

マルチメディア信号解析

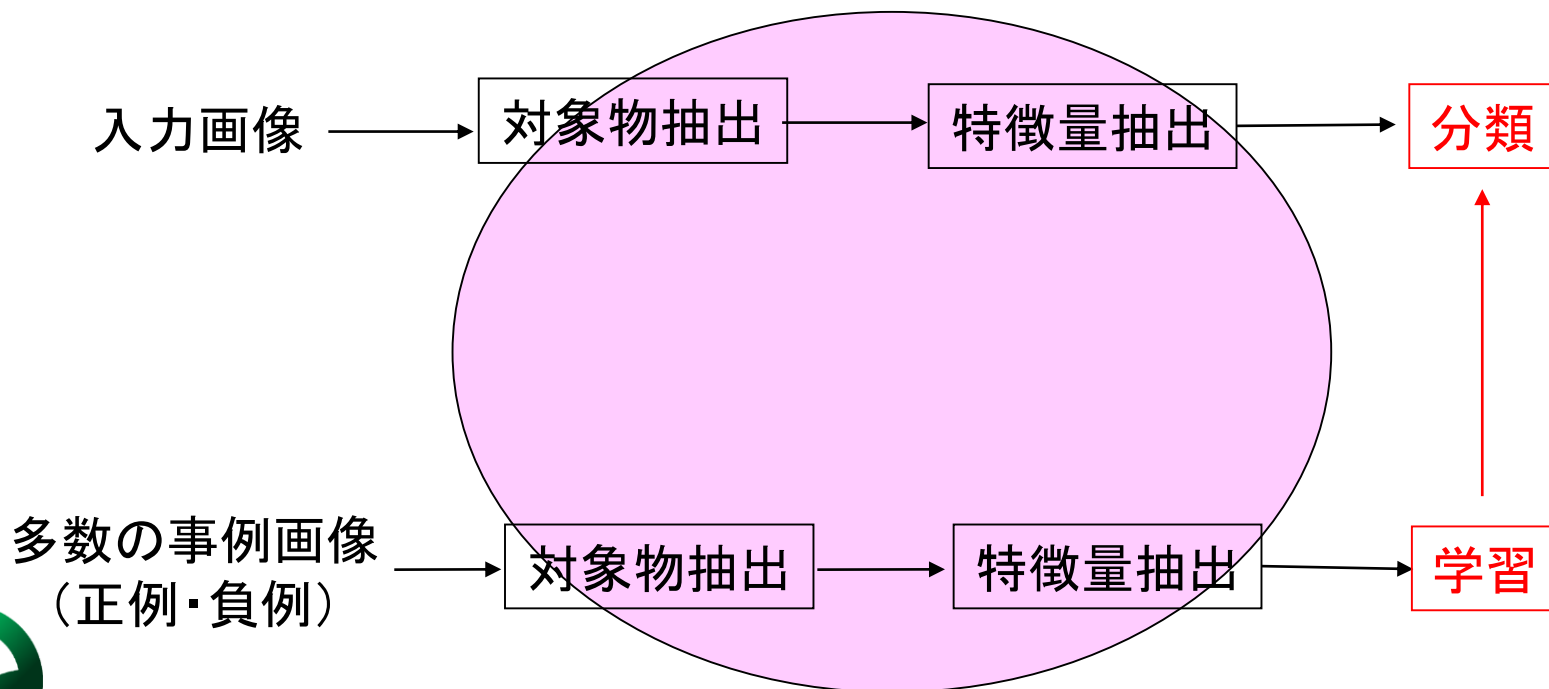


対象物の抽出

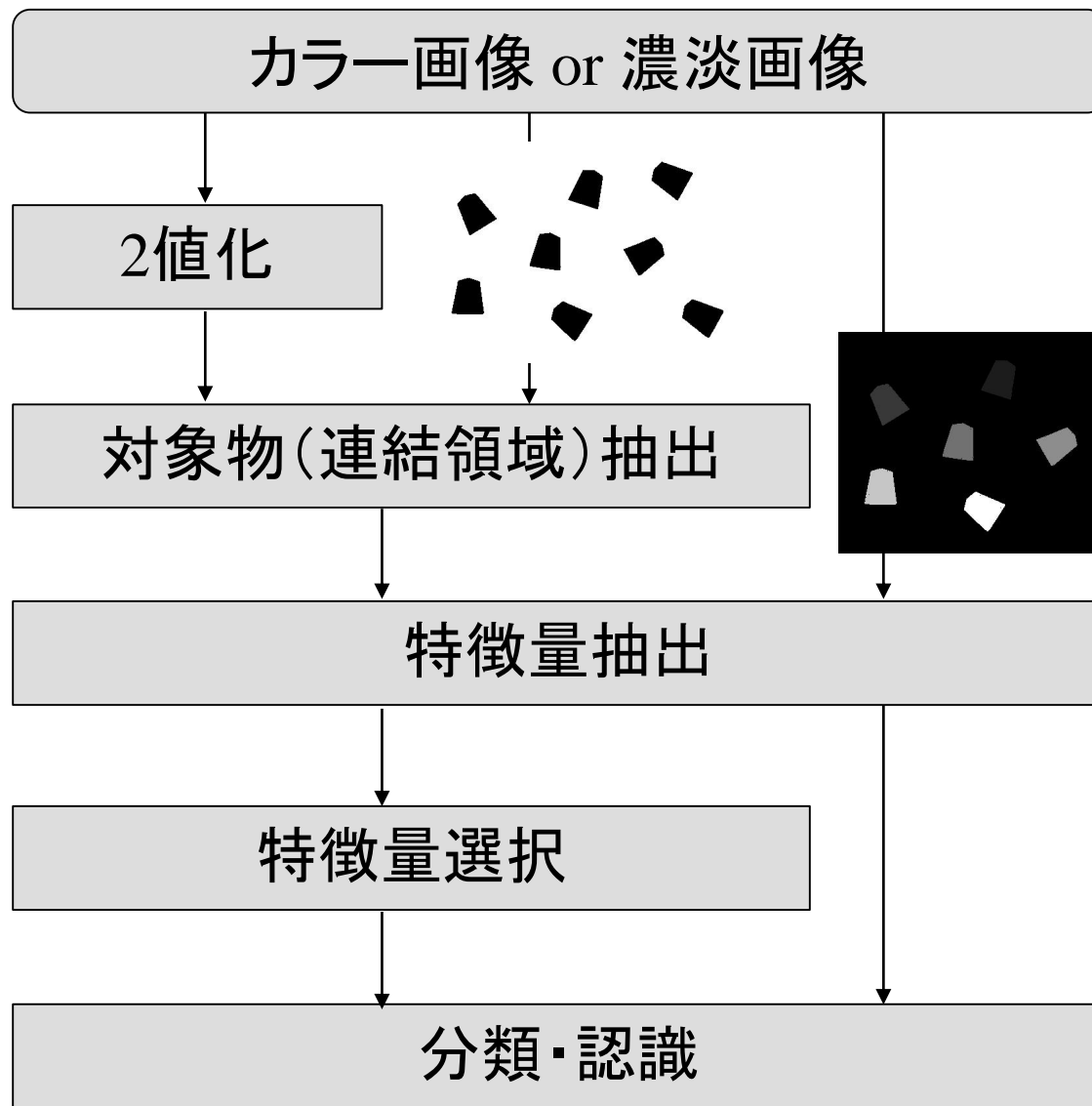


統計的パターン認識

2値化・領域分割などの処理によって
画像から対象物を検出し、色・形・テクスチャなどの
特徴量を計測した後に、それらが何であることを
統計的に分類・認識する。



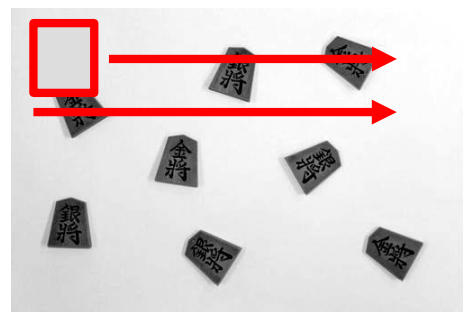
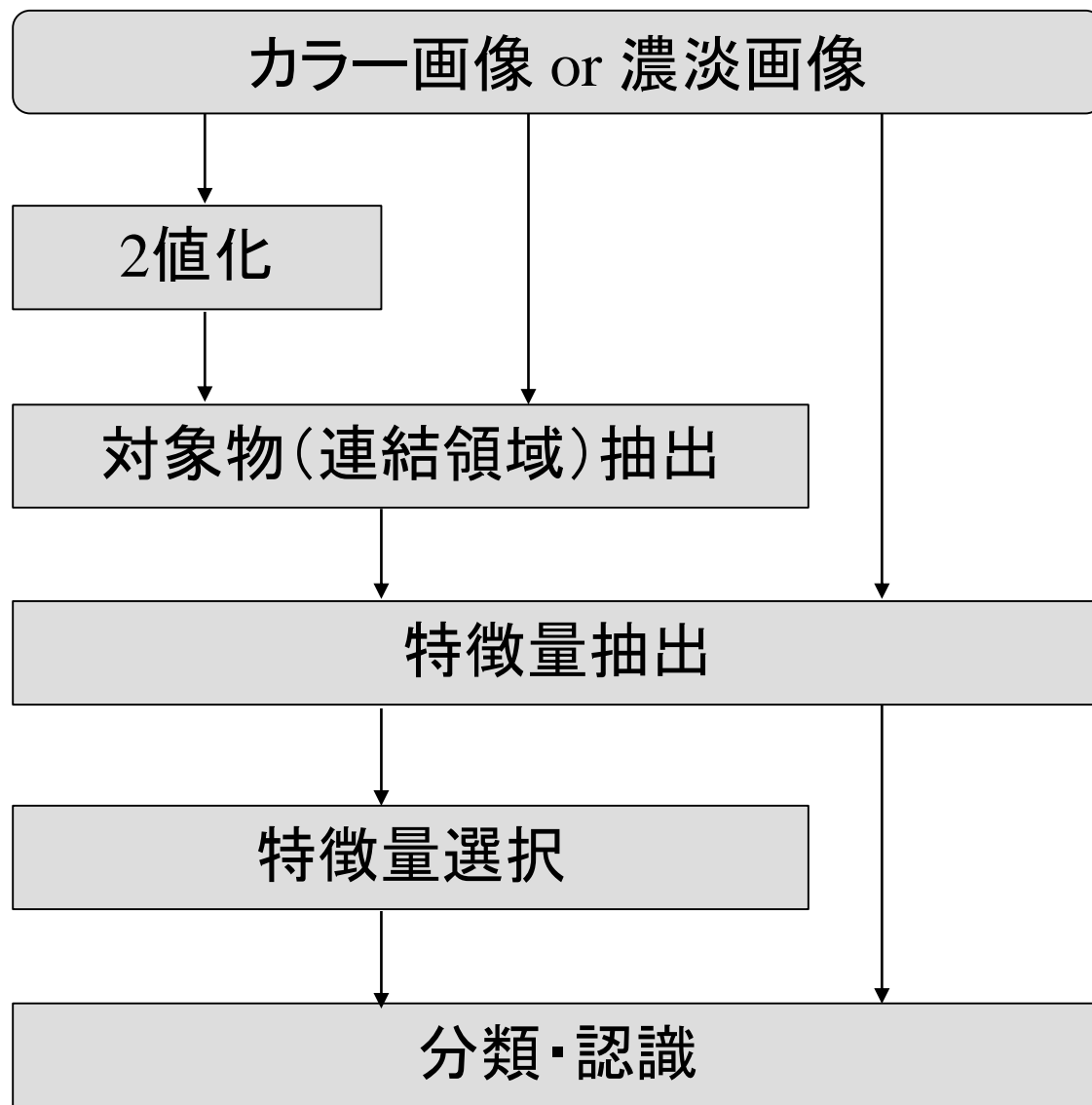
対象物 & 特徴量抽出



金 or 銀？



対象物 & 特徴量抽出



金 or 銀
or それ以外？



対象物の抽出



画像の2値化処理

閾値処理

$$f_t(y, x) = \begin{cases} 1; f(y, x) \geq t \text{ のとき} \\ 0; f(y, x) < t \text{ のとき} \end{cases}$$

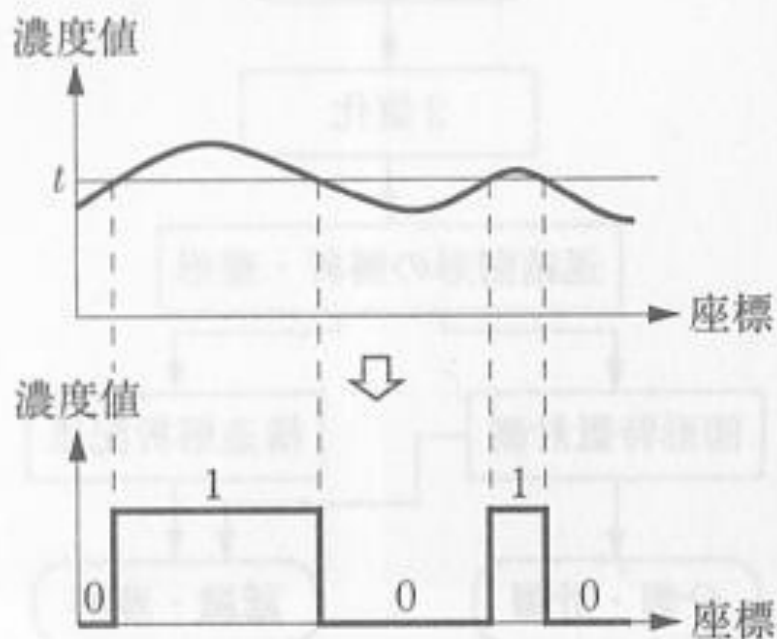


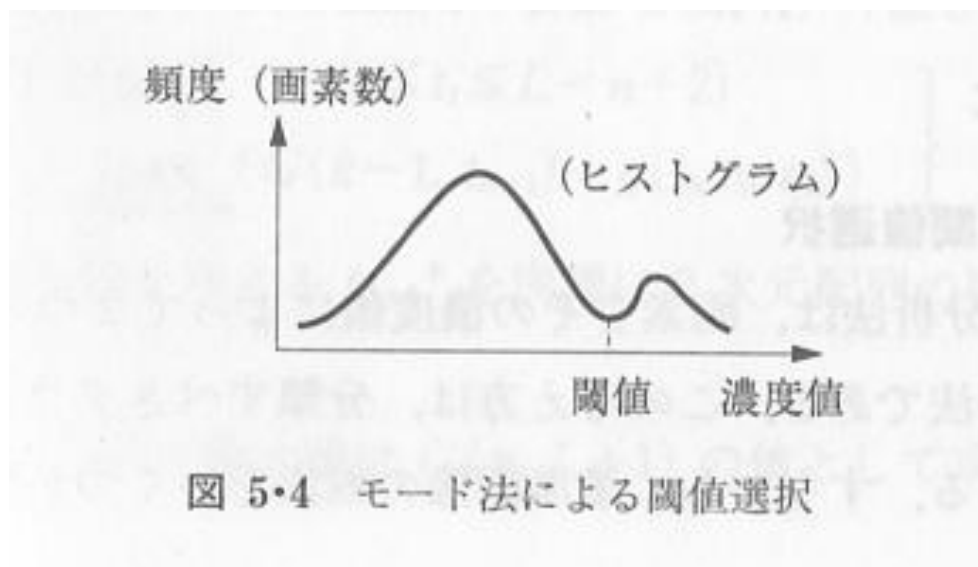
図 5・2 閾値処理



閾値選択

モード法

濃度ヒストグラムが双峰型の分布になる場合、
2つの峰の間の谷底となる濃度値を閾値 t とする。

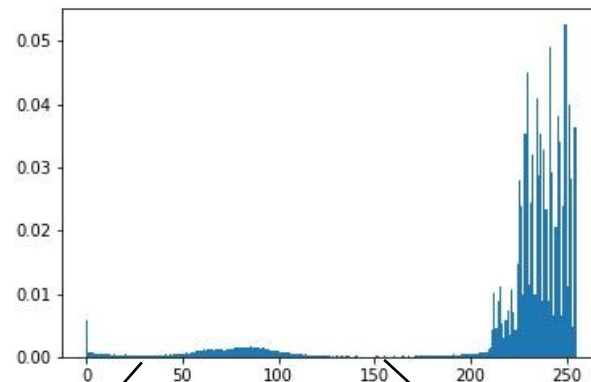


対象図形と背景の濃度値の差が大きい場合に使用



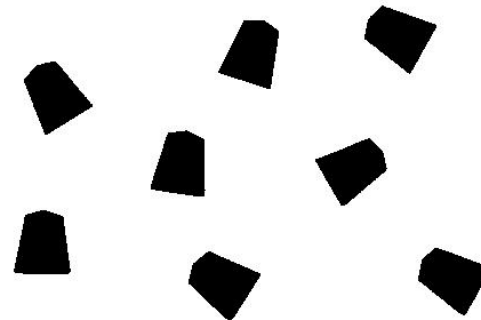
閾値選択

モード法



$t = 25$

$t = 150$



閾値選択

判別分析法(大津の二値化)

閾値 t で2つのクラスに分割したとき、2つのクラスが最もよく分離するようにパラメータ t を決める。

分離度の尺度: クラス間分散とクラス内分散の比

クラス i の画素数: ω_i 、平均濃度値: M_i 、分散: σ_i^2

全画素の平均濃度値: M_T 、分散: σ_T^2

$$\text{クラス内分散: } \sigma_W^2 = \frac{\omega_1 \sigma_1^2 + \omega_2 \sigma_2^2}{\omega_1 + \omega_2}$$

$$\text{クラス間分散: } \sigma_B^2 = \frac{\omega_1 (M_1 - M_T)^2 + \omega_2 (M_2 - M_T)^2}{\omega_1 + \omega_2} = \frac{\omega_1 \omega_2 (M_1 - M_2)^2}{(\omega_1 + \omega_2)^2}$$

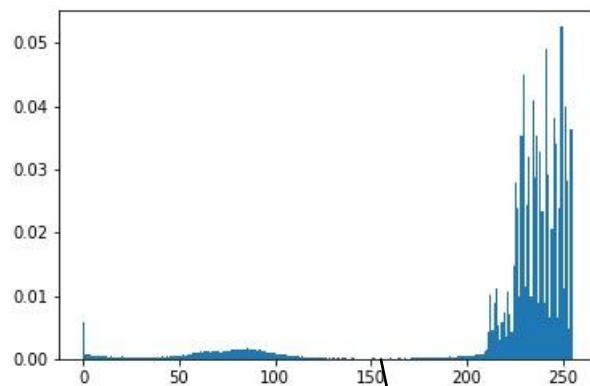
$$\sigma_T^2 = \sigma_W^2 + \sigma_B^2 \text{ より}$$

$$\frac{\sigma_B^2}{\sigma_W^2} = \frac{\sigma_B^2}{\sigma_T^2 - \sigma_B^2} \quad \longrightarrow \quad \sigma_B^2 \text{ を最大にする } t \text{ を求める}$$

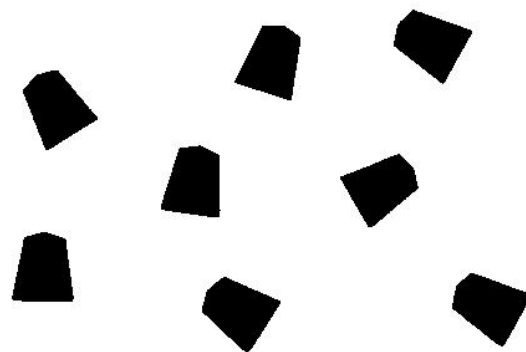


閾値選択

判別分析法



$t = 152$



多値化

反復法による閾値選択アルゴリズム

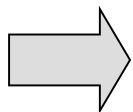
Step1: $n-1$ 個の初期閾値 t_1, t_2, \dots, t_{n-1} を等間隔に選んで n 値化する。
このとき、 $t_0 = 0$, $t_n = L+1$ とし、濃度値が区間 $[t_{i-1}, t_i)$ にある画素を第 i クラスに分類する。

Step2: 各クラスの画素の平均濃度値 $M_i (i = 1, 2, \dots, n)$ を求める。

Step3: 次式により隣り合うクラスの平均間の中点を求め、新たに $n-1$ 個の閾値とする。

$$t_i = \frac{M_i + M_{i+1}}{2} (i = 1, 2, \dots, n-1)$$

Step4: 閾値に変化がなければ終了。変化があればStep2へ

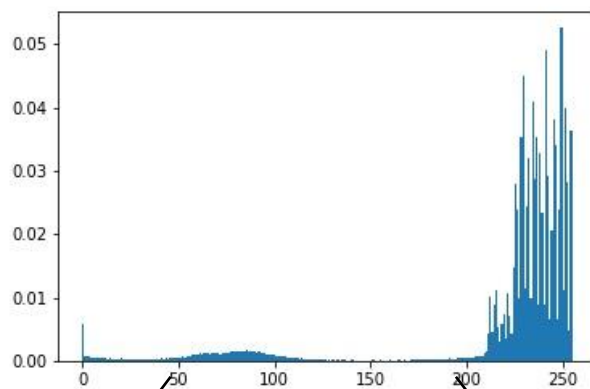


初期閾値が適切でない場合、局所解に収束することがある

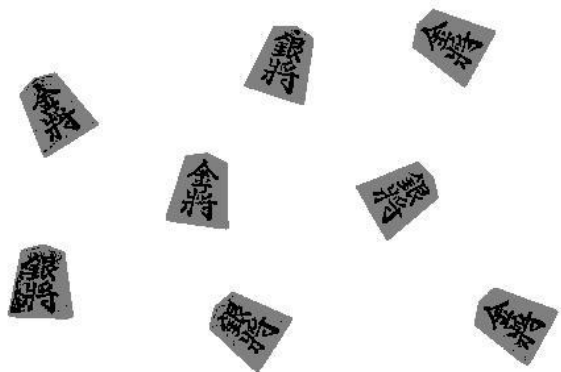


閾値選択

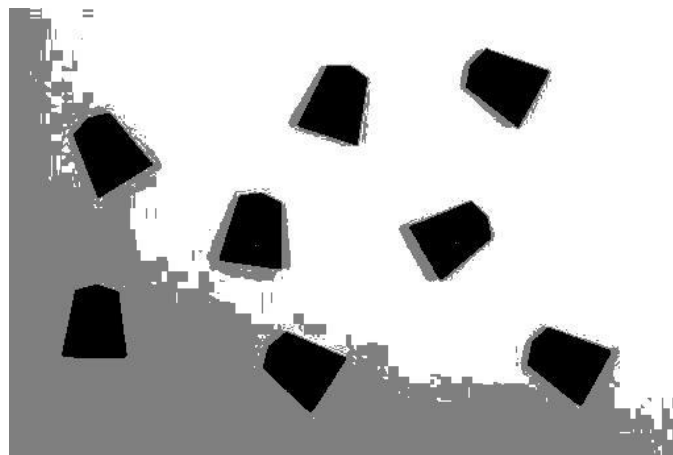
反復法



$$t_1 = 42, t_2 = 68$$



$$t_1 = 186, t_2 = 253$$



多値化

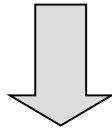
動的計画法による閾値選択

$$\sigma_B^2 = \sum_{i=1}^n \frac{\omega_i}{\omega_T} (M_i - M_T)^2 = \sum_{i=1}^n g(t_{i-1}, t_i)$$

$$G(1, t_1) = g(t_0, t_1) \quad (1 \leq t_1 \leq L - n + 2)$$

$$G(k, t_k) = \max_{k \leq t_{k-1} < t_k} \{G(k-1, t_{k-1}) + g(t_{k-1}, t_k)\}$$

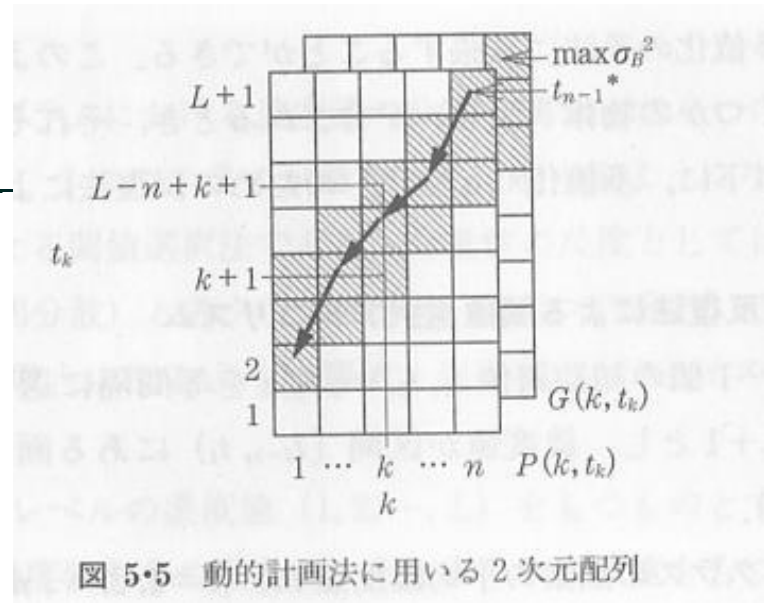
$G(k, t_k)$ の最大値を与える t_{k-1}^* を $P(k, t_k)$ に格納



σ_B^2 の最大値は $G(n, L+1)$

最適閾値は $t_n^* = L+1$

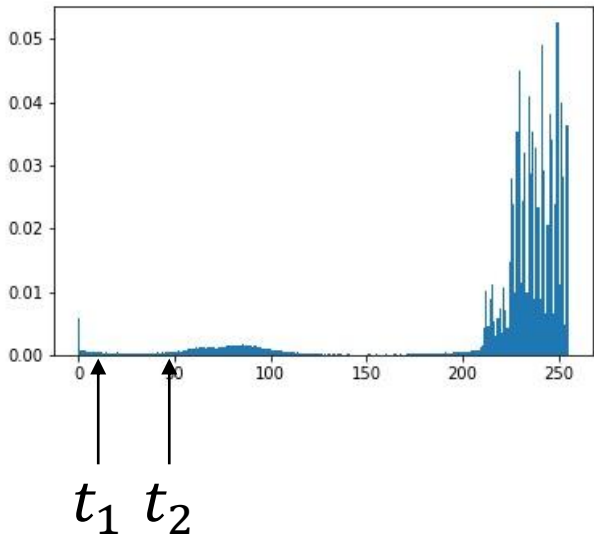
$$t_{k-1}^* = P(k, t_k^*) \quad (2 \leq k \leq n)$$



$$G(1,t_1) = g(t_0,t_1) \quad (1 \leq t_1 \leq L-n+2)$$

$$G(k,t_k) = \max_{k \leq t_{k-1} < t_k} \{G(k-1,t_{k-1}) + g(t_{k-1},t_k)\}$$

$$\sigma_B^2 = \sum_{i=1}^n \frac{\omega_i}{\omega_T} (M_i - M_T)^2 = \sum_{i=1}^n g(t_{i-1},t_i)$$

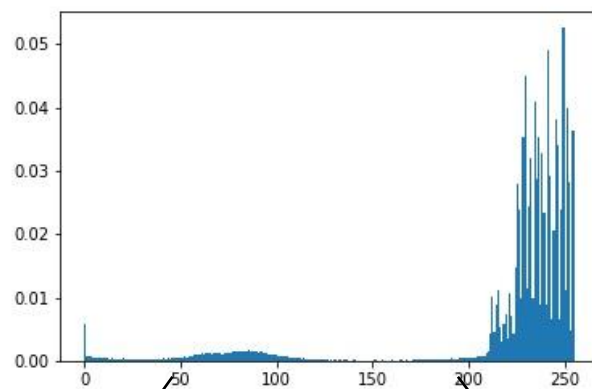


t_k	$G(k,t_k)$	
256	—	
255	$g(0,255)$	
254	$g(0,254)$	
	\vdots	
3	$g(0,3)$	
2	$g(0,2)$	$g(0,1)+g(1,2)$
1	$g(0,1)$	—
	1	2 k

$$\max(g(0,1) + g(1,3), g(0,2) + g(2,3))$$

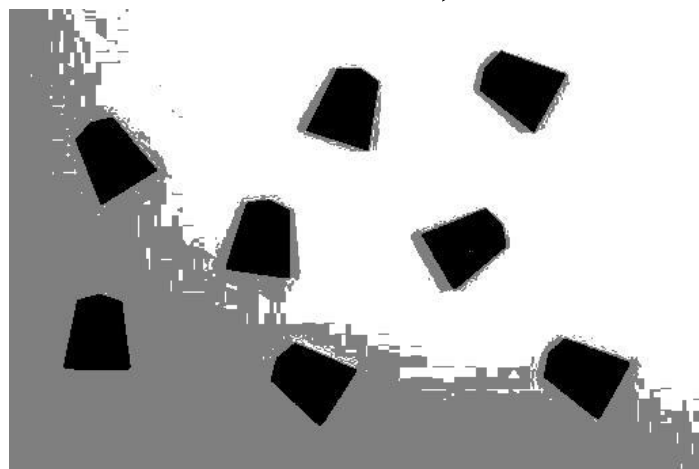
$$\max(g(0,1) + g(1,4), g(0,2) + g(2,4), g(0,3) + g(3,4))$$





$$n = 3$$

$$t_1 = 146, t_2 = 235$$



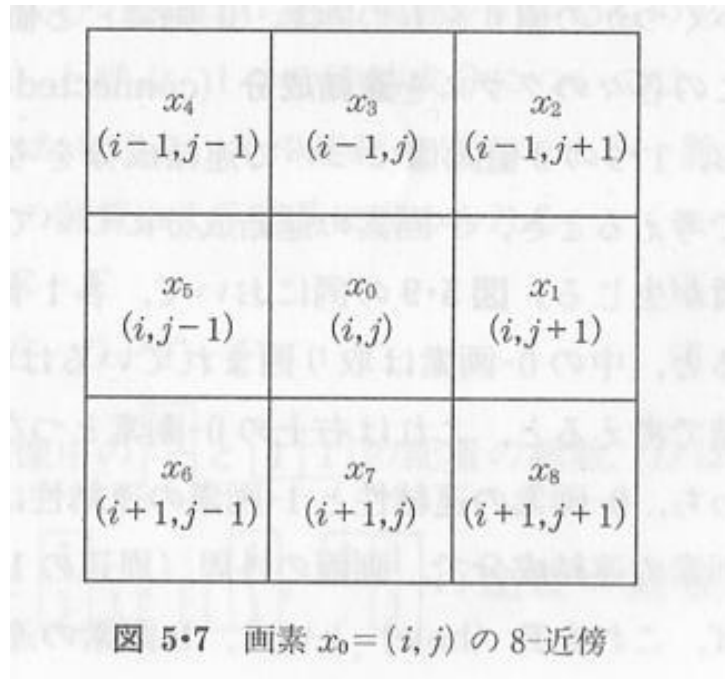
$$n = 4$$

$$t_1 = 53, t_2 = 155, t_2 = 235$$

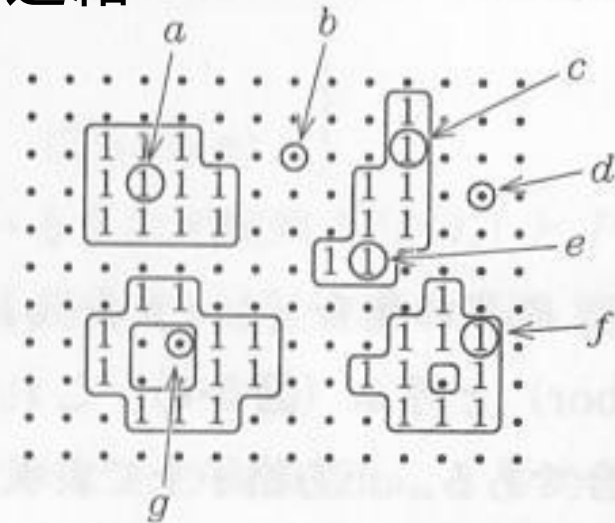


2値画像の連結性

近傍と隣接



連結



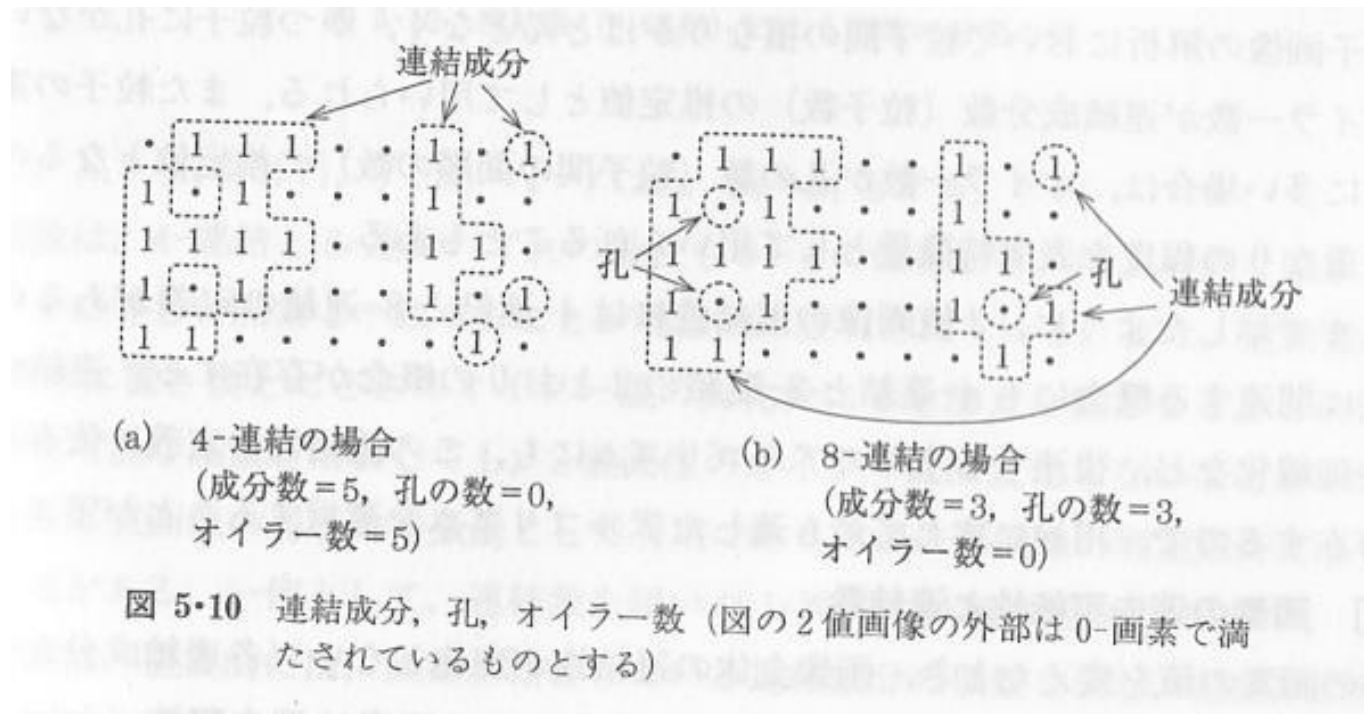
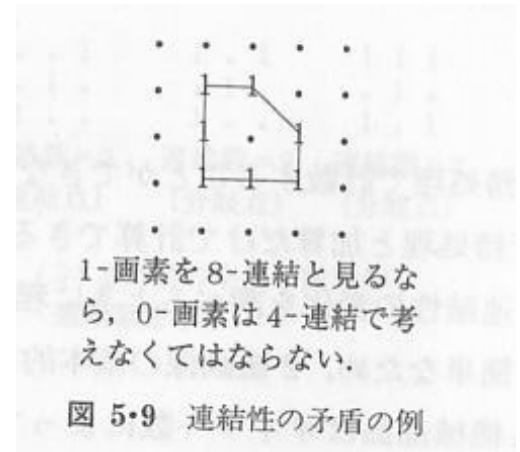
cとe, bとdは連結, 他の
同じ値をもつ画素の対は
非連結

図 5・8 画素の連結



2値画像の連結性

連結成分



連結成分の解析

近傍処理によるラベリング(4-近傍型)

同じ連結成分に属するすべての画素に同じラベル(番号)を、異なる連結成分には異なるラベルを割り当てる

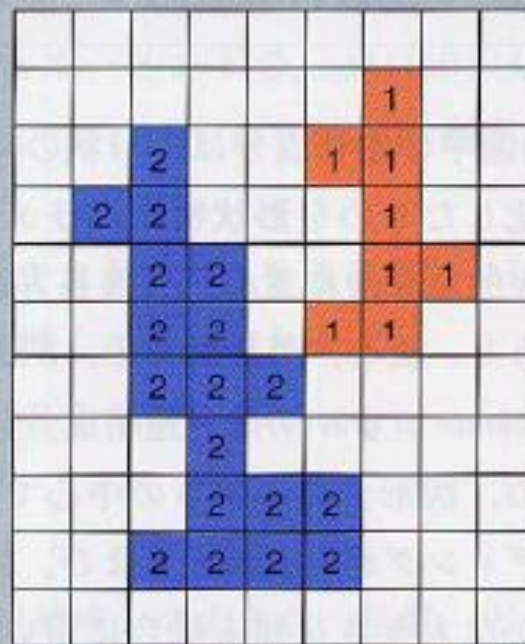
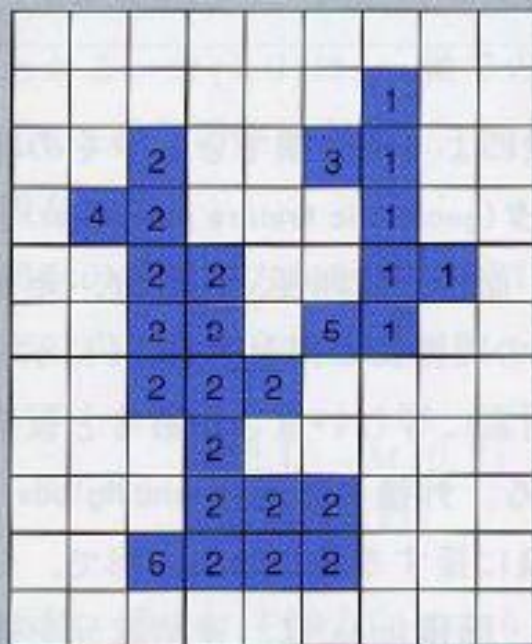
Step1: x_0 に対し、近傍画素 x_3, x_5 が

- どちらもラベル付けされていなければ、新規ラベルを与える。
- 同一ラベルもしくは一方にのみラベルが与えられていれば、そのラベルを与える。
- 異なるラベルが与えられていれば、小さい方のラベルを与え、ラベルの等価性を表すルックアップテーブルに2つのラベルの等価性を記録

Step2: ルックアップテーブルを用いて、画像中のラベルを等価なラベルの中で最小のラベルに付け替える。



近傍処理によるラベリングの例



ルックアップテーブル1

1 ⇔ 3,

4 ⇔ 2,

5 ⇔ 1,

6 ⇔ 2,



テーブルを整理

ルックアップテーブル2

1 ⇔ 3 ⇔ 5,

2 ⇔ 4 ⇔ 6



連結成分の解析

ラン解析によるラベリング(4-／8-近傍型)

同じ連結成分に属するすべての画素に同じラベル(番号)を、異なる連結成分には異なるラベルを割り当てる

- Step1: ラスタ走査で最初に出会う1-画素を見つけ、その画素を含む水平ランを新規ラベルでラベル付けする。
- Step2: ラベル付けしたランの上下に(4-/8-)連結するすべてのランの左端の座標をスタックに記録する(既にラベル付けされているランは無視する)。
- Step3: スタックが空ならStep1に戻る。空でなければ、スタックの先頭から座標を取り出して、その画素を含む水平ランをラベル付けした後Step2に戻る。



ラン解析によるラベリングの例

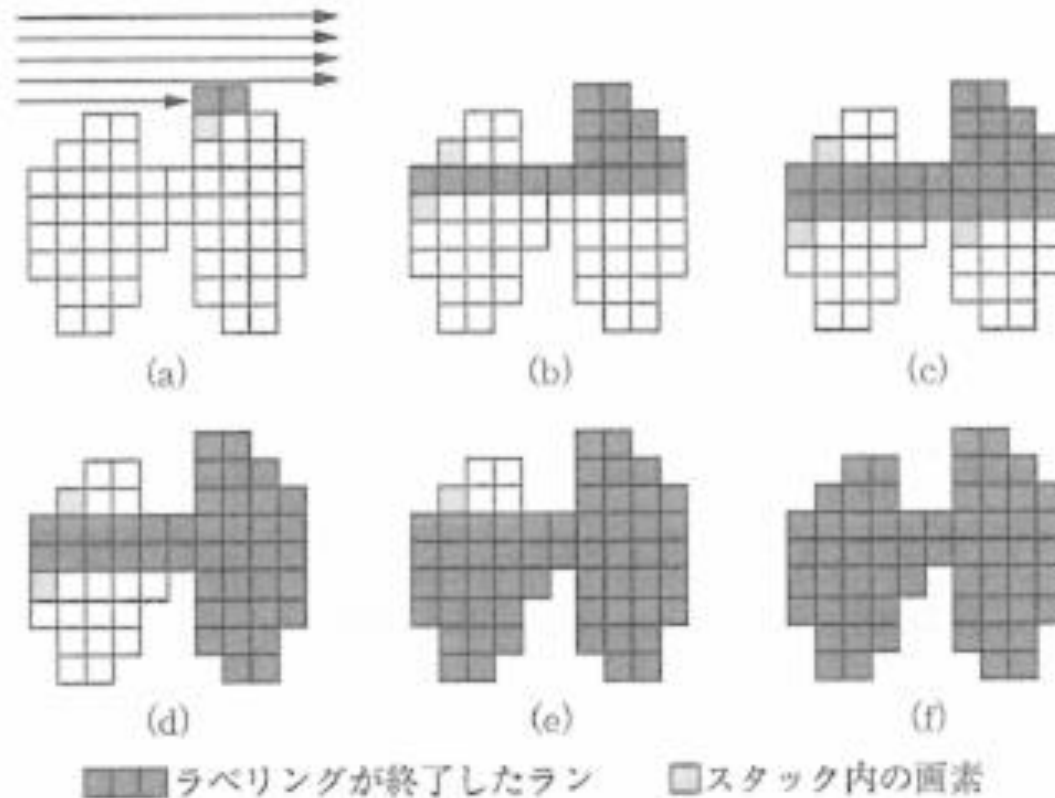


図 5・16 ラン解析アルゴリズムによるラベリングの例



連結成分の解析

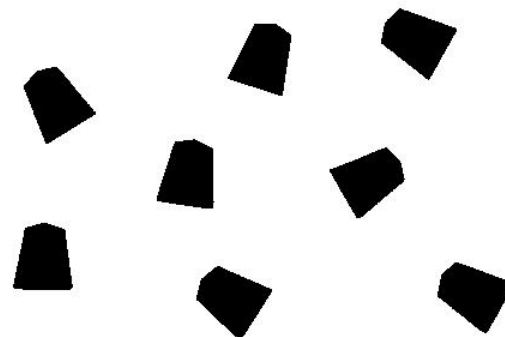
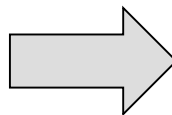
境界追跡

- Step1: 入力画像をTV式ラスタ走査し、左側に0-画素が隣接する1-画素 a_1 を探し、最初の境界画素とする。ただし、既に追跡済みの1-画素は無視する。また、 a_1 の8近傍がすべて0-画素の場合はラスタ走査を続行する。ラスタ走査が終了したら境界追跡を終了する。
- Step2: 境界画素 a_n ($n = 1, 2, \dots, m$) の8近傍において、直前の境界画素 a_{n-1} から順に反時計周りに画素の値を調べ、次の境界画素となる1-画素 a_{n+1} を求める。
ここで、ある m において、 $a_m = a_1$ となったら1本の境界追跡を終了してStep1に戻る。 a_1, \dots, a_{m-1} が1本の境界となる。
また a_1 の直前の境界画素 a_0 は a_1 の左隣の画素と考える。

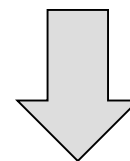




2値化



ラベリング



境界追跡の例

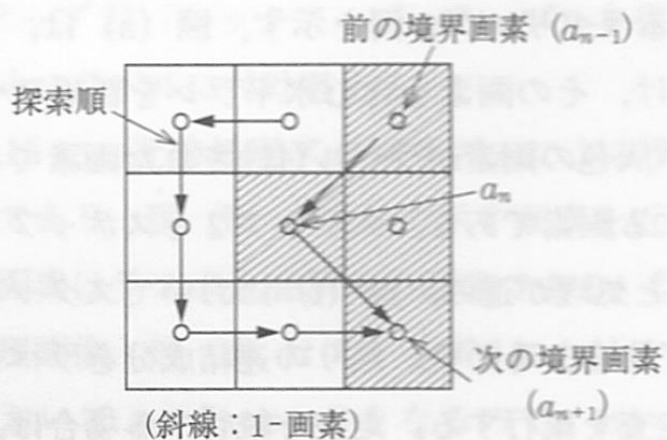


図 5・17 境界追跡における次の境界画素の見つけ方

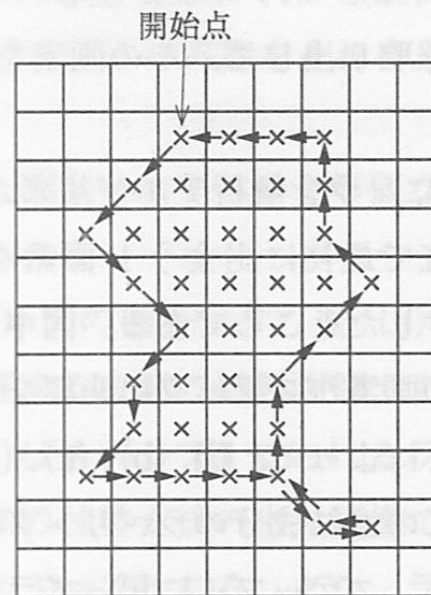
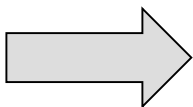


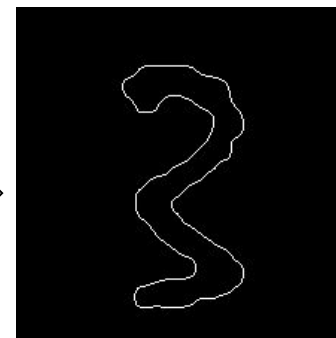
図 5・18 境界追跡の例



2値化



境界



収縮と膨張

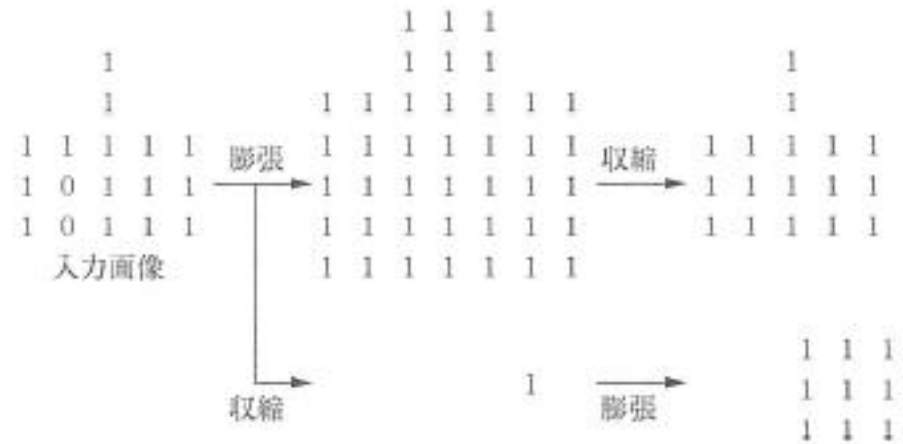


図 5・22 収縮と膨張の例 (8-近傍型, 図の2値画像の外部は0-画素で満たされているものとする)

収縮: 連結成分の境界画素をすべて取り除いて
1画素分小さくする処理

$$g_{ij} = \begin{cases} 0: \text{画素}(i,j) \text{あるいはその4(or 8)近傍のいずれかが0-画素のとき} \\ 1: \text{その他のとき} \end{cases}$$

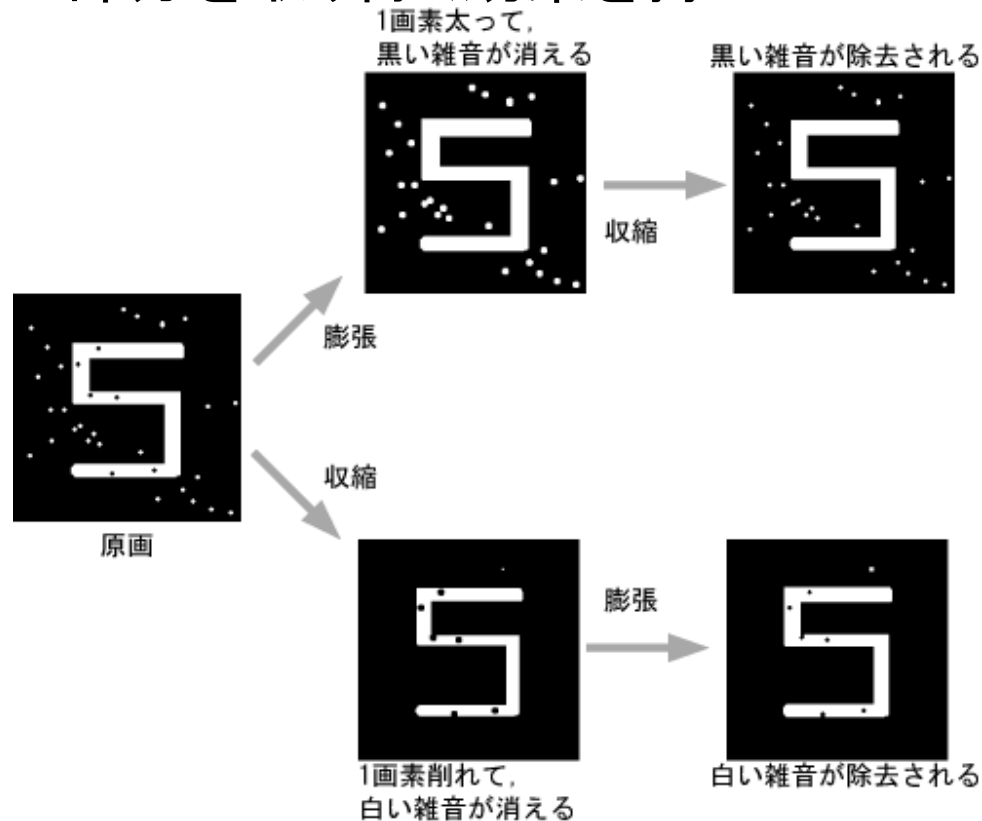
膨張: 連結成分を1画素分大きくする処理

$$g_{ij} = \begin{cases} 1: \text{画素}(i,j) \text{あるいはその4(or 8)近傍のいずれかが1-画素のとき} \\ 0: \text{その他のとき} \end{cases}$$

収縮と膨張

収縮→膨張: 2値画像中の小成分や幅の狭い部分を取り除く効果をもつ

膨張→縮小: 連結成分の小さな孔や幅の狭いくぼみの部分を取り除く効果を持つ



開放と閉鎖

開放: n 回収縮処理を行った後、 n 回膨張処理を行う

閉鎖: n 回膨張処理を行った後、 n 回収縮処理を行う



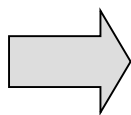
濃淡画像に対する領域分割

領域統合法

(判定基準: 濃度差)

- ① 画像をラスタ走査し、未分類の注目画素を求める
- ② 注目画素の周囲(4or8近傍)で未分類の画素が、同じ画素値を持てば1つの領域として統合する。
- ③ 新たに統合された画素を注目画素として②の処理を行う。新たに統合された画素がなければ①へ戻る。
- ④ それ以上領域が広げられなくなれば終了
- ⑤ 同じラベルを持つ画素の平均画素値を求める。
- ⑥ 平均値の差が最小となる隣接領域を結合する。
- ⑦ ⑤、⑥の処理を繰り返す。

(平均値の差の上限を決める)

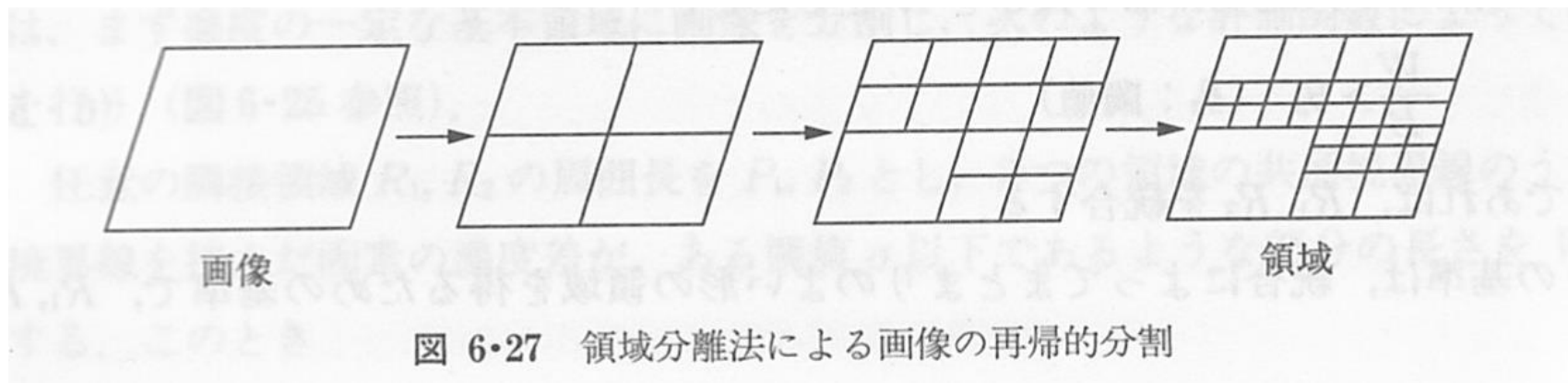


適当な段階で停止させないと、
最終的に画像全体が1つの領域になる



濃淡画像に対する領域分割

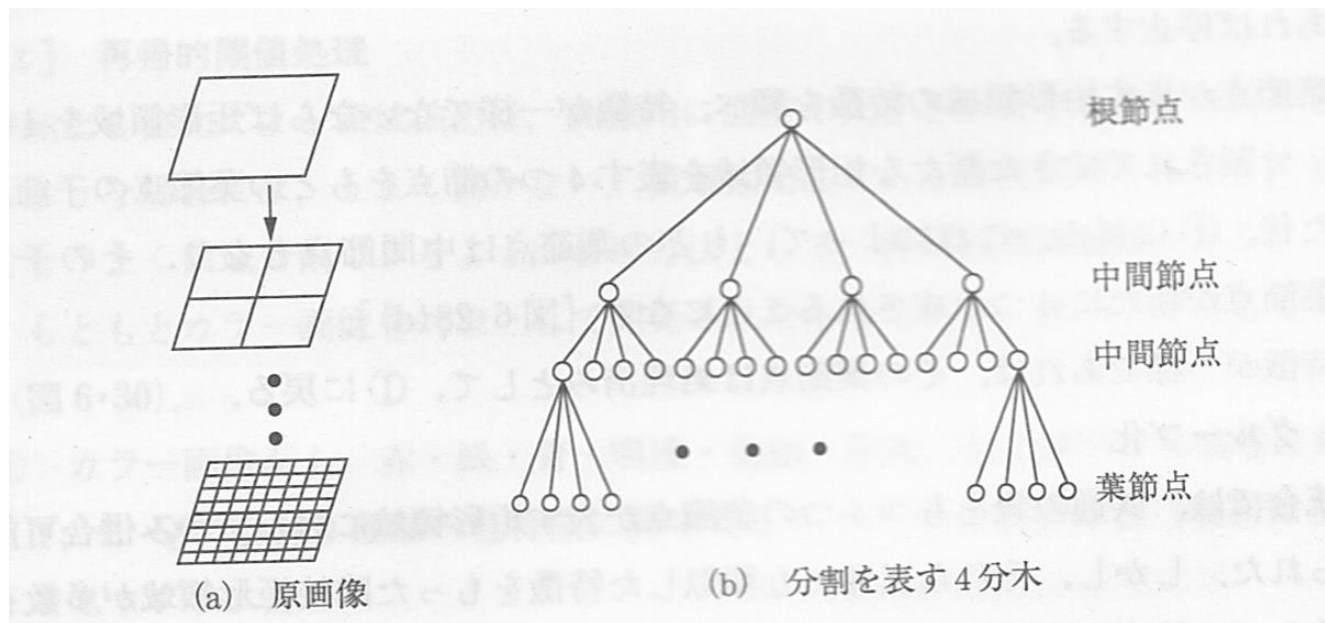
分離統合法：画像を再帰的に小領域に分割し、
特徴の一様な領域を求める。



濃淡画像に対する領域分割

(a) 初期分割

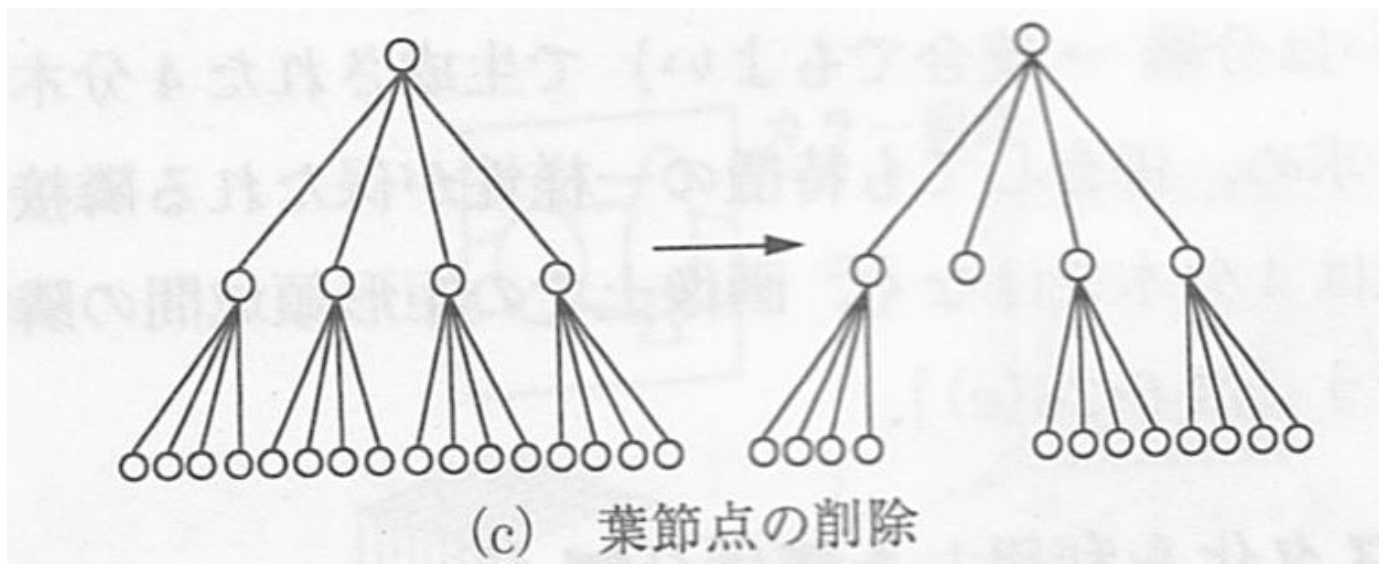
同じ大きさをした初期矩形領域群を生成。



濃淡画像に対する領域分割

(b) 結合

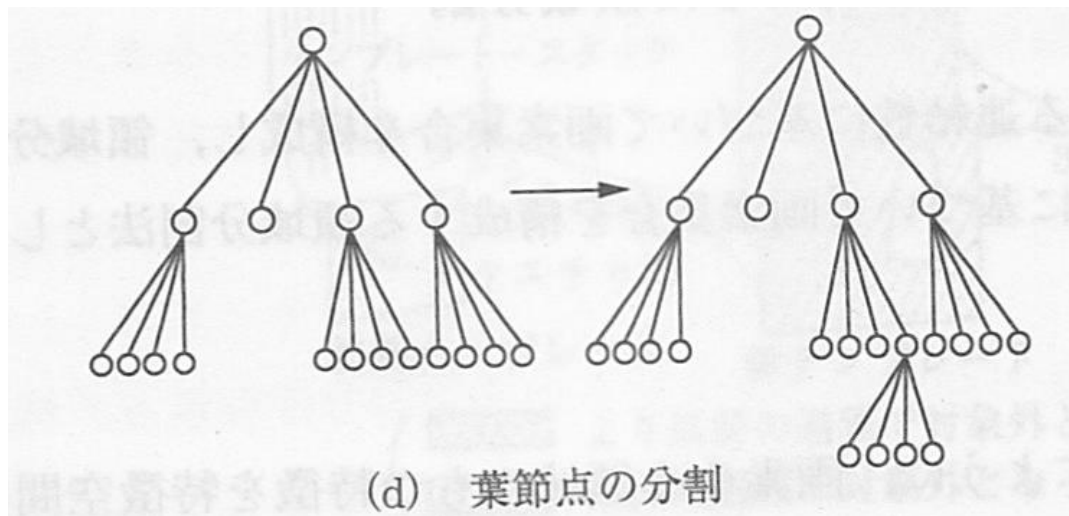
- ① 共通の親節点を持つ4つの葉節点を探す。
- ② 各葉節点を表す矩形領域の特徴が一様であるか調べる。
- ③ 一様であれば、4つの葉節点を削除し①に戻る。



濃淡画像に対する領域分割

(c) 分離

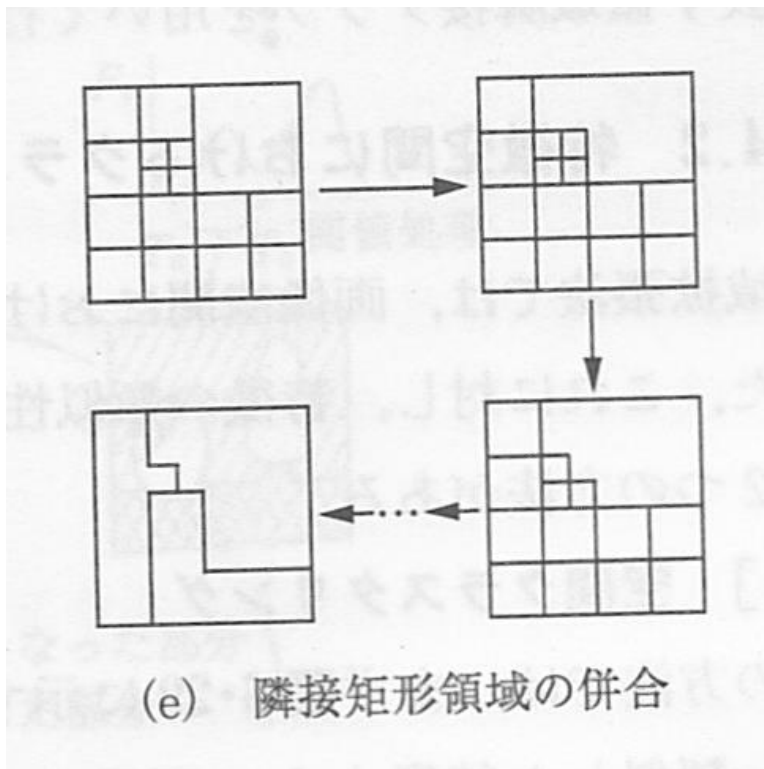
- ① 処理されていない葉節点を探す。
- ② 葉節点が表す矩形領域の特徴が一様でない場合、矩形領域を4分割し、分割されてできた新たな矩形領域をもとの葉節点の子節点とし、①に戻る。
- ③ 特徴が一様であれば、その葉節点処理済みとして①に戻る。



濃淡画像に対する領域分割

(d) グループ化

統合→分離で生成された4分木の葉節点が表す矩形領域間の隣接関係を求め、併合しても特徴の一様性が保たれる隣接矩形領域をすべて併合する。



画像に対する領域分割

空間クラスタリング

(判定基準: 特徴の類似性)

- ① 類似した特徴をもつ画素や小領域の集合を特徴空間におけるクラスタリングにより求め、各画素にそれが属するクラスタを表すラベルを与える。
- ② 連結成分のラベル付けを行う。

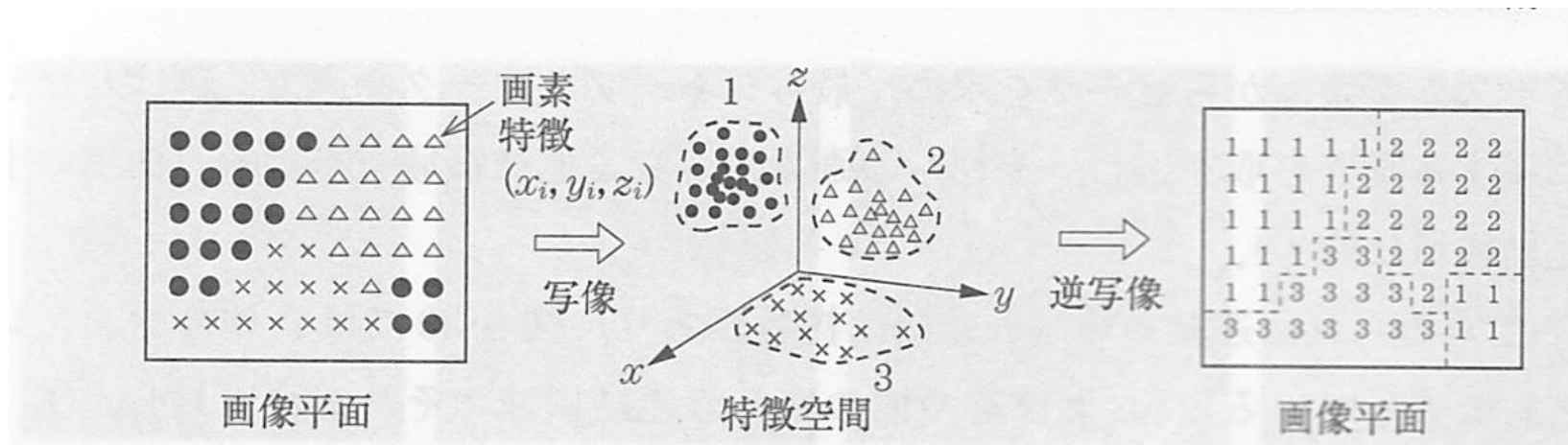


図 6・29 空間クラスタリング



クラスタリング例: K平均法

- $\mathbf{x}_p (p = 1, \dots, N)$ をK個のクラスタに分ける

Step 1: K個のクラスタの中心 $\boldsymbol{\mu}_k$ をランダムに設定する.

Step 2: 各パターンを最も近い中心に割り当てる.

$$h_p = \arg \min_k \|\mathbf{x}_p - \boldsymbol{\mu}_k\|^2$$

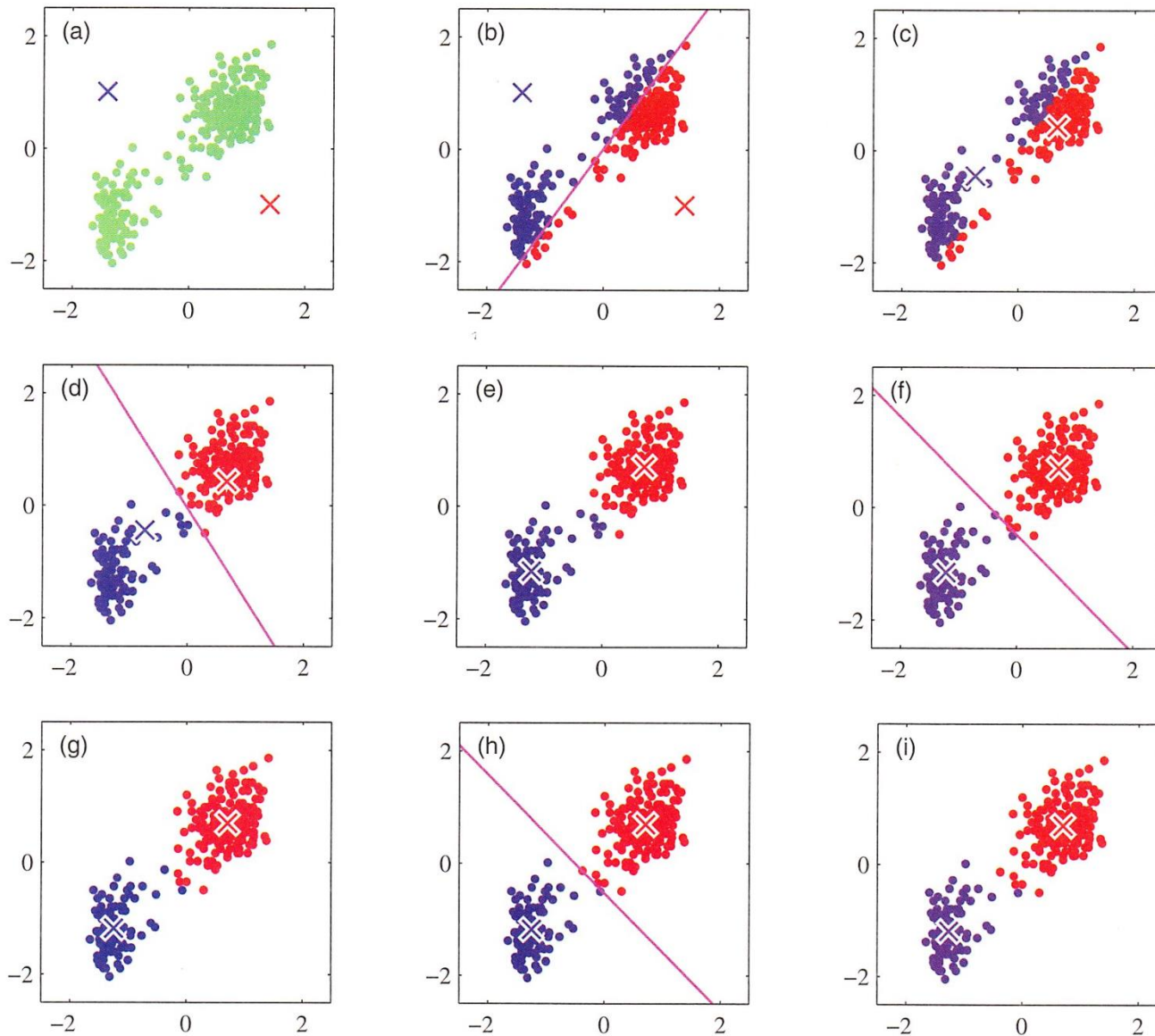
Step 3: クラスタごとに中心を計算しなおす.

$$\boldsymbol{\mu}_k = \frac{\sum_{h_p=k} \mathbf{x}_p}{\sum_{h_p=k} 1} \begin{array}{l} \longrightarrow \text{クラスタ内のパターンの和} \\ \longrightarrow \text{クラスタ内のパターン数} \end{array}$$

すべてのクラスタ中心が変化しなければ終了.

それ以外はStep 2へ戻る.

クラスタリング例: K平均法



線の検出

ハフ変換

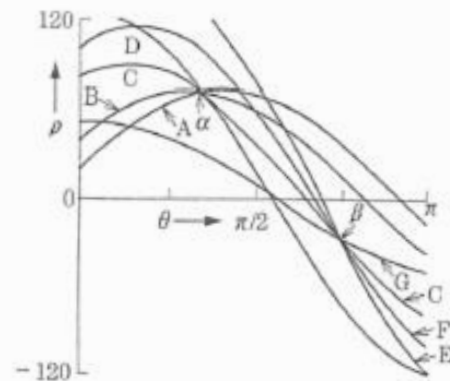
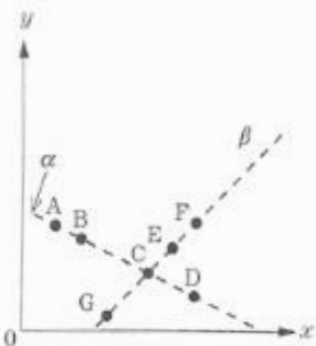
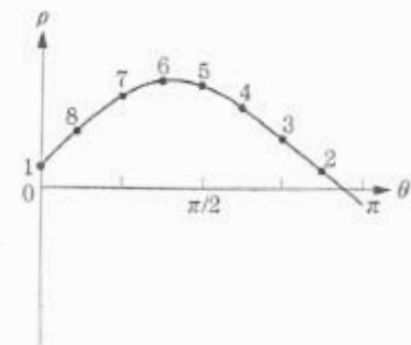
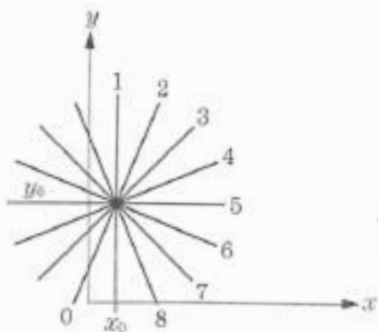
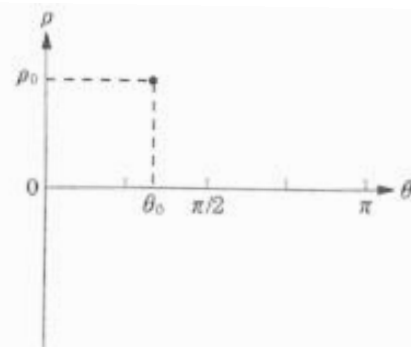
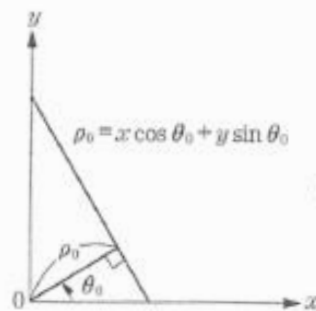
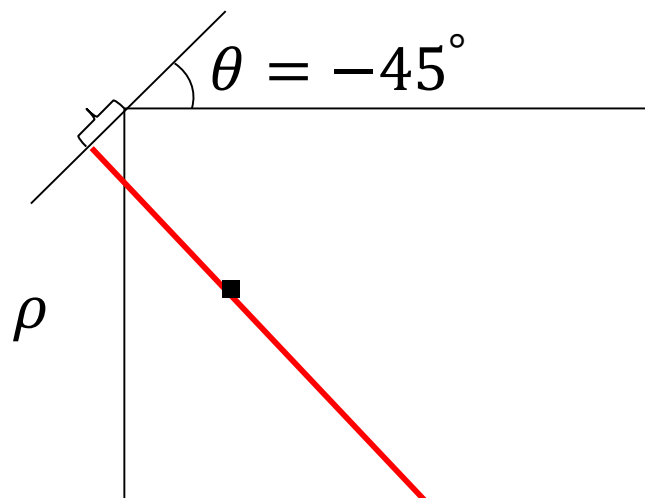
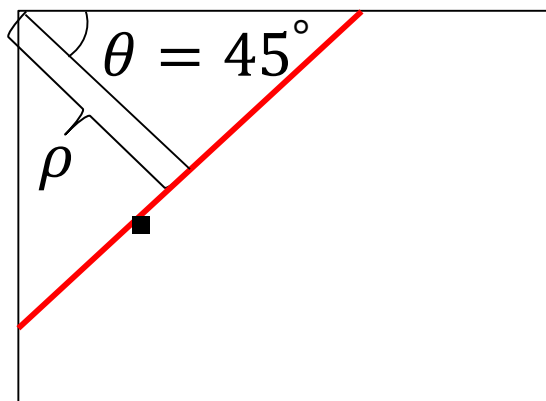
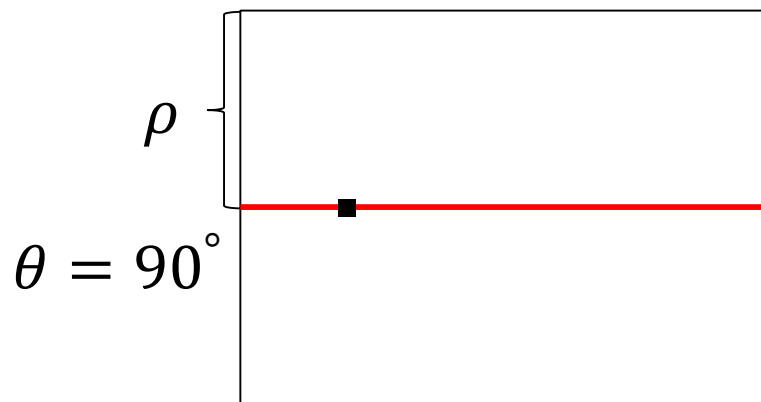
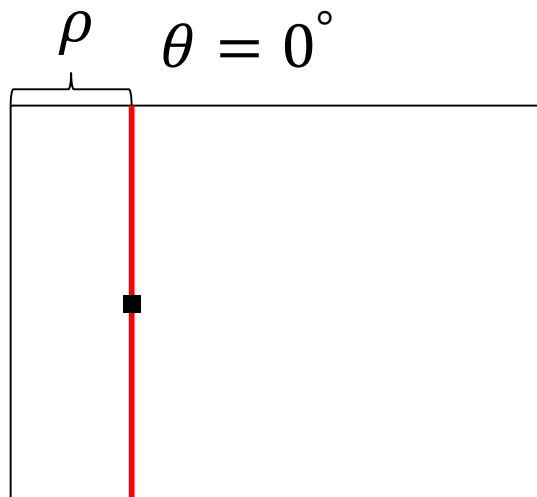


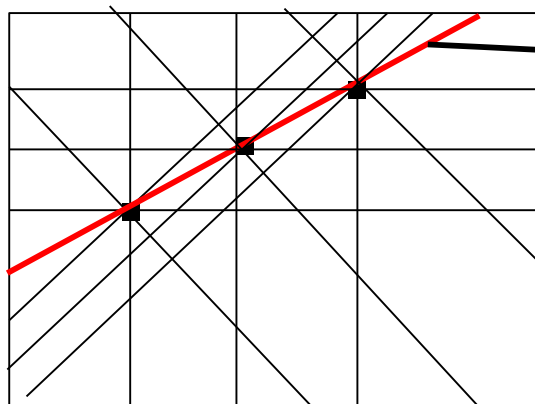
図 6-21 ハフ変換の原理



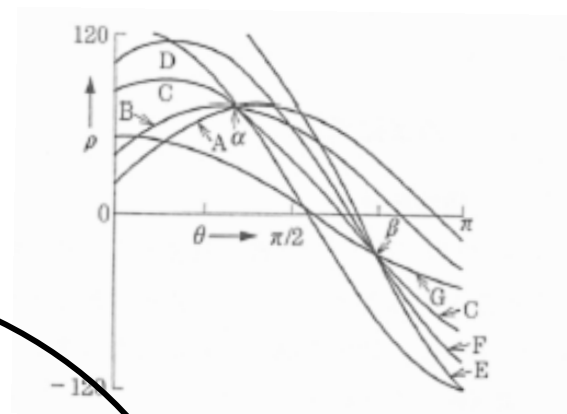
線の検出



線の検出



全点で投票



ρ

θ

	-90		...			90
-max	0	0		0	0	0
	0	0		3	0	0
⋮						1
	0	0		0	0	0
max	0	0		0	0	0



演習課題

※C言語もしくはpythonで実装すること

以下の1、及び2 or 3の処理のプログラムを作成し、与えられた画像により動作確認せよ。

1. 判別分析法と反復法による画像の2値化 (test1.jpg)
2. 動的計画法による画像の多値化 (test1.jpg)
3. ハフ変換による線の検出 (test2.jpg)

- 画像処理部分において、opencvなどの画像処理ライブラリなどは**使わない**こと！
- プログラムの説明(該当するソースコードと共に)、各処理の実行結果、工夫点、感想などを**レポートにまとめよ**。
- レポート(PDF形式)と**ソースコード**をCLEから提出して下さい。
- 提出期限:6月2日(火)