

画像のフィルタリング処理

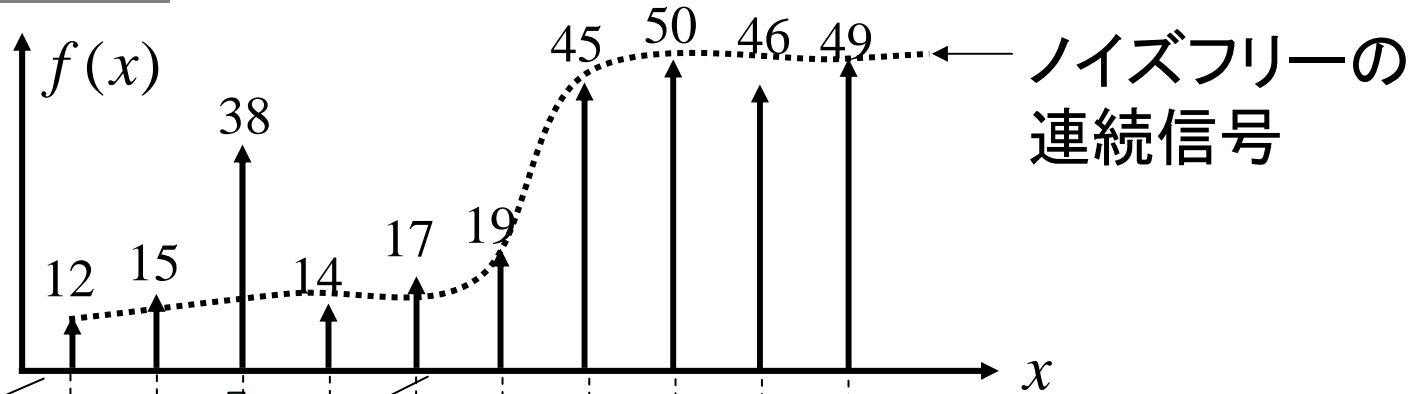
講義内容

- 実空間フィルタリング
 - 平滑化(LPF)
 - エッジ強調(HPF)
 - Laplacian of Gaussian (LOG)フィルタ(BPF)
- 周波数空間フィルタリング
 - LPF, HPF, BPF
 - 周波数選択的フィルタ
- 線形シフトインバリエントシステムと劣化画像復元
 - 線形システム
 - 劣化画像の復元
- MATLABを用いたデモ

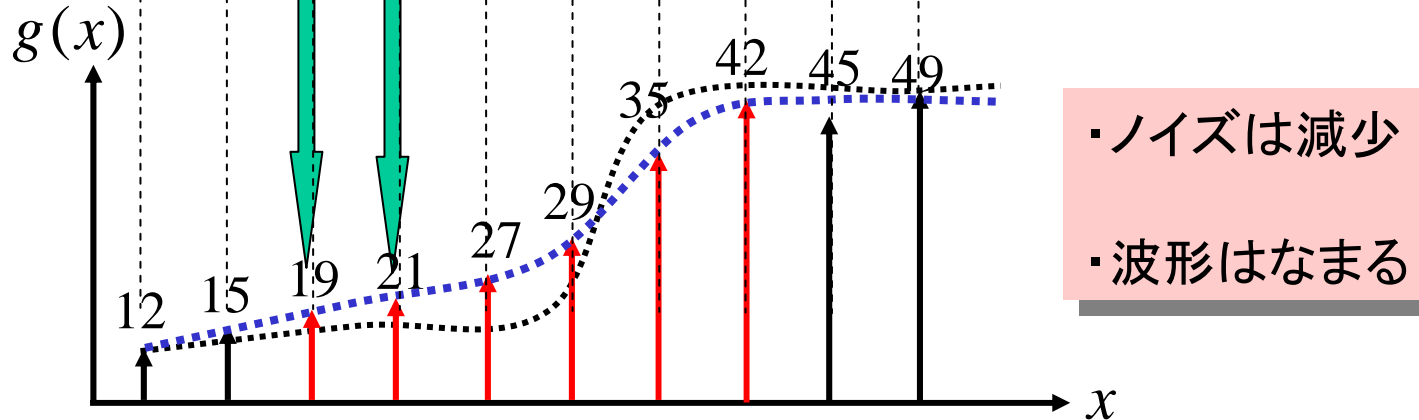
ノイズ除去 (1) 平滑化处理 — 1次元 —

5点の平滑化の場合

処理前

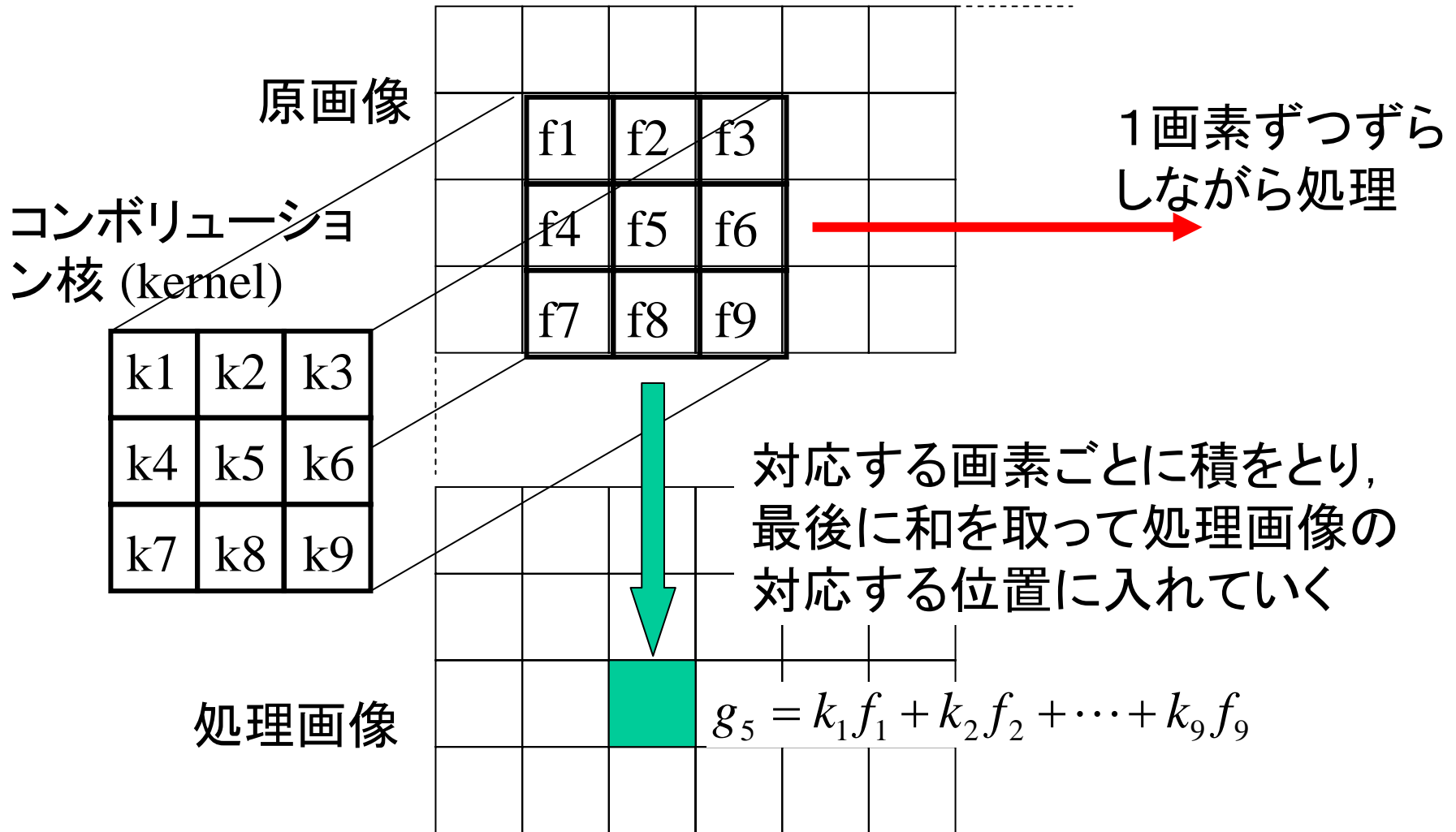


処理後



$$g_n = \frac{1}{5}(f_{n-2} + f_{n-1} + f_n + f_{n+1} + f_{n+2}) = \frac{1}{5} \sum_{i=-2}^2 f_{n+i}$$

デジタル画像に対するコンボリューション処理



ノイズ除去 (1) 平滑化処理 — 2次元 —

3 × 3の平滑化の場合

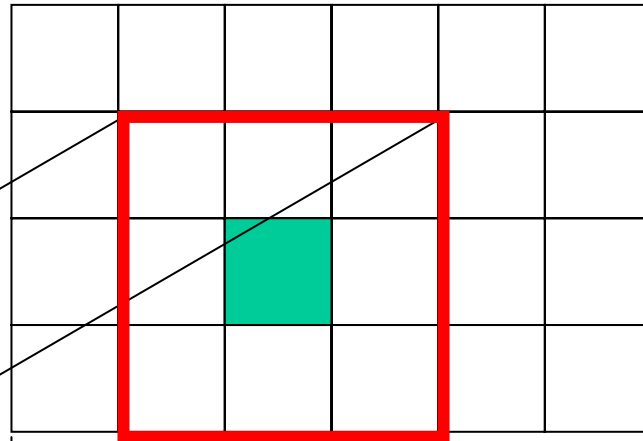
コンボリューション核
(kernel)

k1	k2	k3
k4	k5	k6
k7	k8	k9

||

$\frac{1}{9} \times$

1	1	1
1	1	1
1	1	1

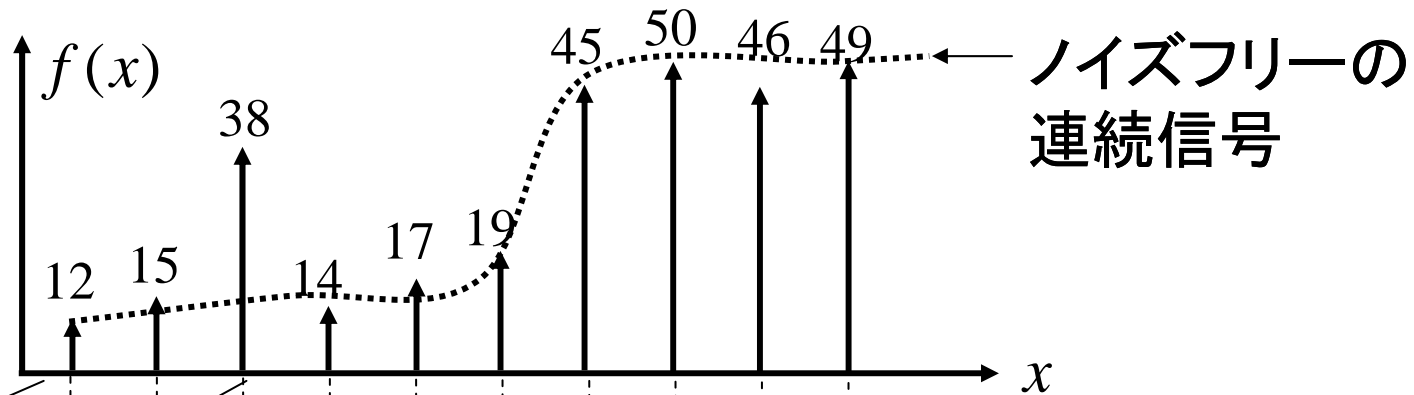


このエリアの平均値を用いる

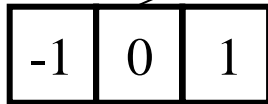
エッジ強調 — 1次元 —

差分フィルタ: 近傍領域の差分値で置き換えていく方法

処理前

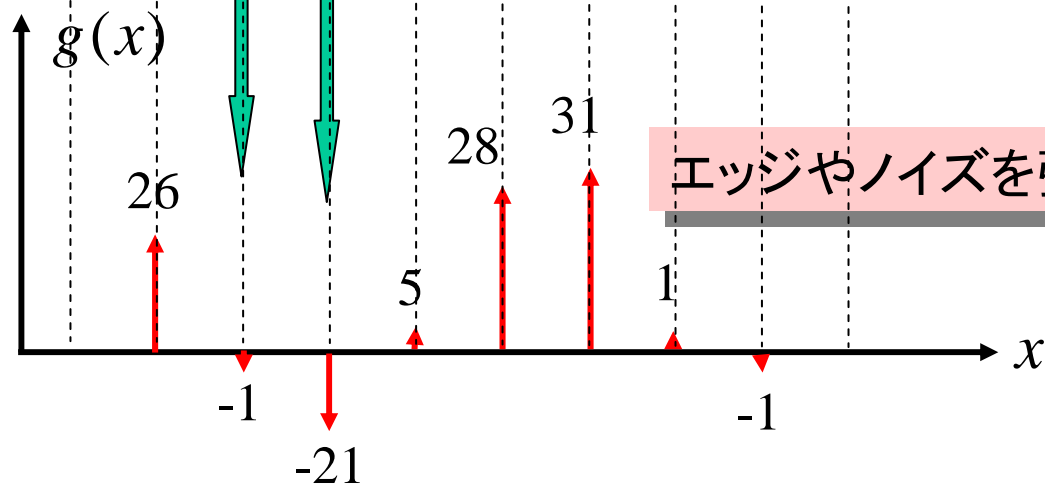


Kernel



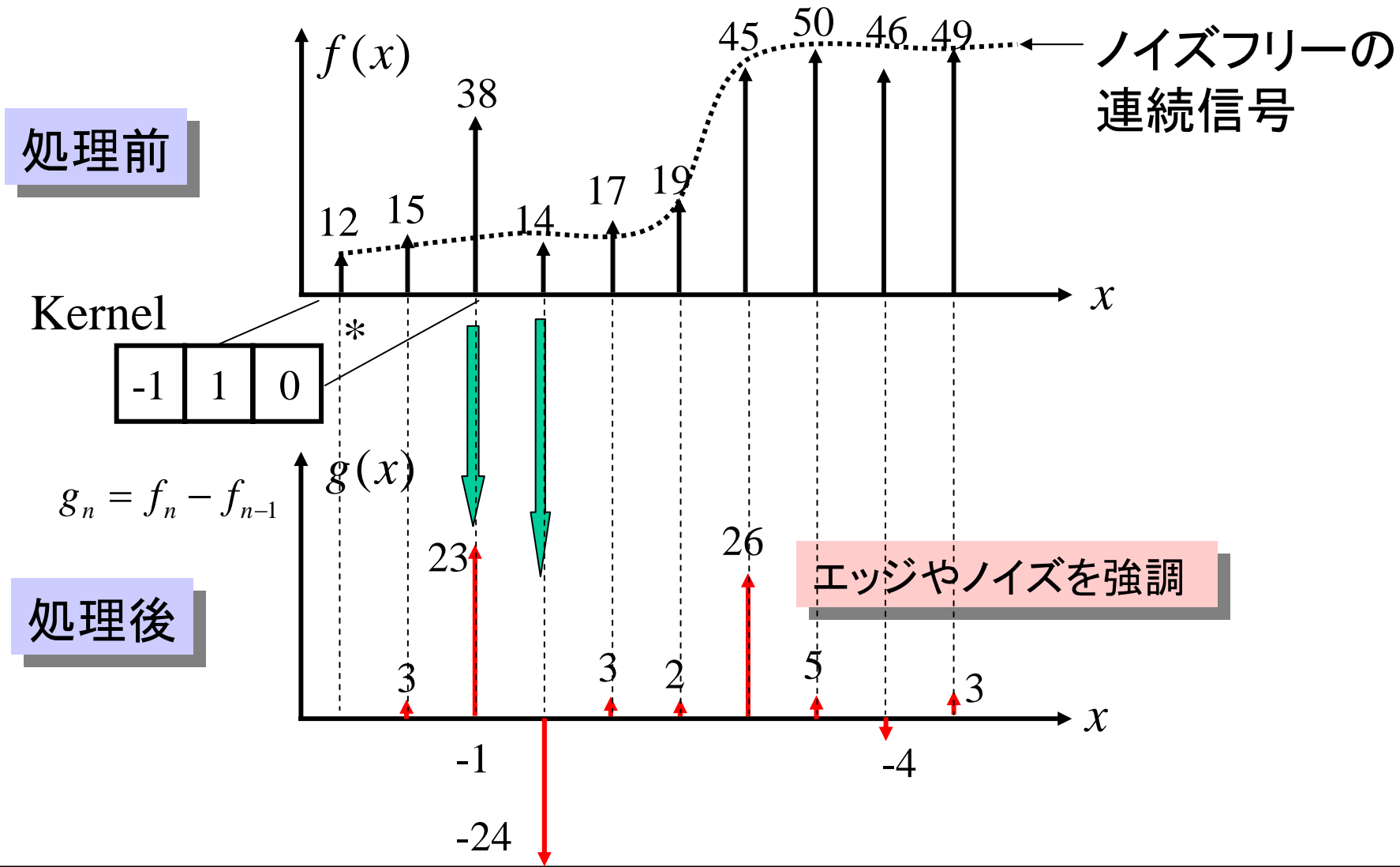
$$g_n = f_{n+1} - f_{n-1}$$

処理後



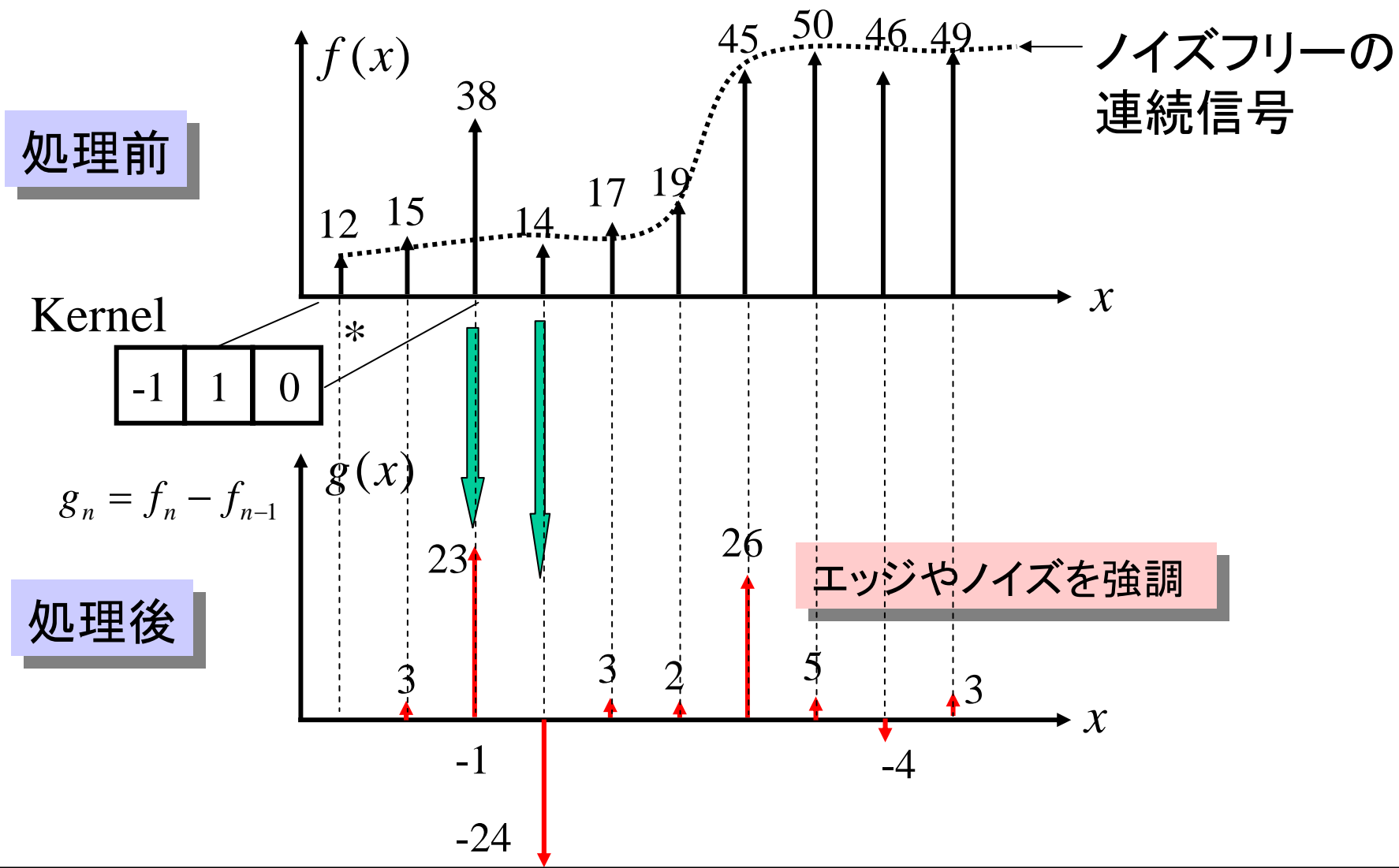
エッジ強調 — 1次元 —

差分フィルタ: 近傍領域の差分値で置き換えていく方法



エッジ強調 — 1次元 —

差分フィルタ: 近傍領域の差分値で置き換えていく方法

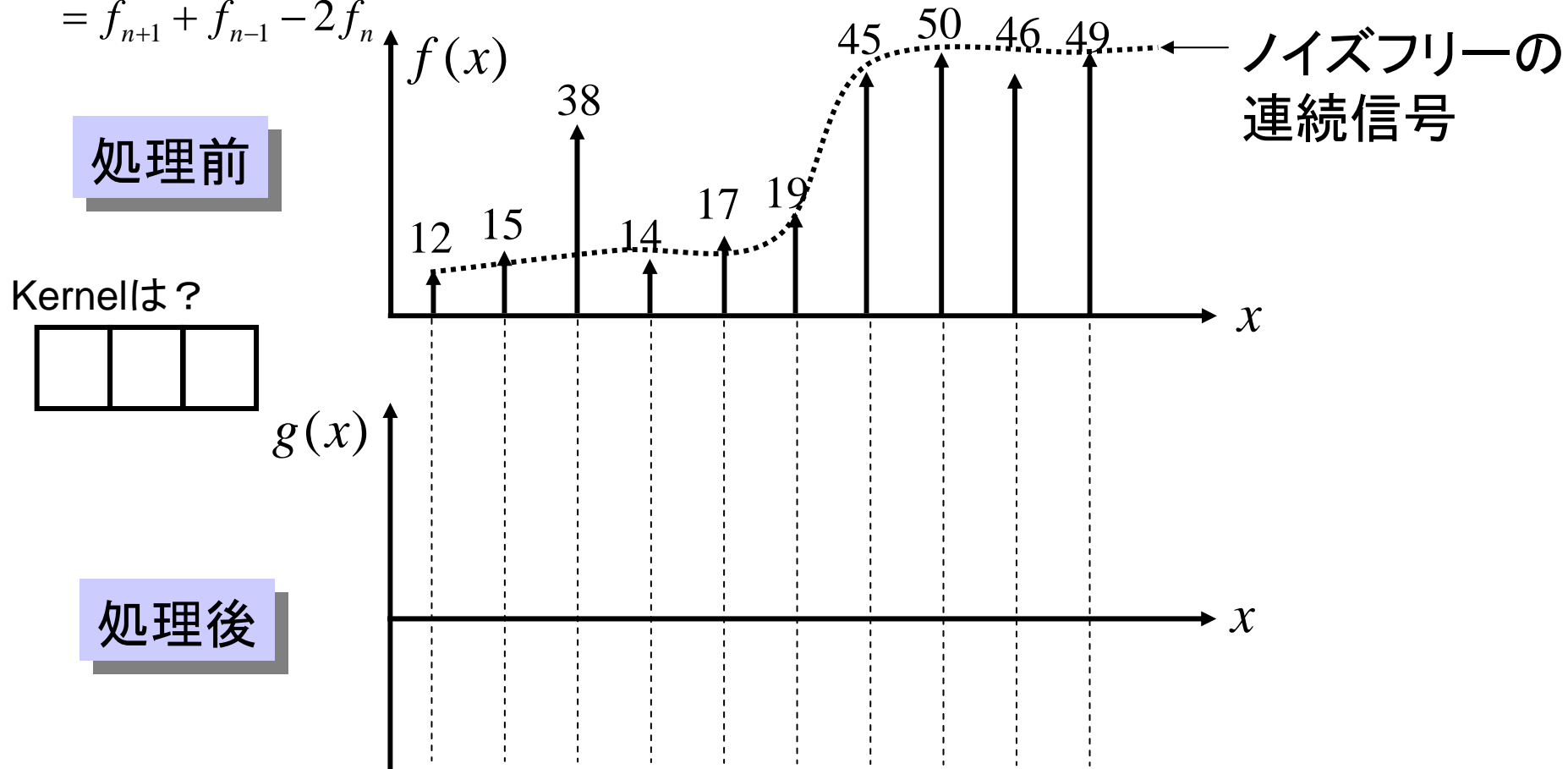


エッジ強調－1次元－ラプラシアンフィルタ

差分フィルタ: 近傍領域の2階微分(ラプラシアン)で置き換えていく方法

$$g_n = (f_{n+1} - f_n) - (f_n - f_{n-1})$$

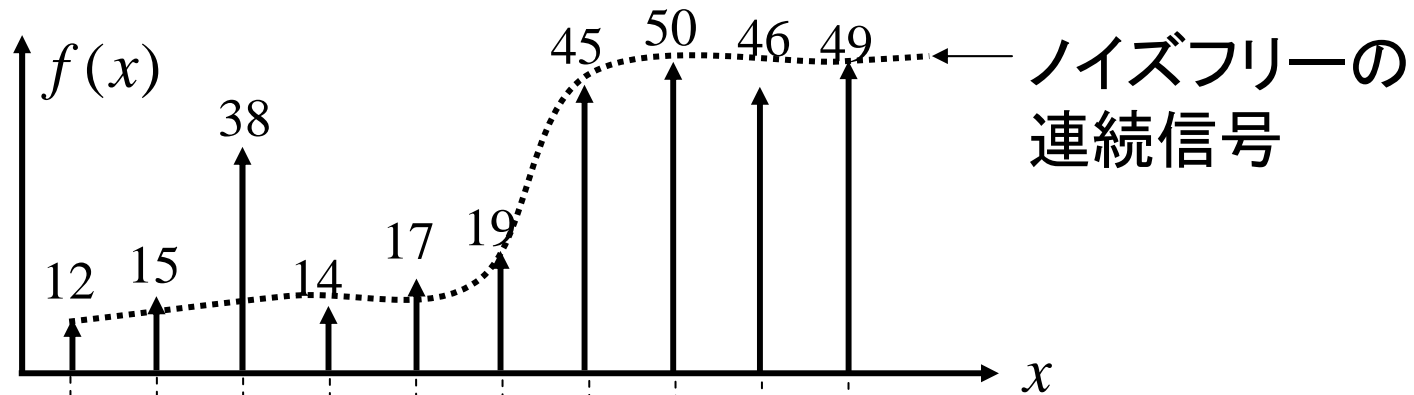
$$= f_{n+1} + f_{n-1} - 2f_n$$



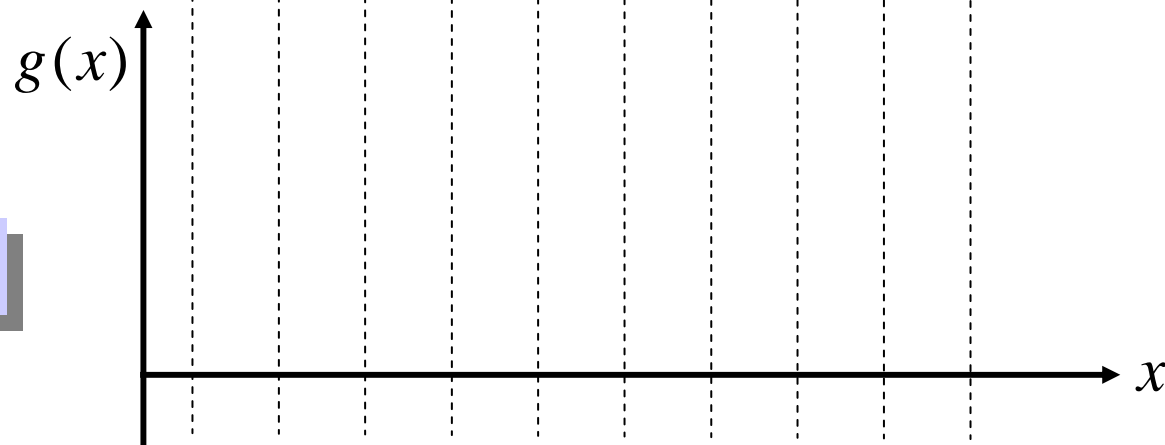
ノイズ除去ー1次元ー メディアンフィルタ

差分フィルタ: 近傍領域の中央値(メディアン)で置き換えていく方法

処理前



処理後



$$g_n = \text{median}\{f_{n-2}, f_{n-1}, f_n, f_{n+1}, f_{n+2}\}$$

注: この処理は線形演算ではなく, コンボリューション処理とは呼ばない

エッジ強調フィルタ — 2次元 —

Laplacian filter_x

			x
	0	-1	0
	-1	4	-1
	0	-1	0
y			

中央と周辺との差分

f1	f2	f3
f4	f5	f6
f7	f8	f9

y方向の2回差分 $(f_8 - f_5) - (f_5 - f_2)$

x方向の2回差分 $(f_6 - f_5) - (f_5 - f_4)$

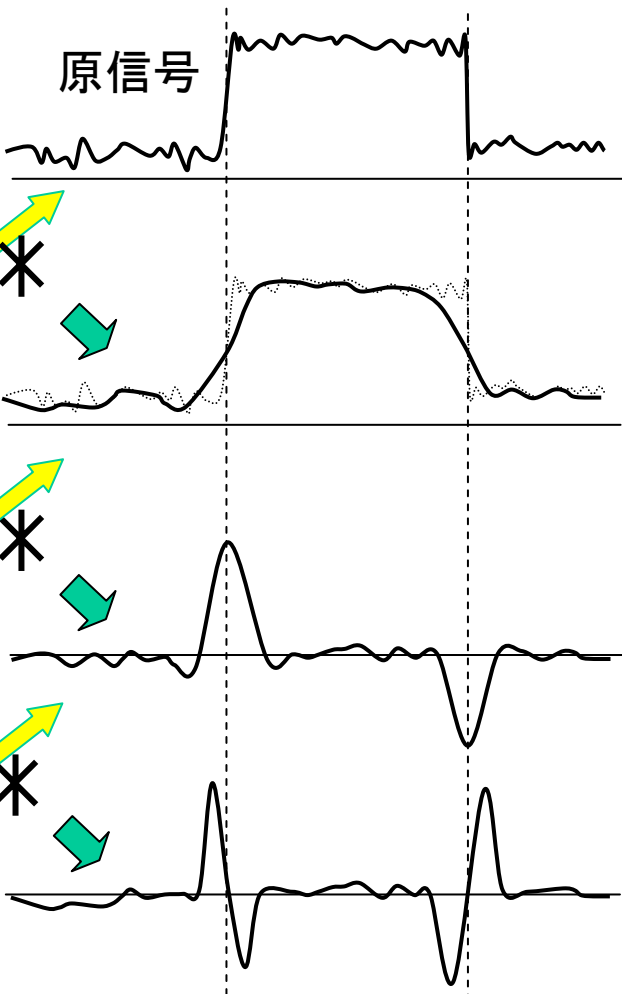
Sobel filter_x

			x
	-1	0	1
	-2	0	2
	-1	0	1
y			

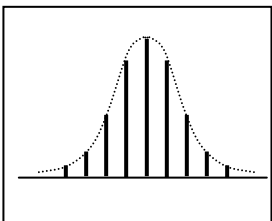
x方向には差分
y方向には平滑化

Laplacian of Gaussian (LoG) フィルタ

1次元信号に対するLOG処理の模式的説明



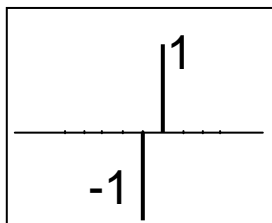
Gaussianを
コンボリューション



* (convolution symbol)

ぼかし処理により
ノイズが低減する

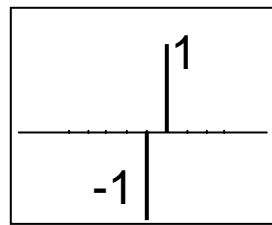
1次微分



* (convolution symbol)

エッジに立ち上がり
が、山や谷になる。

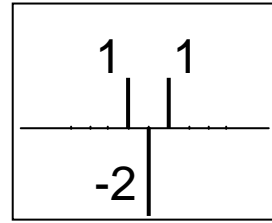
さらに微分
(計2次微分)



* (convolution symbol)

エッジの山や谷
が0近辺の値に
なる。

参考:ラプラシ
アン演算子



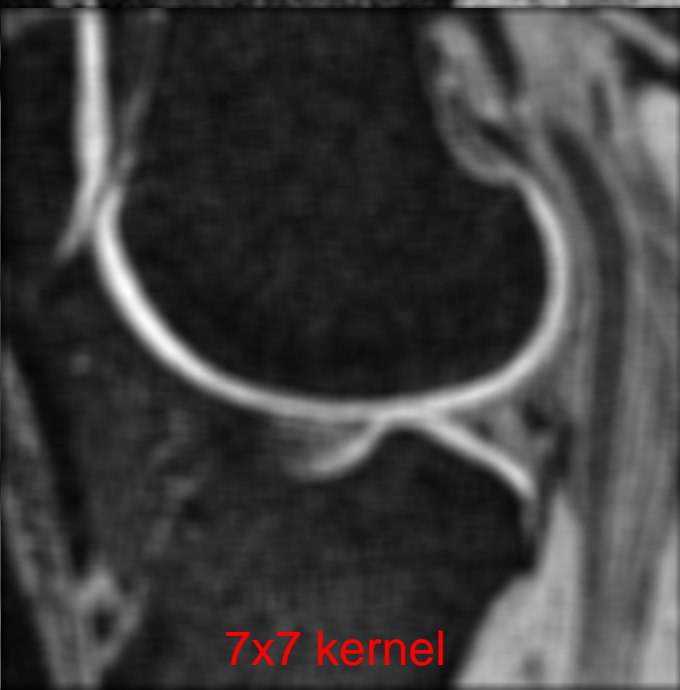
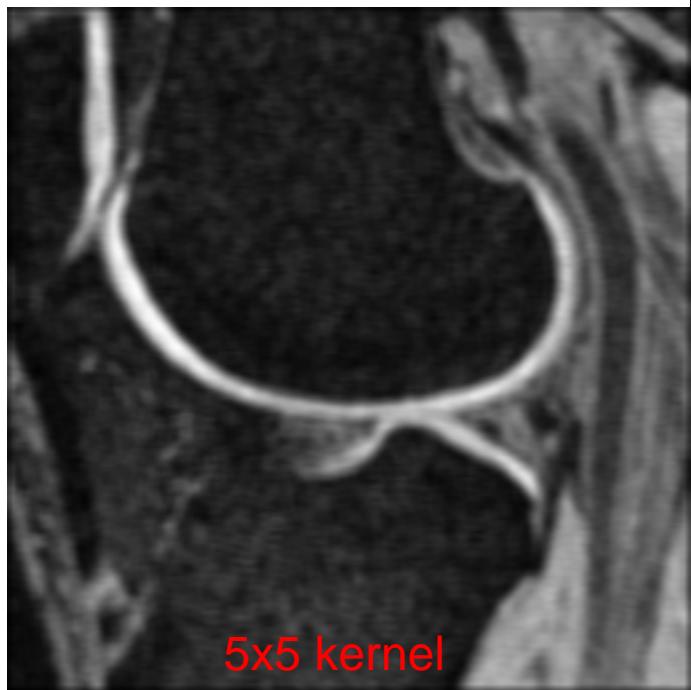
⇒ ゼロクロス法を使って検出すればよい

Kernel:

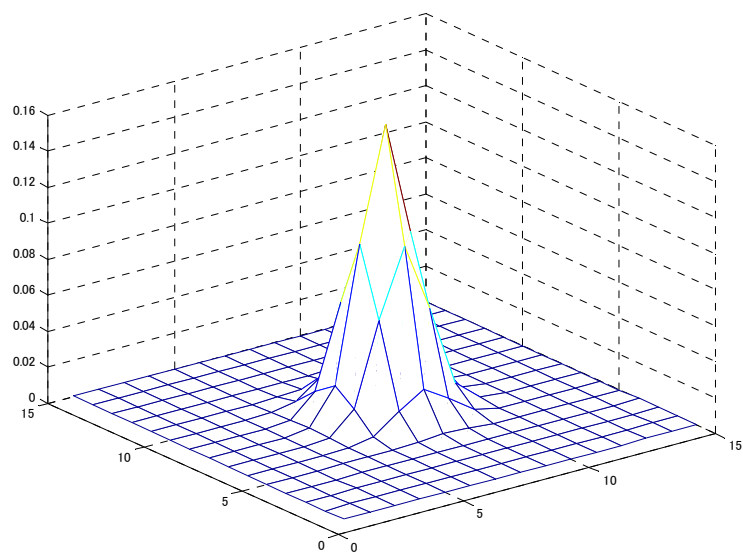
$$\frac{1}{n^2} \begin{bmatrix} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \dots & 1 \end{bmatrix}$$

n

n

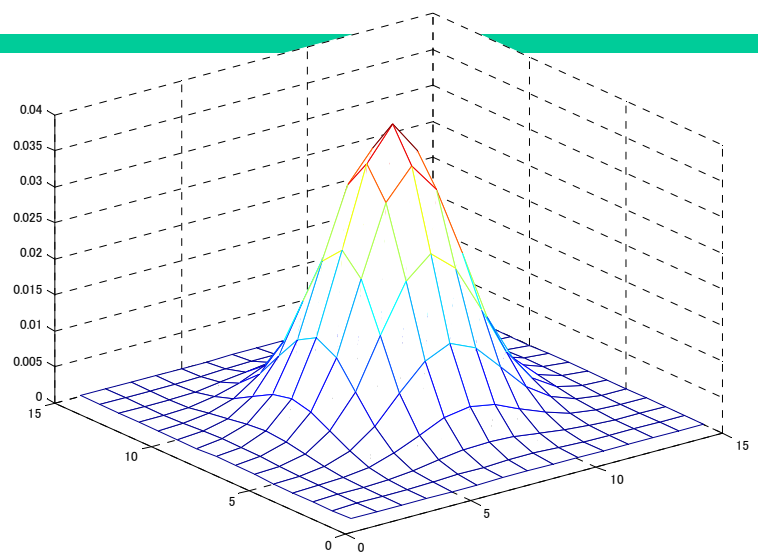


平滑化 フィルタ (Gaussian kernel)

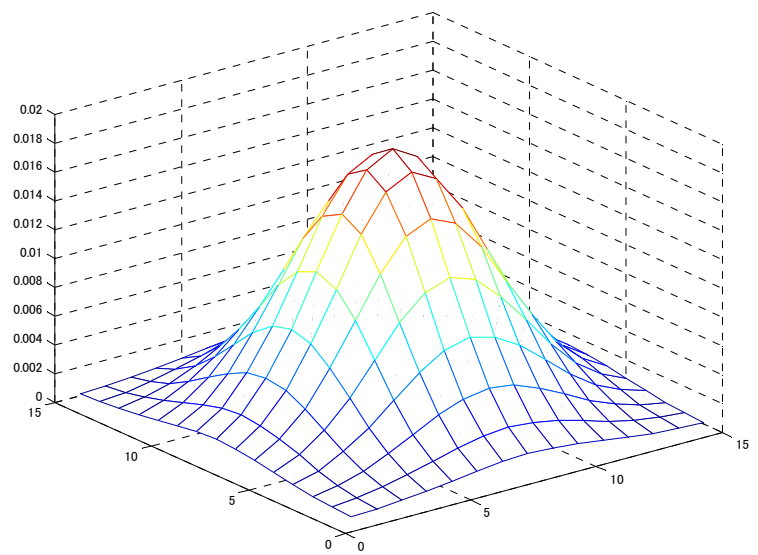


Sigma = 1





Sigma = 2



Sigma = 3

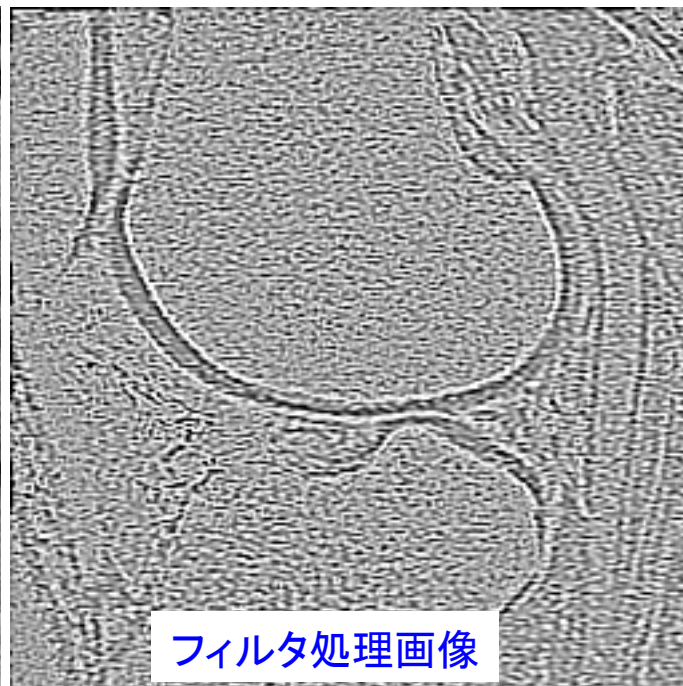


フィルタ処理画像



フィルタ処理画像

Laplacianフィルタ





オリジナル画像

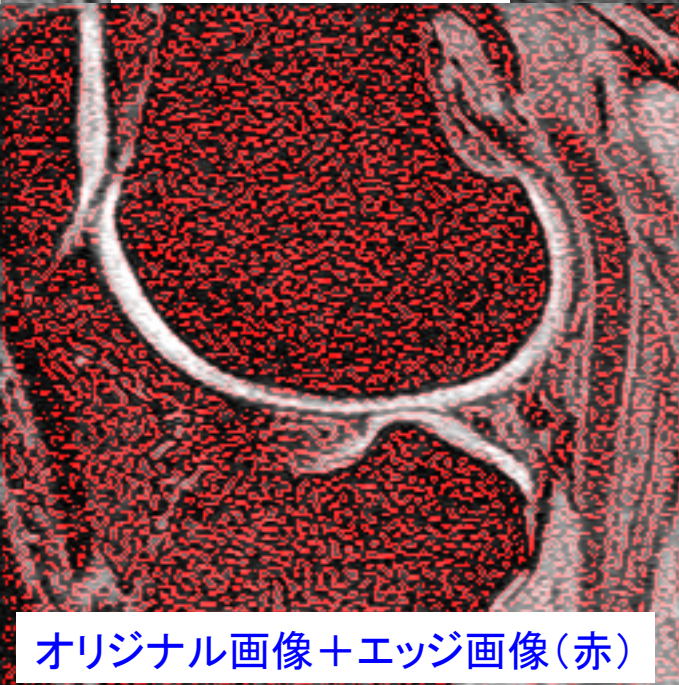


LOGフィルタ後画像

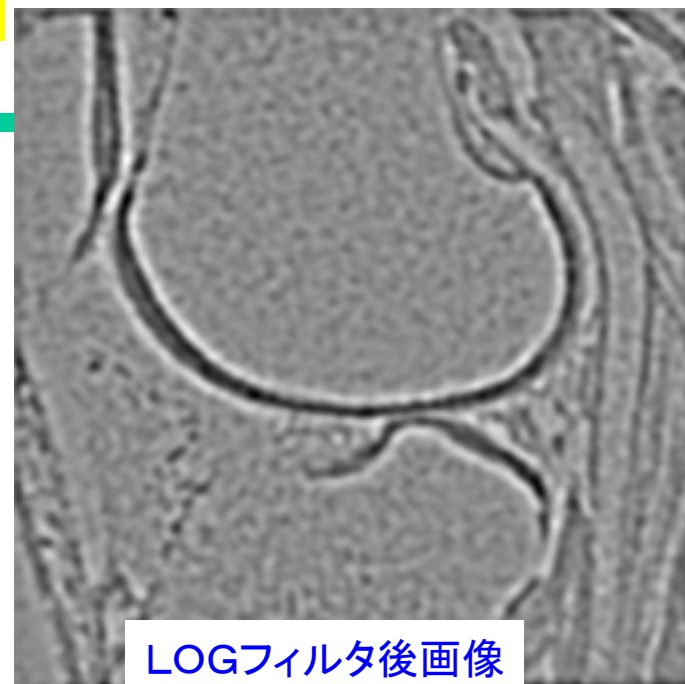
Gaussian
Kernel size:
7x7



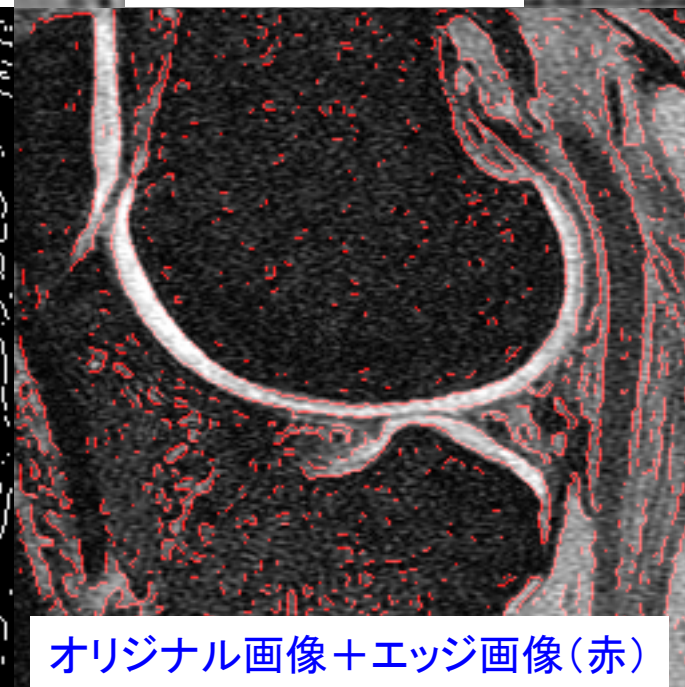
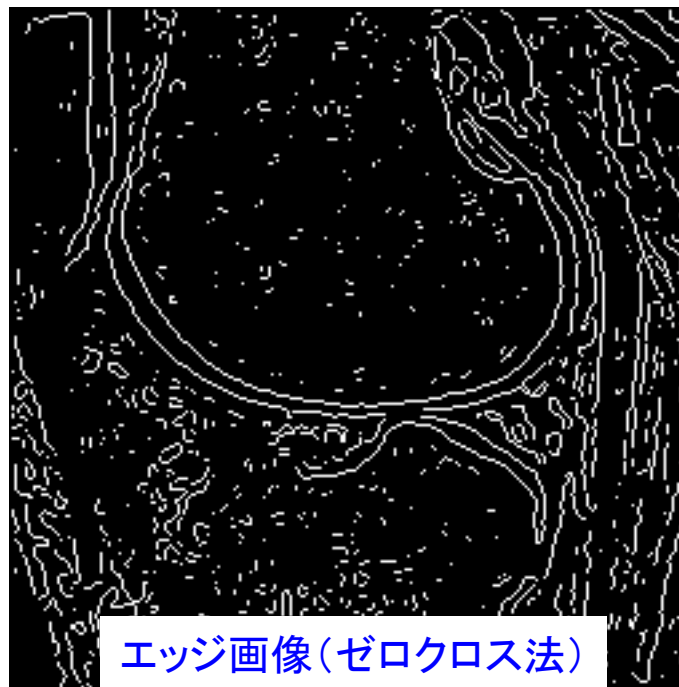
エッジ画像(ゼロクロス法)



オリジナル画像+エッジ画像(赤)



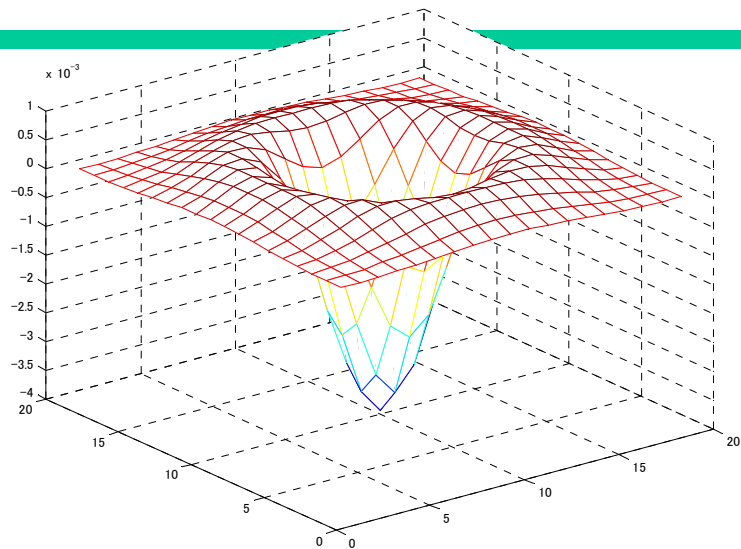
Gaussian
Kernel size:
13x13



エッジ画像(ゼロクロス法)

オリジナル画像+エッジ画像(赤)

LOG フィルタ(Gaussian Kernel size: 19x19)



LOGフィルタ後画像

Gaussian
Kernel size:
19x19

エッジ画像(ゼロクロス法)



オリジナル画像+エッジ画像(赤)

Prewittフィルタ

-1	-1	1
0	0	0
1	1	1

y



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

y

