

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)

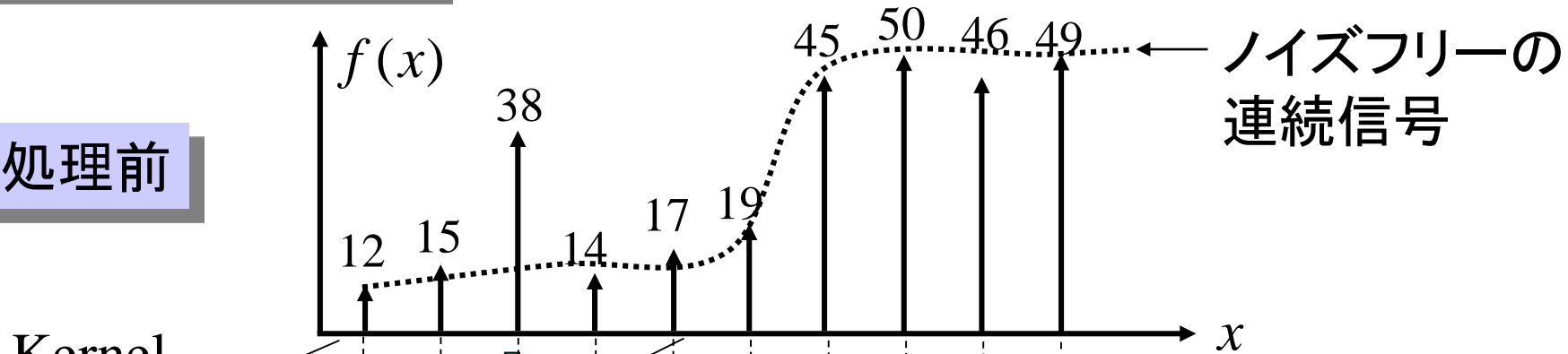
- 周波数空間フィルタリング
 - LPF, HPF, BPF
 - 周波数選択的フィルタ

- 線形シフトインバリア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

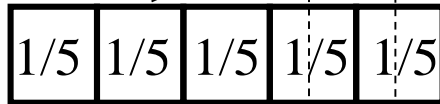
ノイズ除去 (1) 平滑化处理 — 1次元 —

5点の平滑化の場合

処理前



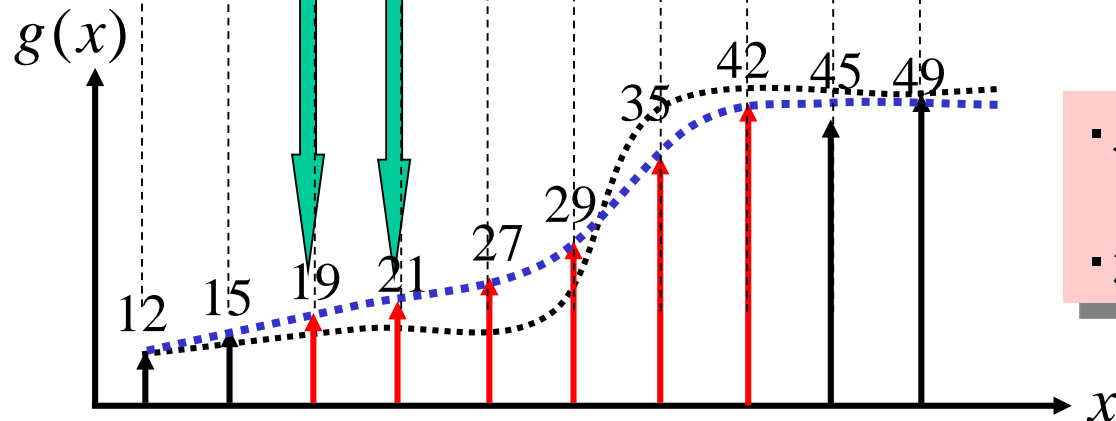
Kernel



5つの値の平均値で置き換えていく

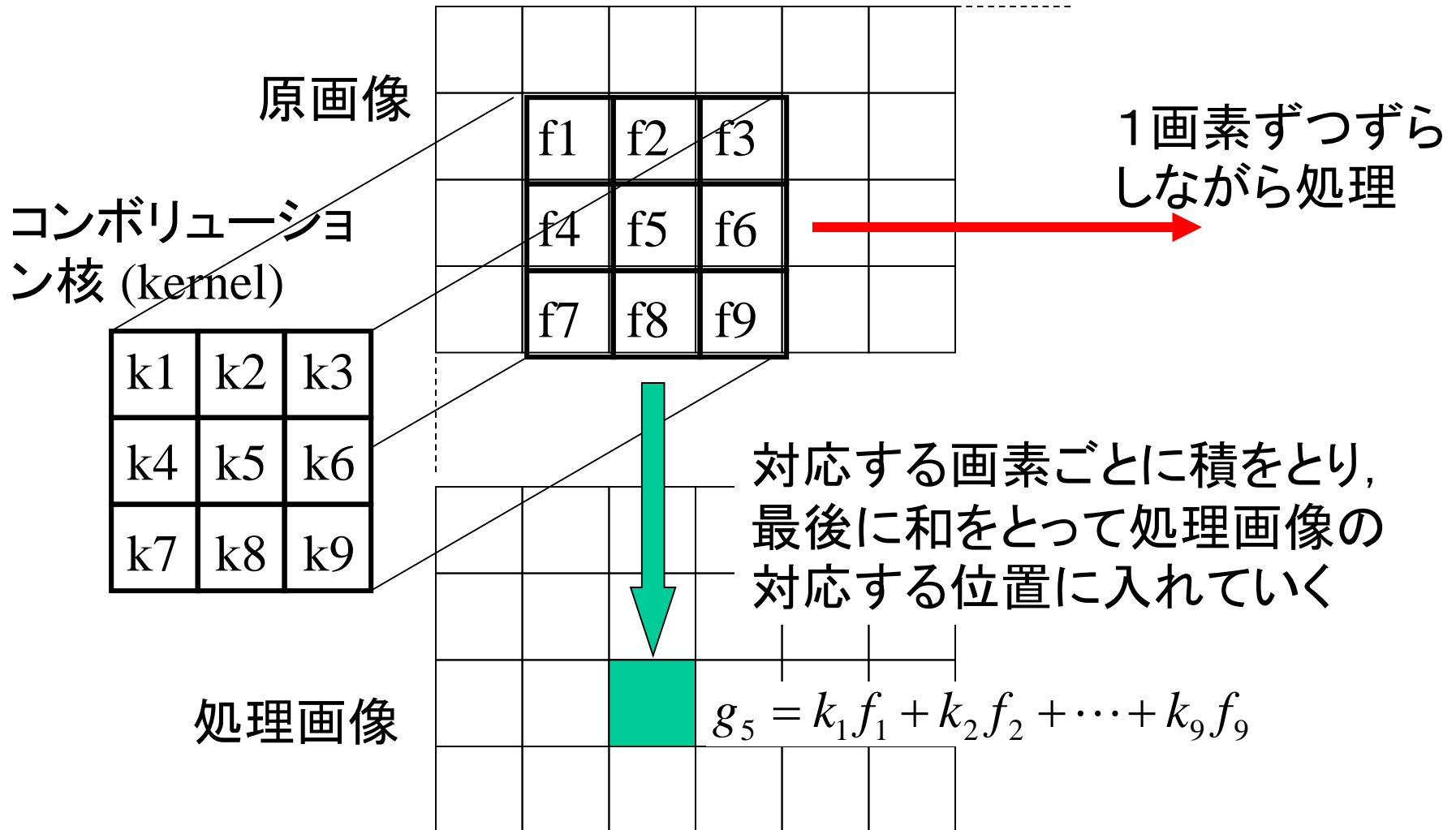
...

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

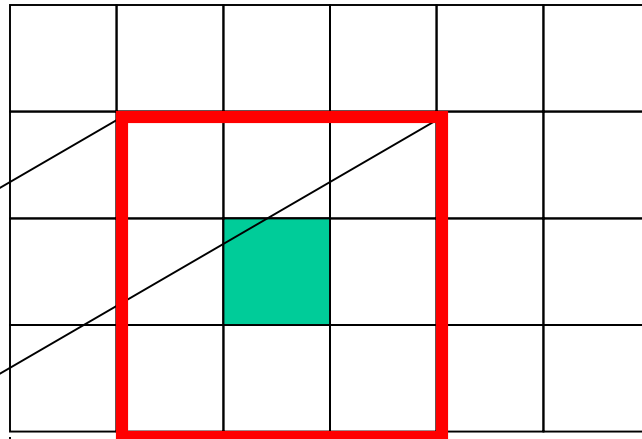
コンボリューション核
(kernel)

k1	k2	k3
k4	k5	k6
k7	k8	k9

||

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

1	1	1
1	1	1
1	1	1

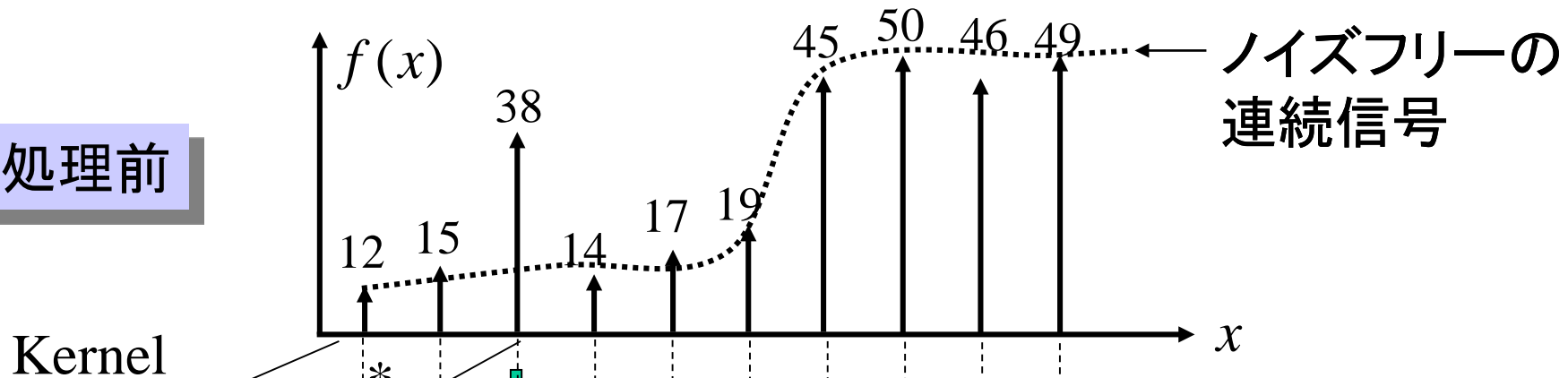


このエリアの平均値を用いる

エッジ強調 — 1次元 —

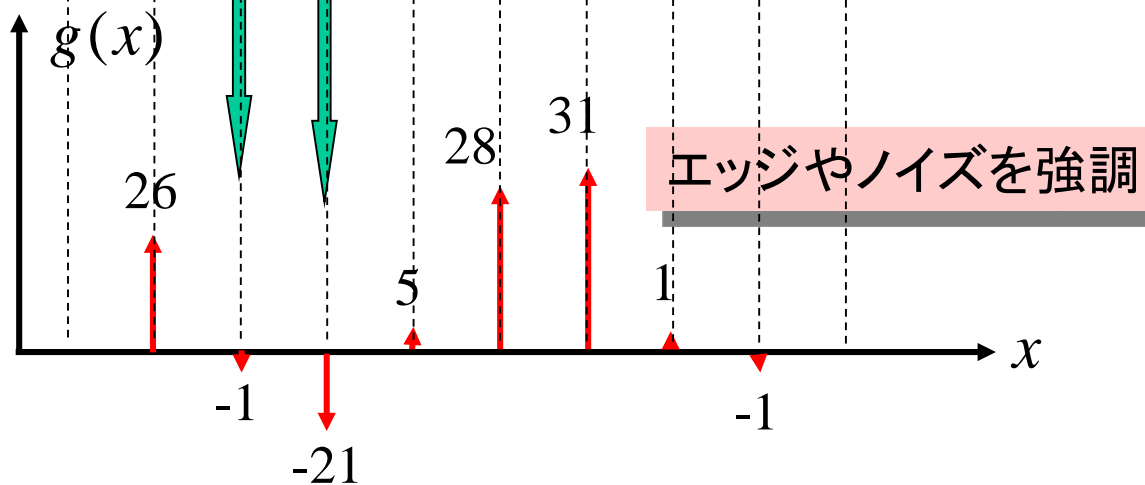
差分フィルタ: 近傍領域の差分値で置き換えていく方法

処理前



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

処理後

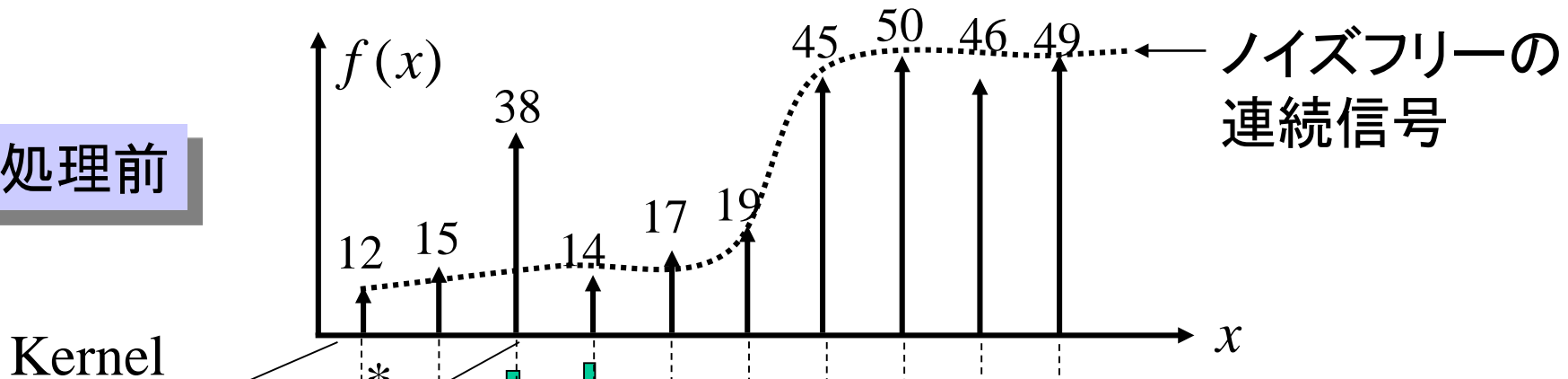


エッジやノイズを強調

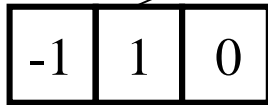
エッジ強調 - 1次元 -

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

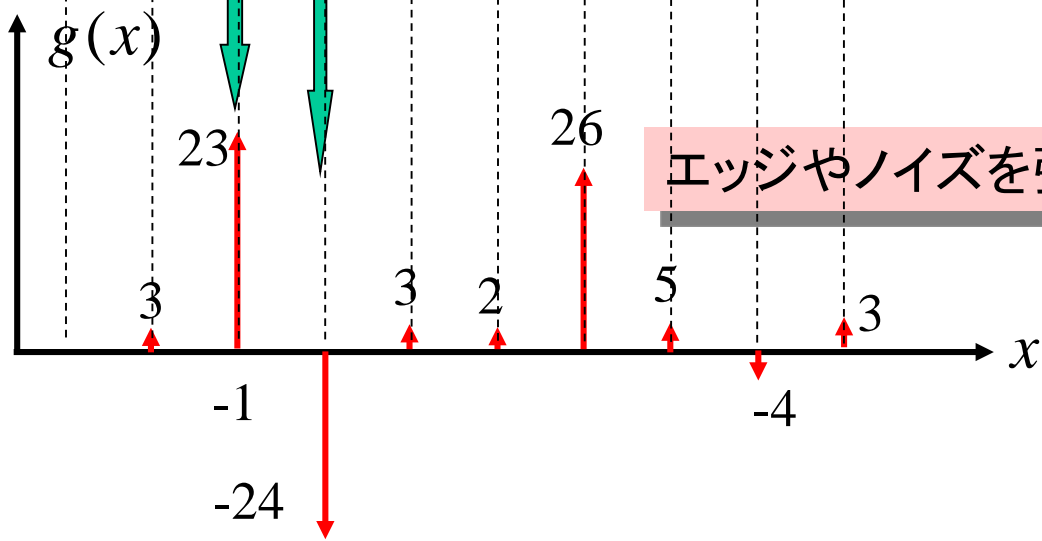


Kernel



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

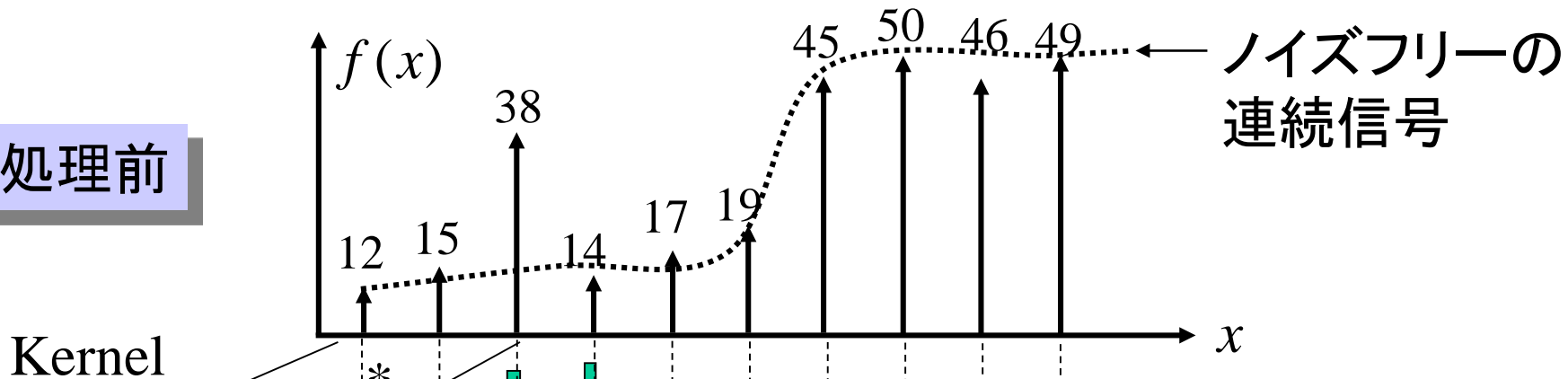
処理後



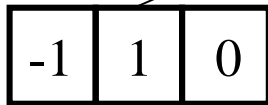
エッジ強調 - 1次元 -

差分フィルタ: 近傍領域の差分値で置き換えていく方法

処理前

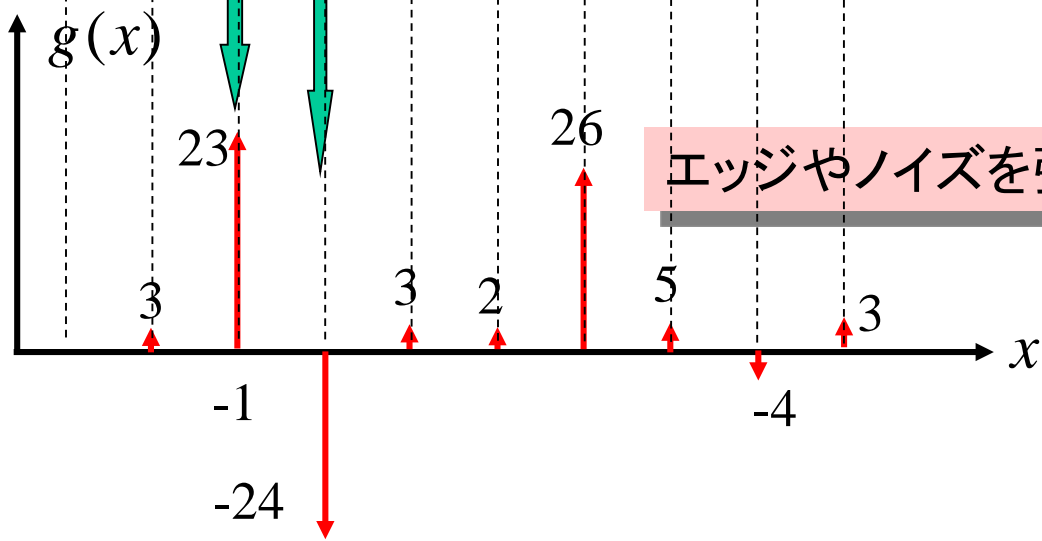


Kernel



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

処理後

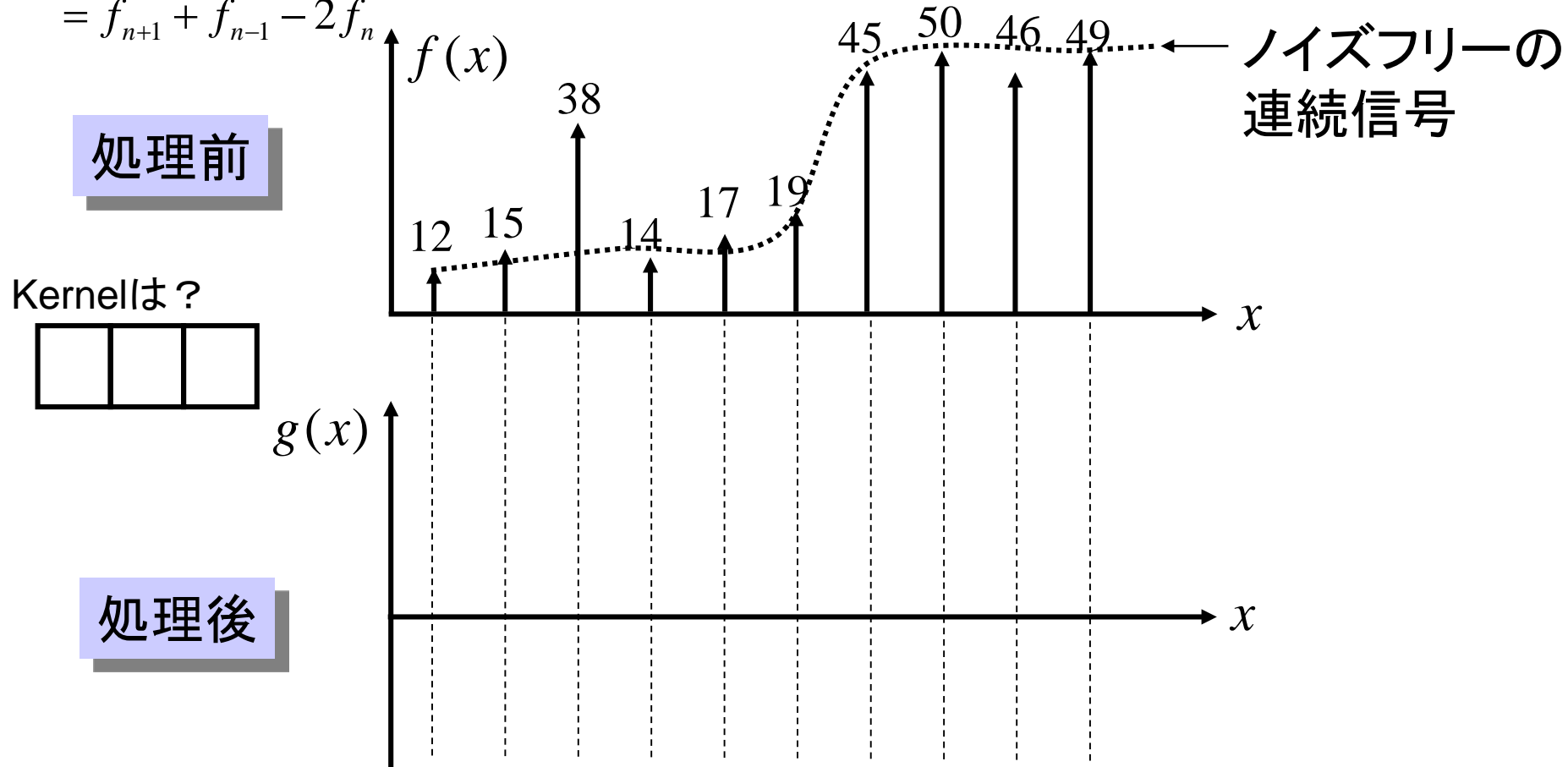


エッジ強調－1次元－ラプラシアンフィルタ

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

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

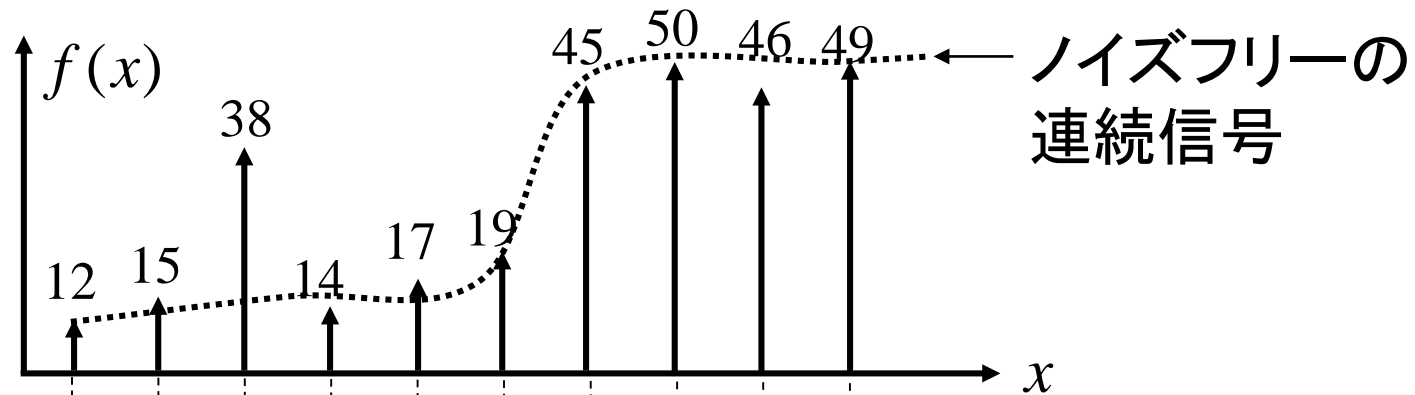
$$= f_{n+1} + f_{n-1} - 2f_n$$



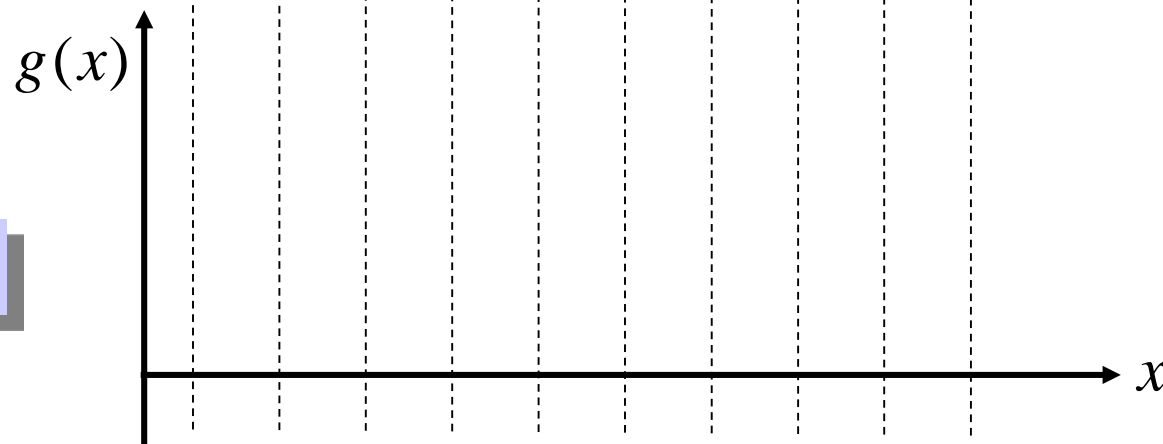
ノイズ除去ー1次元ー メディアンフィルタ

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



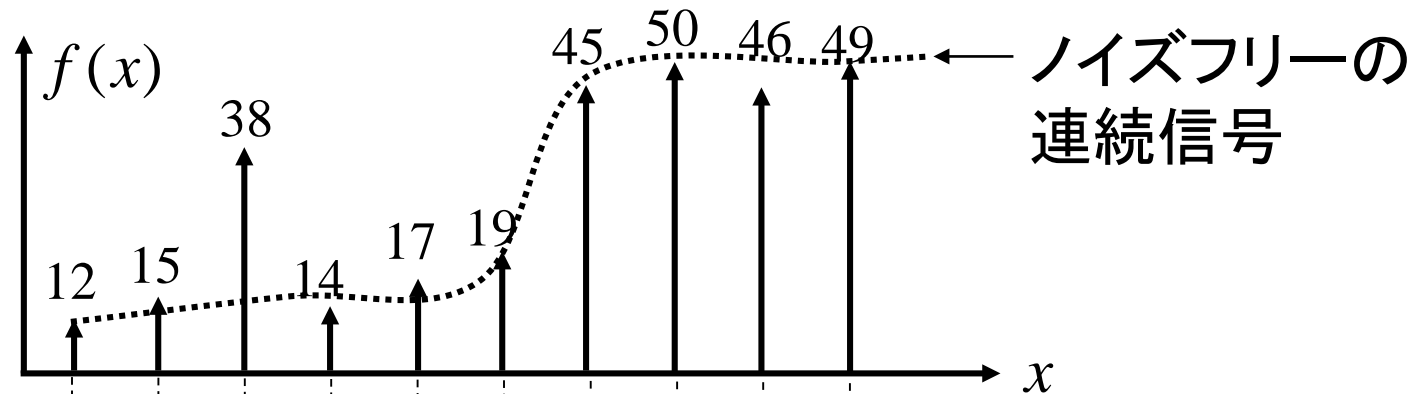
$$g_n = \text{median}\{f_{n-2}, f_{n-1}, f_n, f_{n+1}, f_{n+2}\}$$

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

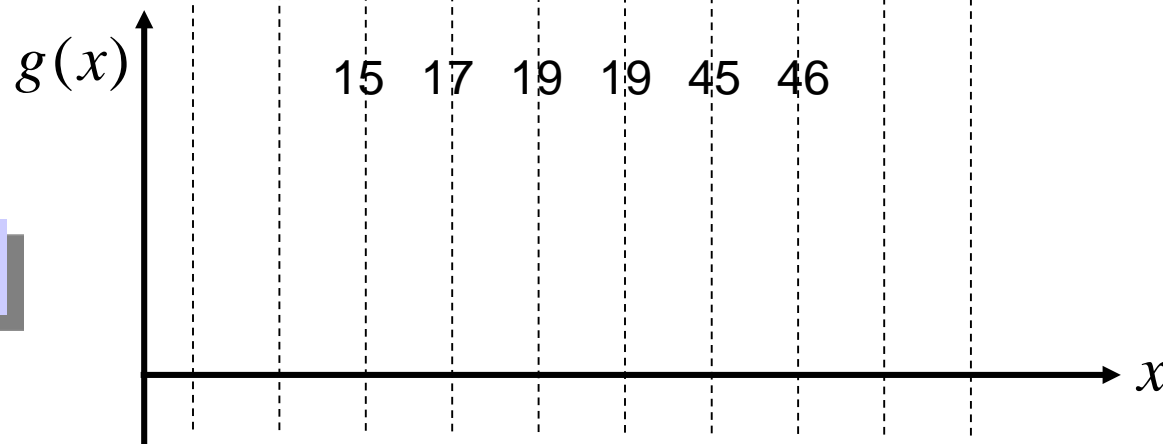
ノイズ除去ー1次元ー メディアンフィルタ

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 $(f_6 - f_5) - (f_5 - f_4)$

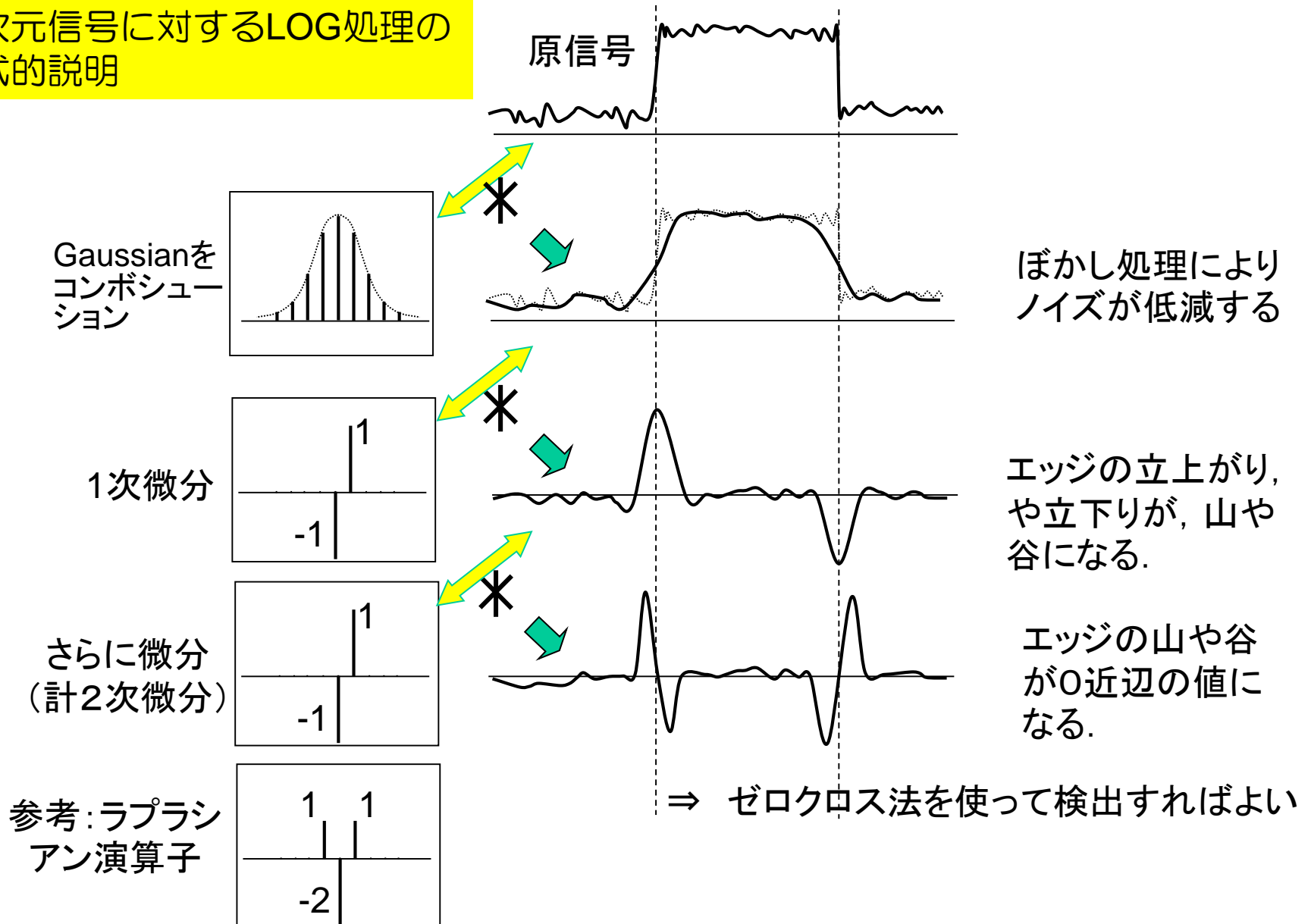
Sobel filter_x

			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

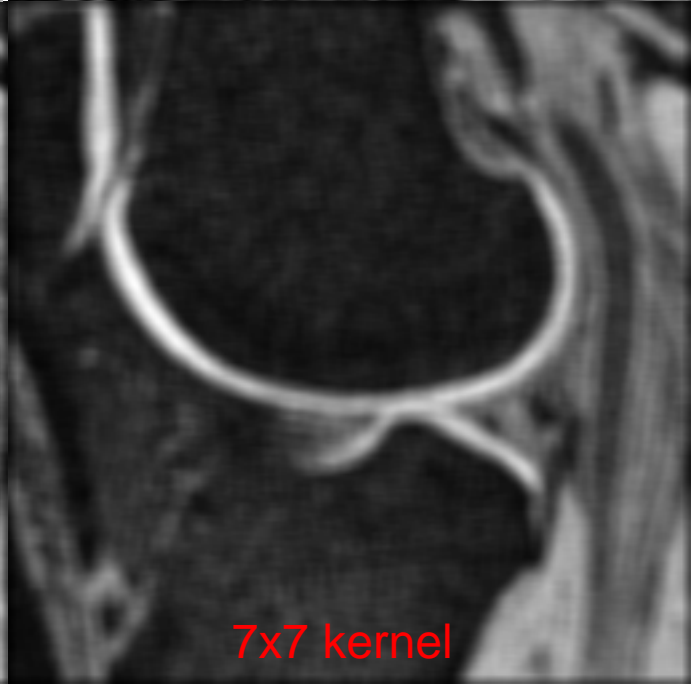
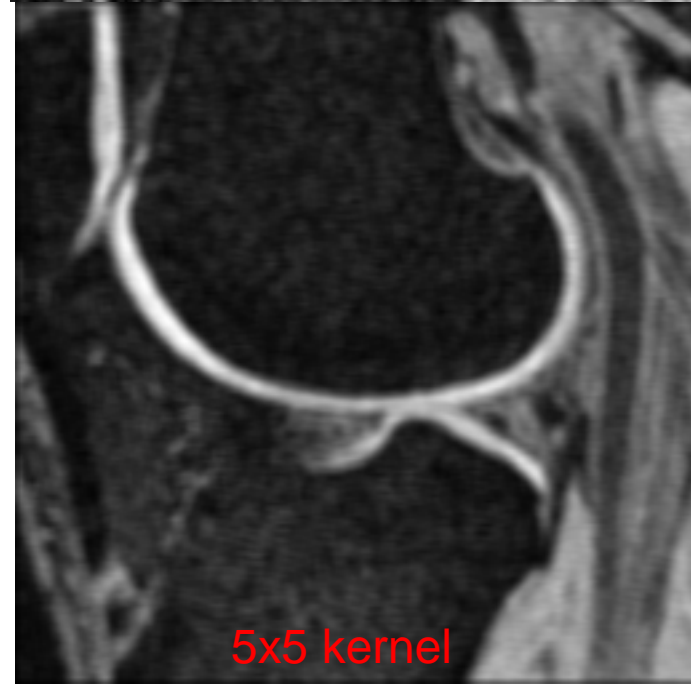
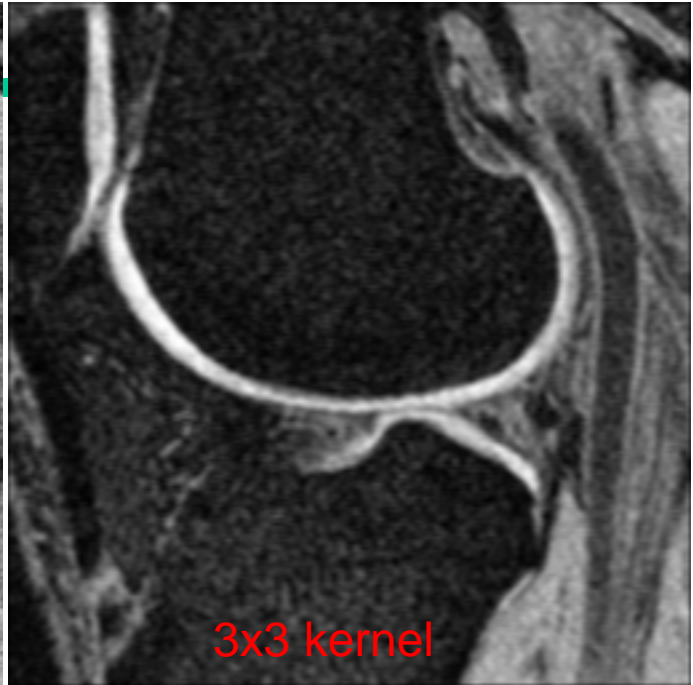
Laplacian of Gaussian (LoG) フィルタ

1次元信号に対するLOG処理の模式的説明

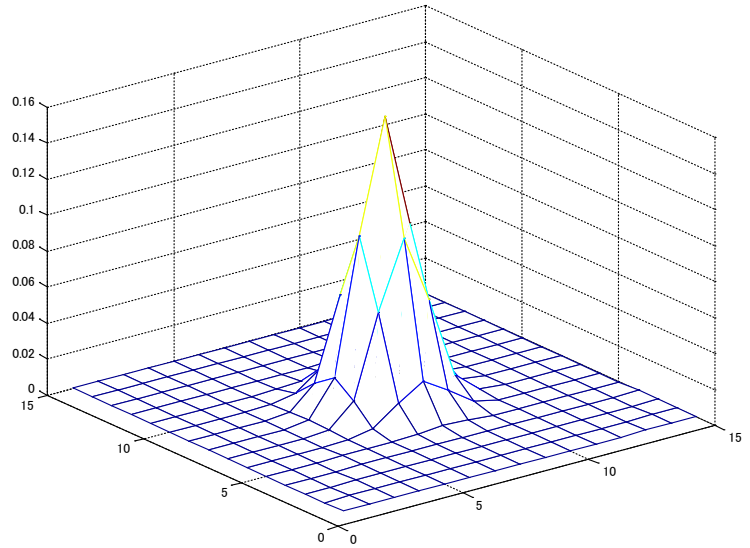


Kernel:

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

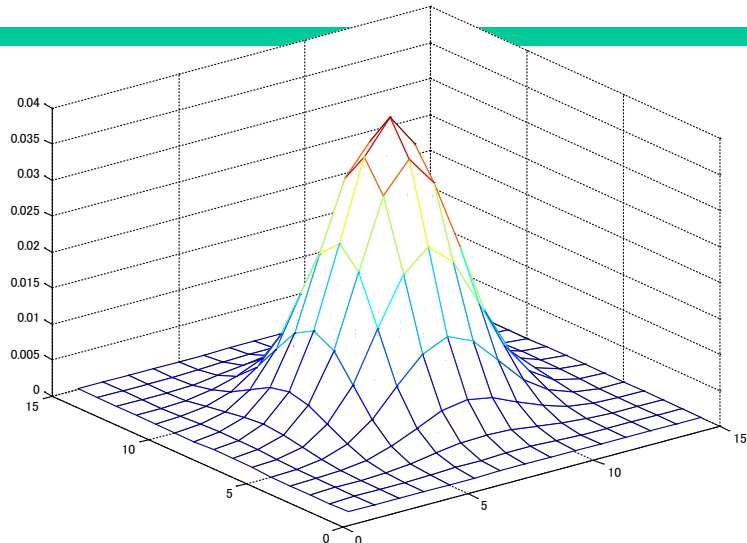


平滑化 フィルタ(Gaussian kernel)

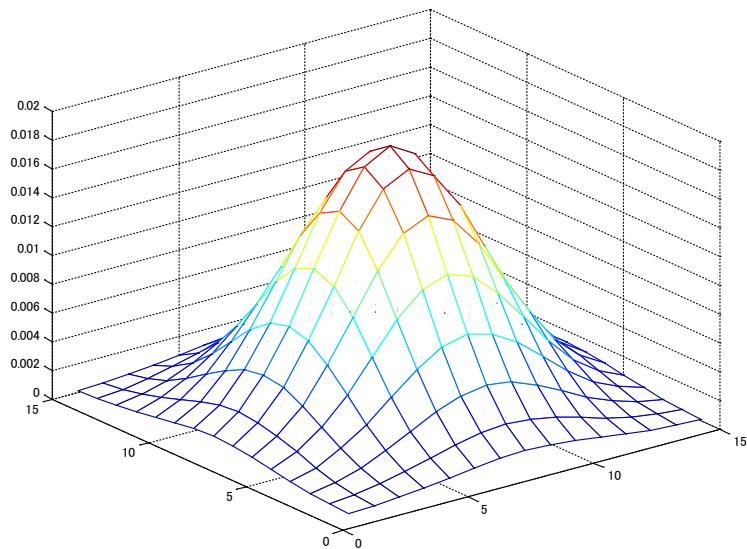


Sigma = 1

平滑化 フィルタ (Gaussian kernel)



Sigma = 2



Sigma = 3

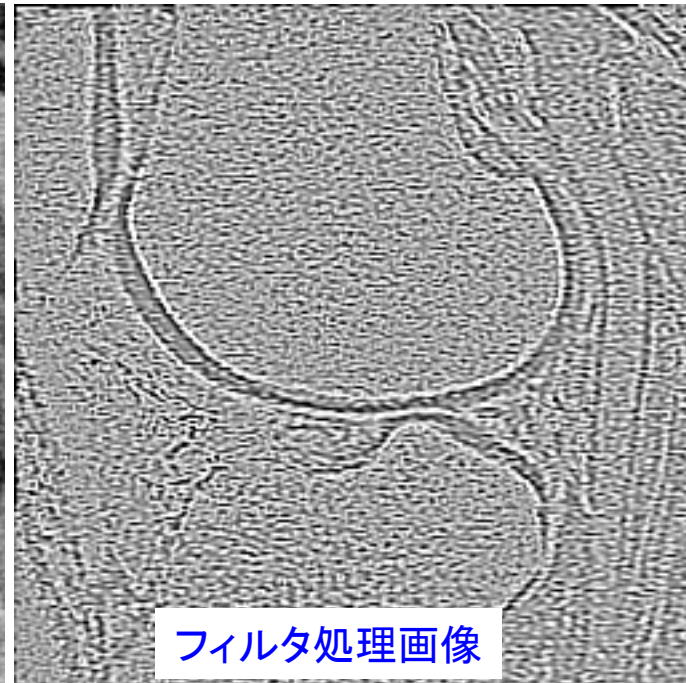


フィルタ処理画像



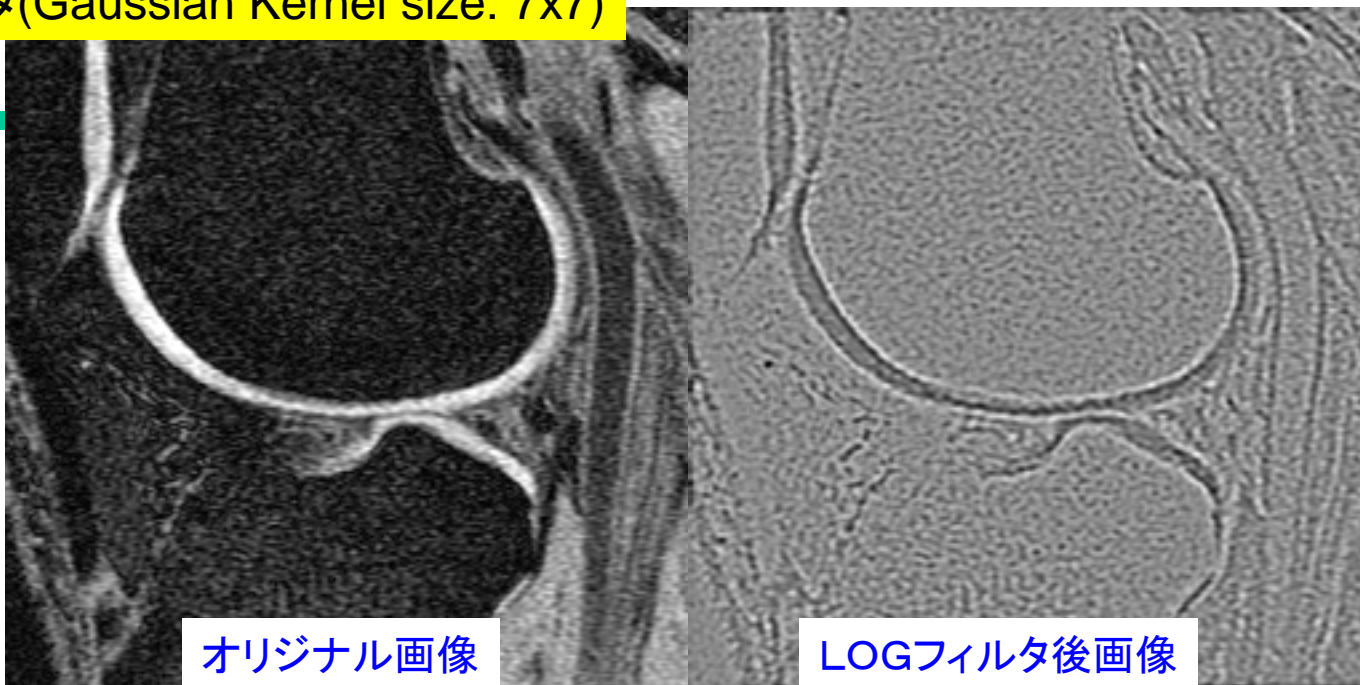
フィルタ処理画像

Laplacianフィルタ



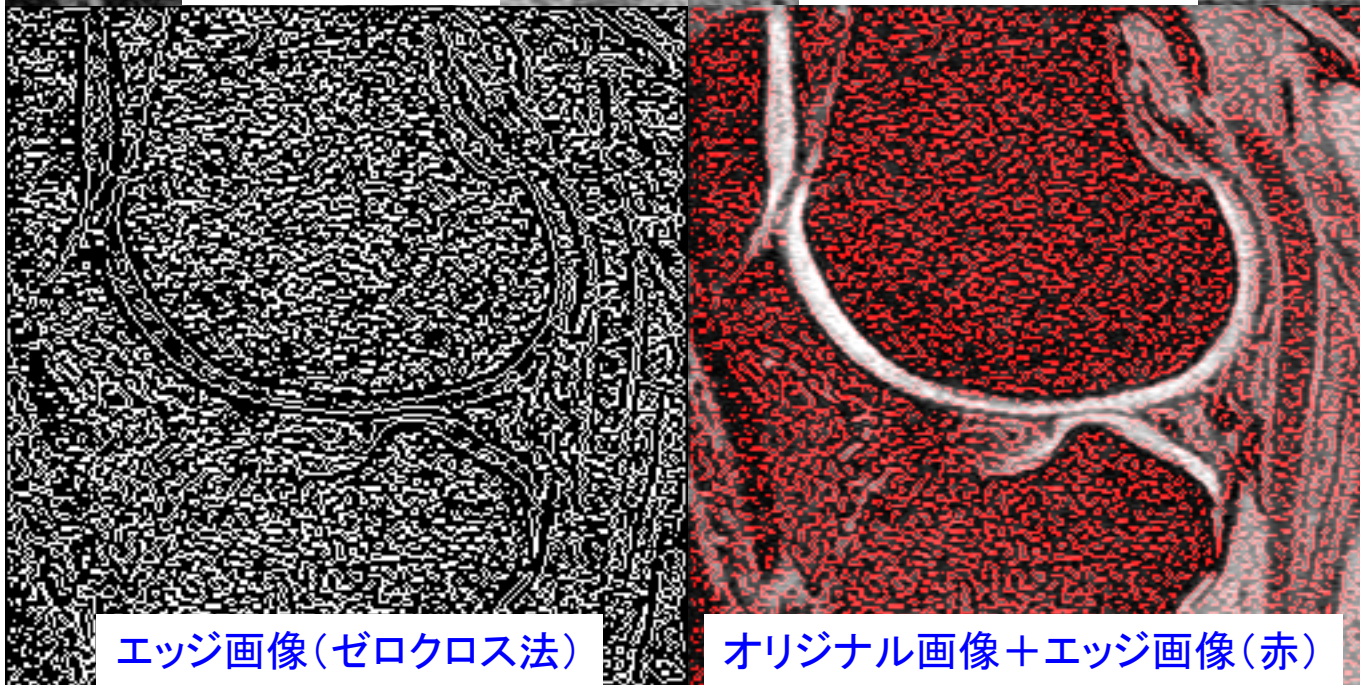
(フィルタ処理の後、負の値も発生する。
画像として表示するために、値が0から255
の範囲になるような階調変換を行っている)

LOG フィルタ(Gaussian Kernel size: 7x7)



オリジナル画像

LOGフィルタ後画像

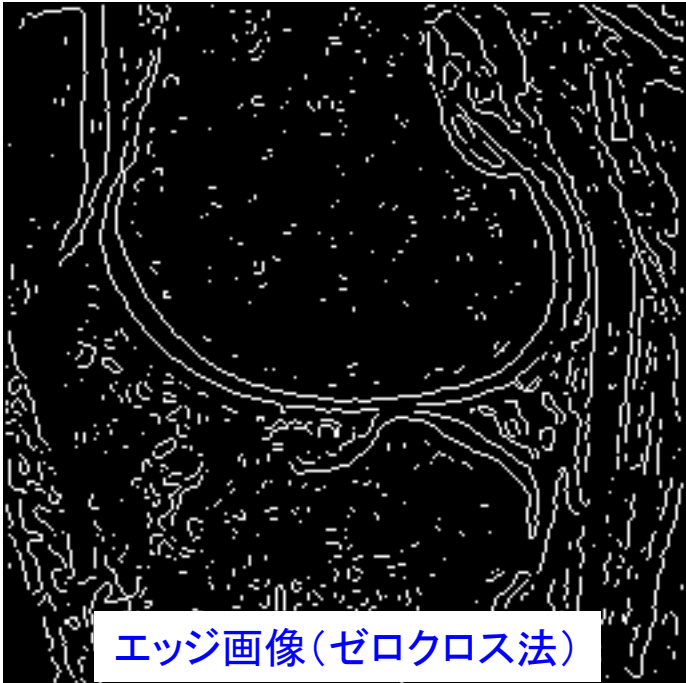
Gaussian
Kernel size:
7x7

エッジ画像(ゼロクロス法)

オリジナル画像+エッジ画像(赤)



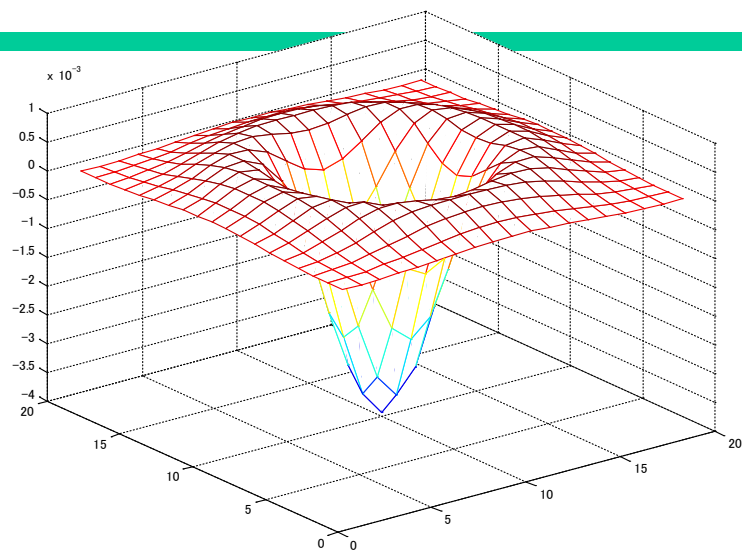
Gaussian
Kernel size:
13x13



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)



Gaussian
Kernel size:
19x19

LOGフィルタ後画像



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

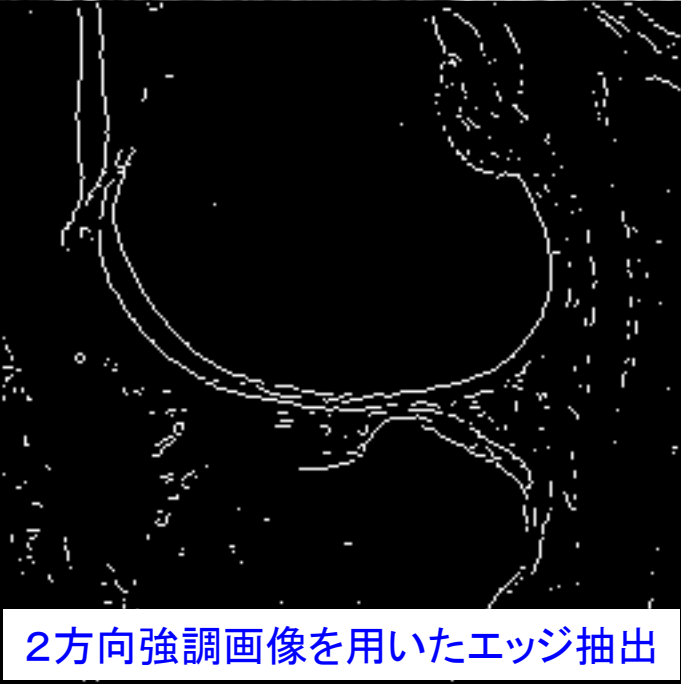
-1	-1	-1
0	0	0
1	1	1

y



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

y



2次元フーリエ変換

講義内容

- 空間周波数の概念
- 2次元フーリエ変換
- 代表的な2次元フーリエ変換対
- 2次元離散フーリエ変換

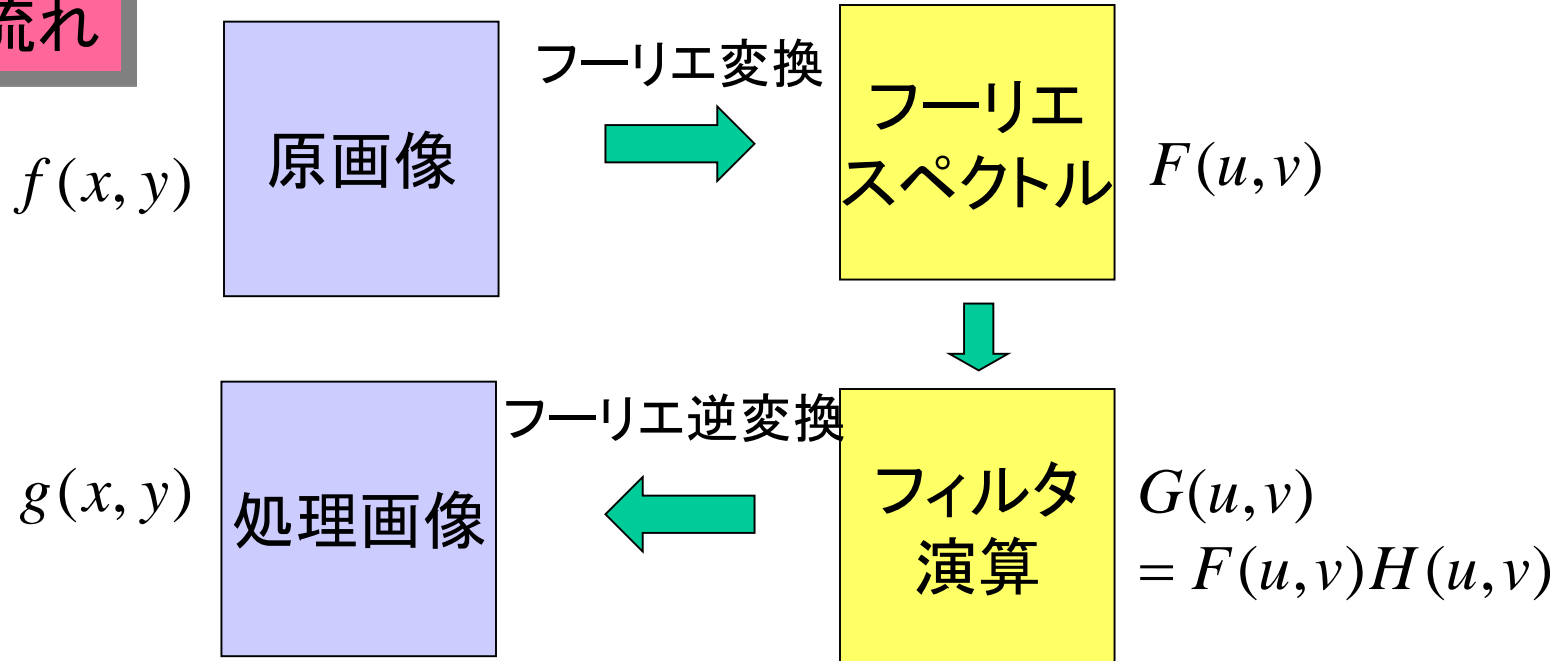
画像のフィルタリング処理

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

フーリエ面での処理

処理の流れ



特徴

周波数成分に対する**自在な**フィルタリングが可能

例

LPF, BPF, HPF, 部分的なフィルタ

(特定周波数成分の除去, 周期構造をもつノイズの除去)

Wiener フィルタ (周波数ごとのSN比を考慮した復元フィルタ)

コンボリューション定理

実空間

フーリエ空間

$$f(x, y) \longleftrightarrow F(u, v)$$

$$h(x, y) \longleftrightarrow H(u, v)$$

$$g(x, y) \longleftrightarrow G(u, v)$$

$$g(x, y) = f(x, y) * h(x, y) \longleftrightarrow G(u, v) = F(u, v) \cdot H(u, v)$$

コンボリューション

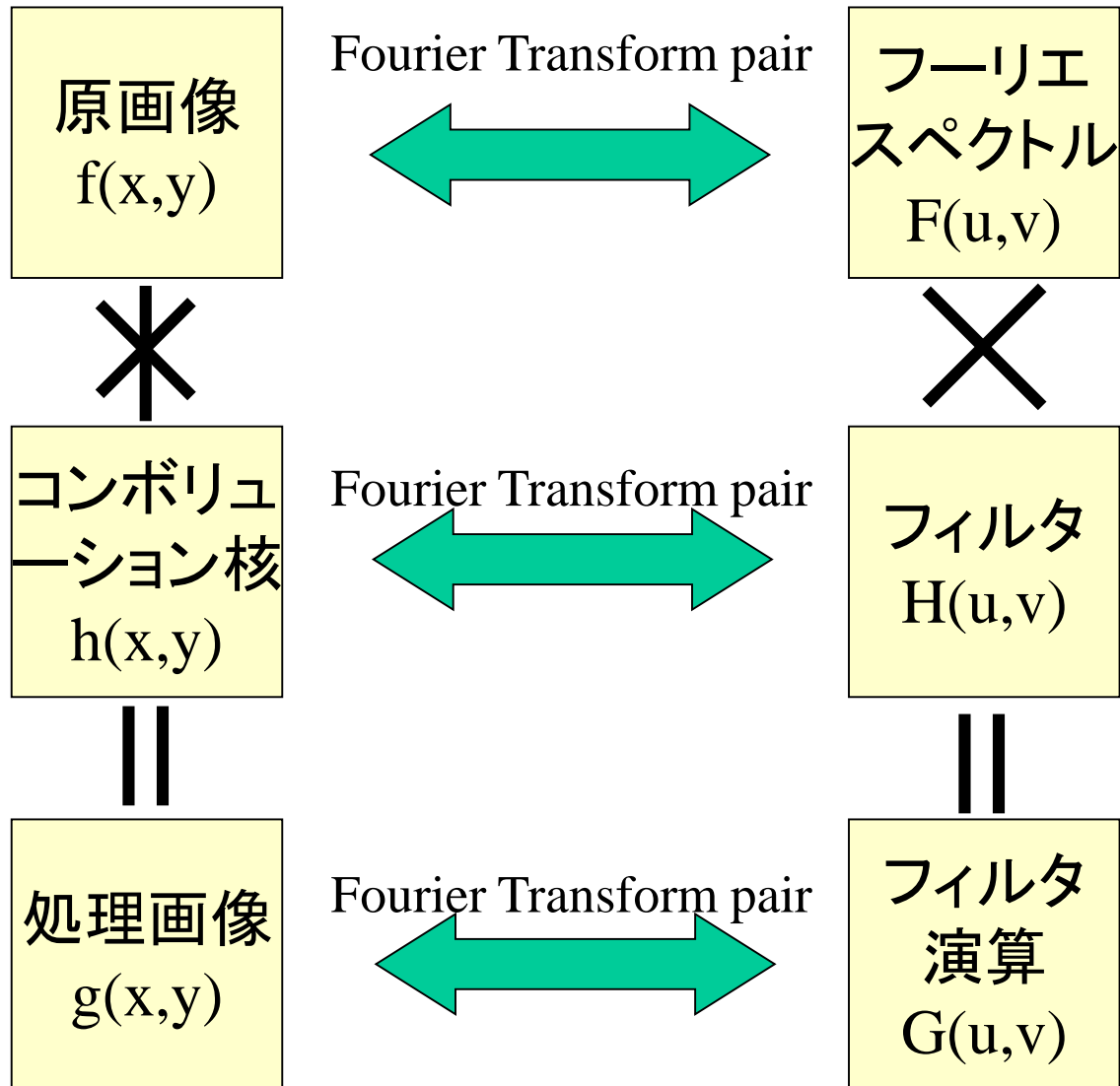
積

$$g(x, y) = f(x, y) \cdot h(x, y) \longleftrightarrow G(u, v) = F(u, v) * H(u, v)$$

積

コンボリューション

処理の等価性

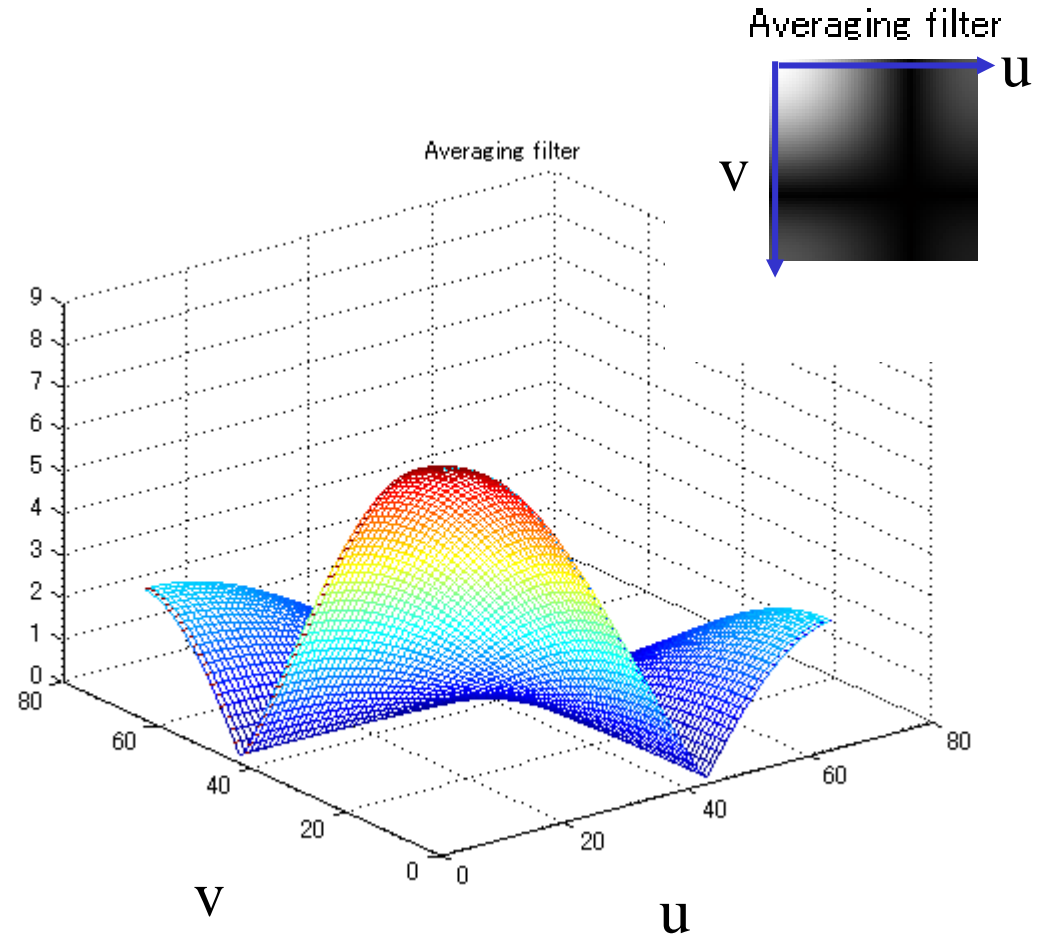
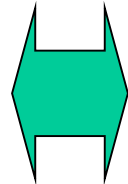


平滑化フィルタ

空間周波数フィルタ

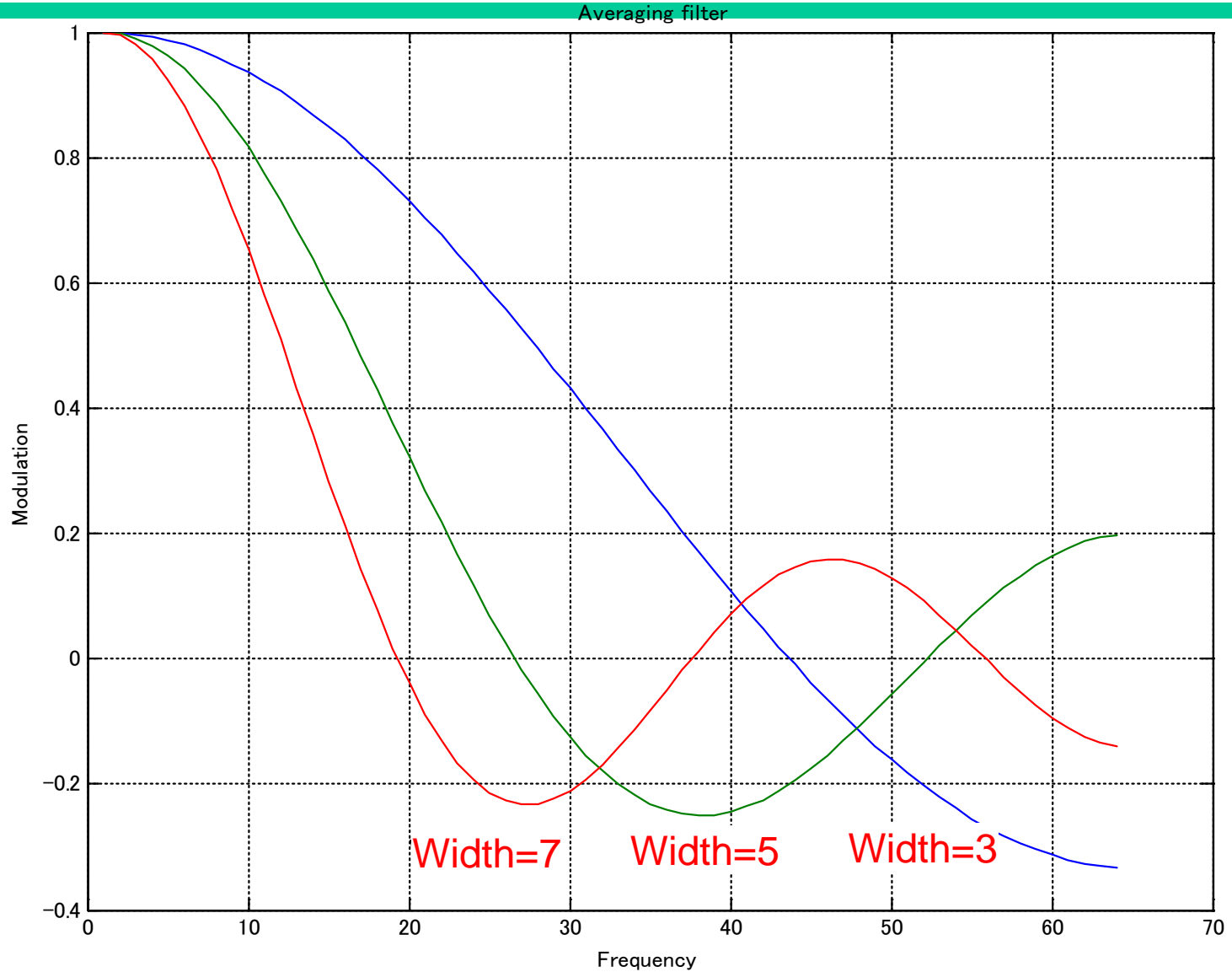
実空間でのフィルタ
(コンボリューション核)

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



(フィルタ特性の絶対値をとって表示)

平滑化フィルタの周波数特性



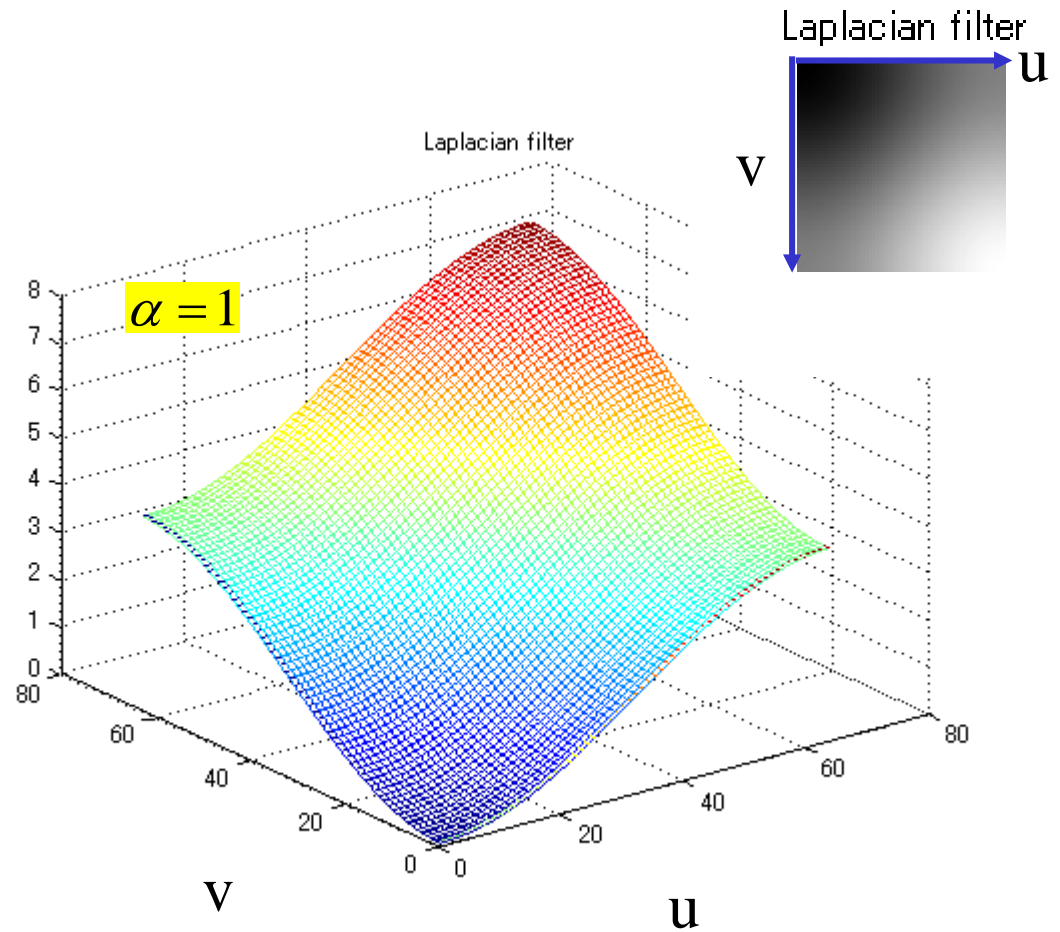
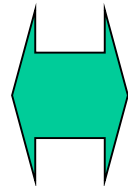
Low pass filter

Laplacianフィルタ

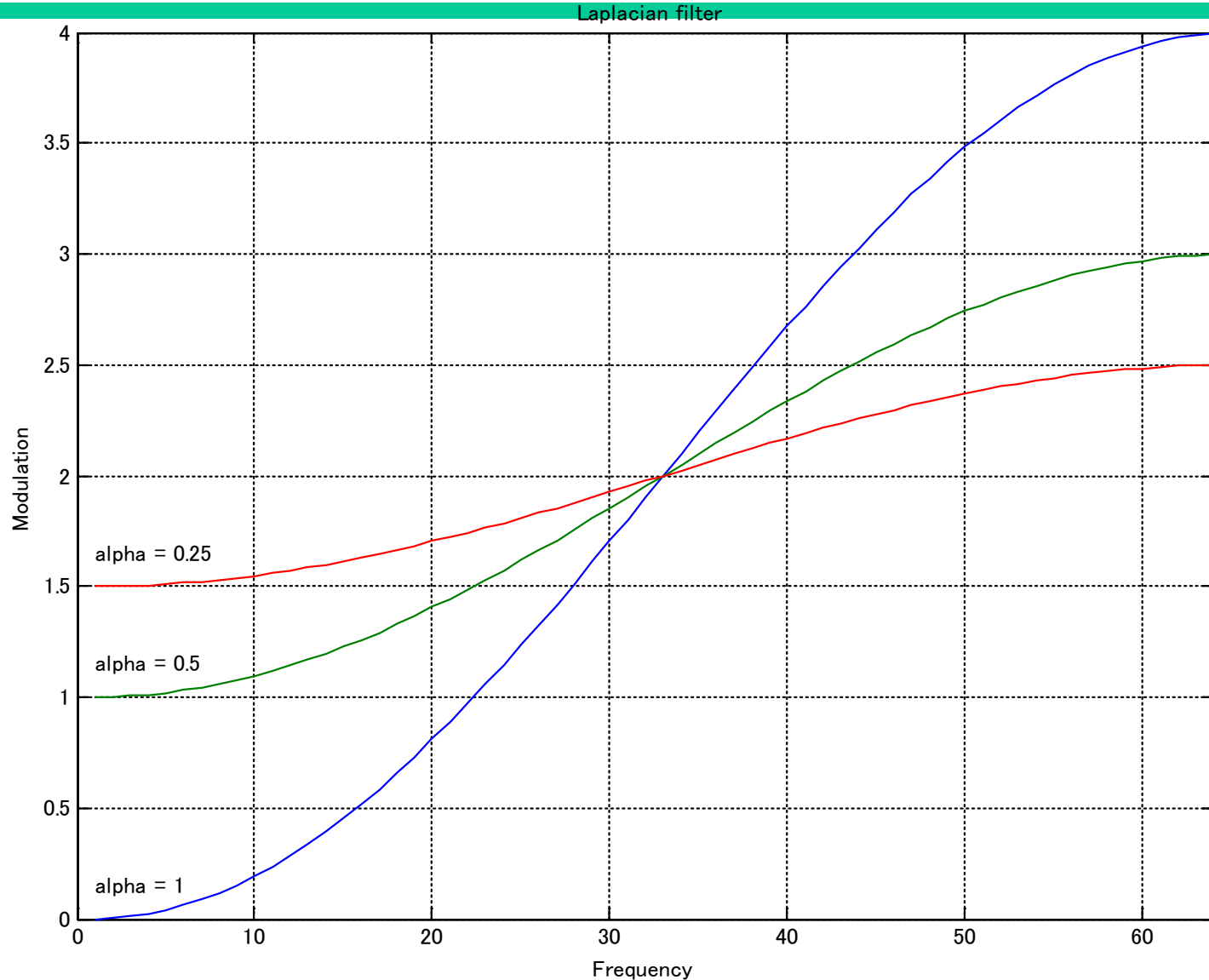
空間周波数フィルタ

実空間でのフィルタ
(コンボリューション核)

0	$-\alpha$	0
$-\alpha$	4	$-\alpha$
0	$-\alpha$	0



ラプラシアンフィルタの周波数特性

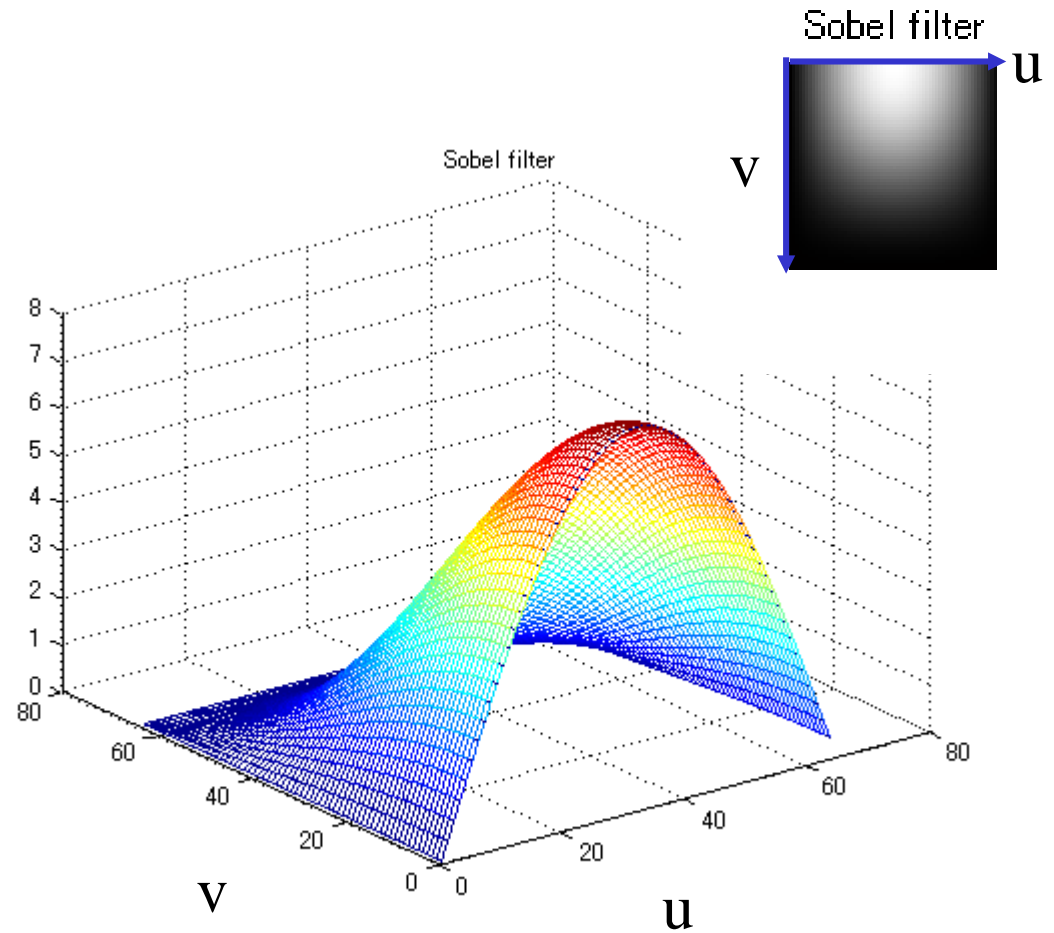
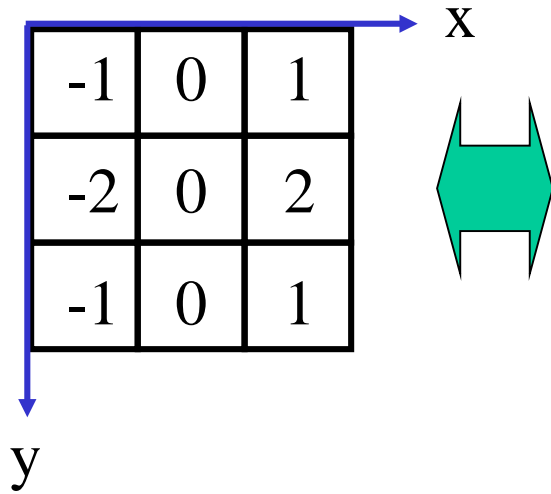


High pass filter

Sobel フィルタ

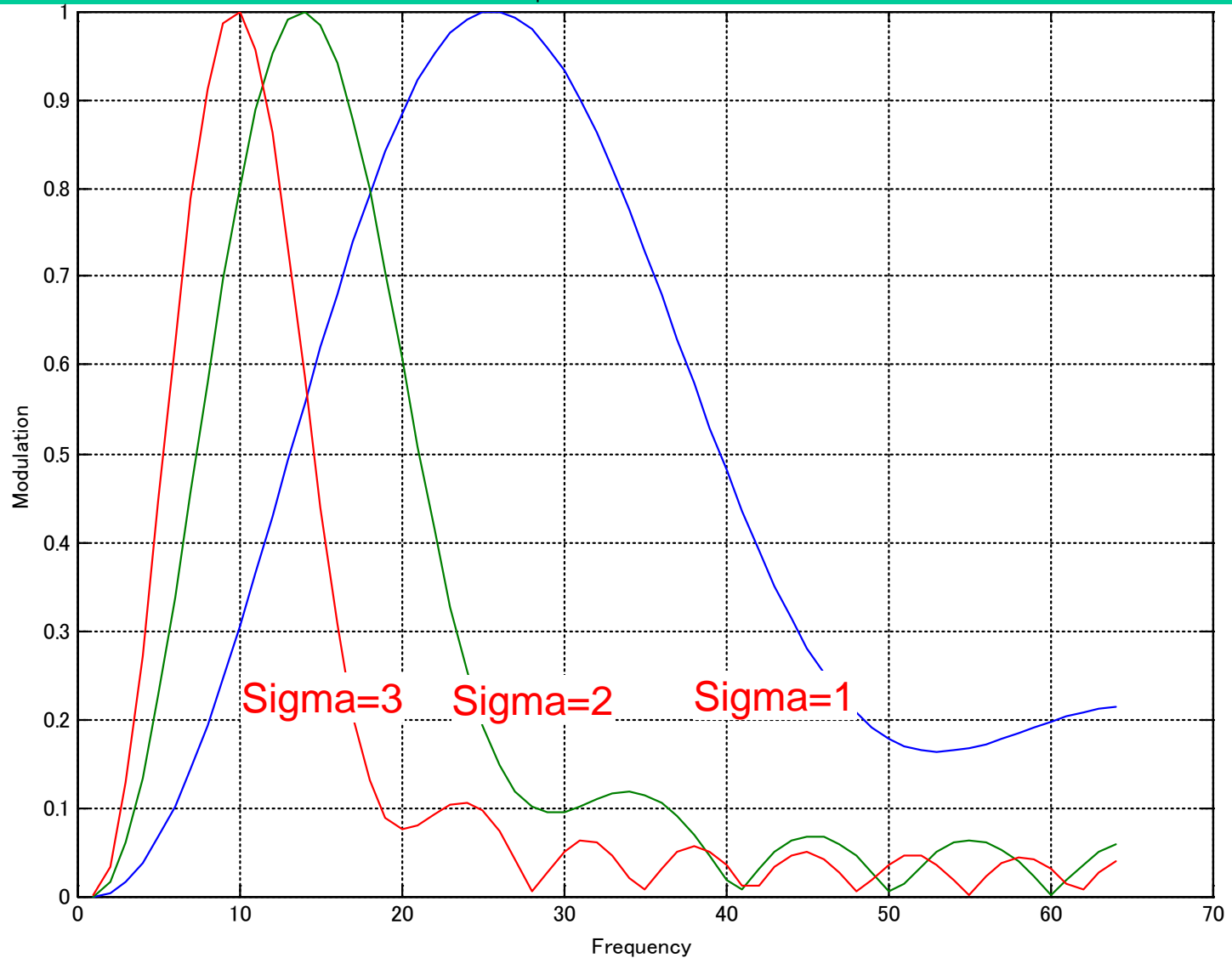
空間周波数フィルタ

実空間でのフィルタ
(コンボリューション核)



LOGフィルタの周波数特性

Laplacian of Gaussian filter

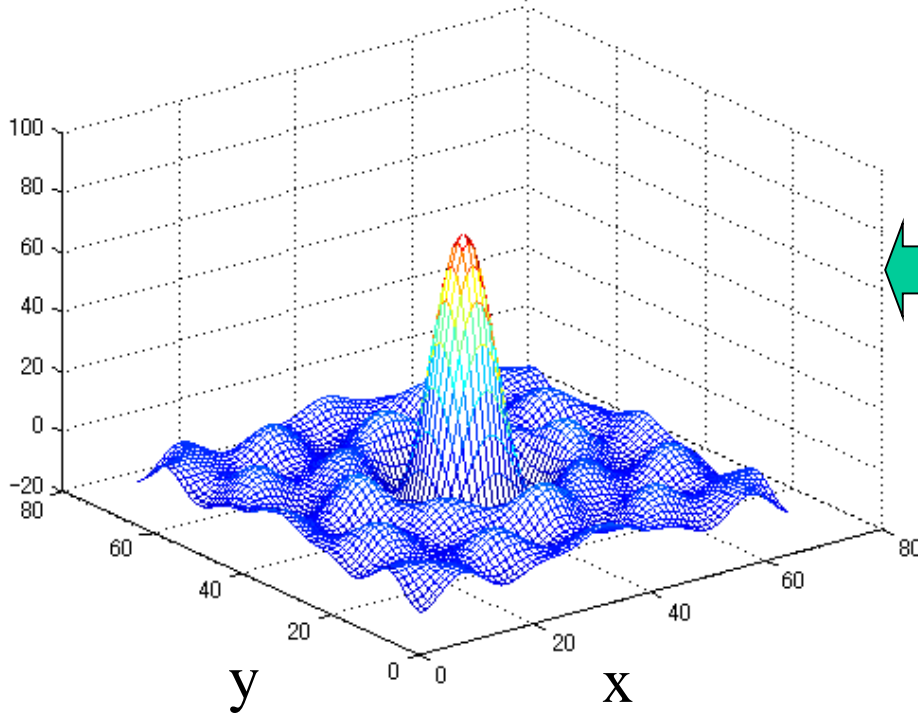


Band pass filter

空間周波数フィルタとコンボリューション核の例

実空間

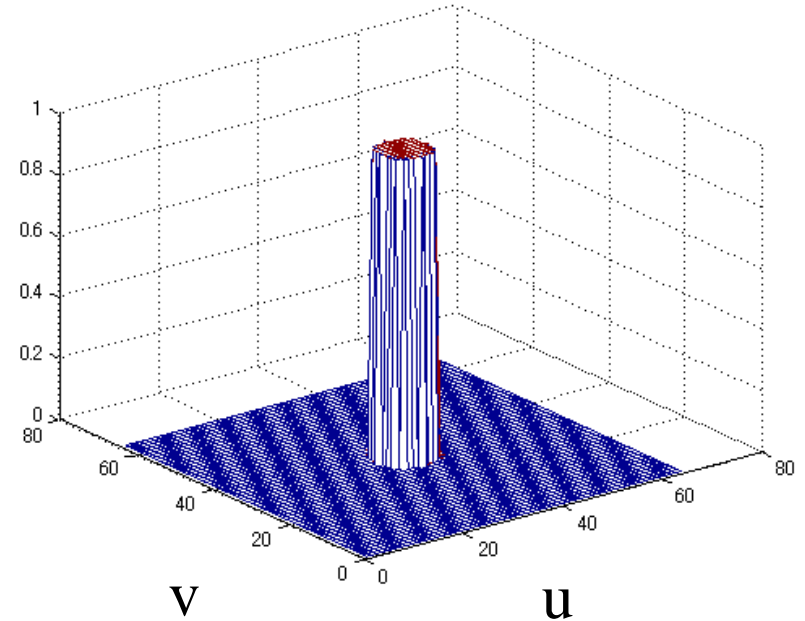
Convolution kernel in spatial domain



コンボリューション核

フーリエ空間

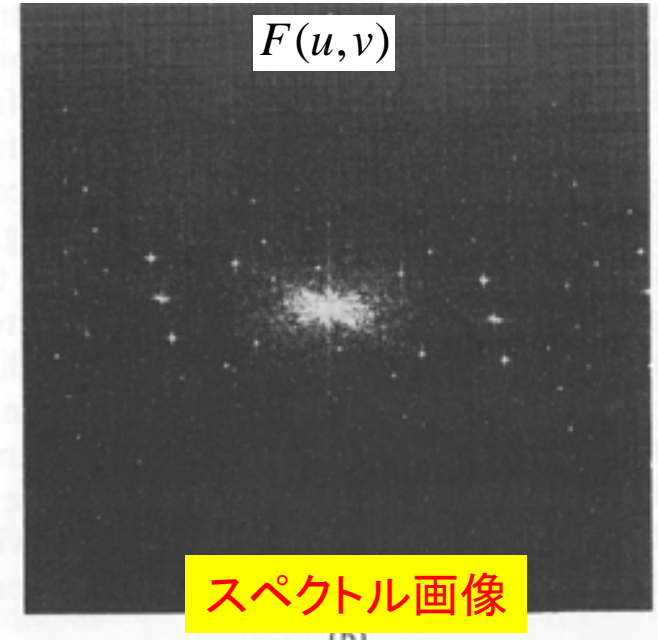
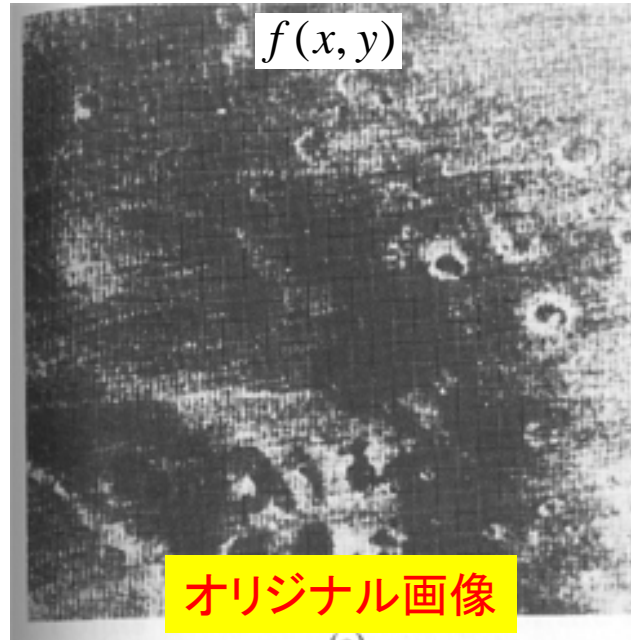
Filter in Fourier domain



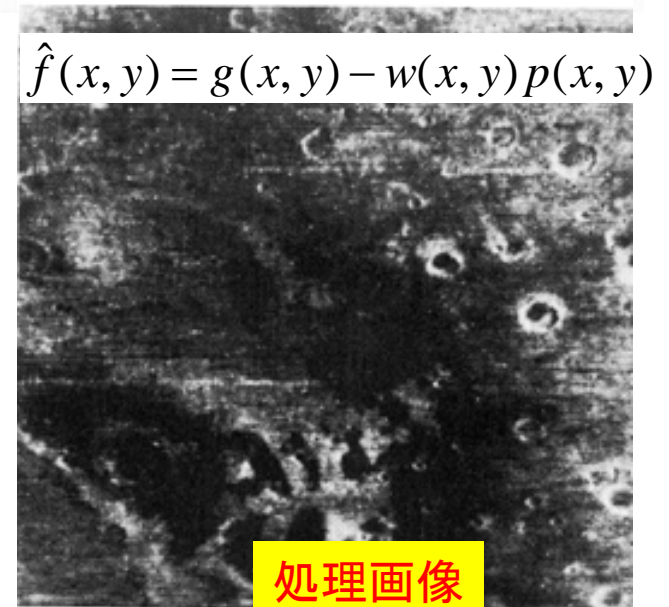
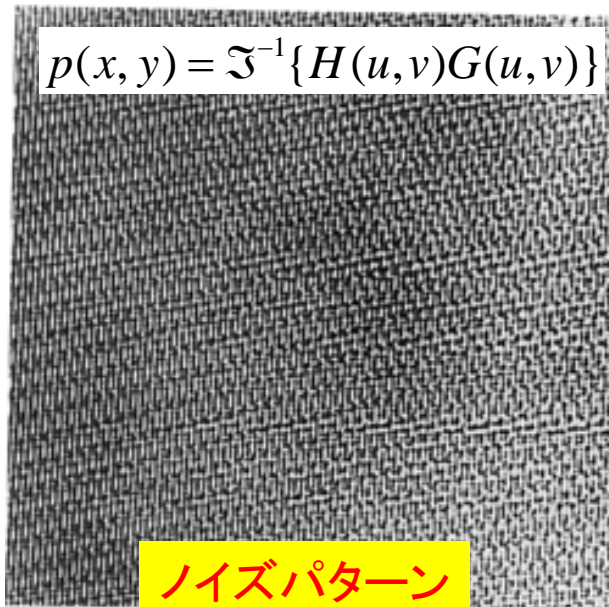
空間周波数フィルタ
Sharp-cut LPF

周期性のあるノイズの低減

周波数空間の一部にノイズのパワーが集中しているようなとき



重み $w(x, y)$ は (x, y) の近傍で推定画像の分散が最小になるように決定.



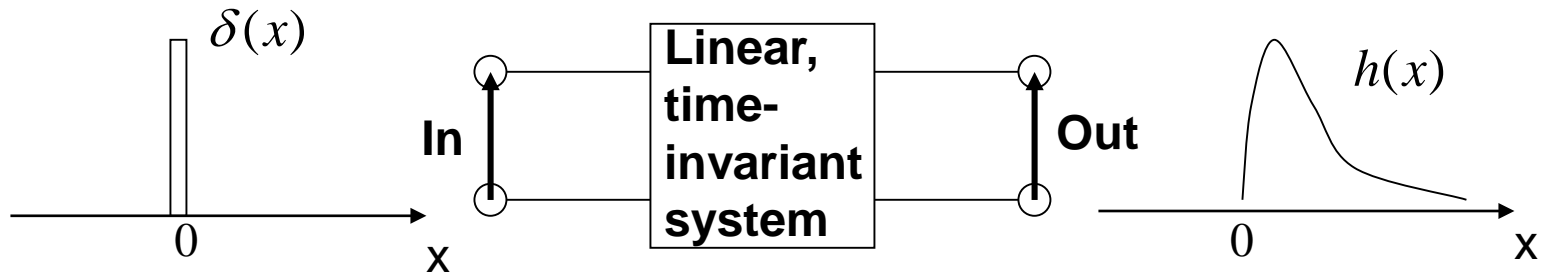
Digital Image Processing, R. C. Gonzalez and R. E. Woodsから引用

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

線形時不変システム また線形シフトインバリエントシステム

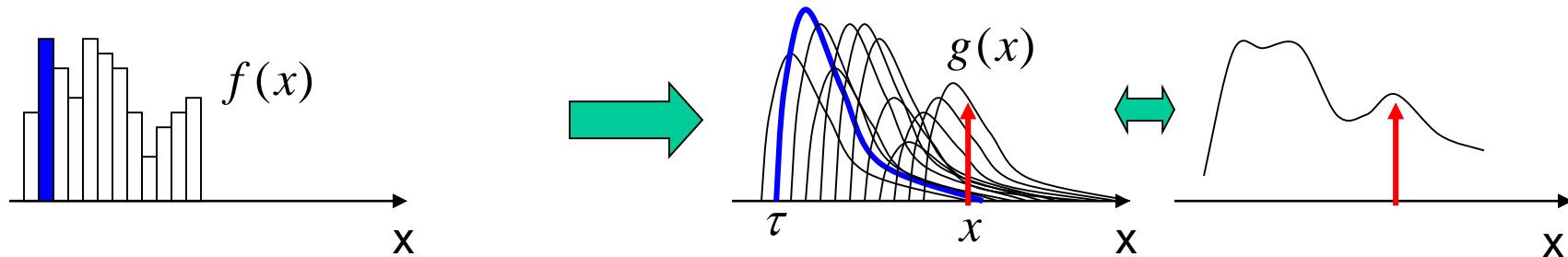


ディラックのデルタ関数
：インパルス関数

デルタ関数入力に対する応答：
インパルス応答

入力信号

出力信号



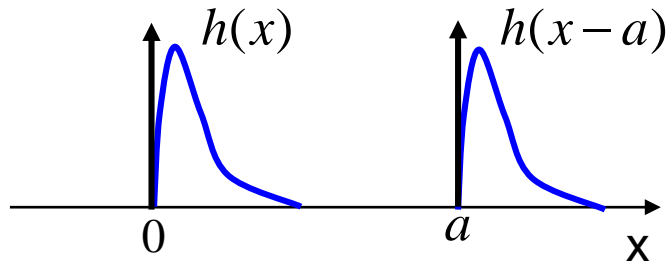
出力信号は入力信号と
インパルス応答との
コンボリューションで
表される。

$$g(x) = \int_{-\infty}^{\infty} h(x-\tau)f(\tau)d\tau$$

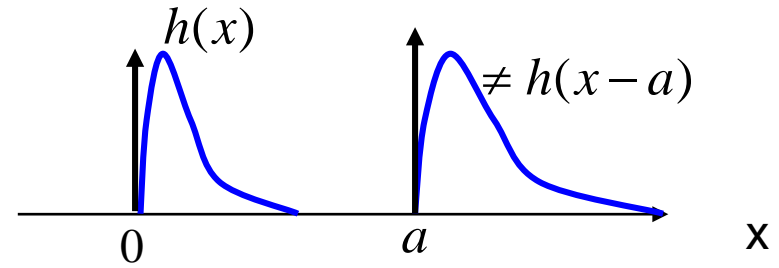
$$= h(x) * f(x)$$

シフトインバリエントシステム

シフトインバリエント：インパルス応答が，シフトによらないこと。

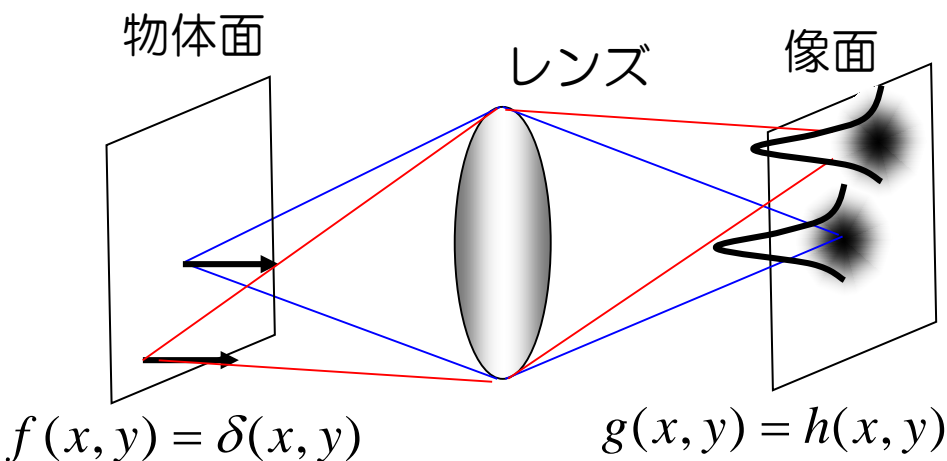


シフトインバリエント

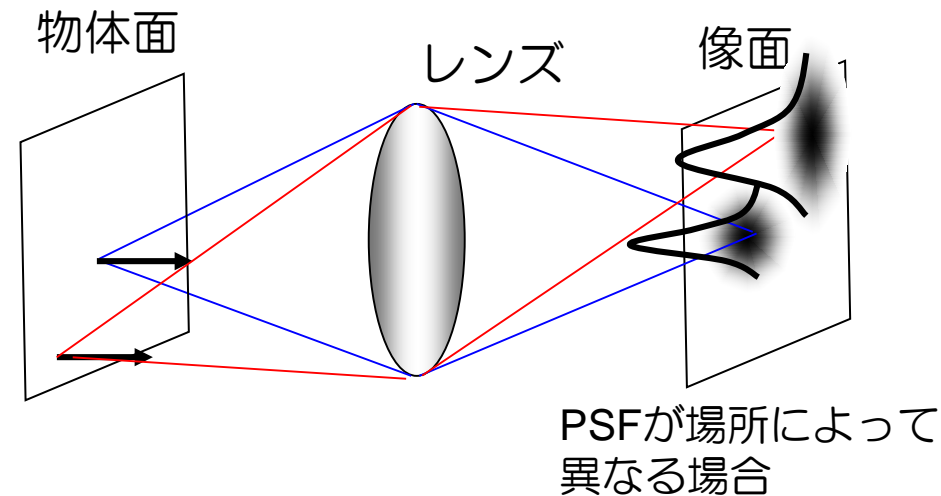


シフトバリエント

2次元（画像）の場合 インパルス応答＝点光源に対するレンズによる像
（点像分布関数 point spread function とよぶ）



シフトインバリエント



シフトバリエント

PSFが場所によって異なる場合

線形システム

線形システム：重ね合わせの原理が成り立つこと

入力 $f(x)$ に対して、 $g(x)$ を出力するシステムを以下のように定義する。

$$g(x) = S\{f(x)\}$$

このシステムが線形であるとは、以下の関係が成り立つことである。

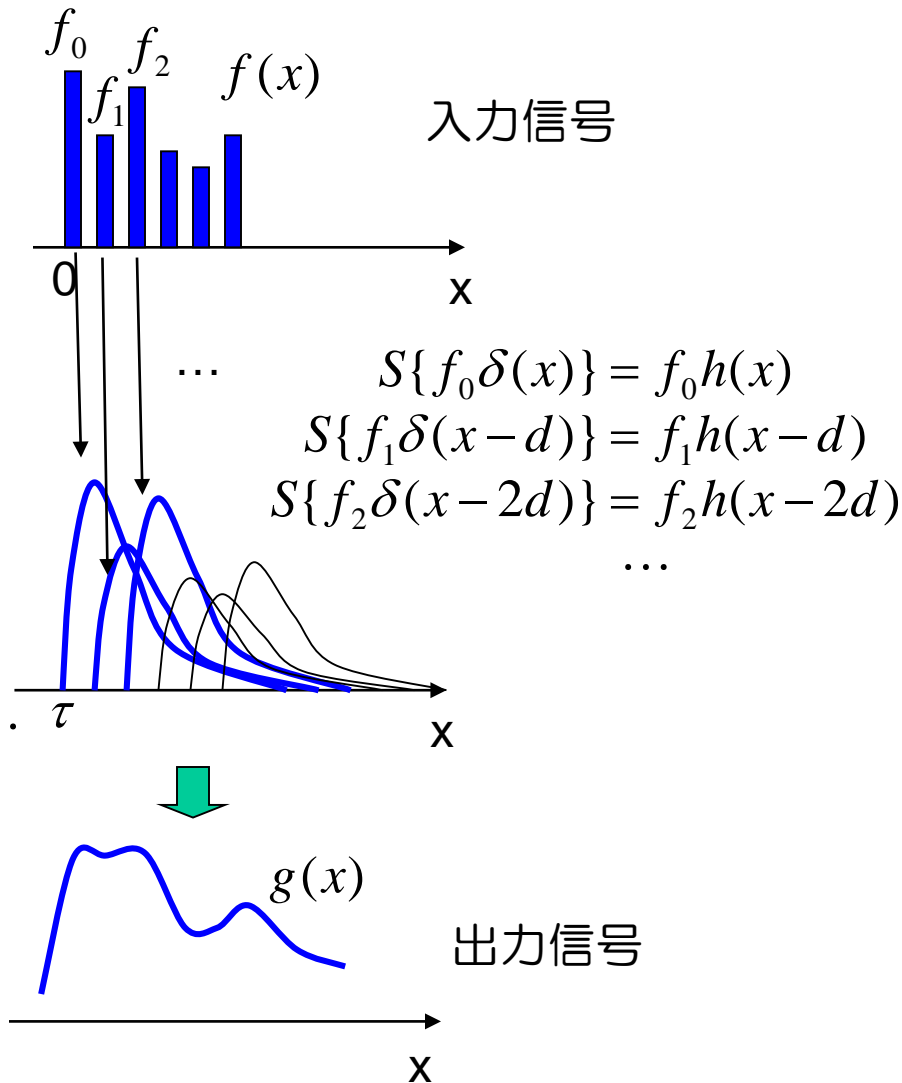
$$\begin{aligned} g(x) &= S\{a_1 f_1(x) + a_2 f_2(x)\} \\ &= a_1 S\{f_1(x)\} + a_2 S\{f_2(x)\} \end{aligned}$$

入力関数：

$$f(x) = f_0 \delta(x) + f_1 \delta(x-d) + f_2 \delta(x-2d) + \dots$$

出力関数：

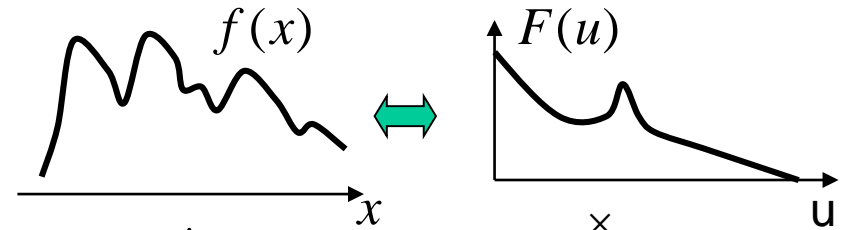
$$g(x) = f_0 h(x) + f_1 h(x-d) + f_2 h(x-2d) + \dots$$



周波数空間で考える

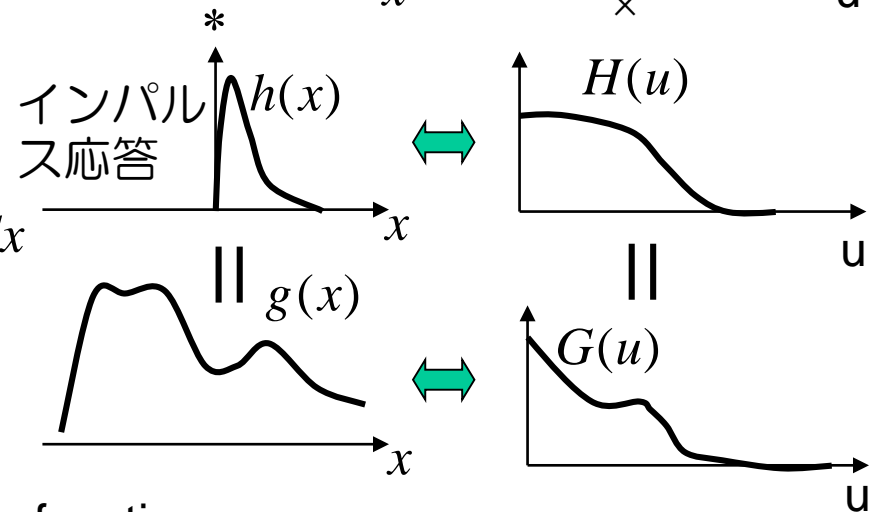
入力信号のスペクトル：

$$F(u) = \int_{-\infty}^{\infty} f(x) \exp(-j2\pi ux) dx$$



出力信号のスペクトル：

$$\begin{aligned} G(u) &= \int_{-\infty}^{\infty} g(x) \exp(-j2\pi ux) dx \\ &= \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} h(x-\tau) f(\tau) d\tau \right] \exp(-j2\pi ux) dx \\ &= H(u) F(u) \end{aligned}$$



$$H(u) = \frac{G(u)}{F(u)} = \frac{\text{output}}{\text{Input}} \quad \text{： 伝達関数 Transfer function}$$

実空間

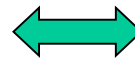
$$\begin{aligned} g(x) &= \int_{-\infty}^{\infty} h(x-\tau) f(\tau) d\tau \\ &= h(x) * f(x) \end{aligned}$$

コンボリューション

フーリエ空間

$$G(u) = H(u) F(u)$$

掛け算



Wiener Filter

劣化画像の復元などに用いられる

理想画像: $f(x, y)$

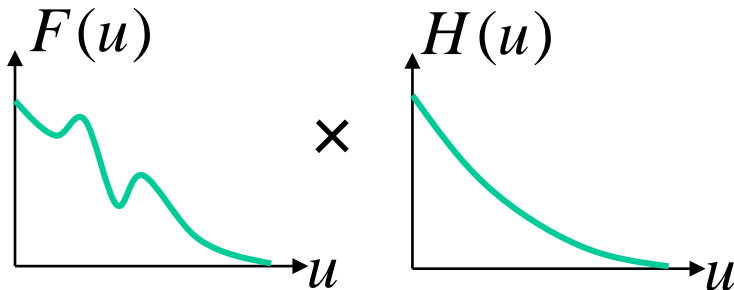
劣化の点像分布関数: $h(x, y)$

劣化画像:

$$g(x, y) = f(x, y) * h(x, y) + n(x, y)$$

フーリエ変換は:

$$G(u, v) = F(u, v)H(u, v) + N(u, v)$$

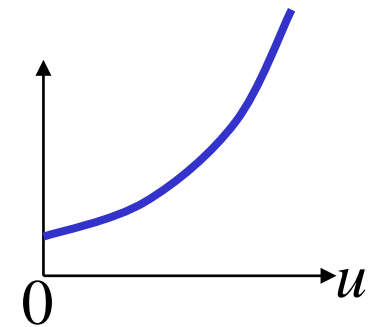


左辺(取得データ)GをHで割ると、

$$\begin{aligned} & G(u, v) / H(u, v) \\ &= F(u, v) + N(u, v) / H(u, v) \end{aligned}$$

Inverse filter:

$$\frac{1}{H(u, v)}$$



Wiener filter:

$$\frac{1}{H(u, v) + \frac{P_N(u, v)}{P_S(u, v)}}$$

ノイズパワー 信号パワー

