

マルチメディア信号解析



出席確認のため、
チャットに出席した旨書いて下さい！！



簡単な画像処理

- ・量子化レベル変更
- ・カラーからモノクロへの変換

- ・コントラスト調整
- ・閾値処理

- ・反転
- ・画像サイズ変更
- ・回転

- ・ノイズ除去
- ・エッジ検出

- ・フーリエ変換
etc.

点処理

局所処理

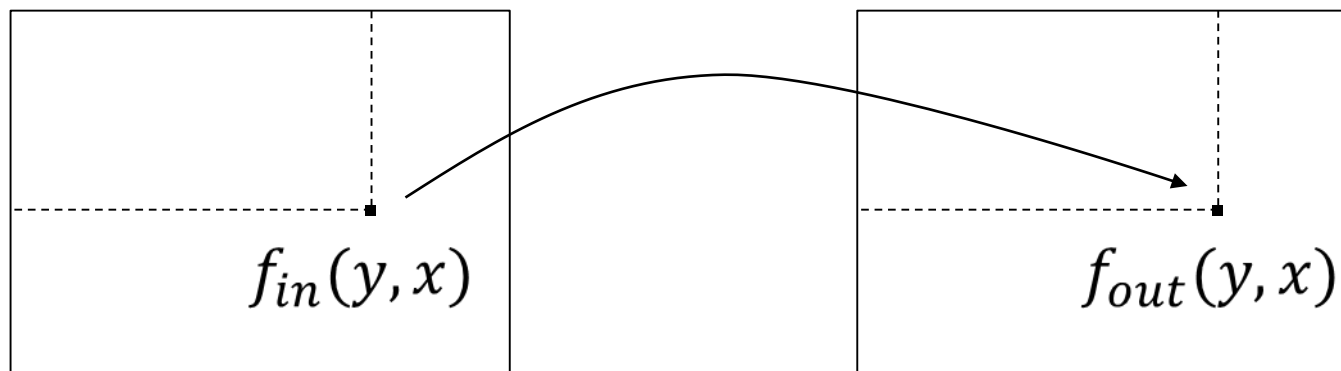
大局処理



簡単な画像処理

- ・量子化レベル変更
- ・カラーからモノクロへの変換

$$f_{out}(y, x) = 0.299f_{in}^R(y, x) + 0.587f_{in}^G(y, x) + 0.114f_{in}^B(y, x)$$



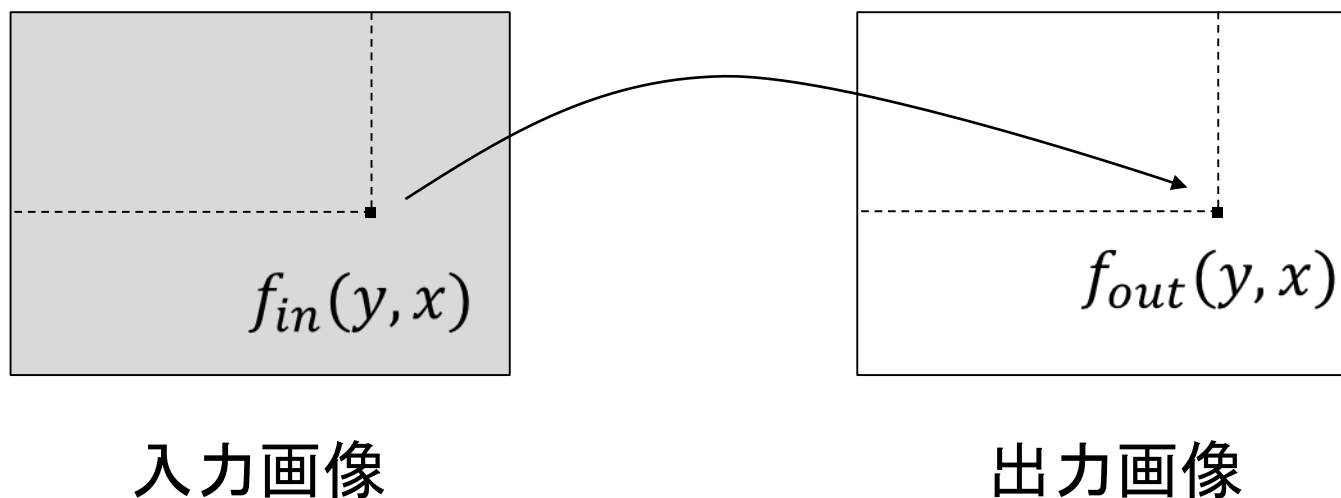
入力画像

出力画像

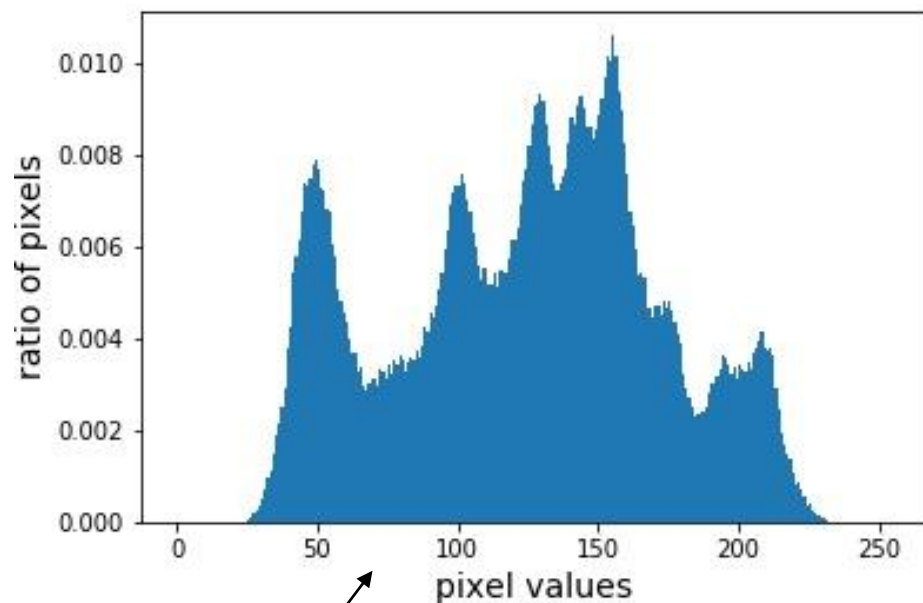


簡単な画像処理

- ・コントラスト調整
- ・閾値処理
 - ヒストグラムの利用？



ヒストグラム



8ビット画像の場合、0～255

各画素の濃度値の情報

ある濃度値をもった画素が
いくつあるか(割合)

空間的情報(位置)は失われる



ヒストグラム

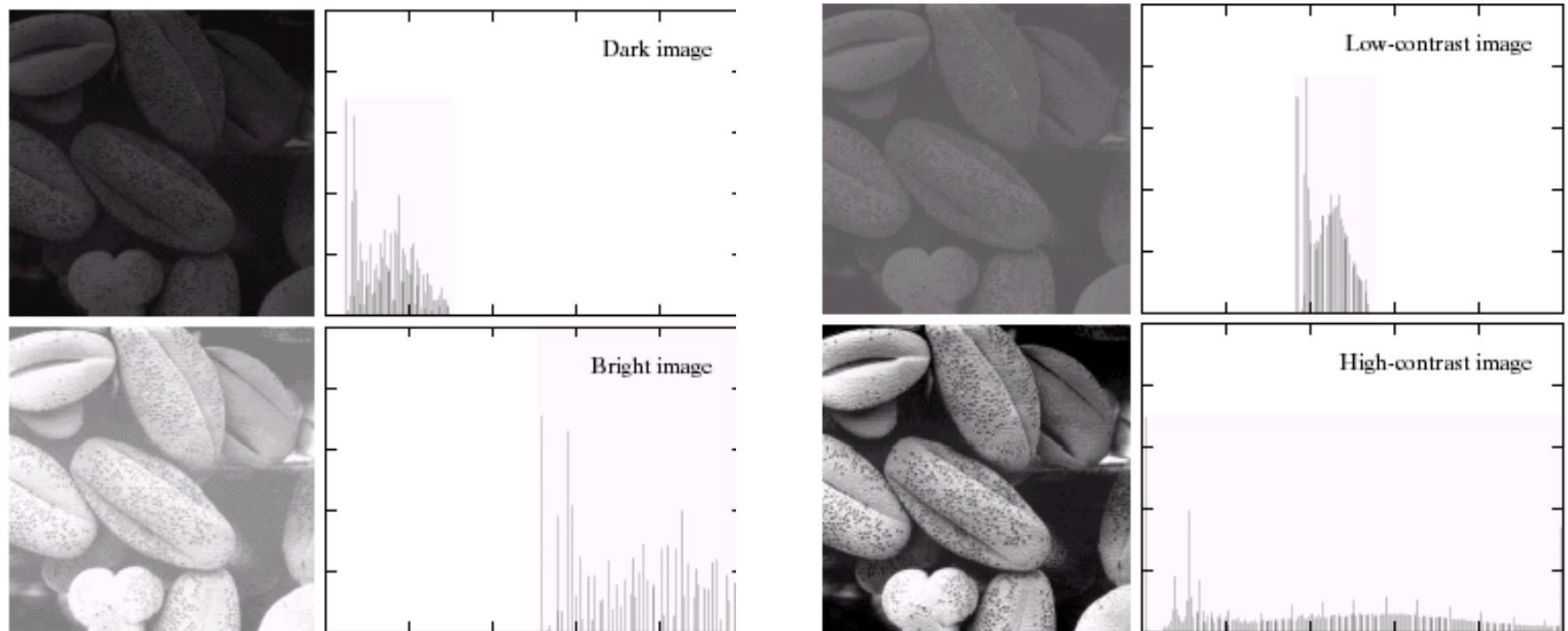
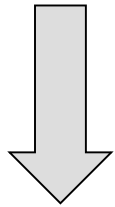
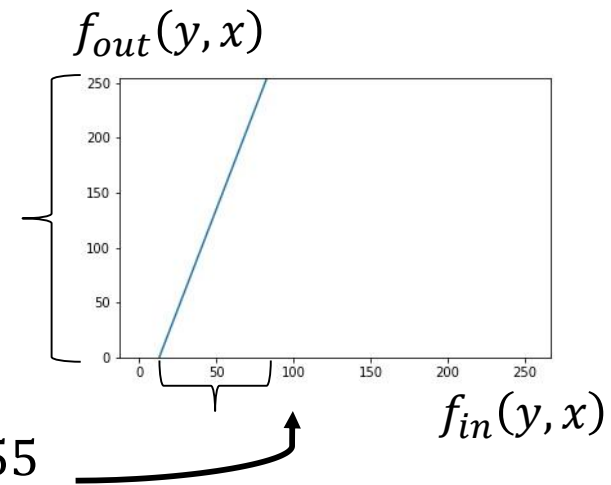
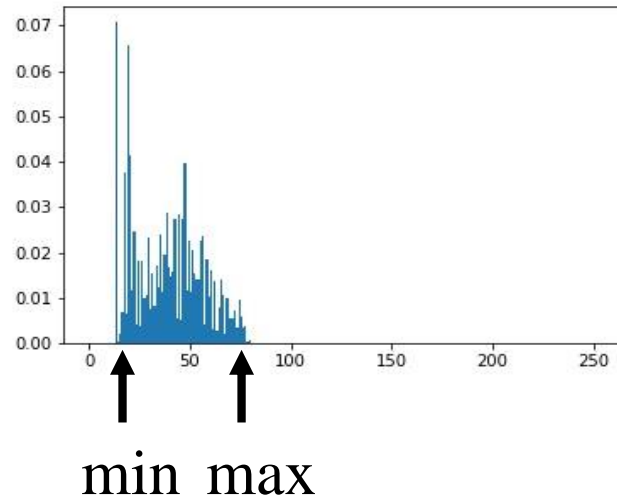
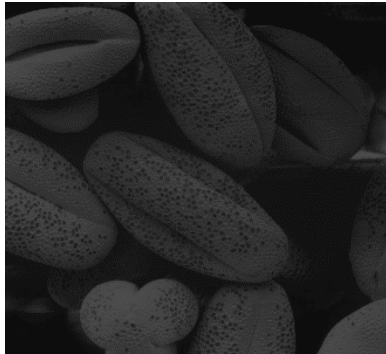


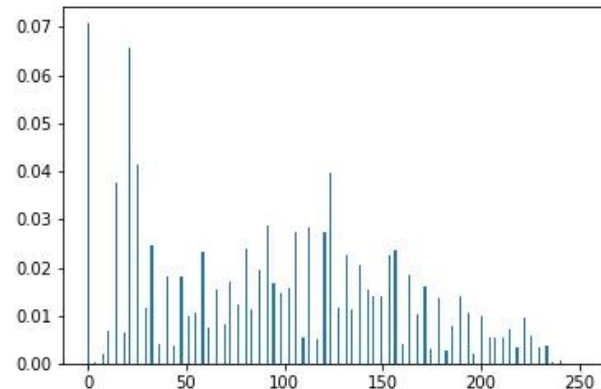
FIGURE 3.15 Four basic image types: dark, light, low contrast, high contrast, and their corresponding histograms. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)



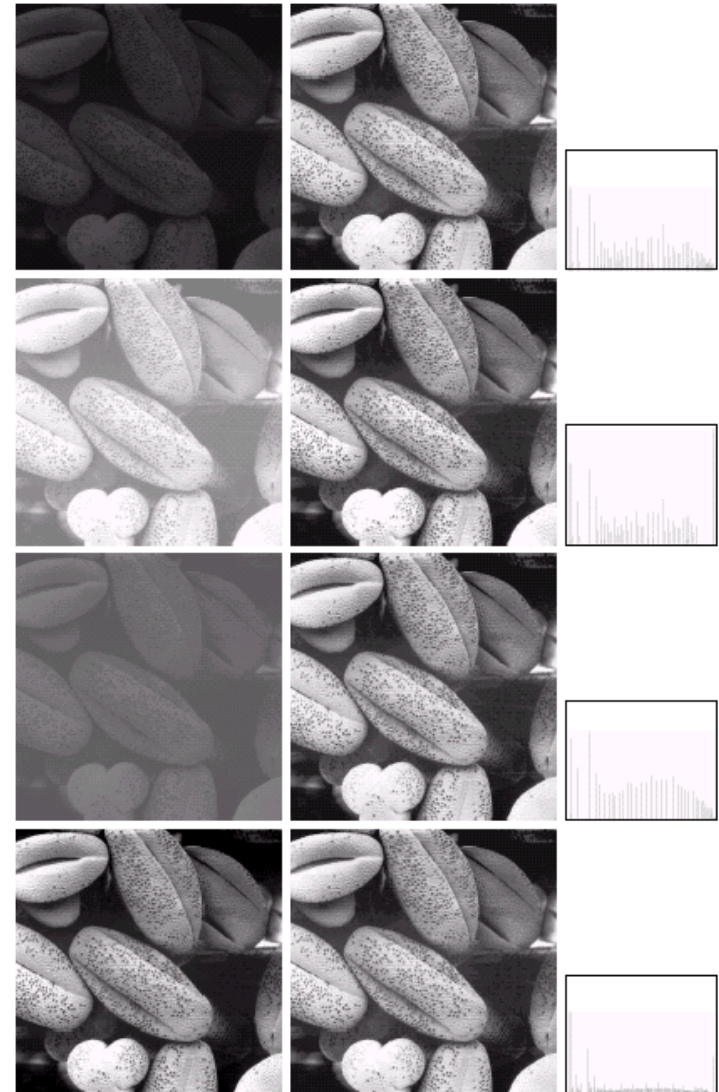
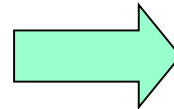
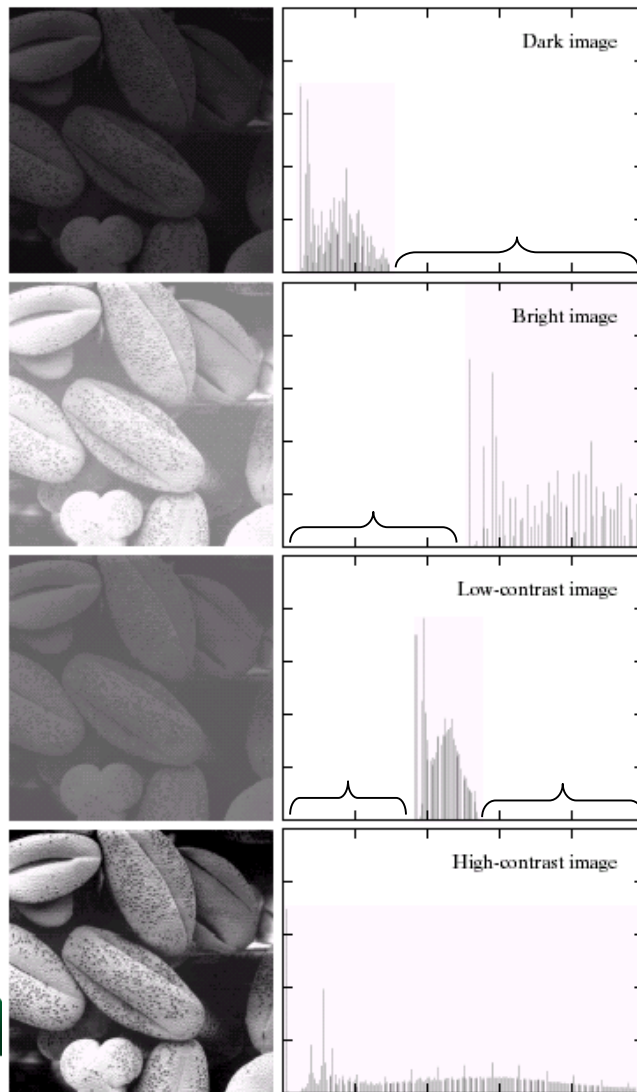
ヒストグラムの利用



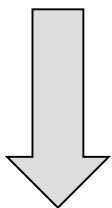
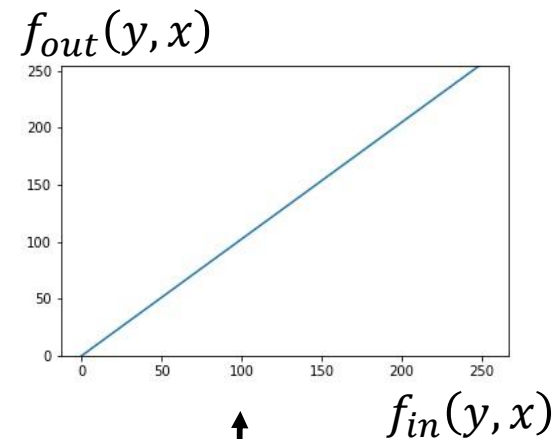
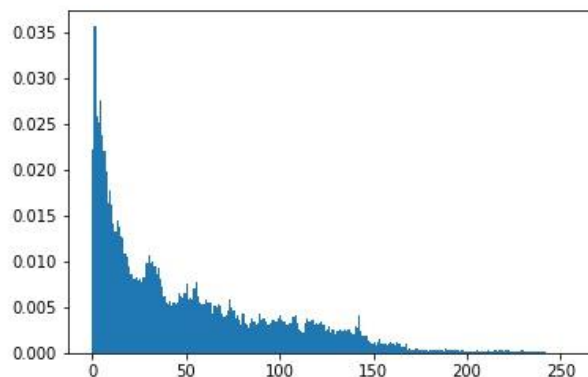
$$f_{out}(y, x) = \frac{f_{in}(y, x) - \min}{\max - \min} \times 255$$



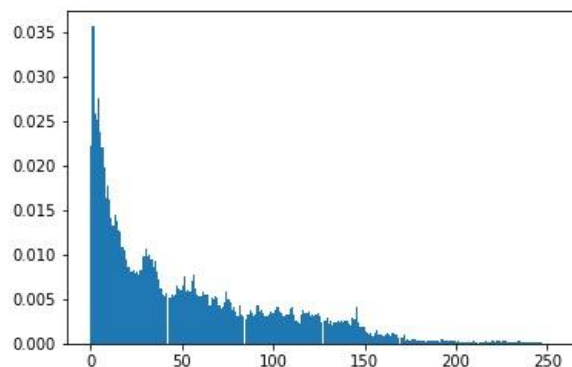
ヒストグラムの利用



ヒストグラムの利用



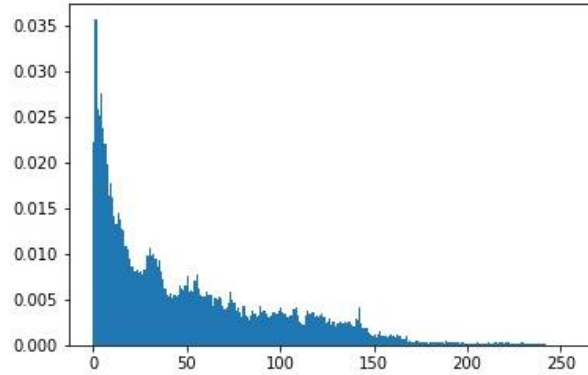
$$f_{out}(y, x) = \frac{f_{in}(y, x) - \min}{\max - \min} \times 255$$



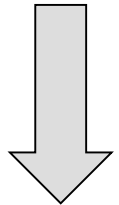
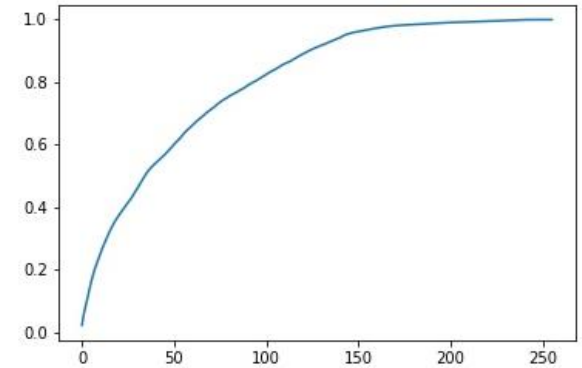
ヒストグラムの利用



ヒストグラム

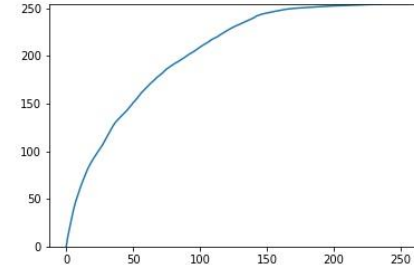


累積分布(cdf)

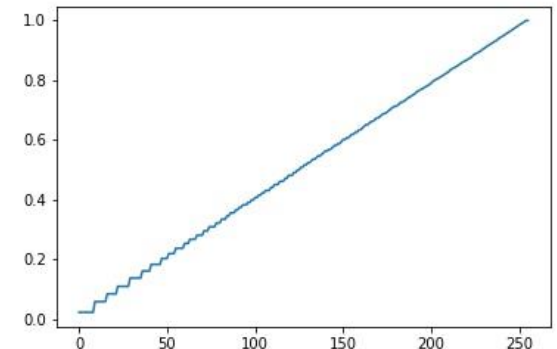
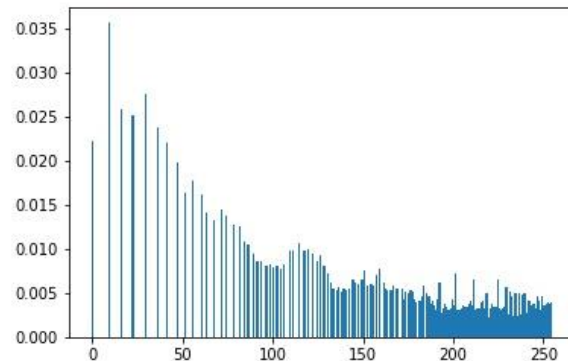


$$f_{out}(y, x) = cdf(f_{in}(y, x)) \times 255$$

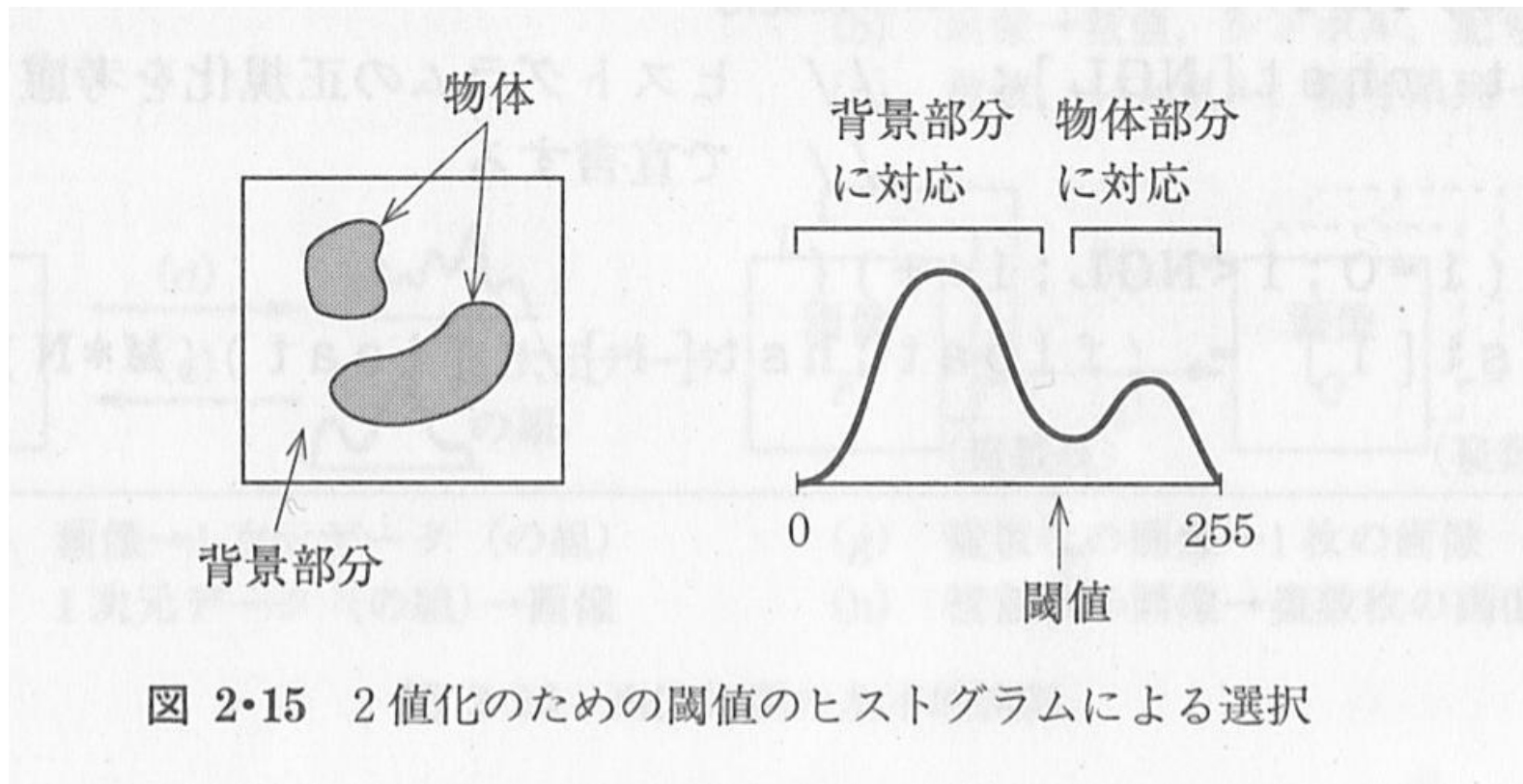
$f_{out}(y, x)$



$f_{in}(y, x)$



ヒストグラムの利用



ヒストグラムの利用

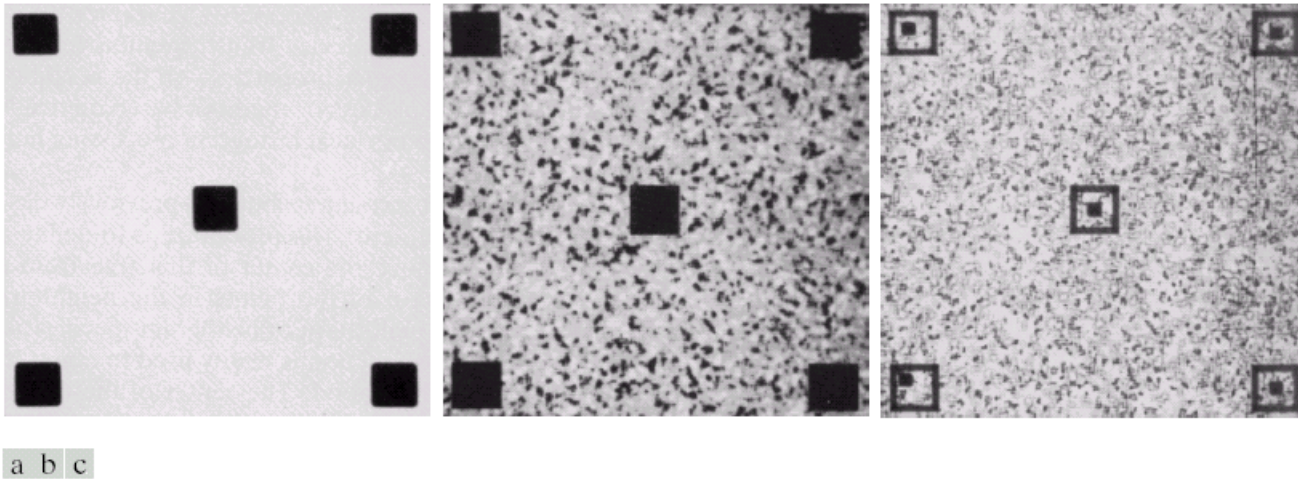
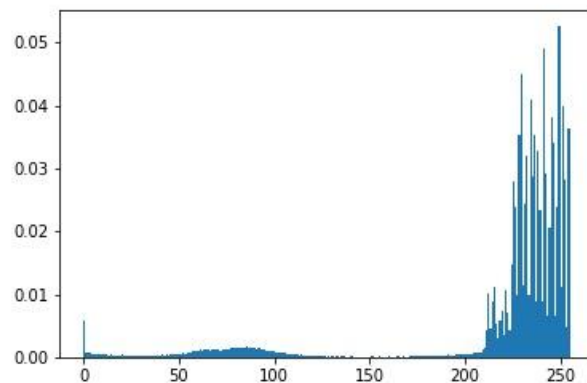
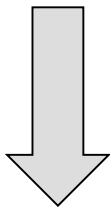


FIGURE 3.23 (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization using a 7×7 neighborhood about each pixel.

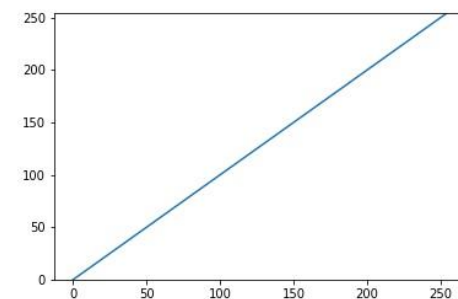


ヒストグラムの利用

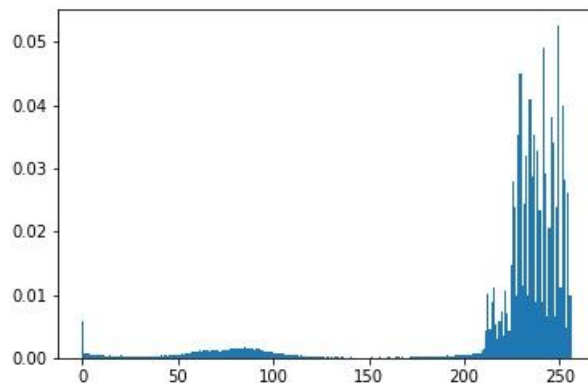


$$f_{out}(y, x) = \frac{f_{in}(y, x) - \min}{\max - \min} \times 255$$

$f_{out}(y, x)$



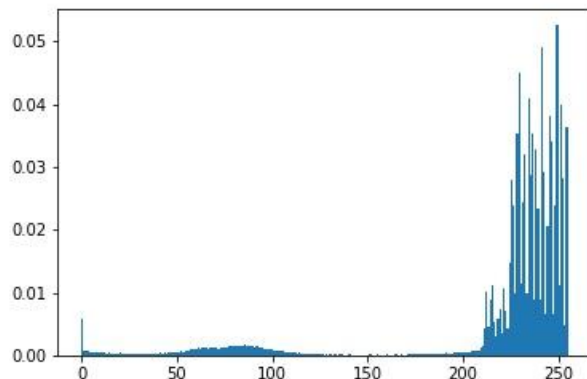
$f_{in}(y, x)$



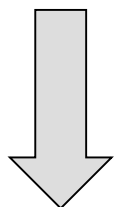
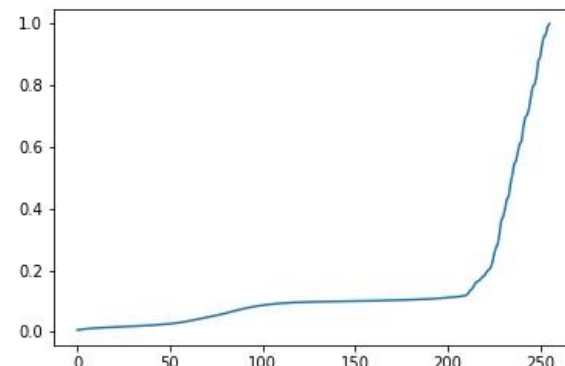
ヒストグラムの利用



ヒストグラム

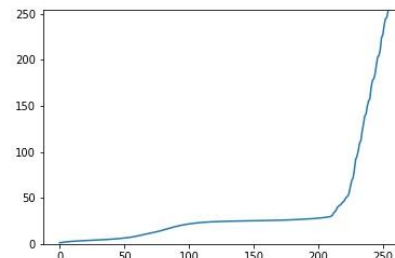


累積分布(cdf)

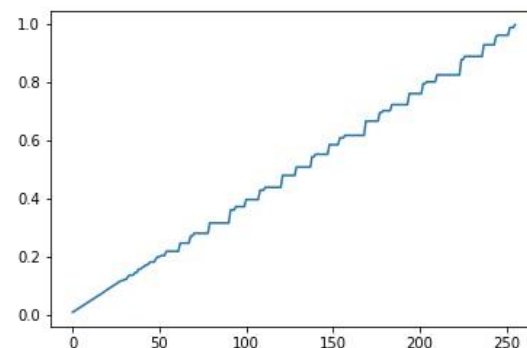
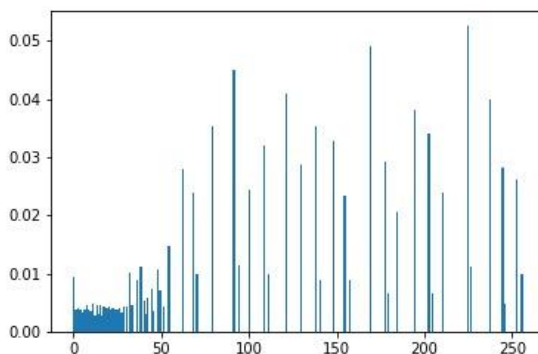
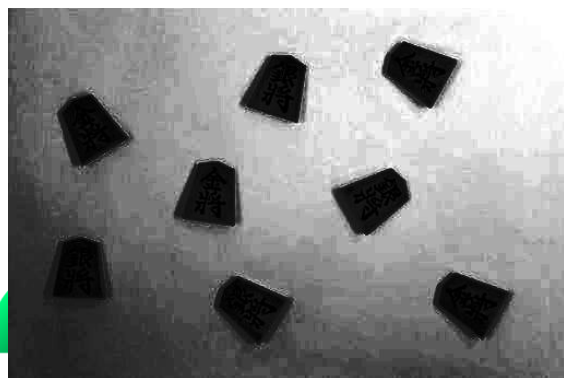


$$f_{out}(y, x) = cdf(f_{in}(y, x)) \times 255$$

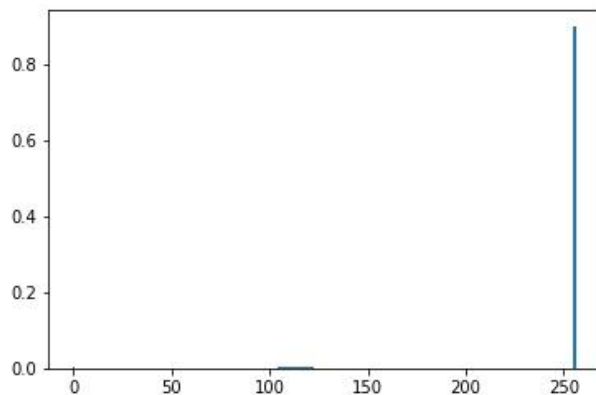
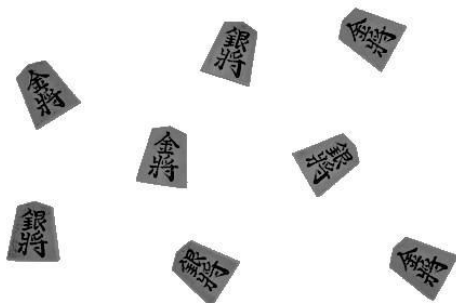
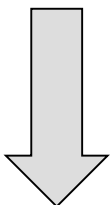
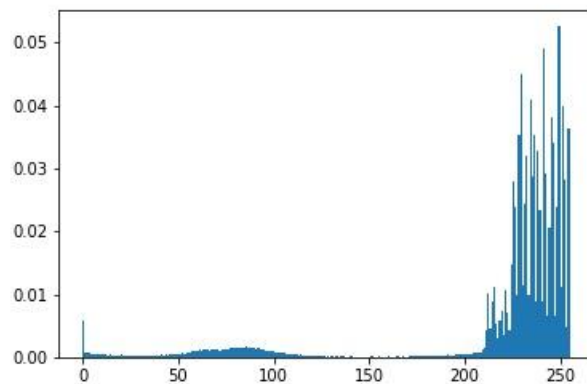
$f_{out}(y, x)$



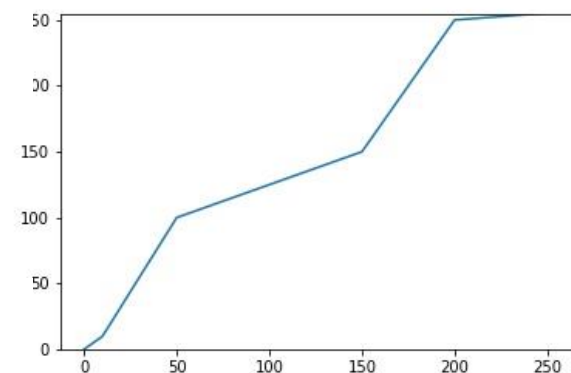
$f_{in}(y, x)$



ヒストグラムの利用



$f_{out}(y, x)$

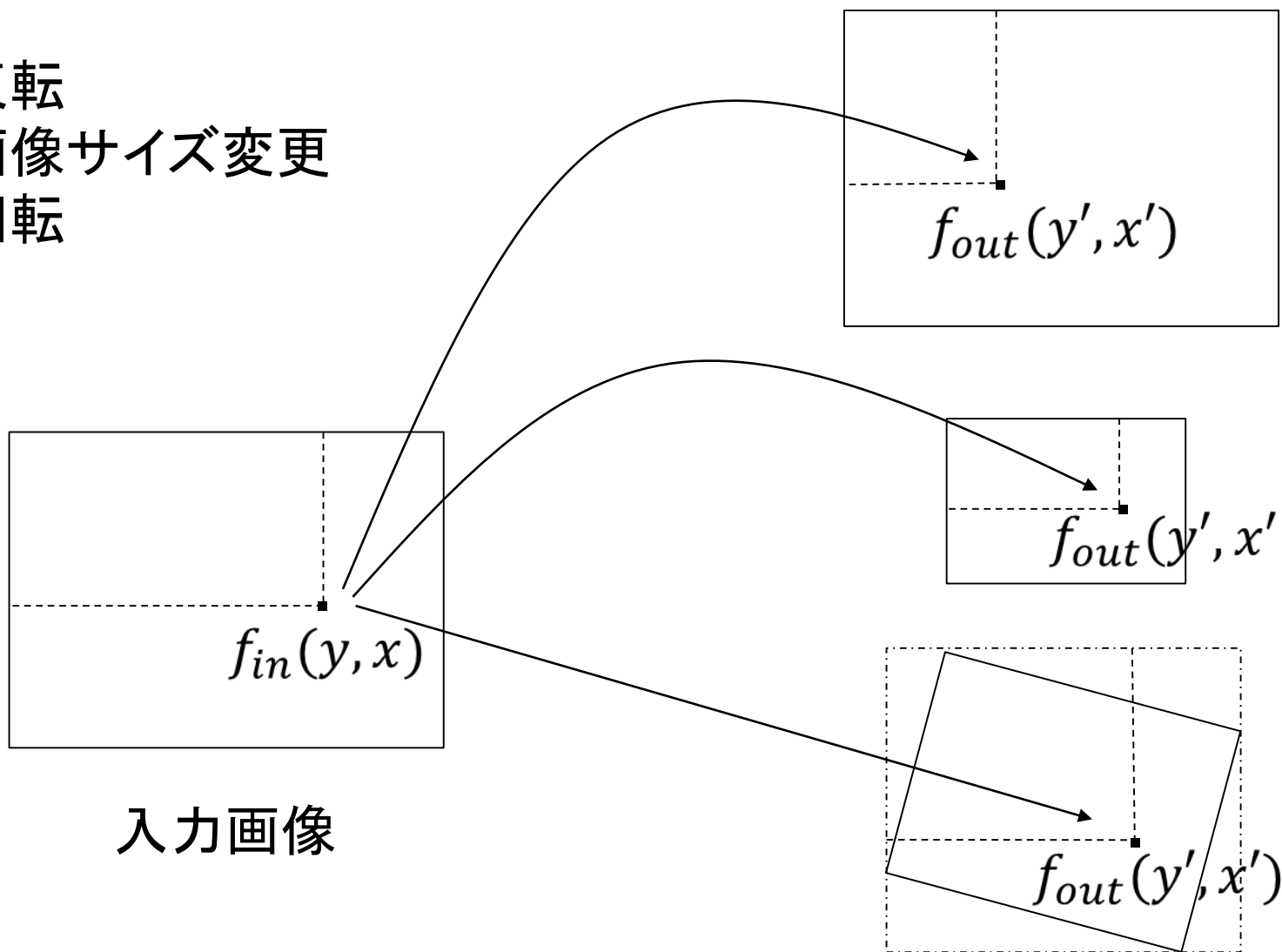


$f_{in}(y, x)$

簡単な画像処理

出力画像

- ・反転
- ・画像サイズ変更
- ・回転



入力画像



幾何變換

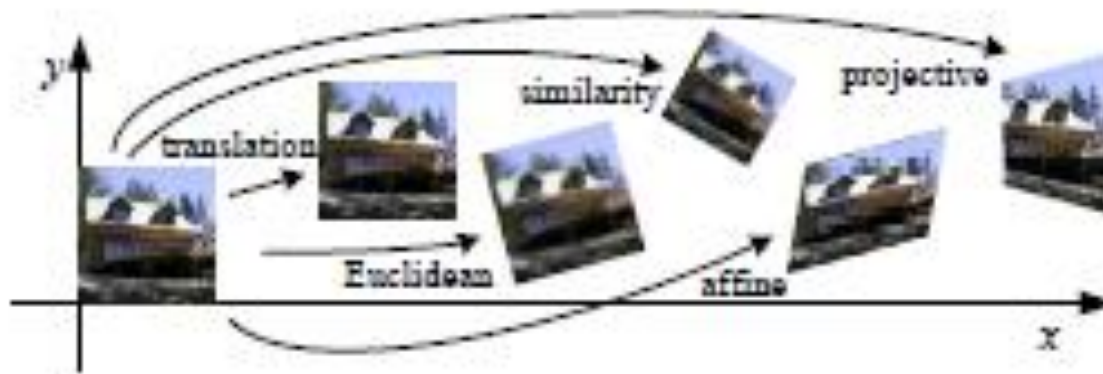


Figure 3.45 Basic set of 2D geometric image transformations.

I : Identity matrix

$$R = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

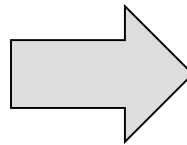
$$x' = \frac{h_{00}x + h_{01}y + h_{02}}{h_{20}x + h_{21}y + h_{22}}$$

$$y' = \frac{h_{10}x + h_{11}y + h_{12}}{h_{20}x + h_{21}y + h_{22}}$$

Transformation	Matrix	# DoF	Preserves	Icon
translation	$\begin{bmatrix} I & t \end{bmatrix}_{2 \times 3}$	2	orientation	
rigid (Euclidean)	$\begin{bmatrix} R & t \end{bmatrix}_{2 \times 3}$	3	lengths	
similarity	$\begin{bmatrix} sR & t \end{bmatrix}_{2 \times 3}$	4	angles	
affine	$\begin{bmatrix} A \end{bmatrix}_{2 \times 3}$	6	parallelism	
projective	$\begin{bmatrix} \tilde{H} \end{bmatrix}_{3 \times 3}$	8	straight lines	



幾何變換

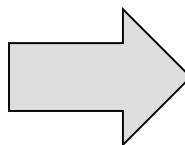


$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a * \cos \theta & -a * \sin \theta & tx \\ a * \sin \theta & a * \cos \theta & ty \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$(a = 1.0, \theta = 30^\circ, tx = 10, ty = -60)$$



幾何変換



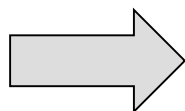
$$\begin{pmatrix} a * \cos \theta & -a * \sin \theta & tx \\ a * \sin \theta & a * \cos \theta & ty \\ 0 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$(a = 1.0, \theta = 30^\circ, tx = 10, ty = -60)$$



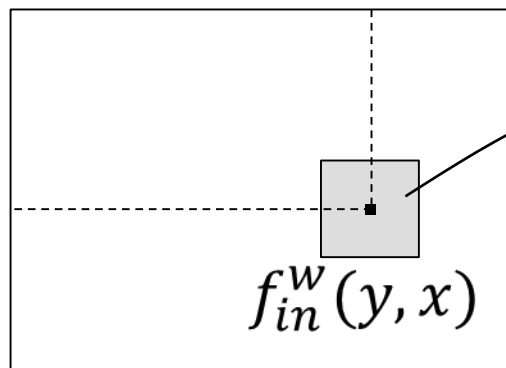
簡単な画像処理

- ・ノイズ除去
- ・エッジ検出

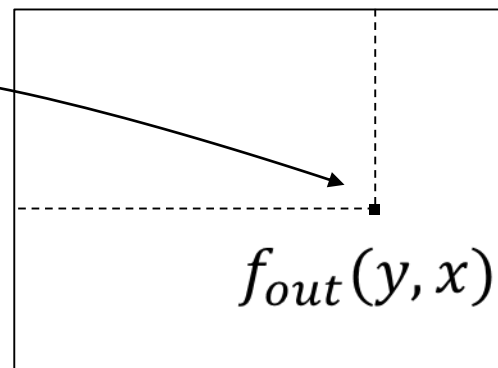


局所処理

(空間フィルタリング)



入力画像



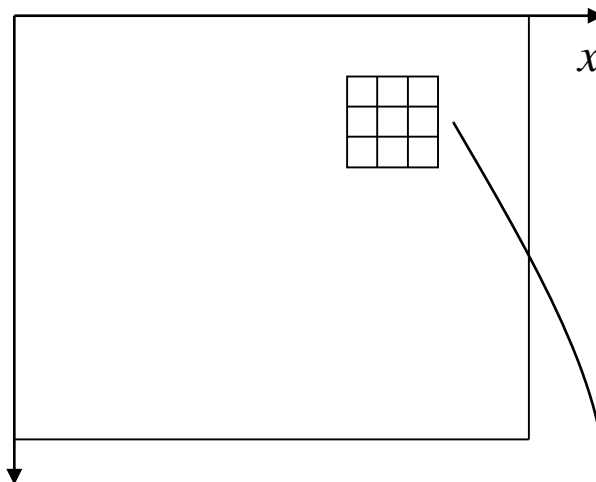
出力画像



局所処理(近傍処理)

$n \times n$ 画素近傍を用いての処理

カーネル



$f_{in}^W(y, x)$

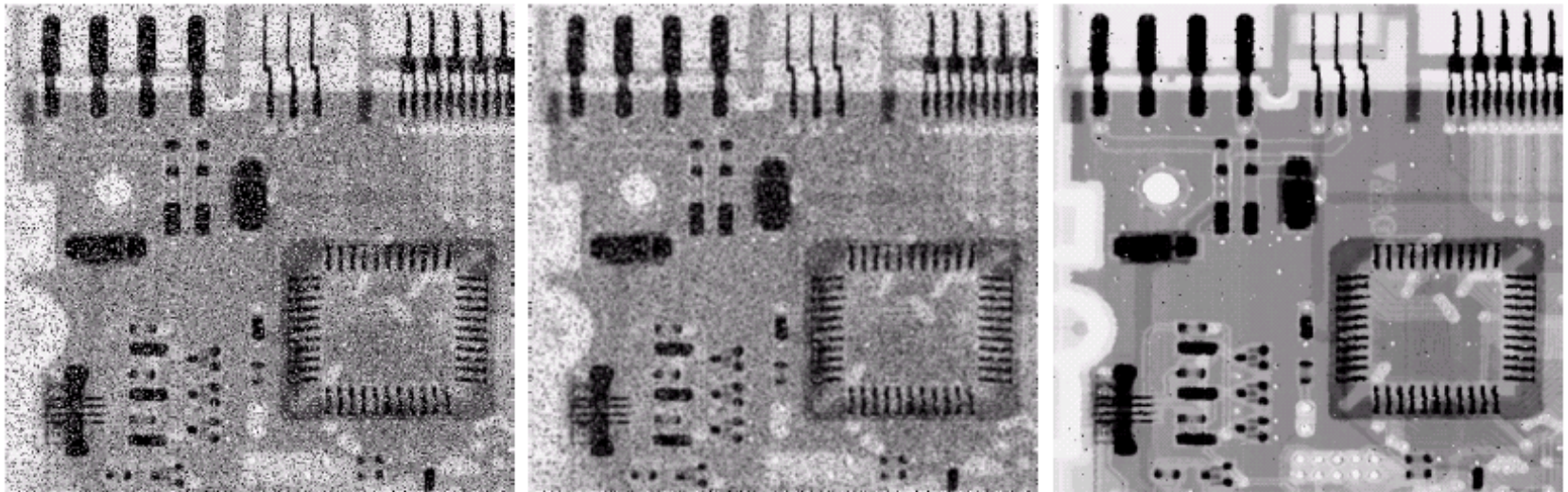
$f(y-1, x-1)$	$f(y-1, x)$	$f(y-1, x+1)$
$f(y, x-1)$	$f(y, x)$	$f(y, x+1)$
$f(y+1, x-1)$	$f(y+1, x)$	$f(y+1, x+1)$

$w(-1, -1)$	$w(-1, 0)$	$w(-1, 1)$
$w(0, -1)$	$w(0, 0)$	$w(0, 1)$
$w(1, -1)$	$w(1, 0)$	$w(1, 1)$

$$f_{out}(x, y) = \frac{\sum_{t=-1}^1 \sum_{s=-1}^1 w(t, s) f_{in}(y+t, x+s)}{\sum_{t=-1}^1 \sum_{s=-1}^1 w(t, s)}$$



局所处理(近傍处理):例



a b c

FIGURE 3.37 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3×3 averaging mask. (c) Noise reduction with a 3×3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)



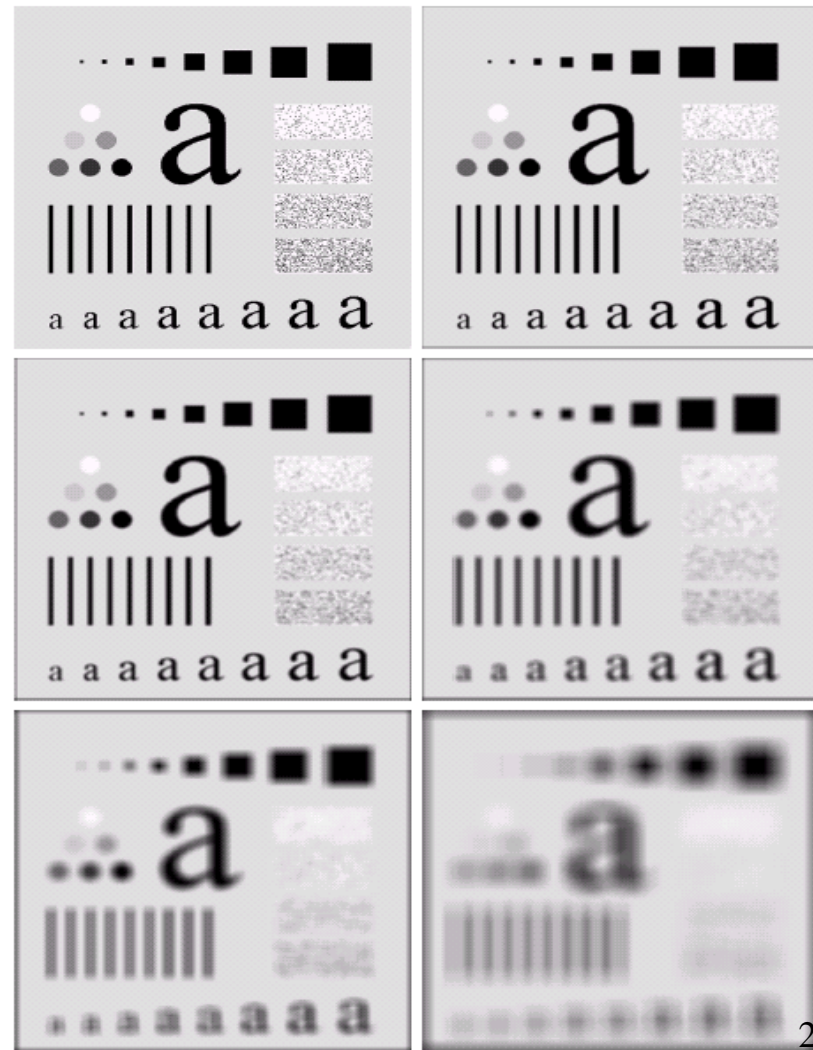
局所処理(近傍処理):例

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$



a b
c d
e f

FIGURE 3.35 (a) Original image, of size 500×500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes $n = 3, 5, 9, 15$, and 35 , respectively. The black squares at the top are of sizes $3, 5, 9, 15, 25, 35, 45$, and 55 pixels, respectively; their borders are 25 pixels apart. The letters at the bottom range in size from 10 to 24 points, in increments of 2 points; the large letter at the top is 60 points. The vertical bars are 5 pixels wide and 100 pixels high; their separation is 20 pixels. The diameter of the circles is 25 pixels, and their borders are 15 pixels apart; their gray levels range from 0% to 100% black in increments of 20% . The background of the image is 10% black. The noisy rectangles are of size 50×120 pixels.



局所処理(近傍処理):例

0	1	0	1	1	1
1	-4	1	1	-8	1
0	1	0	1	1	1

0	-1	0	-1	-1	-1
-1	4	-1	-1	8	-1
0	-1	0	-1	-1	-1

a b
c d

FIGURE 3.40

(a) Image of the North Pole of the moon.
(b) Laplacian-filtered image.
(c) Laplacian image scaled for display purposes.
(d) Image enhanced by using Eq. (3.7-5).
(Original image courtesy of NASA.)



エッジ検出

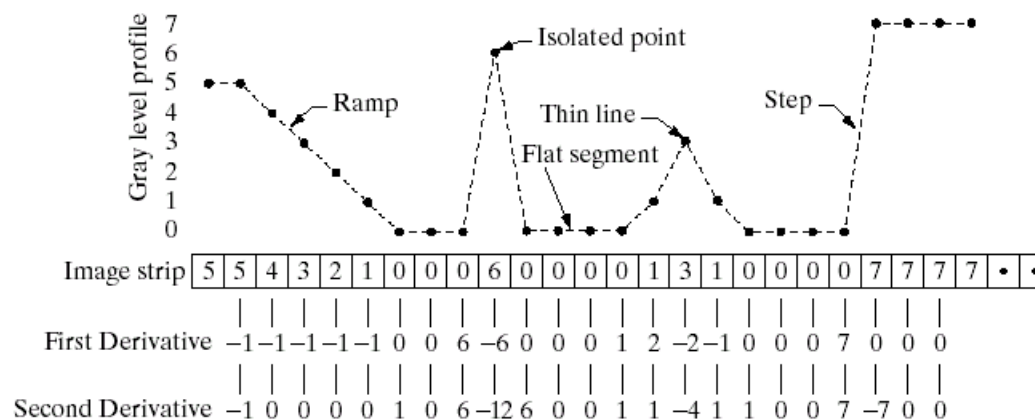
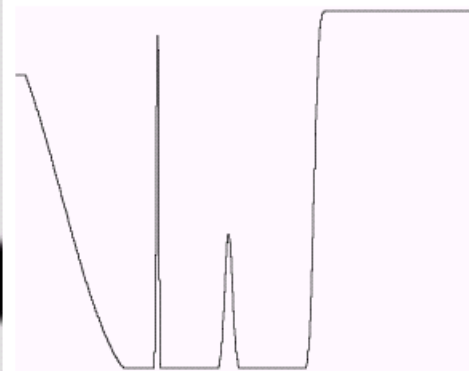
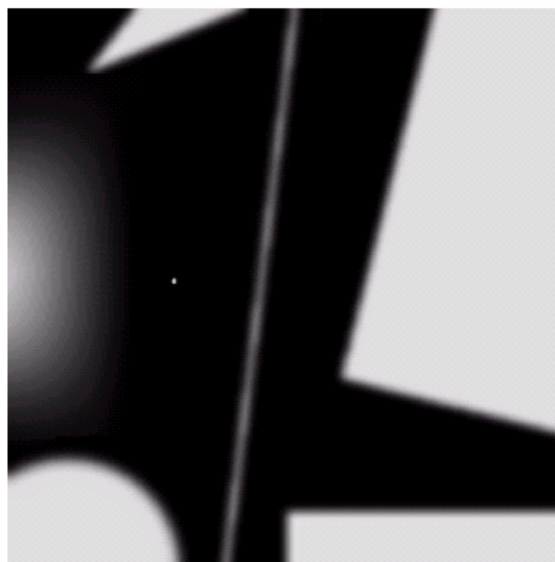
$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

a b
c

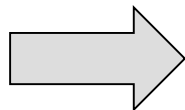
FIGURE 3.38

(a) A simple image. (b) 1-D horizontal gray-level profile along the center of the image and including the isolated noise point. (c) Simplified profile (the points are joined by dashed lines to simplify interpretation).

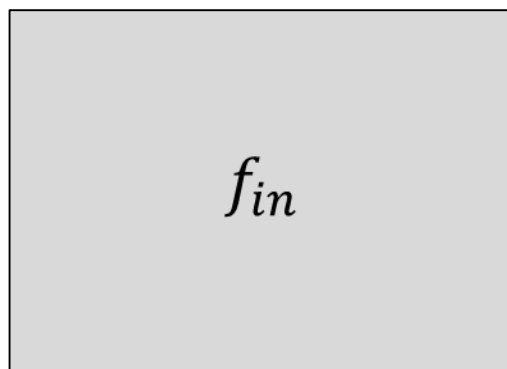
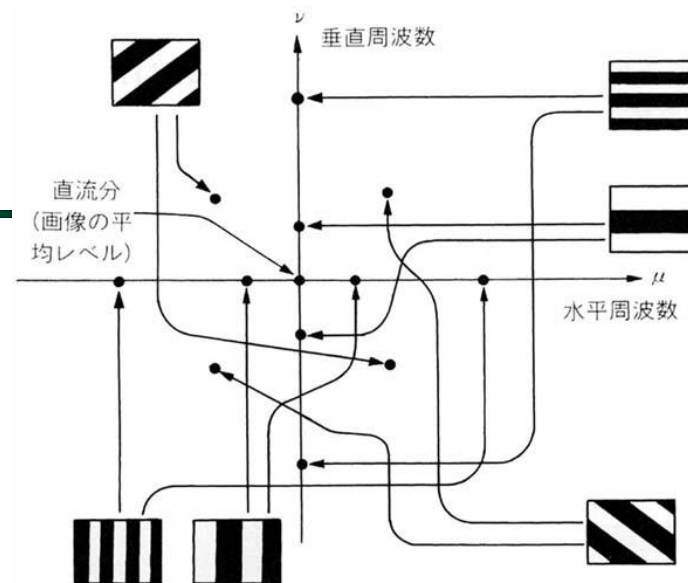


簡単な画像処理

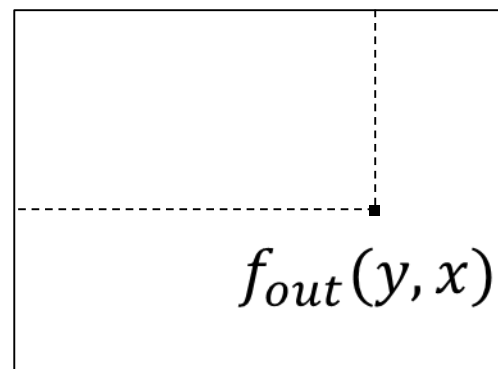
・フーリエ変換



大局処理



入力画像



出力画像



演習課題

※C言語もしくはpythonで実装すること

- コントラストの悪い適当な画像（スマートフォンの撮影画像など）を読み込み、ヒストグラムに基づきコントラストを上げた画像を作成し、保存するプログラムを作成せよ。複数の画像、複数の手法で試すこと。
- 適当な画像を読み込み、幾何変換し、保存するプログラムを作成せよ。背景は白や黒の画素値で補完してよい。
- 適当な画像を読み込み、局所処理をして保存するプログラムを作成せよ。処理はなんでもよい。
- 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：5月5日（火）

出席確認のため、
チャットに出席した旨書いて下さい！！

