2値画像処理Ⅱ-特徴抽出と符号化-(3.3)

- : 2値画像を構成する連結成分の を ことで、連結成分を に する処理。
- : 2値画像を構成する連結成分の を ことで、連結画素を に する処理。
- 元画像を $f(i,j)$、変換後の画像を $g_t(i,j)$ とすると、上記処理(8近傍処理の場合)は以下のようになる。

膨張処理 $g_t(i,j) = \left\{ \right.$ 収縮処理 $g_t(i,j) = \left\{ \right.$

但し,$-1 \le k, l \le 1$で$f(i+k, j+l)$は$f(i,j)$自身と周囲 8近傍画素。

kかlのどちらかが 0の場合は周囲 4近傍画素となる。

- p.38最終行の「膨張した画像」は「収縮した画像」の誤り。
- 図3.11に膨張と収縮処理の適用例がある。
- 図3.11を参考にして、次の元画像を膨張後収縮処理しなさい。
- 次に、同じ元画像を収縮後膨張処理しなさい。
- 但し、処理は4近傍ではなく、8近傍行うものとする。
- また、各処理は1回のみとする。
- なお、画像の周囲は0画素で囲まれているものとする。つまり、実際の画素はないが仮想的に0画素があるものとして近傍を考える。

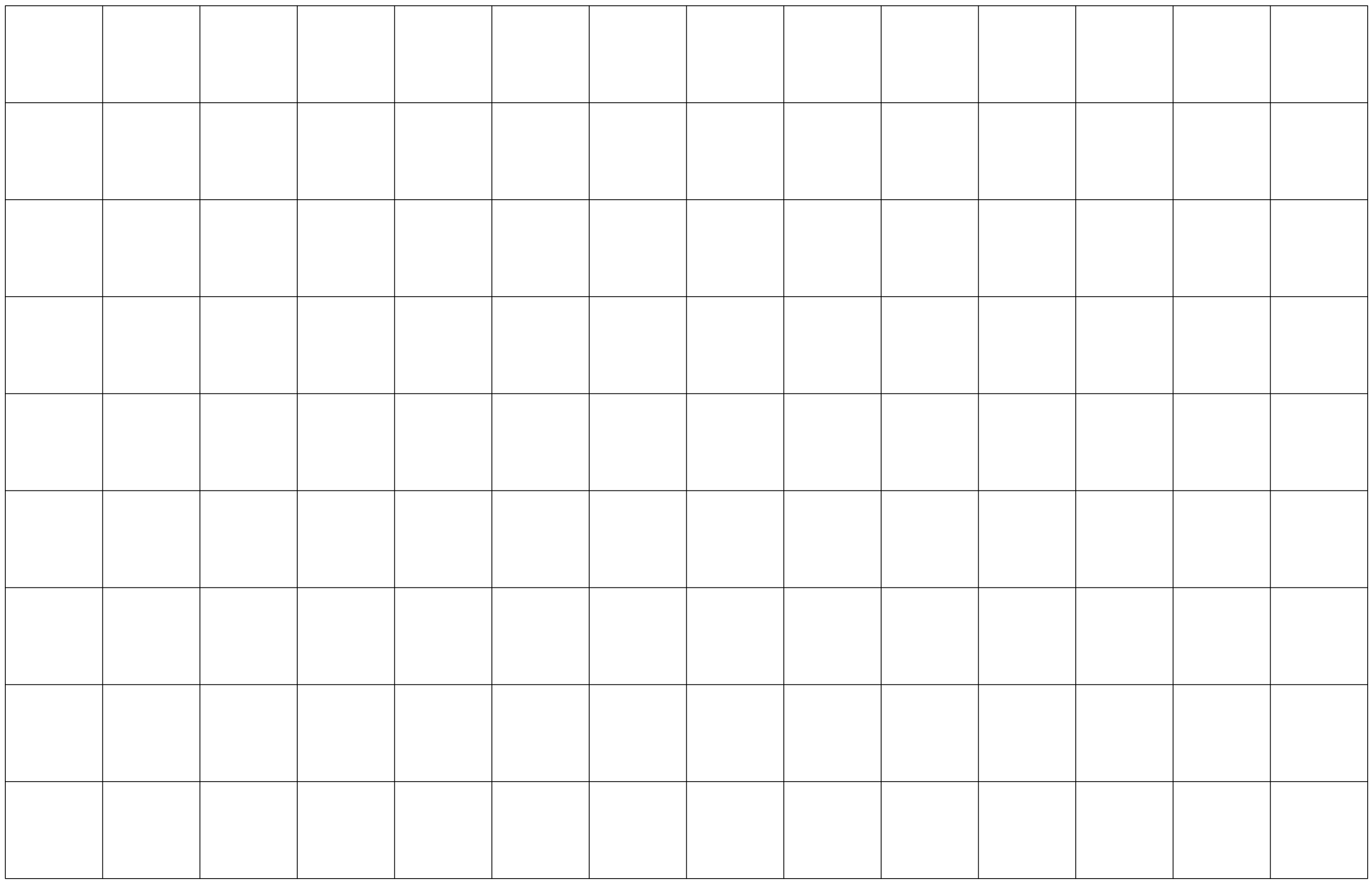
- 元画像

	1	1	1	1	1	1	1						
	1	1	1	1	1	1	1						
	1	1	1	1	1	1	1			1	1	1	
	1	1	1		1	1	1	1		1		1	
	1	1	1	1	1	1	1			1	1	1	
	1	1	1		1	1	1						
	1	1	1		1	1	1						

- 元画像を膨張処理した画像

- 元画像を膨張処理した後、収縮処理した画像

- 元画像を収縮処理した画像



- 元画像を収縮処理した後、膨張処理した画像

- 膨張後収縮の解答

	1	1	1	1	1	1	1						
	1	1	1	1	1	1	1						
	1	1	1	1	1	1	1			1	1	1	
	1	1	1		1	1	1	1		1		1	
	1	1	1	1	1	1	1			1	1	1	
	1	1	1		1	1	1						
	1	1	1		1	1	1						

元画像

膨張後の画像

膨張後収縮の画像

- 収縮後膨張の解答

	1	1	1	1	1	1	1						
	1	1	1	1	1	1	1						
	1	1	1	1	1	1	1			1	1	1	
	1	1	1		1	1	1	1		1		1	
	1	1	1	1	1	1	1			1	1	1	
	1	1	1		1	1	1						
	1	1	1		1	1	1						

元画像

収縮後の画像

収縮後膨張の画像

- 膨張と収縮処理には以下の性質がある。

- した画像と した画像は異なる。
- 膨張後収縮した場合:

 ① は消滅する。(1画素とは限らない)

 ② は消滅する。
 (上記同様、1画素とは限らない。)
- 収縮後膨張した場合:

 ① (飛び出している領域)は消滅する。
 (上記同様、1画素とは限らない。)

 ② (近傍が全て1の画素を
 領域)は消滅する。

距離変換(distance transformation)

- 画像中にある対象物体内の　　　　　　　　をその画素から　　　　(対象物体以外の領域)までの　　　　　　　　　　で置換する処理を　　　　　　　　　　という。
- 変換により得られる画像を　　　　　　　　　　　　　という。
- 対象物体内部の画素ほど大きな値に変換される。
- 距離計算は4近傍と8近傍で異なる。

距離変換アルゴリズム

ア) 初期化：

元画像$f(i, j)$を距離変換初期化画像 $d'(i, j)$に変換する。

$$d'(i, j) = \begin{cases} \\ \end{cases}$$

イ)ラスタ走査：

左上から右下へのラス タ走査で、 $d''(i, j)$に変換する。

$d''(i, j) =$

$d''(i, j) =$

ウ) 逆ラスタ走査：

右下から左上への逆ラ スタ走査で、 $d(i, j)$に変換する。

$d(i, j) =$

$d(i, j) =$

注) p.40中ほどの「ステップ (*b*)」は「ステップ (イ)」の誤り。

4近傍処理

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	0	0	0	0	0	0
0	1	1	1	1	1	1	1	0	0	0	0	0	0
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0	0	0	0	0	0
0	1	1	1	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

2値画像

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	∞	∞	∞	∞	∞	∞	∞	0	0	0	0	0	0
0	∞	∞	∞	∞	∞	∞	∞	0	0	0	0	0	0
0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0
0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0
0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0
0	∞	∞	∞	∞	∞	∞	∞	0	0	0	0	0	0
0	∞	∞	∞	∞	∞	∞	∞	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

初期化後

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	0	0	0	0	0	0
0	1	2	2	2	2	2	2	0	0	0	0	0	0
0	1	2	3	3	3	3	3	1	1	1	1	1	0
0	1	2	3	4	4	4	4	2	2	2	2	2	0
0	1	2	3	4	5	5	5	3	3	3	3	3	0
0	1	2	3	4	5	6	6	0	0	0	0	0	0
0	1	2	3	4	5	6	7	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

ラスター変換後

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	0	0	0	0	0	0
0	1	2	2	2	2	2	1	0	0	0	0	0	0
0	1	2	3	3	3	3	2	1	1	1	1	1	0
0	1	2	3	4	4	4	3	2	2	2	2	1	0
0	1	2	3	3	3	3	2	1	1	1	1	1	0
0	1	2	2	2	2	2	1	0	0	0	0	0	0
0	1	1	1	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

逆ラスター変換後

8近傍処理

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	0	0	0	0	0	0
0	1	1	1	1	1	1	1	0	0	0	0	0	0
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0	0	0	0	0	0
0	1	1	1	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

2値画像

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	∞	∞	∞	∞	∞	∞	∞	0	0	0	0	0	0
0	∞	∞	∞	∞	∞	∞	∞	0	0	0	0	0	0
0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0
0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0
0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0
0	∞	∞	∞	∞	∞	∞	∞	0	0	0	0	0	0
0	∞	∞	∞	∞	∞	∞	∞	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

初期化後

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

ラスター変換後

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	0	0	0	0	0	0
0	1	2	2	2	2	2	1	0	0	0	0	0	0
0	1	2	3	3	3	2	1	1	1	1	1	1	0
0	1	2	3	4	3	2	2	2	2	2	2	1	0
0	1	2	3	3	3	2	1	1	1	1	1	1	0
0	1	2	2	2	2	2	1	0	0	0	0	0	0
0	1	1	1	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

逆ラスター変換後

骨格抽出(skeleton extraction)

- 距離変換画像で　　　　を持つ画素の集合を対象物体の　　　　　　　　　といい、この集合を取り出す処理を　　　　　　　　　　　　　　　という。
- p.40の”skelton”は”skeleton”の誤り(“l”の後に”e”が必要)。
- 距離変換画像 $d(i, j)$ に対して、次の条件を満足する画素が骨格を構成する。

$d(i, j) \geq$

(4近傍)

$d(i, j) \geq$

(8近傍)

- p.41図3.13に4近傍の処理結果がある。下記図で8近傍処理による骨格画素に〇を付けなさい。
- 注)　図3.13は4近傍による距離変換画像であり、8近傍処理の距離変換画像は異なる。

									1	1	
	1	1	1	1	1						
	1	2	2	2	2	1	1	1			
	1	2	3	3	3	2	2	2	1	1	
	1	2	3	3	2	1	1	2	2	1	
	1	2	2	2	1			1	1	1	
	1	1	1	1	1						

- 8近傍処理による骨格抽出(解答)

骨格処理の性質

- 処理の結果と 処理の結果は異なる。
- 一連結成分の骨格が
- 骨格画素と背景までの距離があれば、
  できる。
- 上記性質より に用いられる。
- 骨格は図形の を示すため、文字や図形の認識(特に)に有効である。

- p.41図3.13の結果から元画像を復元してみよう。

									①	①	
	①										
		②									
			③	③	③		②	②		①	
			③	③				②	②		
		②								①	
	①				①						

- 骨格からの復元(解答)

								①	①		
	①										
		②									
			③	③	③		②	②		①	
			③	③				②	②		
		②								①	
	①				①						

骨格画像

1回目の復元画像

2回目の復元画像

細線化(thinning)

- 図形の線幅を細めて幅1画素の中心線を抽出することを　　　　といい、この画像を　　　　　という。

(a)　　　　　　　:輪郭点を構成する画素で　　(1画素と連結している画素)以外の消去可能な点を消去することで輪郭部から　　　にある　　　を構成する画素を抽出する方法。(図3.14)

(b)　　　　　　　を用いる方法:図3.15のパターンを逐一当てはめ、　　　　　　(1から0に変更)することで細線化する方法。図3.15のパターンは、輪郭点(周辺3画素が0)でかつ、端点ではない点(少なくとも周辺2画素は1)を削除するという考えに基づいている。

- 上記マスクパターンを用いる方法では、細線化の結果は図形の中心になるとは限らない。この方法を改良したものに　　　　　　　　　　　　　　　　がある。
- 細線化の条件

(a)　　　　　　　　　な性質(　　　　　、　　など)は保存する。
(b)　　　　　　　とする。
(c)　　　　は削除しない。
(d) 細線化された線は　　　　　　　　にくる。
(e) 細線化された線上に　　　　　　　　。
(f) 元画像を　　　しても中心線の形態は　　　　　　。
(g) 図形の　　　　　で図形は　　　　　　。

輪郭線抽出(border following)

- 図形の輪郭画素を抽出する処理を　　　　　　と
  いい、対象図形を1とすると、手法は次のとおり。

① 画像を　　　　　　して、　　　　　の変化点(対象
  図形の開始点)を検出する。

② 開始点の　　　　　　の値を調べ、新たな変化点
  を求める。この走査を　　　　　という。

注) ここで、背景を0、対象図形を1とした場合、変化
  点は0から1へ変化する点となるが、背景を1 、対
  象図形を0とした場合は1から0への変化点となる。

  また、周囲8画素の検索は　　　　　と
  があり、検索方向と対象図形の追跡方向は一致
  する。

③ 周囲8画素が全て0であれば、　　　　である。

④ 中心画素に　　　　　　　　　を付け、方形走査で検出した　　　　　　　　　する。

⑤ ②に戻り、開始点を新たな変化として方形走査を、　　　　　　　　　が発見されるまで繰り返す。

⑥ 　　　　　　　の付いた画素が　　　　　　となる。

⑦ 上記手法を全ての図形に対する輪郭が検出されるまで繰り返す。このとき、同じ図形の輪郭を抽出することのないように、0から　　　　　　　　へ変化する画素は開始点とはしない。

輪郭抽出の特徴

(a) 方形走査を　　　　に行うと、図形の輪郭も　　　　に追跡される。方形走査を　　　　に行うと、図形の輪郭も　　　　に追跡される。

(b) 孔の輪郭は方形走査とは　　　　に追跡される。つまり、方形走査を　　　　に行うと、孔の輪郭は　　　　に、方形走査を　　　　に行うと孔の輪郭は　　　　に追跡される。

(c) 輪郭線の総数は、図形の輪郭数＋孔の輪郭数である。

注)　　　　と　　　　はどちらも使用される。画像処理では処理の元になる画像という意味で　　　　を用いることが多い。　　　　は写真や絵画で　　　　として使用されることが多い。

線分(line segment)の検出

- 図形から を抽出する処理を という。
- 図3.18に示す3x3のマスクパターンで次の値を調べる。

- $E_i \ (i=1,2,3,4)$ の中で を与える $B_1 \ B_2 B_3$ の方向に が存在する。
- 但し、$E_i \ (i=1,2,3,4)$ にあまり差がない場合、 は存在しない。

3.4 線図形の形状分析と符号化

- 線図形の特徴点には次のものがある。

① :連結数1の画素。但し、線図形を対象としているため、 画素となる。

② :連結数2の画素 と連結。

③ :連結数3の画素。 と連結。

④ :連結数4の画素。 と連結。

(図3.8参照) 上記は線図形を対象としているため、連結数0の はない。

- 画像処理により抽出された線図形にはノイズが含まれるため、次の処理により整形を行う。

(a) :分岐点から着目点(端点)までの髭部分を除去する。

(b) :端点から着目点(近隣にある端点)までの 、かつ (線分の方向と端点を結ぶ方向がほぼ一致する)場合、端点を結ぶ。

(c) :線分の方向に対する をなくして、線分を にする。

線図形の認識

- 線図形がどのような関数で表現できるのかを調べることを　　　　　　　　　という。
- 線図形を表現する関数を求めるには次の方法を用いる。

(a)　　　　　　　　　を用いる方法: 与えられた線分点列に対して直線や円(範囲を限定すると円弧)の方程式を想定し、　　　　　　　　　により関数の係数を決定する方法。

図3.19参照。但し、図3.19は適用する点列の範囲を決定する式 $|f(x,y)|<C$ である。

線図形の認識

- 最小二乗法を用いて以下のように係数を決定する。

直線の場合

与えられた点列 $(x_i, y_i)(i = 0,1,\ldots,n-1)$に対して

最も近似する直線 $y = ax + b$を求めるためには、

を最小とする aとbを 及び より求める。

円の場合

を最小とする aとbを , 及び

より求める。

(a) を用いる方法:

図3.20の直線 α を求める。$\overrightarrow{OB}$ と x 軸とのなす角度は θ であるから、$\overrightarrow{OB}$ の単位ベクトルは となる。

また、直線 α の原点Oからの距離は ρ であるから、点 B の座標は となる。直線 α 上の任意の点を (x, y) とすると、直線 α 上の任意のベクトルは となる。

一方、直線 α と $\overrightarrow{OB}$ は直交するから、ベクトルの内積は 0 となる。従って、

$\therefore$ p.47の(3.19)

となる。

つまり、直線 αは と記述できる。

ここで、θとρを固定すると、直線 αはXY平面上で をパラメータとする直 線となる。一方、(x, y)を固定して をパラメータとして変 更すると、XY平面上では を通る様々な直線 (原点からの距離 ρ 及び直線への垂線と x軸とのなす角度 θが異なる直線) を表す。つまり、$\theta\rho$平面上では を通る曲線となる。

ここで、$\rho = x\cos\theta + y\sin\theta =$ と変形できる。

但し、

つまり、$\rho = x\cos\theta + y\sin\theta$は$\theta\rho$平面上では となる。

複数の点に対する直線 $\rho = x\cos\theta + y\sin\theta$を考え、
これを $\theta\rho$平面上に描画すると、　もし、複数の点
を通る直線が同一直線　であれば、
の　　　　　　　　を取る。つま　り、
$\theta\rho$平面上の複数の曲線は　　　　　　　　。(p.47図3.20)
交点における　　　　　　　　ほど、同一直線　を
構成する　　　　　　　　　　ことを　示し、
なる。

従って、次のようにすれば、ハフ変換を用いて
できる。

与えられた点列 $(x_i, y_i)(i = 0,1,\ldots,n-1)$に対して
を$\theta\rho$平面の正弦曲線に変換
する。全ての点列 $(x_i, y_i)(i = 0,1,\ldots,n-1)$に対して
$\theta\rho$平面の正弦曲線を描画し、
に集中していれば、
という直線が画像中に存在していることが判る。

線図形の符号化

- 線図形を符号で表現することを
といい、次の方法がある。

(a) : が考案した方法で、線分の方向を図3.21(a)に示す で符号化する方法。符号化には を用いる。図3.21(b)に例がある。

(b) :チェイン符号化では符号の割り当てに3ビット用いるが、符号の変化は±1が多い、そこで、符号の差分を計算し、 に を割り当てることで を小さくする方法(これを、 という)である。(図3.22)

(c) :線分の方向を し、
を符号化する方法。アルゴリズムは次のとおり。

① 現在の着目点$p_{i-1}(x_{i-1}, y_{i-1})$ を原点として θ
と k の分割を行う。

② 次の点$p_i'(x_i', y_i')$ を となるように
する。ここで、$p_i'(x_i', y_i')$ の象限とゾーン番号
を$\theta(p_i'), k(p_i')$ とする。

③ 実測点 $p_i(x_i, y_i)$ の象限とゾーン番号が
$\theta(p_i), k(p_i)$ であれば、その

, を
する。(図3.23)

方向差分チェイン符号化とゾーン符号化は1988年に
(Comite Consultatif International Telegraphique et
Telephonique:)で標準勧告された

3.5 団塊図形に対する特徴の抽出

- 広がりを持った図形、あるいは、ある程度固まりを持った図形を　　　　　　といい、次のようにして抽出することができる。

(a)　　　　　　　　　　　　　:同一連結成分に同一番号を割り当てる処理。次の手順で行う。

① 画像を　　　　　　　し、ラベルが割り当てられていない　　　　　　を探す。

② 見つかった　　　　　　　に　　　　　　　を割り当てる。

③ 全ての連結成分にラベルが割り当てられるまで上記処理を繰り返す。

ラベリングにより、　　　　　　　　　を計測できる。

(図3.24参照)

(b)　　　　　　　　　　　: 連結成分の面積を小さくし、
　　　にする処理を　　　　　　という。また、1点となった画素を　　　　　　という。　　　処理結果に対して、　　　　　を行えばよい。処理形態には次の種類がある。

①　　　　　　　　　　: 孔を含まない連結成分(　　　　　　)に対してのみ縮退処理を施す。

②　　　　　　　　　　: 孔を含む連結成分(　　　　　　)に対しても縮退処理を施す。単連結成分も縮退する。

縮退処理の結果、画像中の連結成分を簡単にできる。(p.52 1行目の縮対は縮退の誤り)

(c) 画像中にある連結成分の形状を表す特徴量を
といい、次のものがある。

① :連結成分を構成する画素数。

② :連結成分を構成する の数。
厳密には の総和。このため、
斜め方向の画素間では をかけて補正を行う。

③ :連結成分がどれだけ円に近いかを測る
尺度。次式で表され、真円で を取る。

p.52 (3.22)のxは不要 $or \times$(乗算記号)の誤り。

④ :連結成分を構成する輪郭画素間距離の最大値。

⑤ :絶対最大長に直交する直線に沿った連結成分の最大長さ。

⑥ :水平線(画像の横軸 or x 軸)と絶対最大長の方向とのなす角度。

⑦ :連結成分と面積の等しい円の直径。

(d) :連結成分の概略形状を表す性質の一つ。連結成分をにより分類すると、次のようになる。

① :連結成分の任意の2点を結ぶ線分が全て連結成分のから構成される図形。

② :凸ではない図形。

③ :連結成分を囲む図形。ともいう。

(e) :原点を中心とした

と重心回りの (

あるいは、)がある。

$(p+q)$次のモーメント：$m(p,q)=$

重心モーメント：$M(p,q)=$

但し、$f(i,j)$ $(i,j=0,1,\ldots..,n-1)$は画素(i,j)の値、

(i_g, j_g)は連結成分の重心

連結成分の (連結成分が伸びている方向)と

i軸とのなす角θは次式で与えられる。

$\theta=$

0次モーメント: $m(0,0) = M(0,0) =$

画像$f(i,j)$の

1次モーメント: $m(1,0)$ or $m(0,1)$

$$\frac{m(1,0)}{m(0,0)} = \qquad = i_g , \frac{m(0,1)}{m(0,0)} = \qquad = j_g$$

は連結成分の重心

2次モーメント( )

$M(2,0) + M(0,2) =$

$=$

(d) :輪郭線を表現する を し、展開した で連結成分の特徴を表す方法。

① :ZahnとRoskiesにより発見された方法で、 を用いる。図形の 、 、 、 に対して不変な量。

② : Granlundにより発見された方法で、 による を用いる。

③ : を意味するPhaseから名づけられた方法で、 を用いる。図形の 、 、 、 に対して不変な量となる。また、 にも適用可能。

Z型フーリエ記述子

p.54図3.28(a)に示す点Aから輪郭線上でl離れた距離にある点における輪郭線の接線の偏角を$\theta(l)$とすると、輪郭線の長さがLのとき、

$\theta(l)$は　　　　　となる。つまり、　　　　　　　　　となる。

しかしながら、$l=0$を代入すると　　　　　　　　となり、輪郭線を一周

すると、元の値に戻らない。そこで、

を導入する。すると、

$\theta_n(L)=$　　　　　　　　　　　　となり、輪郭線を一周すると

偏角は元に戻る。この　　　　　　　　$\theta_n(l)$を　　　　　　　し、その

係数を特徴量とする。

G型フーリエ記述子

始点から輪郭線上でl離れた距離にある点を$z(l)=(x(l),y(l))$とする。この点を　　　　で考えると、　　　　　　となるので、この　　　　を

し、その係数を特徴量とする。