画像処理

２値画像処理

宮崎大学 工学部 情報システム工学科

第 ９ 回

本日の内容

2値画像処理

- 諸定義
  近傍，連結性，連結成分，
  オイラー数，連結数，消去可能
- 処理
  ラベリング，境界追跡
  距離変換
  収縮膨脹，細線化

２値画像処理 (P.137)

二値画像

白／黒 の２つの値で表現された画像

白: 0-画素，黒: 1-画素

利用場面

- 文書，図形画像の処理
- 物体とそれ以外の背景への分離

※単純である分，理論的な体系化がしやすい

近傍 (P.143)

画素 (i, j) の近くにある画素

4-近傍: 画素の上下左右の４点

8-近傍: 4-近傍に斜めの４点を加えた８点

x_4 $(i-1, j-1)$	x_3 $(i-1, j)$	x_2 $(i-1, j+1)$
x_5 $(i, j-1)$	x_0 (i, j)	x_1 $(i, j+1)$
x_6 $(i+1, j-1)$	x_7 $(i+1, j)$	x_8 $(i+1, j+1)$

連結性

隣接: 画素同士が近傍内にある

連結: 画素 a〜画素 b 間で，同じ画素値をもつ隣接した画素からなる経路が存在する

連結成分: 互いに連結している画素の集合

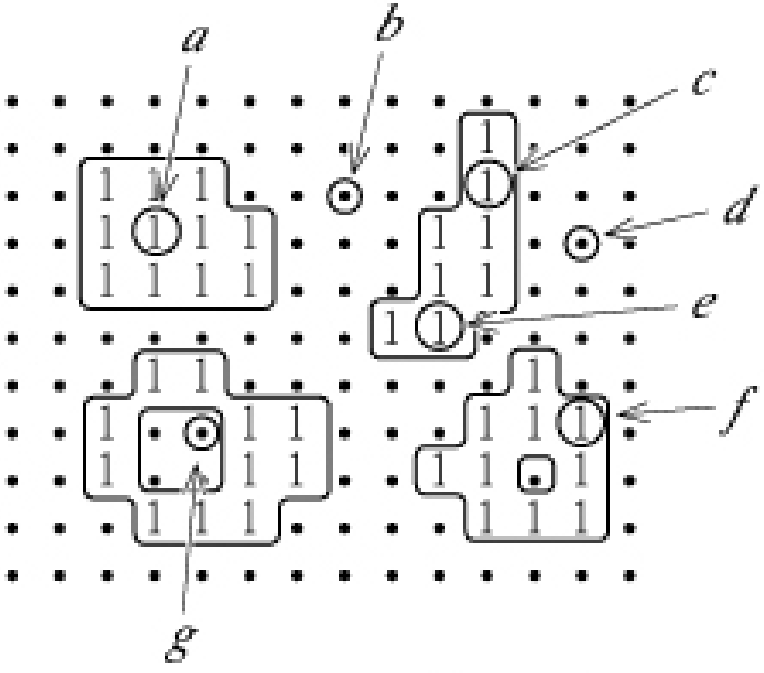
※ 4-近傍／ 8-近傍の場合が考えられる

4-隣接／ *8*-隣接，*4*-連結／ *8*-連結，

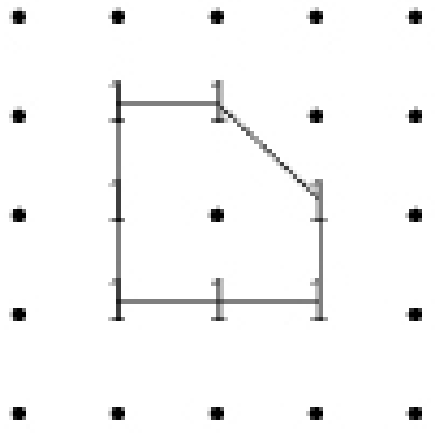
4-連結成分／ *8*-連結成分

連結成分の例

4-近傍を採用するか8-近傍を採用するかで連結成分は異なる



c と e, b と d は連結．他の同じ値をもつ画素の対は非連結



1-画素を8-連結と見るなら，0-画素は4-連結で考えなくてはならない．

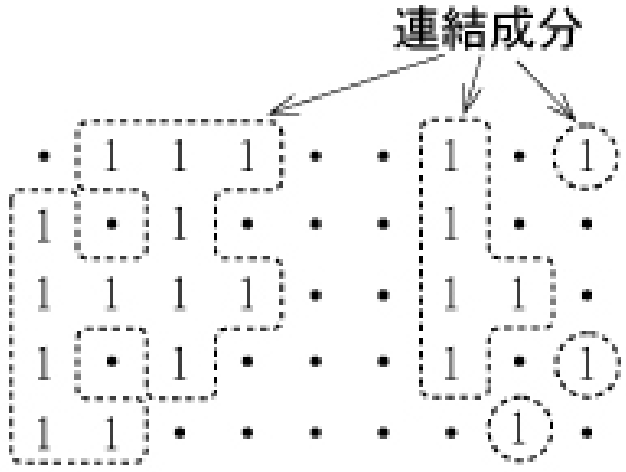
オイラー数 (示性数)

1-画素の連結成分の数から孔の数を引いたもの

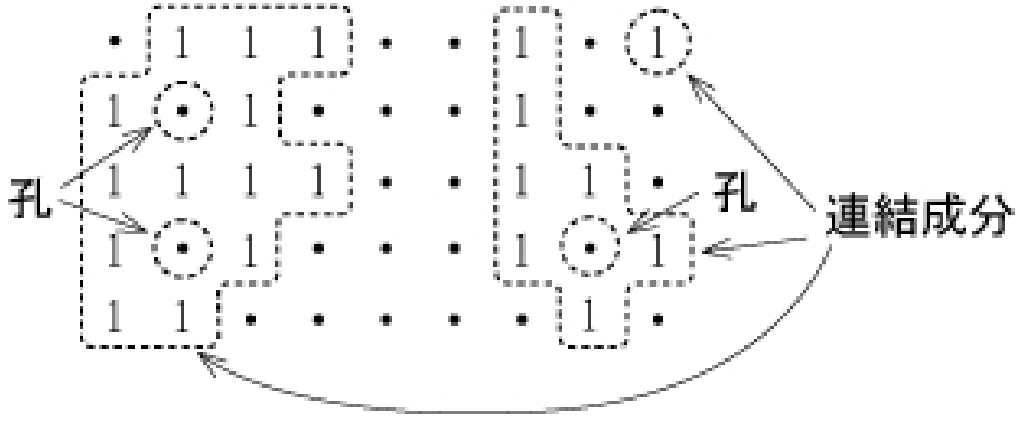
孔: 外周と連結していない 0-画素連結成分

単連結成分: 孔を含まない 1-画素連結成分

多重連結成分: 孔を含む 1-画素連結成分



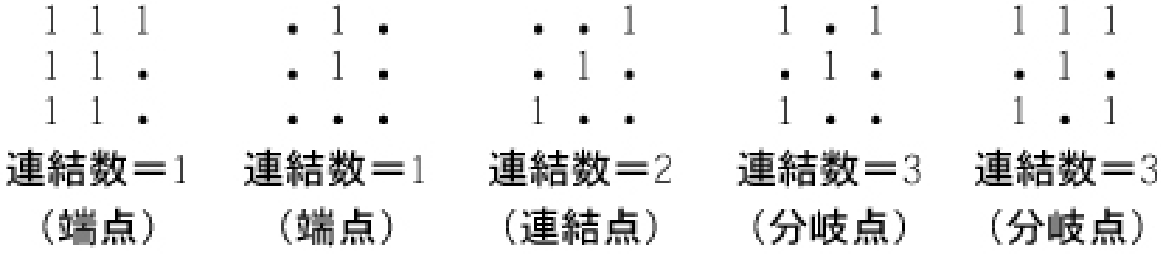
(a) 4-連結の場合
(成分数＝5, 孔の数＝0,
オイラー数＝5)



(b) 8-連結の場合
(成分数＝3, 孔の数＝3,
オイラー数＝0)

連結数

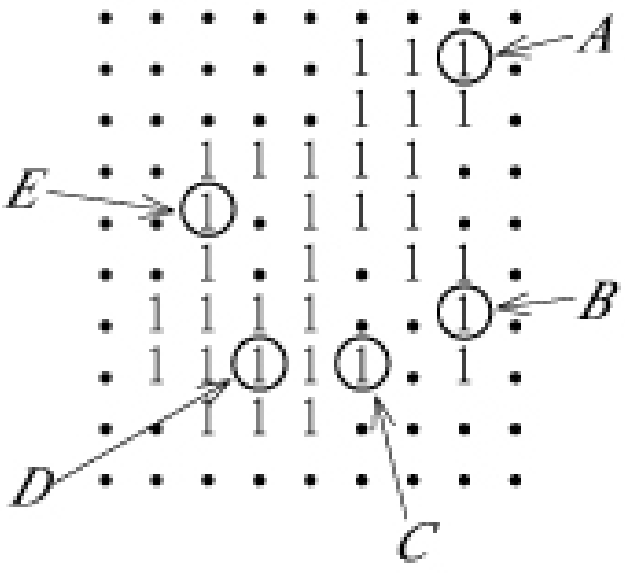
中央の 1-画素を 0-画素に変えた時のオイラー数の変化+1



連結数＝0：孤立点
または内部点
1：端点
2：連結点
3：分岐点
4：交差点

消去可能

- 連結数が 1 な画素
- その 1-画素の値を 0-画素に変えても，連結性（オイラー数）が変化しない



A と C は消去可能

B を消去すると図形が分離

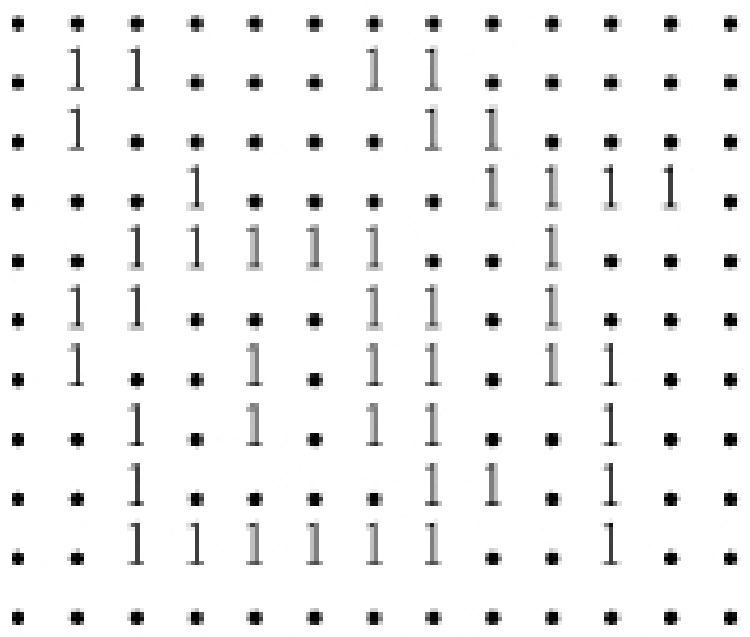
D を消去すると孔が生成

E を消去すると孔が消滅

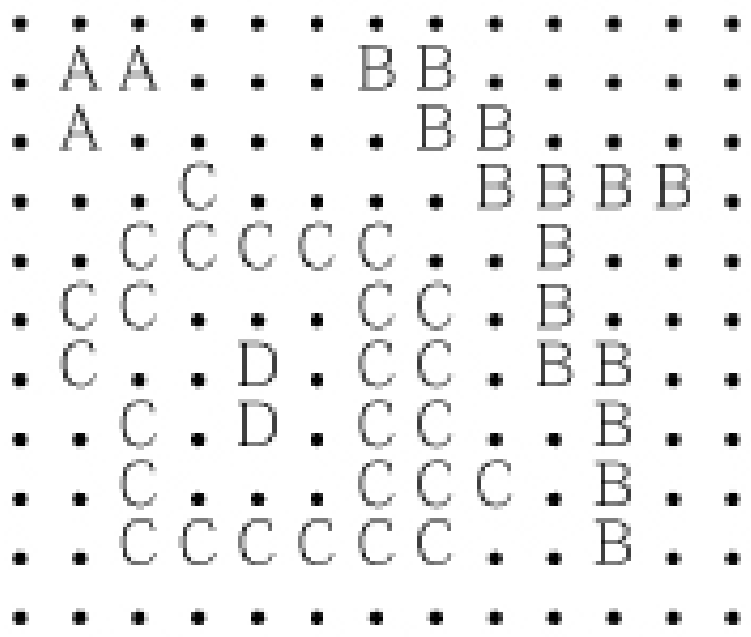
} 消去不可能

ラベリング

連結成分毎に同じラベル（番号）を割り当てる



(a)　入力画像



(b)　ラベリング結果

ラベリングの例

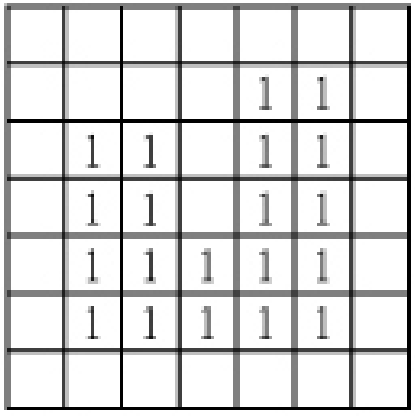
2 値画像

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

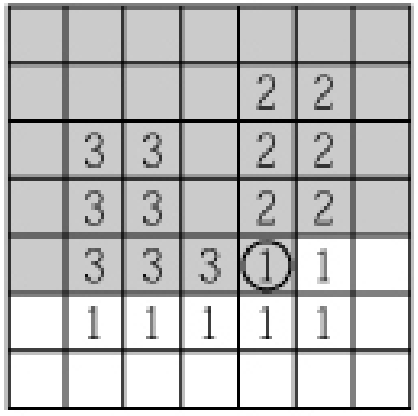
1-画素を 4-近傍で

1-画素を 8-近傍で

ラベリングアルゴリズム（近傍処理）

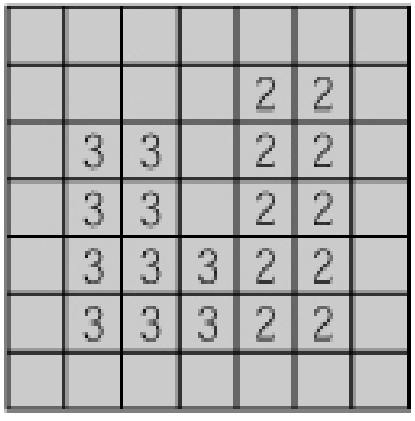


(a)



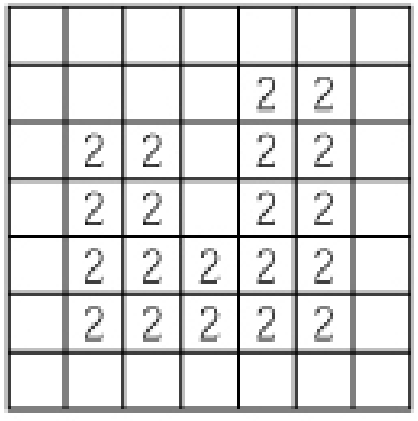
i	
1	1
2	2
3	3

(b)



i	LUT(i)
1	1
2	2
3	2

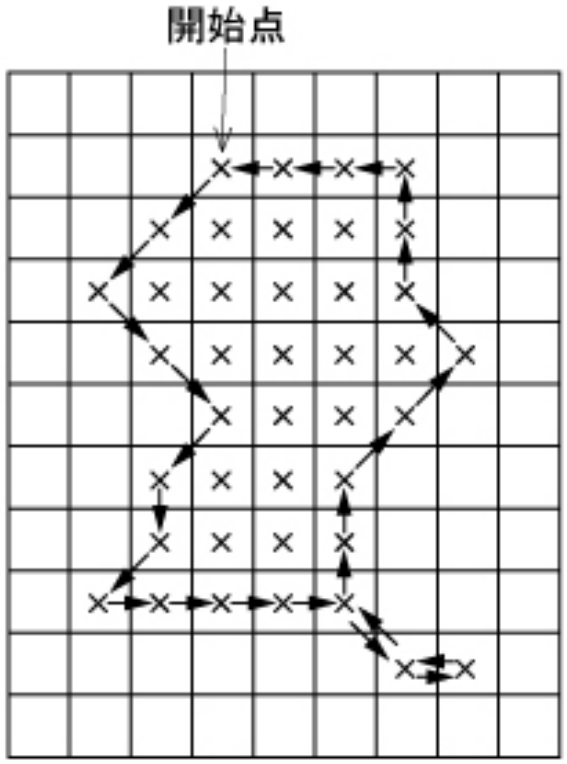
(c)



(d)

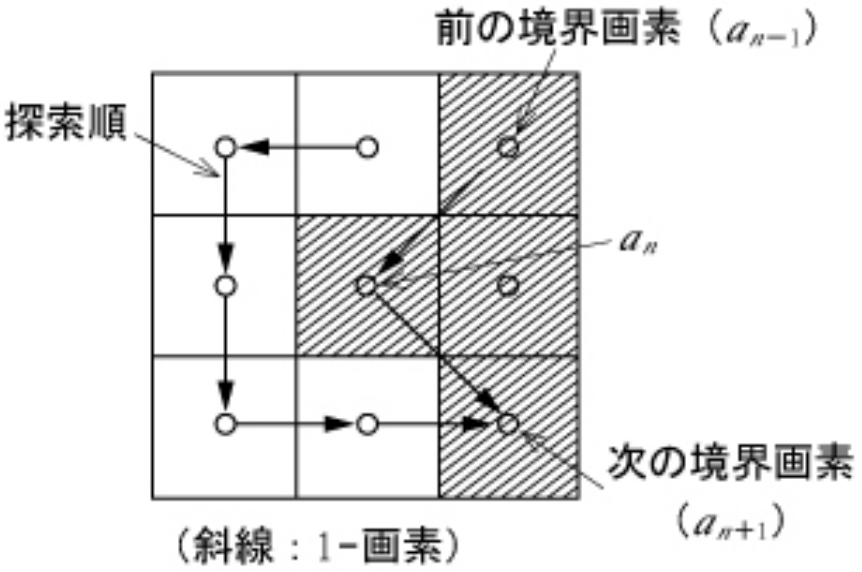
境界追跡

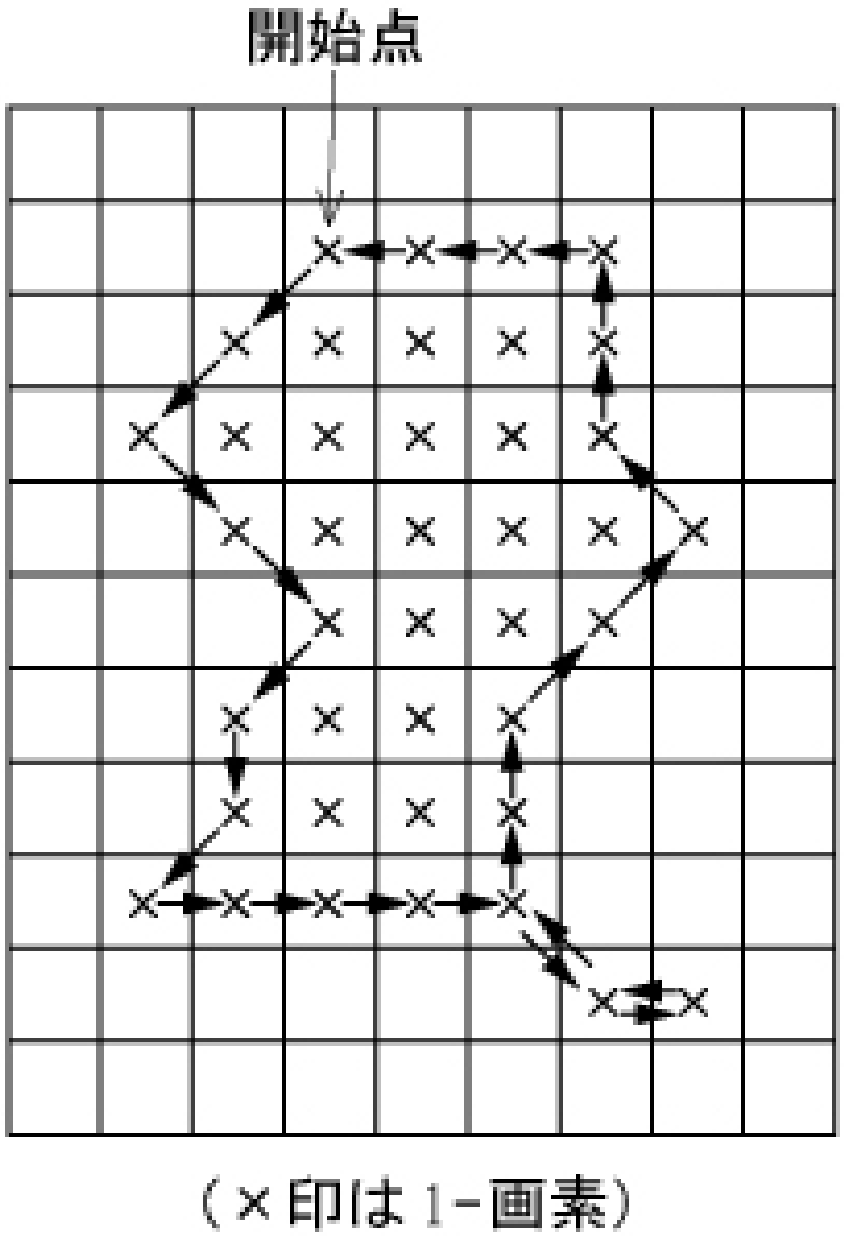
連結成分において 1-画素と 0-画素の境界部分を抜きだす



（×印は 1−画素）

境界追跡アルゴリズム

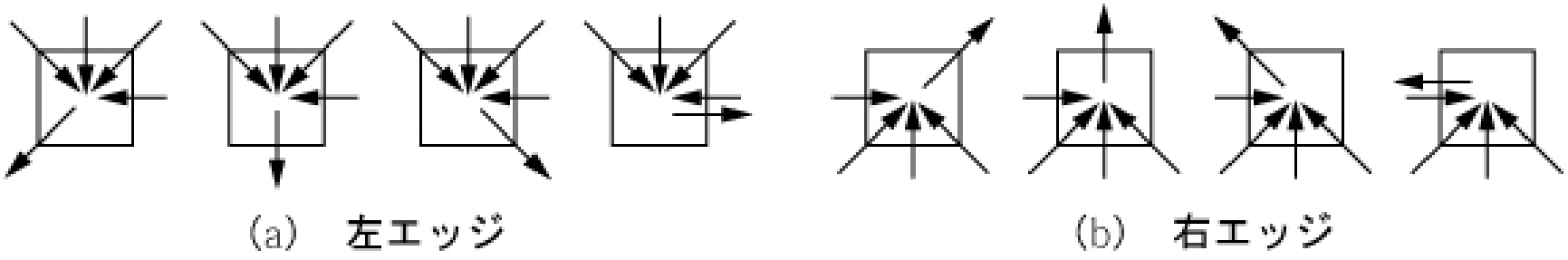




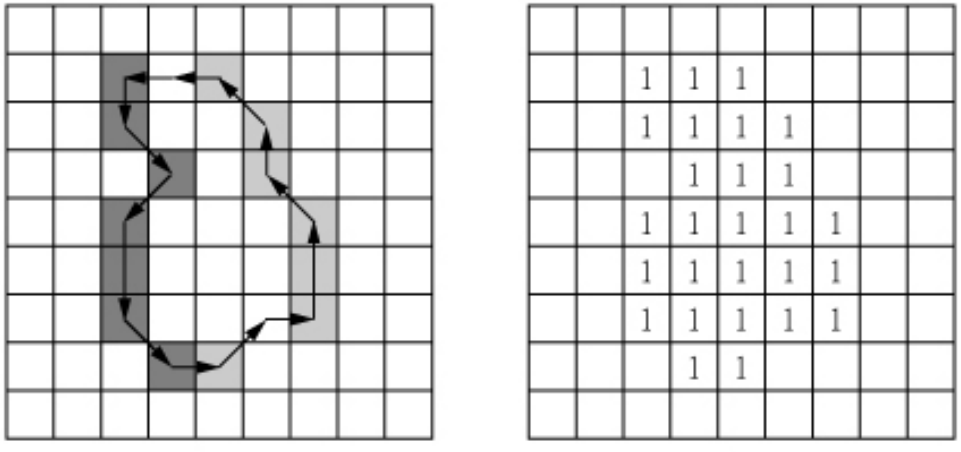
境界からの原画像の復元

境界から原画像が復元可能

1. 右エッジ・左エッジを抜きだす
2. 左エッジと右エッジの間を塗りつぶす



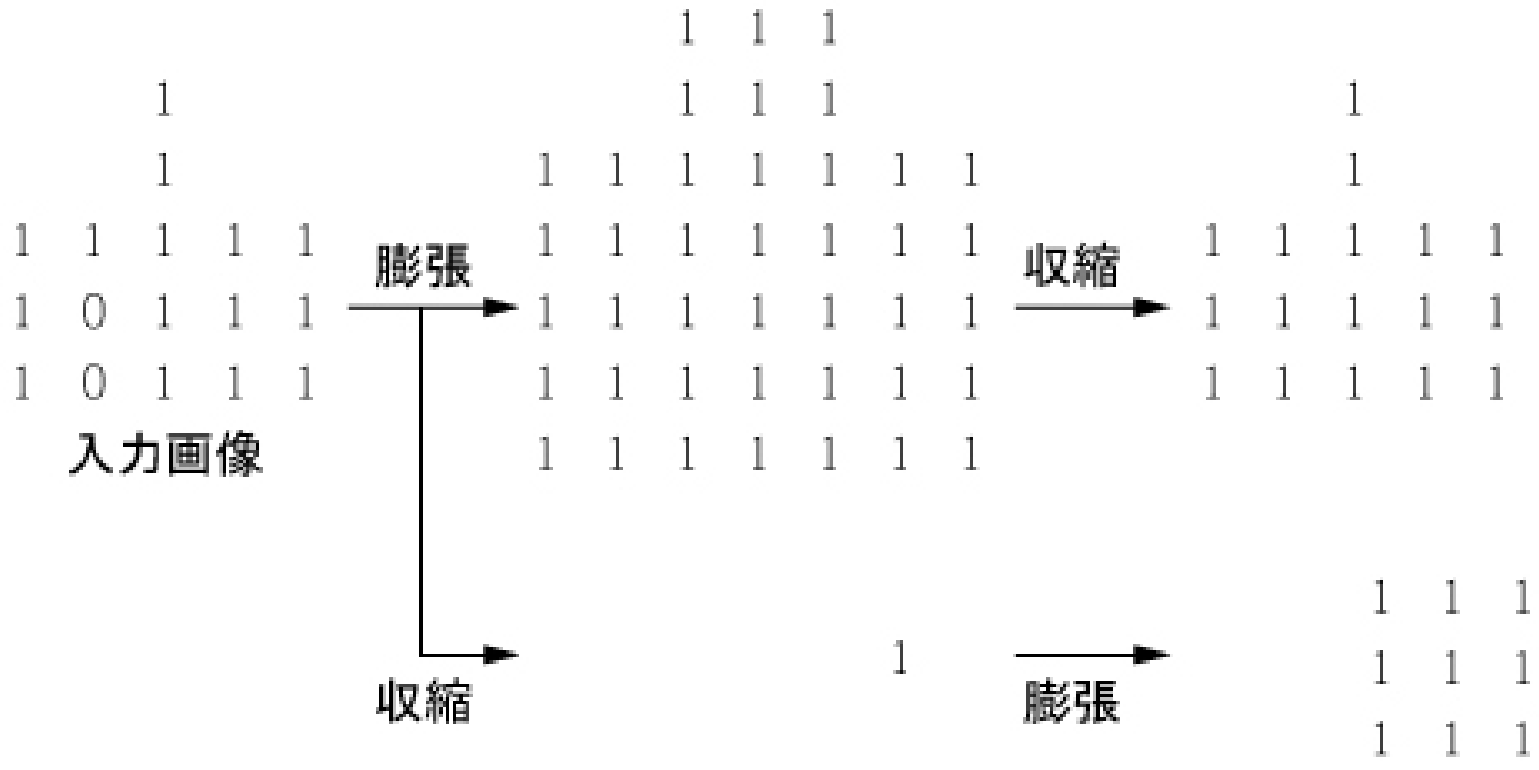
(a)　左エッジ　　(b)　右エッジ



(a)　　(b)

収縮膨脹

収縮 (Erosion): 境界画素を取り除き１画素小さくする
膨脹 (Dilation): 境界画素を追加し１画素大きくする
※小成分や小さな孔を検出／消滅させるために，組み合わせて使用される



収縮膨脹アルゴリズム

膨脹

$$g(i,j) = \begin{cases} 1 & : (i,j)\text{またはその近傍のいずれかが 1-画素} \\ 0 & : \text{その他} \end{cases}$$

収縮

$$g(i,j) = \begin{cases} 0 & : (i,j)\text{またはその近傍のいずれかが 0-画素} \\ 1 & : \text{その他} \end{cases}$$

膨張収縮の例

2 値画像

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

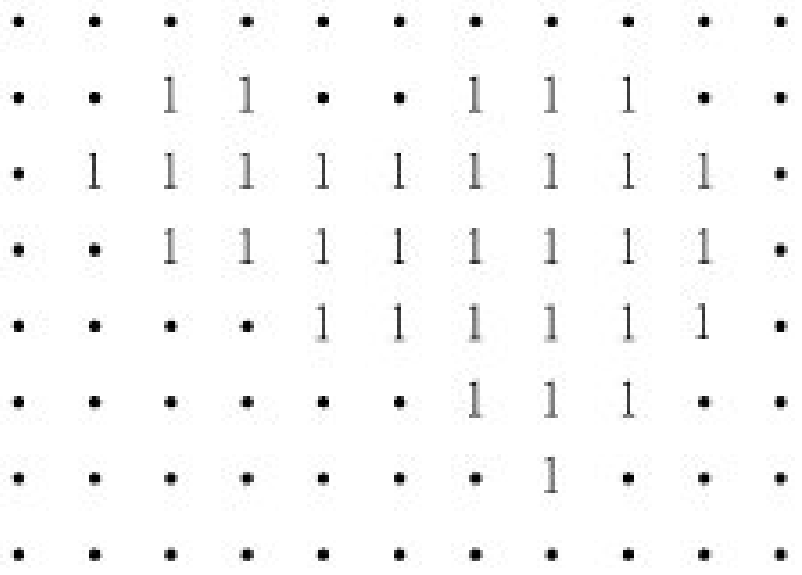
4-近傍で膨張⇒収縮：
クロージング

4-近傍で収縮⇒膨張：
オープニング

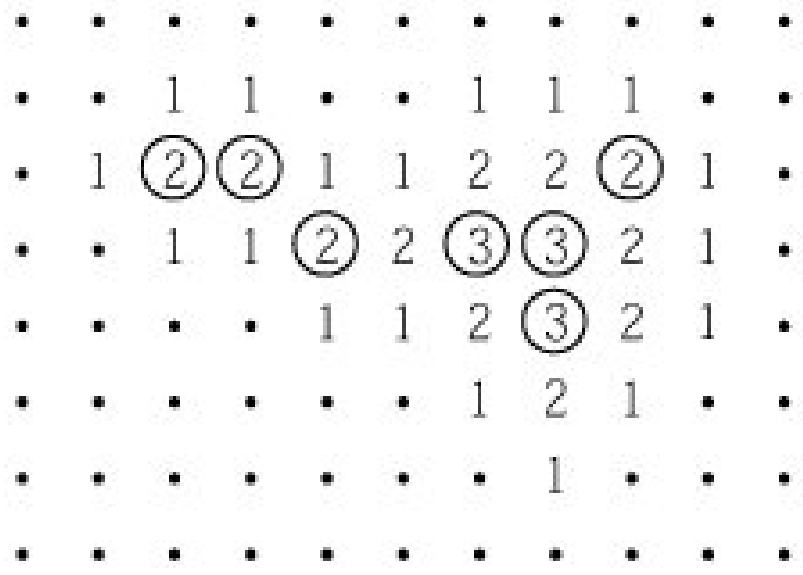
距離変換

各画素から 0-画素までの距離を新たな画素値とする画像への変換

骨格: 距離の極大値をもつ画素の集合



(a)　入力画像



(b)　距離変換（◯印）骨格

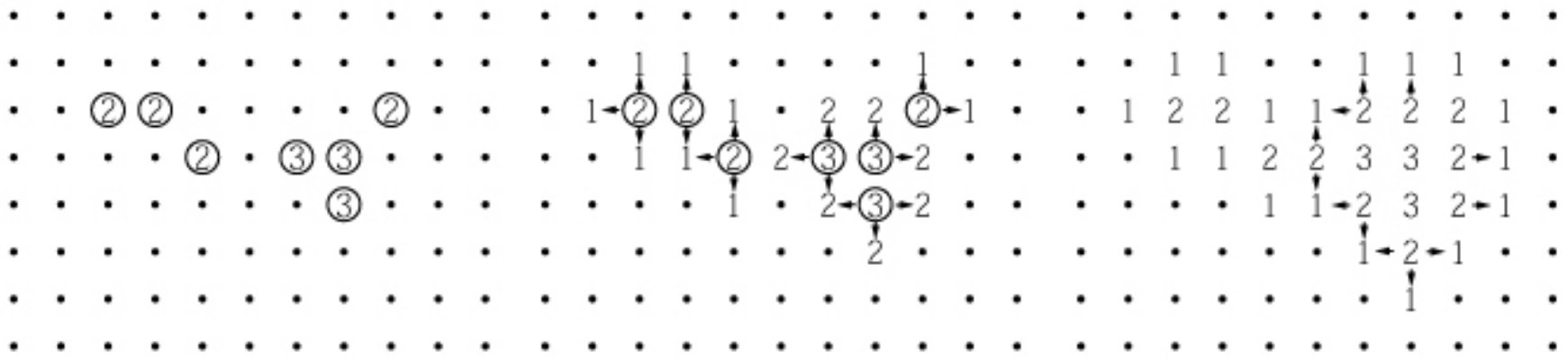
距離変換アルゴリズム(逐次型)

$$d_{ij} = \min\{|i-u| + |j-v|; f_{uv} = 0\}$$

1. 順走査により下記を求める
   $s_{ij} = \min(g_{ij}, s_{i-1,j} + 1, s_{i,j-1} + 1)$
   ただし $g_{ij} = \begin{cases} M & : f_{ij} = 1 \text{(M は十分大きな値)} \\ 0 & : f_{ij} = 0 \end{cases}$
2. 逆走査により下記を求める
   $h_{ij} = \min(s_{ij}, h_{i+1,j} + 1, h_{i,j+1} + 1)$

逆距離変換

骨格（＋その点の距離値）が与えられるともとの２値画像を復元可能



(a)　骨格　　(b)　距離値の 4−近傍への伝搬　　(c)　距離値の 4−近傍への伝搬（2 回目）

細線化

与えられた図形から 線幅 1 の中心線を抽出 (図 5.31)

望ましい性質

- 線の位置がもとの図形のほぼ中心
- 図形の連結性が保存
- 中心線が縮まない
- 境界の凹凸による「ひげ」が生じない
- 原図形を回転しても中心線の形状が不変
- 交差部で中心線が歪まない

細線化アルゴリズム

条件 1: 境界条件 4-近傍に 0-画素が１つ以上存在
条件 2: 連結性条件 4-連結数が 1
条件 3: 非端点条件 8-近傍に 1-画素が 3 つ以上存在

- 条件 1-3 を満たす 1-画素を 0-画素に変換
- 変換する画素がなくなるまで繰り返す