

デ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2章

不変認識とデジタル図形の

輪郭線記述

2.1 まえがき

輪郭線はテクスチャに比べデータ量が少ないにもかかわらず、図形認識のための重要な情報を含んでいる。特に工業部品などの硬質の物体は、その輪郭線により認識できることが多い。図形を輪郭線と見立て、その輪郭線を種々の曲線記述法によってパラメータ記述したものが特徴としてよく用いられる。

本章では、まず基礎理論として、画像認識における不変認識とその特徴について述べる。次いで、一般的によく用いられている輪郭線の記述法および、距離の定義等のデジタル図形を取り扱う場合の諸問題について述べる。

2.2 画像認識と変換

画像認識を行うには、画像から特徴を抽出する必要がある。このとき、画像は画像空間から特徴空間へ写像され、ここで抽出された特徴の比較によって認識が行われる。認識結果は、抽出された特徴に大きく左右される

ので、特徴は適切に抽出される必要がある。しかしながら、我々人間が同じものとして認識する画像も、計算機に入力する画像のデータとして考えると全く別なものとなることが多い。たとえば、2次元平面上の星型を異なった2つの位置から撮影したとき（図 2.1）、2つの画像は、人間が見た場合どちらも星型が写っている画像（図 2.2）として認識できる。しかし計算機に入力する画像データとして考えるとこれらは全く異なった数値の集合体（図 2.3）である。計算機は基本的に数値の演算・比較程度の処理しか行うことができないため、これらの数値の集合体を単純に演算・比較しても、同一なものとして認識するのは困難である。このため、2つの画像から、撮影位置に依存しない特徴をうまく抽出した上で特徴空間上での比較を行う必要がある。

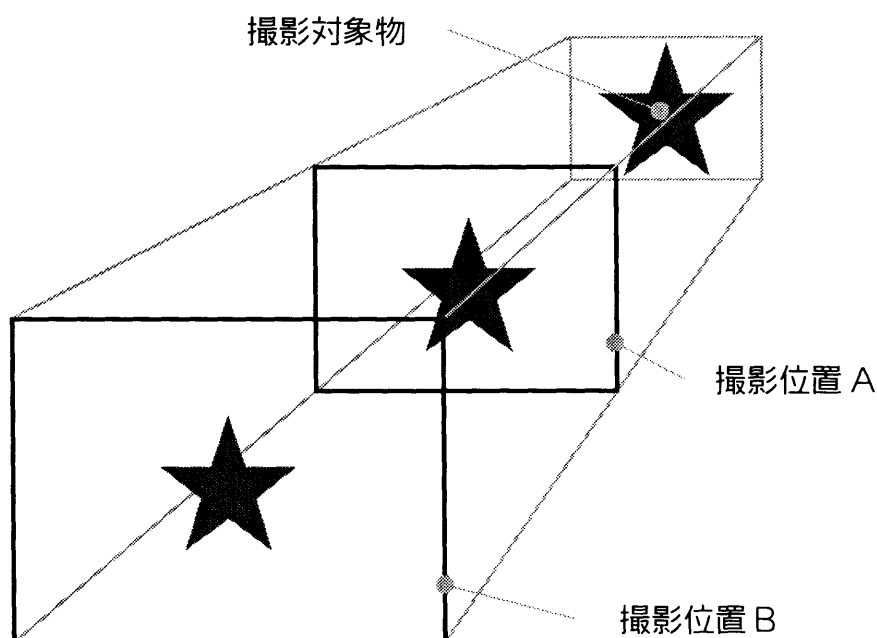
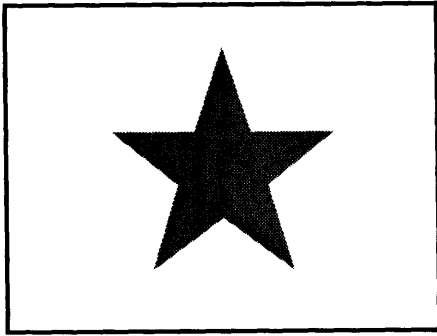
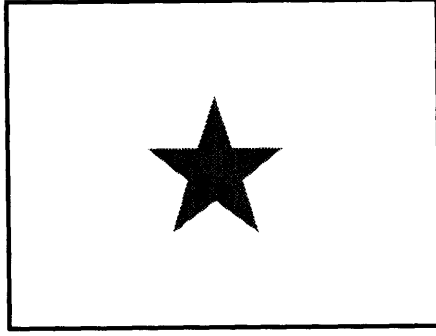


図 2.1 異なった位置での撮影の例

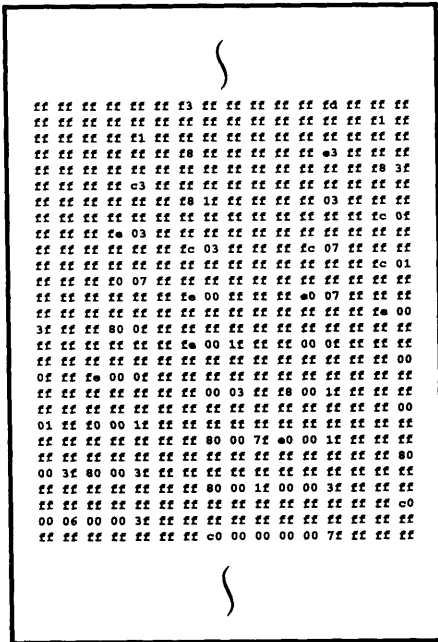


(a) 位置 A で撮影された画像

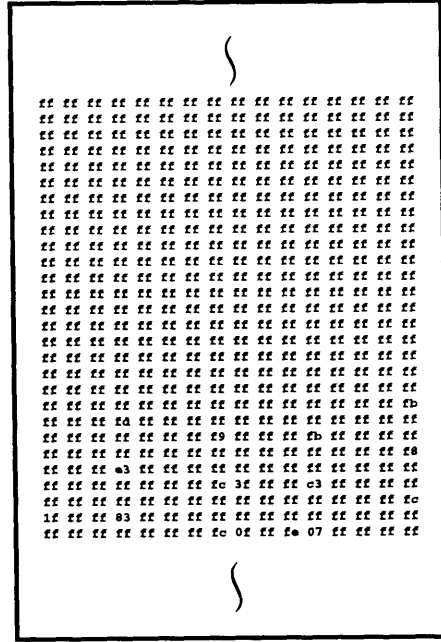


(b) 位置 B で撮影された画像

図 2.2 それぞれの位置で撮影された画像(人間が見た場合)



(a) 位置 A で撮影された画像



(b) 位置 B で撮影された画像

※ モノクロビットマップ形式 (黒画素=0, 白画素=1
として 16 進数表記), データの一部のみ示す。

図 2.3 それぞれの位置で撮影された画像(計算機に入力されるデータ)

この例の場合，二つの画像は，スケール変換の関係にあるので，スケール変換に対して不変な特徴量を用いた認識（スケール不変認識）を行わなければならない。このように，ある変換の関係にある2つの画像を同一のものとして認識するためには，その変換に対して不変な特徴を用いた認識を行う必要がある。このような認識のことをその変換に対する不変認識と呼び，変換の種類については主に次のようなものがある。

1. シフト変換
2. 回轉變換
3. スケール変換
4. アフィン変換

これらのうち，シフト変換・回轉變換についてはまとめてユークリッド変換と呼ぶ。これにスケール変換を加えたものを，相似変換と呼ぶ。アフィン変換は，相似変換にせん断変形(deformation, shear)やある一方向への拡大（伸縮）を加えたものであり，図 2.4のような関係にある。

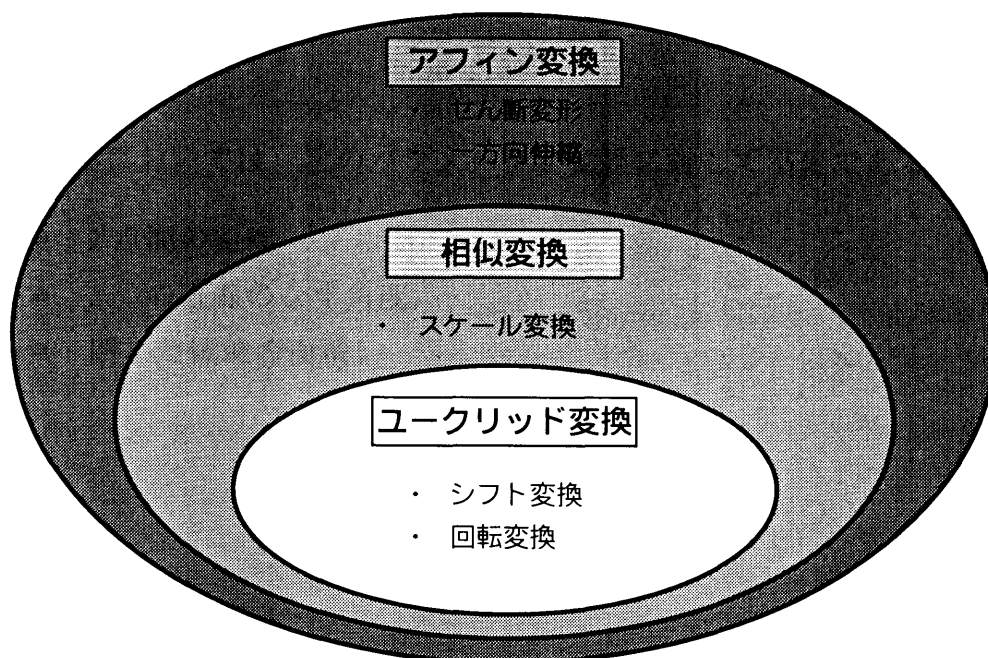


図 2.4 各種の変換の関係

2.2.1 シフト（平行移動）変換

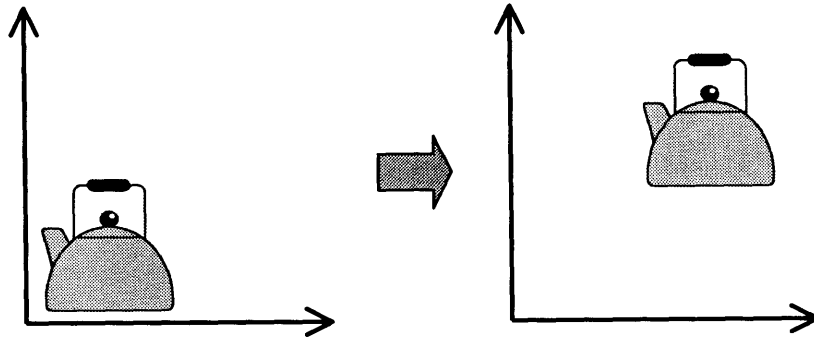


図 2.5 シフト変換

2次元平面上の点 $\mathbf{x} = (x, y)$ から、2次元平面上の点 $\mathbf{x}' = (x', y')$ へのシフト（平行移動）変換は、シフト成分 $\mathbf{s} = (s_x, s_y)$ を用いて次のように表すことができる。

$$\mathbf{x}' = \mathbf{x} + \mathbf{s} \quad (\text{式 2.1})$$

シフト変換においては、次のパラメータが変換に対して不変である。

- 2点間の距離
- 2つの直線のなす角度
- 閉じた領域の面積

2.2.2 回転変換

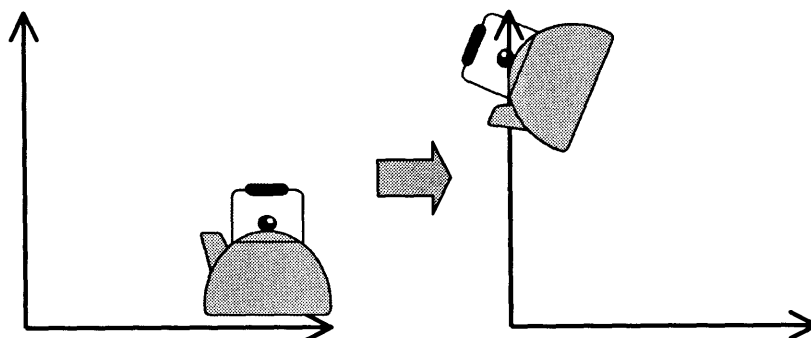


図 2.6 回転変換

2次元平面上の点 $\mathbf{x}=(x, y)$ から、2次元平面上の点 $\mathbf{x}'=(x', y')$ への回転変換は次のように表すことができる。

$$\mathbf{x}' = \mathbf{R}\mathbf{x} \quad (\text{式 2.2})$$

ここで、 \mathbf{R} は回転を表す 2×2 行列であり、回転角 θ を用いて次のように表す。

$$\mathbf{R} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \quad (\text{式 2.3})$$

回転変換においてもシフト変換同様に、次のパラメータが変換に対して不変である。

- 2点間の距離
- 2つの直線のなす角度
- 閉じた領域の面積

また、シフト変換と回転変換をあわせたものであるユークリッド変換は式 2.1 および式 2.2 より次のように表す。

$$\mathbf{x}' = \mathbf{R}\mathbf{x} + \mathbf{s} \quad (\text{式 2.4})$$

2.2.3 スケール変換

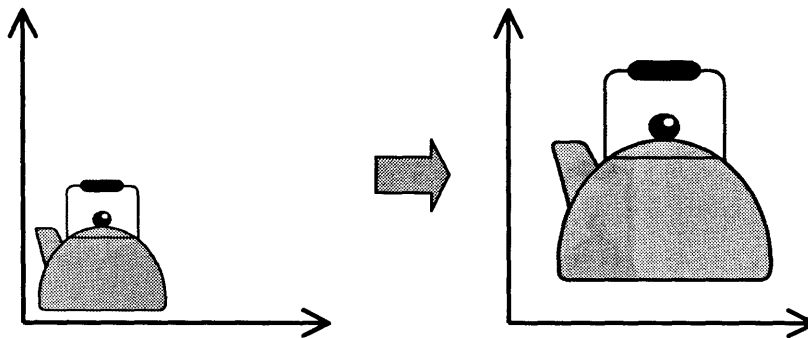


図 2.7 スケール変換

2次元平面上の点 $\mathbf{x} = (x, y)$ から、2次元平面上の点 $\mathbf{x}' = (x', y')$ へのスケール変換は、スケールファクター k を用いて次のように表すことができる。

$$\mathbf{x}' = k\mathbf{x} \quad (\text{式 2.5})$$

以下にスケール変換に対して不変となるものを示す。

- 2点間の距離の比
- 2つの直線のなす角度
- 閉じた領域の面積の比

スケール変換では、シフト変換・回転変換同様に、2つの直線のなす角度は不変であるが、2点間の距離や閉じた領域の面積は不変とはならない。ただし、2点間の距離の比や閉じた領域の面積の比は不変である。

また、ユークリッド変換にスケール変換を加えた相似変換は、式 2.3 お

よび式 2.4 により次のように表す。

$$\mathbf{x}' = \mathbf{kRx} + \mathbf{s} \quad (\text{式 2.6})$$

2.2.4 アフィン変換

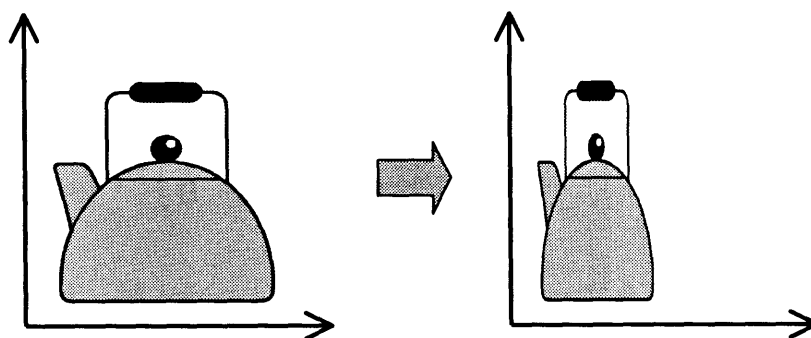


図 2.8 アフィン変換(伸縮)

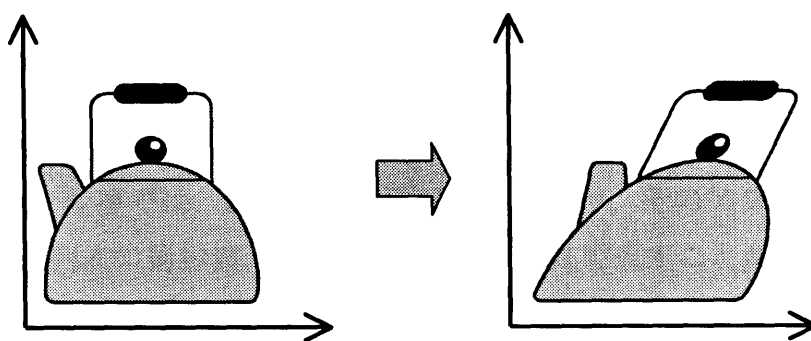


図 2.9 アフィン変換(せん断)

2次元平面上の点 $\mathbf{x} = (x, y)$ から、2次元平面上の点 $\mathbf{x}' = (x', y')$ へのアフィン変換は、 2×2 行列 \mathbf{A} を用いて次のように表すことができる。

$$\mathbf{x}' = \mathbf{Ax} + \mathbf{s} \quad (\text{式 2.7})$$

ここで、 s はシフト成分である。アフィン変換は、相似変換に伸縮やせんだんを加えたものであり、行列 A は、相似変換における回転行列 R をスケールファクター k 倍したのも含まれ、より一般的には次式のように表される。

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad (\text{式 2.8})$$

たとえば、 x 軸方向への伸縮(図 2.8)の場合、行列 A は、次のような形式となる。

$$A = \begin{pmatrix} a & 0 \\ 0 & 1 \end{pmatrix} \quad (\text{式 2.9})$$

ここで、 a は x 軸方向へのスケールファクターである。また、図 2.9 のように x 軸方向に関して傾斜させたせん断変換の場合、行列 A は、次のような形式となる。

$$A = \begin{pmatrix} 1 & b \\ 0 & 1 \end{pmatrix} \quad (\text{式 2.10})$$

ここで、 b は x 軸方向への傾斜の度合いを示す。

アフィン変換では、ある閉じた領域の面積 S と、そのアフィン変換後の面積 S' は行列式の絶対値 $|\det A|$ を用いて次式で表すことができる。

$$S' = |\det A| S \quad (\text{式 2.11})$$

つまり、閉じた領域の面積の比はアフィン変換の下で不変である。しかしながら、2つの直線のなす角度や2点間の距離の比は、もはや不変とはならない。

アフィン変換は、対象となる画像とカメラ視点との間に十分な距離がある場合、アフィン変換を透視変換の近似として使用できることが知られ

ている[68][69]。また、文字の斜体化やくせ字をアフィン変換とみなすことにより、アフィン不変な特徴を用いた文字認識等への応用が可能である。

2.2.5 変換と不変認識

これまで述べてきた不変なパラメータは、変換の度合いがどの程度であっても変わることのないパラメータである。このような不変なパラメータを用いることにより、画像上でのさまざまな変化に影響されない認識を行うことが可能となる。画像認識法の多くがこのような不変なパラメータをもとにした特徴を用いている。本論文においては、3章で、距離の比が相似不変であることを利用した認識方法を、また、4章で面積比がアフィン不変であることを利用した認識方法を述べる。

2.2.6 不変認識の問題点

これまで種々の変換と、不変認識について述べた。しかしながら、このような不変認識でも問題がないわけではない。たとえば、現実的にはありえない変形や、同一のものとは思えないほど変形したものに関してなど、たとえ変換に対しては不変であっても同一のものとして認識すべきでない場合が考えられる。

これに対して、変換の度合いをより現実的なものとする方法や変換の範囲を制限する方法も研究されている。たとえば、佐藤、Cipollaらは変換量が少ない場合のみ不変となるような擬似不変 (quasi-invariant) を定義し、マッチングを行う方法を提案している[39]。また、井上らは、アフィン変換に不変な点パターンマッチングに基づく不変認識方法を提案した上で、その方式が容易に相似不変の範囲に制限できることを示している[66]。

2.3 デジタル図形の輪郭線情報による記述

画像の境界や輪郭線には、その情報量は少ないにもかかわらず認識のための重要な情報が含まれている。画像を処理する場合、その輪郭線を抽出・記述し、これにもとづいて画像の性質や特徴を把握することはきわめて自然な画像処理の方法であるといえる。このような閉曲線画像処理に関してはこれまでも多くの研究があり[19]-[21]，実用的にも数々の提案がなされている[17][53]。

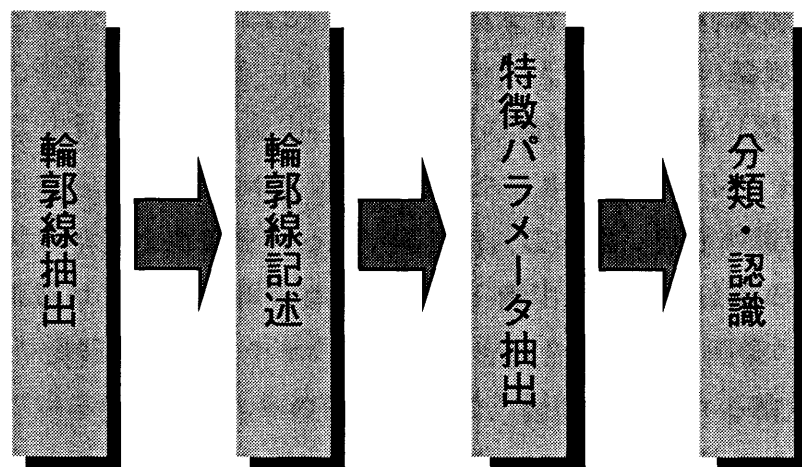


図 2.10 図形認識における輪郭線情報の処理の流れ

図形をその輪郭線情報に着目して処理する場合、輪郭線の抽出、その記述、その記述からの特徴パラメータの抽出、さらに分類・認識などの一連の処理が存在する（図 2.10）。この中で具体的な処理のための最も重要な点は、曲線（輪郭線）をコード化（1次元信号に変換）する際の記述法にある。つまり、曲線を記述し、それ以降の処理の質をよくするためには、特に次に示すような3点を考慮した記述法が望まれる。

1. 前節で述べたようなシフト変換やスケール変換等の変換に対してその記述が不変であること
2. 曲線の記述から元の曲線が再生可能であること
3. デジタル化誤差の影響を受けにくいものであること

などである。本節では、主に種々の曲線記述法について検討する。

2.3.1 輪郭線による図形の記述

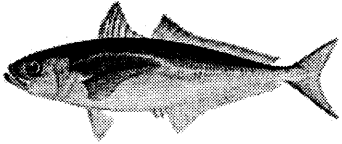
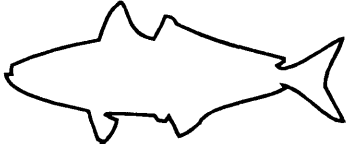
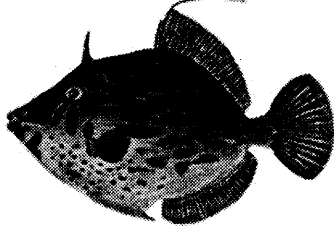
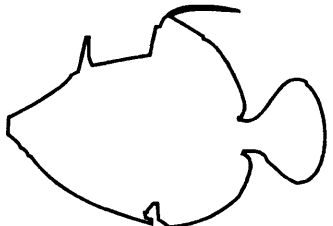
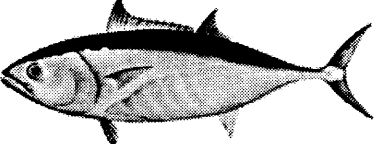
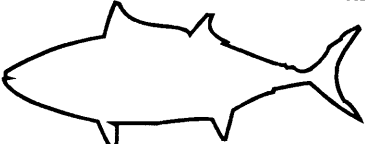
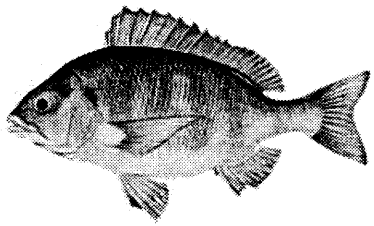
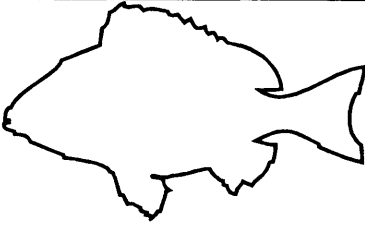
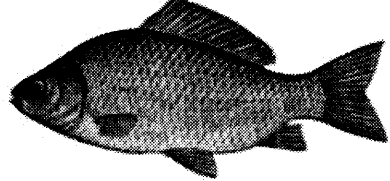
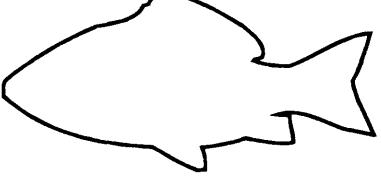
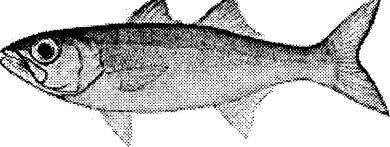

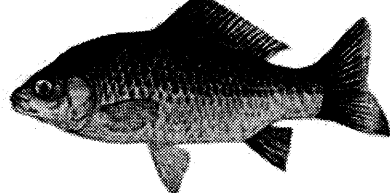
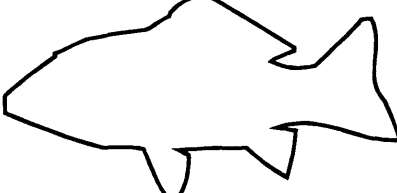
画像の中でも、工業部品のように比較的単純な画像の認識は、特に図形認識と呼ばれる。ここでは、比較的単純な背景上での対象物の輪郭線が認識のための重要な手がかりとなる。また、人間の横顔の輪郭線には重要な情報が含まれている。瀧山らは、人間の横顔の輪郭線（図 2.11）から、曲率値などをもとにし、24次元の特徴ベクトルを求め、シフト・スケール変換に対し不変な認識を行っている[17]。この他、魚種の識別においても魚の輪郭線は重要な情報を持っている。瀧山、小山らは魚の輪郭線（表 2.1）のもつ情報とテクスチャの持つ情報を統合させた魚画像の識別方法を提案している[18]。これらの例のように、画像に含まれる輪郭線等の線図形情報が特徴としてよく用いられている。このような特徴を用いた認識手法では、処理データ量をもとの画像データから大幅に削減することが可能であり、さらに各種の変換に対して不変な認識を比較的容易に実現することができる。

カメラやイメージスキャナ等の画像処理装置を使って処理を行うシステムでは、フレーム内での画像（図形）の位置・尺度・回転などの入力時の状態に対して、常に安定したデータをもたらすような輪郭線記述法が望まれる。すなわち各種の変換に対して不変な輪郭線の記述法が望まれる。



図 2.11 横顔画像とその輪郭線

表 2.1 魚画像とその輪郭線の例

魚種	魚画像	輪郭線
アジ		
カワハギ		
マグロ		
クロダイ		
ヘラブナ		
ムツ		
コイ		

2.3.2 輪郭線記述の種別

輪郭線の記述法には，C.T.Zahn[19]の Curvature Function，白井[22]の曲率関数，安居院ら[23]の偏角関数，上坂[21]の P 型記述子，その他多くの方法が知られている。それぞれの記述子に特徴があり，若干手を加えられることによって広く応用されている。これらはいずれもその記述を直接使用したり，あるいはフーリエ変換してそのスペクトルを用いる（フーリエ記述子）などの方法で認識，および識別へと応用されている。これらは大別してスペクトルから元の輪郭線を再生させることを前提としているものと，そうではないものに分けられる。また，記述に用いる変数によって以下の 3 種類に分けられる。

1. 座標値を直接用いる方法

これは，入力のある座標系と考えると，曲線中の点の座標で直接記述する方法である。

たとえば，G.H.Grandlund[20]は，画像を複素平面と考え，輪郭線を実数方向および虚数方向の座標値の数列で記述する方法で文字認識に応用している。

2. 輪郭線の傾きを用いる方法

これは，座標系に対する輪郭線の傾き（その点における接線の傾き）を使って輪郭線を記述する方法である。この記述は図形の平行移動（シフト変換）に関して不変であり，スケール変換に対しては，輪郭線の長さを正規化しておけば不変となる。また，回転に対しては，記述それ自体は変化するが，多くは簡単な操作で不変なパラメータを得ることができる。たとえば，8 連結のデジタル曲線を 8 方向の傾きを表すコードで記述したものがフリーマンのチェインコードである（図 2.12）。

安居院，中嶋ら[23]は，閉曲線を偏角関数と呼ばれる，接線の傾きを使った関数で記述する方法を提案しており，湖の識別を行っている。また上坂[21]は，輪郭線の傾き（偏角）を複素単

位円の位相と考えると、その複素数値関数で記述する方法（P型記述子と呼ばれる）を考案し、このフーリエスペクトルが特徴パラメータとして優れた性質を持つことを理論的、実験的に示している。また、瀧山ら[53]はP型記述子のパワースペクトルを横顔シルエット認識に応用している。

3. 輪郭線の曲がり具合を用いる記述法

これは、輪郭線を輪郭線自身に対する曲がり具合で記述する方法である。理論的には、前項の輪郭線の傾きを用いる記述法のような類の記述法を微分すればこの記述となる。したがって、シフト変換に加えて回転変換に対しても不変ということになる。またスケール変換に対しては、輪郭線の長さを正規化すれば不変となる。また閉曲線に対して、その記述は周期関数となる性質を持ち合わせている。このために、フーリエスペクトルを不変パラメータとして用いることが可能となる。

白井[22]の曲率関数は、曲がり具合を角（ラジアン）で表したものであり、曲がりの大きいコーナー部の検出等に使用されている。また安居院、中嶋ら[24]は、曲率関数のパワースペクトルを使用して、アニメーションの閉領域の同定に応用している。

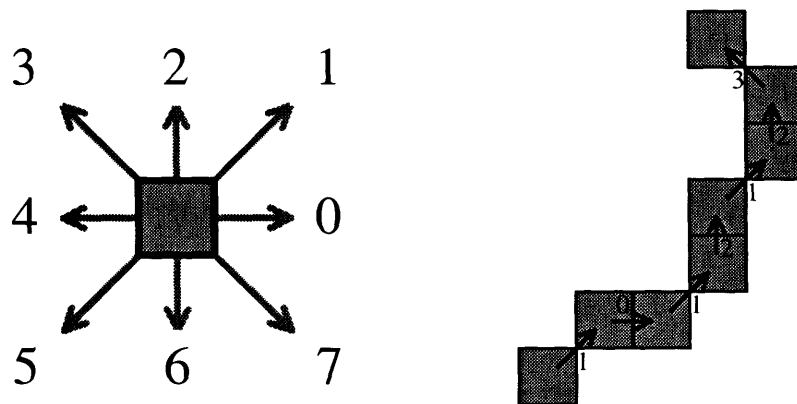


図 2.12 チェインコードと曲線のチェインコード化

2.3.3 偏角関数

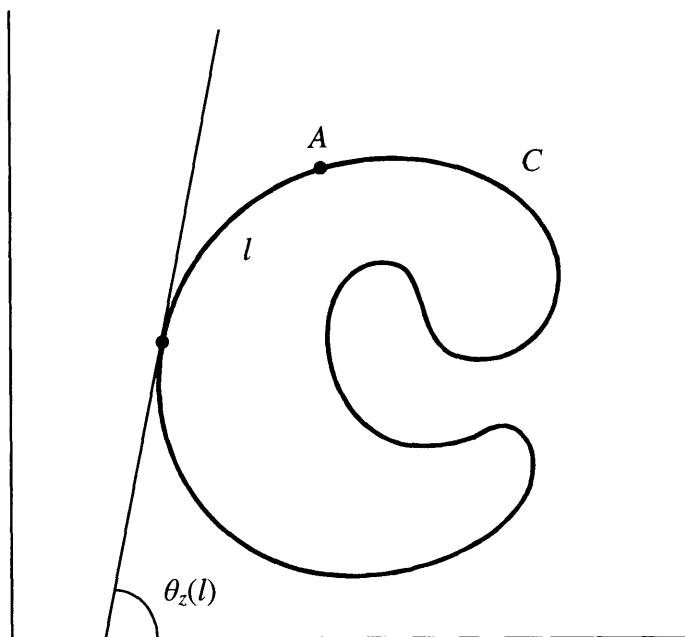


図 2.13 閉曲線の偏角

図 2.13に示すように、平面上の輪郭線 C 上に任意の点 A をとり、 A から C に沿った弧の長さを l とする。 l を媒介変数（パラメータ）として輪郭線 C を $Z(l)$ で表す。ただし l の符号は反時計回りを正とする。 C を一周したときの長さを L として、次に定義する $\phi_z(l)$ で閉曲線を表す。

$$\phi_z(l) = \theta_z(l) - 2\pi \frac{l}{L} \quad (\text{式 2.12})$$

ただし、 $\theta_z(l)$ は距離 l での点の接線の偏角である。

この $\phi_z(l)$ の図形的な意味は次のように考えられる。図 2.14 のように閉曲線 C の接線ベクトルを $r(l)$ で表す。また一方曲線 C と周囲長が等しい円を考えて、その円の下点を始点としたときの接線ベクトルを $t(l)$ で表せ

ば、 $r(l)$ と $t(l)$ とのなす角 ϕ が、その点における閉曲線 C の偏角関数 ϕ となる。

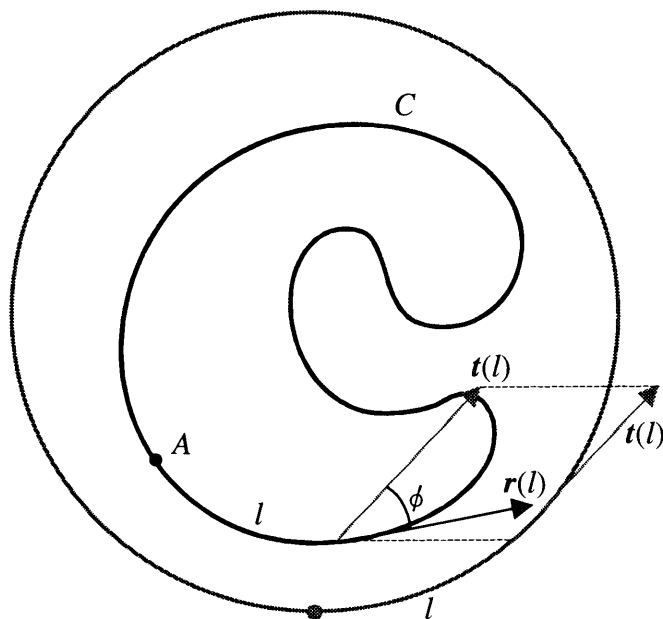


図 2.14 偏角関数 ϕ

$\phi_z(l)$ は周期 L の周期関数であり、閉曲線の大きさに依存するために、次式のように $\phi_z^*(x)$ を用いて区間 $[0, 2\pi]$ に正規化する。

$$\phi_z^*(x) = \theta_z \left(\frac{L}{2\pi} x \right) - x \quad (\text{式 2.13})$$

このとき、 $\phi_z^*(0) = \phi_z^*(2\pi) = \theta_z(0)$ である。また、 $\phi_z^*(x)$ は閉曲線のシフト変換やスケール変換に対して不変な記述である。

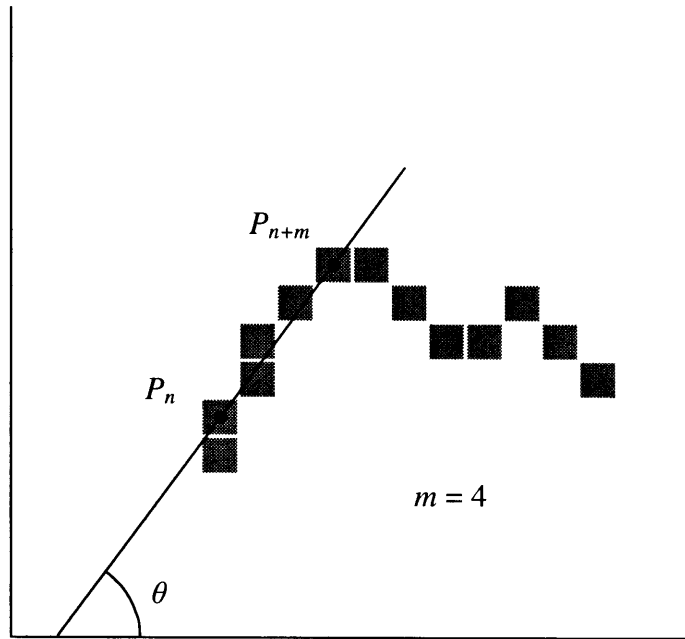


図 2.15 デジタル曲線における偏角

デジタル曲線に対する偏角を、図 2.15に示す。注目すべき画素を P_n とし、これよりも m 個だけ先の画素を P_{n+m} として、この2点を直線で結びこれを P_n における接線とする。偏角はこの接線が x 軸となす角であるから以下の式で与えられることになる。

$$\theta_z^*(n) = \tan^{-1} \left(\frac{y_{n+m} - y_n}{x_{n+m} - x_n} \right) \quad (\text{式 2.14})$$

ここで、 (x_{n+m}, y_{n+m}) , (x_n, y_n) はそれぞれ画素 P_{n+m} , P_n の座標である。また、 m はステップ数で、スケールに比例させる必要がある定数である。ここで、ステップ数はあまり小さすぎると曲線の形の大まかな情報に欠け、雑音や後述するデジタル化誤差に影響されやすく、逆に大きすぎると、局所的な情報に欠けるということになってしまう。この意味でステップ数の設定は重要であり、十分な注意が必要である。

2.3.4 曲率関数

曲率関数 $C(l)$ は、閉曲線の接線の偏角 $\theta(l)$ を微分したものである。すなわち、

$$C(l) = \frac{d\theta(l)}{dl} \quad (\text{式 2.15})$$

デジタル曲線の場合、微分のかわりに差分を用いて得られる。すなわち、

$$\begin{aligned} C(n) &= \theta(n) - \theta(n - m) \\ &= \tan^{-1} \left(\frac{y_{n+m} - y_n}{x_{n+m} - x_n} \right) - \tan^{-1} \left(\frac{y_n - y_{n-m}}{x_n - x_{n-m}} \right) \end{aligned} \quad (\text{式 2.16})$$

で得られることになる。

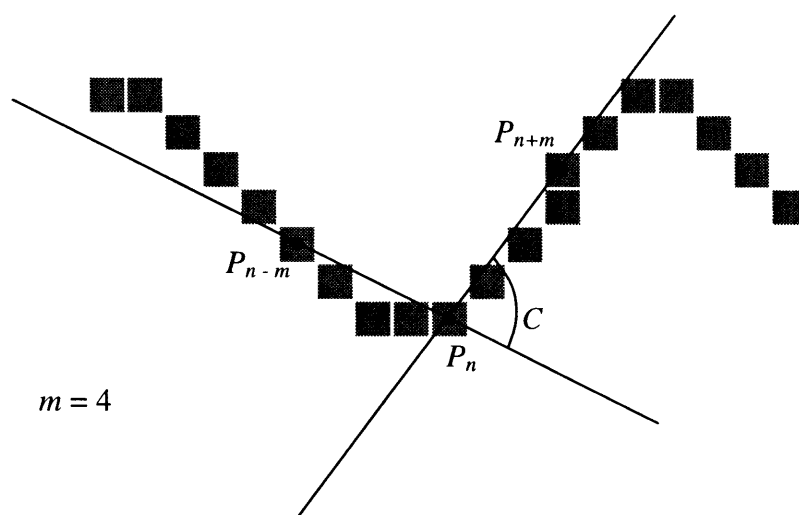


図 2.16 デジタル曲線における曲率

図 2.16に示すように、注目する画素 P_n と、これよりも m 個だけ前後の画

素 P_{n-m} と P_{n+m} とをそれぞれ直線で結んで、これらが交わる補角における曲率と定義する。

曲率関数は、曲線が閉曲線であれば、周囲長を周期とする周期関数となる。パラメータ n が時計回りを正の方向とするならば、閉曲線の凸の部分に対して正の値（ラジアン）を示す（ n が反時計回りを正の方向とするのであれば、正負は逆転する）。また閉曲線のシフト・回転変換に対しては不変であり、スケール変換に対しても周囲長を正規化すれば不変となる。さらに始点の移動に対しては、その位相がずれるだけであるので、このパワースペクトルを不変認識における特徴パラメータとすることが可能となる。

また、曲率関数は閉曲線の曲がりの大きいコーナー部でそのピーク値をとるので、視覚的に重要な特徴点であるコーナーの検出が容易である。この特徴を利用して、瀧山ら[17]は横顔のシルエットからのあご、口、鼻、額といった点の測定に応用している。

2.3.5 フーリエ記述子

輪郭線を周波数領域で表す方法としてフーリエ記述子がある。前項（曲率関数）で述べたような、座標値、傾き、曲がり具合などで輪郭線を記述したものを被展開関数としてフーリエ級数に展開し、得られるスペクトルの組を輪郭線のフーリエ記述子と呼ぶ。輪郭線記述が離散数値関数であるならば、これを、 $p(j), j=0, 1, \dots, n-1$ (n は十分大きい正の整数) として、フーリエ記述子 q は、次の離散フーリエ変換式で得られる。

$$q(k) = \frac{1}{n} \sum_{j=0}^{n-1} p(j) \exp\left(-2\pi i \frac{jk}{n}\right) \quad (\text{式 2.17})$$

$$k = 0, 1, \dots, n-1$$

$$i = \sqrt{-1}$$

以降では、G型記述子、Z型記述子、P型記述子などの代表的なフーリエ記述子について述べる。ただし、説明の便宜上、次のような仮定をしておく。

1. 処理の対象となる輪郭線は、複素平面上に与えられている折れ線図形であるとする。
2. 与えられている折れ線図形は各線分の長さがどれも等しいものとする。
3. 折れ線図形とその記述は一対一に対応しており、変換および逆変換が一意にできるものとする。

2.3.5.1 G型記述子

長さが δ である n 本の線分をつなぎ合わせてできる折れ線図形を C とする。 C は各線分における端点の集合で表せられる。この端点の座標を $(x(j), y(j))$ とすると、

$$C = \{x(j), y(j)\}, \quad j = 0, 1, \dots, n \quad (\text{式 2.18})$$

をもとの輪郭線とみなすことができる。

xy 座標値を使って、複素数値関数 z を次のように定める。

$$z(j) = x(j) + iy(j) \quad j = 0, 1, \dots, n \quad (\text{式 2.19})$$

このようにすると、複素数値関数 z は、折れ線図形 C と同一視することができる。点 $z(0), z(n)$ が C の端点であり、 C が閉曲線の場合には、 $z(0) = z(n)$ である。2番目の仮定により、折れ線の各辺の長さはすべて等しいので、次式

$$\delta = |z(j) - z(j-1)| \quad j = 1, 2, \dots, n \quad (\text{式 2.20})$$

が成り立っている。与えられた任意の輪郭線を上の複素数値関数で近似的に表すことは容易にできるので、一般性を失うことなく式 2.20 を満たす z が輪郭線 C としてはじめに与えられているものとする。

この複素数値関数 z を被展開関数としてフーリエ級数に展開し得られたスペクトル

$$e(k) = \frac{1}{n} \sum_{j=0}^{n-1} z(j) \exp\left(-2\pi i \frac{jk}{n}\right) \quad (\text{式 2.21})$$

を折れ線図形の「G型記述子」と呼ぶ。

2.3.5.2 Z型記述子

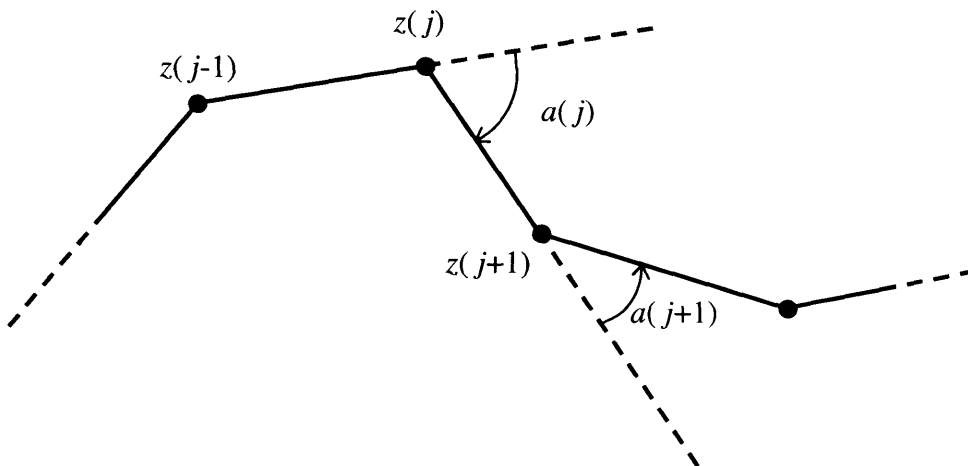


図 2.17 補角

図 2.17に示すように、折れ線図形 C 上に隣り合った線分をベクトル

$z(j) - z(j-1)$ とベクトル $z(j+1) - z(j)$ とし, これらがなす角 (補角) を反時計回り方向に正に, あるいは時計回り方向に負に測って, その値を $a(j)$ とする。 $a(j)$ は,

$$-\pi \leq a(j) < \pi \quad (\text{式 2.22})$$

の範囲におさまるようにとる。ただし, $a(0)$ はベクトル $z(1) - z(0)$ と x 軸とのなす角 (偏角) と定める (図 2.18)。この $a(j)$ を点 $z(j)$ における曲率という。

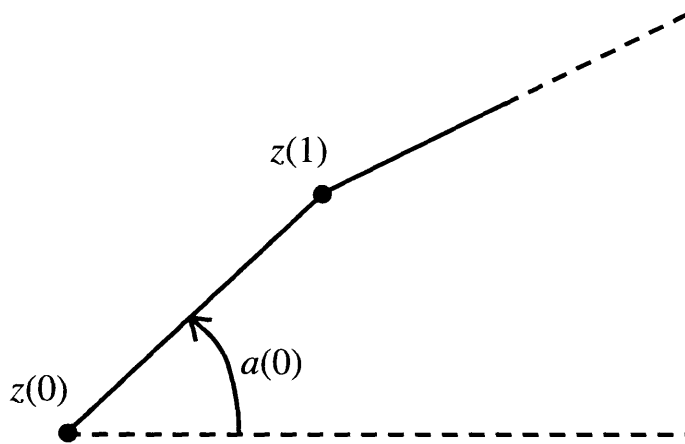


図 2.18 偏角

次に曲率関数 a を用いて折れ線図形 C の偏角関数 θ を,

$$\theta(0) = a(0)$$

$$\theta(j) = \theta(j-1) + a(j), \quad j=1, 2, \dots, n-1 \quad (\text{式 2.23})$$

と定める。偏角関数はもともと滑らかな連続輪郭線上の曲率を輪郭線に沿って積分したものであるが, この概念を折れ線図形に当てはめると前式のようなになる。

この θ を使って

$$\phi(j) = \theta(j) - \theta(0) - 2\pi \frac{j}{n} \quad (\text{式 2.24})$$

とおき, これのフーリエ変換を

$$d(k) = \frac{1}{n} \sum_{j=0}^{n-1} \phi(j) \exp\left(-2\pi i \frac{jk}{n}\right) \quad (\text{式 2.25})$$

とし, $d(k)$ を折れ線図形の「Z型記述子」と呼ぶ。

2.3.5.3 P型記述子

偏角 θ を用いて，複素数値関数 ω を

$$\omega(j) = \exp(i \theta(j)), \quad j=1, 2, \dots, n-1 \quad (\text{式 2.26})$$

と定め，これを折れ線図形 C の「P 表現」と呼ぶ。前式において， ω を折れ線図形 C の信号としてみたとき，ちょうど位相(Phase)の場所に θ が位置している。P 表現の P はこのことに由来している。

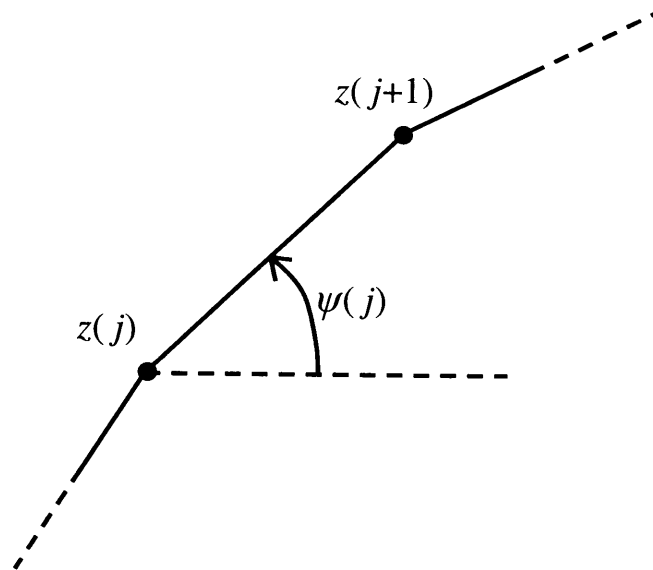


図 2.19 x 軸とのなす角 $\psi(j)$

曲率と偏角関数の定め方から容易にわかるように，ベクトル $z(j+1) - z(j)$ と，x 軸とのなす角 ψ (図 2.19) は $\theta(j)$ と 2π の整数倍を除いて等しい。したがって，

$$\begin{aligned}\exp(i\theta(j)) &= \exp(i\psi(j)) \\ &= \frac{z(j+1) - z(j)}{\delta}\end{aligned}\quad (\text{式 2.27})$$

が成り立ち、式(2.26)により、

$$\varpi(j) = \frac{z(j+1) - z(j)}{\delta}\quad (\text{式 2.28})$$

が得られる。逆に $\varpi(j), j = 1, 2, \dots, M$ が与えられれば、座標は

$$z(j) = z(0) + \delta \sum_{\gamma=0}^{j-1} \varpi(\gamma), \quad j = 1, 2, \dots, n \quad (\text{式 2.29})$$

によって求めることができる。

この P 表現 $\varpi(j)$ をフーリエ変換して得られる結果が、「P 型記述子」と呼ばれるものである。すなわち、折れ線図形 C の P 型記述子 $c(k)$ は

$$c(k) = \frac{1}{n} \sum_{j=0}^{n-1} \varpi(j) \exp\left(-2\pi i \frac{jk}{n}\right), \quad k = 1, 2, \dots, n-1 \quad (\text{式 2.30})$$

によって得られる。この記述子は、輪郭線のシフト変換、スケール変換に対して不変である。また、この折れ線図形が反時計回りに α だけ回転したとき、折れ線図形 $c'(k)$ は、

$$c'(k) = \exp(i\alpha) c(k) \quad (\text{式 2.31})$$

となる。したがって、両者の絶対値は等しい。始点の移動についても、同様に絶対値を取ることによって不変な結果が得られる。

以上のように、折れ線図形を P 型記述子によって表し、その絶対値をとれば、対象のシフト・回転・スケール変換および輪郭線記述の際の始点の位置によらない不変な記述が得られる。瀧山、王らは、P 型記述子のパ

ワースペクトルを横顔シルエット認識に応用して高い識別結果を得ている
(図 2.20)。

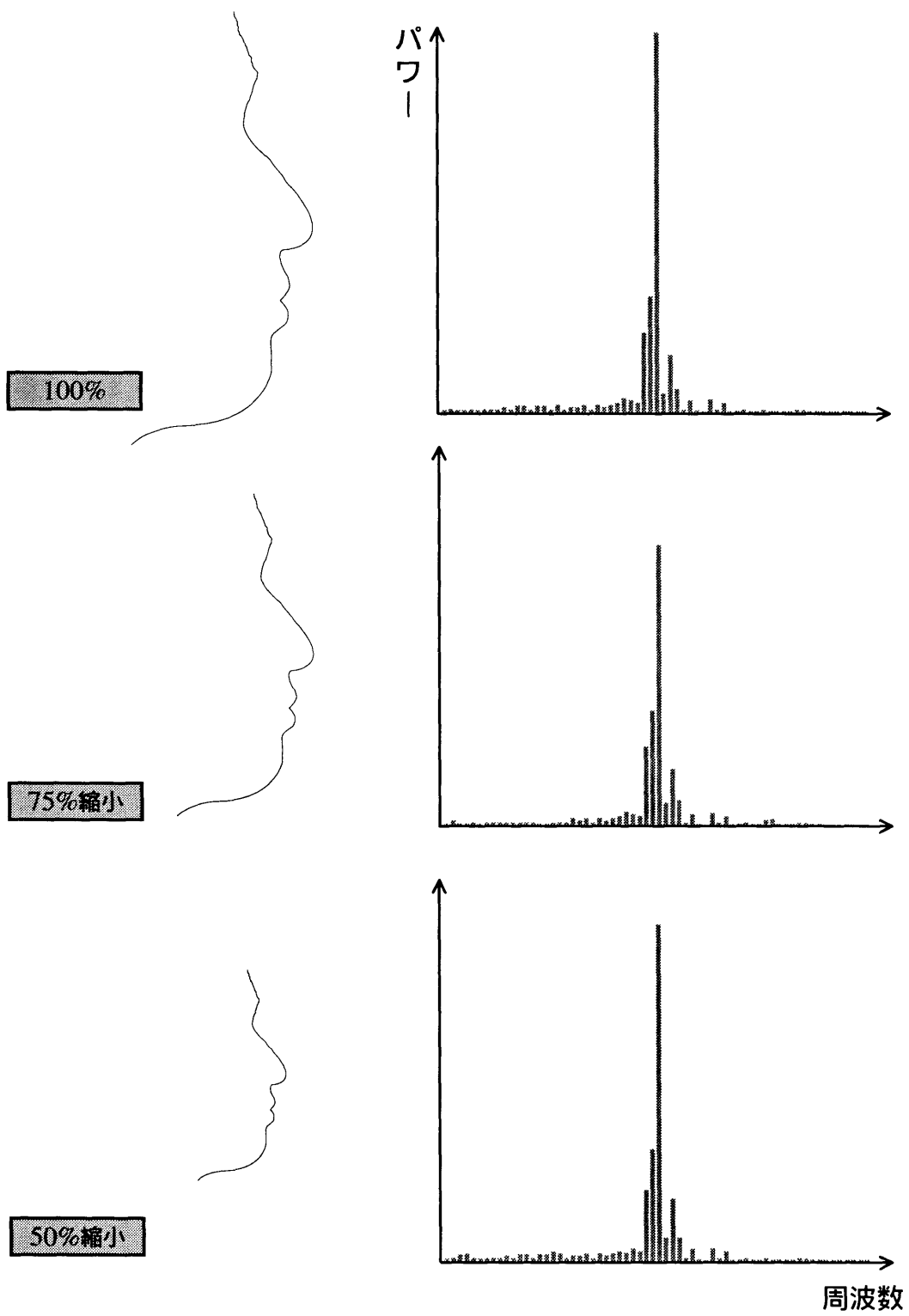


図 2.20 横顔輪郭線のパワースペクトルの例

2.4 デジタル図形を取り扱う際の問題

画像処理システムを使って多くの画像をパターンとして入力し、相似変換やアフィン変換などに対して不変認識を行うためには、デジタル化された画像をさらにサンプリングして正規化する必要がある。

図 2.21は、画像の輪郭線によるパターン認識の過程を示している。輪郭線記述をデジタル画像に適用して各種の変換に対して不変な処理を行う場合には、

- 距離をどう定義するか
- 正規化と曲線記述子によるパラメータ表記の順序
- デジタル化誤差をどの程度考慮するか

などの多くの問題点がある。本節では、正規化・輪郭線記述過程における上記のような諸問題について述べる。

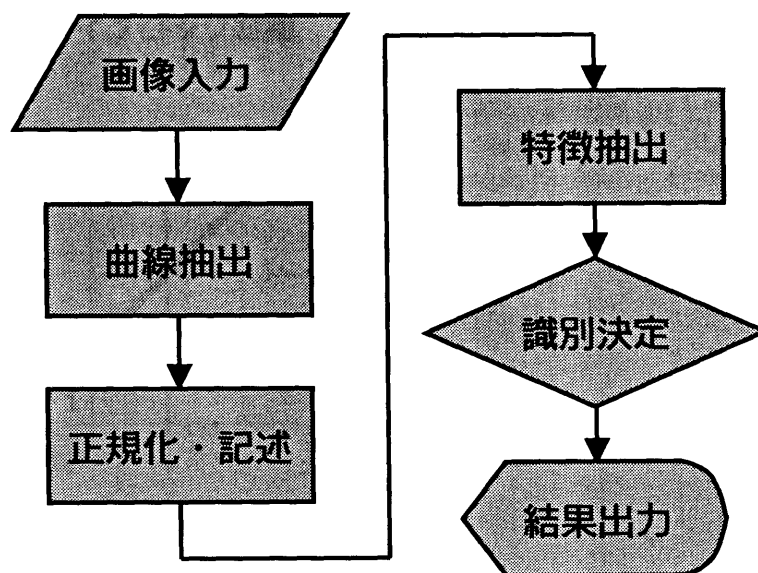
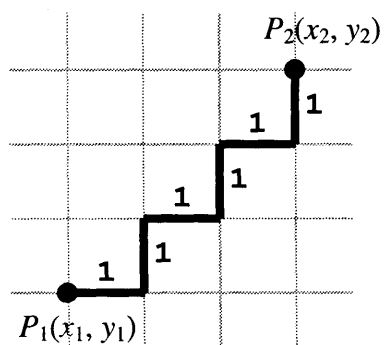


図 2.21 画像の輪郭線によるパターン認識過程

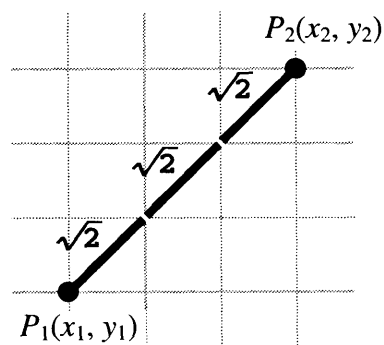
2.4.1 距離の定義

直角格子によって標本化されたデジタル輪郭線を記述する際には、輪郭線に沿った各画素間の距離の総和である「長さ」というものの取り扱いが問題になってくる。

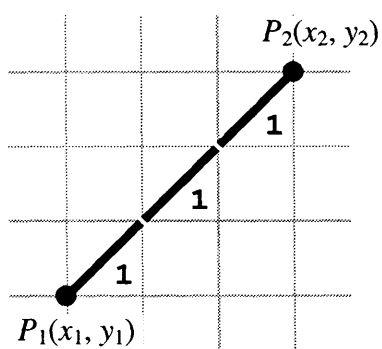
通常、8連結のデジタル輪郭線の長さを測るには、図 2.22に示すような3通りの距離の定義がよく用いられる。



(a) シティブロック距離



(b) ユークリッド距離



(c) チェス盤距離

図 2.22 デジタル輪郭線における距離

2.4.1.1 シティブロック距離(city block distance)

図 2.22(a)に示す 2 点 $P_1(x_1, y_1)$, $P_2(x_2, y_2)$ 間のシティブロック距離を d_a とすると,

$$d_a = |x_2 - x_1| + |y_2 - y_1| \quad (\text{式 2.32})$$

で表せる。輪郭線図形の長さをシティブロック距離で表す場合、輪郭線を構成する画素のうち隣り合う画素間が 4 連結の結合の場合 1, そうでない場合 2 とし, その総和が輪郭線の長さになる。シティブロック距離は, 4 近傍距離(4-neighbor distance)とも呼ばれる。

2.4.1.2 ユークリッド距離(Euclidean distance)

図 2.22(b)に示す 2 点 $P_1(x_1, y_1)$, $P_2(x_2, y_2)$ 間のユークリッド距離を d_b とすると,

$$d_b = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (\text{式 2.33})$$

で表せる。輪郭線図形の長さをユークリッド距離で表す場合、輪郭線を構成する画素のうち隣り合う画素間が 4 連結の結合の場合 1, そうでない場合 $\sqrt{2}$ とし, その総和が輪郭線の長さになる。

2.4.1.3 チェス盤距離(chess board distance)

図 2.22(c)に示す 2 点 $P_1(x_1, y_1)$, $P_2(x_2, y_2)$ 間のチェス盤距離を d_c とすると,

$$d_c = \max(|x_2 - x_1|, |y_2 - y_1|) \quad (\text{式 2.34})$$

で表せる。輪郭線図形の長さをチェス盤距離で表す場合、輪郭線を構成する画素の総数が輪郭線の長さに相当する。チェス盤距離は、8 近傍距離 (8-neighbor distance)とも呼ばれる。

2.4.1.4 中心画素からの距離の違い

正方格子における中心画素からの距離を、上述したそれぞれの距離で測ったときの値を図 2.23に示す。このように、方式によって距離の値が大きく異なるような複数の距離の定義が共存しているのは、輪郭線の距離の意味合いの効力が、そのデジタル画像内に限定されているからであることは言うまでもない。

6	5	4	3	4	5	6
5	4	3	2	3	4	5
4	3	2	1	2	3	4
3	2	1	0	1	2	3
4	3	2	1	2	3	4
5	4	3	2	3	4	5
6	5	4	3	4	5	6

(a) シティブロック距離

$\sqrt{18}$	$\sqrt{10}$	3	$\sqrt{10}$	$\sqrt{18}$		
$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{8}$		
$\sqrt{10}$	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{10}$
3	2	1	0	1	2	3
$\sqrt{10}$	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{10}$
$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{8}$		
$\sqrt{18}$	$\sqrt{10}$	3	$\sqrt{10}$	$\sqrt{18}$		

(b) ユークリッド距離

3	3	3	3	3	3	3
3	2	2	2	2	2	3
3	2	1	1	1	2	3
3	2	1	0	1	2	3
3	2	1	1	1	2	3
3	2	2	2	2	2	3
3	3	3	3	3	3	3

(c) チェス盤距離

図 2.23 中心画素からの距離

2.4.2 デジタル化による誤差

前述した3種類の距離による輪郭線図形の長さは共に実際の長さ按比例するものではなく、デジタル化誤差を多分に含んでいるため非線形に変化する。従来、標本化レベルが十分である場合は、どのような距離の定義を使った場合でも、そのデジタル化誤差は黙認してかまわないとされてきた。しかし、現実にはこのようなデジタル化誤差を問題とすべきことが起きる場合がある。

図 2.24の(a), (b)は同一の直角三角形であるが、座標系が45度ずれている。左下の端を始点として輪郭線に沿ってチェス盤距離を用い、等間隔に4つの点をサンプリングした結果を黒画素で表示している。結果は、同一の図形にもかかわらず、それぞれのサンプリング点の位置は異なったものとなっている。

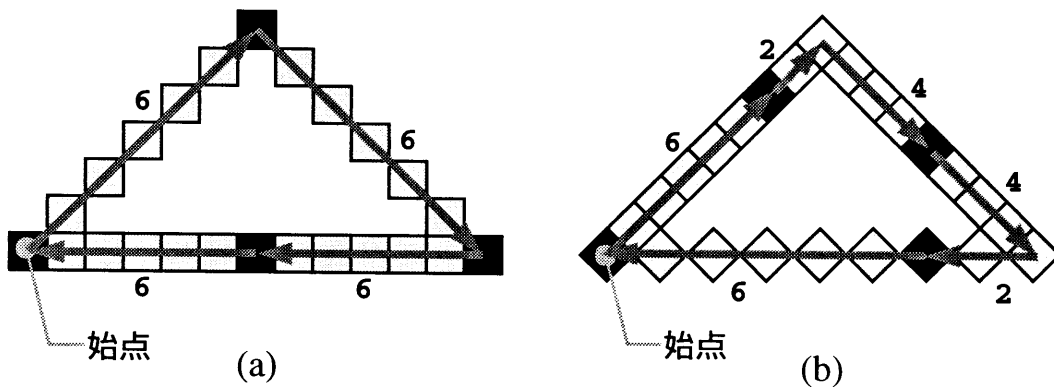


図 2.24 デジタル化誤差

輪郭線記述法の多くは、輪郭線に沿った弧の長さをパラメータとした関数である。このためこれらは実際の長さの変化に対して非線形に変化し、その性質は大きく左右されるため安定性に欠けるのである。特に画像(図

形)の回転に対しては、画素の総数が同じものであってもそのユークリッド距離は変わってしまう。これはデジタル輪郭線の性質上避けることのできない欠点である。

このような欠点を抑えたデジタル化の方式として、Golay[67]が提案した正六角形格子(hexagonal grid)を用いたものがある(図 2.25)。これは、標本点が正六角形の配列をしたものであって、それぞれの点は等距離に6つの隣接点を持つものである。距離が1と $\sqrt{2}$ の2種類のところに隣接点を持つ直角格子方式と比べて、正六角形格子方式は隣り合う画素との距離がただ一つに決められるために、デジタル化誤差が少なくすむことになる。しかしながら、このような方式は多くの画像処理アルゴリズムや画像の変換に対して不都合があるため、あまり広くは用いられていない。

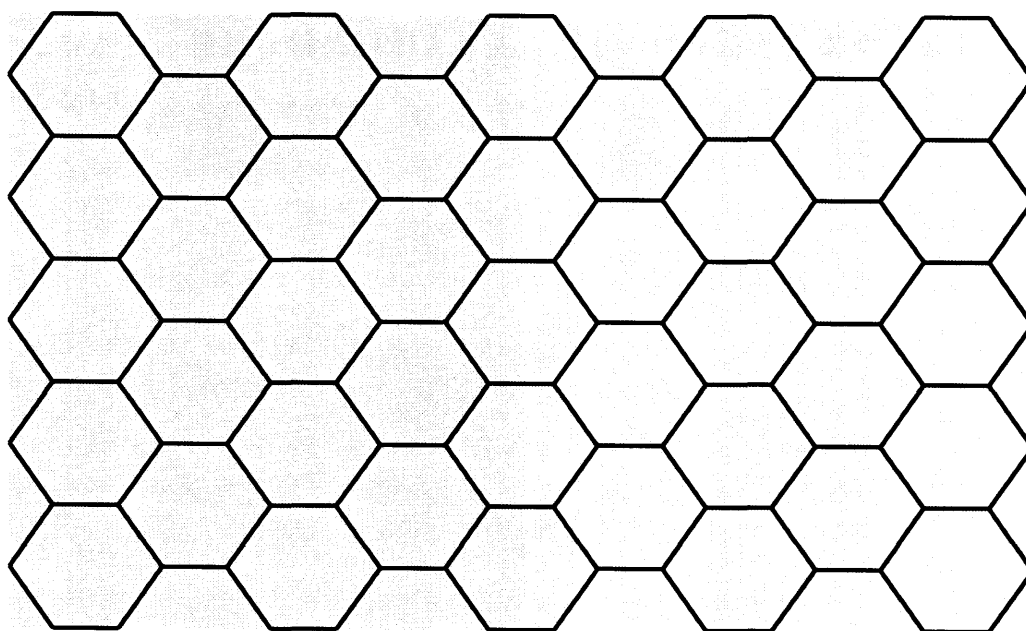


図 2.25 正六角形格子

現在のデジタル画像処理の分野では、ほとんどが直角格子を用いたデジタル化を採用している。したがって、先の欠点を抑えるために、デジタル化時の誤差をできるだけ補正していくための事後処理を行うことが必要である。先に述べたシティブロック距離 d_a 、ユークリッド距離 d_b 、そしてチェス盤距離 d_c を用いて、2つの画素間の距離を測った場合、それらの間の大小関係は次のようになる。

$$d_a \geq d_b \geq d_c \quad (\text{式 2.35})$$

前述したそれぞれの距離の定義において輪郭線の長さを測ったときの性質を調べてみると、次のようなことがわかる。チェス盤距離を用いた長さよりもシティブロック距離を用いた長さの方が長くなるのは明らかであるが、その値の差は非常に大きい。また、直線部の多い単純な形の閉曲線であると、ユークリッド距離を用いた長さが実際の長さに近い値を示す。しかしながら、複雑な輪郭線が入り込んだものになると、必ずしもそうとはいえない。

2.4.3 図形の連結性

曲線（図形）には大きく分けて単連結な曲線とそうでない（非単連結）曲線がある。ここで単連結とは、互いに連結している画素群を一つの連結成分としたとき、曲線が一つの連結成分からできているもののことを呼ぶ（図 2.26）。

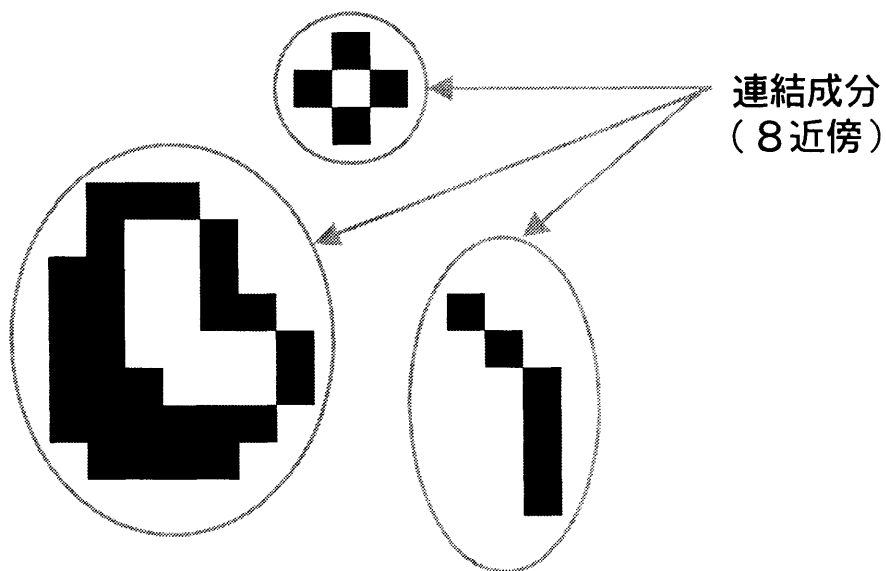


図 2.26 連結成分

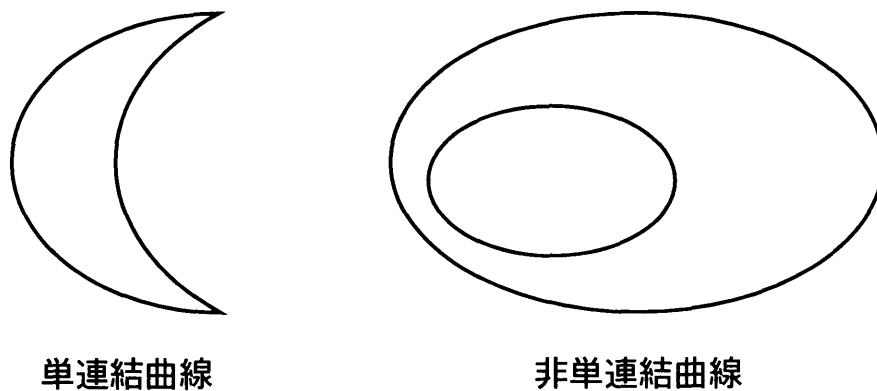


図 2.27 単連結曲線と非単連結曲線の例

先に述べたフーリエ記述子等の曲線記述法は、いずれも本質的には単連結な曲線にのみ有効に適用される。つまりこのような従来法は、通常の方法では二重曲線などの非単連結な曲線には向いていないといえる。これは、従来法ではデータすなわちデジタル曲線における画素の並びが一意でない場合、互いにある種の変換の関係にある曲線同士を認識できないからである。任意に与えられた単連結でない曲線に対して、画素の並びを一意に決めることは一般にはむずかしい。

2.5 まとめ

本章では、基礎理論として画像の変換と不変認識の関係、一般的によく用いられる種々の曲線記述法とその問題点について述べた。

不変認識を行うためには位置や大きさなど画像のさまざまな変化（変換）に対して不変な特徴を特徴パラメータとして用いる必要がある。種々の変換に対して不変なパラメータとして、画像の輪郭線を曲線と見立てた上で、その曲線を P 型記述子をはじめとするフーリエ記述子等の曲線記述法により表記し、このような曲線表記を特徴パラメータのもととする方法がよく用いられている。

しかしながらこのような従来の曲線記述法を用いた認識方法では、対象となる曲線（図形）が単連結である必要があり、非単連結な曲線に対しては適用が難しい。

本研究ではこのような問題点に着目し、図形の画素間の情報を用いる、新たな概念を利用し、非単連結な図形にも適用可能な認識方法について提案しており、後の章にて述べる。第 3 章では、図形を構成する画素間の距離情報を利用した相似変換に対して不変な認識方法を、また、第 4 章では図形を構成する画素間の面積の情報を用いたアフィン変換に対して不変な認識方法について述べる。