

画像処理2019ー4

画像処理の基礎

- * 点演算 (階調補正, しきいち処理)
- * 局所演算 (雑音除去, エッジ抽出)
- * 連結領域、トポロジー

パターン認識過程

パターン観測

入力, 量子化, 標本化など

前処理

雑音除去, 濃度変換,
領域検出, セグメンテーションなど

特徴抽出

幾何特徴抽出, 次元圧縮
統計的パターン解析, 動き解析,
スペクトル解析など

$\mathbf{x} = (x_1, x_2, \dots, x_n)$

パターン識別

ベイズ決定則, 最近傍則,
HMM, ニューラルネットなど

$\mathbf{x} = \text{argmax}(g_1(\mathbf{x}), g_2(\mathbf{x}), \dots, g_n(\mathbf{x}))$

後処理

表示, 分類,

1. 前処理

■ 1. 1 点演算

□ 階調補正

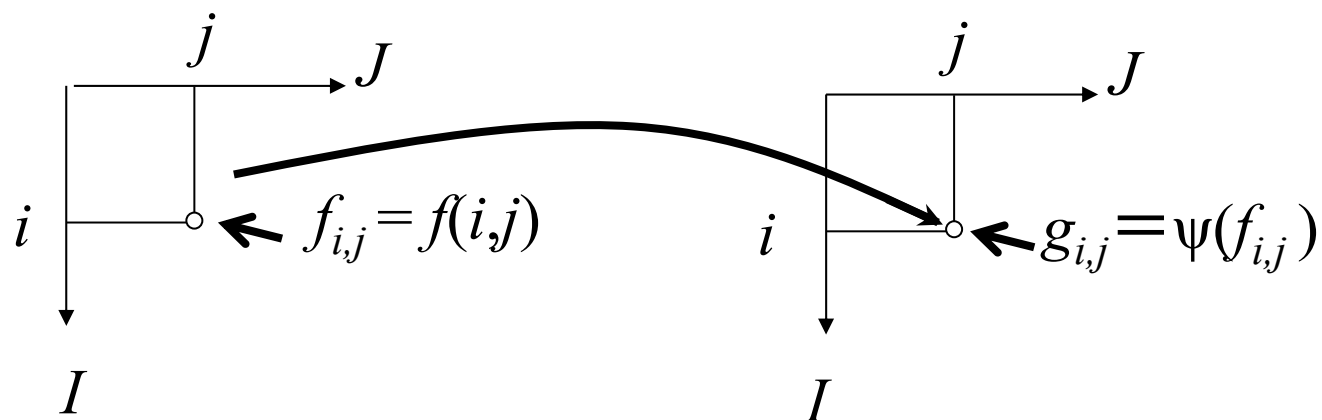
■ 伸長, 圧縮

■ 2値化

□ しきい値処理

■ しきい値決定法

点演算

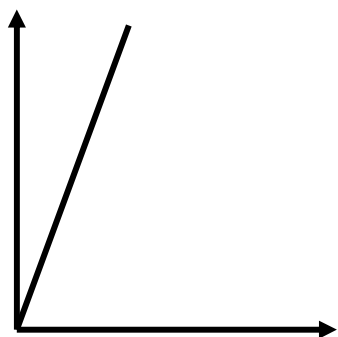


出力画像の各画素値

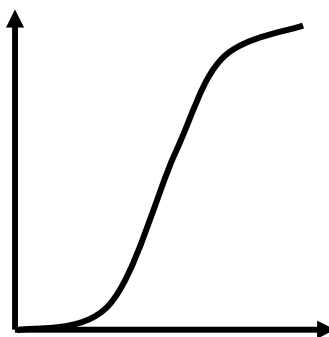
入力画像の同じ位置の画素値のみに依存

点演算の例(1): 階調補正

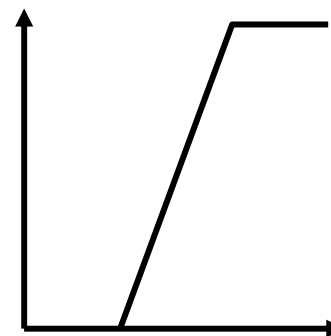
入力画像を人間にとって見やすく変換



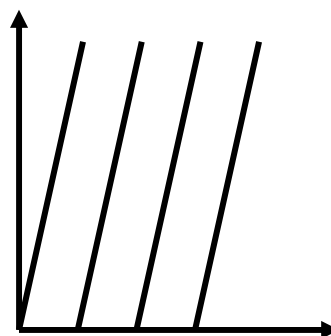
範囲の一樣伸張



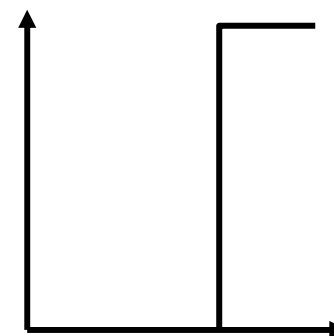
非線形変換



窓処理

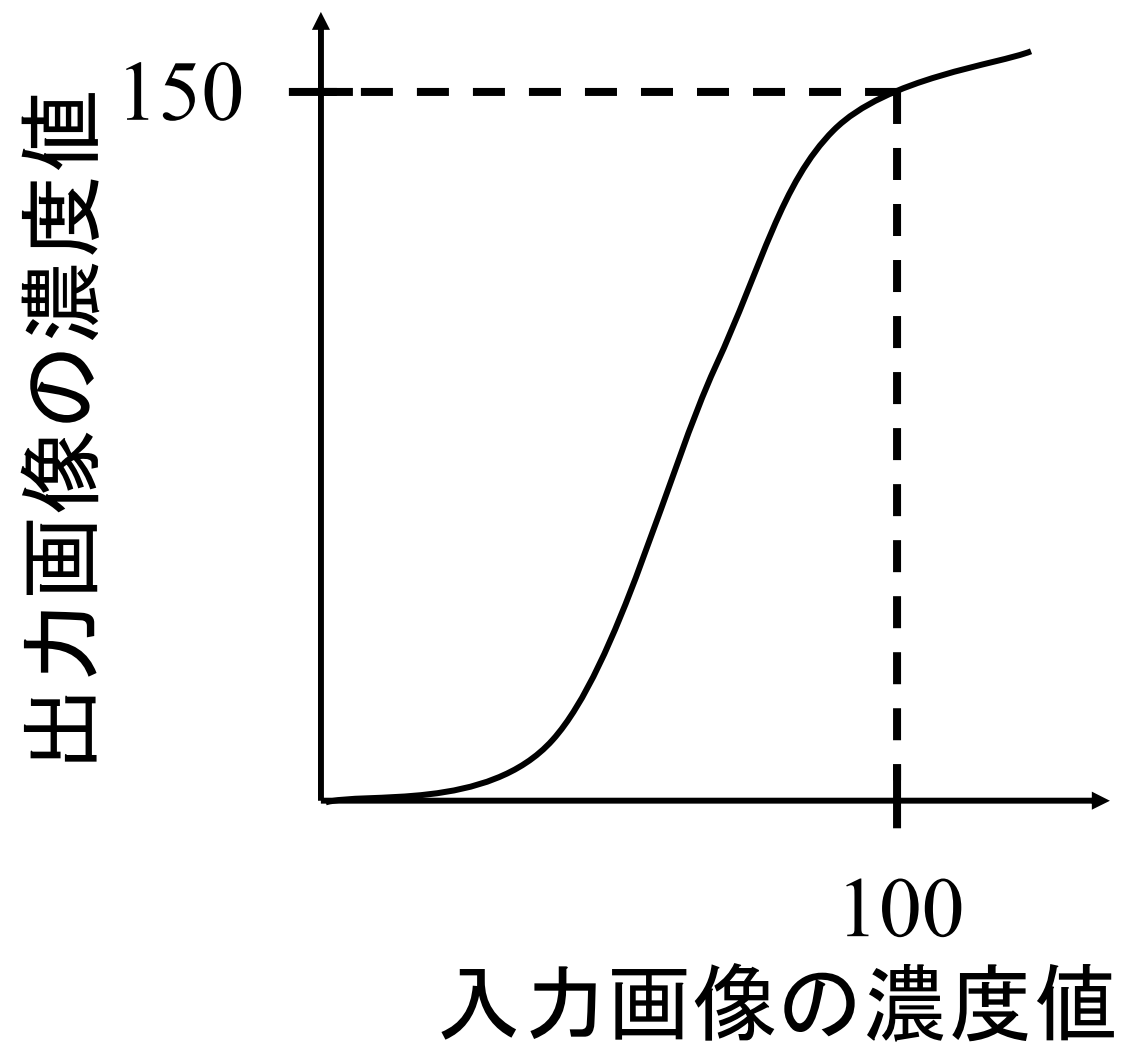


等高線表示



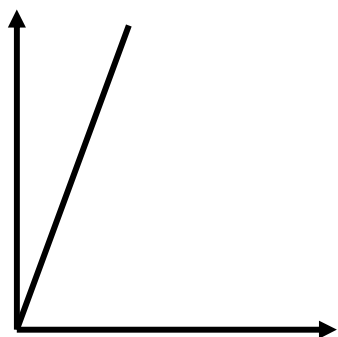
しきい値処理

グラフの見方

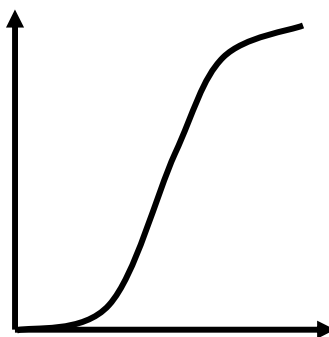


階調補正

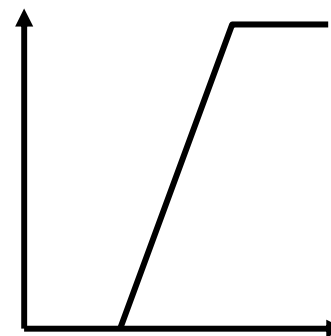
入力画像を人間にとって見やすく変換



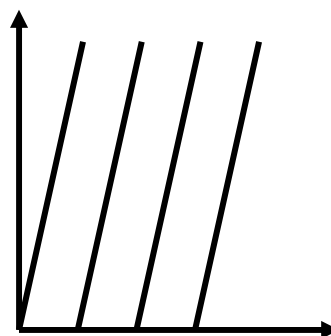
範囲の一樣伸張



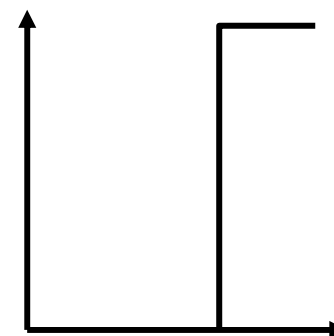
非線形変換



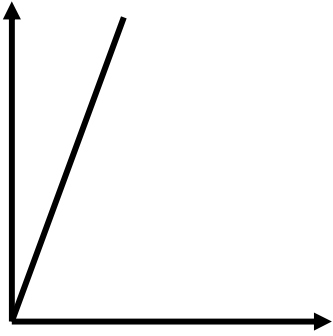
窓処理



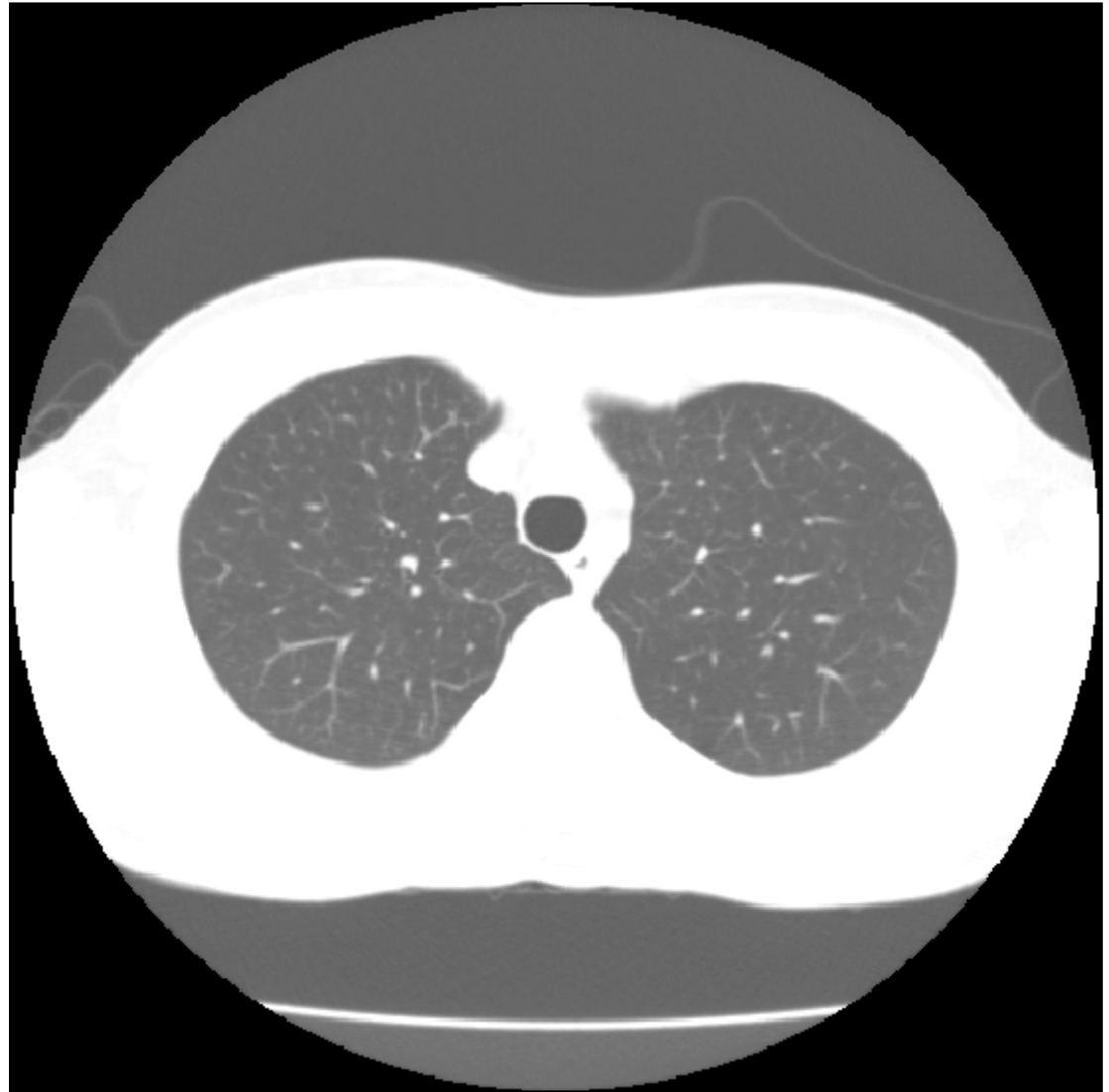
等高線表示

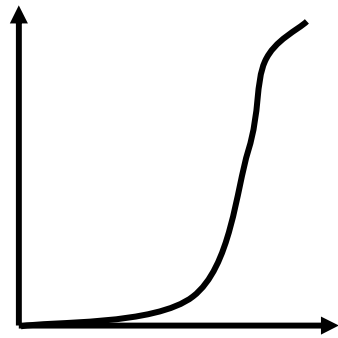


しきい値処理

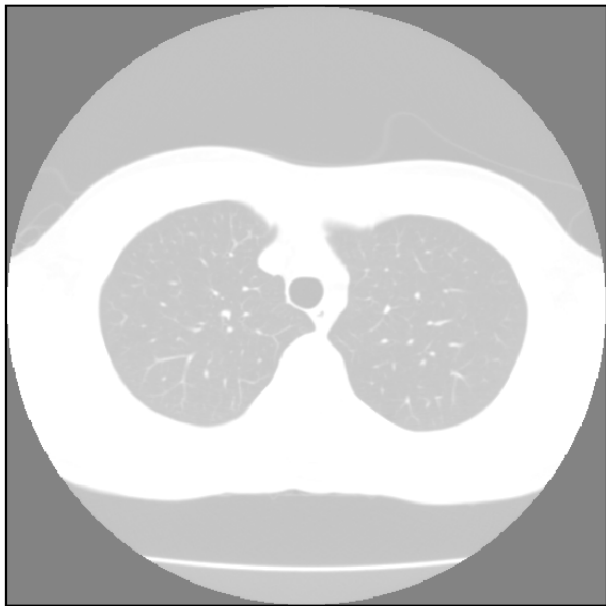


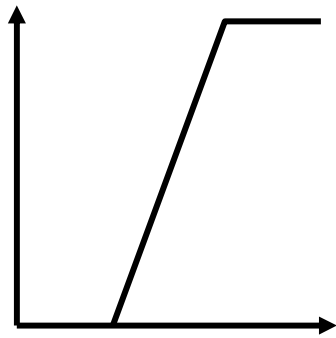
範囲の一樣伸張



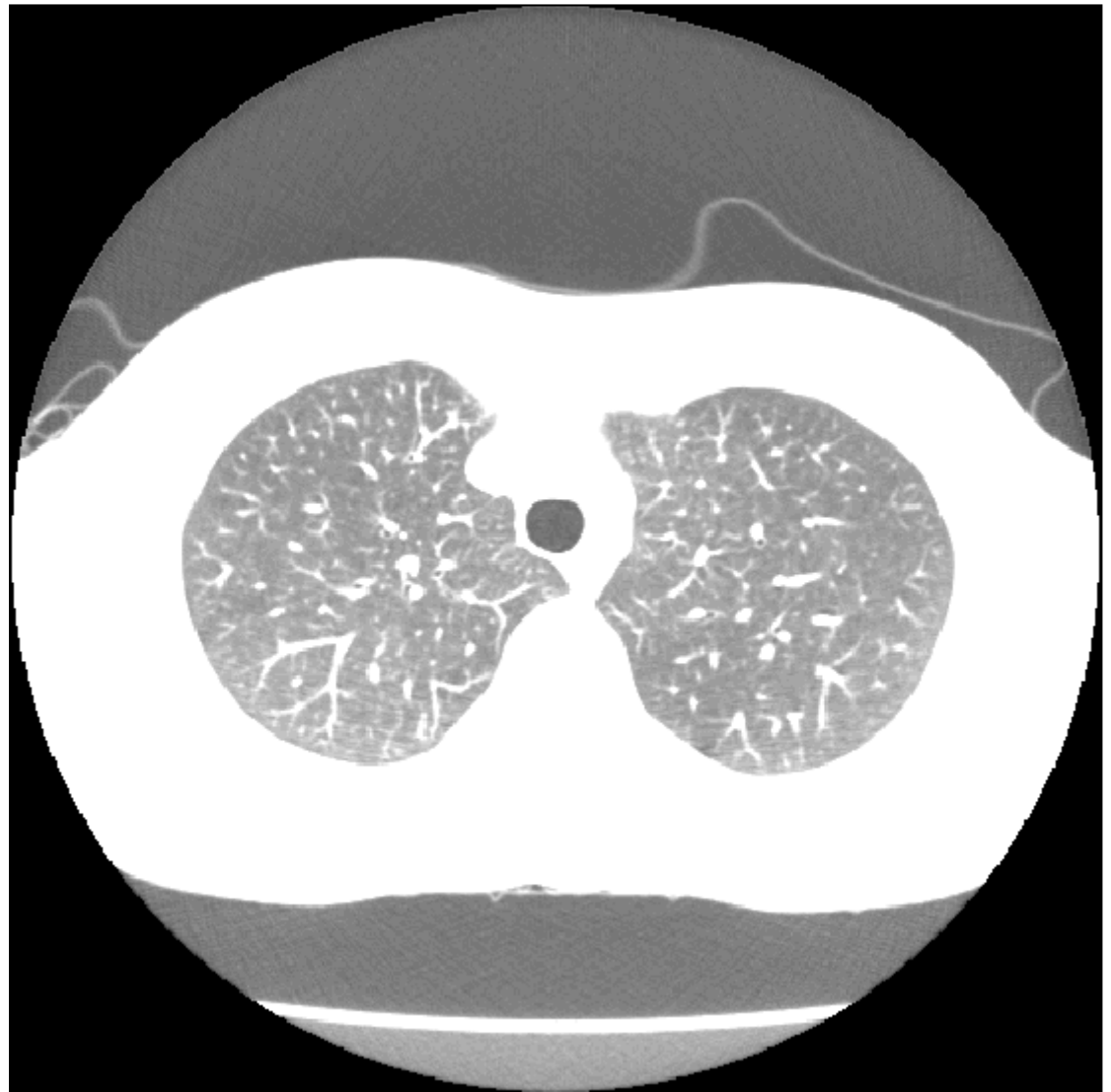
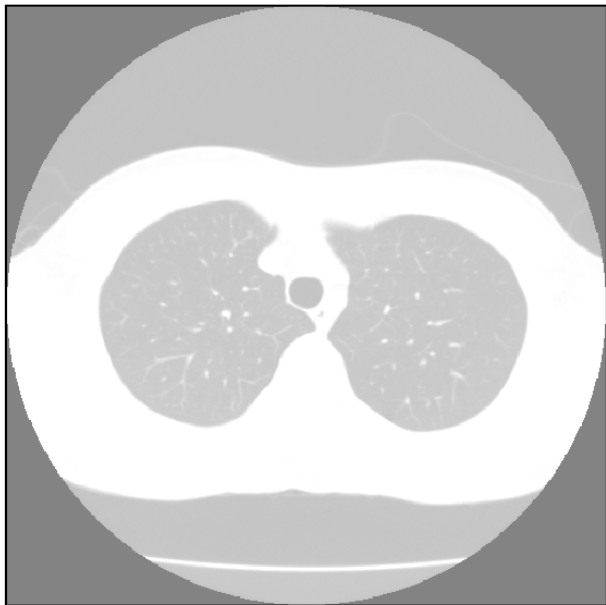


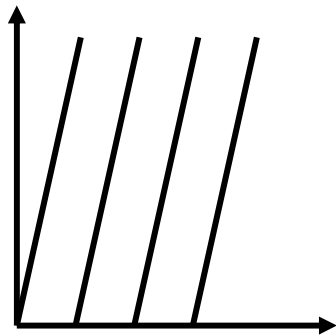
非線形変換



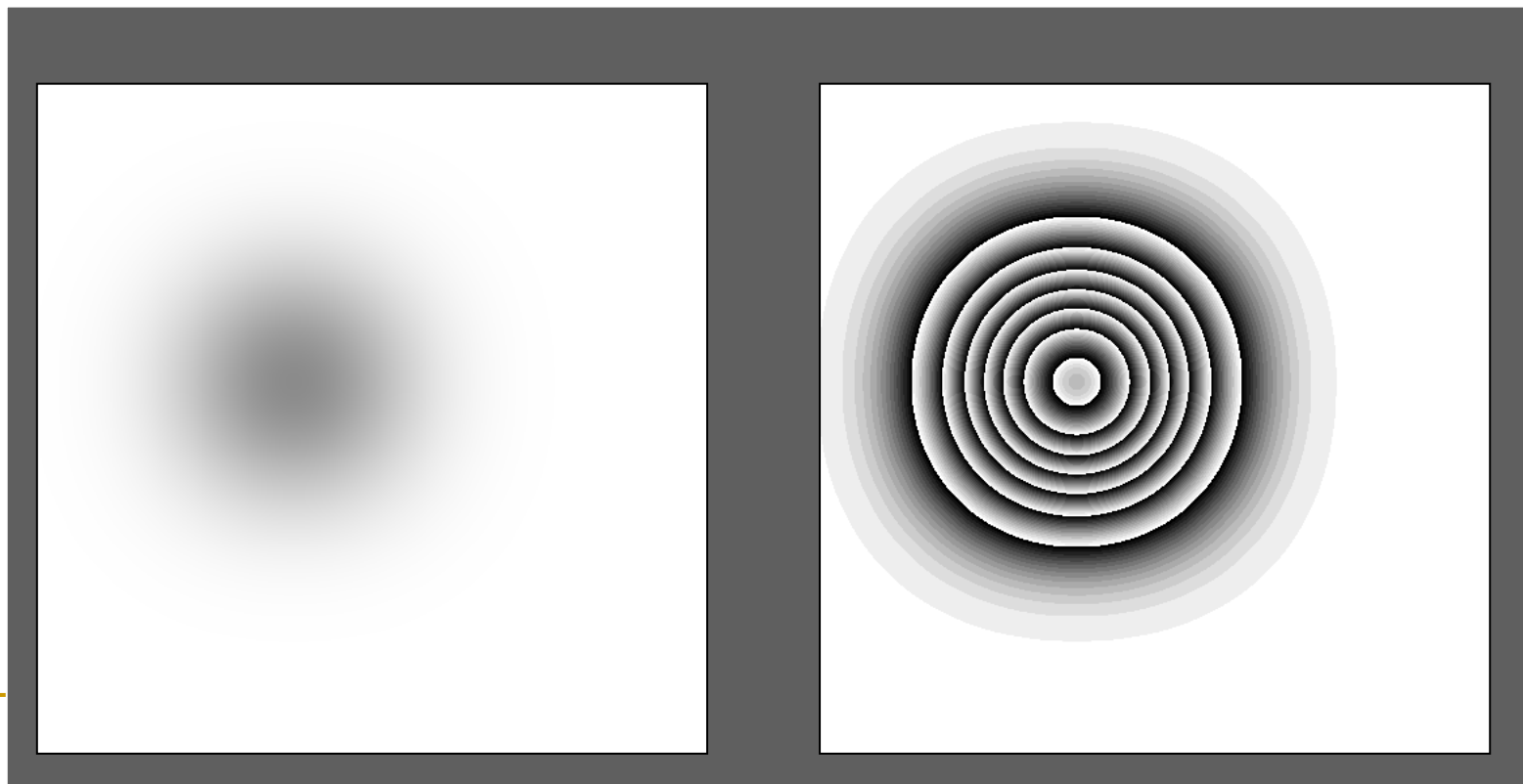


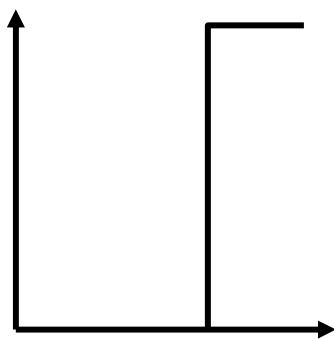
窗处理



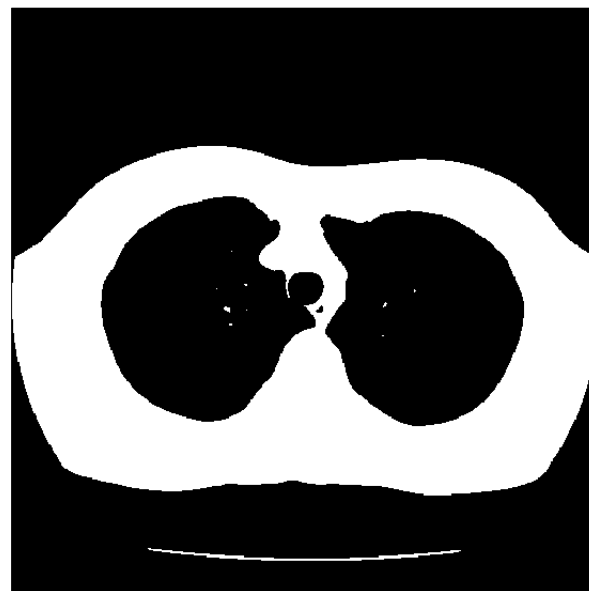
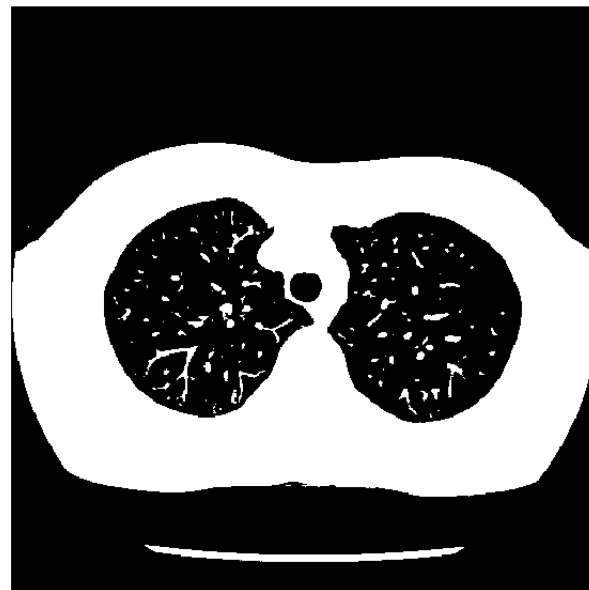


等高線表示





しきい値処理





γ 補正 (ガンマほせい)

表示デバイス (CRT, 液晶) の実際の発光出力 O と, 駆動信号 I の間には, 次式で表される関係がある.

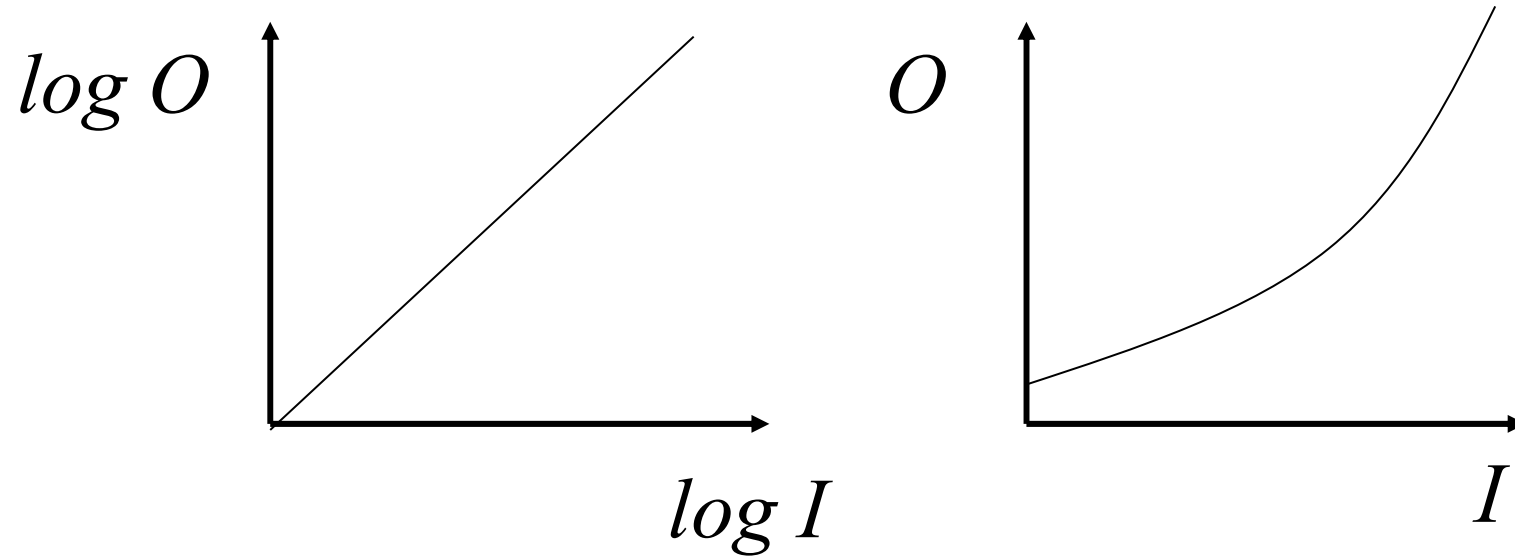
$$O = I^\gamma \quad \gamma: \text{定数}$$

このような非直線性を示す特性を γ 特性と呼ぶ.

このため, 表示デバイスを駆動するときは, γ 特性を逆補正した信号で行う必要がある. たとえば, 標準RGB色空間では, $\gamma=2.2$ と言われているので, $1/\gamma=0.45$ の曲線で補正する必要がある.

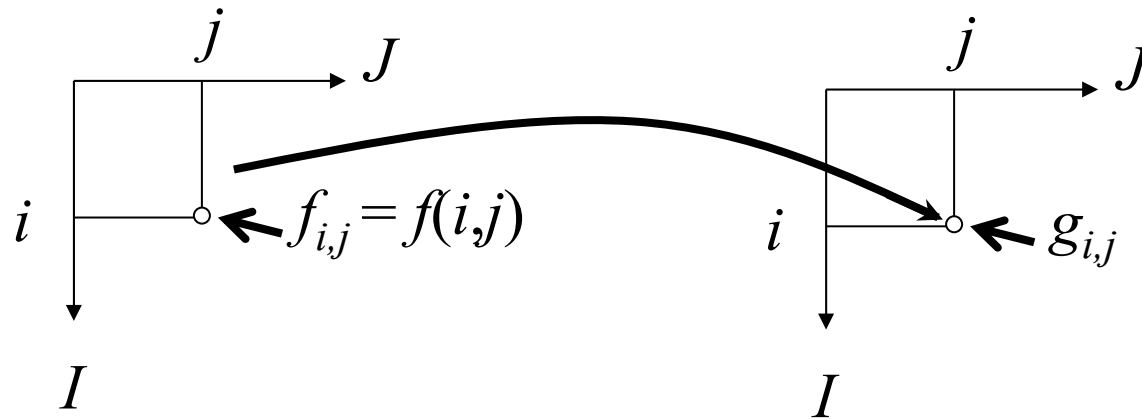
昔のテレビはこの γ 特性があったので、放送局で補正した信号を作り送り出していた。

$$\log O = \gamma \log I$$



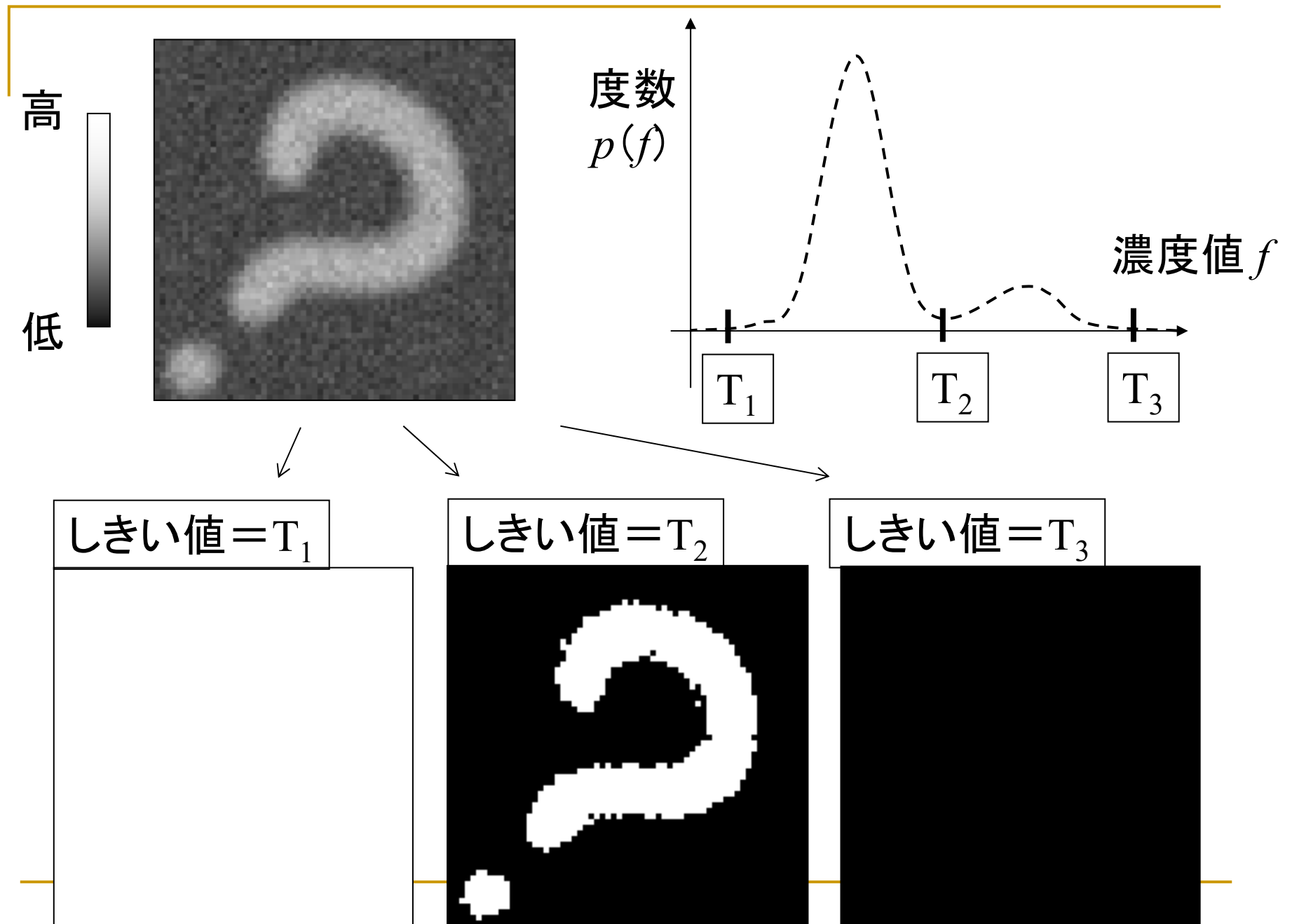


点演算の例(2) 二値化処理



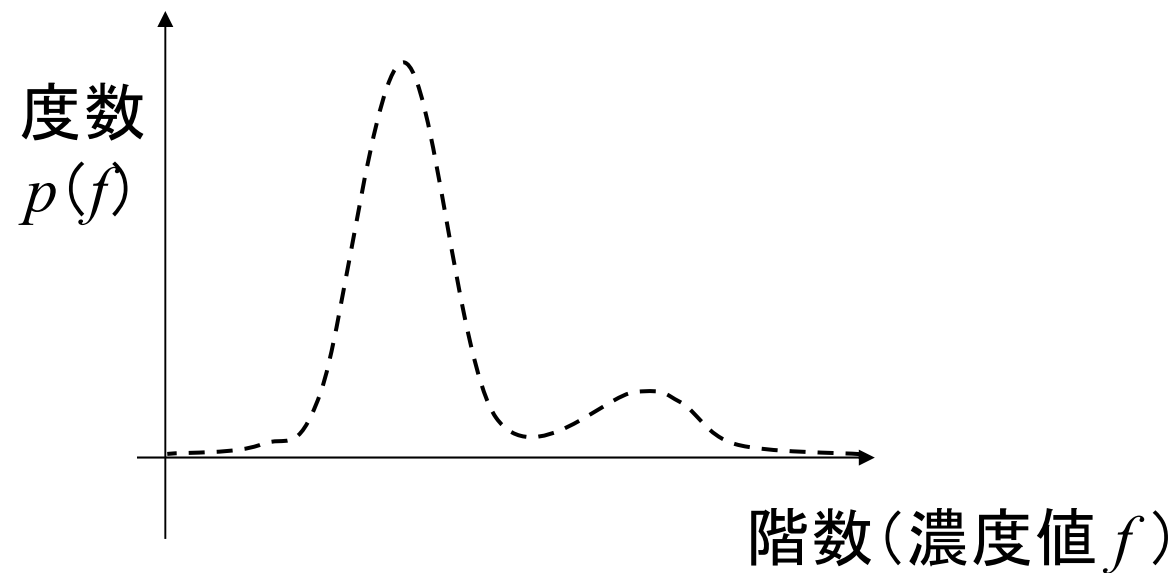
$$g_{i,j} = \begin{cases} 1 & f_{i,j} \geq T \\ 0 & f_{i,j} < T \end{cases} \quad (T: \text{しきい値})$$

どうやってしきい値を決定するか?



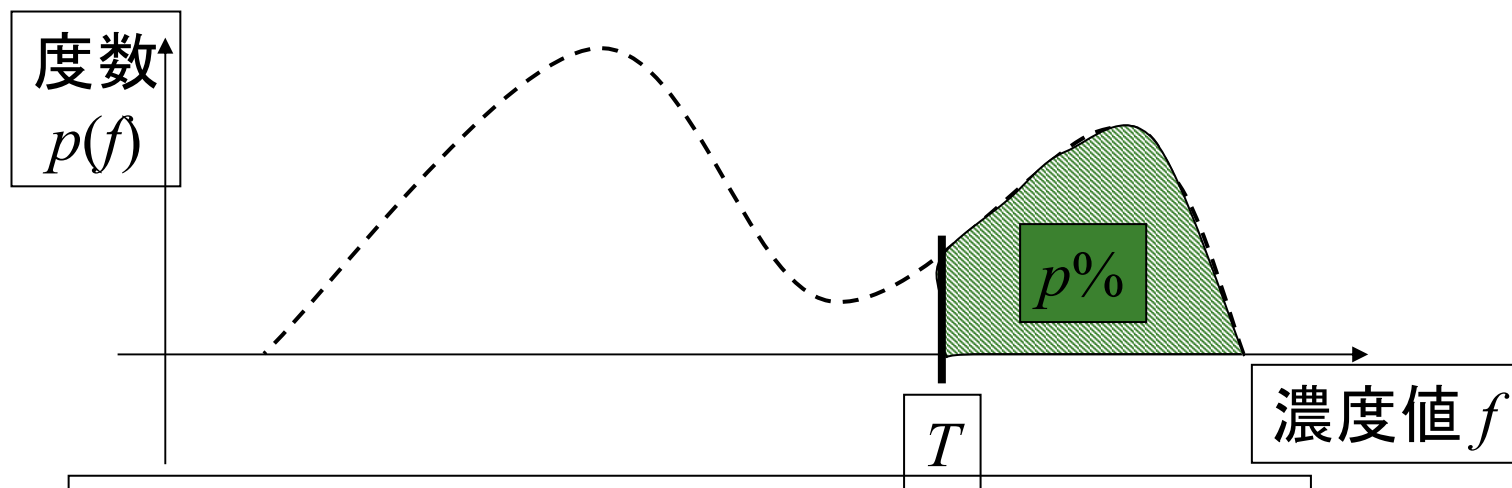
用語： ヒストグラム(histogram)

■ 度数分布図



p -タイル法

【しきい値 T 】 = 濃度値の確率分布の $p\%$ 点

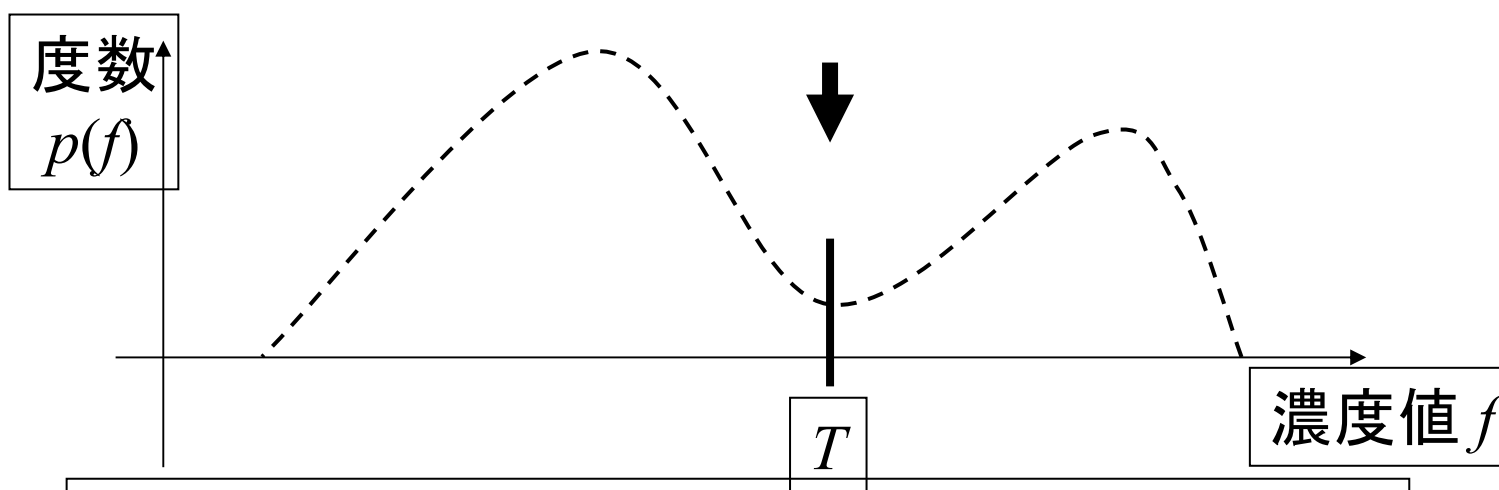


利用可能な場合
画像全体に対する図形の割合が一定の場合

問題点
図形の割合が一定でない場合には利用不可

モード法

【しきい値 T 】 = 濃度値の確率分布の極小点



利用可能な場合

図形と背景の濃度値分布がある程度異なる場合

問題点

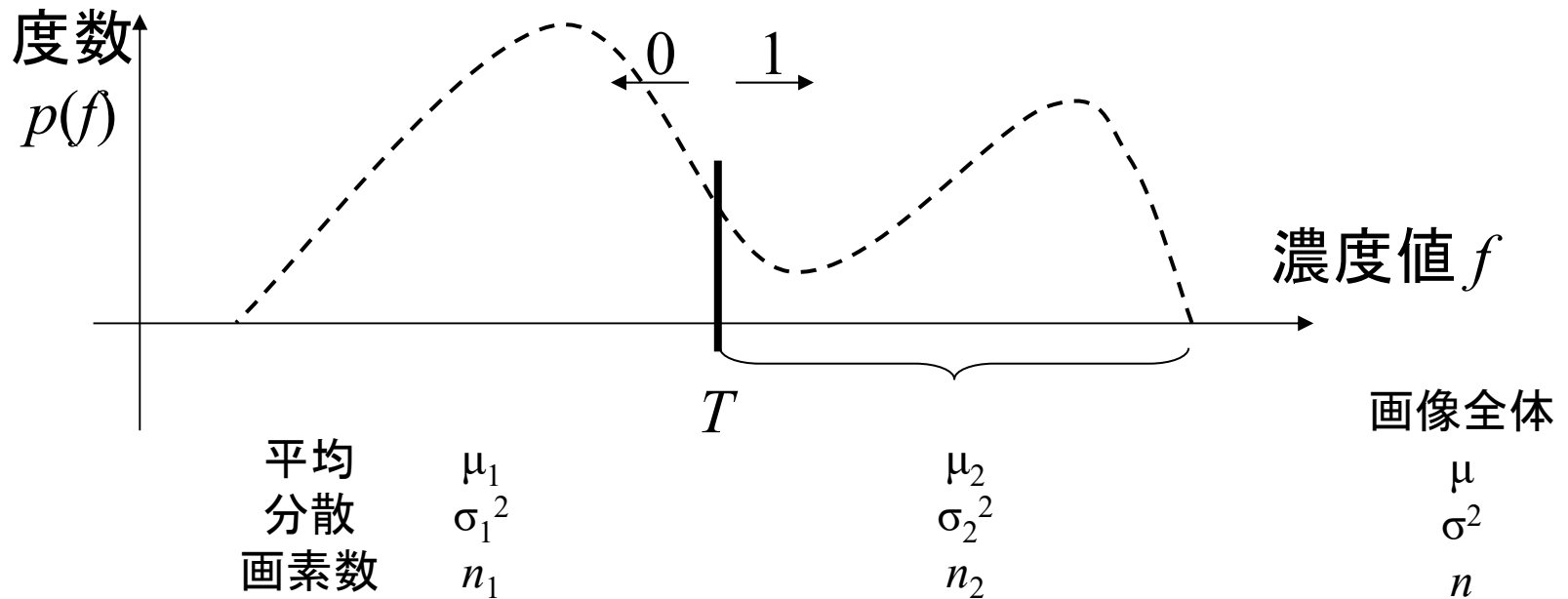
雑音やヒストグラム作成時のパラメータによる影響大

理由: 雑音による極小値との区別が困難

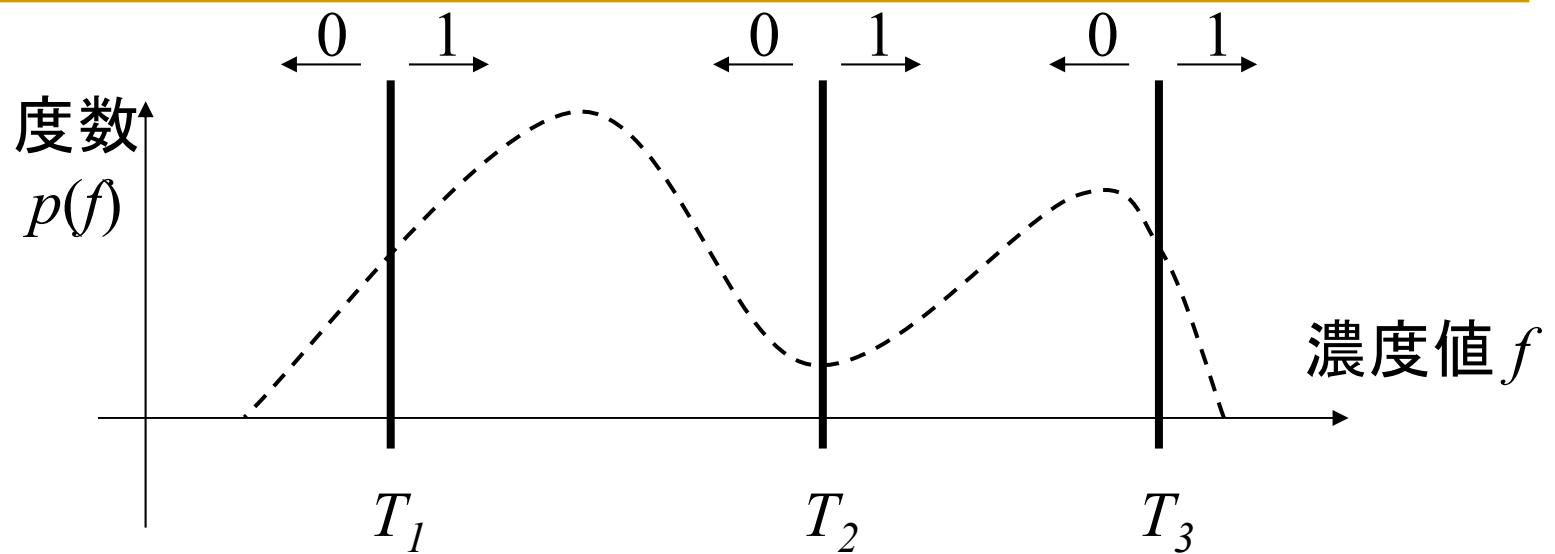
ヒストグラムの形状は階級の幅によって変化

判別分析法(大津のしきい値選定法)

【しきい値 T 】 = しきい値処理後に「1-画素と0-画素領域の濃度分布間の分離度を最大」とする濃度値



$$\text{分離度} = \frac{n_1 n_2 (\mu_1 - \mu_2)^2}{n (n_1 \sigma_1^2 + n_2 \sigma_2^2)}$$



$$\text{分離度} = \frac{n_1 n_2 (\mu_1 - \mu_2)^2}{n (n_1 \sigma_1^2 + n_2 \sigma_2^2)}$$

← クラス間分散 σ_B

← クラス内分散 σ_W

$$\sigma_B > \sigma_y, \sigma_R$$

山と山の間が遠い

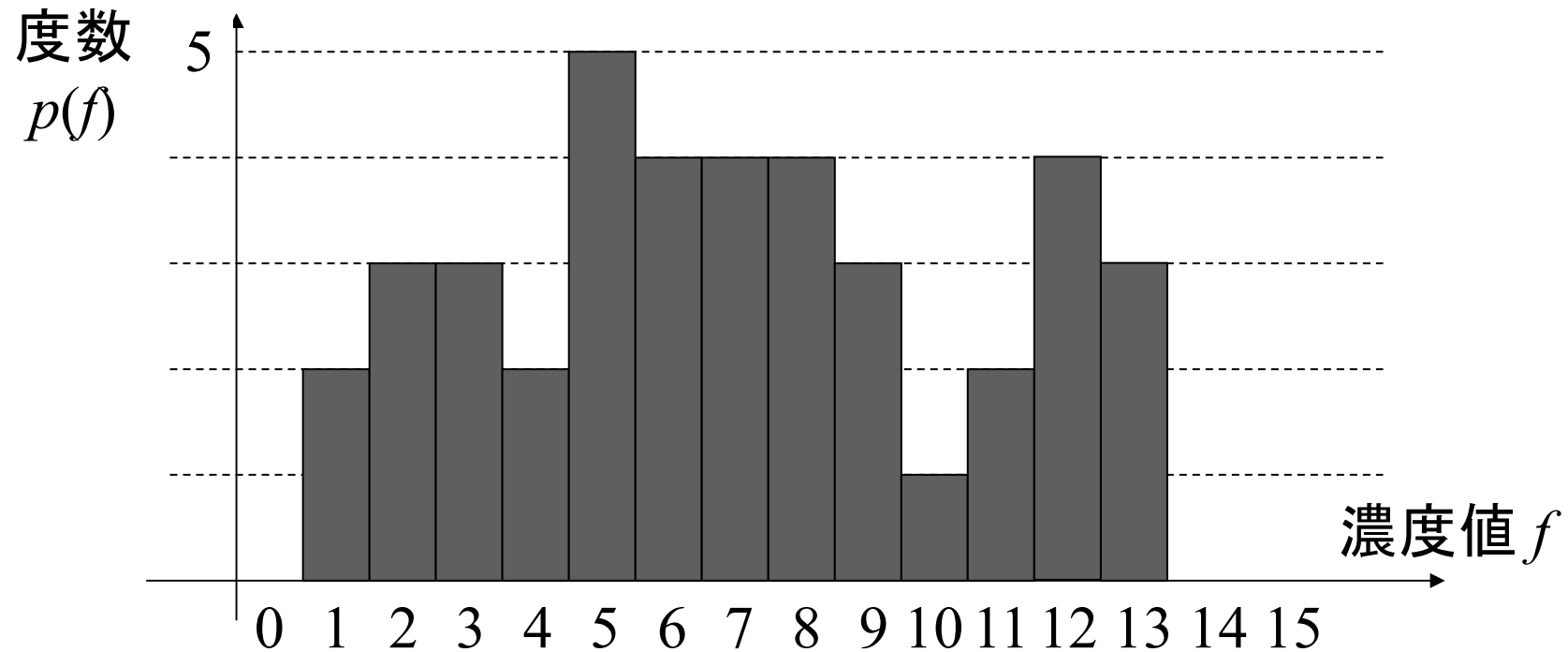
$$\sigma_W < \sigma_y, \sigma_R$$

それぞれの山がコンパクト

T_2 において分離度最大！

プログラムチャレンジ 2

次のヒストグラムを持つ画像に対して p -タイル法(22.5%点), モード法, および, 判別分析法でしきい値を決定せよ.



問題2

ヒストグラムの階級の幅を倍にしてモード法を適用せよ.

問1. ヒストグラムと似て呼び名が異なる次のもの(1次元のデータ構造)について説明せよ.

- プロファイル(Profile, 断面)
- 周辺分布(marginal distribution, projection profile)

■

1. 2 局所フィルタ・局所処理

- 局所フィルタは
 - 局所フィルタの移動走査適用
 - Convolution (畳み込み演算)

- 濃淡画像
 - 差分フィルタ(線形, 非線形)

 - 平滑化フィルタ(線形, 非線形)

- 2値画像
 - モルフォロジーフィルタ

 - 連結成分, 距離変換, 細線化 ←次回

その前に

- 近傍と距離についての概念

4近傍と8近傍

X4	X3	X2
X5	X0	X1
X6	X7	X8

X0に対して{x1, x3, x5, x7}が4近傍
X0に対して{x1, x2, ..., x7}が8近傍

中心画素からの距離

2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2

8近傍距離
チェス盤距離

2次元
 $d(Pa, Pb) =$
 $\max\{|x_a - x_b|, |y_a - y_b|\}$
 L_∞ ノルム

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

4近傍距離
市街地距離
マンハッタン距離

$d(Pa, Pb) =$
 $|x_a - x_b| + |y_a - y_b|$

L_1 ノルム

$2\sqrt{2}$	$\sqrt{5}$	2	$\sqrt{5}$	$2\sqrt{2}$
$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$
2	1	0	1	2
$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$
$2\sqrt{2}$	$\sqrt{5}$	2	$\sqrt{5}$	$2\sqrt{2}$

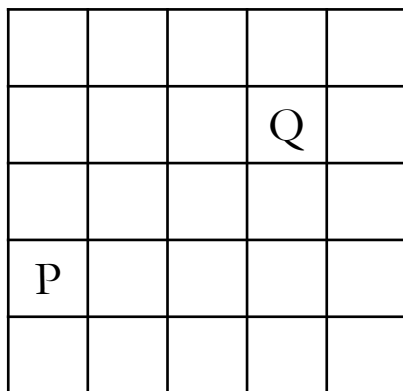
ユークリッド距離

$d(Pa, Pb) =$
 $(|x_a - x_b|^2 + |y_a - y_b|^2)^{1/2}$

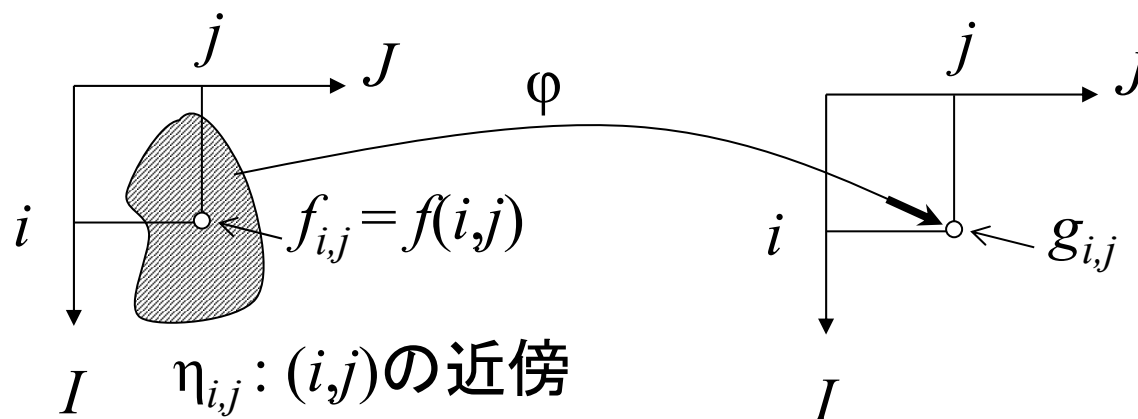
L_2 ノルム

問2

・2点PQ間の距離を, 4近傍距離, 8近傍距離, ユークリッド距離でそれぞれ求めよ.



(I) 濃淡画像の局所処理



線形フィルタ

$$g_{i,j} = \sum_{(k,l) \in \eta_{i,j}} \omega_{k,l} f_{i+k,j+l}$$

重み係数

非線形フィルタ

$$g_{i,j} = \phi(f_{i+k,j+l}) \quad ((k,l) \in \eta_{i,j})$$

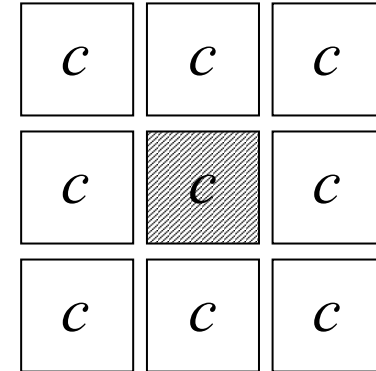
Max, Min等の非線形関数 (i, j) を原点とする局所座標

線形平滑化フィルタ

a) 一様重み平滑化フィルタ

$$g_{i,j} = \sum_{(k,l) \in \eta_{i,j}} \underbrace{c}_{\text{定数}} f_{i+k,j+l}$$

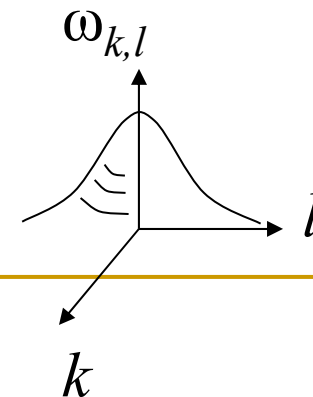
$\omega_{k,l}:$



b) ガウス型平滑化フィルタ

$$g_{i,j} = \sum_{(k,l) \in \eta_{i,j}} \frac{1}{2\pi\sigma^2} \exp\left(-\frac{k^2 + l^2}{2\sigma^2}\right) f_{i+k,j+l}$$

$\omega_{k,l}:$



線形平滑化フィルタ

目的: ノイズの除去、平滑化

ノイズの種類: ランダムノイズなど

原理: 重み付平均

a) 一様重み平滑化フィルタ

重みが一様

計算量小

b) ガウス型平滑化フィルタ

正規分布重み

計算量大、なめらか

非線形平滑化フィルタ

a) Maxフィルタ

特徴

$$g_{i,j} = \max_{(k,l) \in \eta_{i,j}} (f_{i+k,j+l}) \quad \leftarrow \text{窪みを除去}$$

b) Minフィルタ

$$g_{i,j} = \min_{(k,l) \in \eta_{i,j}} (f_{i+k,j+l}) \quad \leftarrow \text{突起を除去}$$

c) Medianフィルタ

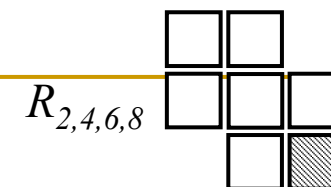
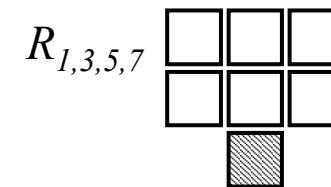
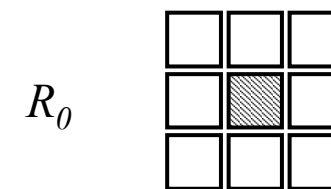
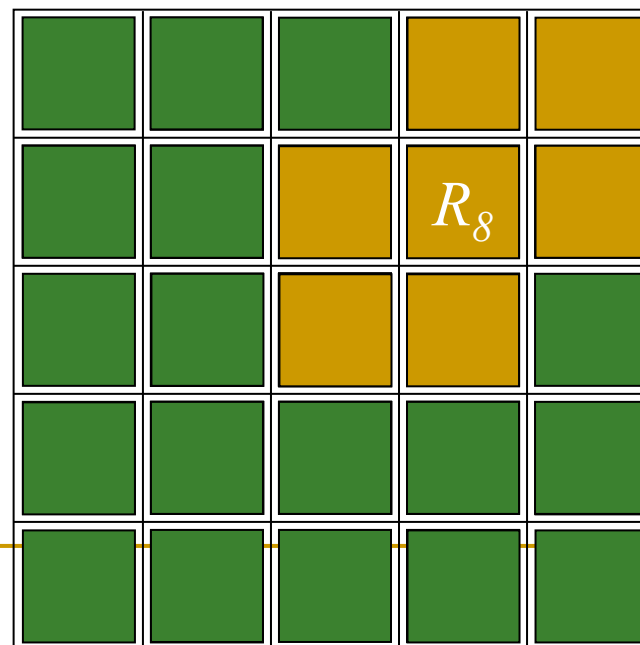
$$g_{i,j} = \underbrace{\text{median}_{(k,l) \in \eta_{i,j}} (f_{i+k,j+l})}_{\text{濃度値の中央値}} \quad \leftarrow \text{孤立した窪みや突起を除去}$$

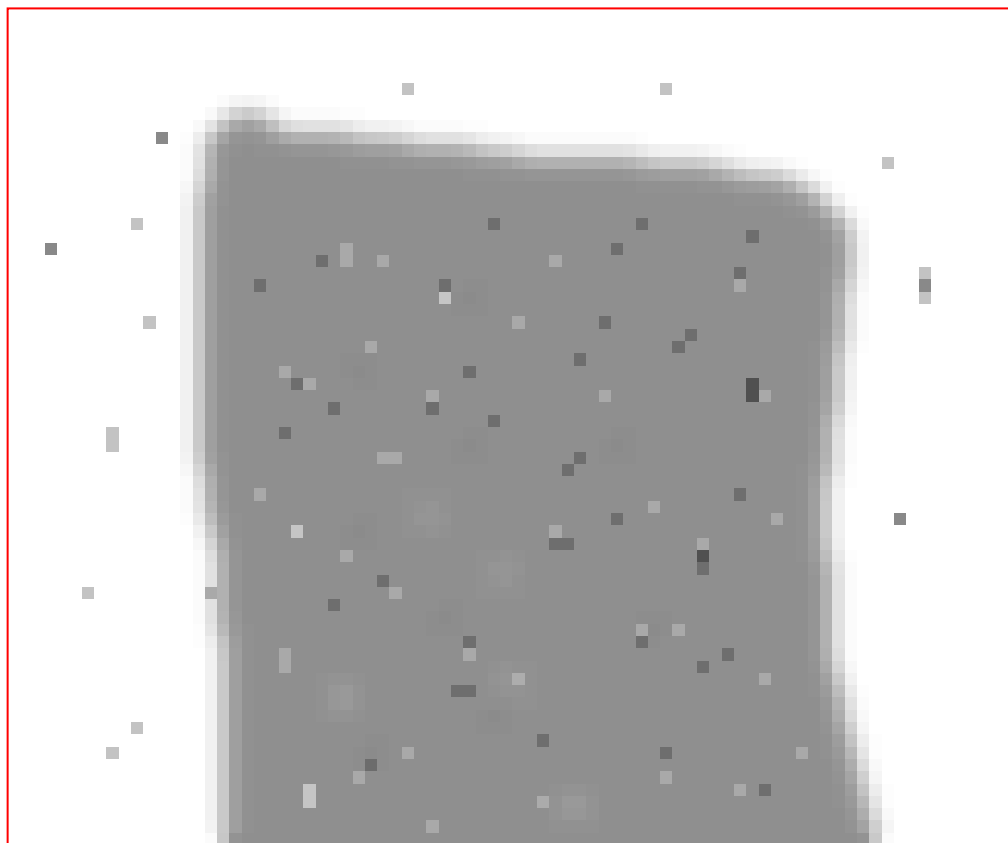
濃度値の中央値

d) エッジ保存平滑化

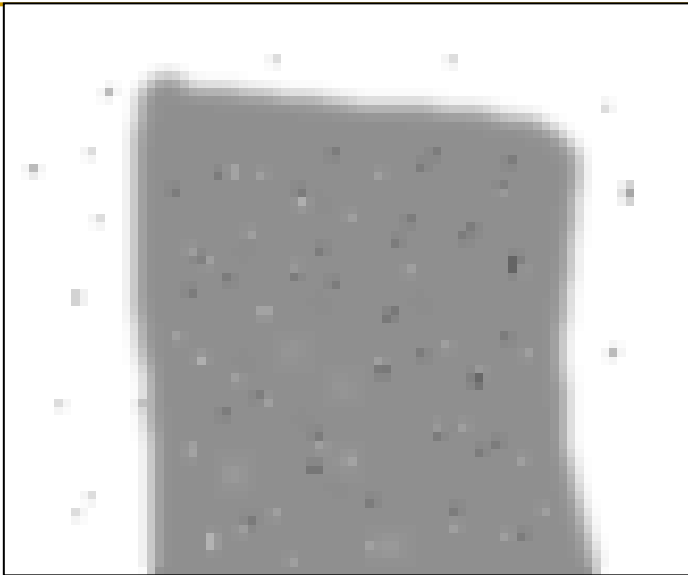
特徴

← エッジを避けて平滑化する

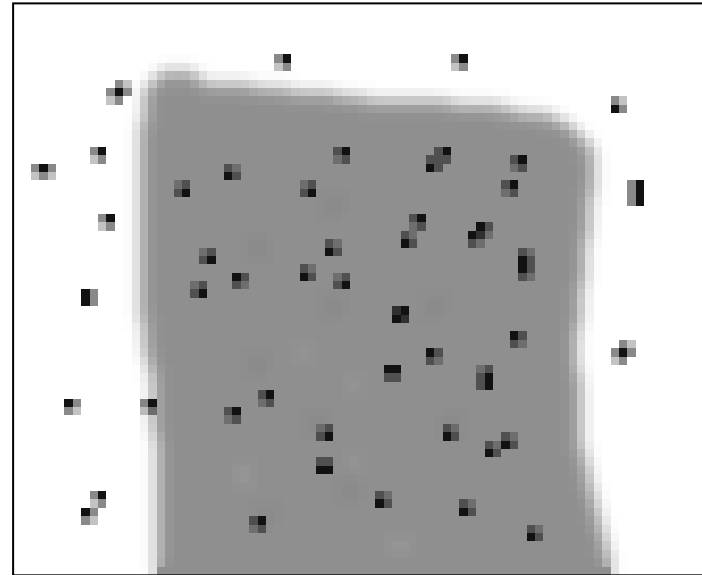




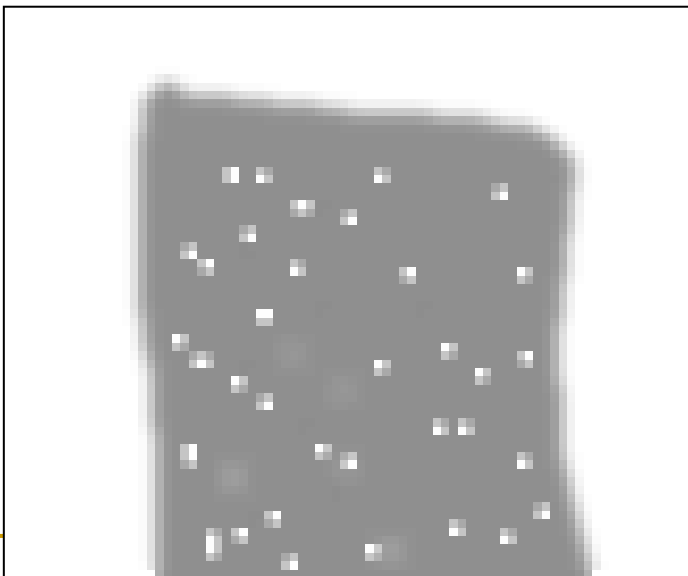
入力画像に加法的なノイズ



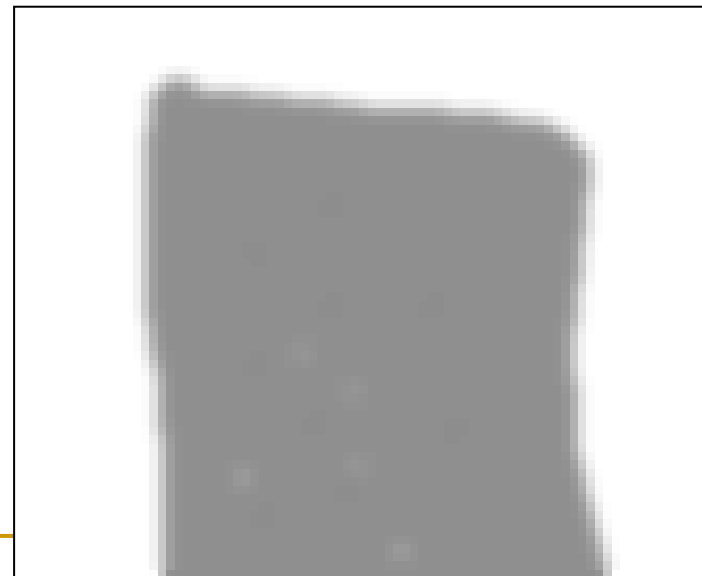
一様重み



最大値



最小値



メディアン



線形差分フィルタ

用途: エッジ、線分の強調

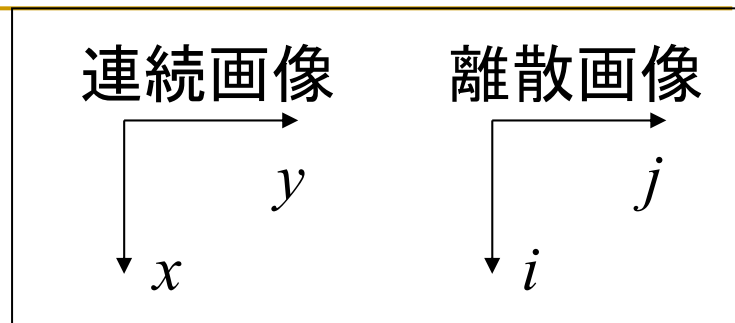
エッジの種類: ステップエッジ、ランプエッジ、ルーフエッジ、リッジ

原理: 微分演算

- a) 1階差分
エッジの部分が強調される。方向性あり
- b) 2階差分
エッジの両側が強調される。方向性あり
- c) グラディエント
エッジの傾きと方向が得られる
- d) ラプラシアン
方向性のない2階差分

線形差分フィルタ

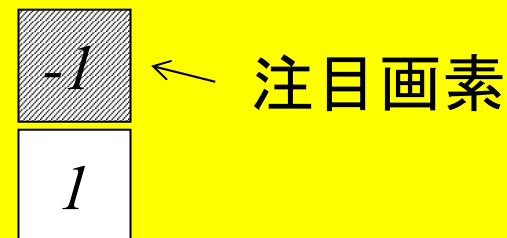
a) 一階差分



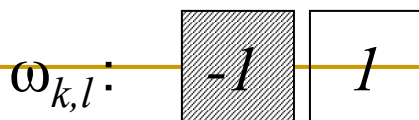
$$g_{i,j} = f_{i+1,j} - f_{i,j} \cong \frac{\partial f}{\partial x} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x}$$

$$+) \frac{\begin{array}{|c|c|} \hline f_{i,j} & \times (-1) \\ \hline f_{i+1,j} & \times (1) \\ \hline \end{array}}{g_{i,j}} \quad \text{重み係数 } \omega_{k,l}$$

$\omega_{k,l}$ の簡易表記法



$$g_{i,j} = f_{i,j+1} - f_{i,j} \cong \frac{\partial f}{\partial y} = \lim_{\Delta y \rightarrow 0} \frac{f(x, y + \Delta y) - f(x, y)}{\Delta y}$$



b) 2階差分

$$g_{i,j} = f_{i+1,j} - 2f_{i,j} + f_{i-1,j} \cong \frac{\partial^2 f}{\partial x^2}$$

$$\left(= \underbrace{(f_{i+1,j} - f_{i,j})}_{\text{1階差分}} - \underbrace{(f_{i,j} - f_{i-1,j})}_{\text{1階差分}} \right)$$

1階差分

$\omega_{k,l}:$

$$\begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$$

$$g_{i,j} = f_{i,j+1} - 2f_{i,j} + f_{i,j-1} \cong \frac{\partial^2 f}{\partial y^2}$$

$\omega_{k,l}:$

$$\begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$$

c) グラディエント (HOG, Histogram of Oriented Gradient, 使われる)

$$\nabla f_{i,j} = (f_{i+1,j} - f_{i,j}, f_{i,j+1} - f_{i,j}) \cong \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right)$$

$$\omega_{k,l}: \begin{array}{|c|} \hline -1 \\ \hline 1 \\ \hline \end{array} \quad \begin{array}{|c|c|} \hline -1 & 1 \\ \hline \end{array}$$

ベクトル: 輝度勾配の大きさと方向

d) ラプラシアン

$$\nabla^2 f_{i,j} = f_{i+1,j} + f_{i-1,j} + f_{i,j+1} + f_{i,j-1} - 4f_{i,j} \cong \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

$$= (f_{i+1,j} - 2f_{i,j} + f_{i-1,j}) + (f_{i,j+1} - 2f_{i,j} + f_{i,j-1})$$

$$\omega_{k,l}: \begin{array}{|c|} \hline 1 \\ \hline -2 \\ \hline 1 \\ \hline \end{array} \quad \begin{array}{|c|c|c|} \hline 1 & -2 & 1 \\ \hline \end{array} \quad \begin{array}{|c|} \hline 1 \\ \hline 1 & -4 & 1 \\ \hline 1 \\ \hline \end{array}$$

問3

A.画像 `Ipimage *image`に, 濃淡8bitの画像がはいっている. (幅: W 画素, 高さ: H 画素) (C言語)

1. しきい値処理 (しきい値: T で, $f(i,j) < T$ ならば0, その他は1)で `output [j][i]`を出力とするプログラムを書きなさい

2. 横方向に延びるエッジを強調する1階差分 (横方向 = j 方向) のプログラム (コア部分) を書きなさい

B. ラプラシアンフィルタの中核となる式を書きなさい

非線形差分フィルタ

a) 方向性一階差分

$$g_{i,j} = \max(f_{i+1,j} - f_{i,j}, f_{i,j+1} - f_{i,j})$$

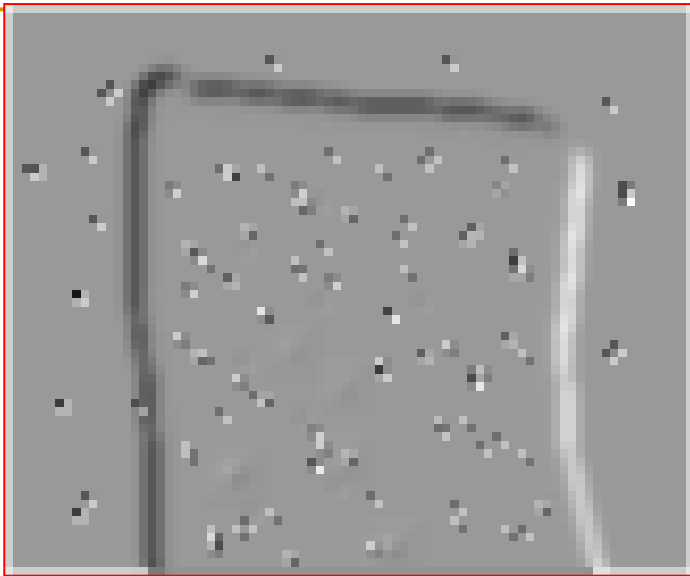
b) 方向性二階差分

$$g_{i,j} = \max(f_{i-1,j} - 2f_{i,j} + f_{i+1,j}, f_{i,j-1} - 2f_{i,j} + f_{i,j+1})$$

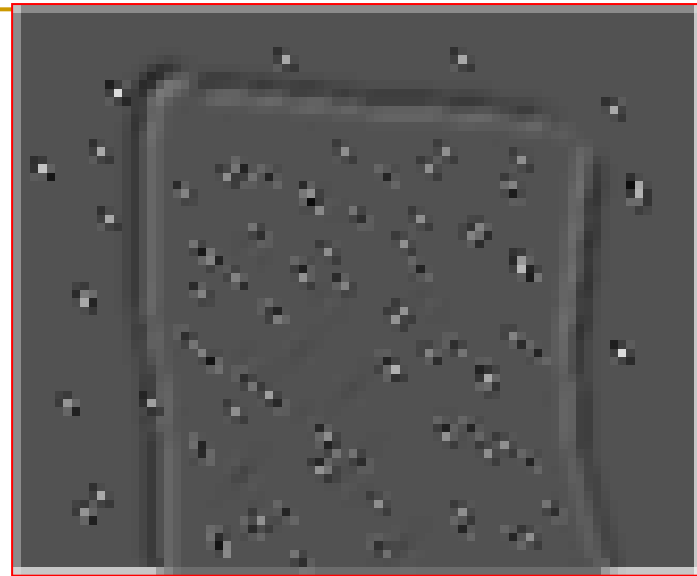
c) Rangeフィルタ

$$g_{i,j} = \max_{(k,l) \in \eta_{i,j}} (f_{i+k,j+l}) - \min_{(k,l) \in \eta_{i,j}} (f_{i+k,j+l})$$

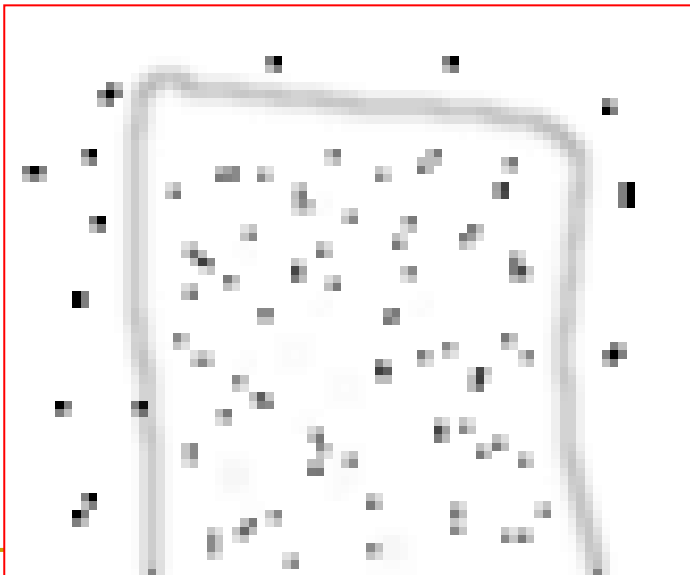




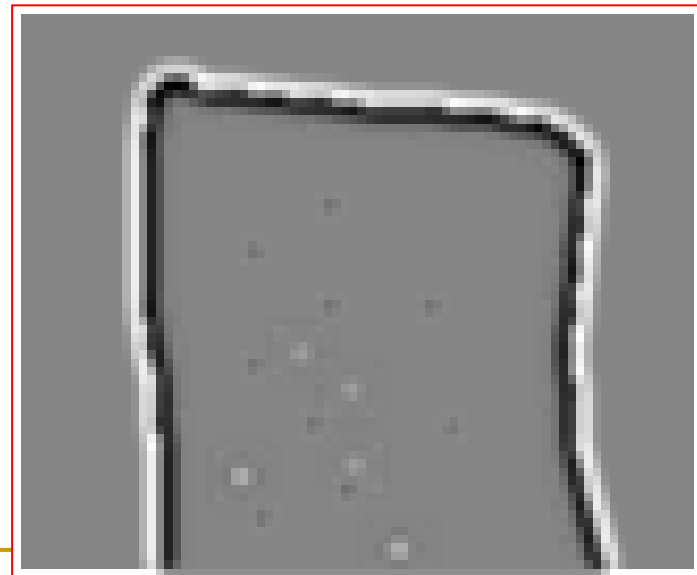
斜め方向一階差分



斜め方向二階差分



レンジフィルタ



メディアン+ラプラシアン

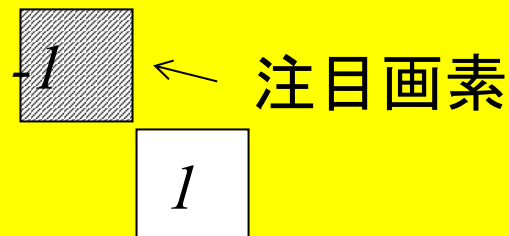
a) 方向性一階差分

$$g_{i,j} = f_{i+1,j+1} - f_{i,j}$$

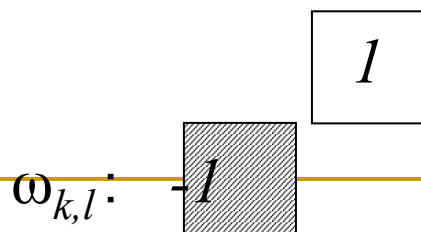
$$\begin{array}{r} +) \begin{array}{|c|c|} \hline f_{i,j} & \times (-1) \\ \hline f_{i+1,j+1} & \times (1) \\ \hline \end{array} \\ \hline g_{i,j} \end{array}$$

重み係数 $\omega_{k,l}$

$\omega_{k,l}$ の簡易表記法

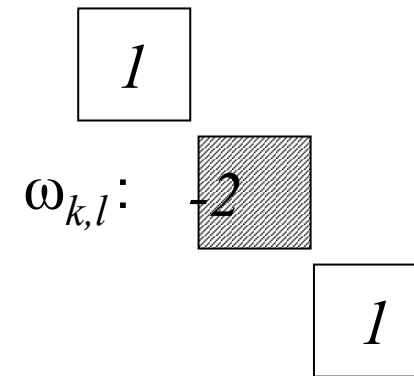


$$g_{i,j} = f_{i-1,j+1} - f_{i,j}$$



b) 方向性2階差分

$$g_{i,j} = f_{i+1,j+1} - 2f_{i,j} + f_{i-1,j-1}$$



エッジ検出アルゴリズムとしての 差分フィルタ

- 1次差分値の絶対値大→エッジ(図形輪郭)
- 他のよく知られたフィルタ(x方向(図)とその90度回転)

□ Prewitt(3x3)

-1	0	1
-2	0	2
-1	0	1

$\omega_{k,l}:$

-1	0	1
-1	0	1
-1	0	1

□ Sobel $\omega_{k,l}:$

- 平滑化+差分

□ Roberts

$$\sqrt{\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}^2 + \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}^2}$$

エッジ検出フィルタ(つづき)

■ Canny エッジフィルタ手順

1. 平滑化(ガウシアン)
2. 微分(1階差分)
3. 微分値の大きさと傾きの計算
4. 尾根の抽出(非極値の削除)
5. 尾根の接続(ヒステリシスしきい値処理)



問4

次の画像の中心画素(網目)の次の値を求めよ.

(1) 3x3平滑化フィルター(平均値)

(2) 3x3ラプラシアンフィルタ

(3) 縦方向2次差分フィルタ

(4) 最大値フィルタ

(5) メディアンフィルタ

(6) Sobelフィルタ(縦エッジ抽出)

(7) Sobelフィルタ(横エッジ抽出)

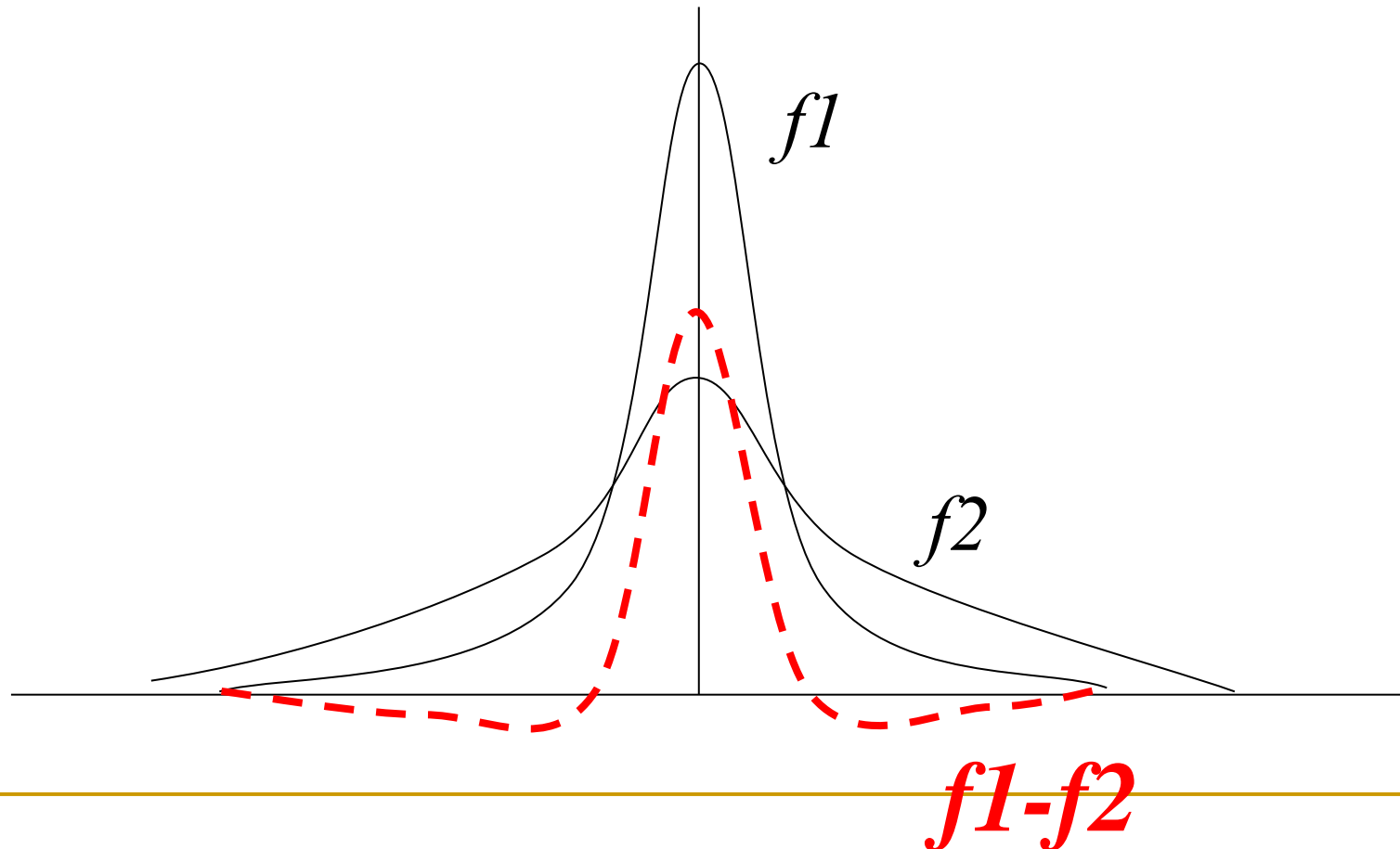
1	2	3
6	5	4
3	2	1

—

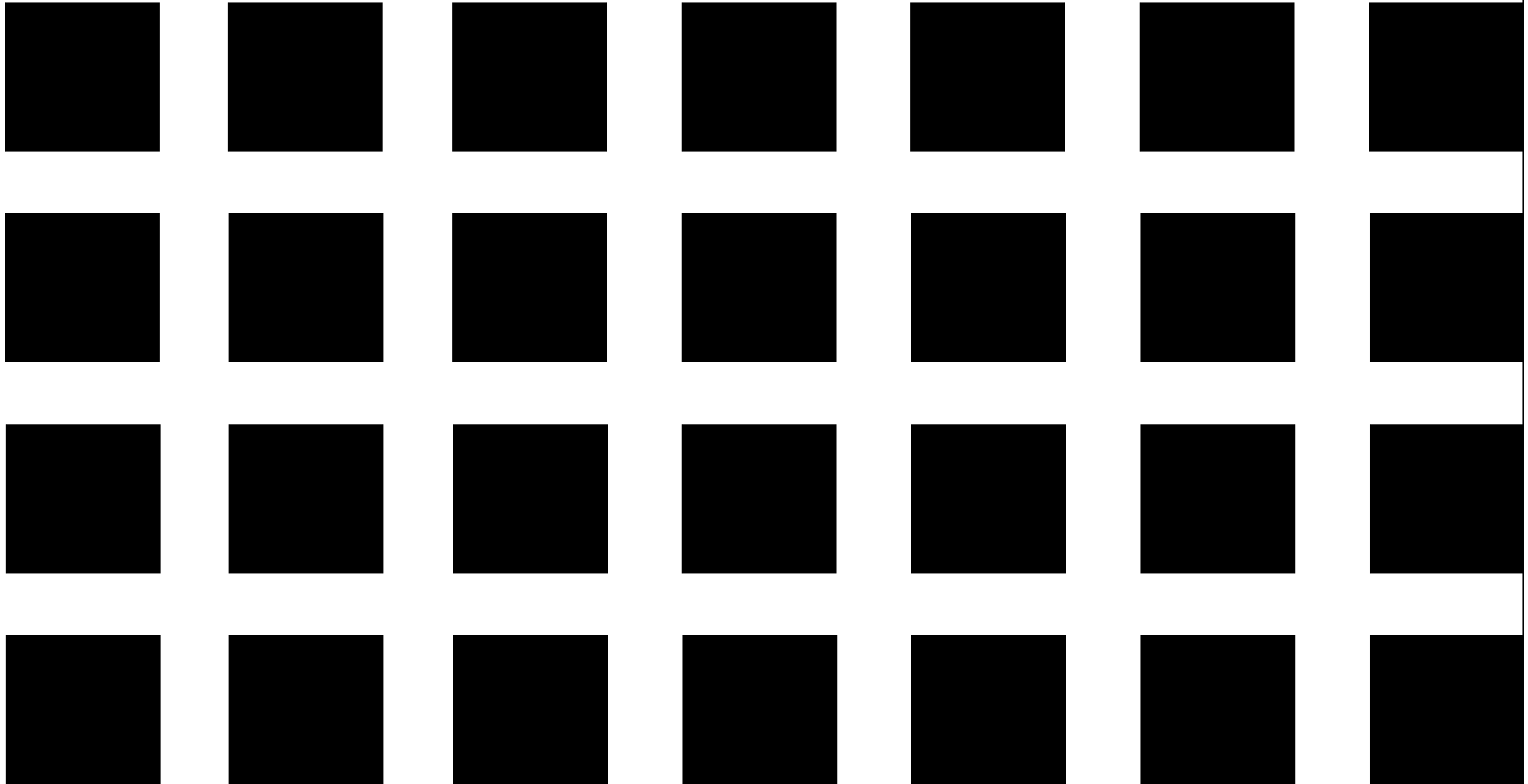
DOG関数(difference of Gaussian)

(Marr-Hildrethオペレータ)

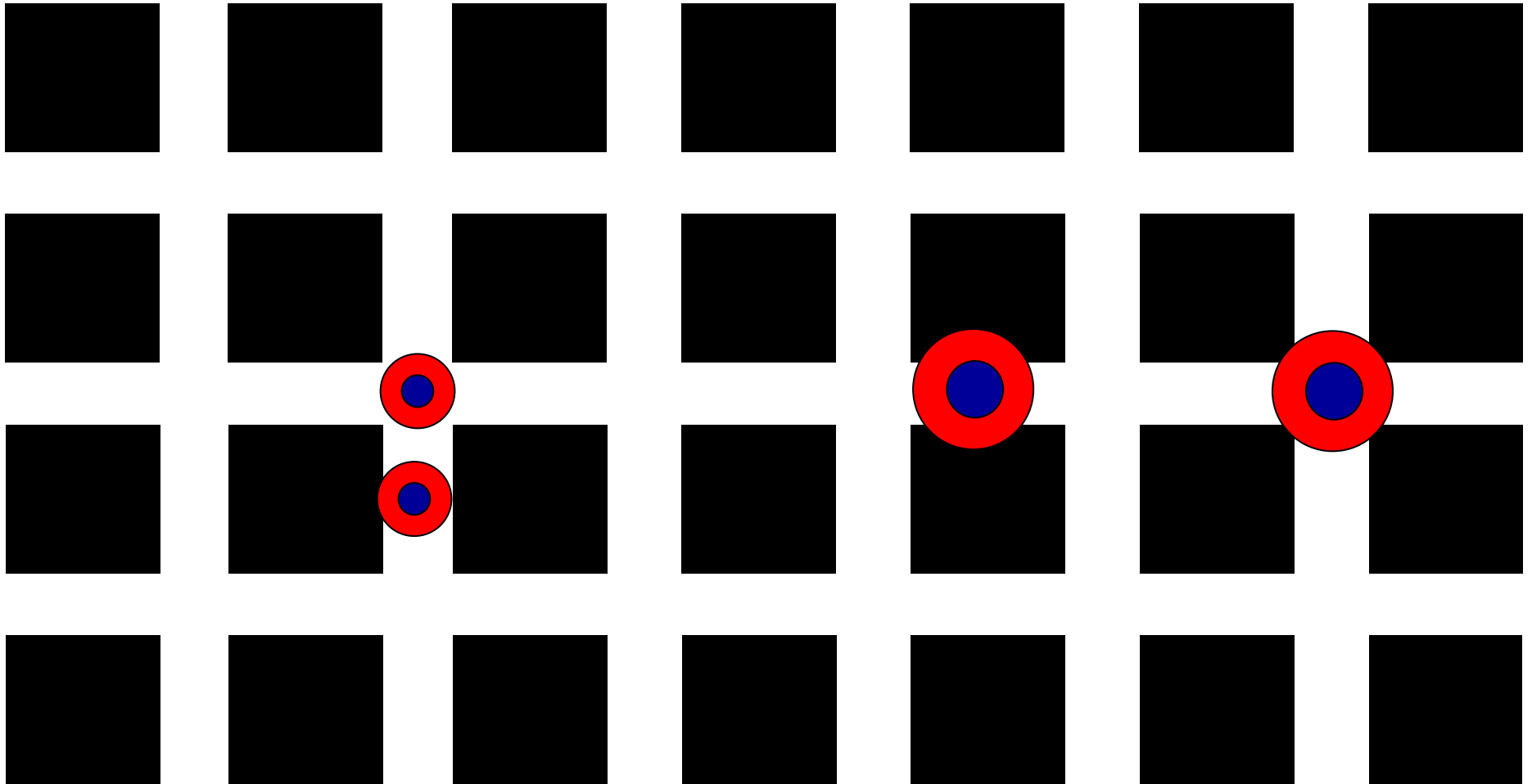
SIFT特徴(Shift-Invariant Feature Transform)で使われる。



(Hermann Grid)



(Hermann Grid) 中心窩と周辺視



演習1

学籍番号

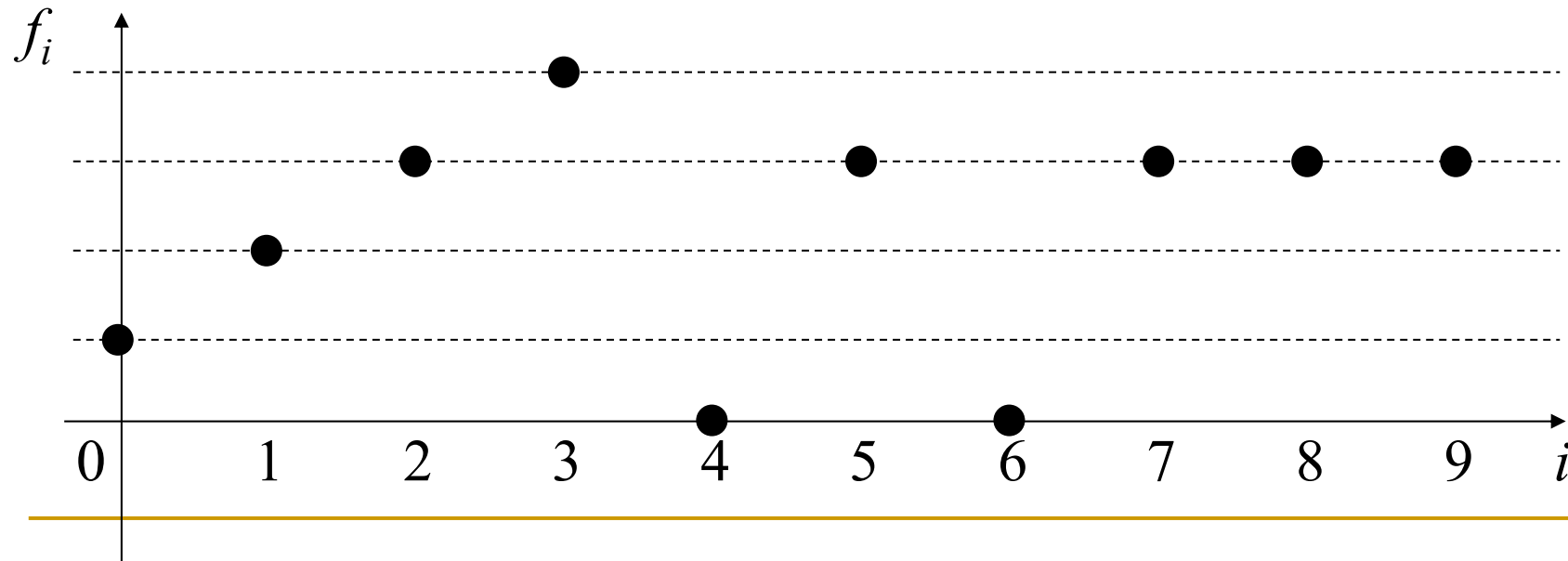
氏名

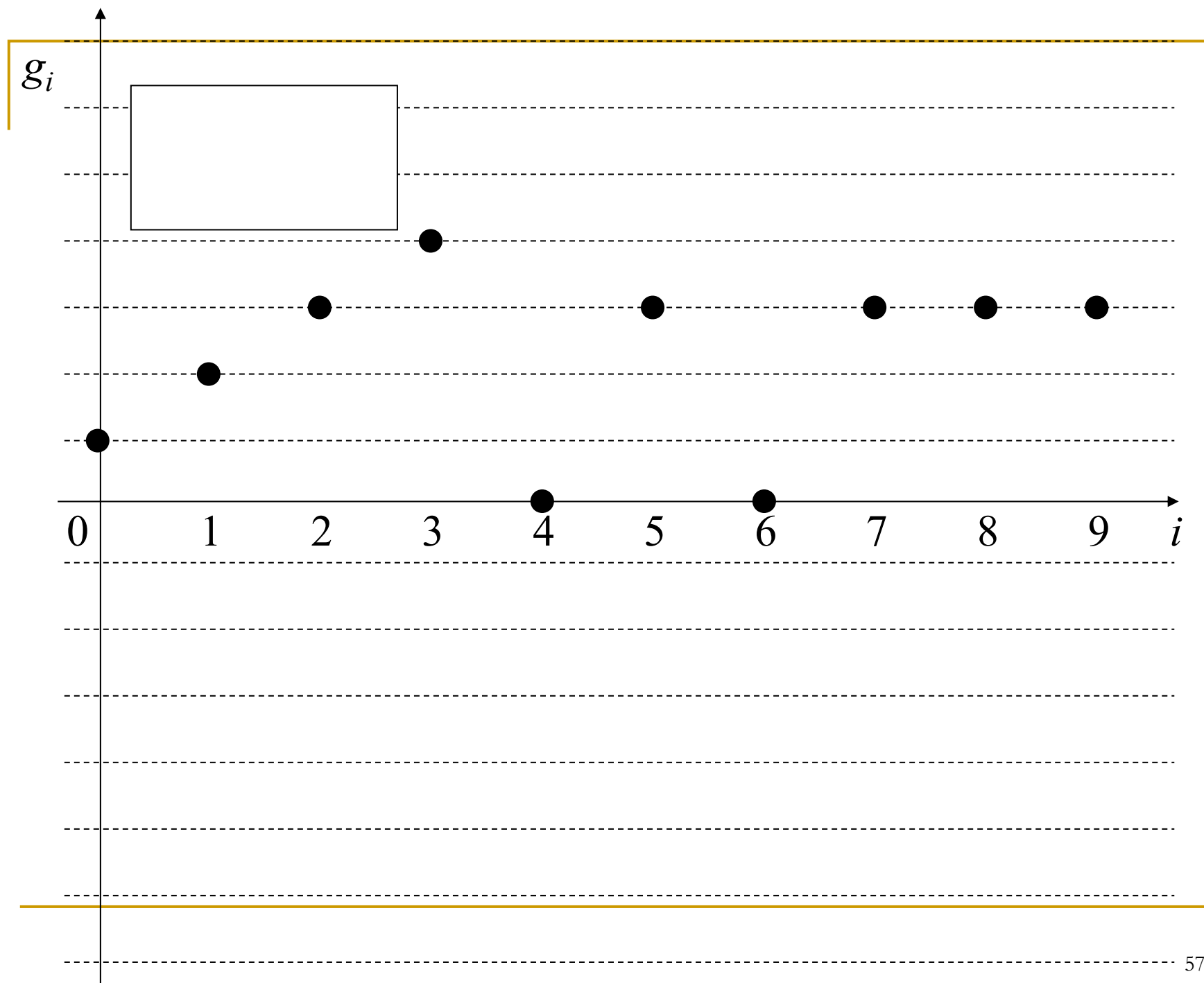
以下の一次元信号に対して,

- ・差分: 一階差分, 二階差分
- ・平滑化: 一様重み, Max, Min, Median

(全てマスクサイズ=3. 一様重みの係数 $c=1/3$)

を適用した結果を示せ. ただし, 縁の信号は $f_i = f_0$ ($i < 0$),
 $f_i = f_9$ ($i > 9$)とし, 濃度値の再量子化は小数第一位を四捨五入して行う.



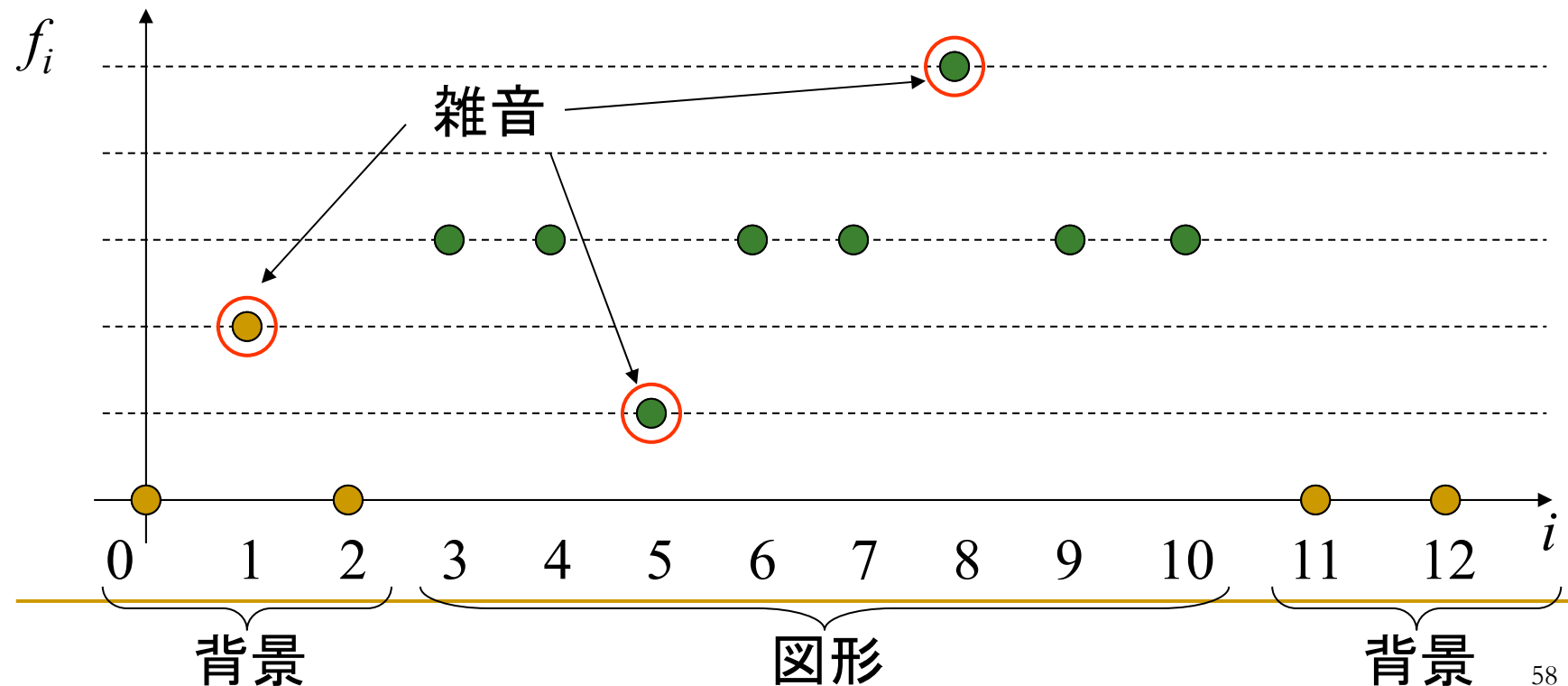


演習2(11／1)

学籍番号

氏名

濃度値が一様な図形と背景に対して矢印の位置に雑音を加わっていたとする. この時, これらの雑音除去に最も適しているフィルタを, 演習1の一樣重み, Min, Max, Medianの中から選び, その根拠を述べよ. ただし, 縁の信号の扱い方や濃度値の再量子化の方法は前問と同じとする.



局所処理の機能による分類

(1) 平滑化フィルタ

近傍領域内の濃淡を滑らかにする

(2) 差分フィルタ

局所的な濃淡変化の大きい部分を検出・強調

(3) 局所パターンマッチング

局所領域の濃淡分布の標準形との類似性評価

(4) 局所統計量フィルタ

近傍領域内の統計量をもとめる

(統計量により他の分類と同じ働きをすることもある)

(5) モルフォロジーフィルタ

入力画像の局所領域内のパターンとの包含関係に基づく
集合演算で定義. パターンマッチングに類似. 整形

1, 3 モルフォロジーフィルタ

入力画像Fと適当な構造要素 $B = \{b_{ij} = 0\}$

dilation ($F \oplus B$): $g_{ij} = \max_{(p,q) \in B} \{f_{pq} + b_{s_{p-i,q-j}}\}$

erosion ($F \ominus B$): $g_{ij} = \min_{(p,q) \in B} \{f_{pq} - b_{s_{p-i,q-j}}\}$

$b_{s_{ij}}$ は構造要素の原点对称図形

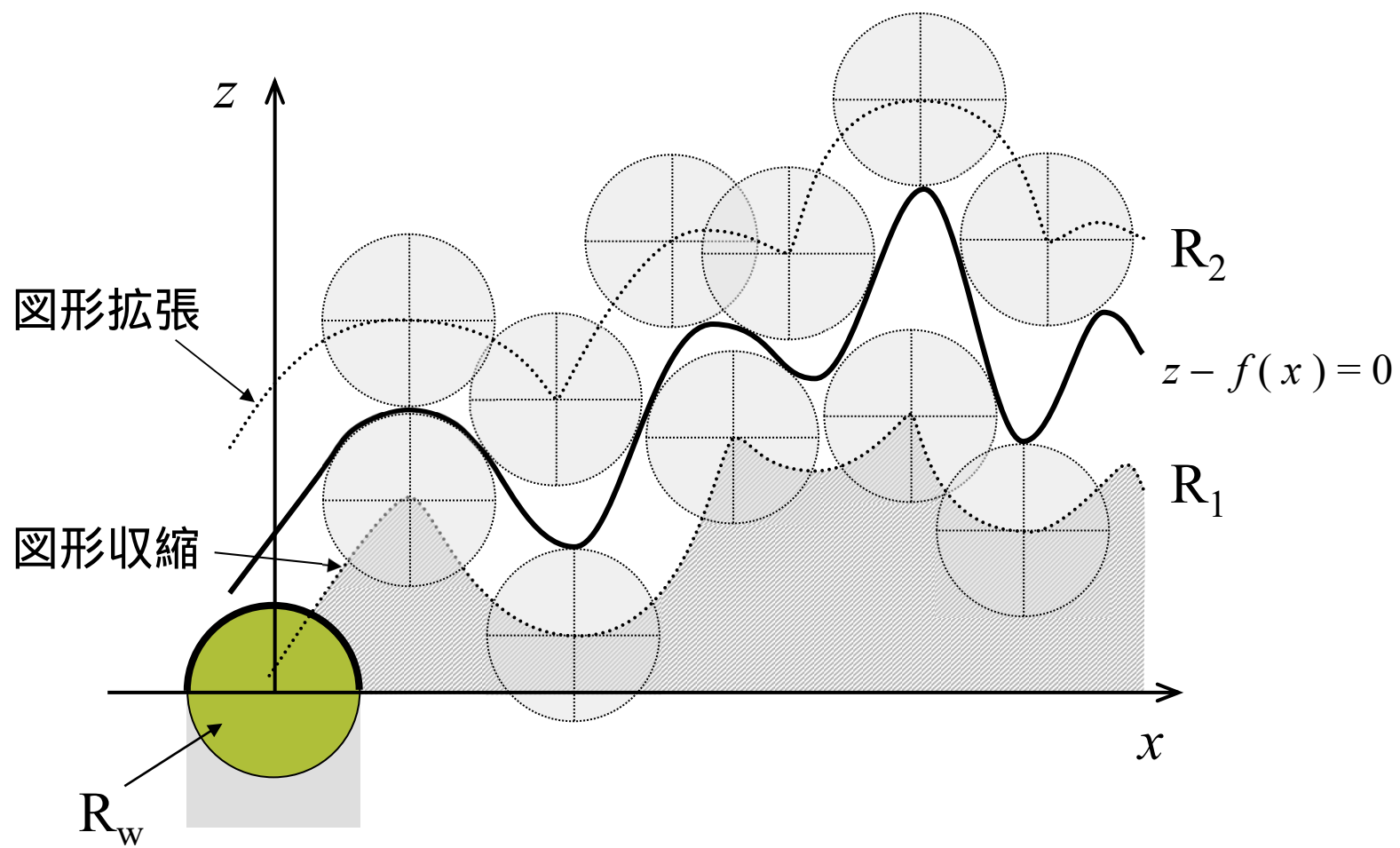
$b_{ij} = 0$ の場合が多い

dilate(拡張), erode(浸食)

dilationとerosionの組み合わせ

closing ($F \oplus B \ominus B$)

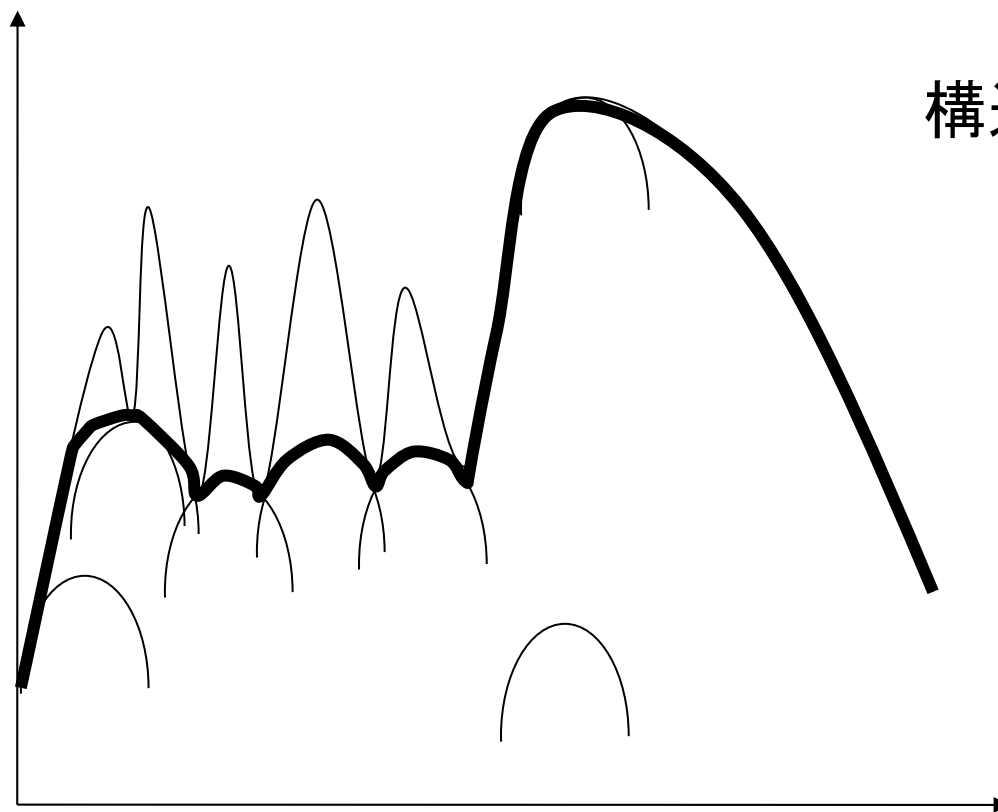
opening ($F \ominus B \oplus B$)



モルフォロジーフィルタの組み合わせ

Tophat変換: $F \leftarrow (F \ominus B) \oplus B$

$f(x)$



構造要素でopening

x

モルフォロジーフィルタの組み合わせ

$f(x)$

Tophat変換: $F \leftarrow (F \ominus B) \oplus B$

エッジ抽出などに用いられる

構造要素でopeningしたものを
元の画像から引く

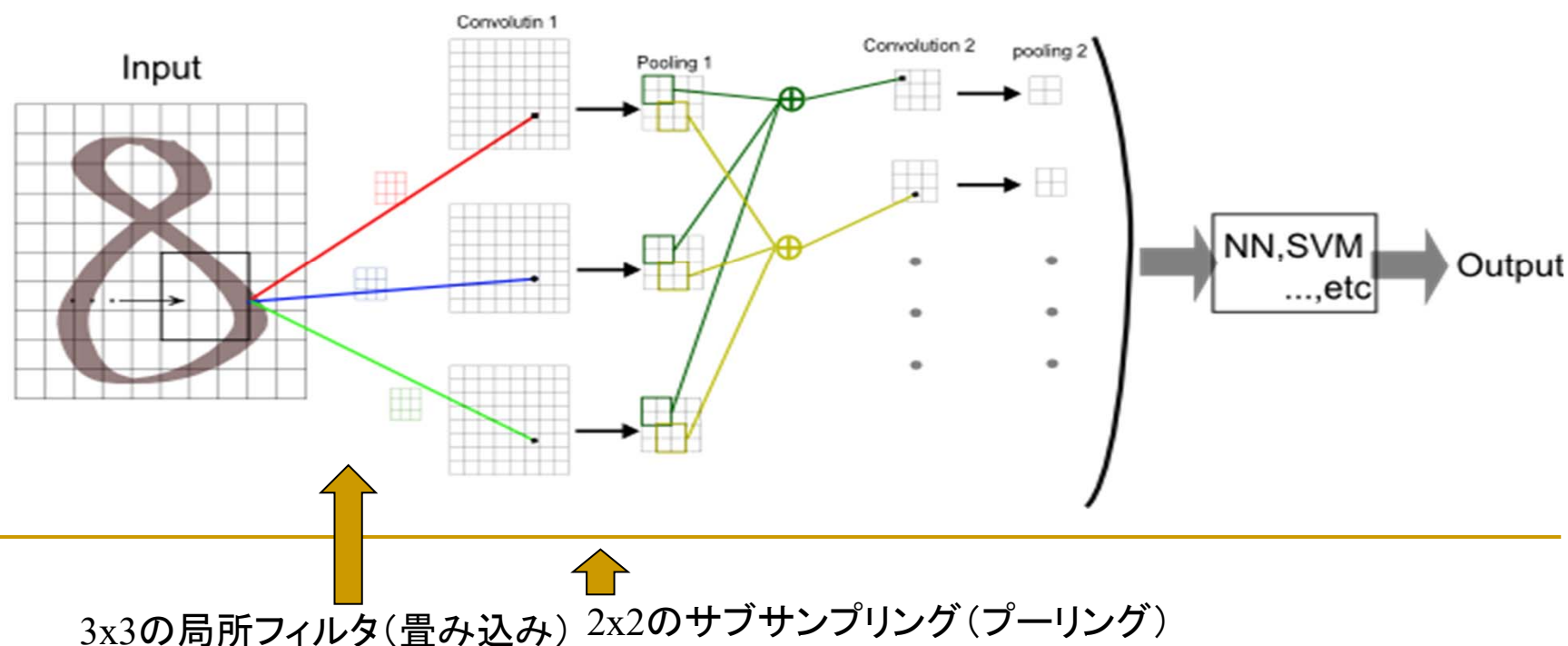
x

ちょっとしたポイント

- 局所演算で、周辺画素の計算は定義できないのでは？
- なぜガウス型平均値フィルタがよいのか？
- 計算の結果が負になる場合、画像形式は？
- カラー画像の場合、RGBそれぞれ独立にフィルタをかけるのか？
- 線形フィルタの良さはどこに？
- プログレッシブ符号化とは？

深層学習と局所演算

- 多層ニューラルネットワーク
 - ニューラルネットワークは人間の神経網を模倣
 - 計算量が膨大
 - 大量の学習データが必要 (ImageNetなど)
- CNNは画像処理の「多段の局所演算＋最終段のNN」に相当する。局所演算のオペレータ(重み)を機械学習するところが異なる(表現学習)。



Lu CunのLeNet-5

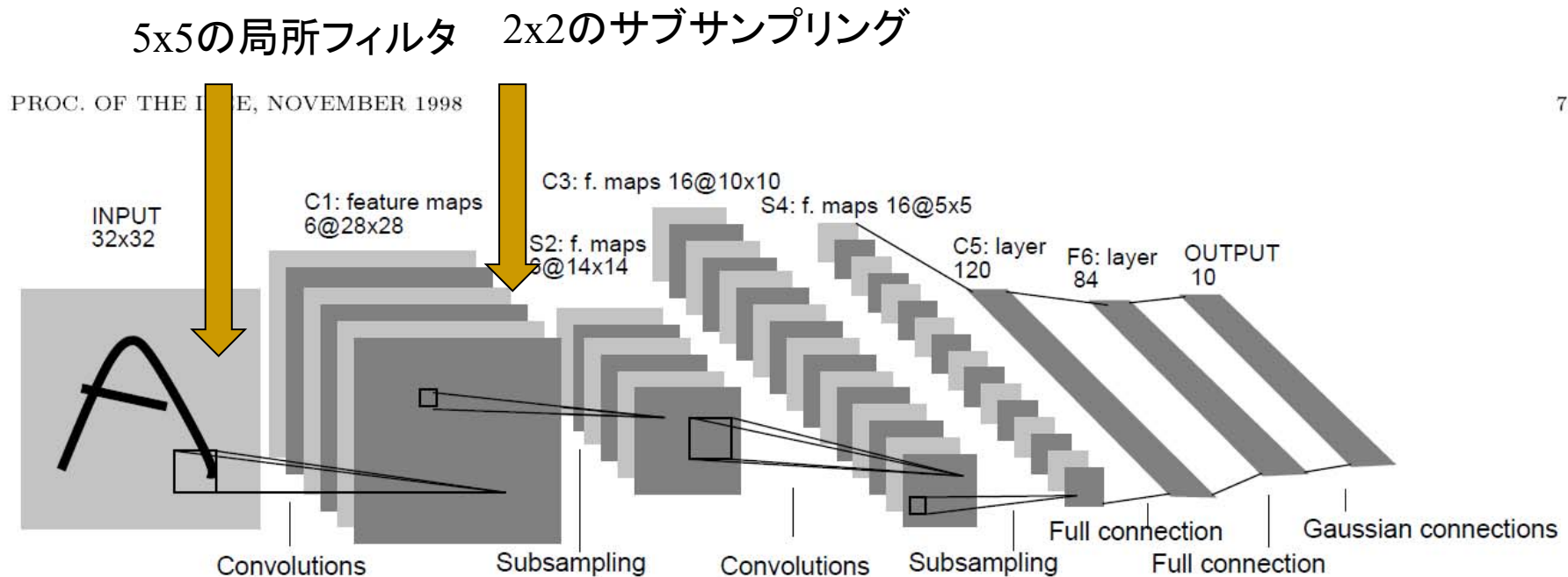


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

- PROC. OF THE *IEEE*, NOVEMBER 1998. 1. *Gradient-Based Learning Applied to Document. Recognition.* Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner

1段, 2段目の学習済み畳み込みフィルタの可視化

Visualizing and Understanding Convolutional Networks

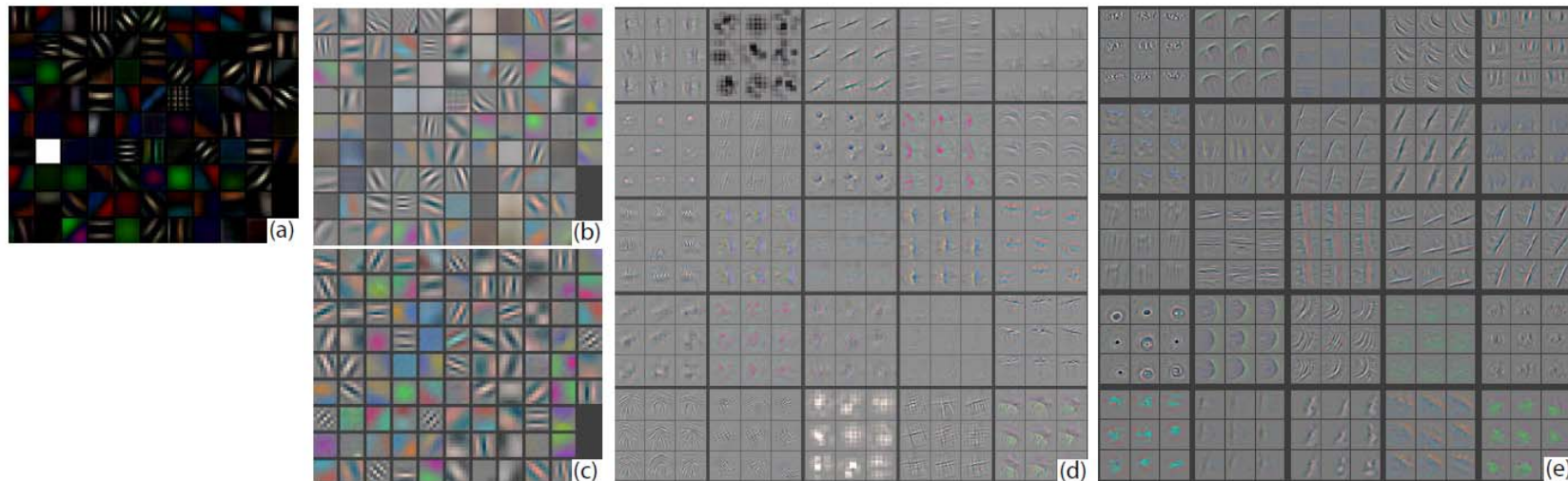


Figure 6. (a): 1st layer features without feature scale clipping. Note that one feature dominates. (b): 1st layer features from (Krizhevsky et al., 2012). (c): Our 1st layer features. The smaller stride (2 vs 4) and filter size (7x7 vs 11x11) results in more distinctive features and fewer “dead” features. (d): Visualizations of 2nd layer features from (Krizhevsky et al., 2012). (e): Visualizations of our 2nd layer features. These are cleaner, with no aliasing artifacts that are visible in (d).

[arxiv.org](https://arxiv.org/abs/1311.2901)より

Visualizing and understanding convolutional networks

MD Zeiler, R Fergus - European Conference on Computer Vision, 2014 - Springer

深層学習(2)

■ 事件

- ILSVRC2012(ImageNet Large Scale Visual Recognition Competition (ILSVRC)、一般物体認識の画像処理コンペ)でそれまでの手法より10%以上誤り率の低い結果を出した。

Model	Top-1 (val)	Top-5 (val)	Top-5 (test)
<i>SIFT + FVs [7]</i>	—	—	26.2%
1 CNN	40.7%	18.2%	—
5 CNNs	38.1%	16.4%	16.4%
1 CNN*	39.0%	16.6%	—
7 CNNs*	36.7%	15.4%	15.3%

Table 2: Comparison of error rates on ILSVRC-2012 validation and test sets. In *italics* are best results achieved by others. Models with an asterisk* were “pre-trained” to classify the entire ImageNet 2011 Fall release. See Section 6 for details.

深層学習(3)

- 解説スライド(「slideshare ディープラーニング」で検索)
http://www.slideshare.net/nlab_utokyo/deep-learning-40959442

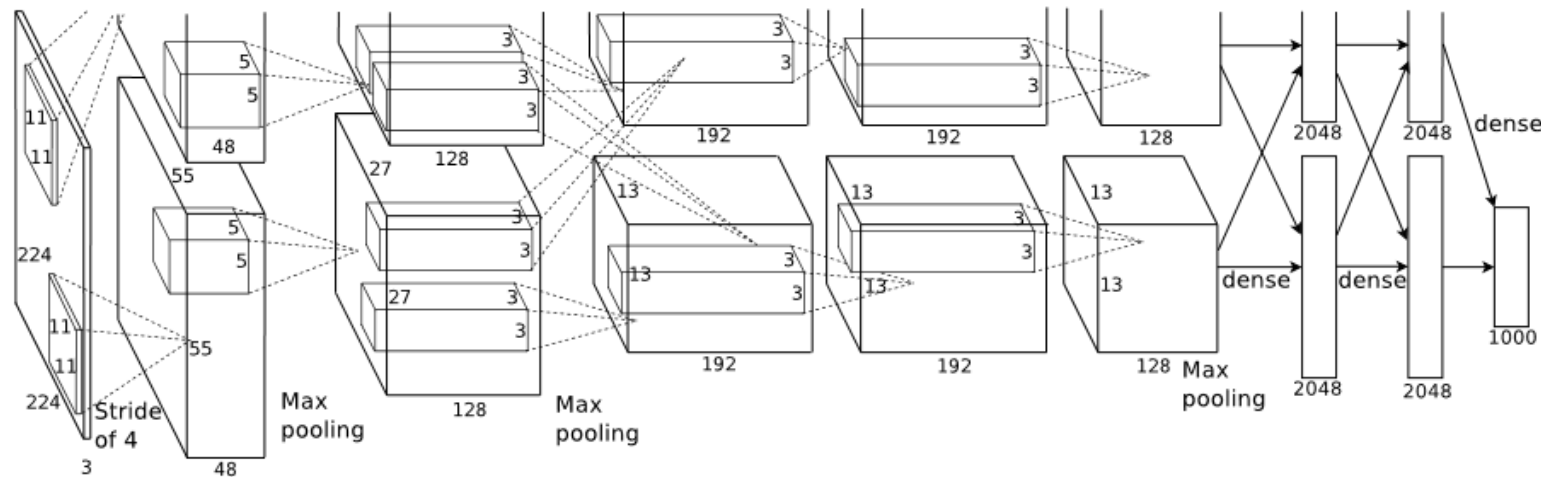


Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

深層学習(4) 局所フィルターとの関係 まとめ

- CNNにおける畳み込み層には, 歴史的に提案されてきた局所フィルターが多く含まれている
 - クラフティング(工夫)したものが自動に作られた
 - 未知の局所フィルターが学習されているのは興味深い
- プーリング相も同様
→表現学習(特徴学習)と呼ばれる所以
- 全結合層は多層パーセプトロン

深層学習(5) トレンド

- ネットワークの多段化: パワーゲーム
- ネットワーク構成のデザイン(アーキテクチャ)の工夫による性能向上
- マルチモーダル化: 映像とテキストの同時学習
- 転移学習: 学習データの削減

2. トポロジ(位相)とジオメトリ(幾何)

■ トポロジ

- 図地判定
- 対象物の位置関係
- 対象物の数

■ ジオメトリ

- 対象物の大きさ
- 対象物の形

ジオメトリ

図形の幾何学的性質

■ 画像の モーメント特徴量

$$m_{ij} = \sum_{x,y \in R} x^i y^j f(x, y)$$

画像 f の $(i + j)$ 次モーメント

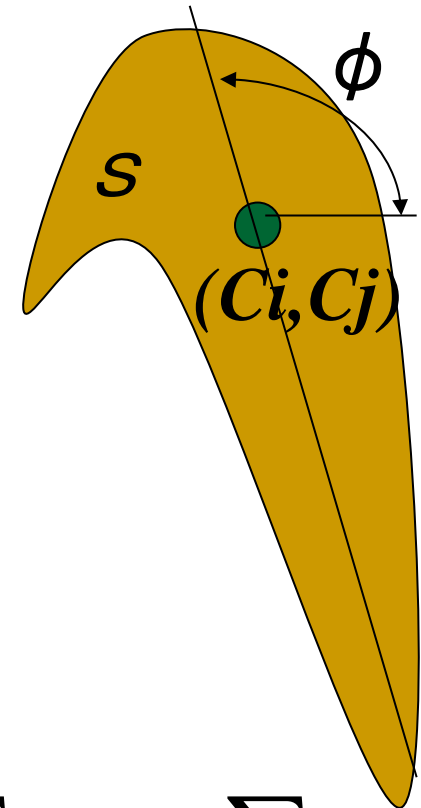
- 0次モーメント: 面積

$$m_{00} = \sum_{x,y \in R} f(x, y)$$

- 1次モーメント: 重心

$$C = (m_{10} / m_{00}, m_{01} / m_{00}) = \frac{1}{m_{00}} \left(\sum_{x,y \in CC} x f_{x,y}, \sum_{x,y \in CC} y f_{x,y} \right)$$

- 2次モーメント → 重心まわりの傾き



$$\tan^2 \phi + \frac{m_{20} - m_{02}}{m_{11}} \tan \phi - 1 = 0$$

慣性モーメント

■ 2次慣性モーメント

- 原点から

$$m_f = m_{20} + m_{02}$$

- 直線から

$$m_{\theta} =$$

慣性主軸

他の幾何学特徴

- 周囲長

- 細長さ

面積／縮退回数²(縮退フィルター:min)

最小外接矩形の長辺／短辺の比

- 円らしさ

- 周囲長²／面積

- 対称性

トポロジ(位相)

2. 1. 近傍

■ 2次元画像

- 4近傍 ($x \in N^{[4]}(y)$) : 注目画素と辺を共有する画素の集合
- 8近傍 ($x \in N^{[8]}(y)$) : 注目画素と点を共有する画素の集合

■ 3次元画像

- 6近傍 ($x \in N^{[6]}(y)$) : 注目画素と「面」を共有する画素の集合
- 18近傍 ($x \in N^{[18]}(y)$) : 注目画素と「面」を共有する画素の集合
- 26近傍 ($x \in N^{[26]}(y)$) : 注目画素と「面」を共有する画素の集合

演習3

- 近傍を図示せよ
 - 4近傍
 - 8近傍
 - 6近傍
 - 18近傍
 - 26近傍
 - 18近傍と26近傍を図示する(宿題).

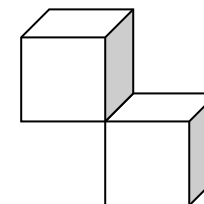
2. 2. 連結

- 図と地の関係を前提
- 2次元画像
 - 4連結
 - 8連結
- 隣接
- 連結成分

ルビンの壺

●2画素 x, y の隣接関係

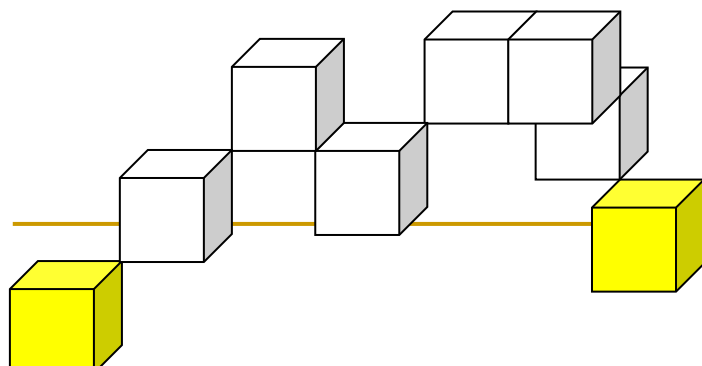
$$x \in N^{[k]}(y) \quad y \in N^{[k]}(x)$$



画素 x と y は互いに k 隣接する

18, 26隣接している2画素

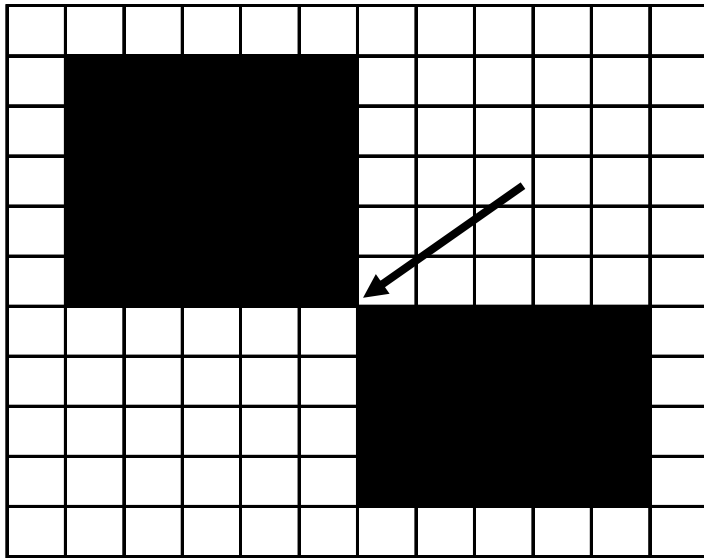
●同じ値を持つ2個の画素 x_1, x_2 に対して, それらと同じ値を持つ画素の系列 ($y_0 (= x_1), y_1, \dots, y_n (= x_2)$) が存在し, すべての $i (1 \leq i \leq n)$ に対して, y_{i-1} と y_i が k 隣接しているならば, 画素 x_1 と x_2 は k 連結の意味で連結している.



26連結している2画素

図形と背景の連結性

図形(1画素)と背景(0画素)の連結性を同時に議論する必要がある場合は矛盾を避ける必要がある.



2次元での例

図形を8連結で考えた場合は, 連結成分数は1
矢印の部分で背景が斜めに連結してはいけない.

→背景は4連結

表があるところには、
かならず裏がある

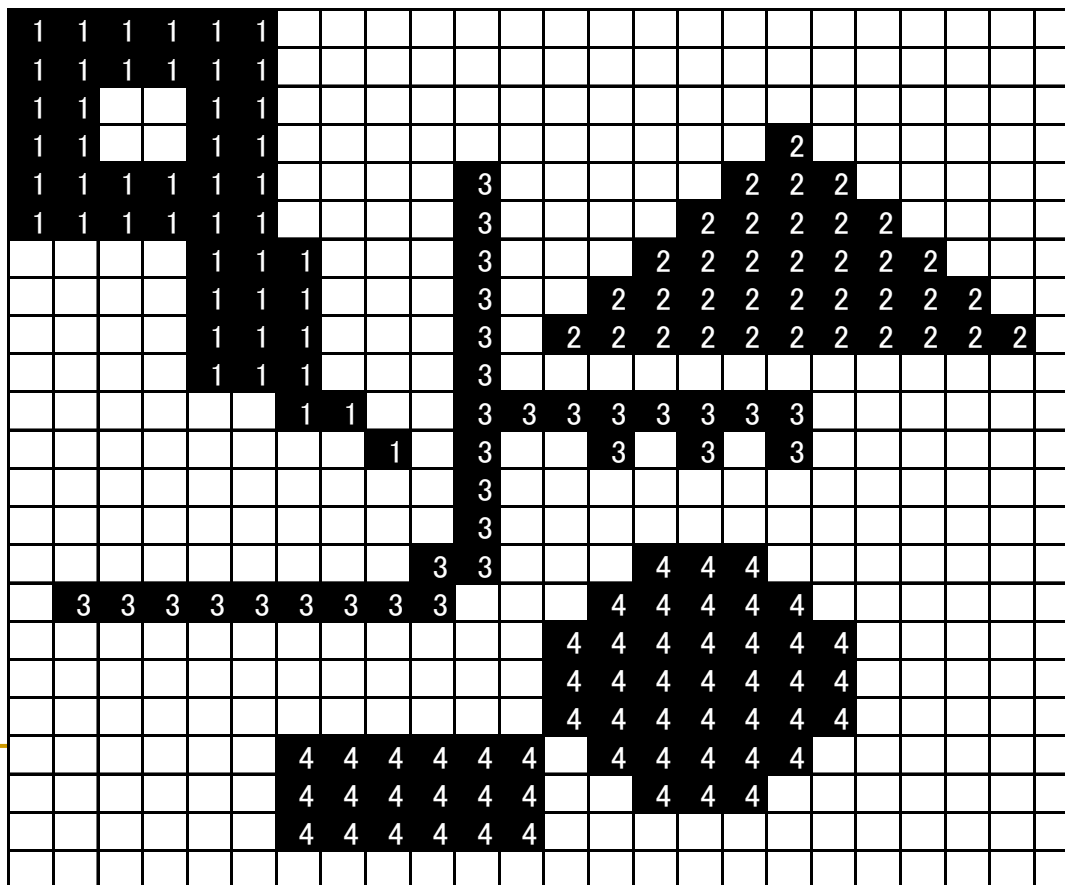
光があるところには必ず影(陰)がある

相対(双対): dual

連結成分 (connected component)

定義 任意の画素 x に対して, それと連結しているすべての画素集合

→ x を含む連結成分



2次元画像のラベル付け(ラベリング)

アルゴリズム1

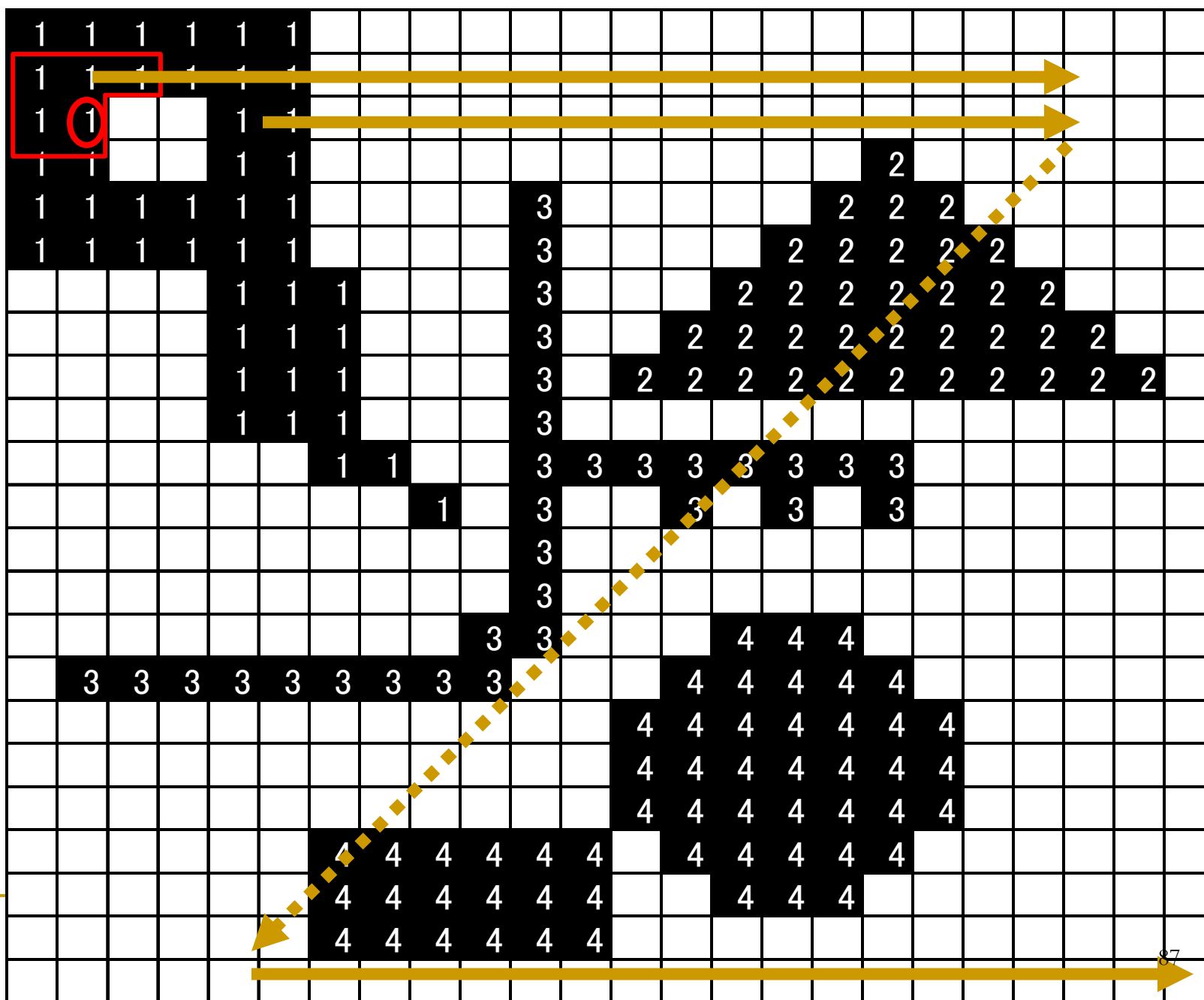
入力 $F = \{f_{ij}\}$,

出力(ラベル画像) $L = \{l_{ij}\}$, l_{ij} には各連結成分のラベルを表す正整数 λ が入る

ラベル表: $T(i)$ 作業用1次元配列

1) 初期設定 $\lambda \leftarrow 0$, 画素(2, 2)から走査開始

2) $f_{ij} = 1$ ならば 3)へ. $f_{ij} = 0$ ならば $l_{ij} \leftarrow 0$ として 4)へ



3) 近傍中で走査済みの画素(最大4個)のラベル値を lp とする. これらに対応するラベル表 $T(lp)$ が異なる n 個の値を持つとし, これを小さい順にソート L_1, L_2, \dots, L_n .

3-1) $n=0$ (走査済みの近傍がすべて背景)の場合 3-2)へ. $n=1$ (正值が一種類)の場合 3-3)へ. それ以外は 3-4)へ.

3-2) $\lambda \leftarrow \lambda + 1, T(\lambda) \leftarrow \lambda, l_{ij} \leftarrow \lambda$ 4)へ

3-3) $l_{ij} \leftarrow L_1$ 4)へ

3-4) $l_{ij} \leftarrow L_1, L_2, \dots, L_n$ なる γ に対して $T(\gamma) = L_1$

4) 全画素が終了したら 5)へ. その他の場合は次の対象画素に進み 2)へ

5) $T(\lambda)$ を通し番号にして, その最大値を記憶

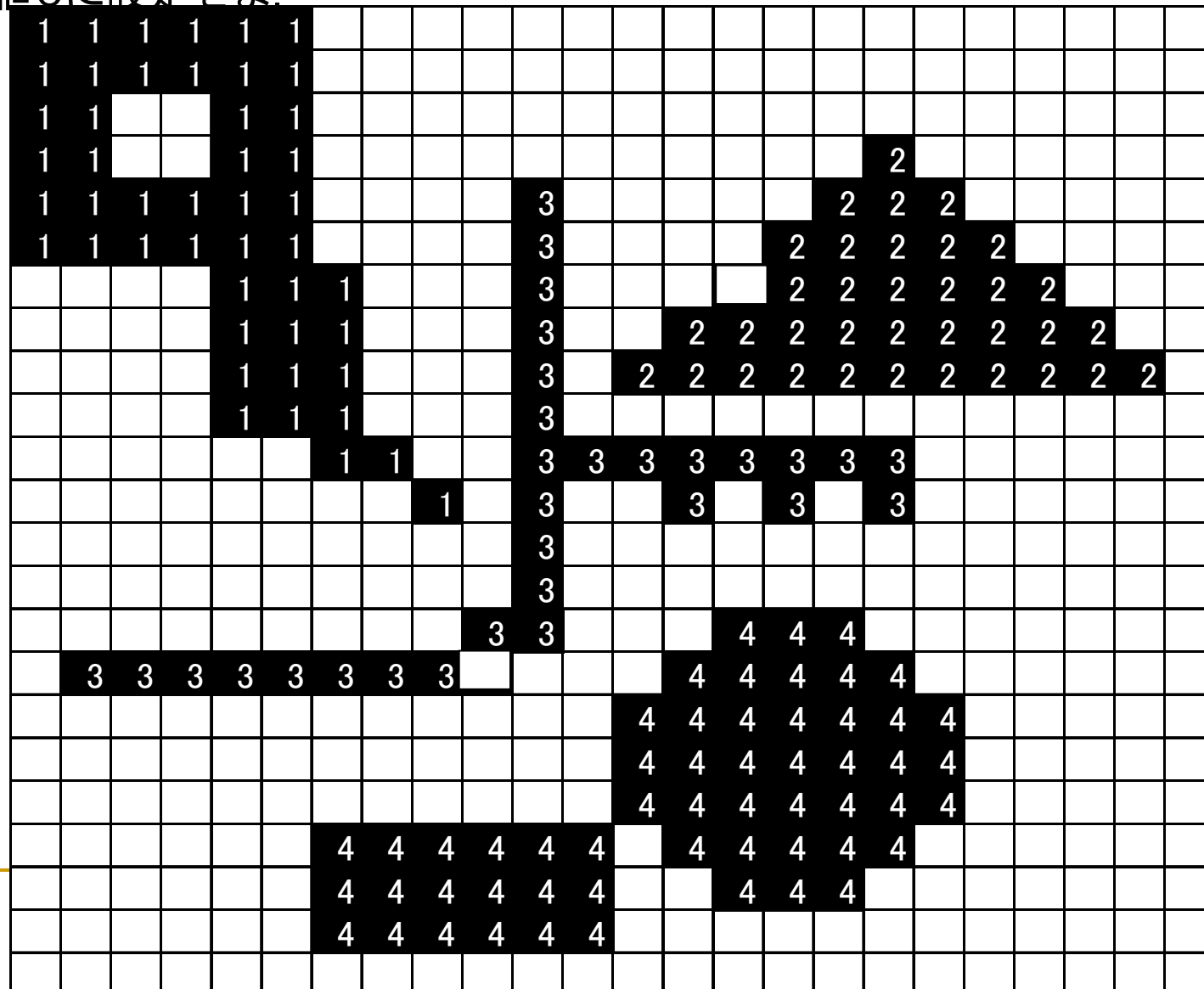
6) ラベルの付け直し

特徴

ラベルの付け直しが複雑

画像を2回走査するだけでよい

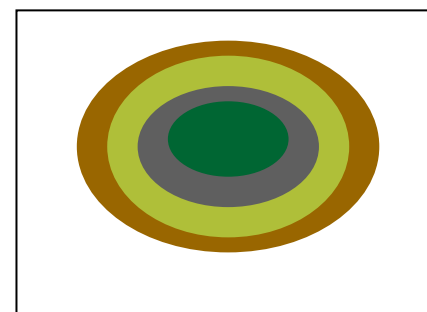
宿題: 次の画像をしきい値0.9で2値化し, それに対してアルゴリズム1のラベリング(8連結)を実行せよ. ただし, ステップ(5)の直前で停止し, その時点でのラベル画像Lとラベル表Tを別紙に表せ. 前処理として周囲1行1列は値0に設定せよ.



2. 3 距離

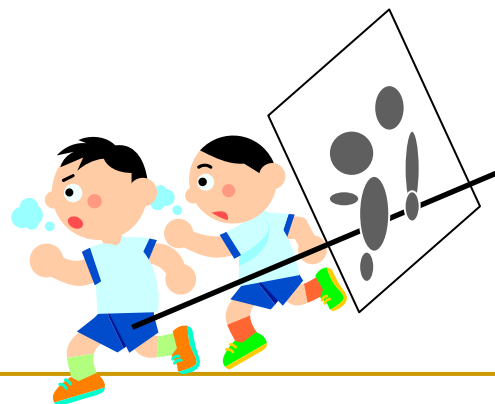
- 地(背景)との境界から図を構成する画素までの距離

- 距離画像



X 視点から物体までの距離

- 距離画像(range image)



距離関数

2(3)次元ベクトルの対からなる直積空間から非負実数値への写像で, 非負性, 同一性(反射律), 対称律, 三角不等式が成り立つものを距離関数(測度)とよぶ. (c.f. 同値関係)

可変近傍パス

4, 8(6, 18, 26)の固定近傍の系列でパスを構成する.

計算が単純な固定近傍でユークリッド距離に近づけることができる. ただし, 使用する近傍の系列を記憶する必要あり. 逐次処理が基本

距離変換

定義 距離変換

入力2値画像 $F = \{f_{ijk}\}$, から出力 $D = \{d_{ijk}\}$ への変換

$$d_{ijk} = \begin{cases} \min \{d[(i,j,k), (p,q,r); f_{pqr} = 0]\} & (i,j,k) \text{ s.t. } f_{ijk} = 1 \\ 0 & (i,j,k) \text{ s.t. } f_{ijk} = 0 \end{cases}$$

- ・図形上の各画素にそこから背景までの最短距離を入れる処理
- ・図形境界から一層ずつ削っていくとき, k回目に削られる画素に値kを代入

距離変換

入力2値画像 $F = \{f_{ijk}\}$, から出力 $D = \{d_{ijk}\}$ への変換

$$d_{ij} = \begin{cases} \min \{d[(i,j), (p,q); f_{pq}=0]\} & (i,j) \text{ s.t. } f_{ij}=1 \\ 0 & (i,j) \text{ s.t. } f_{ij}=0 \end{cases}$$

関数 d は固定近傍, 可変近傍距離やユークリッド距離をとる. ただし, ユークリッド距離の場合は, 図形境界から一層ずつ削っていく...という考えではできなくなる.

背景に接していても距離値が違える場合がある.

逆距離変換

入力距離値画像 $F = \{f_{ij}\}$, から $G = \{g_{ij}\}$ への変換

入力画像の各画素の距離値の情報から元の図形の形状を復元する

- ・入力画像の各点において、距離変換時の距離関数を用いた円(球)を配置し、それらの和集合を出力する変換と考えてもよい

スケルトン

入力画像 $F = \{f_{ij}\}$, その距離変換を $G = \{g_{ij}\}$ とする

定義1 G の部分集合で, 逆距離変換により元の図形 F を復元することができる

定義2 G の極大点集合

背景から距離1の画素集合を0にする処理を φ , 図形から距離1の画素集合を1にする処理を μ とする.

定義3 すべての k に対して, $\varphi^{k-1}(F)$ に含まれていて, $\mu(\varphi^k(F))$ に含まれないがその集合

スケルトンのあいまい性

- ・極大点の定義が明確でない. 近傍が違えば異なる.
- ・図形を復元するための点の選び方は一意ではない.

画素数最小という基準を定めてもなお一意性は保証されない

- ・妥当な時間でスケルトンを求める範囲のアルゴリズムの複雑さ
- ・距離関数によって特性が大きく変わる

距離変換のアルゴリズム

4近傍型距離変換の主要部分の擬似プログラム

アルゴリズム2

入力2値画像, 距離変換結果: $f(h,w)$

注)画像の端一ラインは背景として扱う

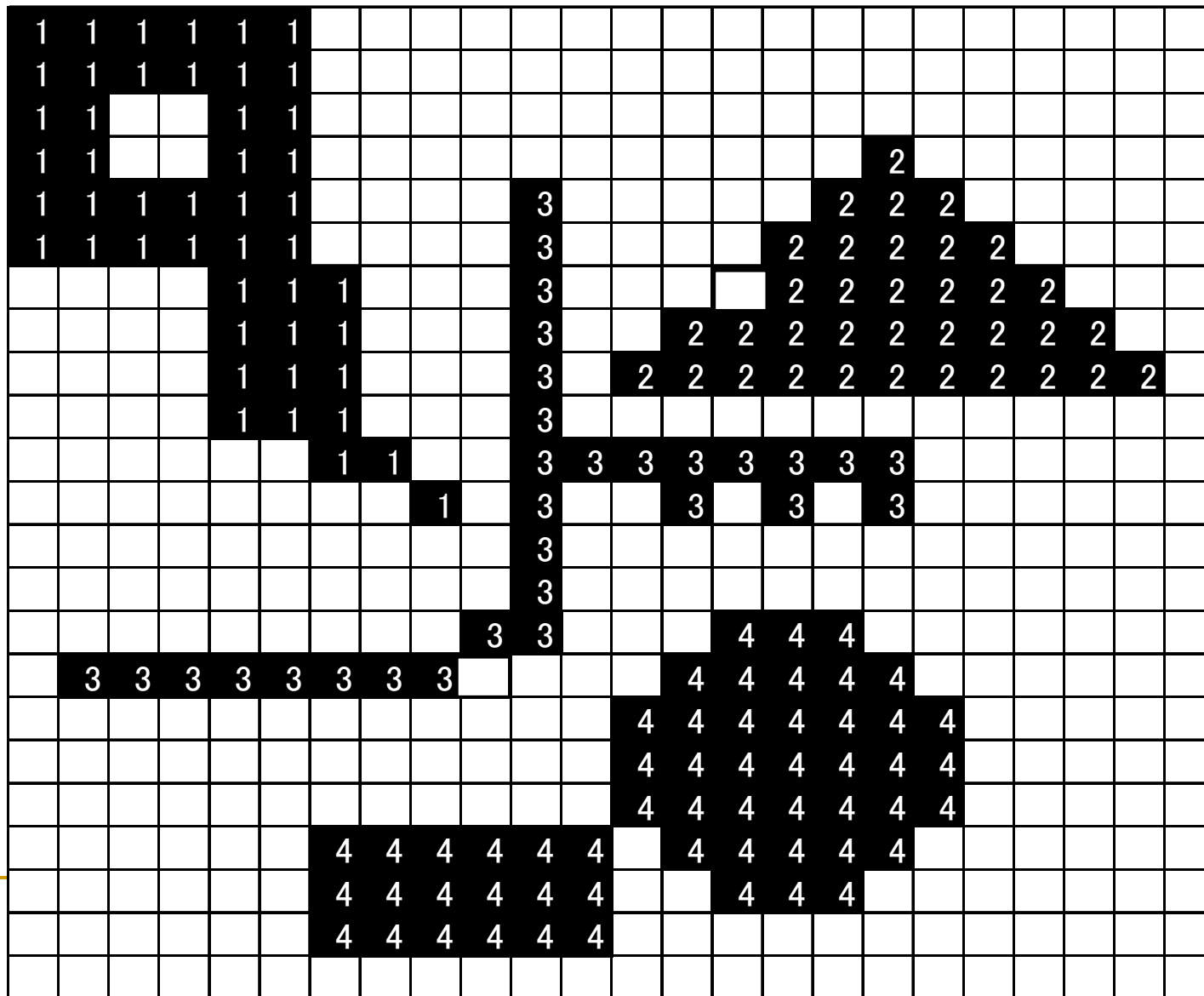
```
for(j=1; j<h-1; j++){  
    for(i=1; i<w-1; i++){ //画像を左上からスキャン  
        if( f(i,j) != 0 ){  
            nmin=min(f(i,j-1), f(i-1,j))+1;  
            f(i,j)=nmin;  
        }  
    }  
}
```

続き

```
for(j=h-2; j>=1; j--){  
    for(i=w-2; i>=0; i--){  
        //画像を右下からスキャン  
        if( f(i,j) != 0){  
            nmin=min(f(i,j), f(i,j+1)+1, f(i+1,j)+1);  
            f(i,j)=nmin;  
        }  
    }  
}
```

配列の走査を3次元的にし、最小値を求める部分を3次元固定近傍に変更すればそのまま利用できる

宿題2: 次の画像に対して, しきい値1.5で2値化し, アルゴリズム2の距離変換(4近傍)を実行し, 最終結果の距離画像Fを別紙に表せ.



その他の重要な項目

- 境界・境界追跡
- 領域塗りつぶし
- 点の幾何学的特徴
- オイラー数
- 曲率係数
- 連結数
- 画素の分類(頂点部, 穴部, 平地部, 鞍部, 尾根部, 谷部, 斜面部. . .) 濃淡画像
- Octree/Quadtree

画像演算