

# 画像処理 2値画像

藤田 一寿

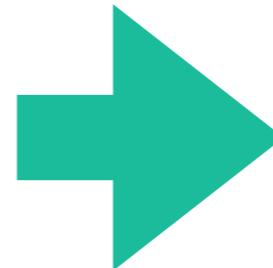
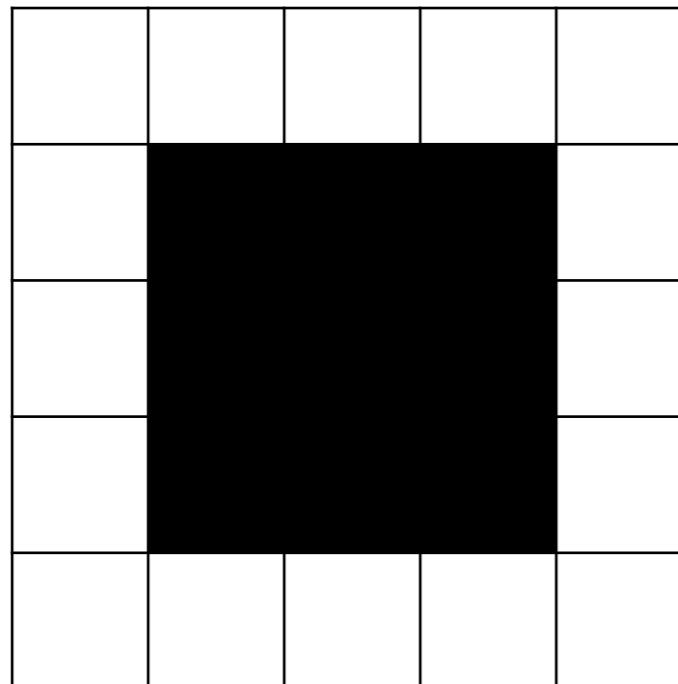
## ■ 2値画像とは

---

- ・白と黒の2色しか用いない画像
- ・2値画像もしくはバイナリ画像と呼ばれる
- ・画像識別で用いられることがある。とくに文字識別を行う場合によく用いられる。

## ■ 2値画像の例

- ・2値画像は白と黒の2色のみで構成される画像である.
- ・白を0, 黒を1で表した場合の例を示す.

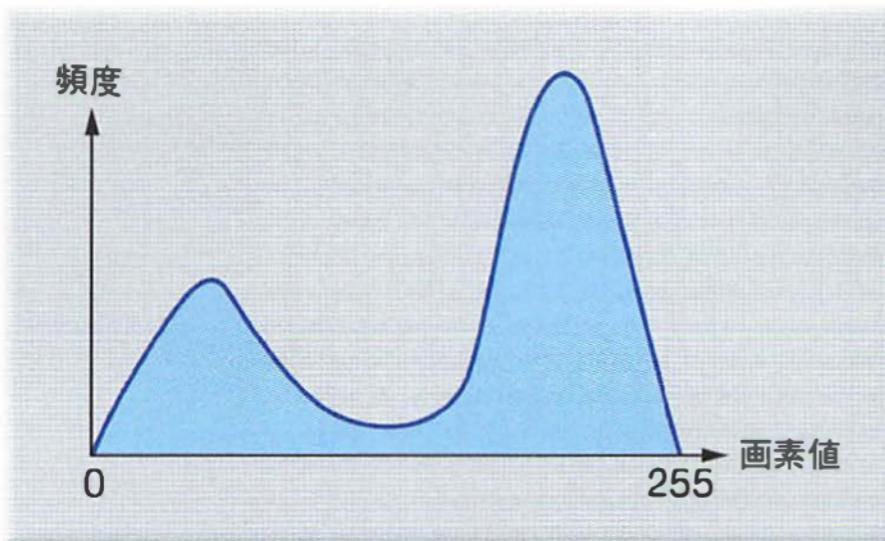


0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

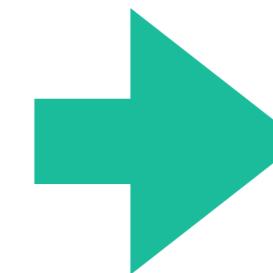
白と黒の値は0, 1でも1, 0でもよい.

# ■ 2値化とは

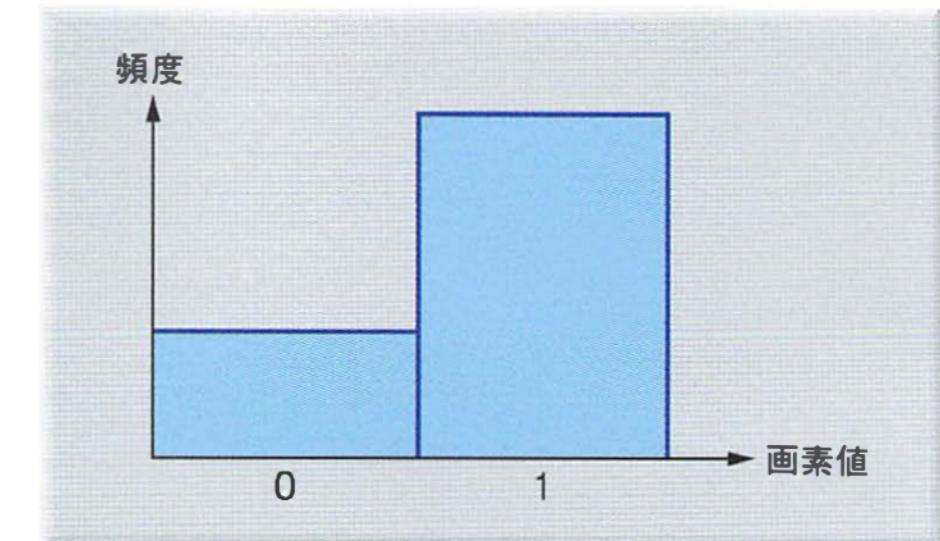
- ・画像を(2値画像, バイナリ画像)に変換すること



0から255の値



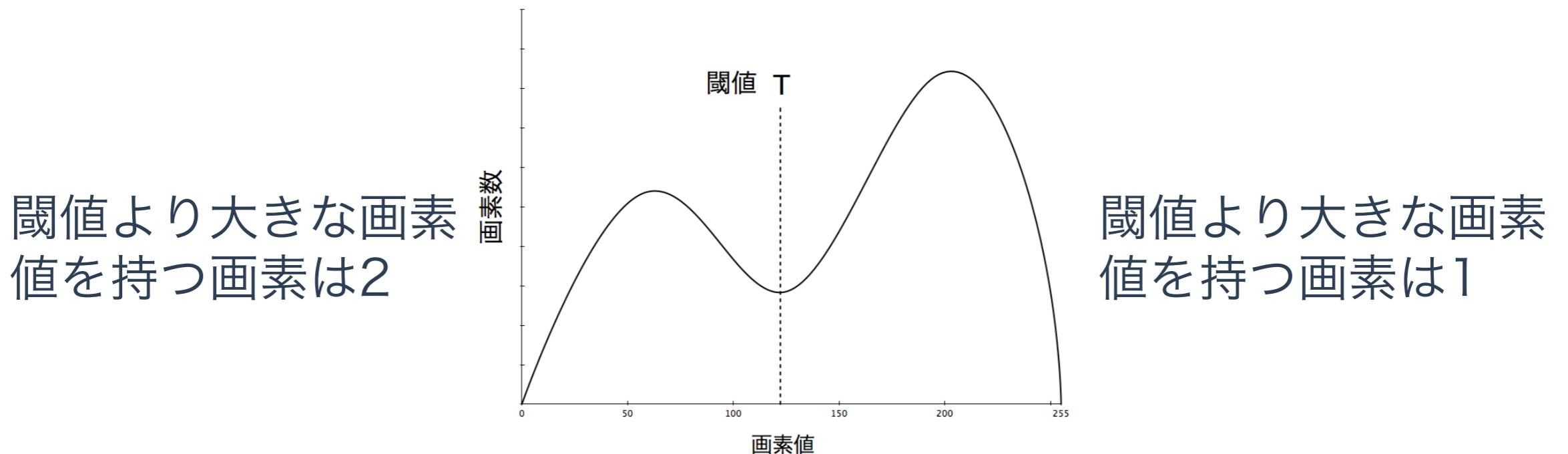
2値化



0か1の値

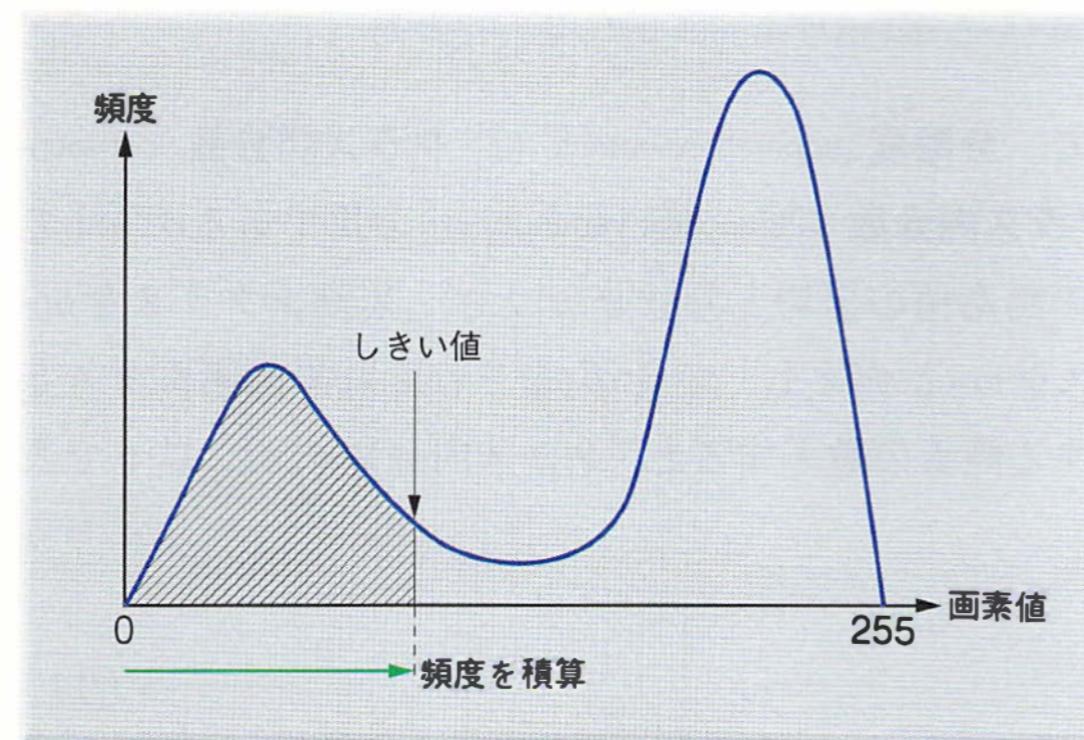
## ■ 2値化のやり方

- グレースケール画像の $(x, y)$ にある画素の画素値を $g(x, y)$ とすると、2値画像の $(x, y)$ にある画素の画素値を $b(x, y)$ は、
  - $b(x, y) = 1 \text{ if } g(x, y) \geq T$
  - $b(x, y) = 0 \text{ if } g(x, y) < T$
- と表される。 $T$ は閾値（いきち、慣用読みとして、しきいち、も可）である。



## p-タイル法

- ・2値化するためには閾値の値を決める必要がある。
- ・p-タイル法は、予め白と黒の領域の広さが予測できる場合に有効な閾値を決める方法である。
- ・p-タイル法では、低い画素値から画素の数を数え、ある値を超えたときの画素値もしくは超える直前の画素値を閾値とする。



■図9.3—p-タイル法によるしきい値の決め方

## ■ モード法

---

- ・ヒストグラムの谷の部分を閾値とする。
- ・ヒストグラムに山が明らかに2つあり、明確な谷が存在する場合には有効だが、それ以外の場合では良い方法とはいえない。

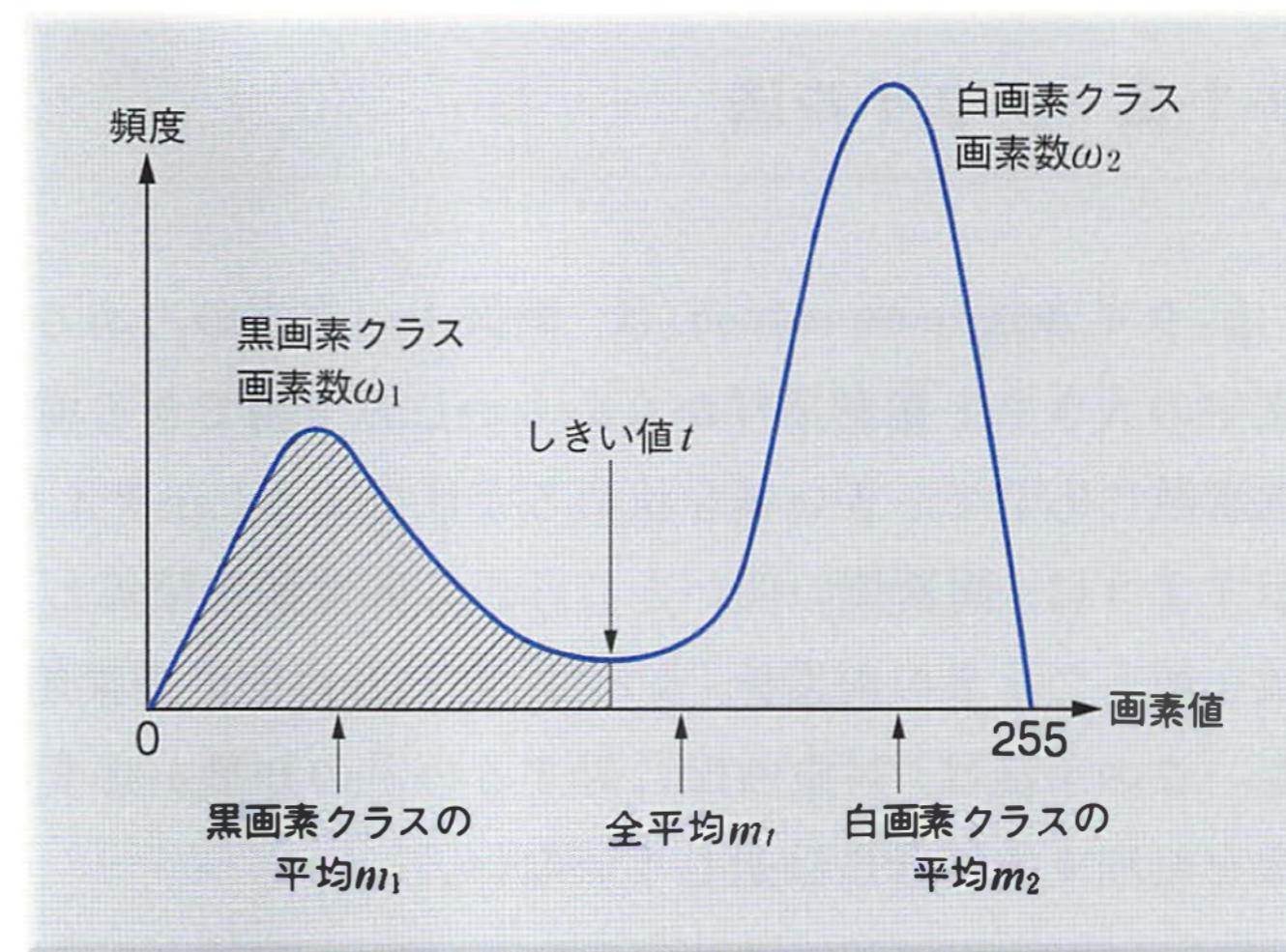
# ■ 判別分析法

- 黒画像の分布(クラス)と白画像の分布(クラス)の分離度を最大にするよう閾値を決める
- 分離度

クラス間分散

$$\frac{\sigma_b^2}{\sigma_w^2}$$

クラス内分散



■図9.4——判別分析法によるしきい値 $t$ の決め方

## ■ クラス内分散

$$\sigma_w^2 = \frac{\omega_1 \sigma_1^2 + \omega_2 \sigma_2^2}{\omega_1 + \omega_2}$$

$\omega_1$  黒画素クラスの画素数

$\sigma_1^2$  黒画素クラスの分散

$\omega_2$  白画素クラスの画素数

$\sigma_2^2$  白画素クラスの分散

## ■ クラス間分散

$$\begin{aligned}\sigma_b^2 &= \frac{\omega_1(m_1 - m_t)^2 + \omega_2(m_2 - m_t)^2}{\omega_1 + \omega_2} \\ &= \frac{\omega_1\omega_2(m_1 - m_2)^2}{(\omega_1 + \omega_2)}\end{aligned}$$

$m_1$  黒画素クラスの平均

$m_2$  白画素クラスの平均

$$m_t = \frac{\omega_1 m_1 + \omega_2 m_2}{\omega_1 + \omega_2}$$

# ■ 分離度

---

分離度

$$\frac{\sigma_b^2}{\sigma_w^2} = \frac{\sigma_b^2}{\sigma_t^2 - \sigma_b^2}$$

閾値Tに値を変えながら分離度を計算し、分離度が最大となるTを採用する。

## ■ 2値画像特有の処理

- 2値画像は2値しか取り扱わないため、グレースケール画像やカラー画像では行わない特有の処理がいくつかある。
- 連結性
- 輪郭追跡
- 収縮と膨張処理
- ラベリング
- 形状特徴パラメタ
- 距離
- 細線化

## ■ 連結性

---

- 2値画像は2値しか無いため、黒（白）のひとまとまりの領域を定義することが容易である。そこで連結という概念を用いる。

# ■ 近傍と連結

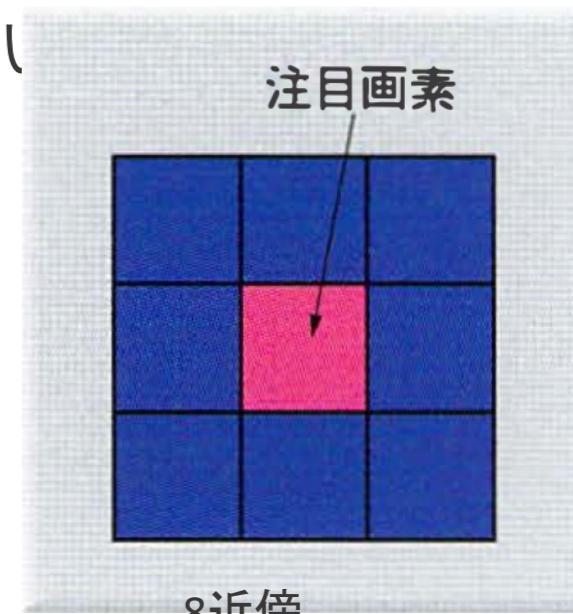
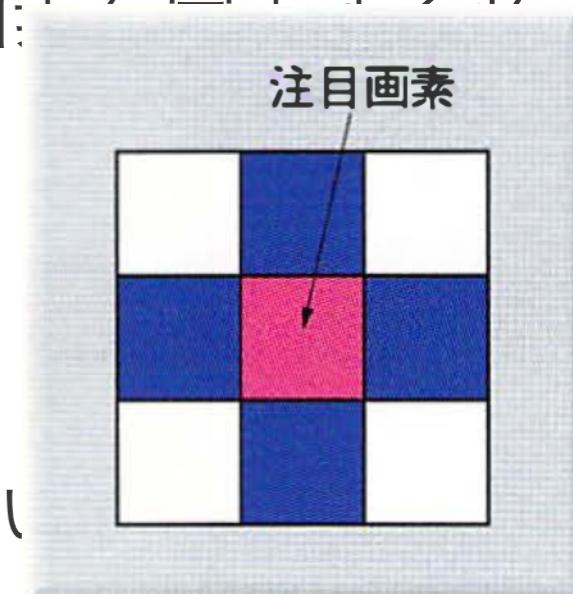
- デジタル画像の画素は格子状に並んでいる。黒い画素が繋がっている(連結)しているかどうかを決めるには、隣(近傍)の画素をどうする必要がある。

- 4近傍

- 注目画素に対し上下左右にある画素を隣とする。
- その近傍に対し連結していると定義したものを4連結とい

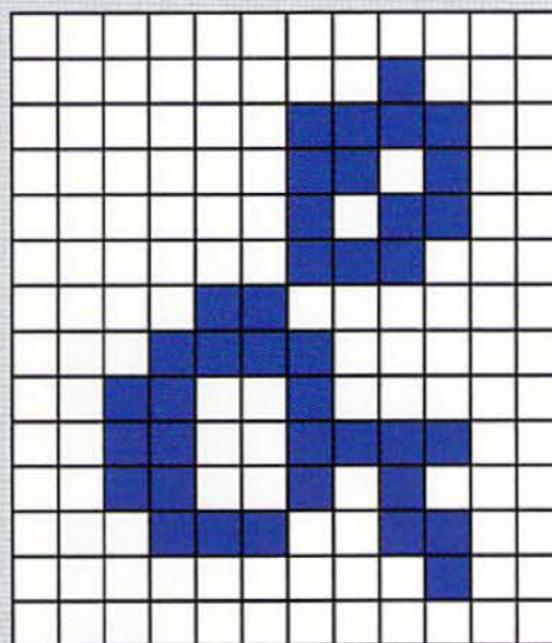
- 8近傍

- 4近傍に加え斜め方向にある画素も隣とする。
- その近傍に対し連結していると定義したものを8連結とい

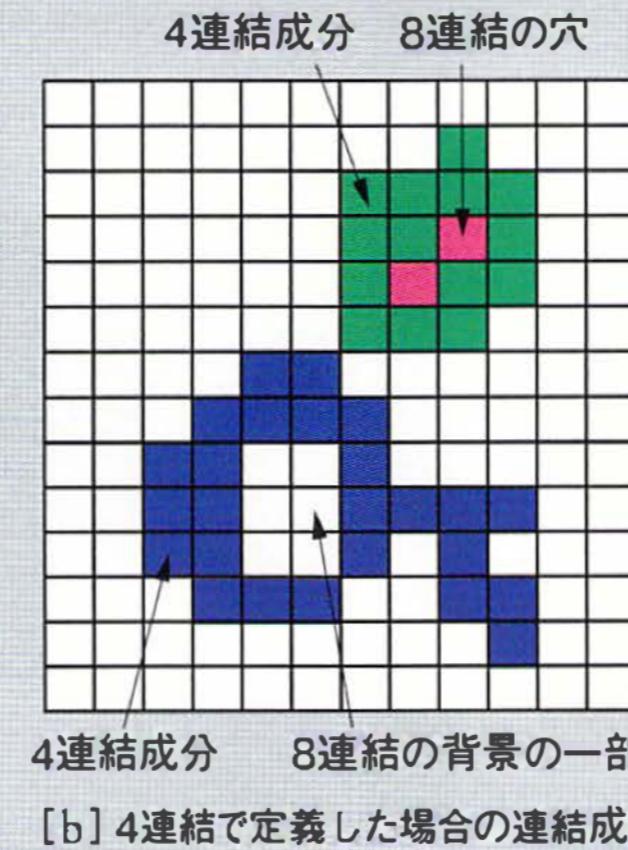


# ■ 連結成分

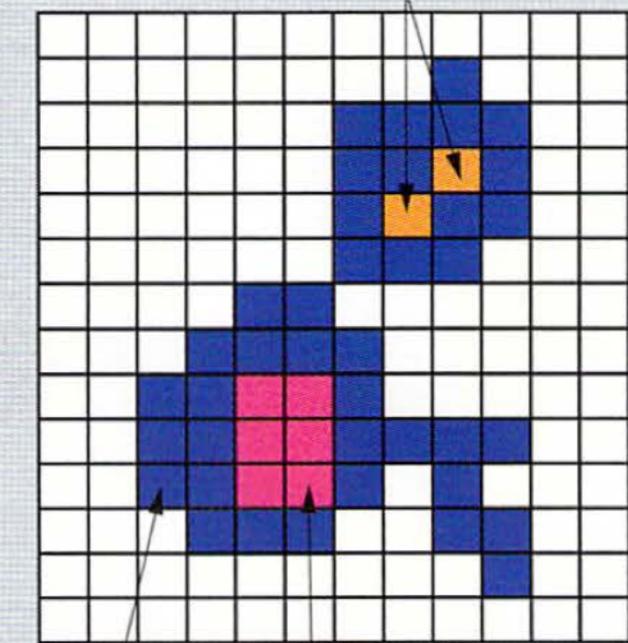
4連結で黒画素の連結性で定義する場合、背景と穴を8近傍で定義する。  
8連結で黒画素の連結性で定義する場合、背景と穴を4近傍で定義する。



[a] 2値画像の例



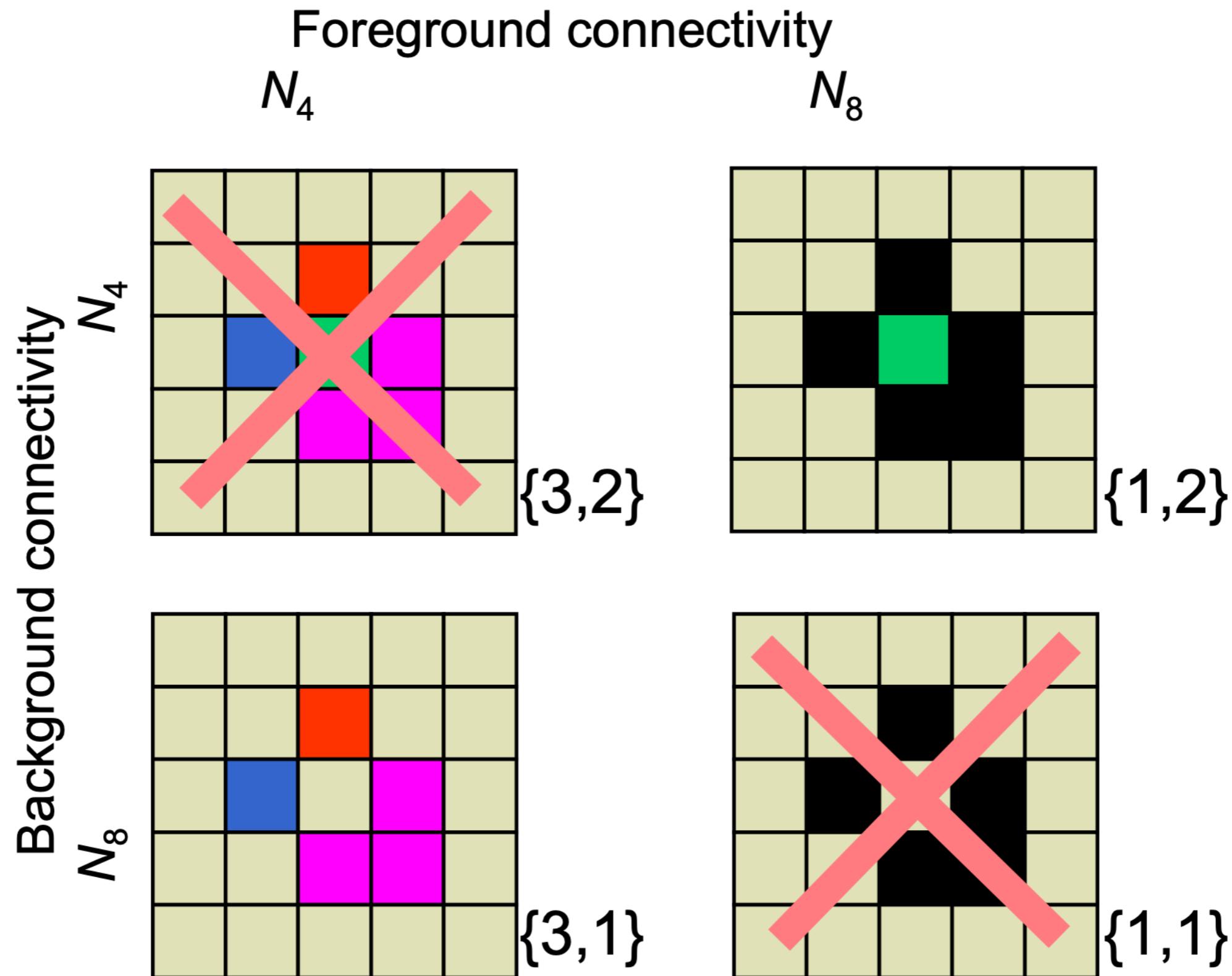
[b] 4連結で定義した場合の連結成分



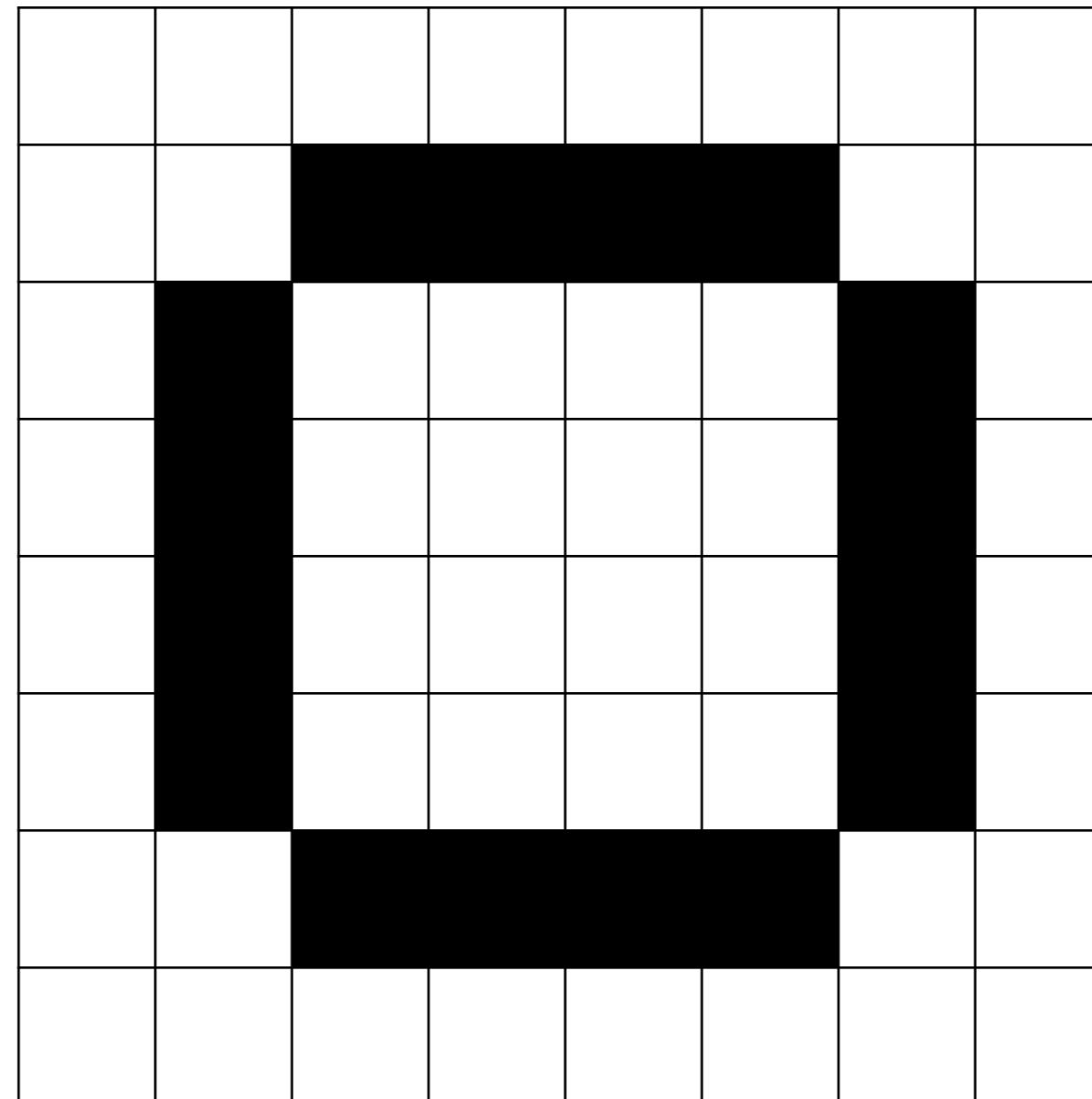
[c] 8連結で定義した場合の連結成分

連結成分の内部にあり、背景と連結していない白画素のクラスタを穴と呼ぶ

# 連結成分



# 問題



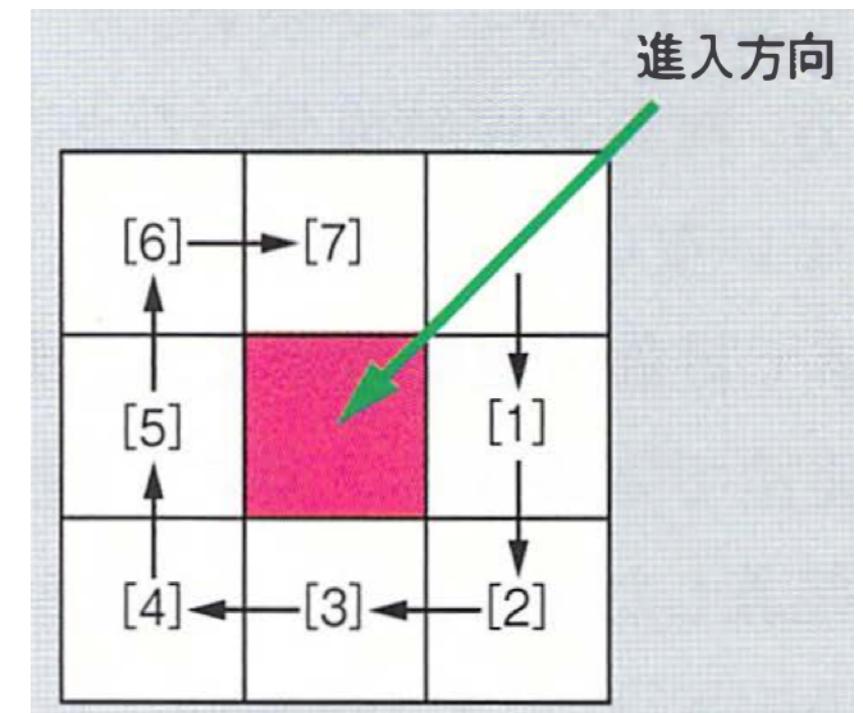
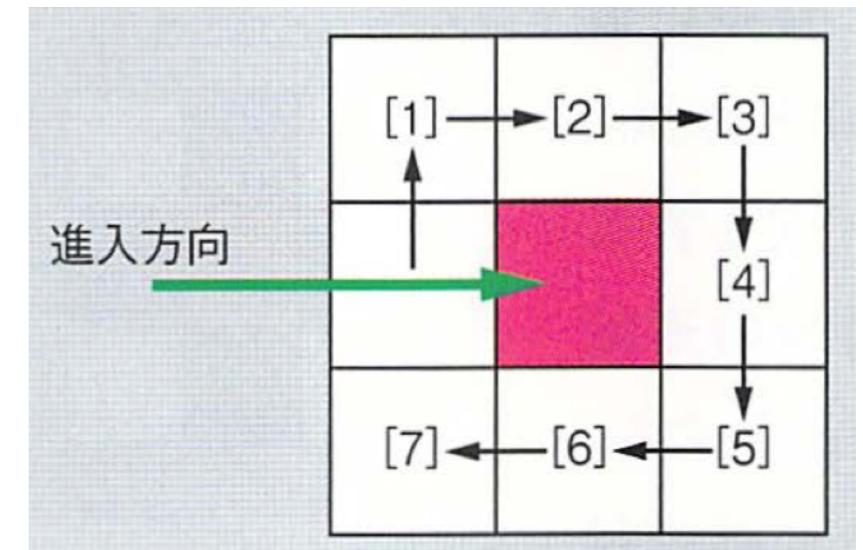
4連結で黒の連結性を定義した場合：連結成分 、 穴

8連結で黒の連結性を定義した場合：連結成分 、 穴

# 輪郭追跡

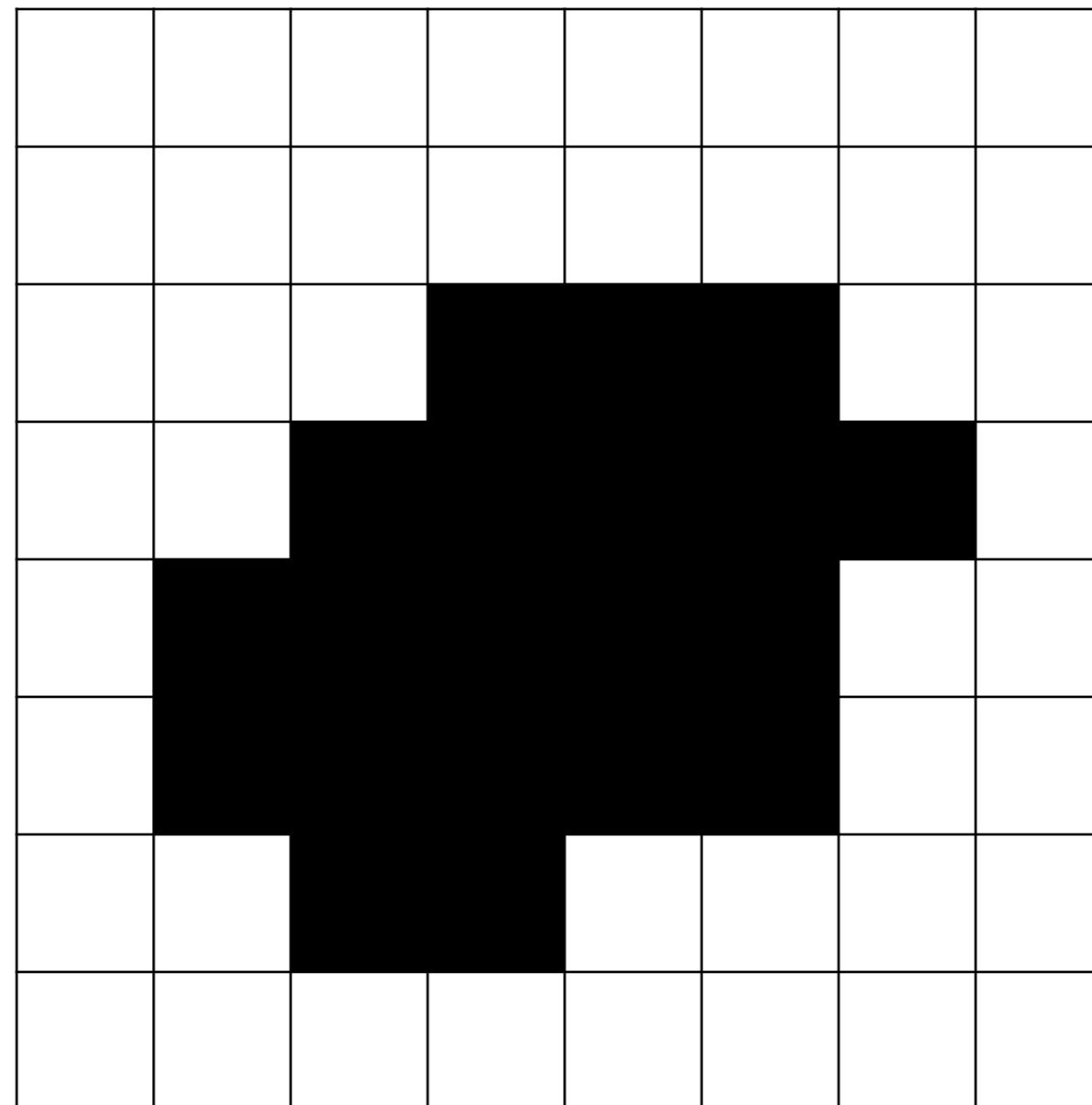
- 連結成分の境界を求めることを、輪郭追跡と呼ぶ。
- 処理手順（8連結の場合）

- ラスタスキャンによって白画素から黒画素に変わった画素を探索する。ラスタスキャンとは、画像の左上を起点に左端から右へ画素を調べ、右端に着いたら行を一つ下がり左端から右へ画素を調べることである。
- 図に示す例のように、その進入方向を基点に番号順に右回りに黒画素を探索する。
- 見つかった画素に移動する。黒画素が開始点で、かつ、つきの移動点が追跡済みの場合は、処理を終了し追跡結果を登録する。そうでない場合は2の処理を繰り返す。



教科書は時計回りで走査しているが、反時計回りで走査する文献も多い

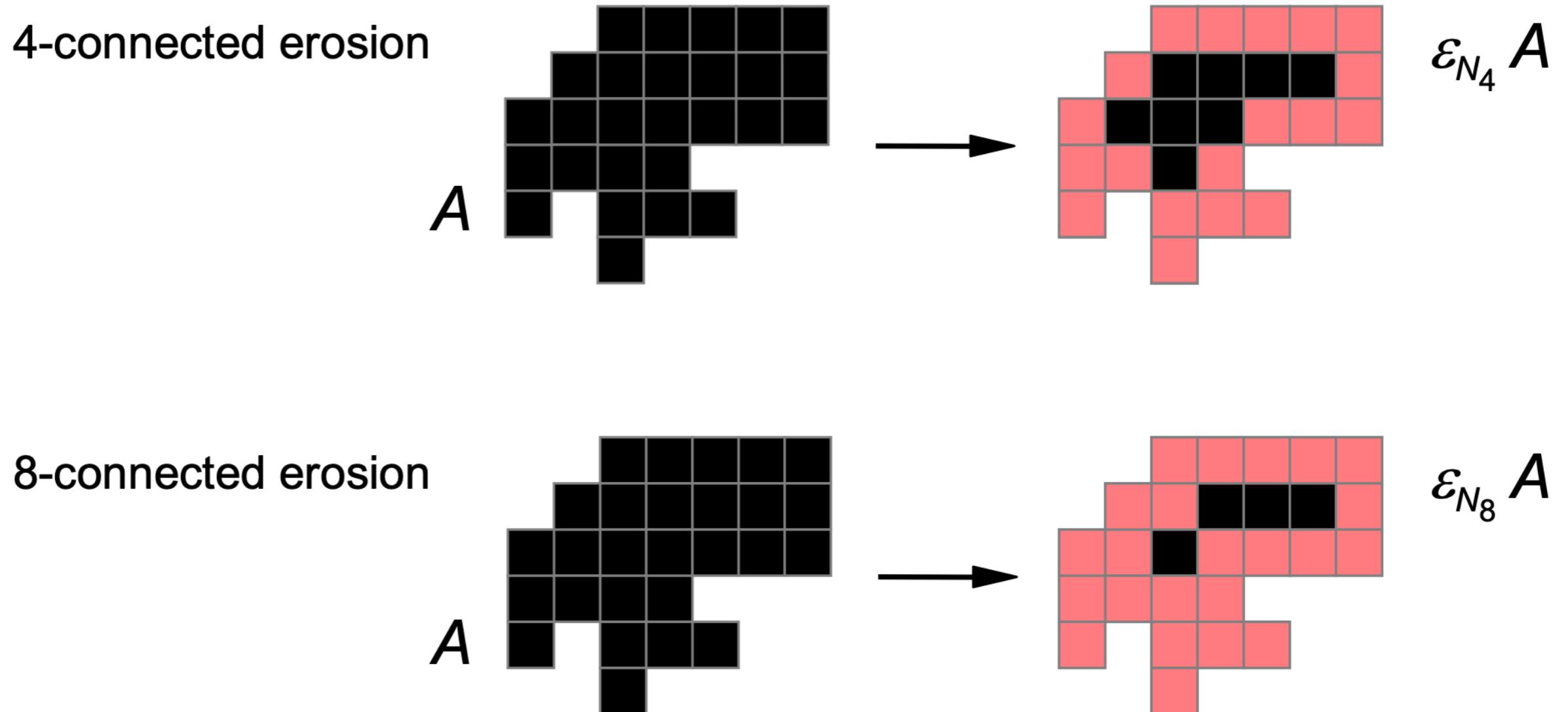
# 問題



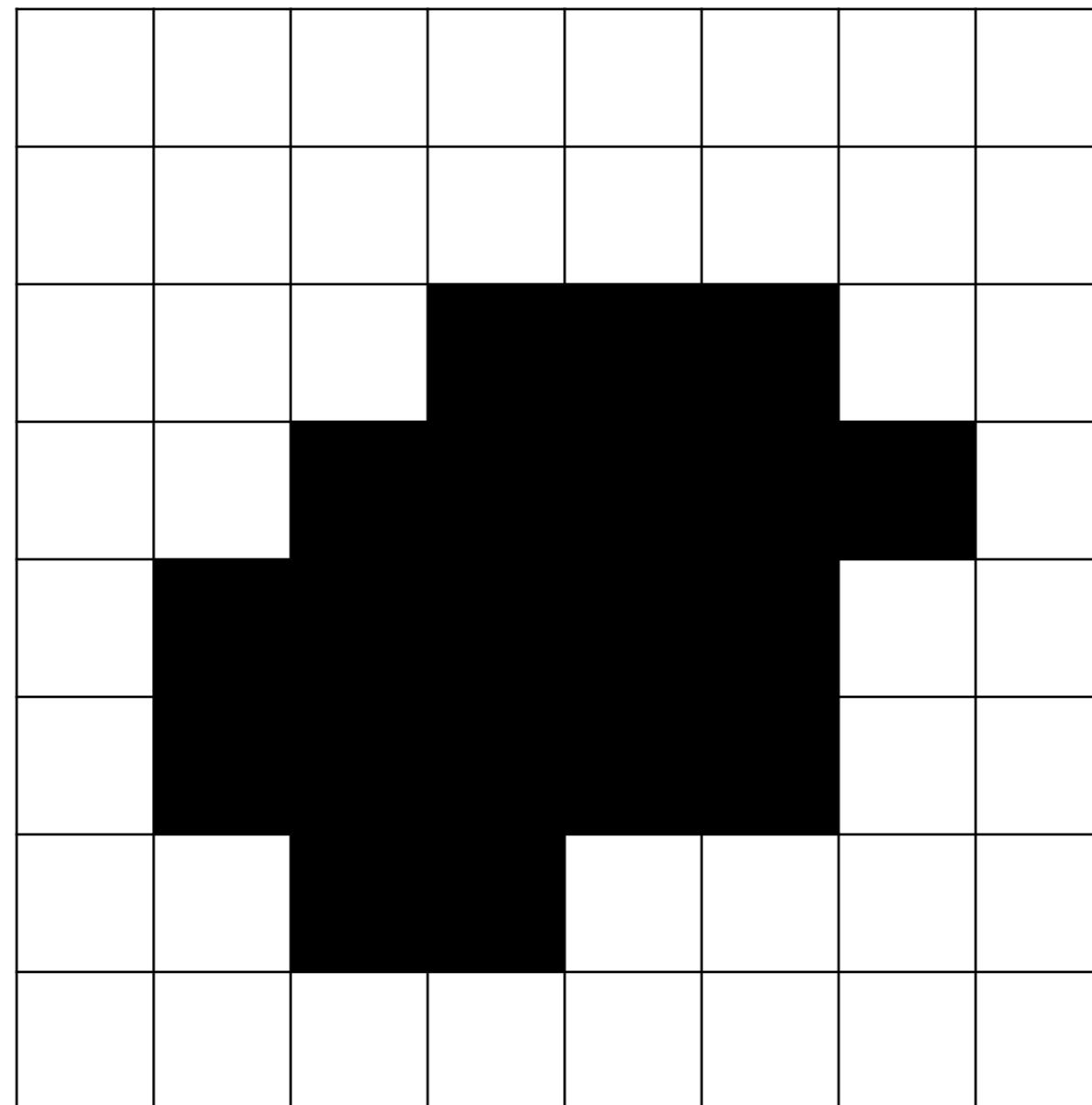
この画像の輪郭追跡をせよ

# ■ 収縮

- 背景または穴に接する対象の画素を取り除く（白に変える）処理を収縮と呼ぶ



# 問題

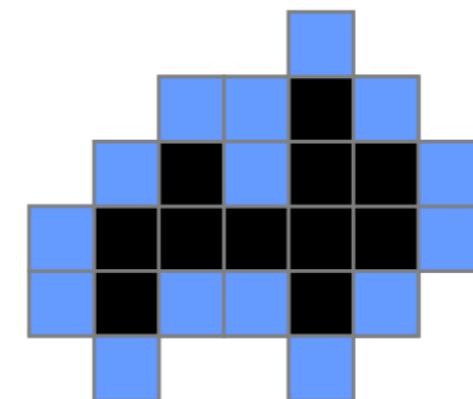


この画像の4連結収縮および8連結収縮した結果をかけ

# 膨張

- 背景または穴に接する対象の画素に、画素を一回り加える（黒くする）。

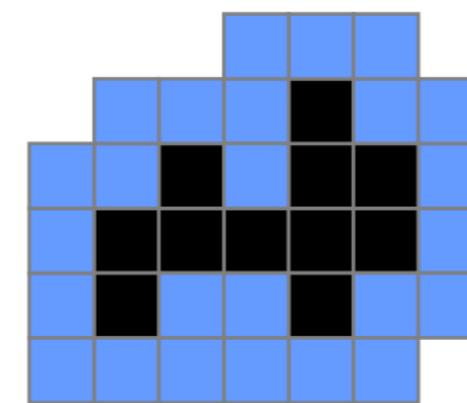
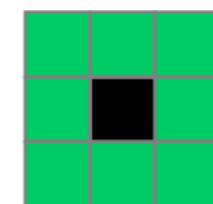
4-connected dilation



$$\delta_{N_4}$$

$$\delta_{N_4} A$$

8-connected dilation



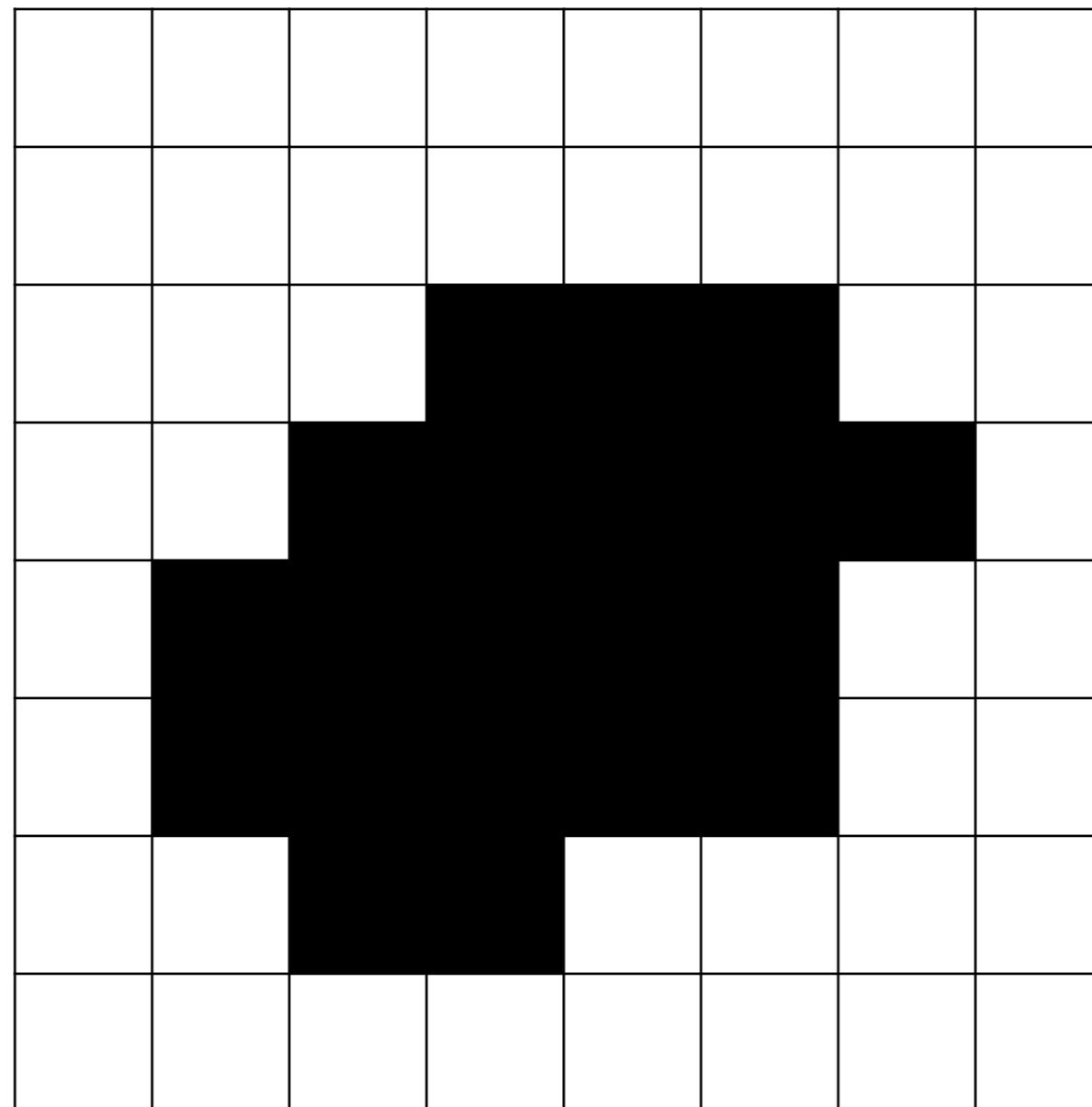
$$\delta_{N_8}$$

$$\delta_{N_8} A$$

## 問題

- 中心の1画素だけ黒く他は白い極めて大きい画像がある。この画像に4連結膨張および8連結膨張処理何度か行った場合、黒い部分はどのような形になるか答えよ。

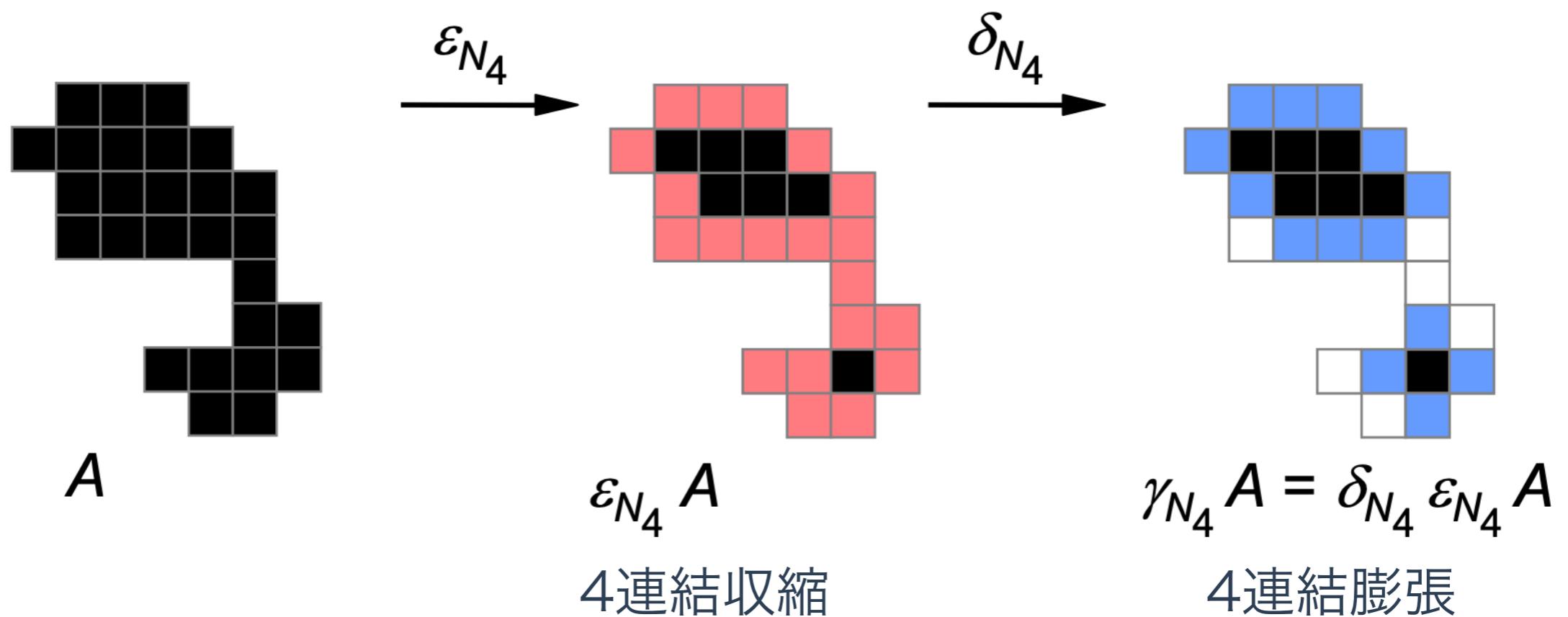
# 問題



この画像の4連結膨張および8連結膨張した結果をかけ

# ■ オープニング

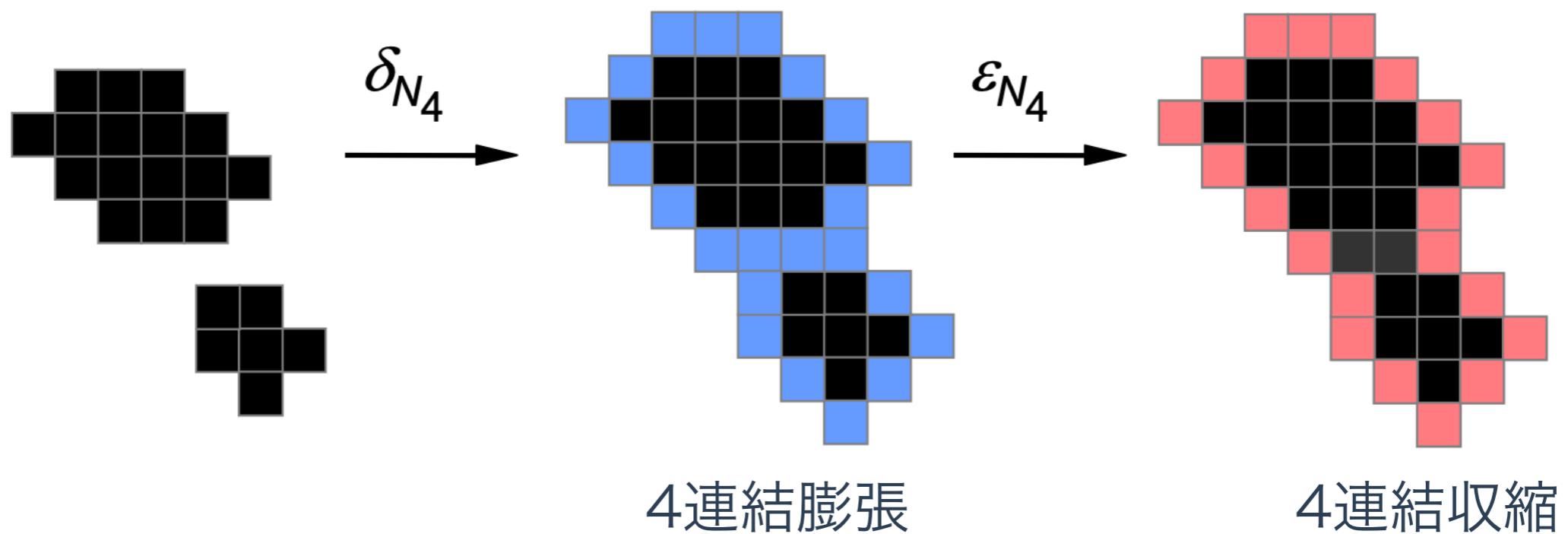
- 同じ回数、収縮して膨張する処理



接続を切断する  
小さな黒画素を消す

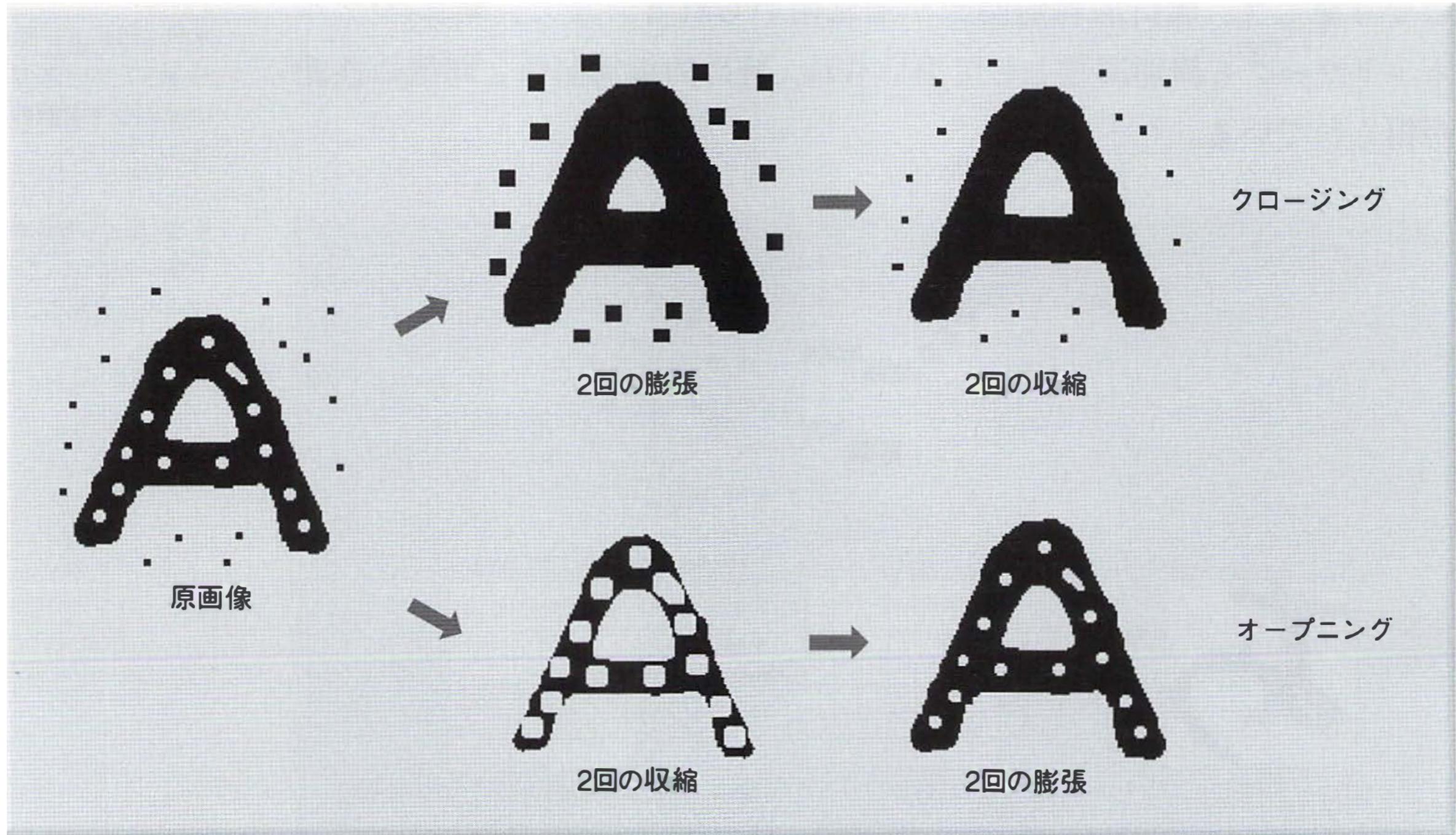
# クロージング

- 同じ回数、膨張して収縮する処理

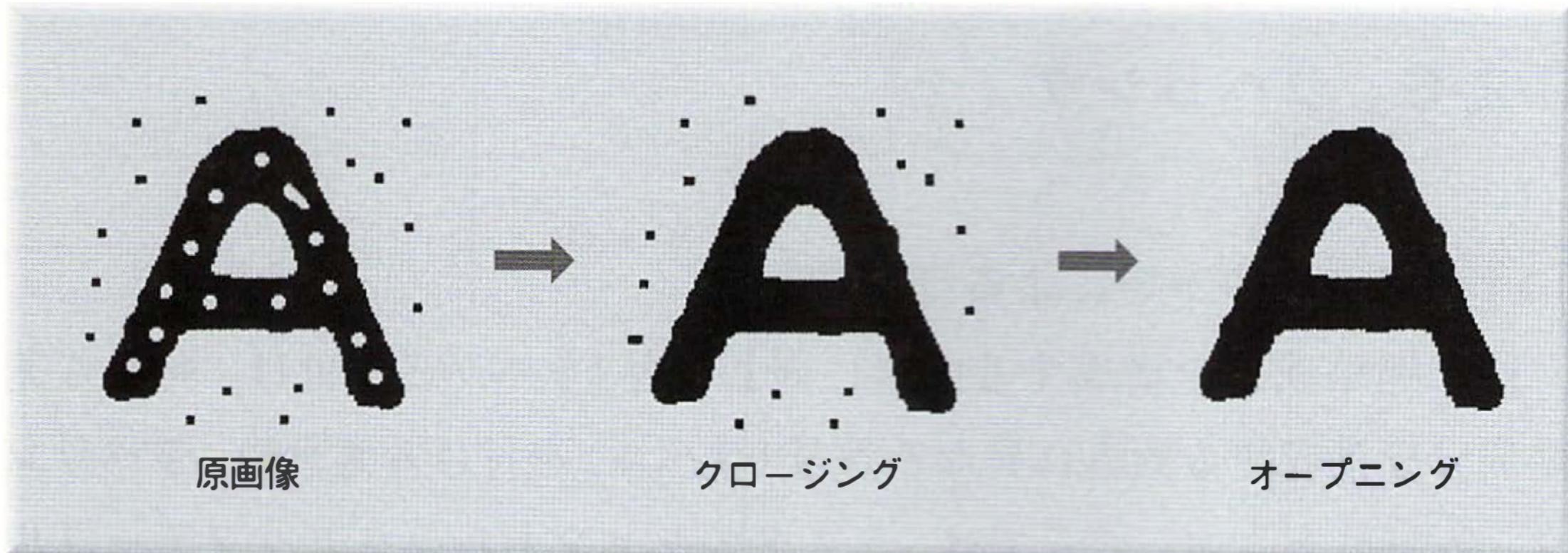


分断された連結成分をつなげる  
穴をふさぐ  
輪郭を滑らかにする

## ■ 例



## ■ クロージングとオープニングによるノイズ除去



画像取り込みで生じる小さな穴や小さな連結成分を、クロージングとオープニングで除去できる。クロージングとオープニングの収縮、膨張の回数は除去したいノイズの大きさで決める。

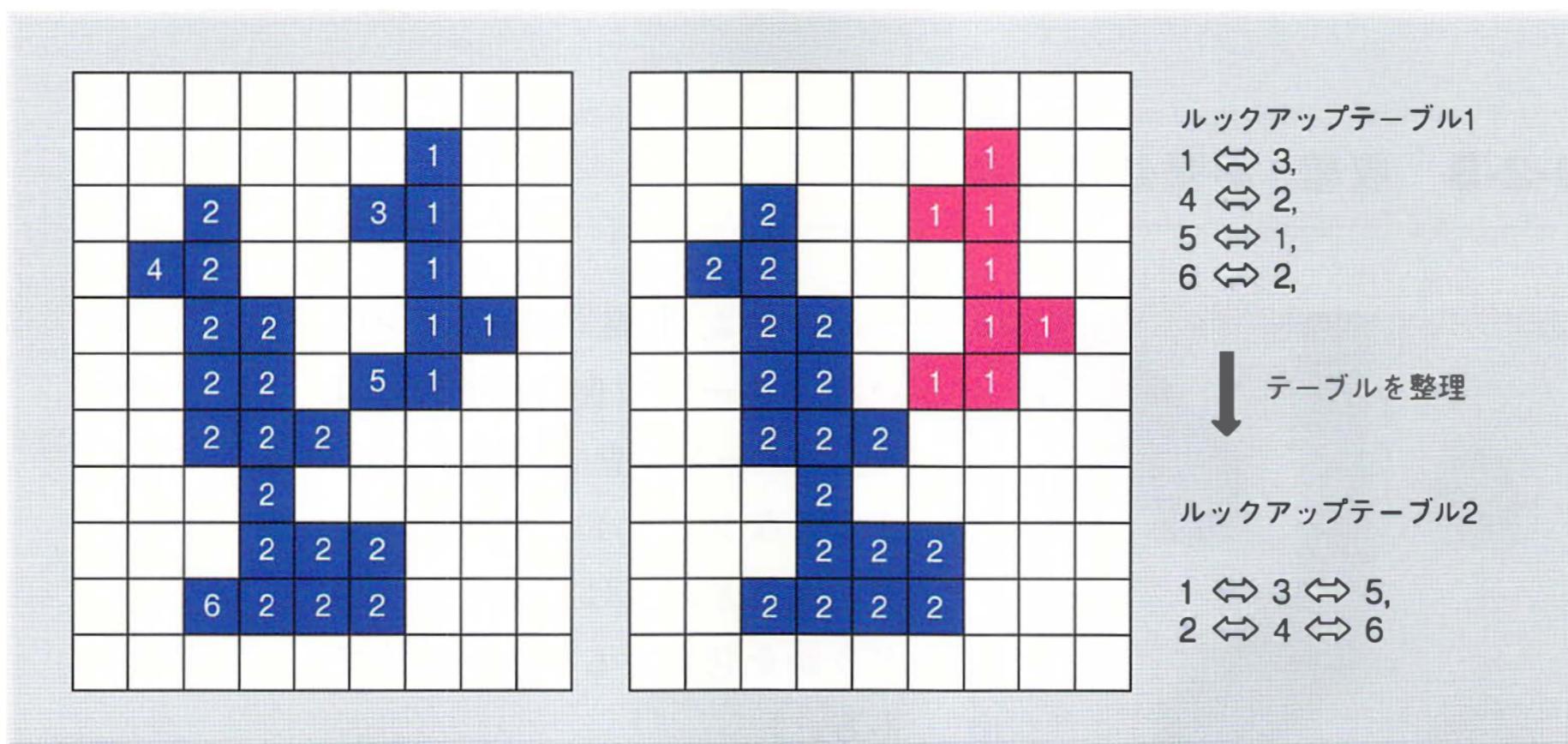
## ■ ラベリング

---

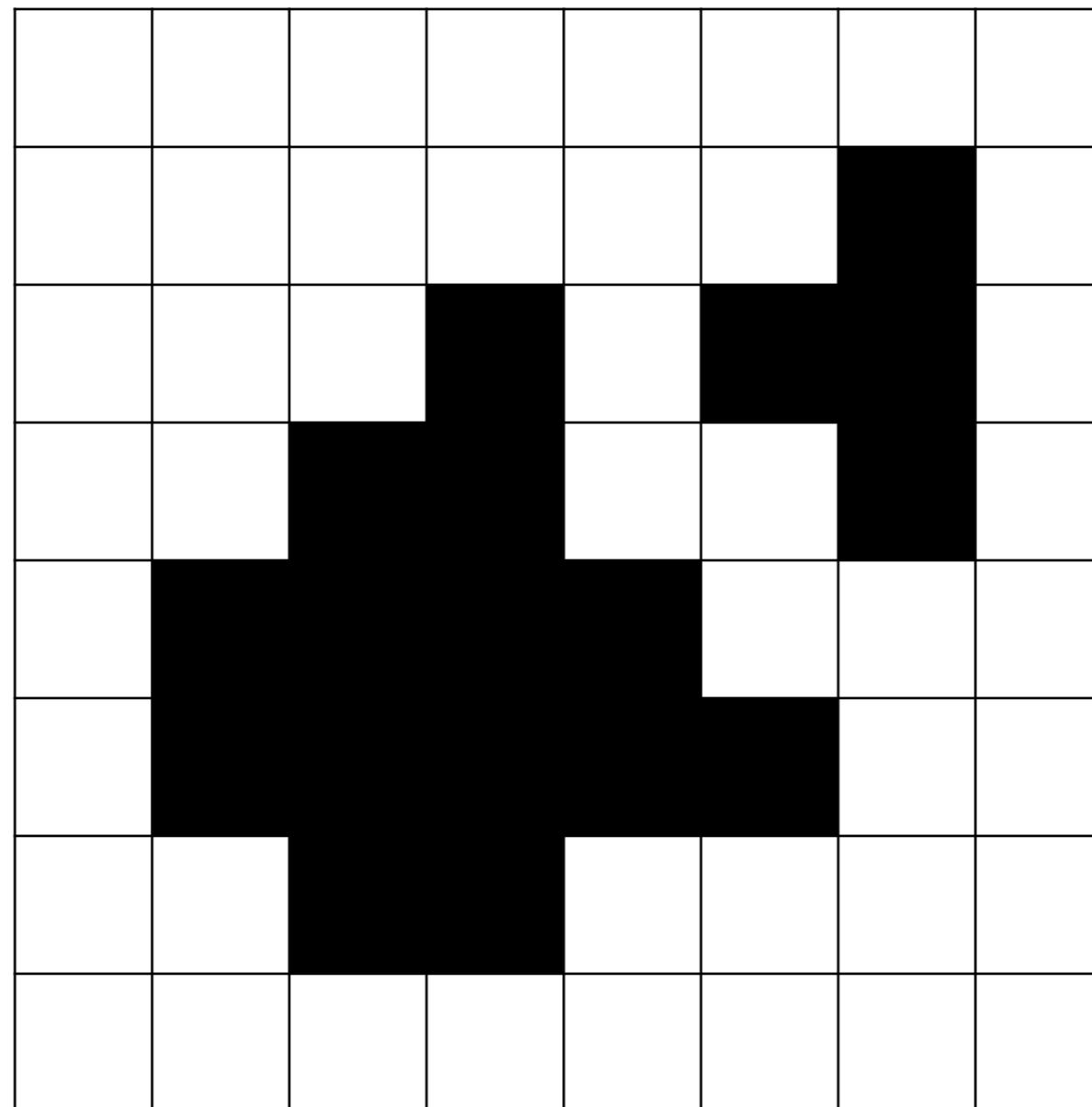
- ・同じ連続成分に所属する画素に同じ番号（ラベル）をつける処理

# ■ ラベリングの手順 (4近傍)

1. ラスタスキャンを行いラベルが付いていない画素を調べ、見つかったら注目画素とする。
2. 注目画素の上の画素がラベルを持つとき、上の画素のラベルを注目画素につける。左画素がラベルをもち、注目がそのラベルと異なるとき、ルックアップテーブルにそれらのラベルが同一の連結成分に属することを記録する。
3. 注目画素の上の画素が白画素、左の画素がラベルを持つとき、そのラベルを注目画素につける。
4. 注目画素の上も左も白がその時、新しいラベルを注目画素につける。
5. 操作する画素がなるなるまで1から繰り返す。
6. 再度ラスタスキャンを行い、ルックアップテーブルを参照しながら、同一の連結成分に属するラベル群から、最も小さいラベルを選んでつけ直す。



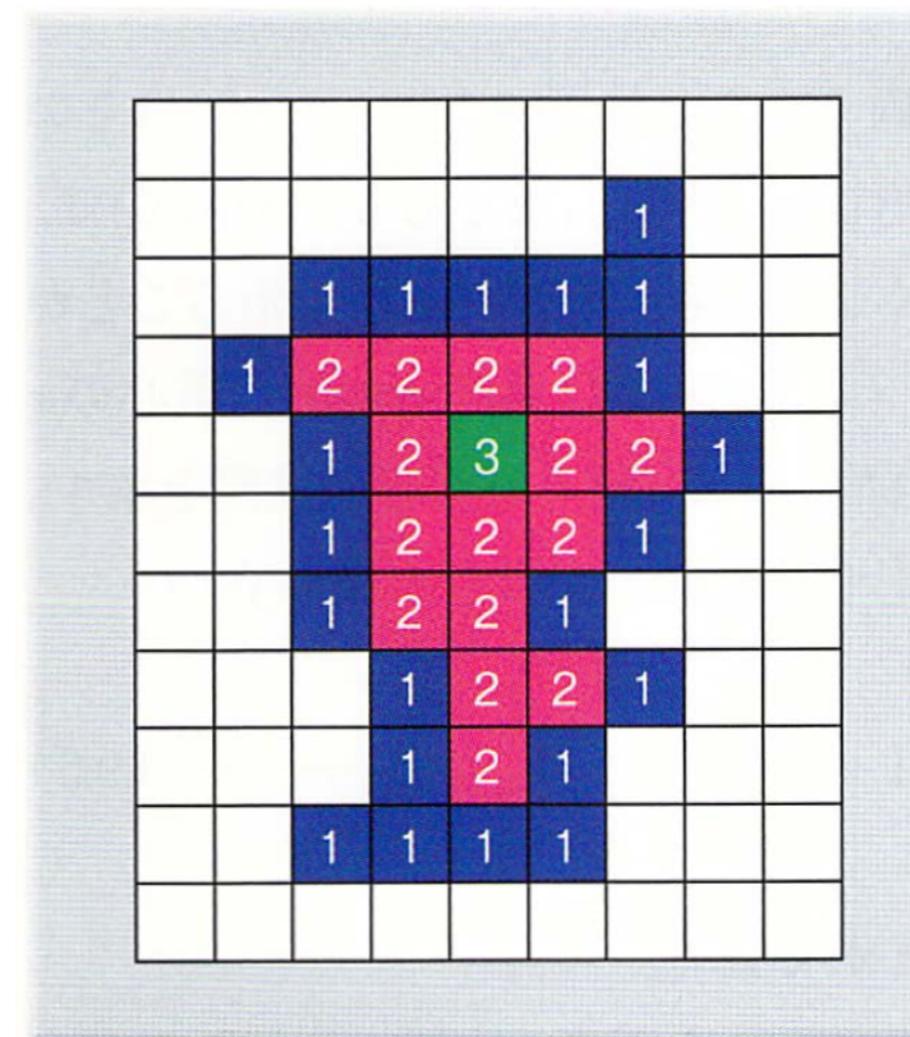
# 問題



この図形のラベリングをせよ

# ■輪郭追跡による距離変換画像

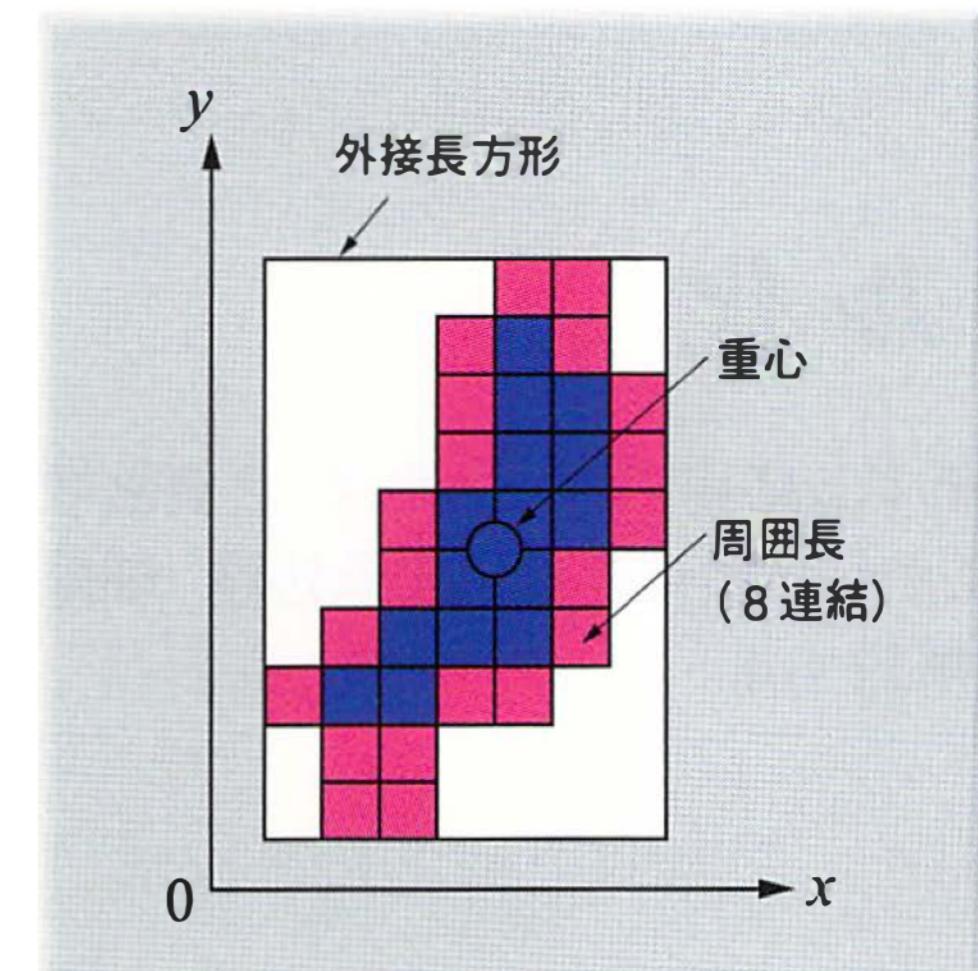
- 距離変換画像
  - 連結成分の各画素に周辺から距離の値を書き込んだもの



■図9.17——8連結のラベリング処理で背景からの距離変換画像を求めた例

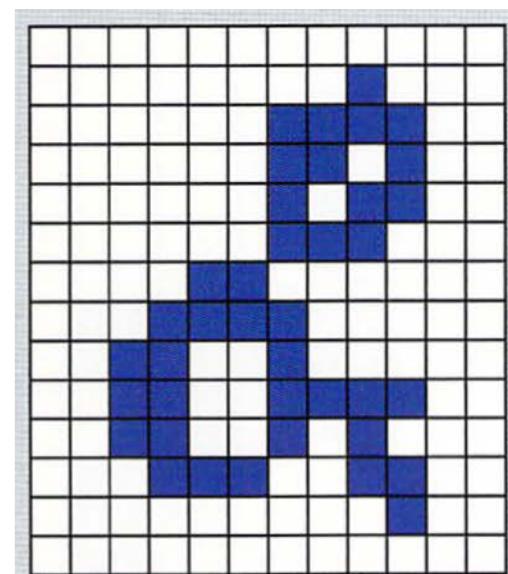
# ■ 形状パラメタ

- 形状パラメタ
  - 連結成分の形状の特徴を数値化したもの
- 重心
  - 連結成分の画素に等しい重さがあるとしたときの図形全体の重さの中心
- 外接長方形 (Bounding box)
  - 対象とする連結成分に接する最小の長方形
  - 対象の大まかな大きさを知る場合に用いる
- 面積
  - 連結成分を構成する画素数



# ■ 形状パラメタ

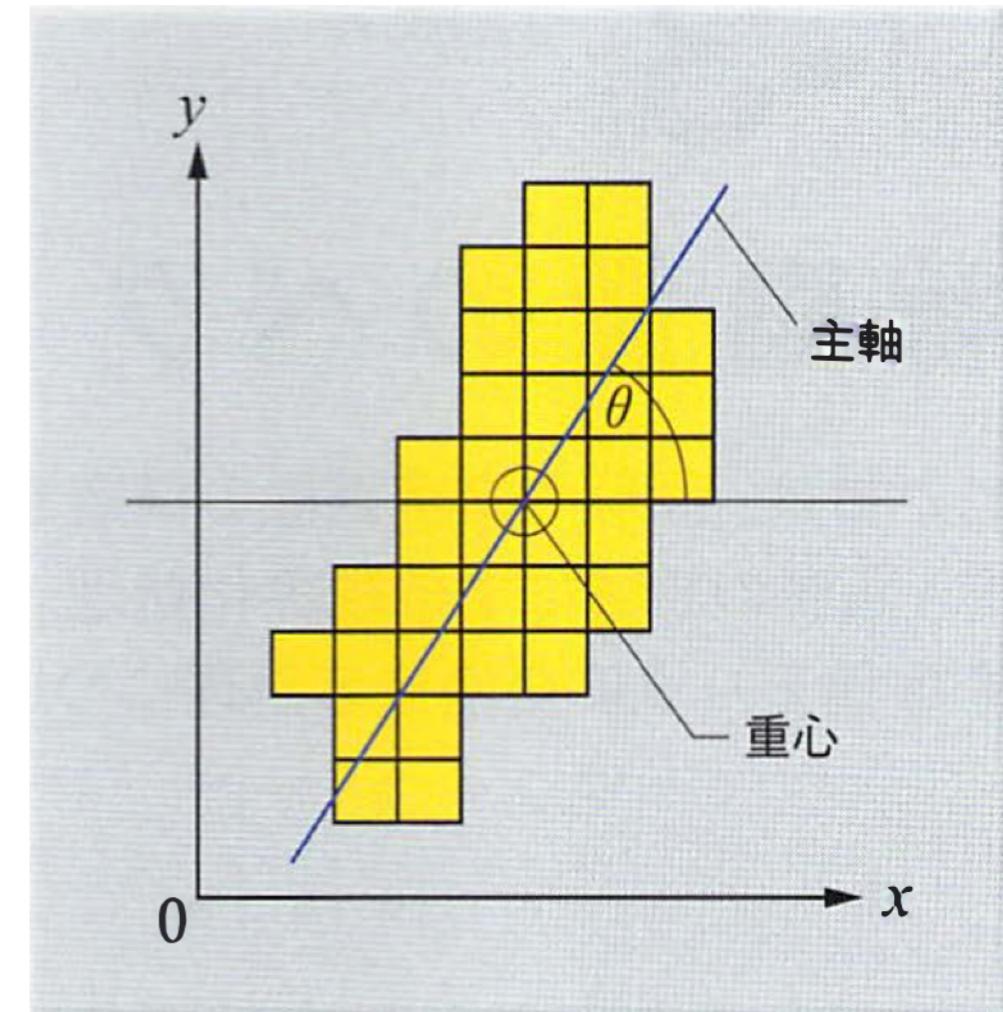
- 周囲長
  - 輪郭追跡を一周する移動量
  - 8連結の場合, 上下左右に追跡移動する数C1, 斜めに追跡移動する数C2とすると周囲長は
  - 4連結の場合, 上下左右に追跡移動する数が周囲長  
$$C_1 + \sqrt{2}C_2$$
- 円形度
  - 図形がどれだけ円に近いかを表す尺度
  - 面積S, 周囲長をLとした場合
  - 対象が円のとき1となり, 複雑な図形ほど小さな値になる
- オイラー数 
$$4\pi S/L^2$$
  - 連結成分の数から穴の数を引いたもの



## ■ モーメント特徴

$$M(p, q) = \sum f(i, j)i^p j^q$$

- ・上記の式で表される画像の特徴量
- ・ $M(0, 0)$ は面積を表す。
- ・ $M(1, 0)/M(0, 0)$ はx軸方向の重心
- ・ $M(0, 1)/M(0, 0)$ はy軸方向の重心
- ・主軸の方向は次の方程式を解くことで求まる



$$\tan^2 \theta + \frac{M(2, 0) - M(0, 2)}{M(1, 1)} \tan \theta - 1 = 0$$

# ■ 距離

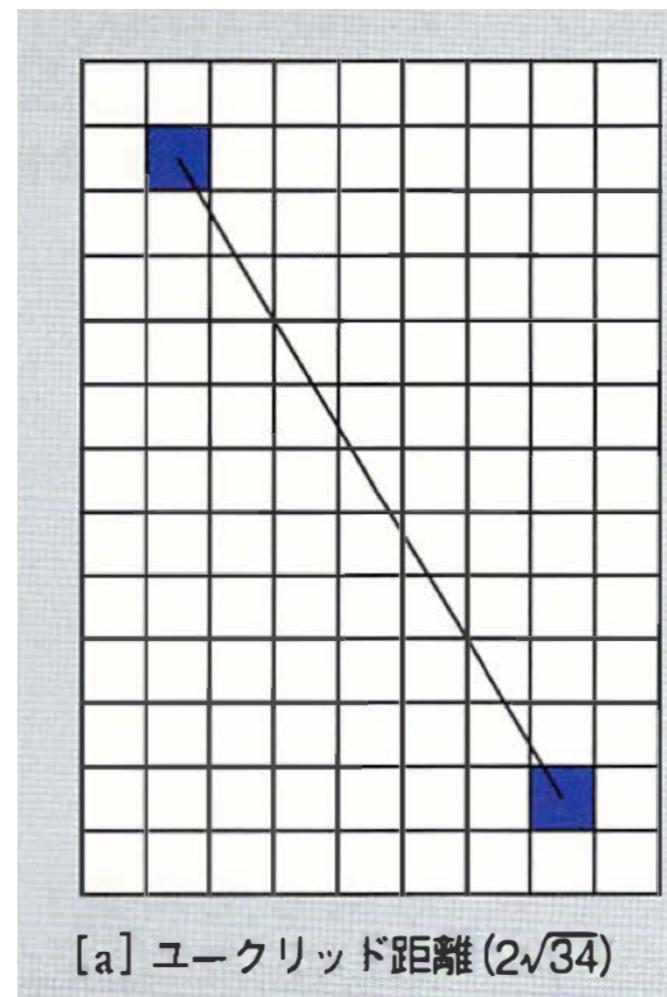
---

- 画像上での距離
- ユークリッド距離
- マンハッタン距離(市街地距離)
- チェス盤距離

## ■ ユークリッド距離

- 我々が距離や長さという場合は大抵ユークリッド距離のことである。 $L_2$ 距離とも呼ばれる。
- $(x_a, y_a), (x_b, y_b)$ の2点間の距離は、次のように表せる。

$$L = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$$

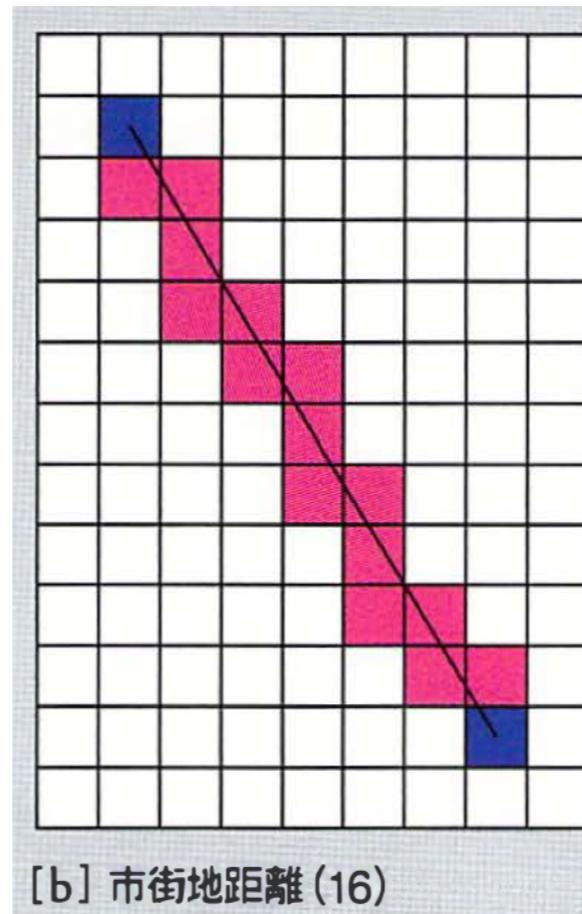


## ■ マンハッタン距離（市街地距離）

- ・碁盤の目のような都市上で、 $(x_a, y_a), (x_b, y_b)$ の2点間の距離を考えた場合、その距離は

$$L = |x_a - x_b| + |y_a - y_b|$$

- ・のように表すことが出来る。これをマンハッタン距離（市街地距離、 $L_1$ 距離）という。
- ・4近傍の移動距離となる。

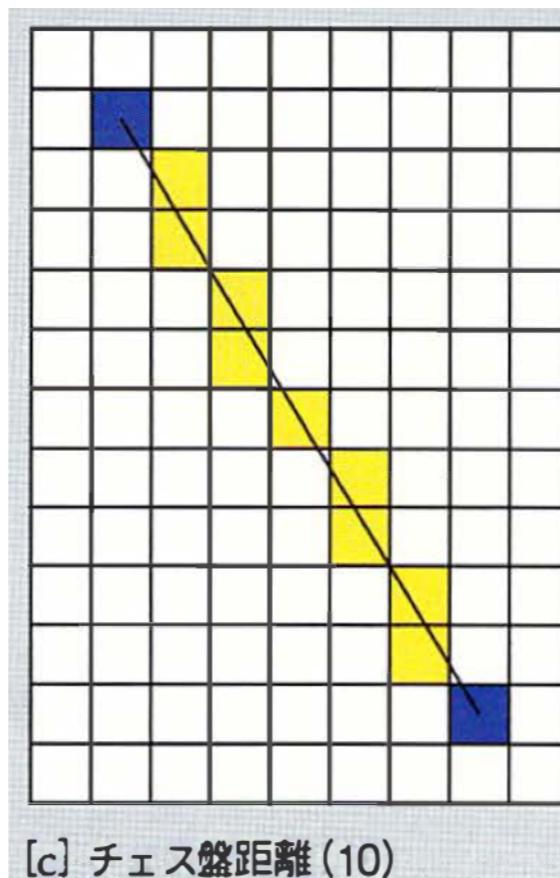


## ■ チェス盤距離（チェビシェフ距離）

- 斜め移動1と考えた場合、 $(x_a, y_a), (x_b, y_b)$ の2点間の距離は

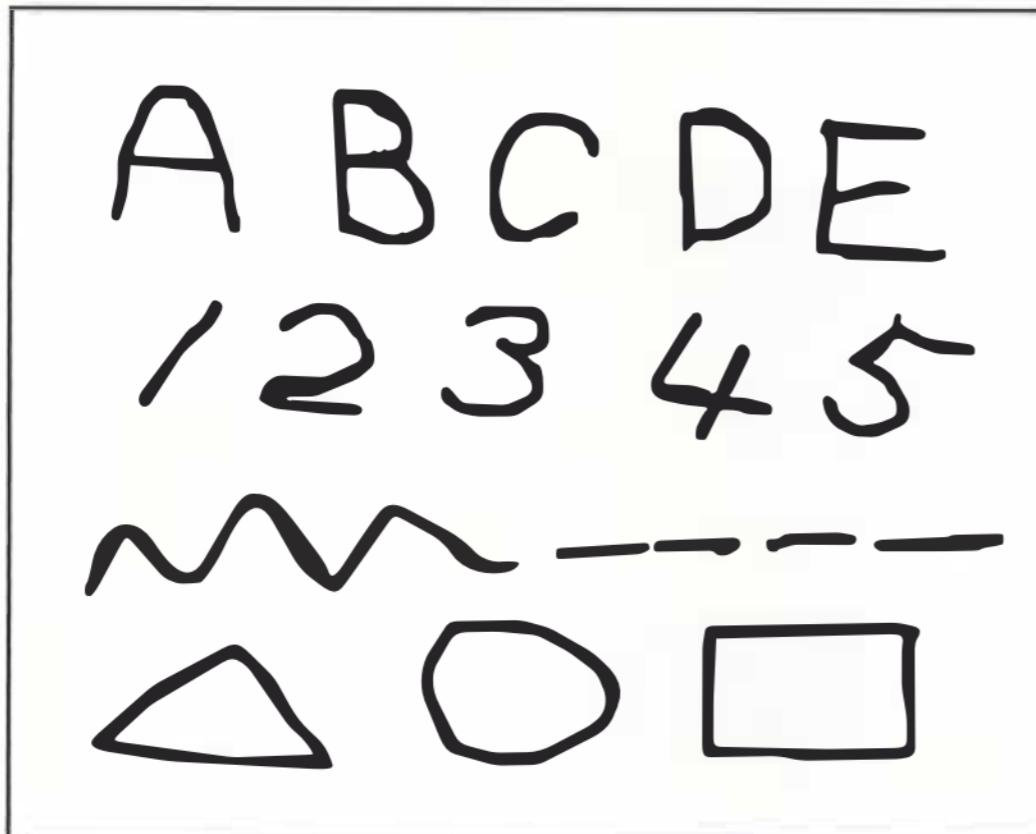
$$L = \max(|x_a - x_b|, |y_a - y_b|)$$

- のように表すことが出来る。これをチェス盤距離（チェビシェフ距離、 $L_\infty$ 距離）という。
- 8近傍の移動距離となる。

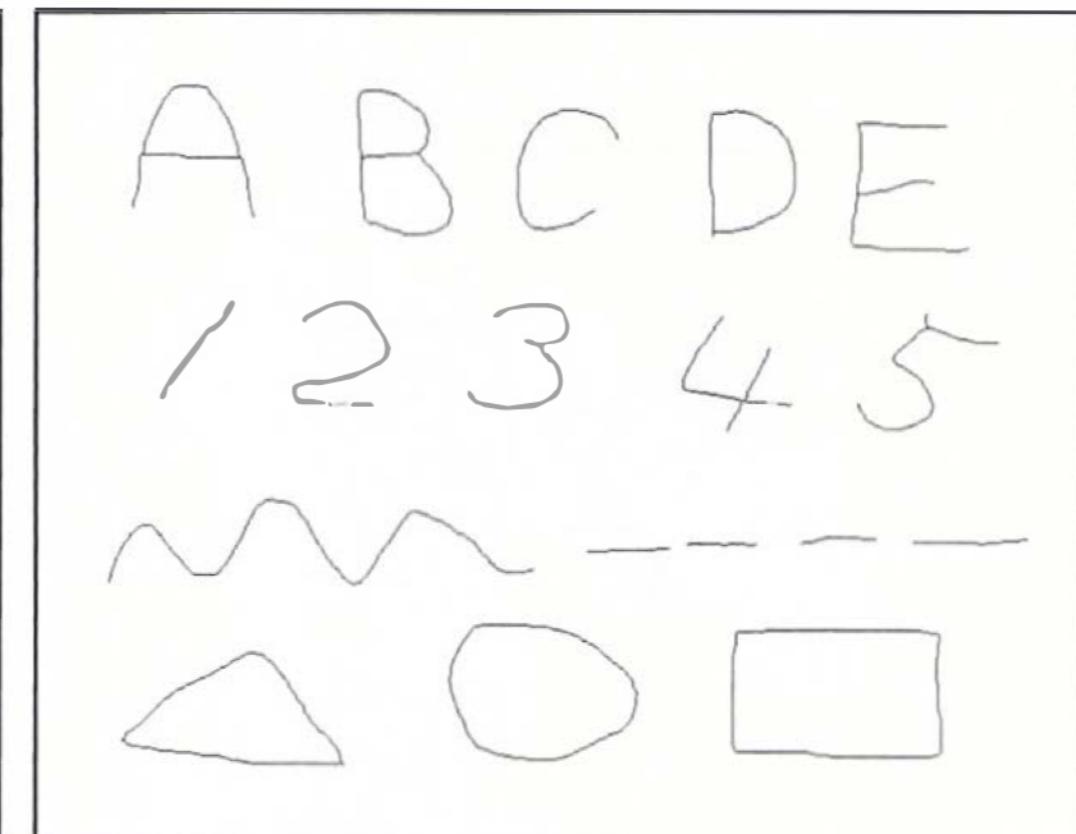


## ■ 細線化（骨格化）

- ・領域成分から線幅1画素の中心線を抽出すること
- ・画像識別の前処理やベクトル化のために用いる。



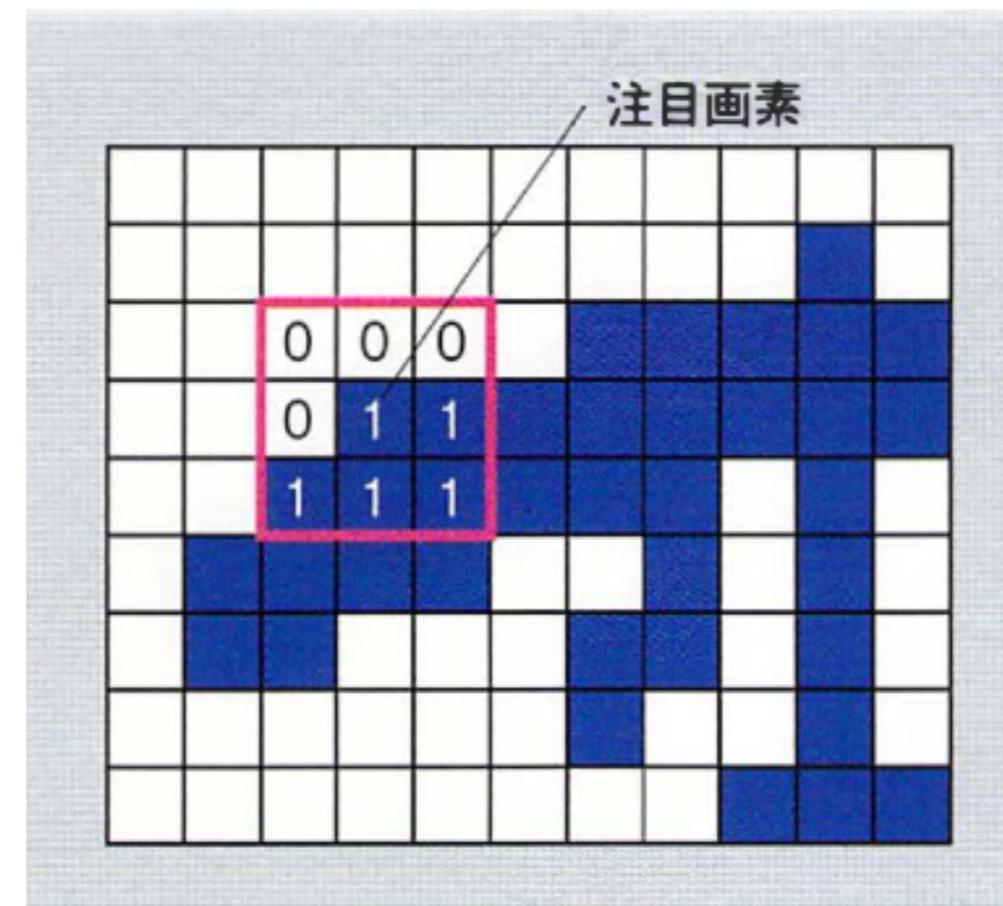
[a] 文字や図形の2値画像



[b] 細線化画像

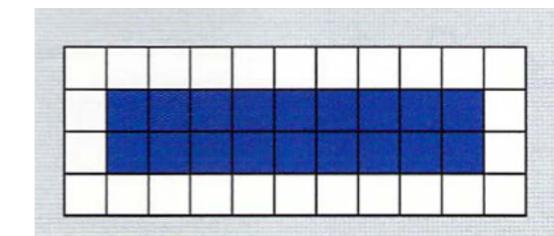
# ■ 細線化手法

- ・注目画素を中心とした $3 \times 3$ の領域の画素値パターンにより、注目画素の値が決まる。
  - ・境界線上の黒画素である。
  - ・白画素に変更しても連結性が保存される
  - ・端点でない
  - ・以上の条件を満たす注目画素は白画素とする。



# ■ 逐次法と並列法

- 逐次法
  - 画素値の更新をすぐさま対象画像に反映させ, 次の処理を行う
  - 細線化の結果が中心になりにくく.
- 並列法
  - 画素値の更新用の検証用画像と更新を反映させる更新用画像を用意する. 全画素処理が済んだら, 更新用画像を検証用画像とする.
  - 2画素幅の線画像が消滅する.



# ■ 特徴点抽出

- ・細線化によって得られた画像の画素を, 端点, 分岐点, 孤立点, 通過点の特徴点に分類する.

0	0	0
0	1	0
*	1	*

*	0	0
1	1	0
*	0	0

*	1	*
0	1	0
0	0	0

0	0	*
0	1	1
0	0	*

0	0	0
0	1	0
1	0	0

1	0	0
0	1	0
0	0	0

0	0	1
0	1	0
0	0	0

0	0	0
0	1	0
0	0	1

[a] 端点

*	1	*
1	1	1
0	0	0

0	1	*
0	1	1
0	1	*

0	0	0
1	1	1
*	1	*

*	1	0
1	1	0
*	1	0

1	0	0
0	1	0
1	0	1

1	0	1
0	1	0
1	0	0

1	0	1
0	1	0
0	0	1

0	0	1
0	1	0
1	0	1

[b] 分岐点 (3分岐)

*	1	*
1	1	1
*	1	*

1	0	1
0	1	0
1	0	1

0	0	0
0	1	0
0	0	0

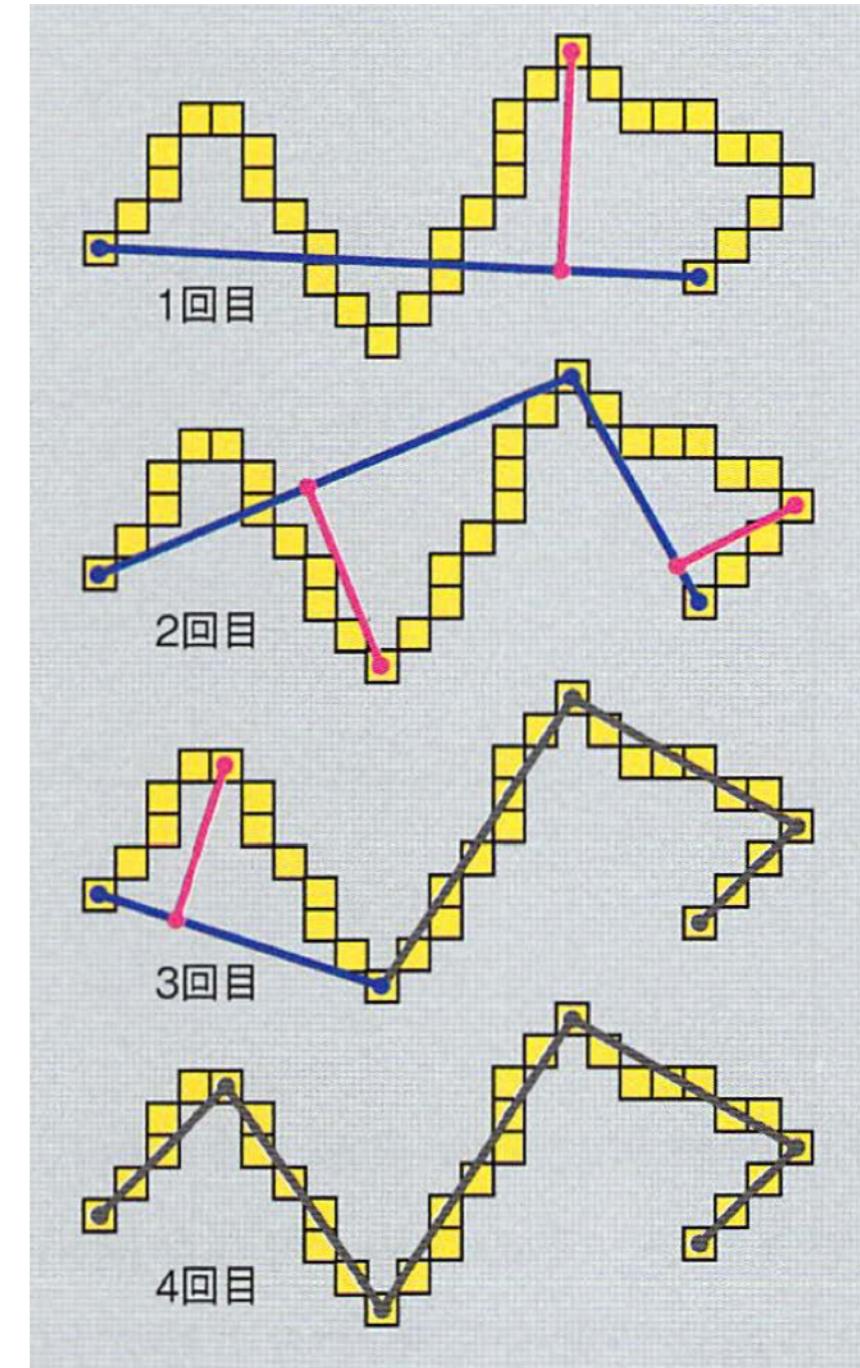
\*は0でも1でもよい。理想的な細線化を行った場合は、\*はすべて0となる。図の論理フィルタによって、端点、分岐点、孤立点を検出できる。それ以外は通過点となる。

[c] 分岐点 (4分岐)

[d] 孤立点

## ■ 2分割法によるベクトル化（線分近似）

1. 画像列の始点と終点を繋いだ直線に対し,  
各画素の距離を計算する。
2. 最大距離が閾値以下の場合は線分として  
採用する。そうでない場合は、その画素  
で画像列を分割し、分割された画素列で1  
の処理を再帰的に繰り返す。
3. 画素列の分割がない場合、処理を終了す  
る。



## ■ 角点検出によるベクトル化（線分近似）

- 画像列の任意の点 $P_i$ に対し、 $k$ 画素離れた前後の点 $P_{i-k}$ と点 $P_{i+k}$ を結ぶ線分のなす角が小さい時、点 $P_i$ を角点とする。
- これでは、なめらかに曲がっている曲線を線分化出来ない。そのため、角点を検出してそれを分割点と下の値、分割点の間を2分割法で線分近似する方法が用いられる。

