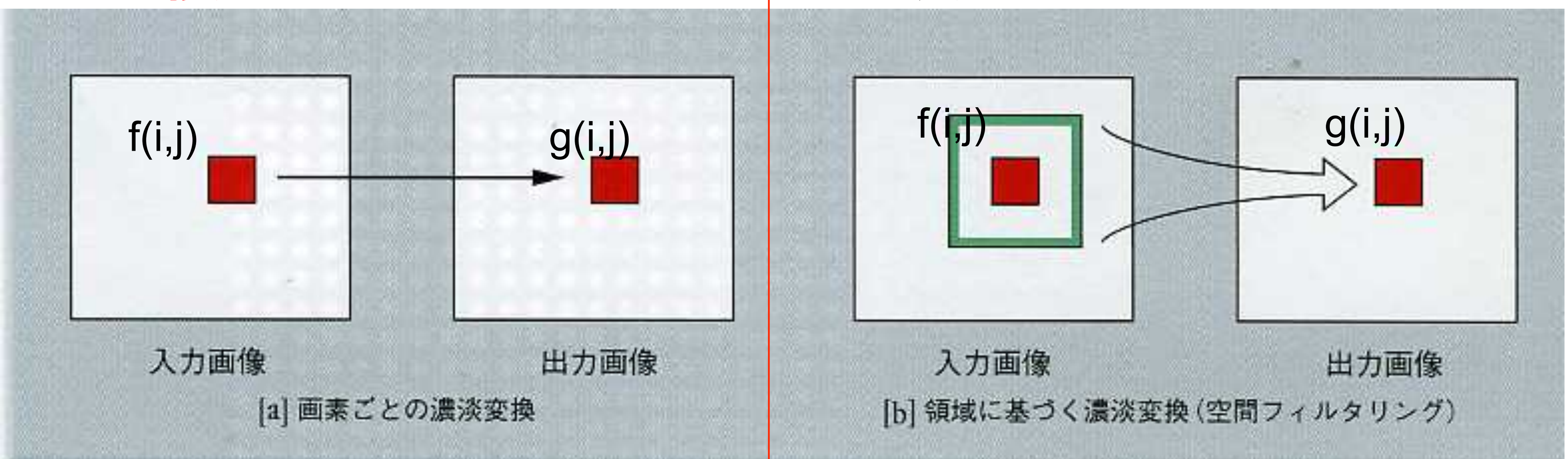
第3回　空間フィルタリング（filtering）（領域に基づく濃度変換）

１．フィルタリング（領域に基づく濃度変換）
３．平均値フィルタ
４．エッジ検出フィルタ
５．鮮鋭化フィルタ

空間フィルタリング(filtering)
(領域に基づく濃度変換)

- 入力画像の対応する画素値だけではなく、その周囲の画素も含めた領域内の画素値を用いた計算を**領域に基づく濃度変換**または**空間フィルタリング**という。



$$g(i,j) = T[f(i,j)]$$

$T[\ ]$: 諧調変換関数

（トーンカーブ）

$$g(i,j) = f * h = \sum_{n=-w}^{w}\sum_{m=-w}^{w} f(i+m, j+n)\cdot h(m,n)$$

$h(m,n)$: フィルタ関数（$kenel$ともいう）

フィルタのサイズ（窓幅）: $(2w+1)(2w+1)$

線形フィルタ(linear filter)

- **線形フィルタ：ノイズ除去フィルタ（平均値フィルタ、gaussianフィルタ）；エッジ検出フィルタ（sobel, Laplacianフィルタなど）**
- **非線形フィルタ：ノイズ除去フィルタ（Medianフィルタなど）**
- **線形フィルタ：出力画像gは入力画像fとフィルタ関数hとの畳込み積分（積和演算）で表される。**

$$g(i,j) = f * h = \sum_{n=-w}^{w} \sum_{m=-w}^{w} f(i+m, j+n) \cdot h(m,n)$$

$h(m,n)$：フィルタ関数（kernelともいう）

フィルタのサイズ（窓幅）：$(2w+1)(2w+1)$

線形フィルタの種類

- 平滑化（スムージング）フィルタ(Low-pass filter)
  - 平均値フィルタ（mean filter)
  - Gaussianフィルタ

  例:　0 0 0 3 0 0 0 → 0 0 1 1 1 0 0 (平均値）
- エッジ検出フィルタ (High-pass filter)
  - Sobel フィルタ
  - Prewittフィルタ
  - Laplaceフィルタ

  例:　0 0 0 1 1 1 1 → 0 0 0 1 0 0 0 (一次微分）

  0 0 1 -1 0 0 0 (二次微分）

具体的な計算(Kernel：3x3)

例えば：カーネル関数が3x3の場合、*w*=1 (-*w*~*w*)

f(i-1,j-1)	f(i-1,j)	f(i-1,j+1)
f(i,j-1)	f(i,j)	f(i,j+1)
f(i+1,j+1)	f(i+1,j)	f(i+1,j+1)

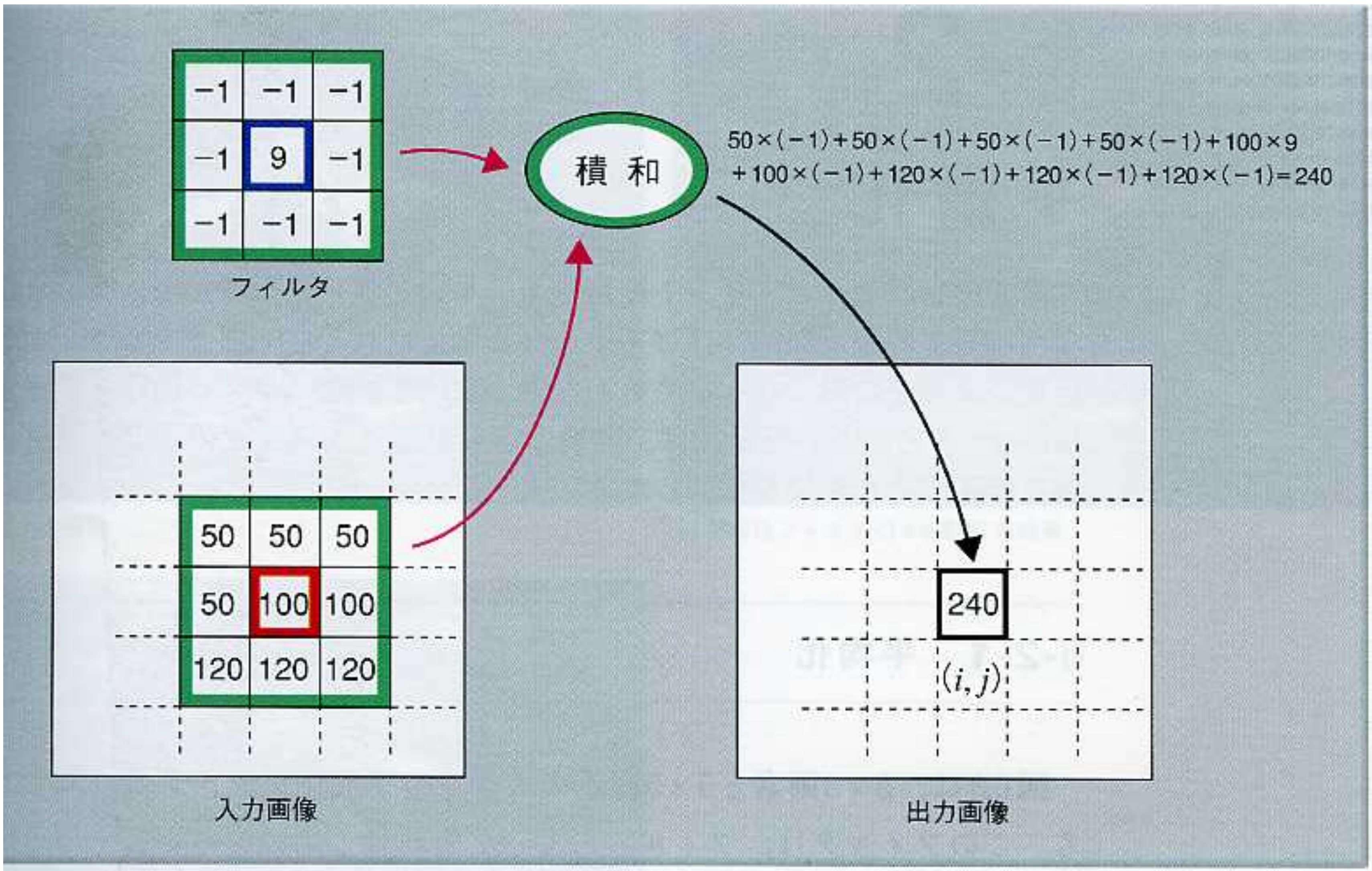
*

h(-1,-1)	h(-1,0)	h(-1,1)
h(0,-1)	h(0,0)	h(0,1)
h(1,-1)	h(1,0)	h(1,1)

$$g(i,j) = f * h = \sum_{n=-w}^{w} \sum_{m=-w}^{w} f(i+m, j+n) h(m,n)$$

$$\begin{aligned} &= h(-1,-1) f(i-1, j-1) + h(-1,0) f(i-1, j) + h(-1,1) f(i-1, j+1) \\ &+ h(0,-1) f(i, j-1) \quad + h(0,0) f(i, j) \quad + h(0,1) f(i, j+1) \\ &+ h(1,-1) f(i+1, j-1) \quad + h(1,0) f(i+1, j) \quad + h(1,1) f(i+1, j+1) \end{aligned}$$

具体的な例



−1 −1 −1
−1 9 −1
−1 −1 −1
フィルタ
積 和
50×(−1)+50×(−1)+50×(−1)+50×(−1)+100×9
+100×(−1)+120×(−1)+120×(−1)+120×(−1)=240
50 50 50
50 100 100
120 120 120
入力画像
240
(i, j)
出力画像

畳み込みによるfilteringのアルゴリズム

画像f：NxN;　　フィルタカーネルh:(2w+1)x(2w+1）

Input f(i,j) (i,j=0,N-1) and h(m,n) (m,n=-w, w)

```
For i=w, N-1-w;
  For j=w, N-1-w;
        g(i,j)=0:
    For n=-w, w;
      For m=-w,w;
         g(i,j) += h(m,n) x f(i+m,j+n);
   next m, n;
next i,j;
```

平均値フィルタ(mean filter)

f(j-1,i-1)	f(j-1,i)	f(j-1,i+1)
f(j,i-1)	f(j,i)	f(j,i+1)
f(j+1,i+1)	f(j+1,i)	f(j+1,i+1)

$$
* \quad \frac{1}{5}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}
$$

$$
g(j,i) = \frac{f(j-1,i) + f(j,i-1) + f(i,j) + f(j,i+1) + f(j+1,i)}{5}
$$

重み付き平均値フィルターの例(Kernel：3x3)

$$\frac{1}{6}\begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 1 & 2 & 1 \\ \hline 0 & 1 & 0 \\ \hline\end{array}$$

$$g(i,j) = f * h = \sum_{u=-m}^{m}\sum_{v=-m}^{m} h(u,v)f(i-u,j-v)$$

$$\begin{aligned}
&= h(-1,-1)f(i-1,j-1) + h(-1,0)f(i-1,j) + h(-1,1)f(i-1,j+1) \\
&+ h(0,-1)f(i,j-1) \quad + h(0,0)f(i,j) \quad + h(0,1)f(i,j+1) \\
&+ h(1,-1)f(i+1,j-1) \quad + h(1,0)f(i+1,j) \quad + h(1,1)f(i+1,j+1)
\end{aligned}$$

$$\begin{aligned}
&= [0\cdot f(i-1,j-1) + 1\cdot f(i-1,j) + 0\cdot f(i-1,j+1) \\
&\ +1\cdot f(i,j-1) \quad +2\cdot f(i,j) \quad +1\cdot f(i,j+1) \\
&\ +0\cdot f(i+1,j-1) + 1\cdot f(i+1,j) + 0\cdot f(i+1,j+1)]/6
\end{aligned}$$

$$= \frac{f(i-1,j) + f(i,j-1) + 2f(i,j) + f(i,j+1) + f(i+1,j)}{6}$$

重み付き平均値フィルタ（weighted mean filter)

f(j-1,i-1)	f(j-1,i)	f(j-1,i+1)
f(j,i-1)	f(j,i)	f(j,i+1)
f(j+1,i+1)	f(j+1,i)	f(j+1,i+1)

*

$$\frac{1}{6}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$g(j,i) = \frac{f(j-1,i) + f(j,i-1) + 2f(i,j) + f(j,i+1) + f(j+1,i)}{6}$$

平均値フィルタ例

$$\frac{1}{9}\times\begin{array}{|c|c|c|}\hline 1&1&1\\\hline 1&1&1\\\hline 1&1&1\\\hline\end{array}$$

(a)　均一な重み

$$\frac{1}{10}\times\begin{array}{|c|c|c|}\hline 1&1&1\\\hline 1&2&1\\\hline 1&1&1\\\hline\end{array}$$

(b)　中央に大きな重み

図 4・13　3×3 の移動平均フィルタ

$$\frac{1}{25}\times\begin{array}{|c|c|c|c|c|}\hline 1&1&1&1&1\\\hline 1&1&1&1&1\\\hline 1&1&1&1&1\\\hline 1&1&1&1&1\\\hline 1&1&1&1&1\\\hline\end{array}$$

(a)　均一な重み

$$\frac{1}{35}\times\begin{array}{|c|c|c|c|c|}\hline 1&1&1&1&1\\\hline 1&2&2&2&1\\\hline 1&2&3&2&1\\\hline 1&2&2&2&1\\\hline 1&1&1&1&1\\\hline\end{array}$$

(b)　中央に大きな重み

図 4・14　5×5 の移動平均フィルタ

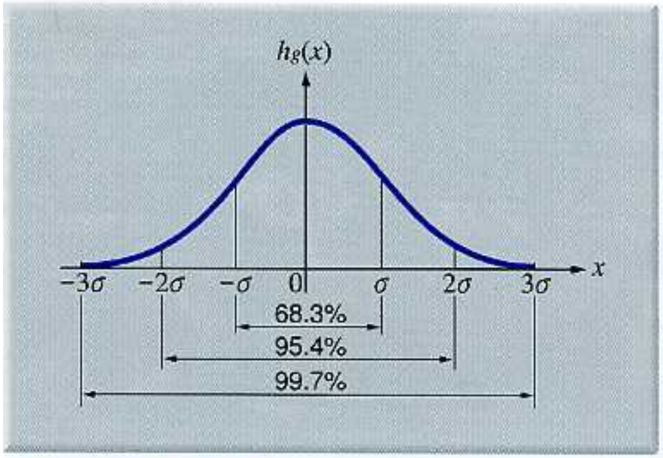
平均値フィルタ出力例

[a] 入力画像

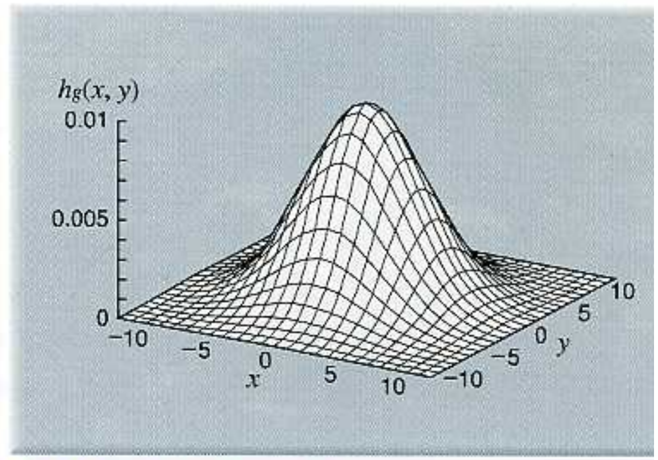
[b] 平均化フィルタ（3×3画素）の結果

[c] 平均化フィルタ（5×5画素）の結果

Gaussianフィルタ（重み付き平均）



■図6.6――ガウス分布



■図6.7――2次元ガウス分布

1次元 $$h_g(x) = \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{x^2}{2\sigma^2})$$

2次元 $$h_g(x, y) = \frac{1}{2\pi\sigma^2}\exp(-\frac{x^2 + y^2}{2\sigma^2})$$

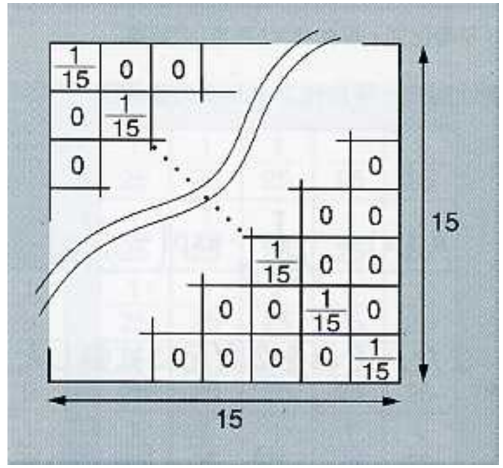
$\frac{1}{16}$	$\frac{2}{16}$	$\frac{1}{16}$
$\frac{2}{16}$	$\frac{4}{16}$	$\frac{2}{16}$
$\frac{1}{16}$	$\frac{2}{16}$	$\frac{1}{16}$

[a] 3×3画素

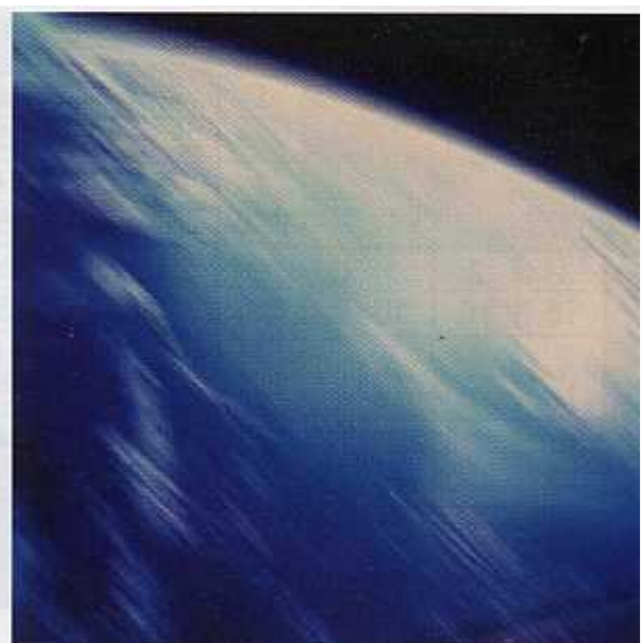
$\frac{1}{256}$	$\frac{4}{256}$	$\frac{6}{256}$	$\frac{4}{256}$	$\frac{1}{256}$
$\frac{4}{256}$	$\frac{16}{256}$	$\frac{24}{256}$	$\frac{16}{256}$	$\frac{4}{256}$
$\frac{6}{256}$	$\frac{24}{256}$	$\frac{36}{256}$	$\frac{24}{256}$	$\frac{6}{256}$
$\frac{4}{256}$	$\frac{16}{256}$	$\frac{24}{256}$	$\frac{16}{256}$	$\frac{4}{256}$
$\frac{1}{256}$	$\frac{4}{256}$	$\frac{6}{256}$	$\frac{4}{256}$	$\frac{1}{256}$

[b] 5×5画素

特定方向の平滑化



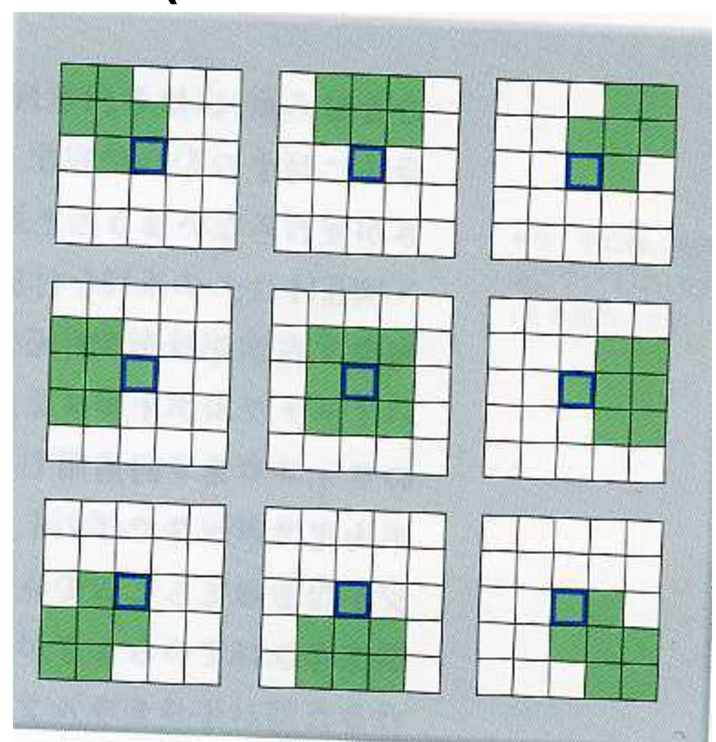
[a] 入力画像



[b] フィルタリング結果

エッジを保存したスムージング

- **平滑化（移動平均法）の欠点：エッジがぼける**
- **エッジを保存したスムージング：下図の9個局所領域内の濃度の分散をそれぞれ計算し、分散が最小(平坦な領域）となる領域の平均濃度をg(i,j)とする。**



メディアンフィルタ(median filter)

- 平滑フィルタ:領域内の画素値の平均(重み付きも含めて)をもとめ、出力画像の画素値とする。
- メディアンフィルタ:領域内の画素値の中央値(メディアン)を出力画像の画素とする。
- 例　ノイズ信号:　　　…,0, 1, 0, 1, 6, 5, 6, 6,…
  平滑化(1x3):　…,0, 0, 1, 2, 4, 6, 6, 6,…
  メディアン(1x3): …,0, 0, 1, 1, 5, 6, 6, 6,…

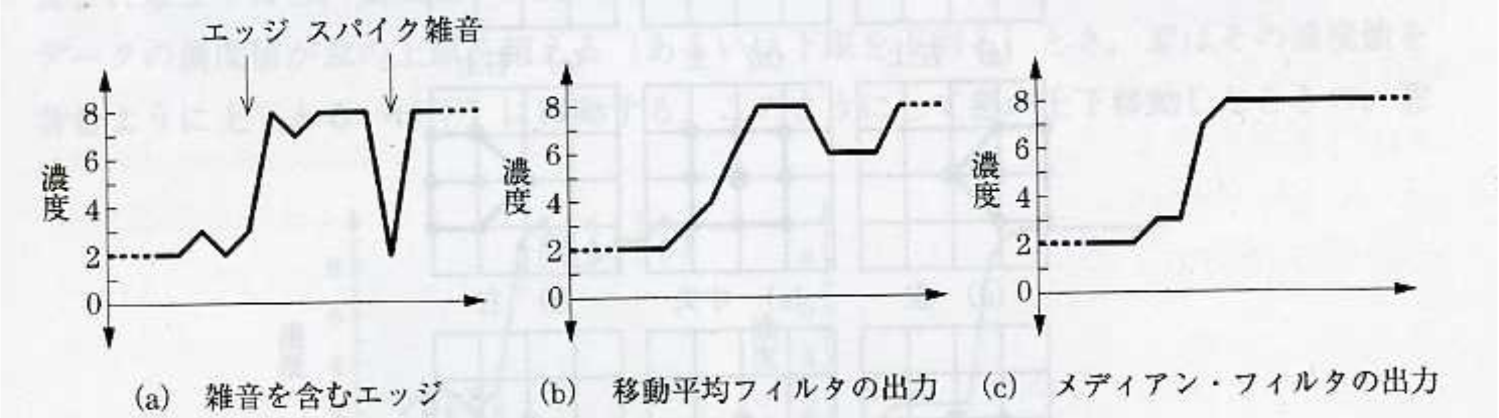


図 4・16　移動平均フィルタとメディアン・フィルタの比較

各種フィルタの比較

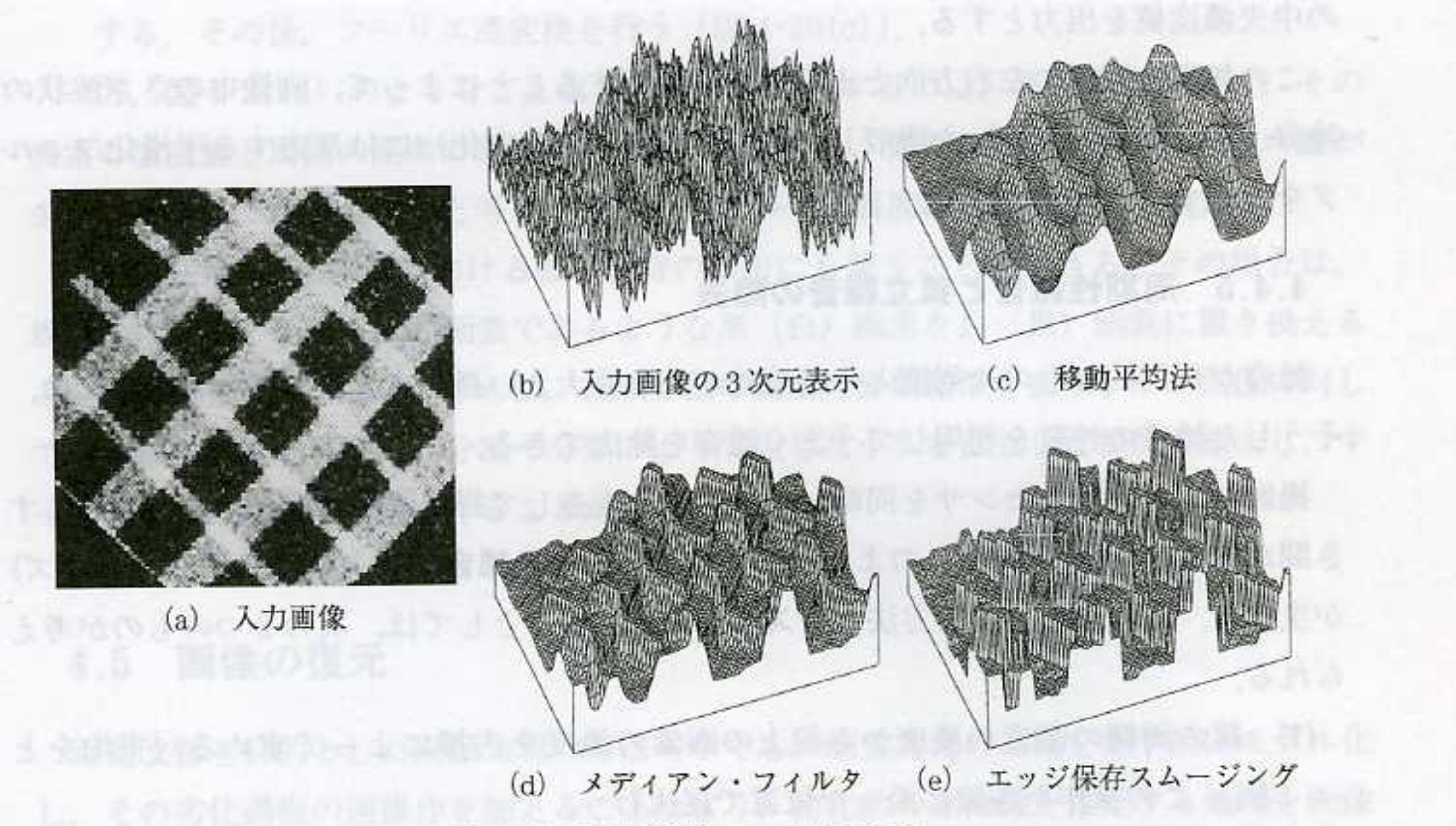


図 4·18 各種平滑化フィルタの比較

ノイズ除去能力：殆ど同じ
エッジ保存能力：平均法 ＜ メディアン ＜ エッジ保存

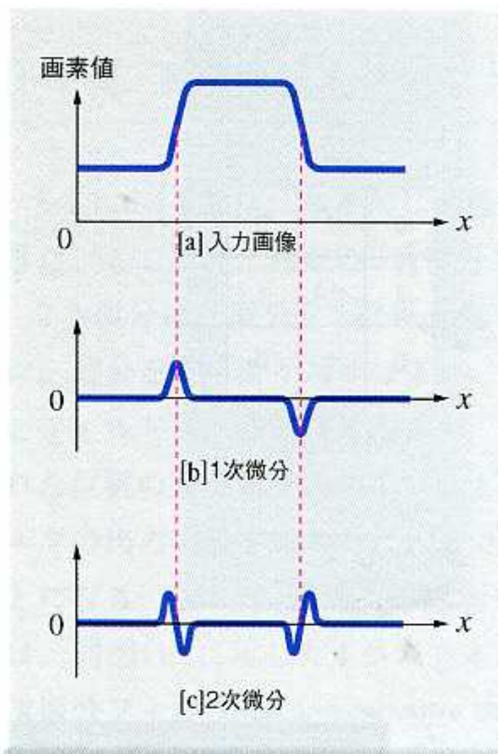
エッジ抽出(high-pass filter)

- 画像の中に急激に変化する部分(高周波成分)をエッジと呼ぶ
- エッジは視覚的に境界線として認識される
- エッジの検出には、勾配(1次微分)やラプラシアン(2次微分)などの微分operatorを用いる。

エッジ信号

勾配:最大値

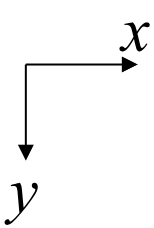
ラプラシアン:零交点



中心差分による偏微分

$$f_x(x,y)=\frac{f\left(x+\Delta x,y\right)-f\left(x-\Delta x,y\right)}{2\Delta x}=\frac{f(x+1,y)-f(x-1,y)}{2}$$

$$f_y(x,y)=\frac{f\left(x,y+\Delta y\right)-f\left(x,y-\Delta y\right)}{2\Delta y}=\frac{f(x,y+1)-f(x,y-1)}{2}$$



0	0	0
-1	0	1
0	0	0

/2

f_x

0	-1	0
0	0	0
0	1	0

/2

f_y

勾配

fの勾配：偏微分f_xとf_yを成分とするベクトル
fの最も急速に変化する方向

$$\nabla f = (f_x, f_y)$$

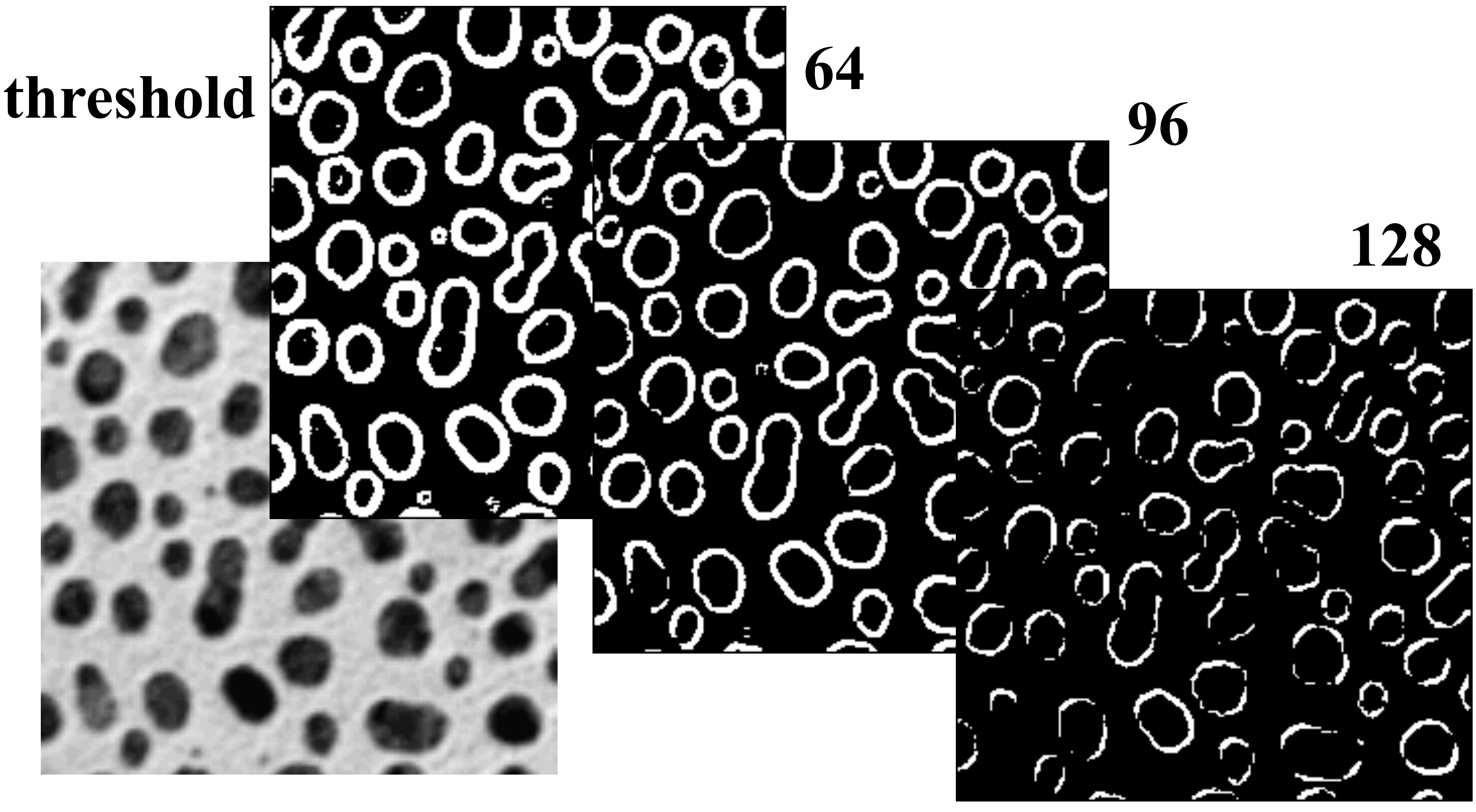
エッジ検出：

$$\left|\nabla f\right| = \sqrt{f_x^2 + f_y^2}$$

2値エッジ画像

$$edge(x, y) = \begin{cases} 255, & \left|\nabla f\right| \geq \theta \\ 0, & otherwise \end{cases}$$

閾値による影響（Sobel filter)



例

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

f_x

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

f_y

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

ノイズの影響を抑えるためのSobelフィルタ、Prewittフィルタ

ノイズを含む画像を微分すると、ノイズを増幅してしまう

ノイズの影響を抑えるために、偏微分の差分計算に、画素値ではなく、その画素を中心とする周辺の平均値を用いる

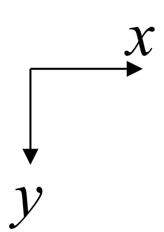
$$f_x(x,y) = \frac{\overline{f(x+1)}^{y} - \overline{f(x-1)}^{y}}{2}$$

$$f_y(x,y) = \frac{\overline{f(y+1)}^{x} - \overline{f(y-1)}^{x}}{2}$$

Sobel Filter

$$\overline{f(x)}^{\,y} = \frac{1}{4}\left\{ f(x, y+1) + 2f(x, y) + f(x, y-1) \right\}$$

$$\overline{f(y)}^{\,x} = \frac{1}{4}\left\{ f(x+1, y) + 2f(x, y) + f(x-1, y) \right\}$$



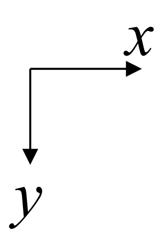
$$\begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array}\ /8 \qquad \begin{array}{|c|c|c|}\hline -1 & -2 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 2 & 1 \\ \hline \end{array}\ /8$$

f_x $\qquad$ f_y

Prewitt Filter

$$\overline{f(x)}^{y} = \frac{1}{3}\left\{f(x, y+1) + f(x, y) + f(x, y-1)\right\}$$

$$\overline{f(y)}^{x} = \frac{1}{3}\left\{f(x+1, y) + f(x, y) + f(x-1, y)\right\}$$



$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} /6 \qquad \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} /6$$

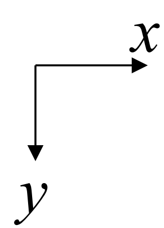
f_x $\qquad$ f_y

ラプラシアン（2次微分）

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

4方向

$$\nabla^2 f = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$



0	1	0
1	-4	1
0	1	0

2値エッジ画像

$$edge(x, y) = \begin{cases} 255, & \left|\nabla^2 f\right| \le \theta \\ 0, & otherwise \end{cases}$$

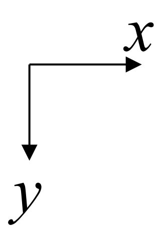
Laplacian filters（エッジ検出）

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

$$f'(j,i) = \frac{-f(j-1,i) - f(j,i-1) + 4f(i,j) - f(j,i+1) - f(j+1,i)}{1}$$

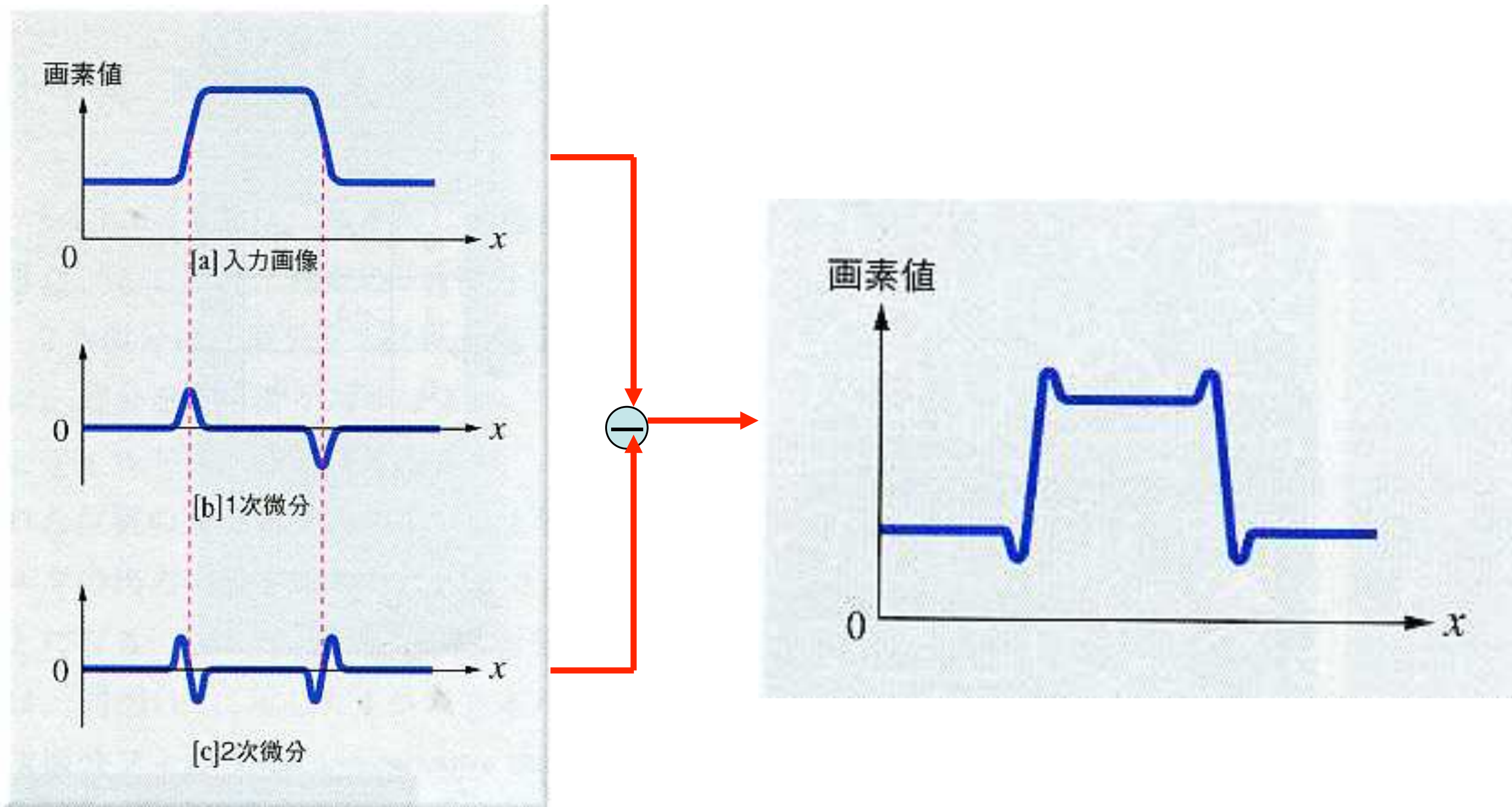
8点ラプラシアン

Sobelフィルタと同じ考え方で、平均値をもちいることによって、ノイズの影響を抑えることができる。＝＞8方向ラプラシアンフィルタ

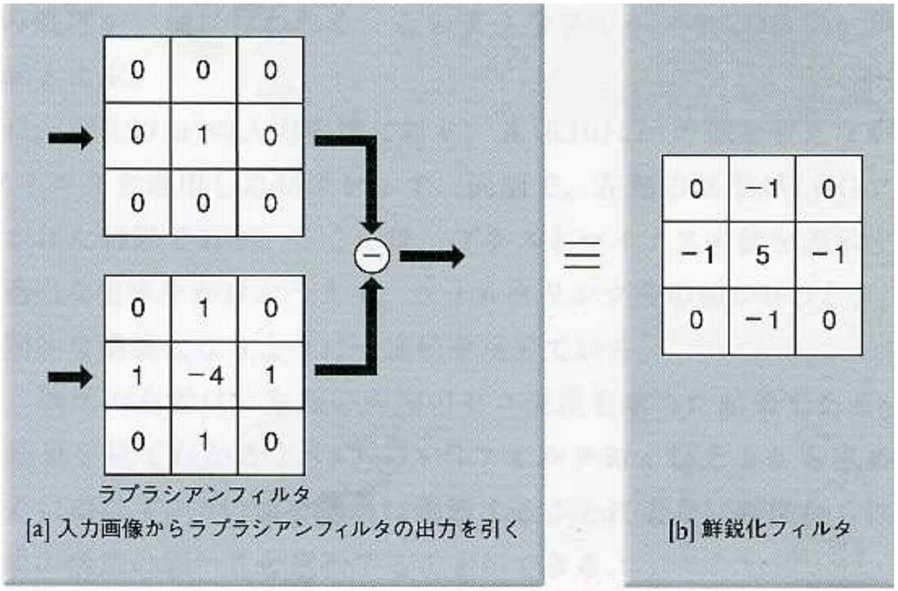


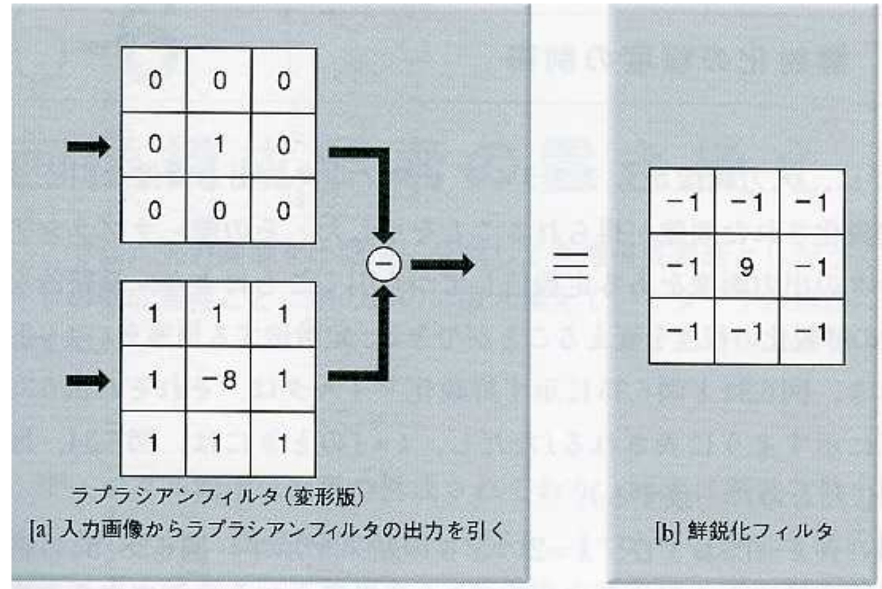
1	1	1
1	-8	1
1	1	1

鮮鋭化フィルタ(sharpening)
（入力画像-ラプラシアン出力）



鮮鋭化結果





a] 入力画像（グレースケール）

[b] 鮮鋭化フィルタの結果

鮮鋭化の程度制御

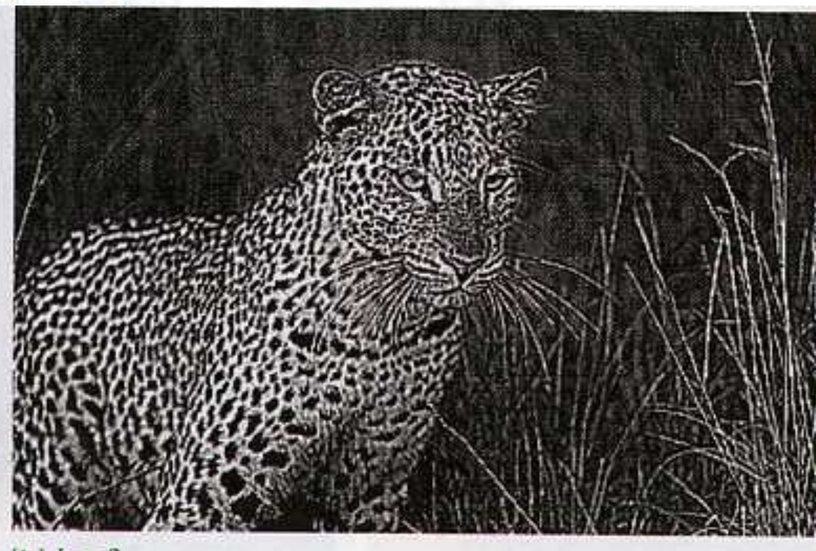
0	−k	0
−k	1+4k	−k
0	−k	0

[a]

−k	−k	−k
−k	1+8k	−k
−k	−k	−k

[b]

[a] $k = 0.5$



[b] $k = 2$

各種フィルターの結果

original image

mean filter =
”smoothing”

Laplace filter

sharpening

Laplacian filterの例(Kernel：3x3)

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

$$g(i,j) = f * h = \sum_{u=-m}^{m}\sum_{v=-m}^{m} h(u,v) f(i-u, j-v)$$

$$\begin{aligned}
&= h(-1,-1) f(i-1, j-1) + h(-1,0) f(i-1, j) + h(-1,1) f(i-1, j+1) \\
&+ h(0,-1) f(i, j-1) \quad + h(0,0) f(i, j) \quad + h(0,1) f(i, j+1) \\
&+ h(1,-1) f(i+1, j-1) \quad + h(1,0) f(i+1, j) \quad + h(1,1) f(i+1, j+1)
\end{aligned}$$

$$\begin{aligned}
&= [0 \cdot f(i-1, j-1) - 1 \cdot f(i-1, j) + 0 \cdot f(i-1, j+1) \\
&-1 \cdot f(i, j-1) \quad + 4 \cdot f(i, j) \quad - 1 \cdot f(i, j+1) \\
&+ 0 \cdot f(i+1, j-1) - 1 \cdot f(i+1, j) + 0 \cdot f(i+1, j+1)]/6
\end{aligned}$$

$$= -f(i-1, j) - f(i, j-1) + 4f(i, j) - f(i, j+1) - f(i+1, j)$$

平滑フィルタ(Low-pass filter)

- 周辺の画素に重み（カーネルの各要素）をかけ、平均値をとる。
- その重み係数は、注目する点(i,j)からの距離が大きいほど小さく、小さいほど大きく。
- 重み係数の総和が１になるように、規格化する。

$$\frac{1}{5}\begin{bmatrix}0 & 1 & 0\\ 1 & 1 & 1\\ 0 & 1 & 0\end{bmatrix}$$

3x3平均値フィルタ

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

3x3重み付き平均値フィルタ（中心強調）

$$\frac{1}{140}\begin{bmatrix}1 & 1 & 2 & 2 & 2 & 1 & 1\\ 1 & 2 & 2 & 4 & 2 & 2 & 1\\ 2 & 2 & 4 & 8 & 4 & 2 & 2\\ 2 & 4 & 8 & 16 & 8 & 4 & 2\\ 2 & 2 & 4 & 8 & 4 & 2 & 2\\ 1 & 2 & 2 & 4 & 2 & 2 & 1\\ 1 & 1 & 2 & 2 & 2 & 1 & 1\end{bmatrix}$$

7x7Gaussianフィルタ