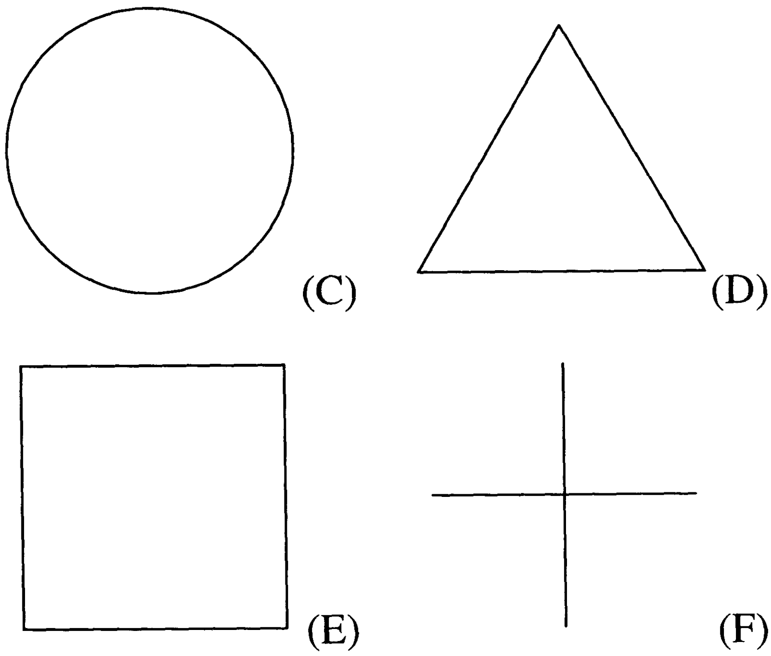


図 3.10 幾何学図形

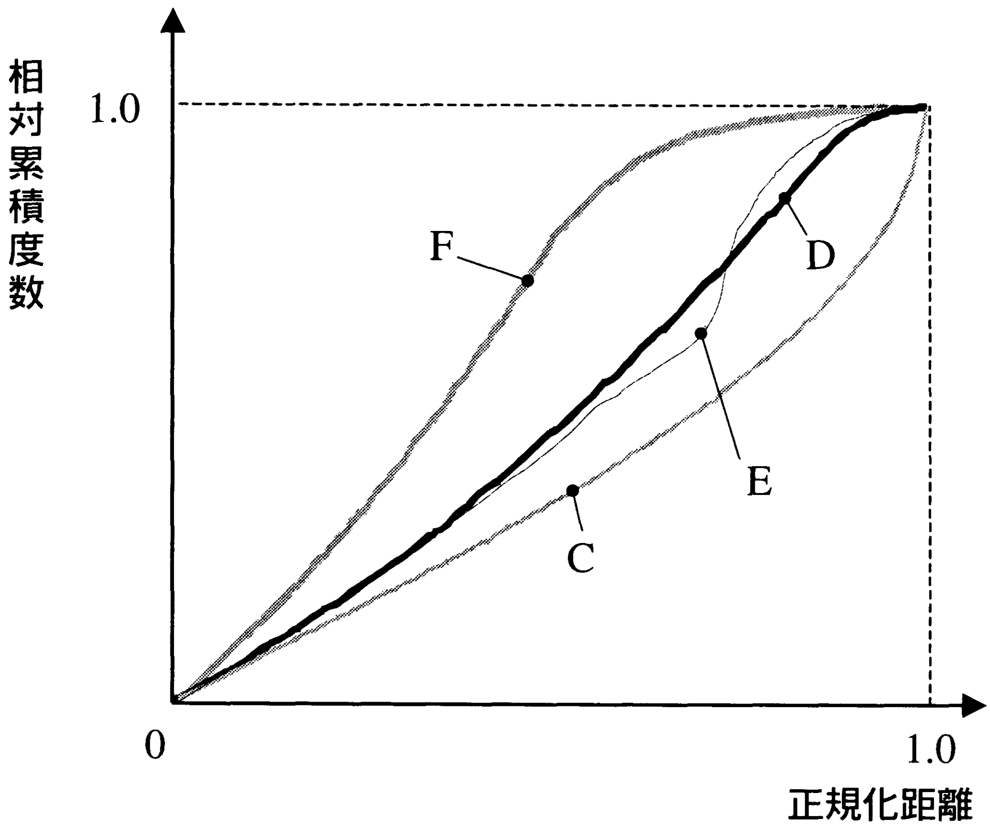


図 3.11 自己距離関数の違い

表 3.3 幾何学図形間の距離

	C	D	E	F
C	0.0	11.2	10.8	26.9
D		0.0	2.2	15.6
E			0.0	16.1
F				0.0

3.4.3　図形の局所的な形状の違いの識別

ここでは，図形の局所的な形状の違いの識別に関する実験を行った。図 3.12に示す 3 種の図形(a),(b),(c)は，矢印付近の形状が異なる図形である。それぞれの自己距離関数を図 3.13に示す。原図形の局所的な形状の違いは自己距離関数を表す曲線の全域的な形の違いとなって現れることがわかる。

さらに3.4.2の実験と同様に，これら 3 種の図形を原図形として実験を行った。その結果，すべての相似図形を正しく識別することができた。各図形間の距離を表 3.4に示す。

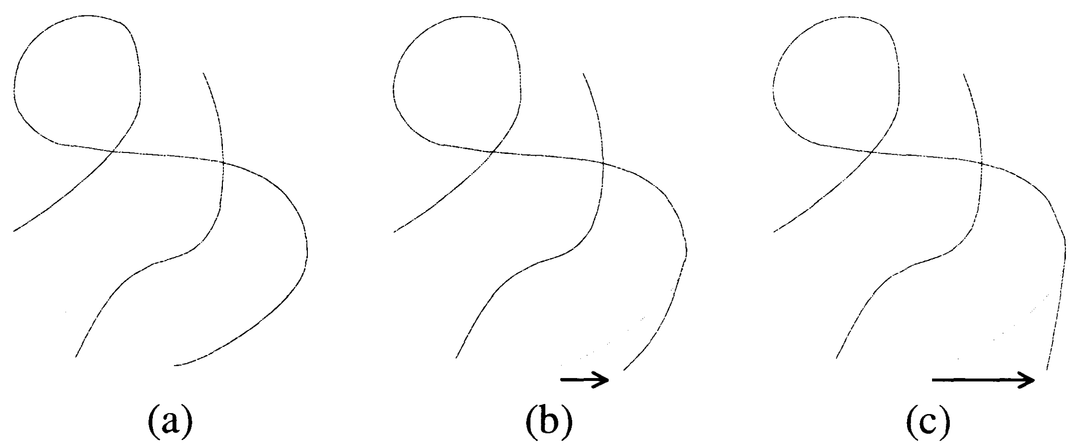


図 3.12 局所的に異なる形状をもつ図形

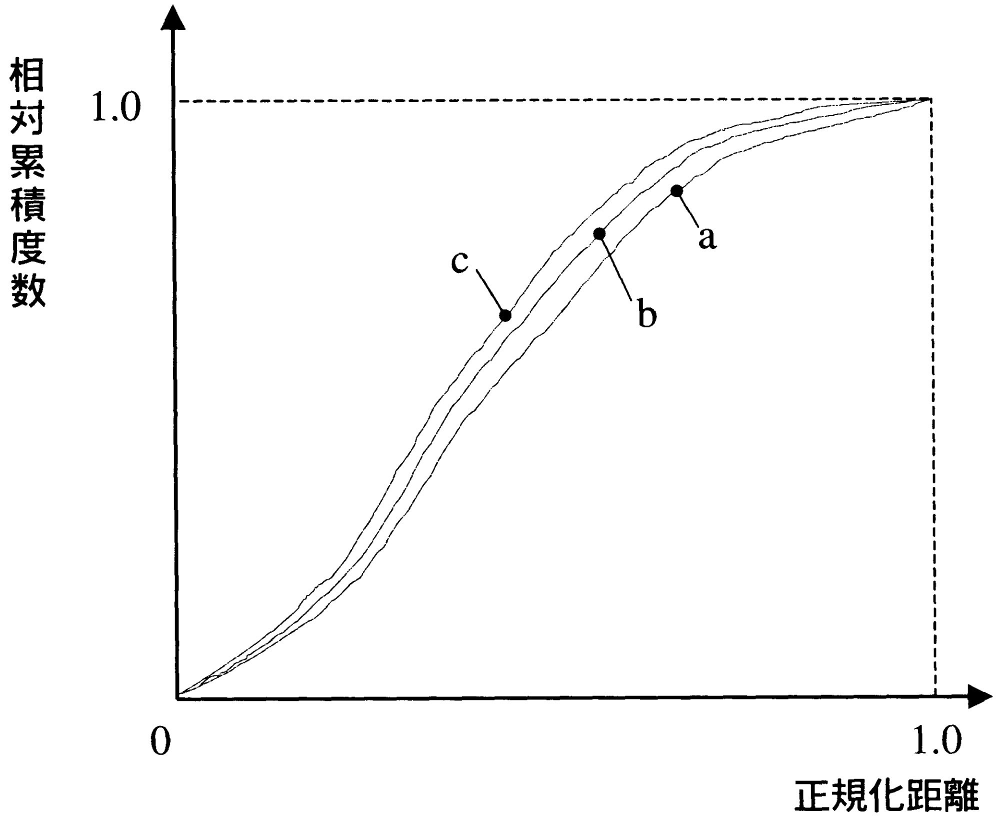


図 3.13 局所的に異なる図形の自己距離関数

表 3.4 局所的に異なる図形間の距離

	a	b	c
a	0.0	3.1	5.4
b		0.0	2.3
c			0.0

3.4.4　複雑な曲線からなる図形の認識

ここでは，いくつかの原図形が与えられた場合に，それらに相似な図形が原図形と同じであると認識できるか否かを調べる実験を行った。

図 3.14にGからOの9種類の原図形を示す。各々の原図形に対して，3.4.1の実験と同様の方法によって256種類の相似図形を作った。したがって，全部で2304個の図形があり，各図形がGからOのどの原図形と相似であるかを認識することが，与えられた問題である。これら9種の原図形との距離は，表 3.5のとおりである。

相似図形と9種の原図形との距離を式3.18を用いて比較し，その最小値を与える標準図形と相似であると決定した。本実験における相似図形の識別率は100%であった。

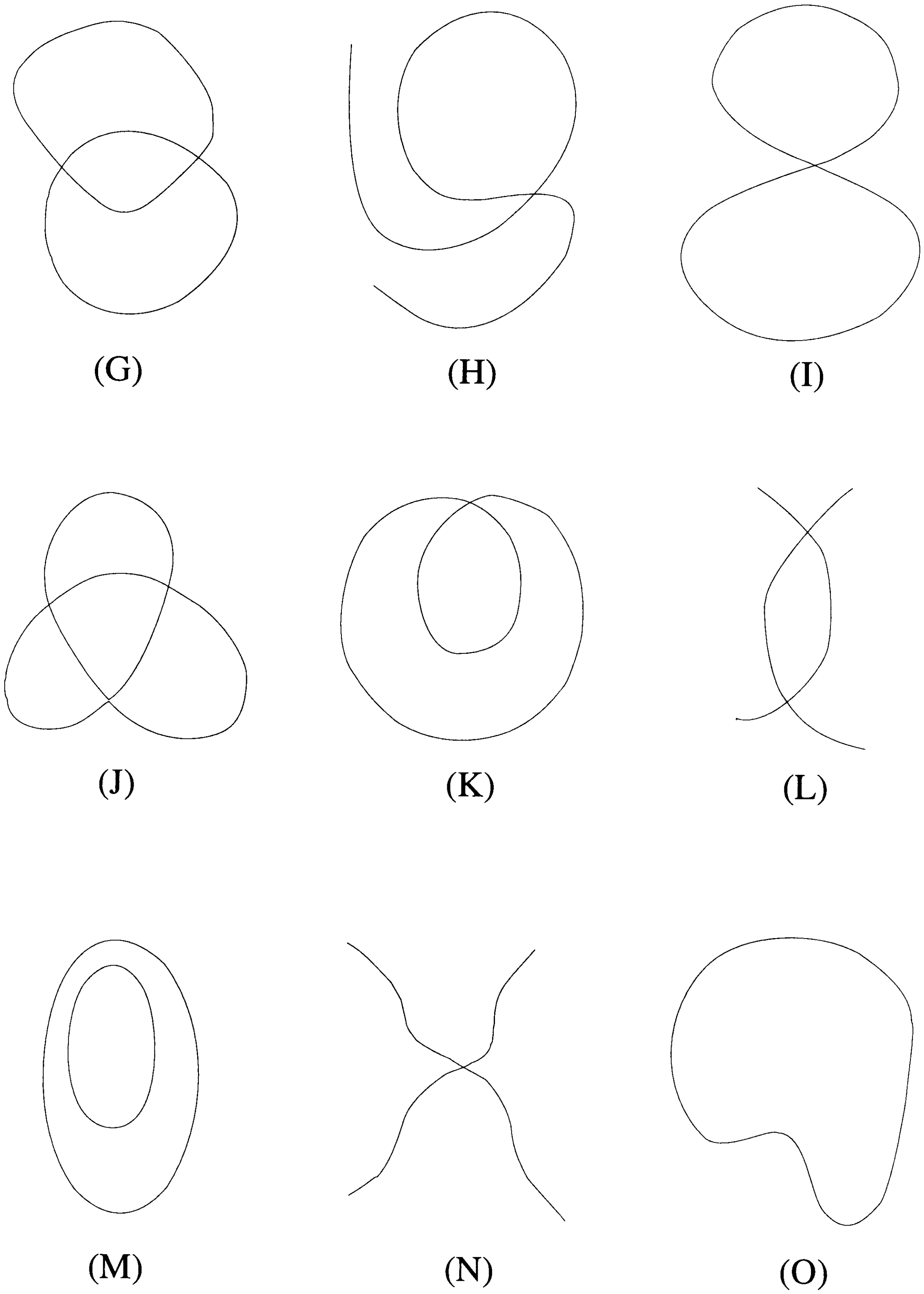


図 3.14　9 種類の図形(G)～(O)

表 3.5 図形(G)～(O)の間の距離

	G	H	I	J	K	L	M	N	O
G	0.0	1.2	1.9	1.4	6.3	10.8	3.5	13.2	6.6
H		0.0	2.2	1.3	6.2	10.9	3.6	13.3	6.4
I			0.0	2.0	5.8	11.4	4.2	13.8	5.9
J				0.0	7.0	10.1	2.9	12.5	7.2
K					0.0	17.1	9.8	19.5	3.1
L						0.0	7.3	3.0	16.9
M							0.0	9.7	9.8
N								0.0	19.3
O									0.0

3.5 まとめ

ディジタル図形を記述する新しい特徴と，その特徴を用いた図形の相似不変認識について述べた。本章で提案したディジタル図形の特徴は，“自己距離関数”と呼ばれるものに基づいており，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。