

画像情報処理1

第1-8回

教員: 陳 延偉

chen@is.ritsumei.ac.jp

クリエイションコア4階知的画像処理研究室

講義内容

- 第1週 デジタル画像処理について
- 第2週 デジタル画像とその表現方法
- 第3週 畳み込み演算とフィルタリング
- 第4週 フーリエ変換と周波数領域における画像処理
- 第5週 画像復元と画像再構成
- 第6週 幾何学的変換線形変換、アフィン変換
- 第7週 2値化と2値画像処理
- 第8週 確認テスト(60分)と解説(30分)

評価方法

- 授業前半の内容について確認テスト 40%
授業後半の内容について確認テスト 40%
プログラミングレポート提出 20%
- 【BCPLレベル1~2】
第8回目と第15回目に確認テストを対面で実施する
【BCPLレベル3~4】
第8回目と第15回目をレポート試験とする

教科書

デジタル画像処理 CG-ARTS

参考書

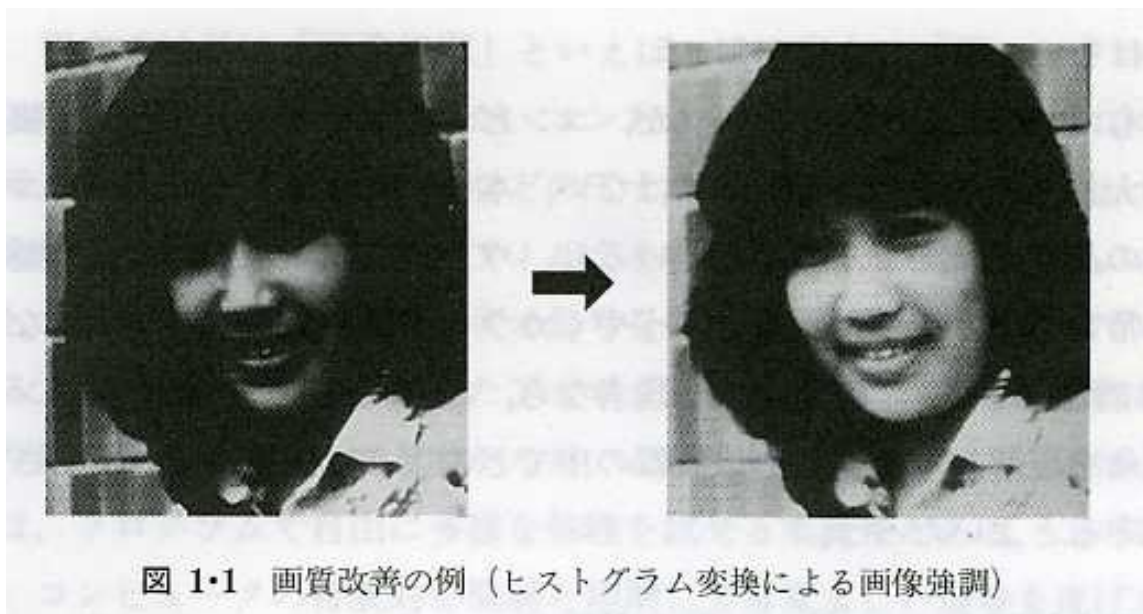
田村秀行編著、「コンピュータ画像処理」、
オーム社

画像処理

1. 見やすくする→画質の改善

入力:画像; 出力:画像

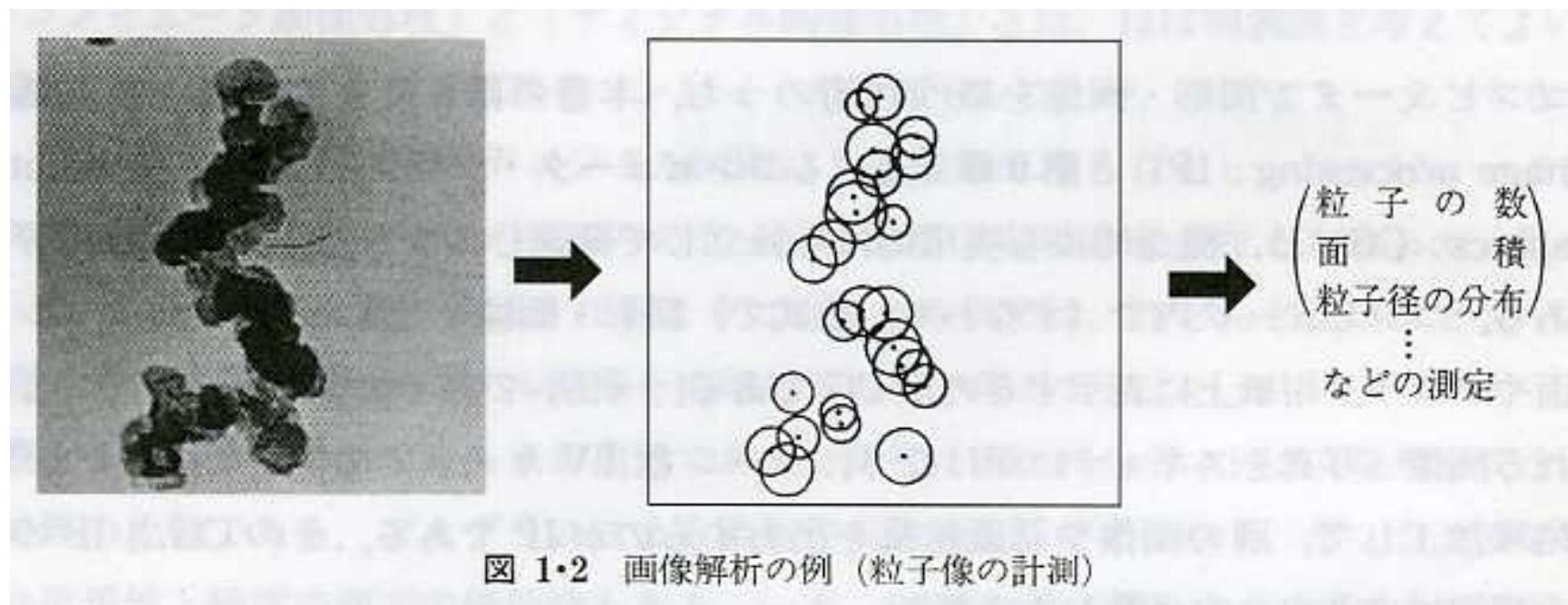
処理内容:ノイズ除去、画像強調、画像復元など



2. 情報を抽出・認識→画像解析、パターン認識

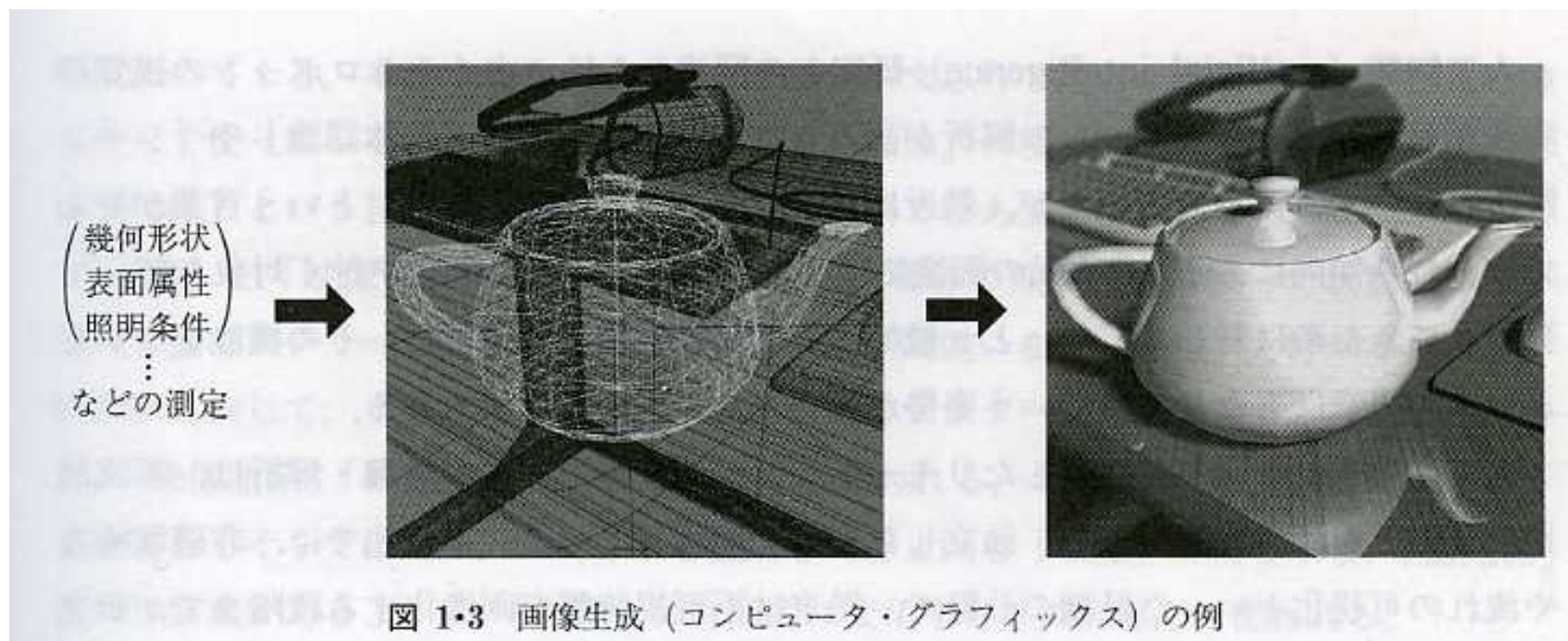
入力: 画像; 出力: 必要な情報、パターン(画像)

処理内容: エッジ検出、特徴抽出、画像分類



3. 画像の人的な生成→コンピュータグラフィックス

入力:関数や点群; 出力:画像or図形



画像処理は特に1と2の処理を意味する。

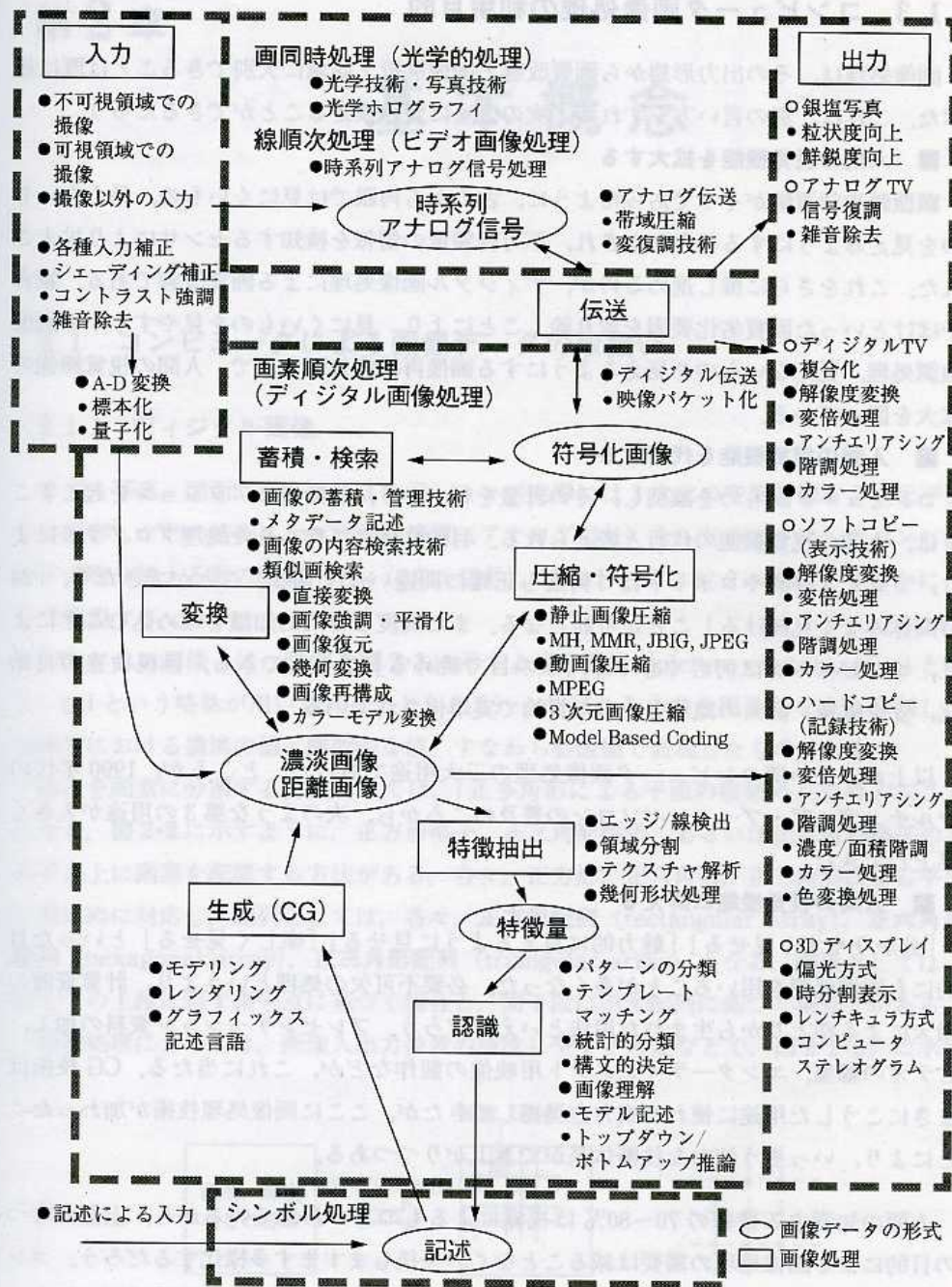
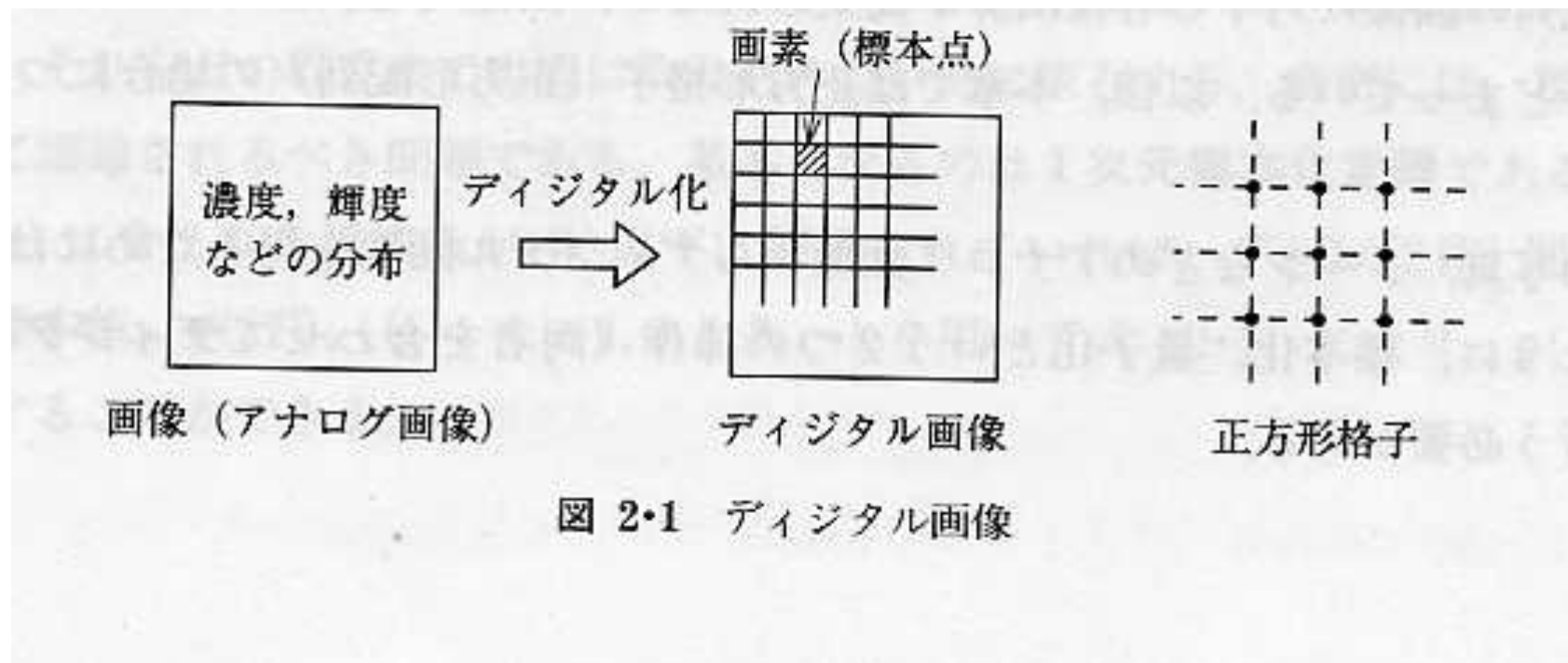


図 1・4 画像処理分野技術マップ

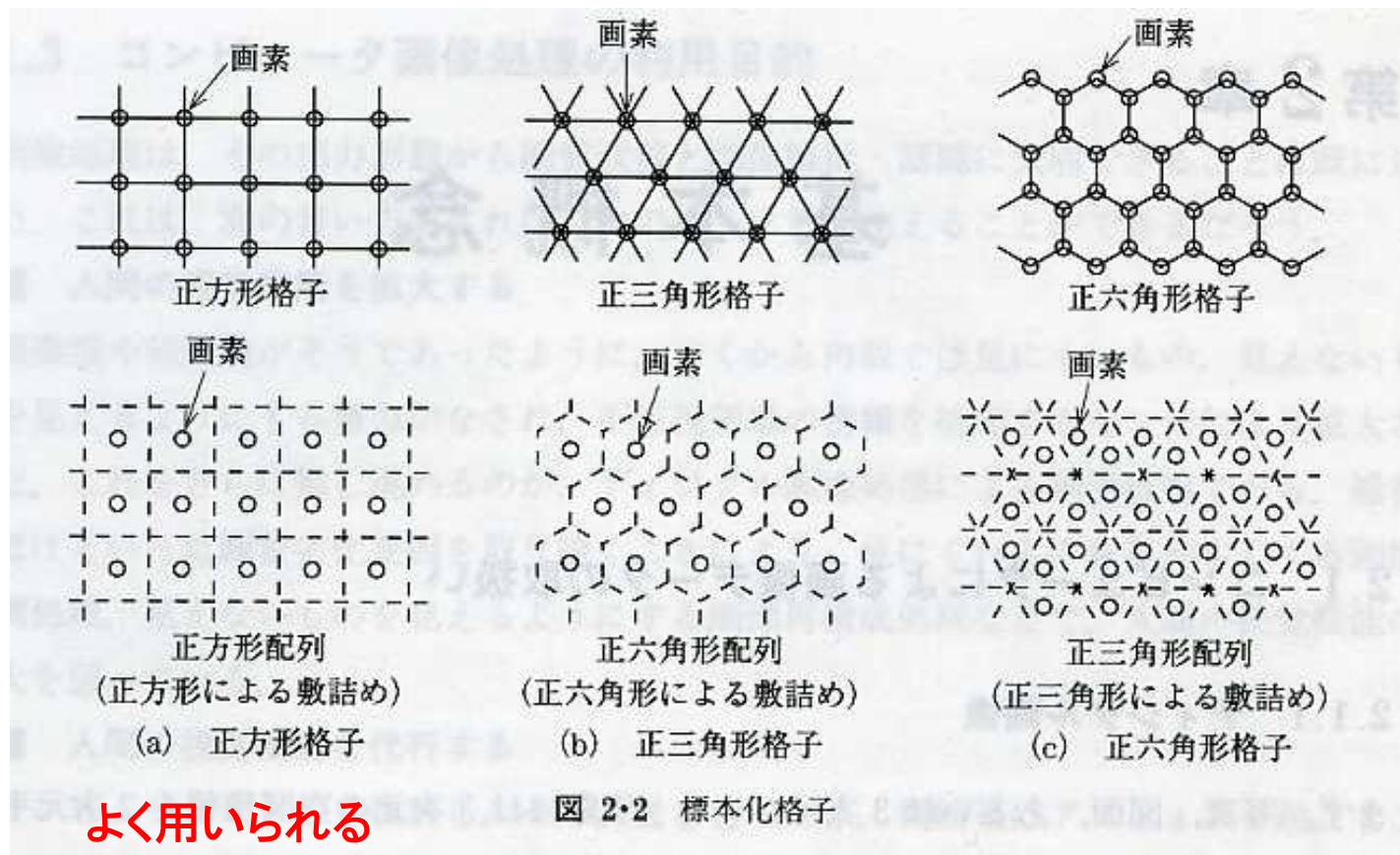
デジタル画像とは？

- 画素(ピクセル)の集合
- 画素:画像の中の「小さな正方形領域」



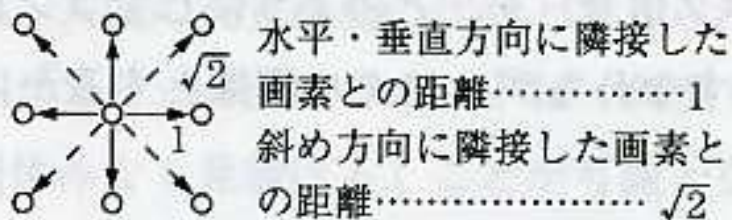
画素に分割する方法

- 正多角形による平面の敷詰め
- 画像処理では、**正方形格子(正方形配列)**が最もよく用いられる

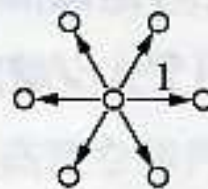


隣接した画素との距離

- 正方形格子では、水平・垂直方向に隣接した画素の距離を1として、斜め方向も隣接した画素との距離は $\sqrt{2}$ となる



(a) 正方形格子（正方形配列）の場合

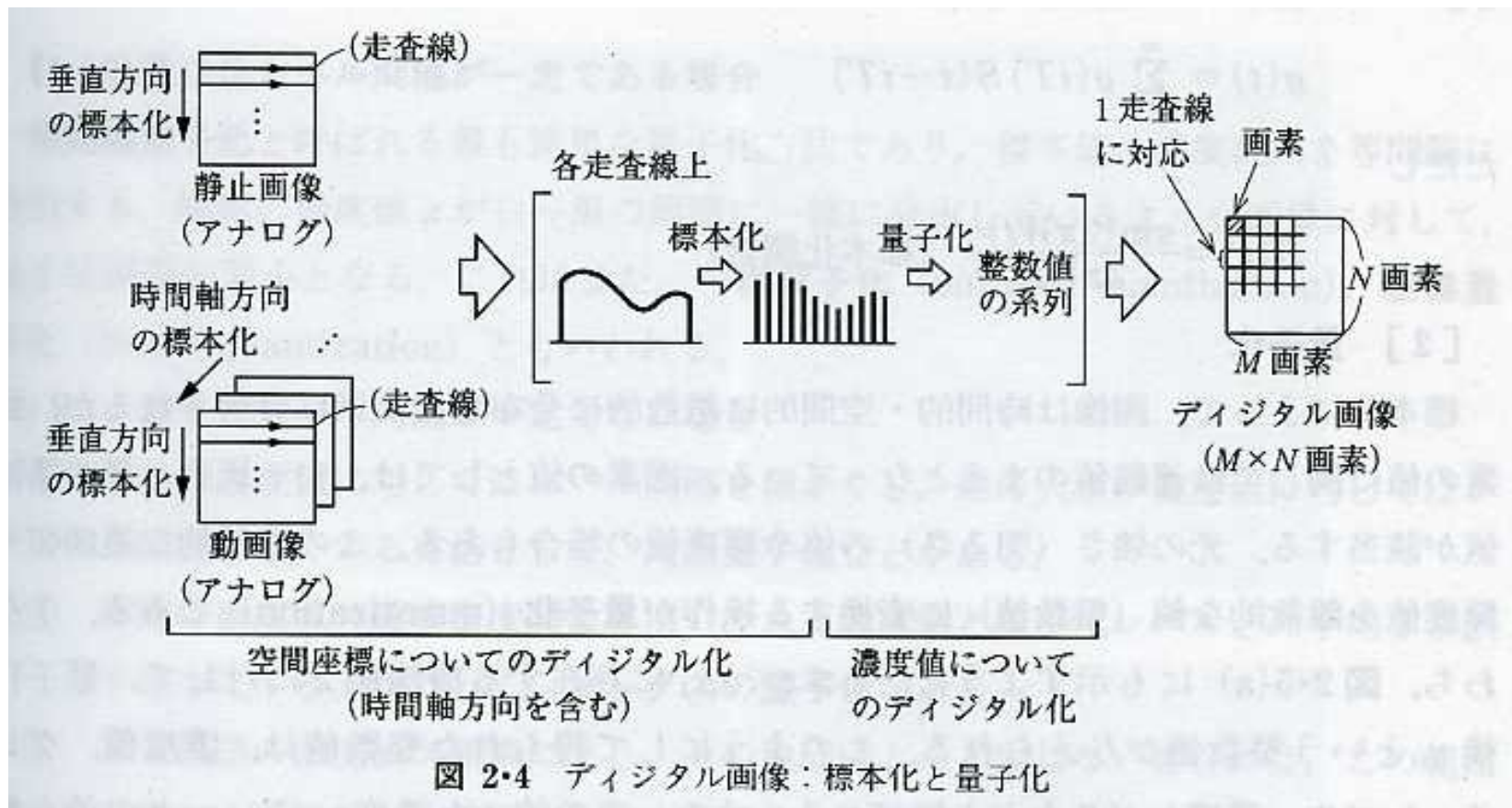


(b) 正三角形格子（正六角形配列）の場合

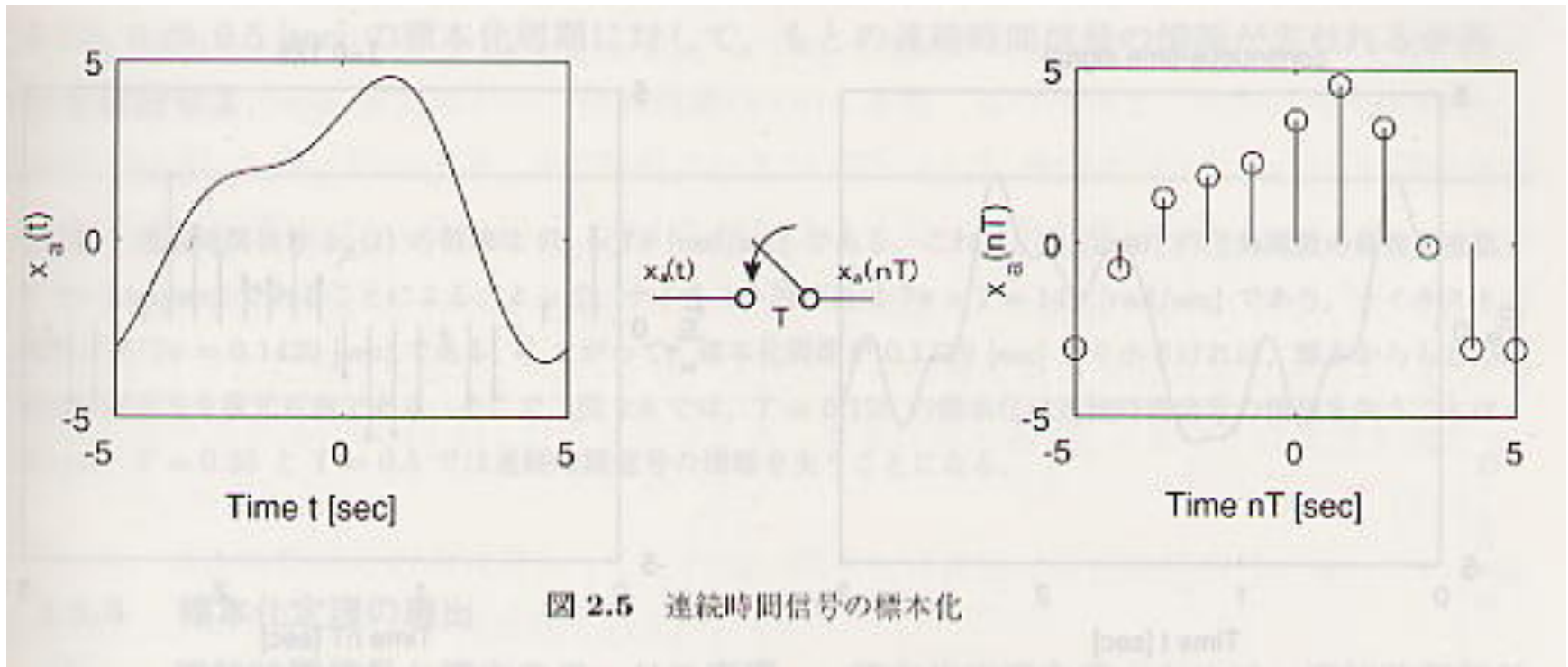
図 2・3 隣接した画素との距離

写真などのアナログ画像からデジタル画像に変換するには

- 標本化: 横軸(空間、時間)を離散的な点の集合に変換する
- 量子化: 縦軸(濃度、強度)を離散的な数(整数値)に変換する

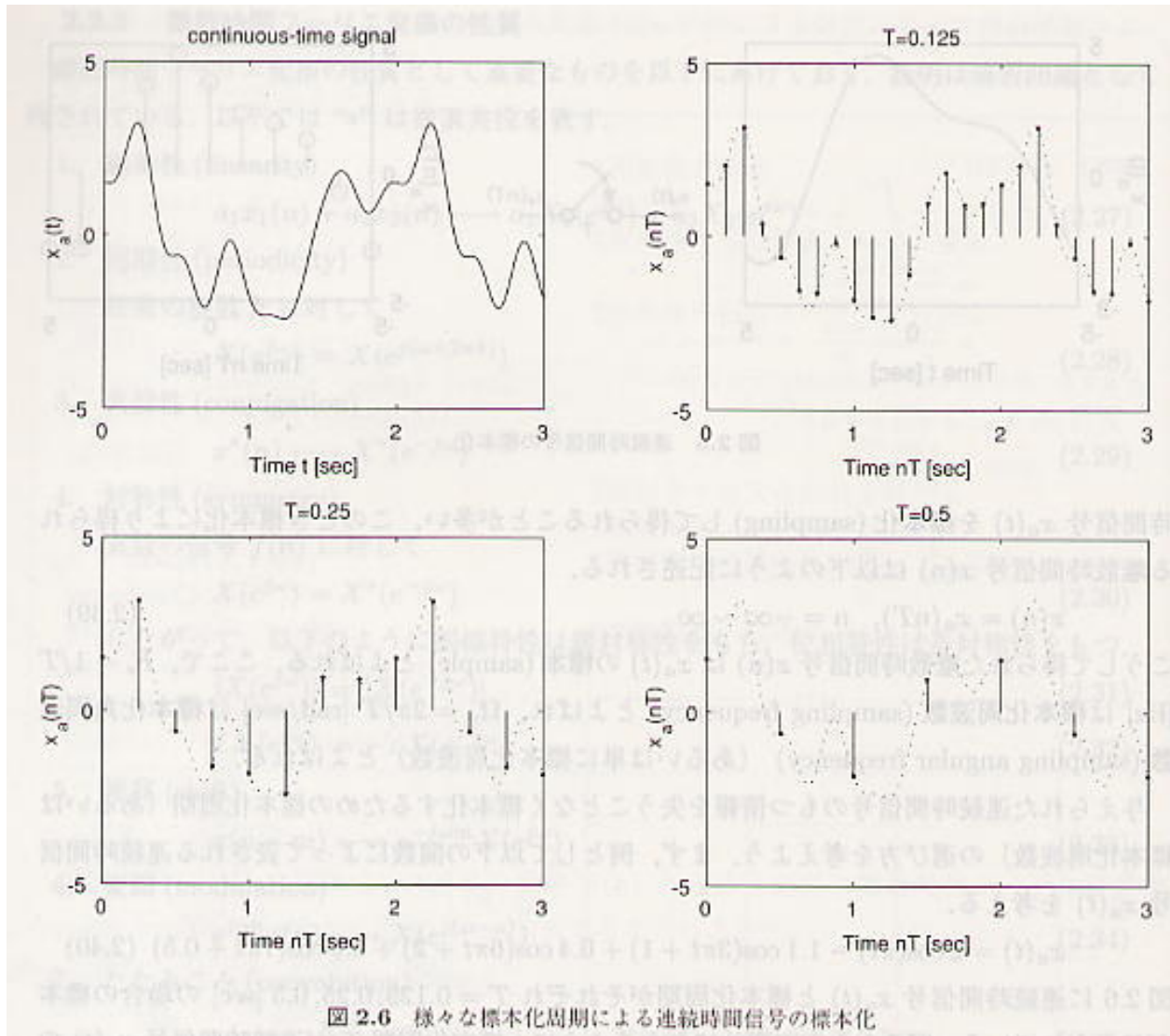


1次元の標本化 (Sampling) 例



$$X(nT) = \sum_{n=0} x(t) \delta(t - nT)$$

標本化定理



W: 最大周波数

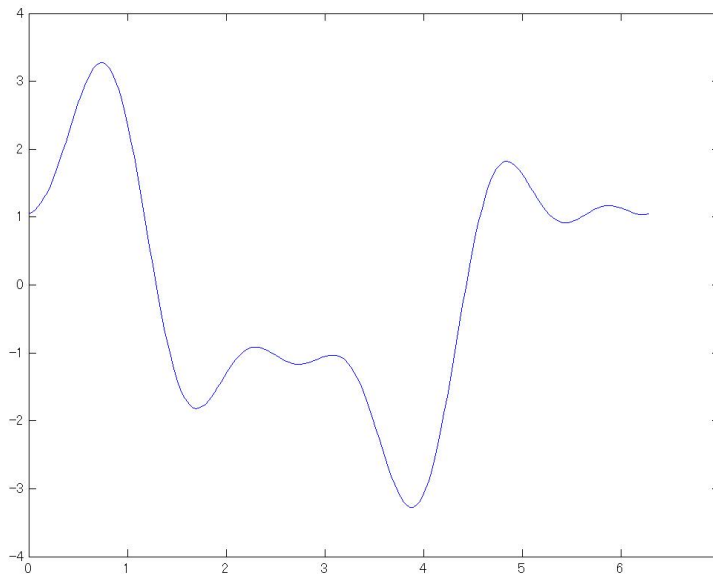


標本化間隔
 $T \leq 1/(2W)$

課題

- 下記の信号に対して、最大な標本化間隔(ナイキスト間隔)を求めなさい。

$$x(t) = 2 \cos(\pi t) - 1.3 \cos(3\pi t + 1) + 0.6 \cos(5\pi t + 2)$$



解答

最大角周波数: 5π
 $W = 5\pi/2\pi = 2.5$ (1/s)

$T = 1/(2W) = 0.2$ (s)

係

256x256

128x128

64x64

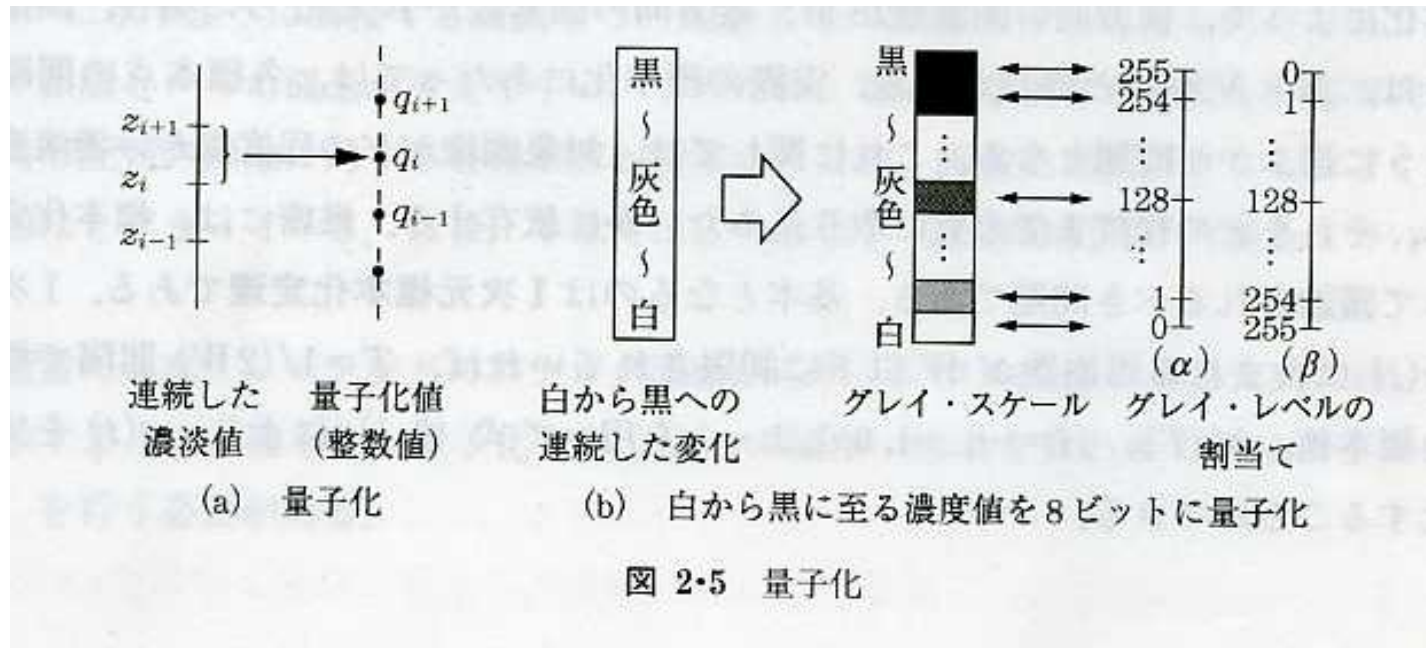
32x32

実際の画像の画素数

漢字	： 32×32～64×64 画素（ただし，文字としてのパターンを区別するだけでなく，細かい部分の形状などを扱うためには，さらに多くの画素数が必要である）
顕微鏡画像	： 256×256～512×512 画素
標準方式の TV 画像	： 720～910×480 画素
高精細 TV (HDTV)	： 1 900×1 100 画素程度
医用 X 線画像	： 2 000×2 000 画素程度
A 4 図面	： 1 700×2 600 画素程度
衛星画像	： LANDSAT (1 バンド)…3 240×2 340 画素 SAR (合成開口レーダ)…8 000×8 000 画素程度
CRT ディスプレイ	： 表示画面の解像度として 1 024×768 画素～2 048×1 536 画素のものが一般に用いられている．2 000×2 000 画素以上のもの（超高精細ディスプレイ）もある．

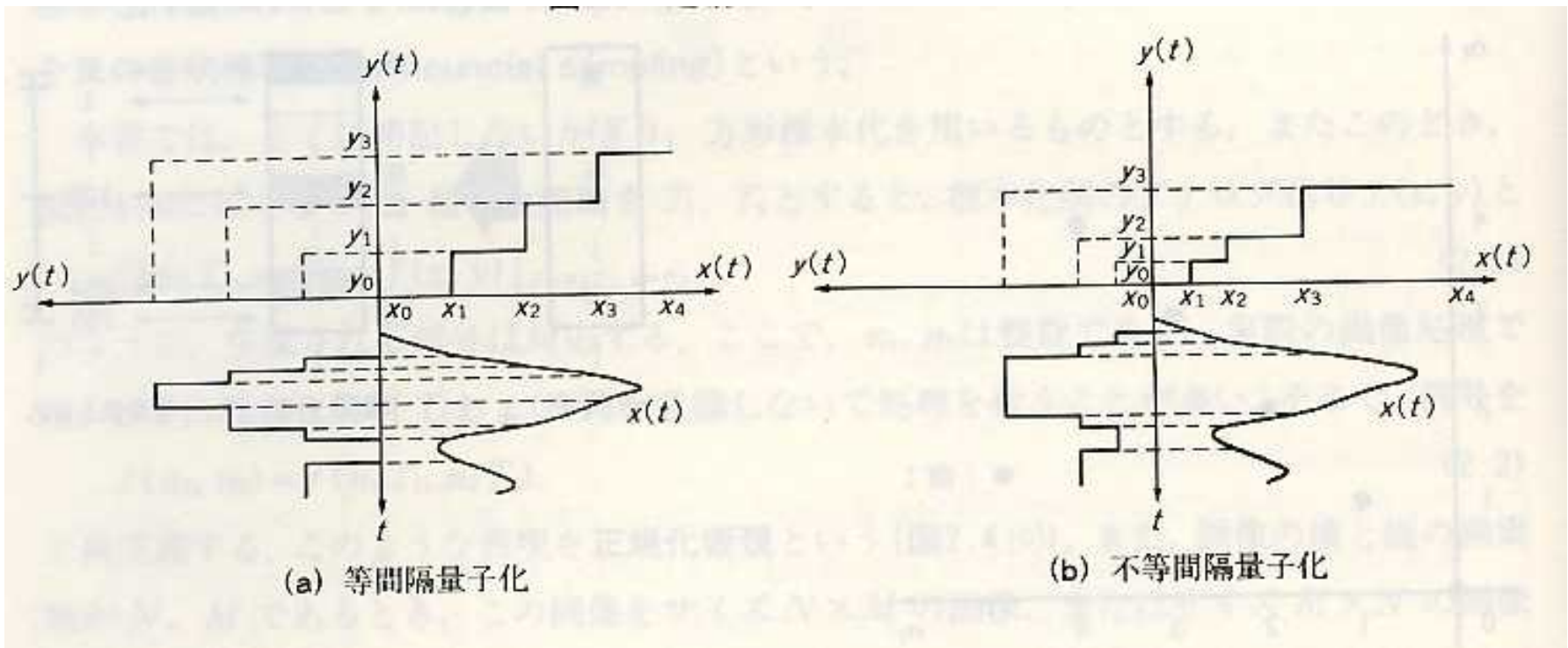
量子化

- デジタル画像に変換するには、横軸の空間を離散化するだけでなく、縦軸の濃度(強度)も離散の値(整数値)に変換する必要がある。その変換プロセスを量子化という。
- 量子化誤差: 真の値 z と濃度レベル q_i との差
- 一般の画像は8bit(256レベル)に量子化する。



量子化方法

- 等間隔量子化(直線量子化):最もよく用いられる
- 不等間隔量子化(対数量子化など)



量子化レベルと画像との関係



256レベル(8bit)

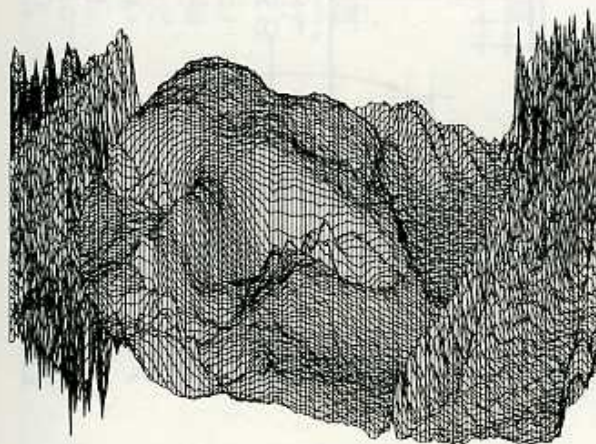


16レベル(4bit)

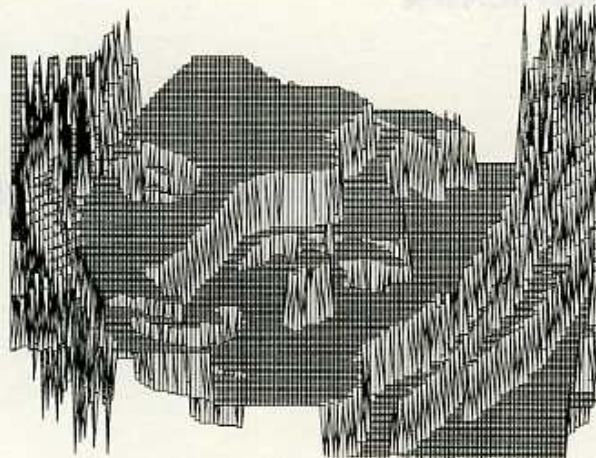


2レベル(1bit)

量子化例

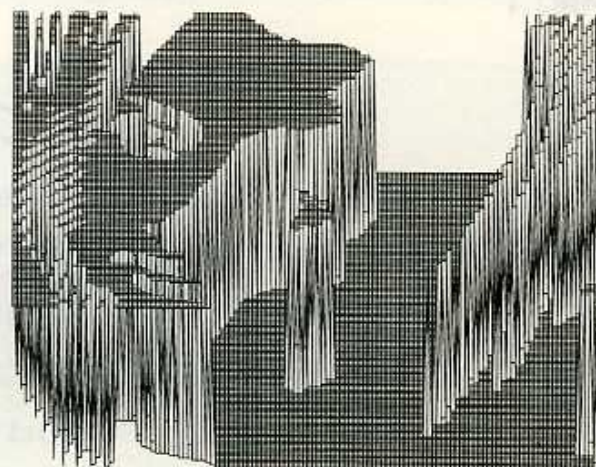
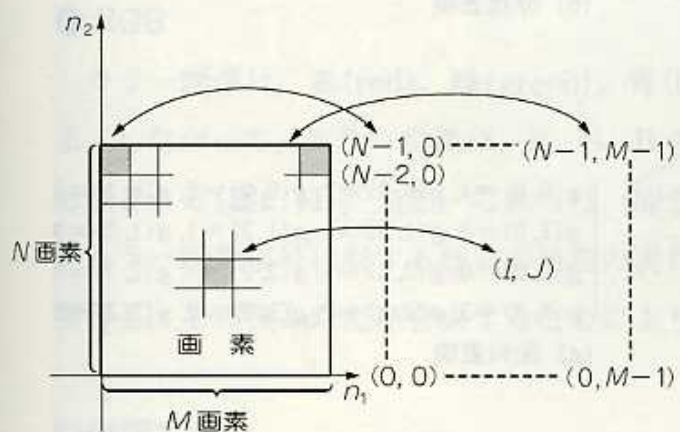


(a) 256 階調



(b) 4 階調

図2.10 画像の x - y 座標表現



(c) 2 階調

ビットとバイト

- 1ビット: 0 1
(0 1)
- 2ビット: 00 01 10 11
(0 1 2 3)
- 3ビット: 000 001 010 011 100 101 110 111
(0 1 2 3 4 5 6 7)
- 4ビット: 0000 0001 0010 0011 0100 0101 0110 0111
1000 1001 1010 1011 1100 1101 1110 1111
(0 1 2 3 4 5 6 7
8 9 10 11 12 13 14 15)
- 1バイト=8ビット: 0~255

写真のサイズ

- 1024x1024 のフルカラー画像(100万画素)
- カラー: (R, G, B)
- R,G,Bはそれぞれ8bit (諧調: 0~255)
- サイズ = $1024 \times 1024 \times 3 \times 8 = 24,769,824 \text{ bit}$
= 3,096,228 byte = 3MB
(3枚のFDDが必要)

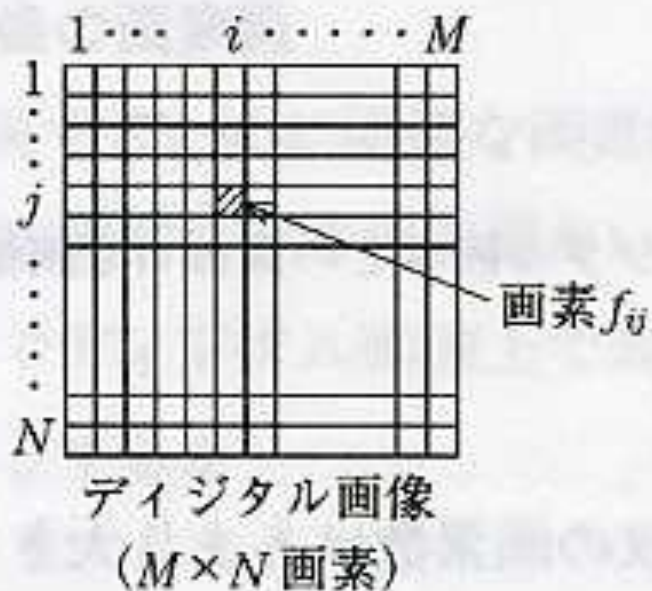
課題: 転送速度が64Mbpsの場合、上記の写真を送信するにはどのくらい時間がかかるか？

画像の種類

画像はピクセル(画素)の集合
各ピクセルは輝度と色などの情報をもつ

- 2値画像 (binary image)
 (各ピクセルに0か1をもつ)
- 濃淡画像 (grayscale image)
 (各ピクセルに0から255の間の整数をもつ)
- カラー画像 (color image)
 (各ピクセルに三つの整数データをもつ)

デジタル画像のコンピュータ内での取り扱い



$$[F] = \begin{bmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,M} \\ f_{2,1} & f_{2,2} & \cdots & f_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ f_{N,1} & f_{N,2} & \cdots & f_{N,M} \end{bmatrix}$$

(a) 慣用的に用いられている表記法

(b) 行列表現 (N行M列行列)

図 2・8 デジタル画像中の画素の表記のしかた

2次元配列を用いた画像表現

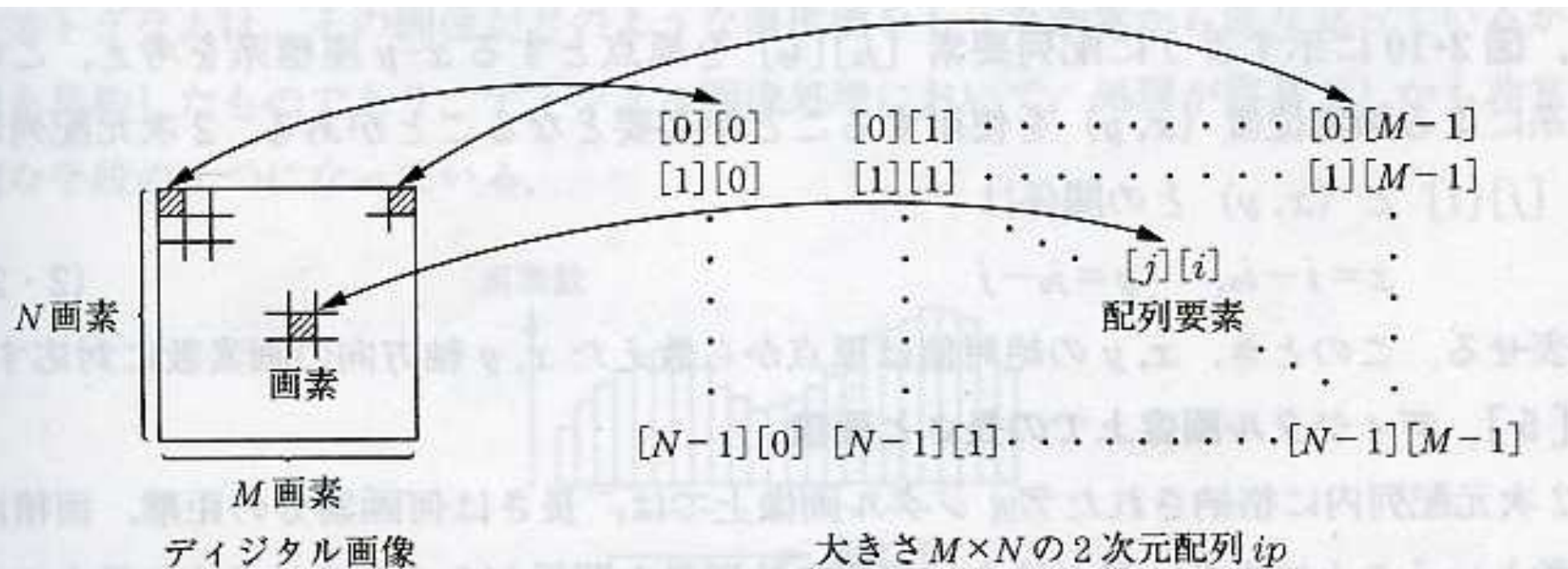


図 2・9 デジタル画像の 2 次元配列への格納 (C 言語の場合)

プログラム例

プログラム 2・1 画面全体の各画素に対する順次アクセス

```
for(j=0; j<N; j++){  
    for(i=0; i<M; i++){  
        ip[j][i] に関する処理  
    }  
}
```

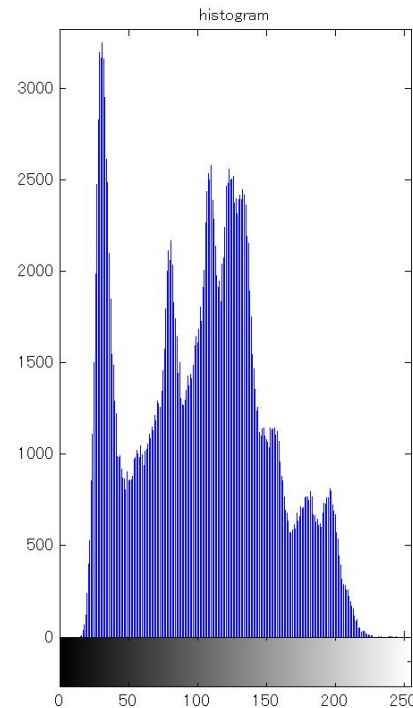
例えば、大きさ $M \times N$ 画素の画像 ip について、各画素の濃度値の総和 sum, および
平均値 ave を求めるプログラムは、プログラム 2・2 のように書くことができる。

プログラム 2・2 画素値の総和と平均値を求めるプログラム

```
int i, j, iave;  
float sum, ave;  
unsigned char ip[N][M]; // ここでは画素値は 8 ビット  
                        // とする  
  
sum=0.0;  
for(j=0; j<N; j++){  
    for(i=0; i<M; i++){  
        sum+=(float)ip[j][i];  
    }  
}  
  
ave=sum/(float)(M*N);  
iave=(int)(ave+0.5); // 小数点以下を四捨五入する
```

ヒストグラム

- 各濃度値に対して、画像中におけるその濃度値をもった画素数をもとめたものを**ヒストグラム**という。
- 横軸を濃度値、縦軸を画素数(or生起確率)



暗い ← ————— → 明るい

ヒストグラムのプログラム例

プログラム 2・3 ヒストグラムの計算

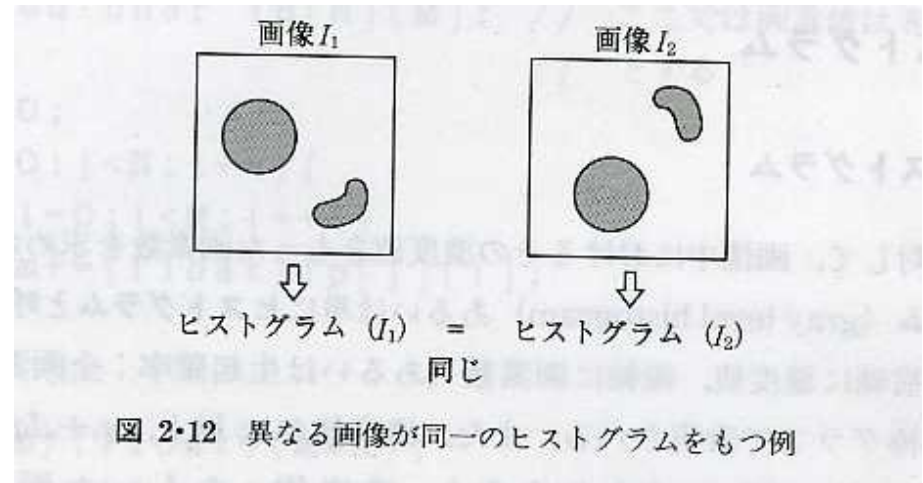
```
int i, j;
int ip[N][M]; // ここでは濃度レベル数が 256 階調 (8 ビット)
int hst[NGL]; // 以上である場合も想定し, int で宣言する

for (i=0; i<NGL; i++) {
    hst[i]=0;
}

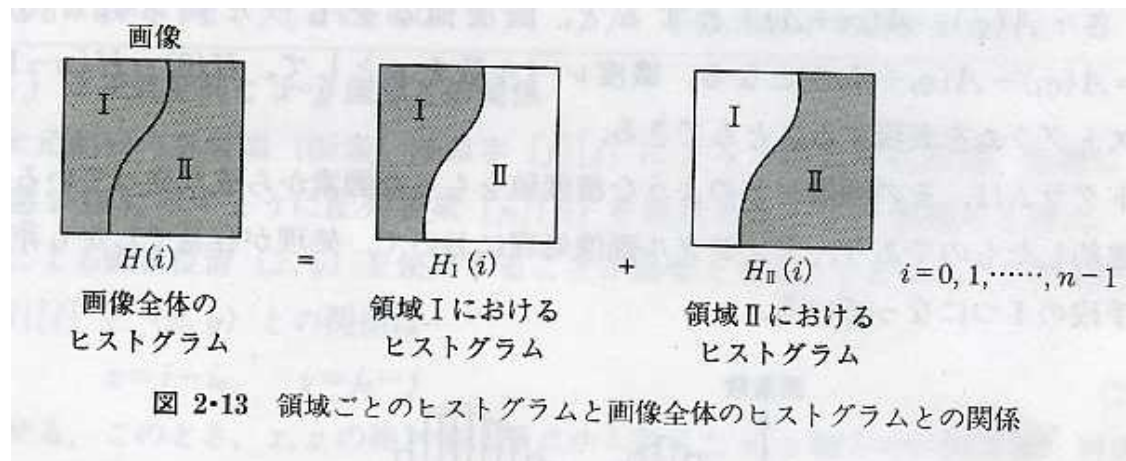
for (j=0; j<N; j++) {
    for (i=0; i<M; i++) {
        hst[ip[j][i]]++;
    }
}
```


ヒストグラムの性質

- 空間情報がないため、異なる画像が同一ヒストグラムをもつことがある。



- 画像をいくつかの領域に分割し、それぞれの領域のヒストグラムの和は画像全体のヒストグラムの和と同じである。



ヒストグラムは適切にスケールされることが重要

コントラストが最もよい

コントラストが低い

一部分の濃淡変化が失われる

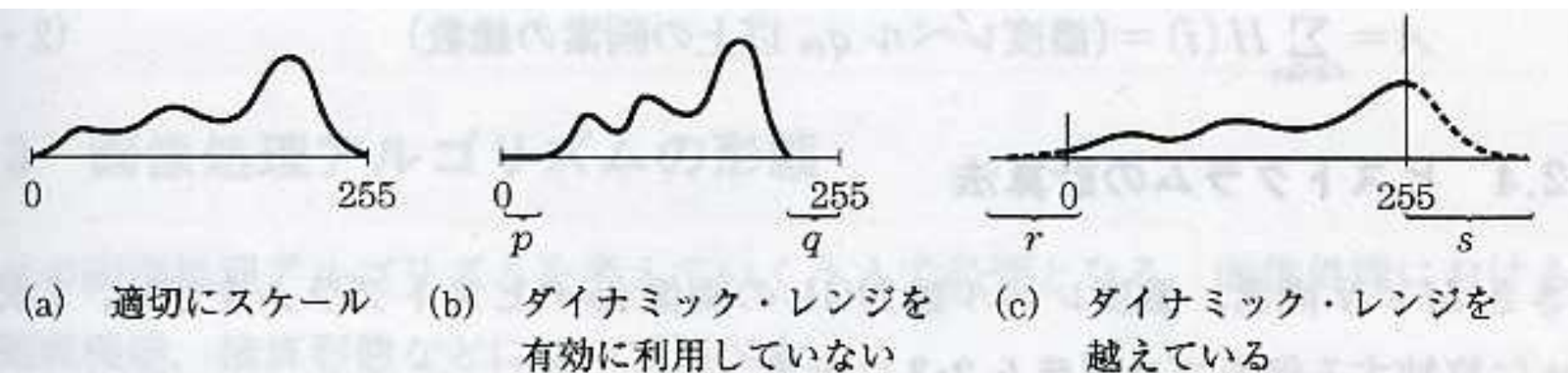
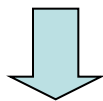


図 2-14 入力画像の性質とヒストグラムとの関係

ヒストグラムのスケール変換

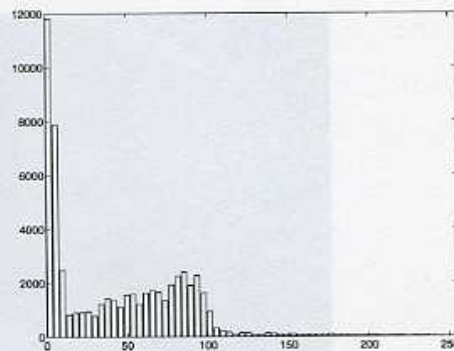
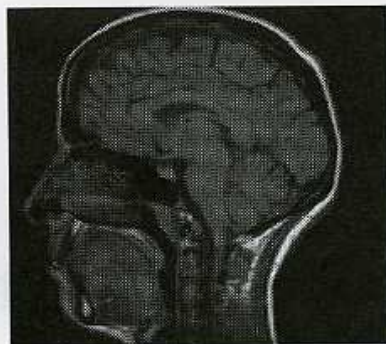
濃度レンジ

a b

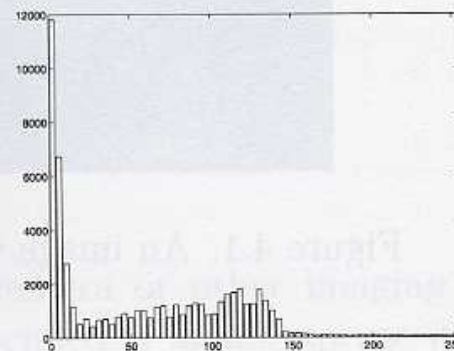
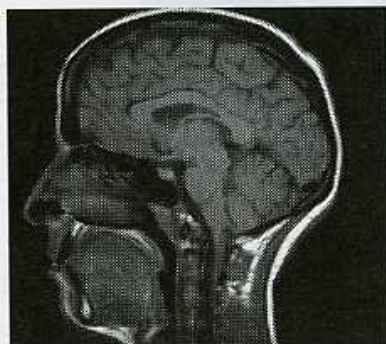


z_1 z_k

$$z' = \frac{z_k - z_1}{b - a} (z - a) + z_1$$



変換前



変換後

ヒストグラムを利用した2値化

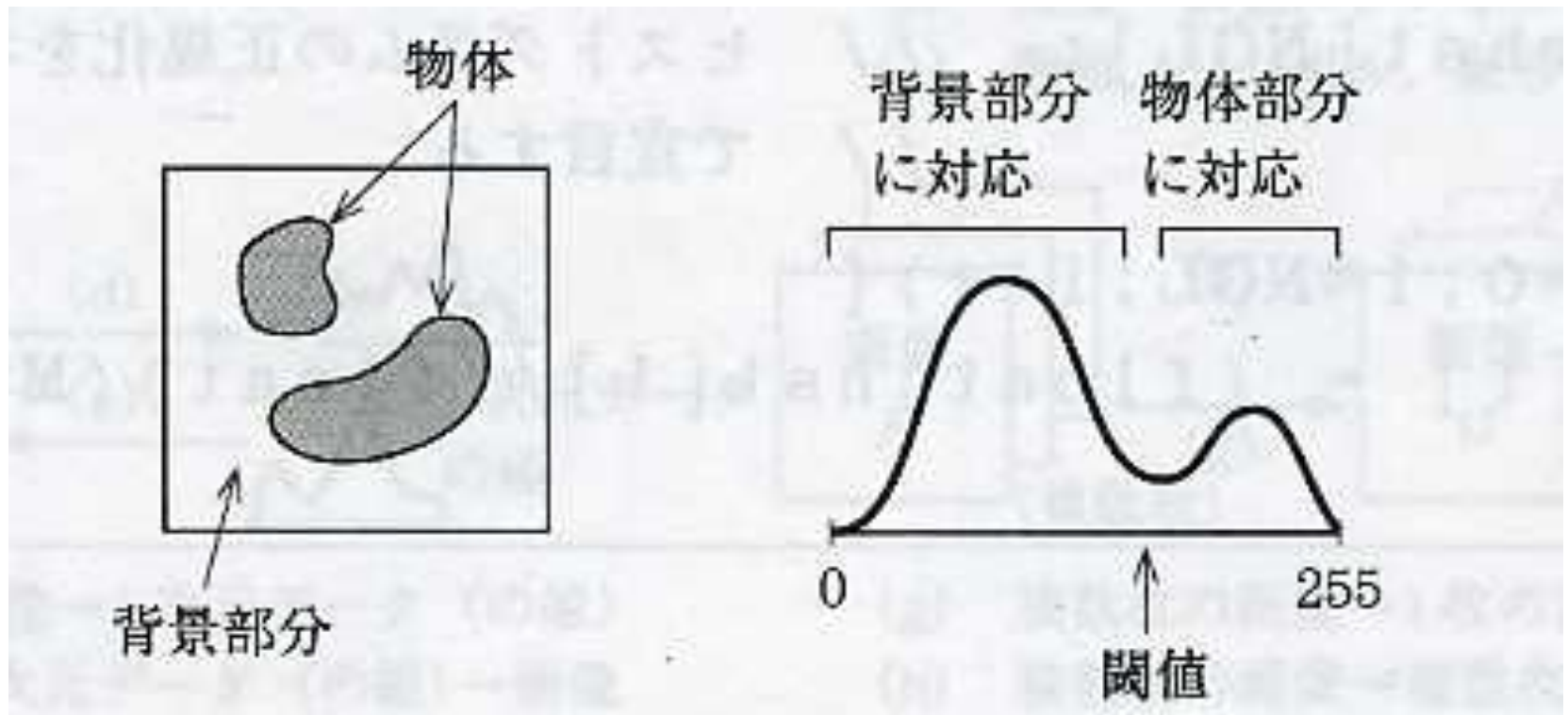


図 2・15 2 値化のための閾値のヒストグラムによる選択