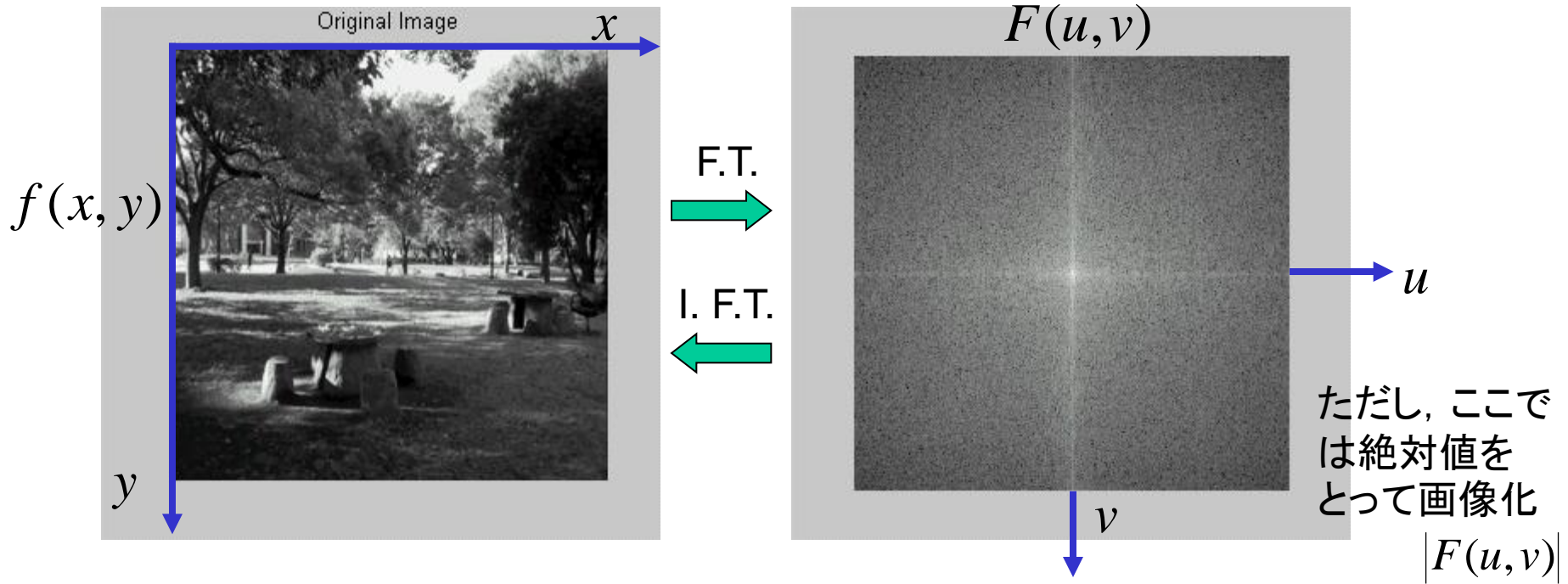


2次元フーリエ変換

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

フーリエ変換と逆変換



連続系

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \exp\{-j2\pi(ux + vy)\} dx dy$$

離散系

順変換

$$F(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \exp\{-j2\pi(ux + vy) / N\}$$

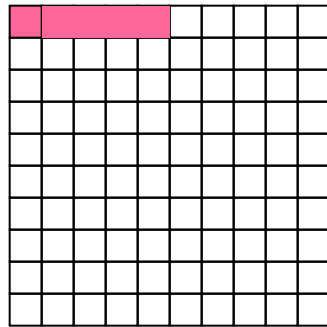
逆変換

$$f(x, y) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} F(u, v) \exp\{j2\pi(ux + vy) / N\}$$

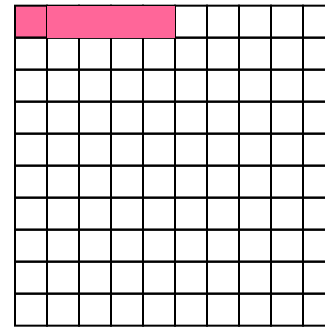
2次元フーリエ変換の具体的なイメージ

離散系での説明

$$F(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \exp\{-j2\pi(ux + vy) / N\}$$



$f(x, y)$



$\exp\{-j2\pi(ux + vy) / N\}$

対応する画素ごとに積をとって
最後に総和をとる。

それでは $\exp\{-j2\pi(ux + vy) / N\}$ はどんなパターンか？

2次元フーリエ変換の具体的なイメージ

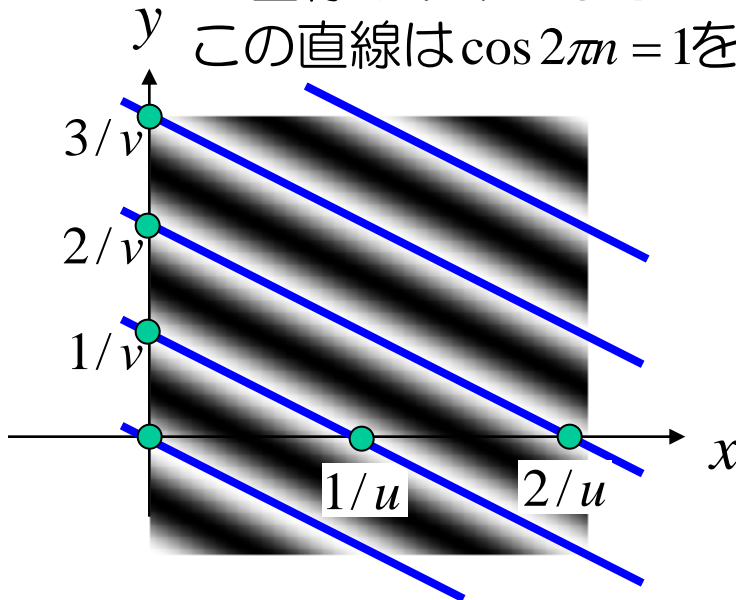
$$\exp\{-j2\pi(ux + vy)\} = \cos 2\pi(ux + vy) - j \sin 2\pi(ux + vy)$$

のうち，実部 $\cos 2\pi(ux + vy)$ に注目して考える。

$$ux + vy = 0, 1, 2, \dots, n, \dots$$

の直線は以下のようなになる。

この直線は $\cos 2\pi n = 1$ を与える。



(u, v) は空間的な波の周波数を与える。

⇒ 『空間周波数』 と呼ばれる。

u : x 方向の周波数成分

v : y 方向の周波数成分

$$ux + vy = 0, 1, 2, \dots, n, \dots$$

において， $y = 0$ とおくと

(すなわち x 軸上に注目すると)，

$$ux = 0, 1, 2, \dots \Leftrightarrow x = 0, 1/u, 2/u, \dots$$

で $\cos(ux) = 1$ となる。

「 u が小さい」 \Leftrightarrow 「間隔が大きい」

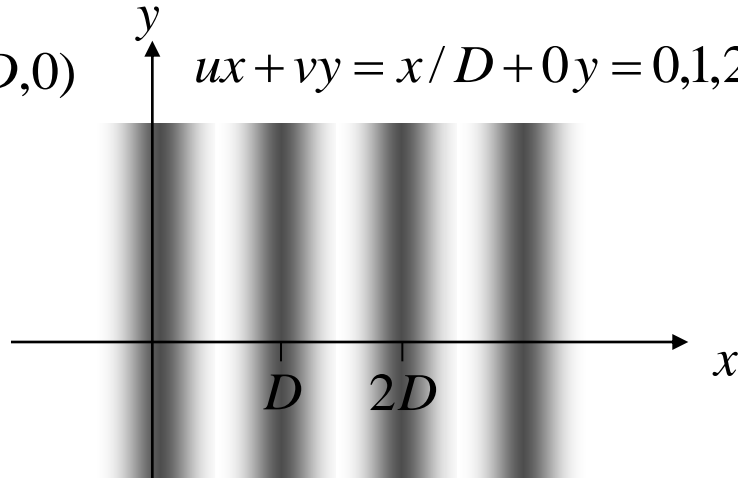
空間周波数の例

$$\cos 2\pi(ux + vy)$$

例 1)

$$(u, v) = (1/D, 0)$$

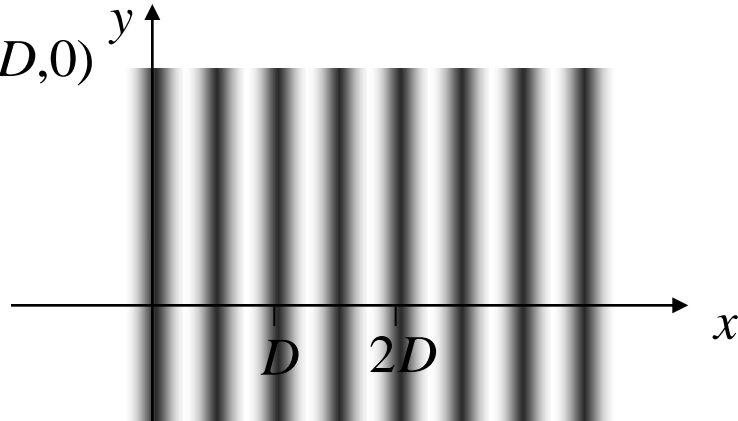
$$ux + vy = x/D + 0y = 0, 1, 2, \dots \Leftrightarrow x = 0, D, 2D, \dots$$



例 2)

$$(u, v) = (2/D, 0)$$

$$ux + vy = 2x/D + 0y = 0, 1, 2, \dots \Leftrightarrow x = 0, D/2, D, 3D/2, \dots$$

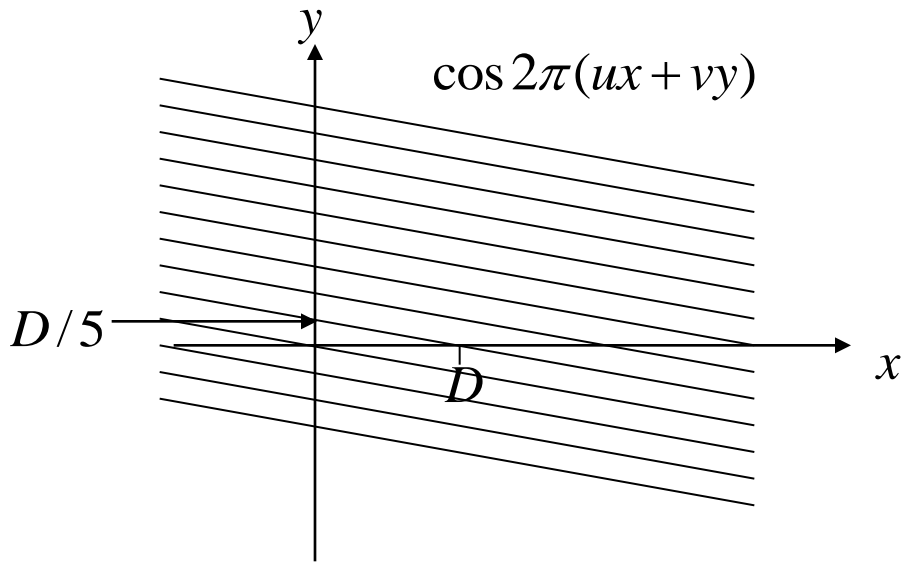


演習

例題 1

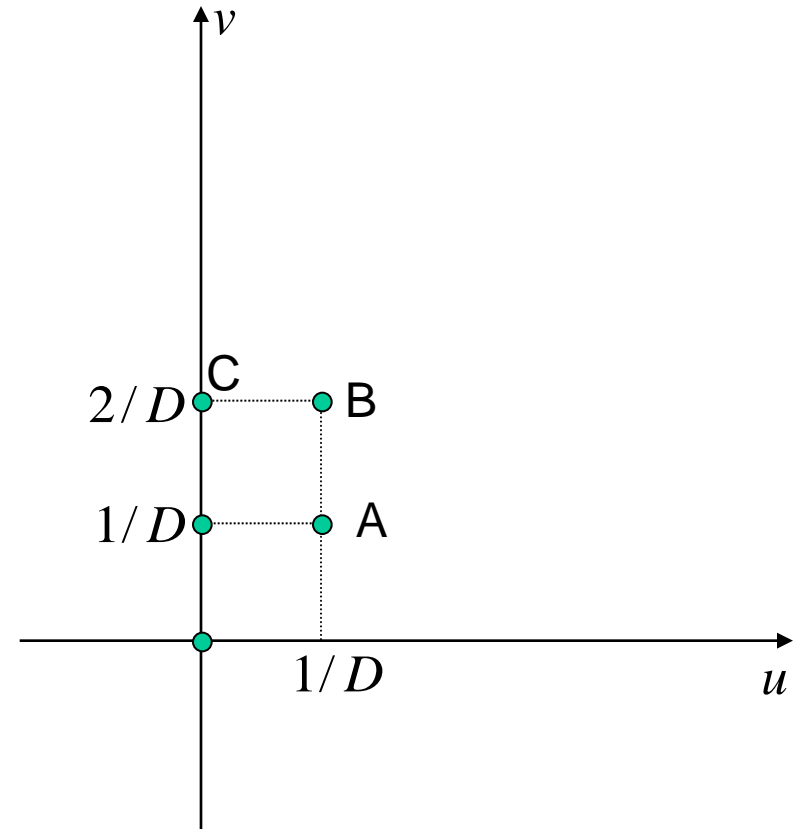
下の図に対応する余弦関数を式で書きなさい。ただし黒い線は1の値をもち、余弦関数の最大値を描いているものとする。

また、その空間周波数の位置を uv 平面上に図示しなさい。

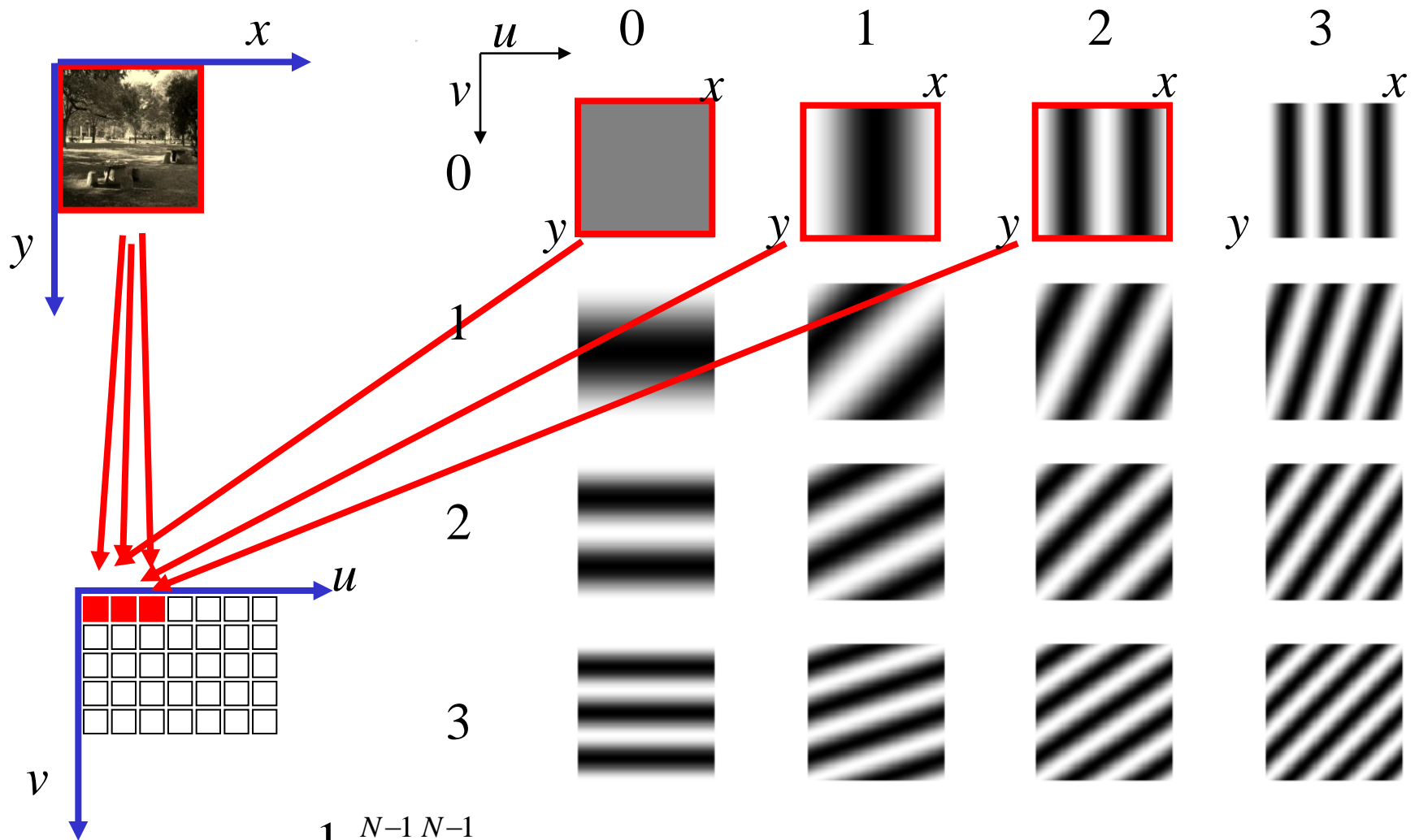


例題 2

下図のA,B,Cの位置に対応する空間周波数のパターン（ x y 面での余弦波のパターン）をスケッチしなさい。

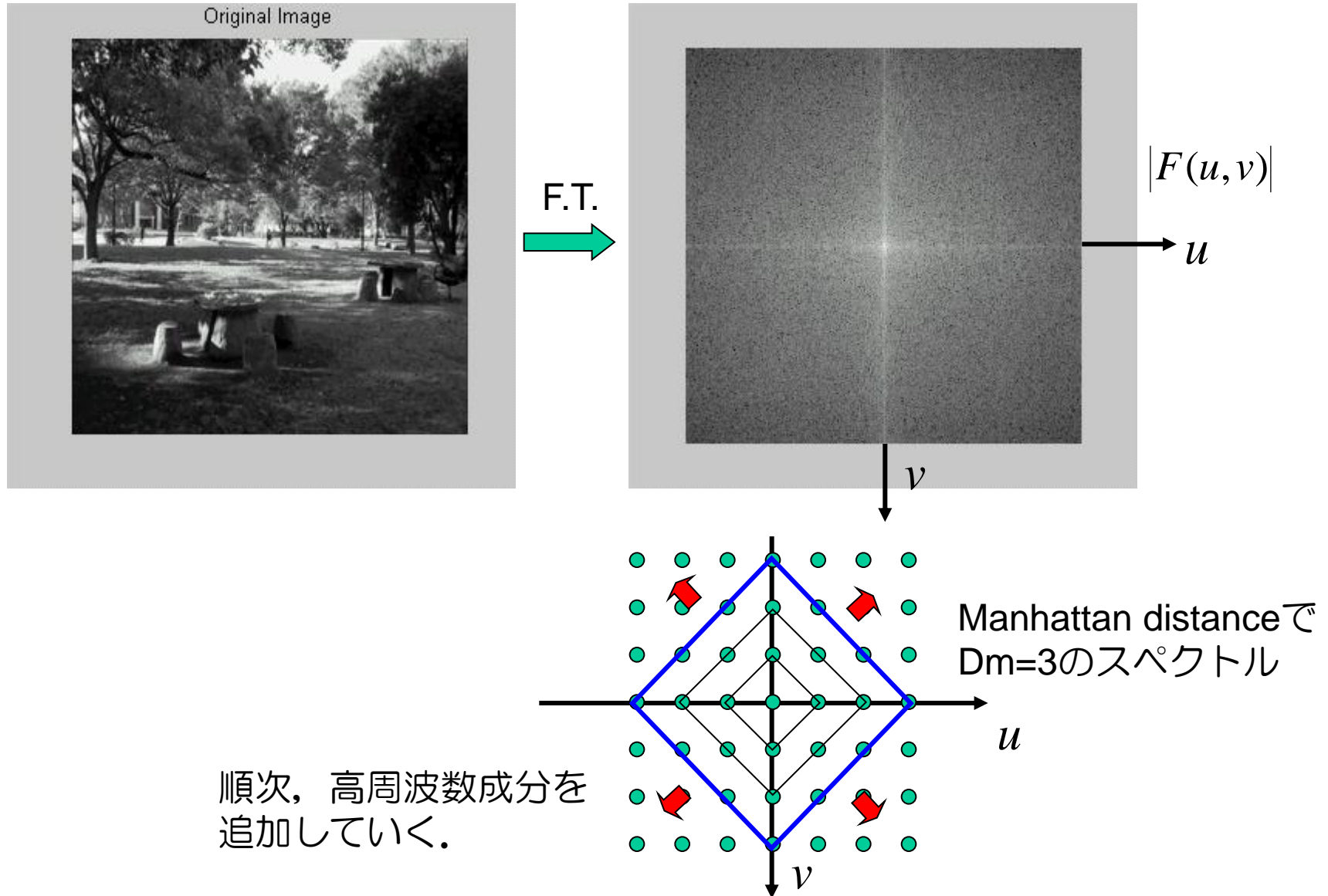


フーリエ変換演算のまとめ

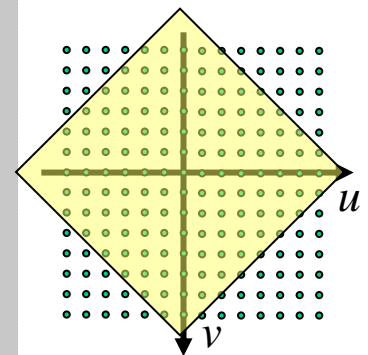
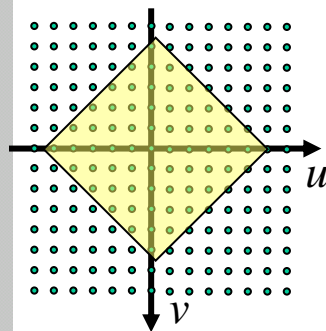
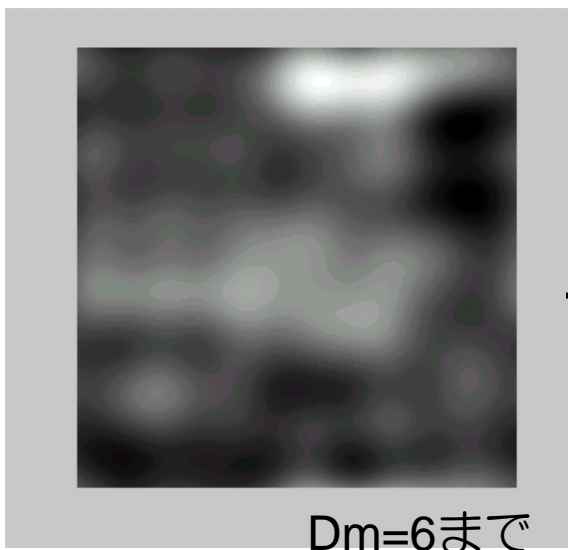
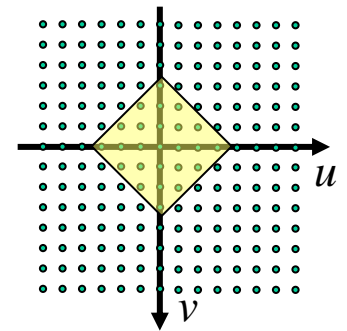
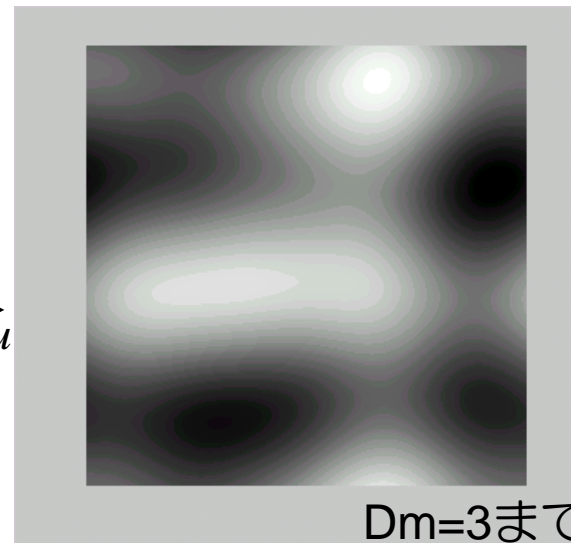
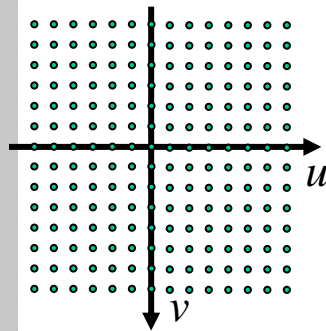


$$F(u, v) = \frac{1}{N} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(x, y) \exp\{-j2\pi(ux + vy) / N\}$$

フーリエの合成のデモ



フーリエの合成のデモ (つづき)



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

代表的な2次元フーリエ変換対(1)

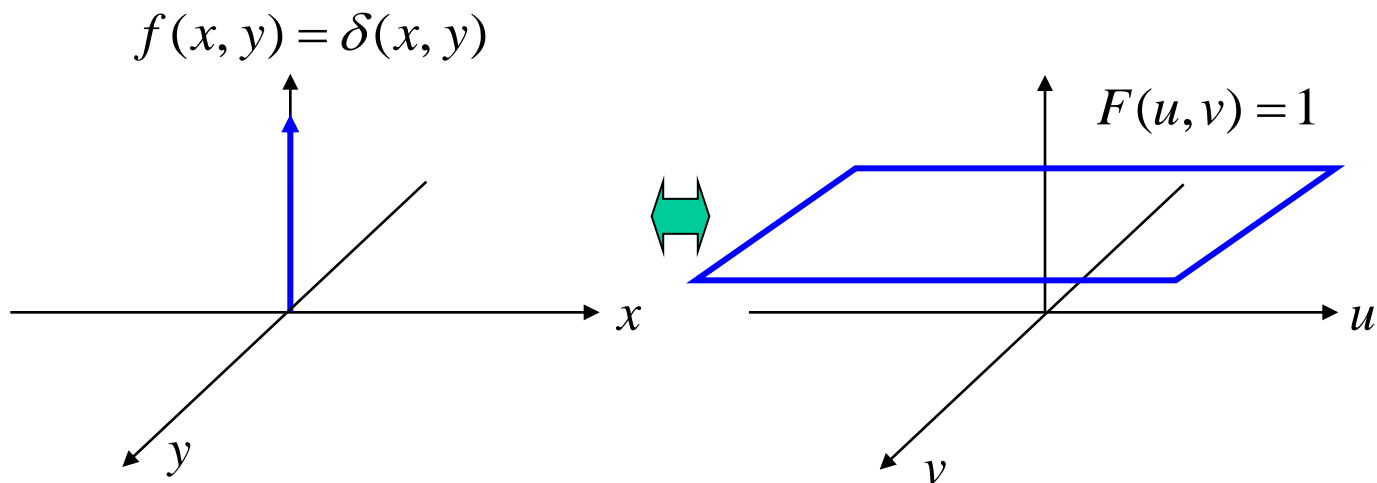
2変数のデルタ関数：

$\delta(x, y) : x = 0, y = 0$ で無限大になり，他で0の関数。



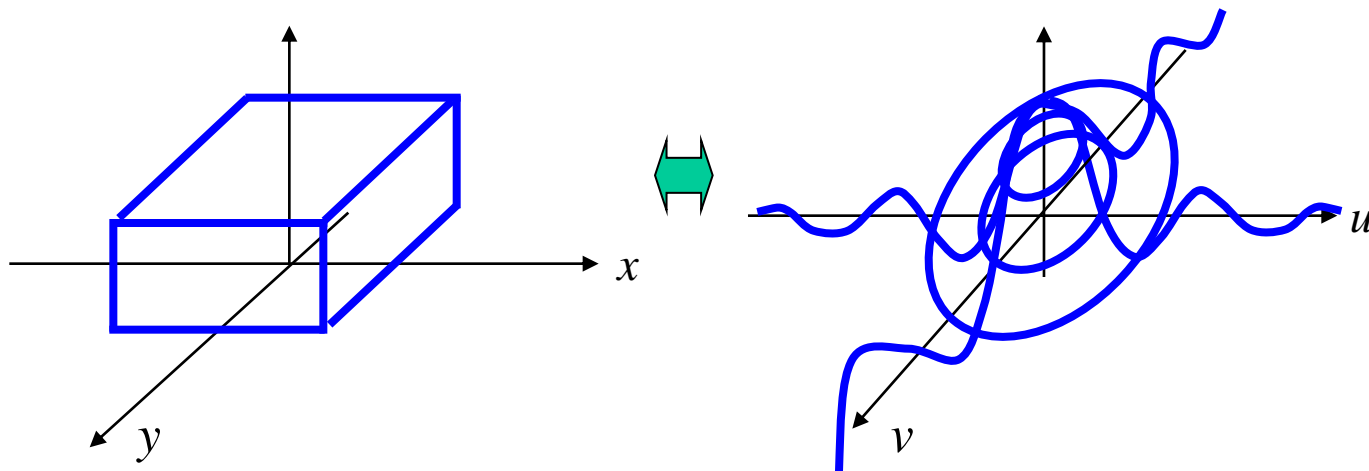
$\delta(x-a, y-b) : x = a, y = b$ で無限大になり，他で0の関数。

$$f(x, y) = \delta(x, y) \Leftrightarrow F(u, v) = 1$$

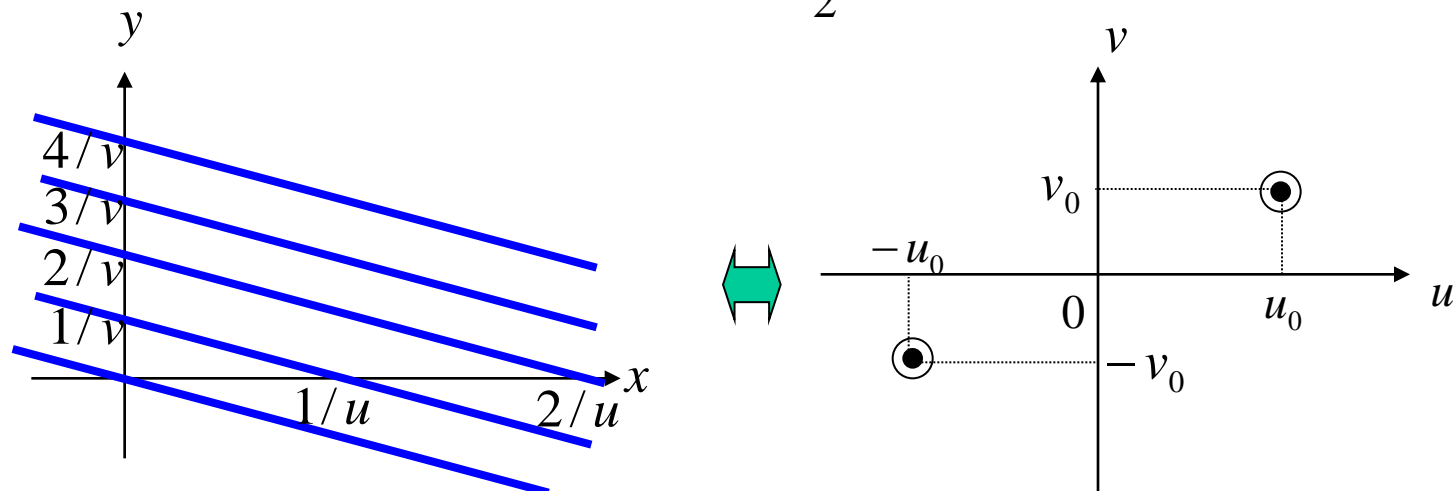


代表的な2次元フーリエ変換対(2)

$$f(x, y) = \text{rect}(x)\text{rect}(y) \Leftrightarrow F(u, v) = \text{sinc}(u)\text{sinc}(v)$$

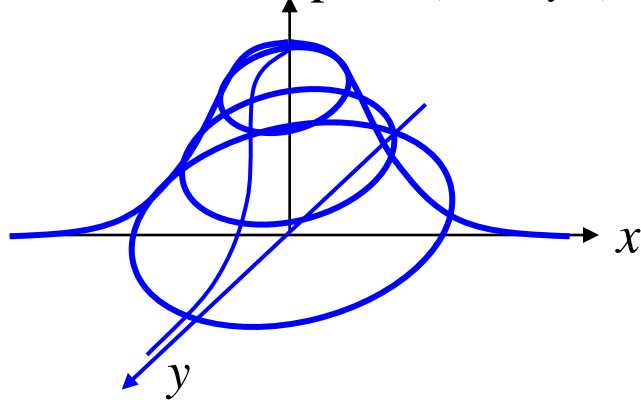


$$f(x, y) = \cos[2\pi(u_0x + v_0y)] \Leftrightarrow F(u, v) = \frac{1}{2} \{ \delta(u - u_0, v - v_0) + \delta(u + u_0, v + v_0) \}$$

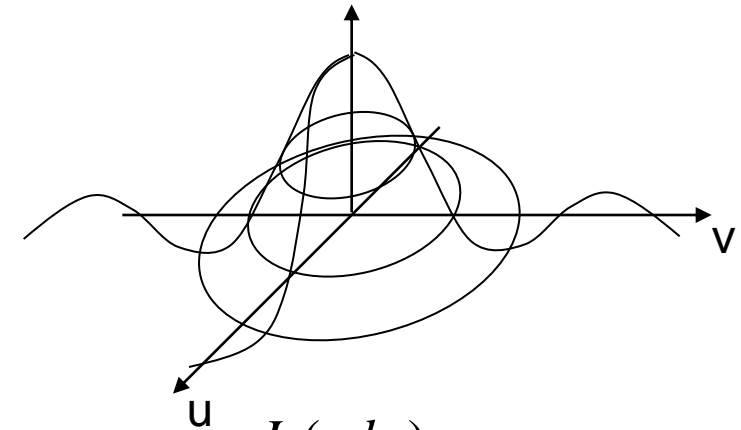
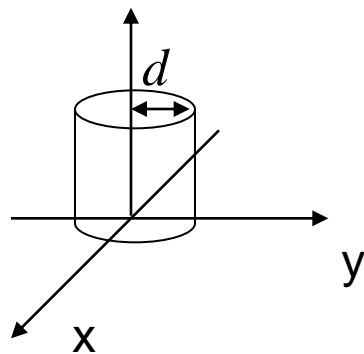
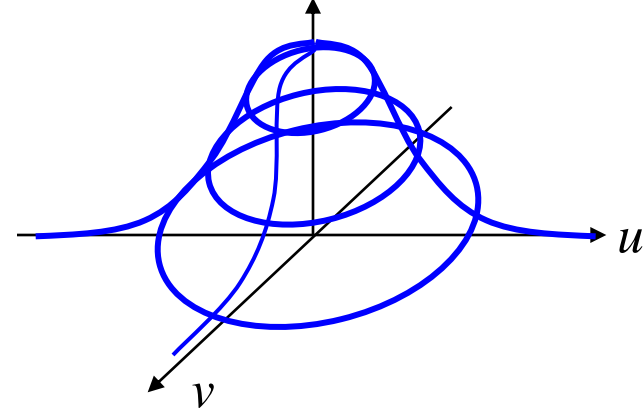


代表的な2次元フーリエ変換対(3)

Gauss関数 $f(x, y) = \exp[-\pi r^2]$
 $= \exp[-\pi(x^2 + y^2)]$



$F(u, v) = \exp[-\pi \rho^2]$
 $= \exp[-\pi(u^2 + v^2)]$



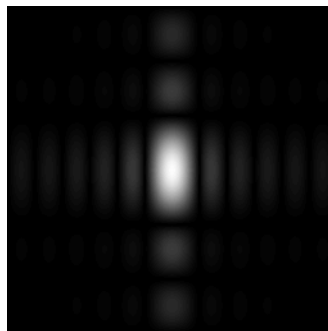
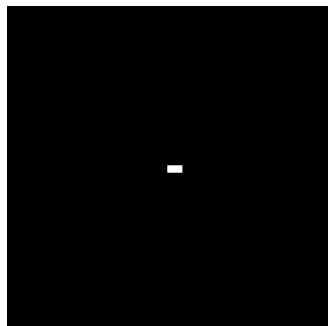
$f(x, y) = \text{circ}\left(\frac{r}{d}\right) \quad r = \sqrt{x^2 + y^2}$

$F(u, v) = \pi d^2 \frac{J_1(\pi d \rho)}{\pi d \rho}, \quad \rho = \sqrt{u^2 + v^2}$

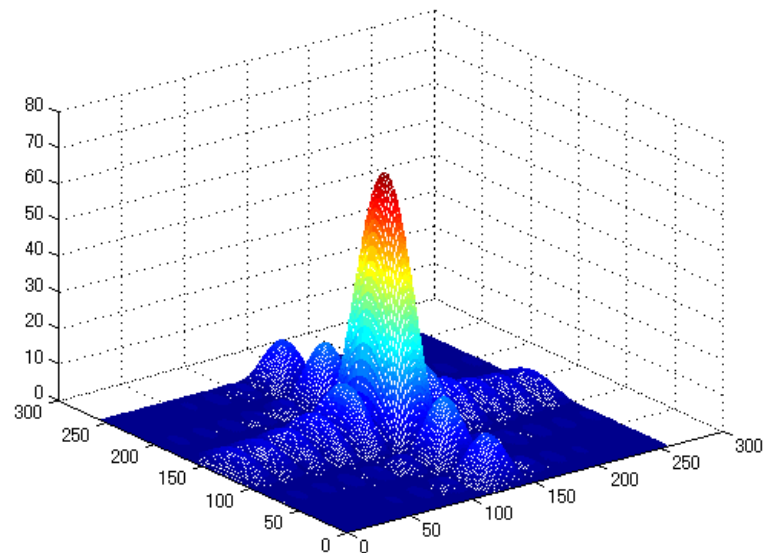
J_1 : ベッセル関数

2次元フーリエ変換の計算例－矩形1－

$$f(x, y) = \text{rect}\left(\frac{x}{a}\right)\text{rect}\left(\frac{y}{b}\right) \Leftrightarrow F(u, v) = \text{sinc}(au)\text{sinc}(bv)$$

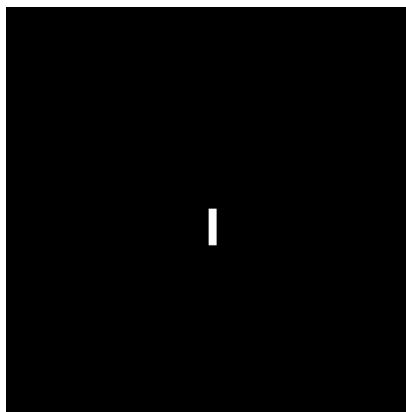


$$a = 12, b = 6$$

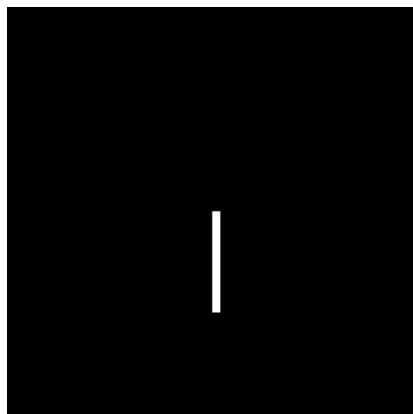
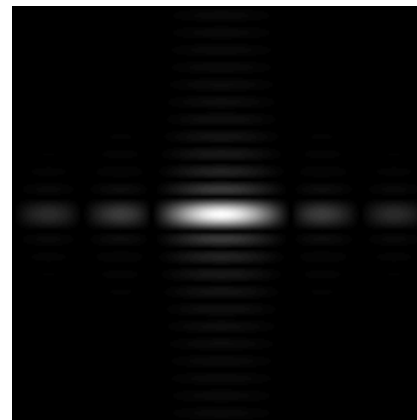


2次元フーリエ変換の計算例－矩形1－

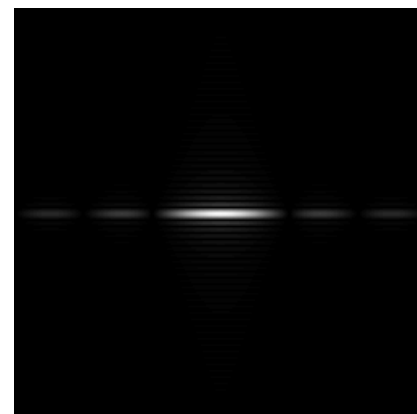
$$f(x, y) = \text{rect}\left(\frac{x}{a}\right)\text{rect}\left(\frac{y}{b}\right) \Leftrightarrow F(u, v) = \text{sinc}(au)\text{sinc}(bv)$$



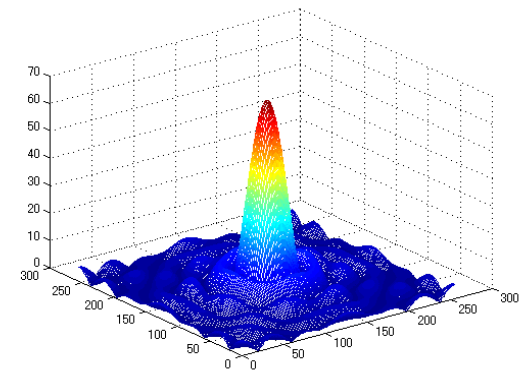
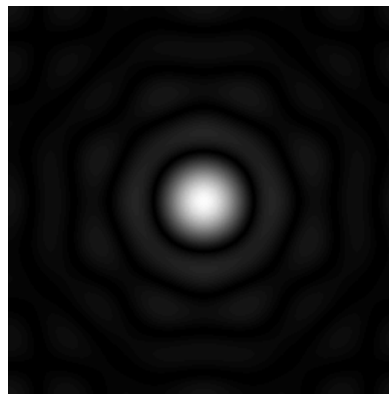
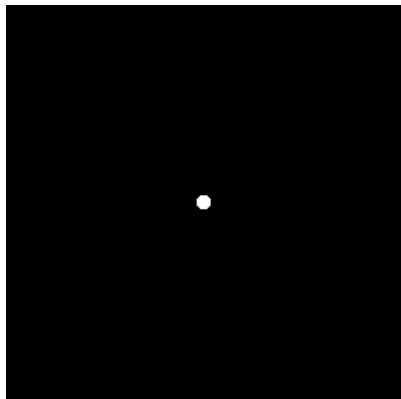
$$a = 6, b = 24$$



$$a = 6, b = 64$$



2次元フーリエ変換の計算例－円形1－



$$f(x, y) = \text{circ}\left(\frac{r}{d}\right) \quad r = \sqrt{x^2 + y^2}$$

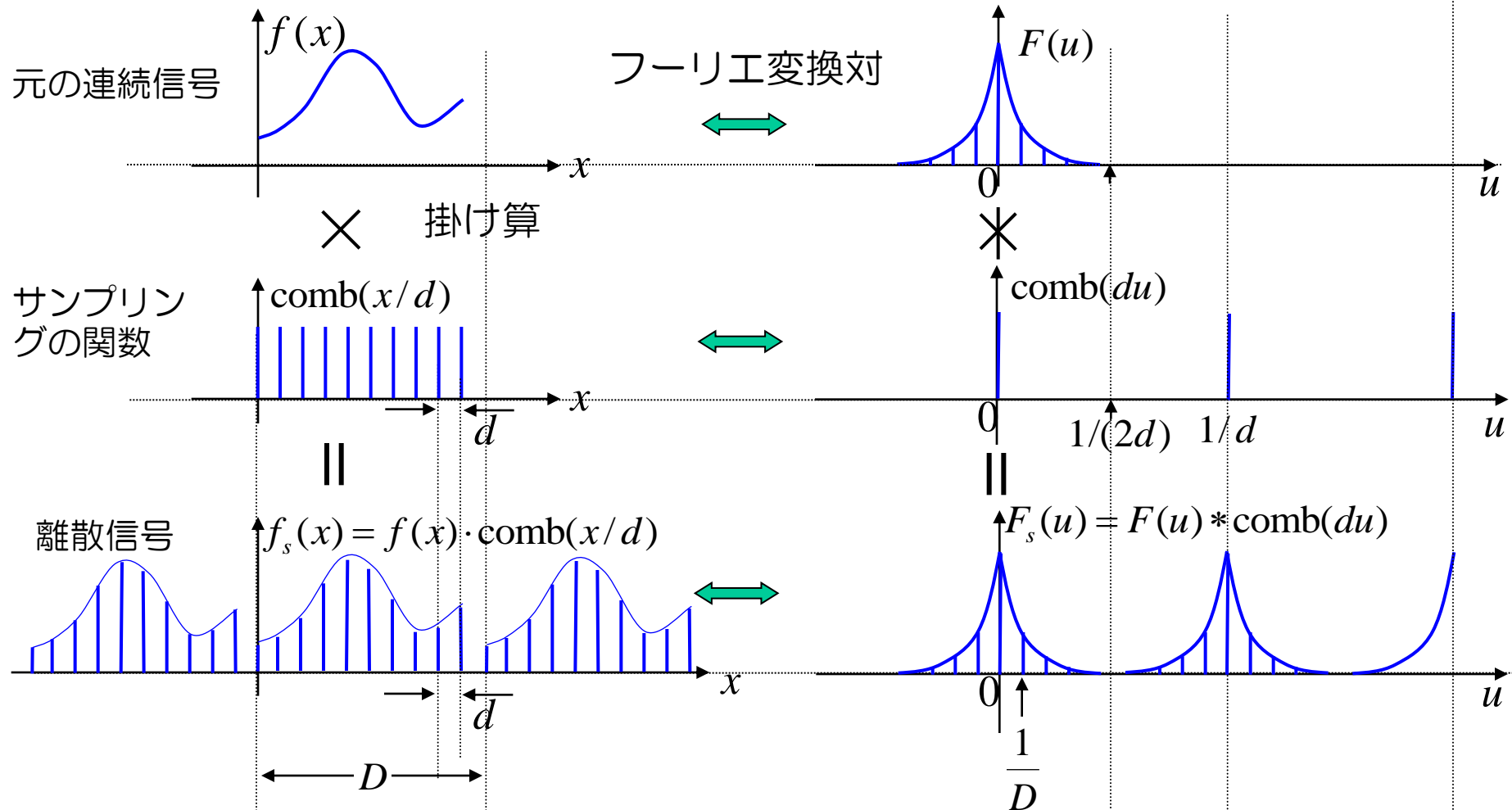
$$F(u, v) = \pi d^2 \frac{J_1(\pi d \rho)}{\pi d \rho}, \quad \rho = \sqrt{u^2 + v^2}$$

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

△ 離散フーリエ変換の概念 — まずは1次元 —

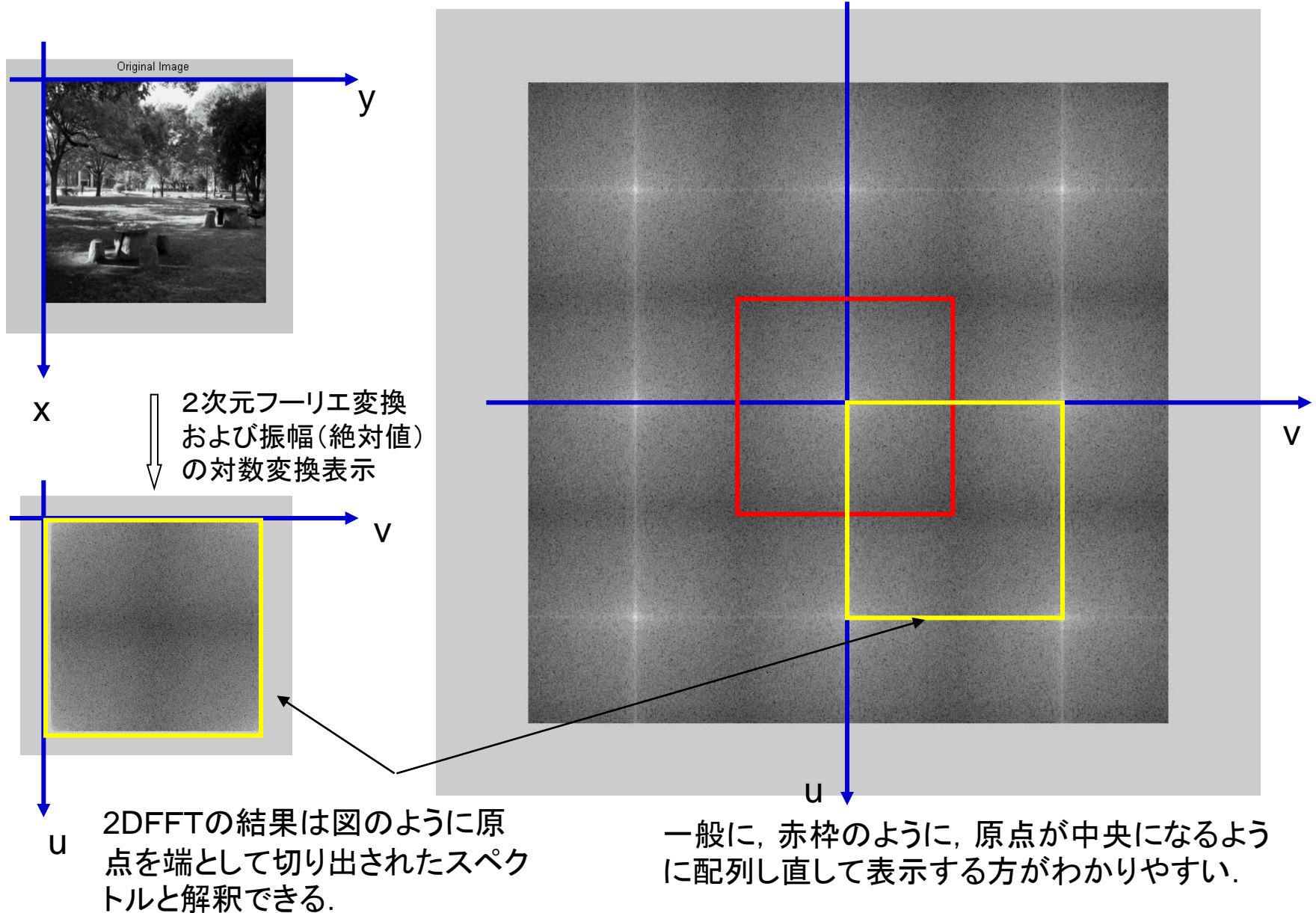


$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \exp(-j2\pi ux / N)$$

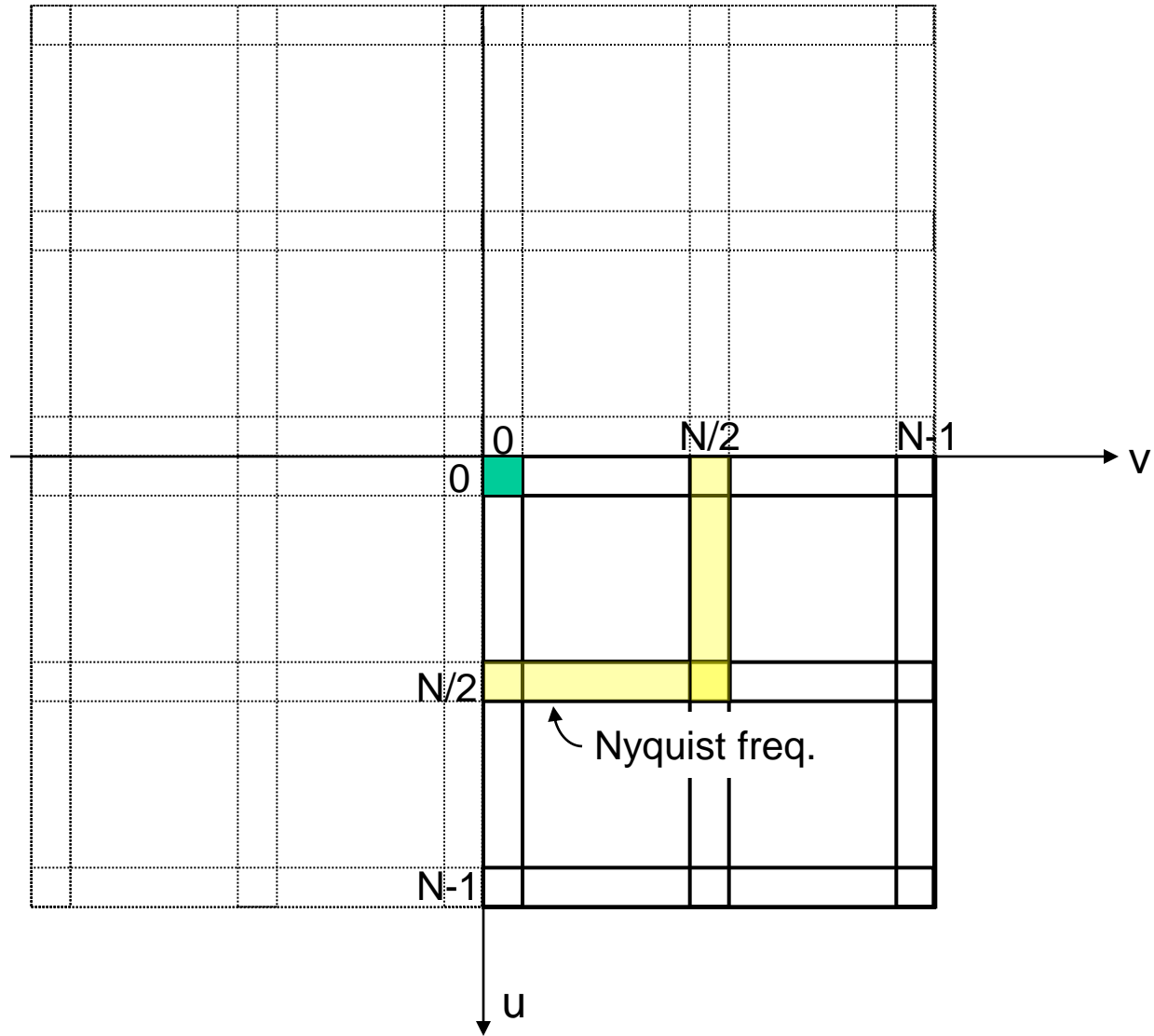
周期 D の正弦波
(余弦波) の成分

D の範囲に対して、基底関数を掛けてフーリエ成分を計算しているということは、暗黙のうちに上記のような実空間信号の周期性を仮定していることになる。

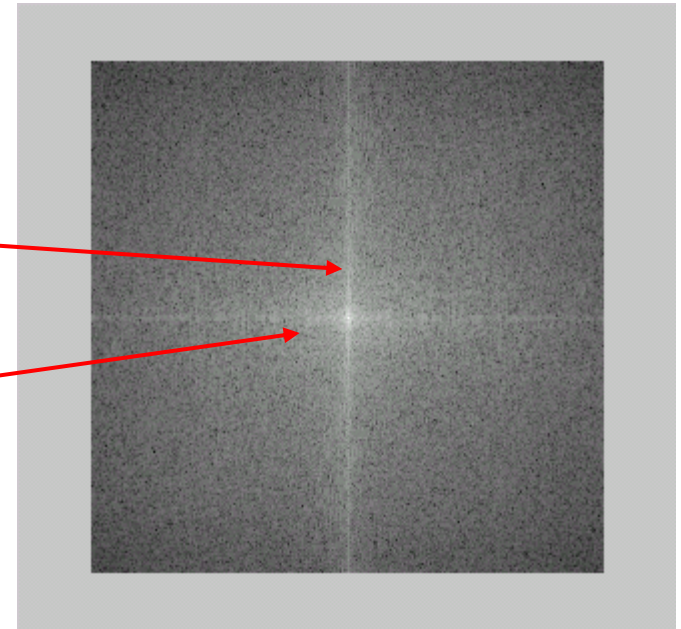
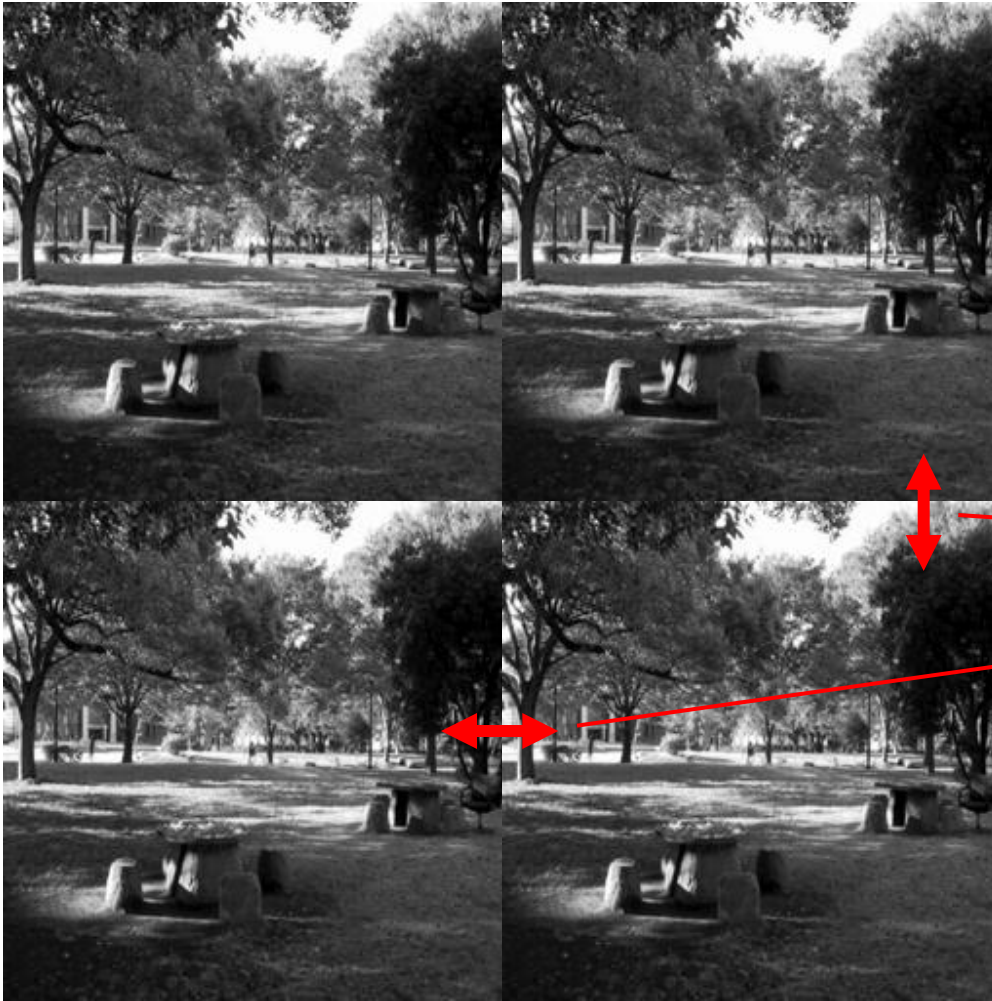
2次元離散フーリエ変換



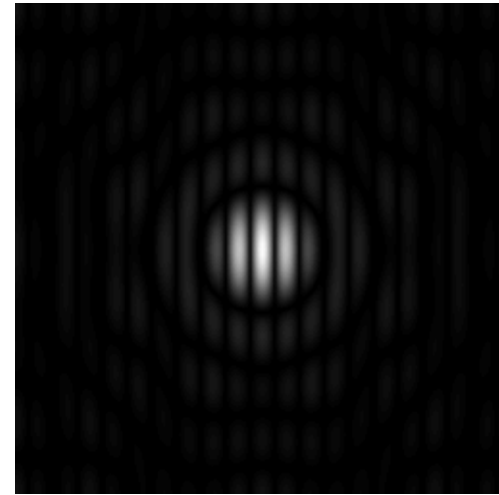
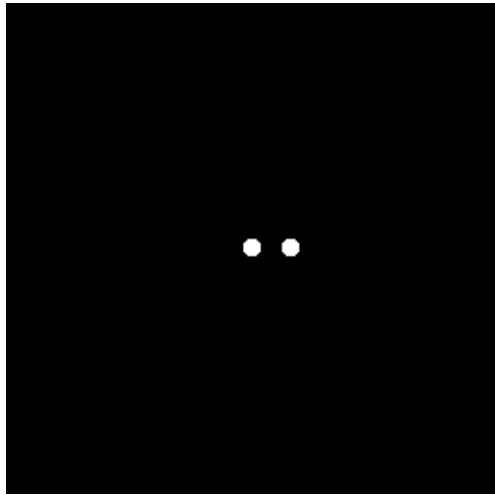
2次元離散フーリエ変換のデータの並び



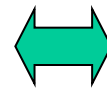
境界部分での不連続によるスペクトル



2次元フーリエ変換の計算例－円形2－



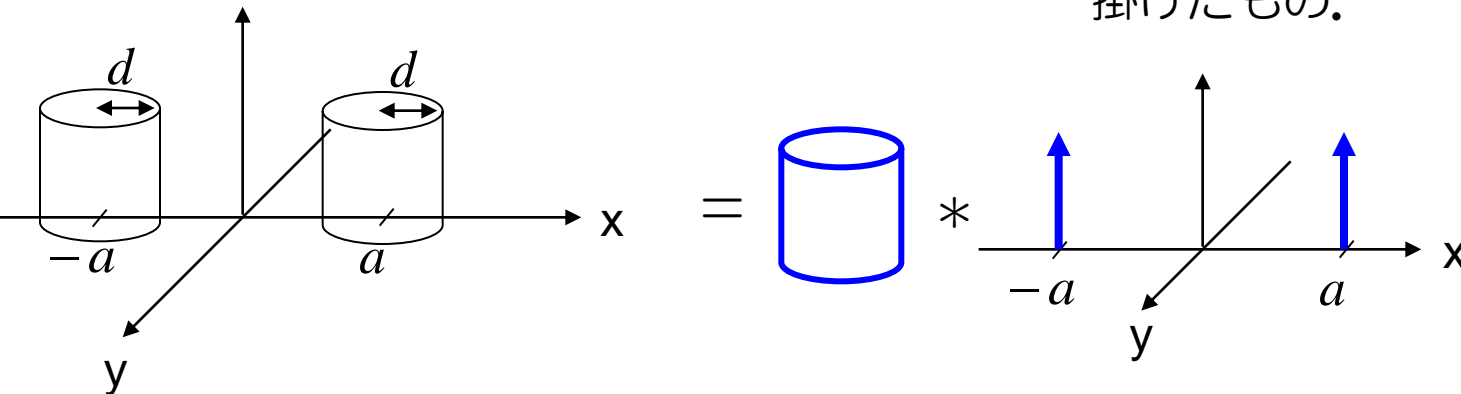
$$\begin{aligned}
 f(x, y) &= \text{circ}(r/d) * \delta(x+a, y) \\
 &\quad + \text{circ}(r/d) * \delta(x-a, y) \\
 &= \text{circ}(r/d) * \{\delta(x+a, y) + \delta(x-a, y)\}
 \end{aligned}$$



$$F(u, v) = \pi d^2 \frac{J_1(\pi d \rho)}{\pi d \rho} \times 2 \cos(2\pi a u)$$



ベッセル関数に $\cos(2\pi a u)$ を掛けたもの。



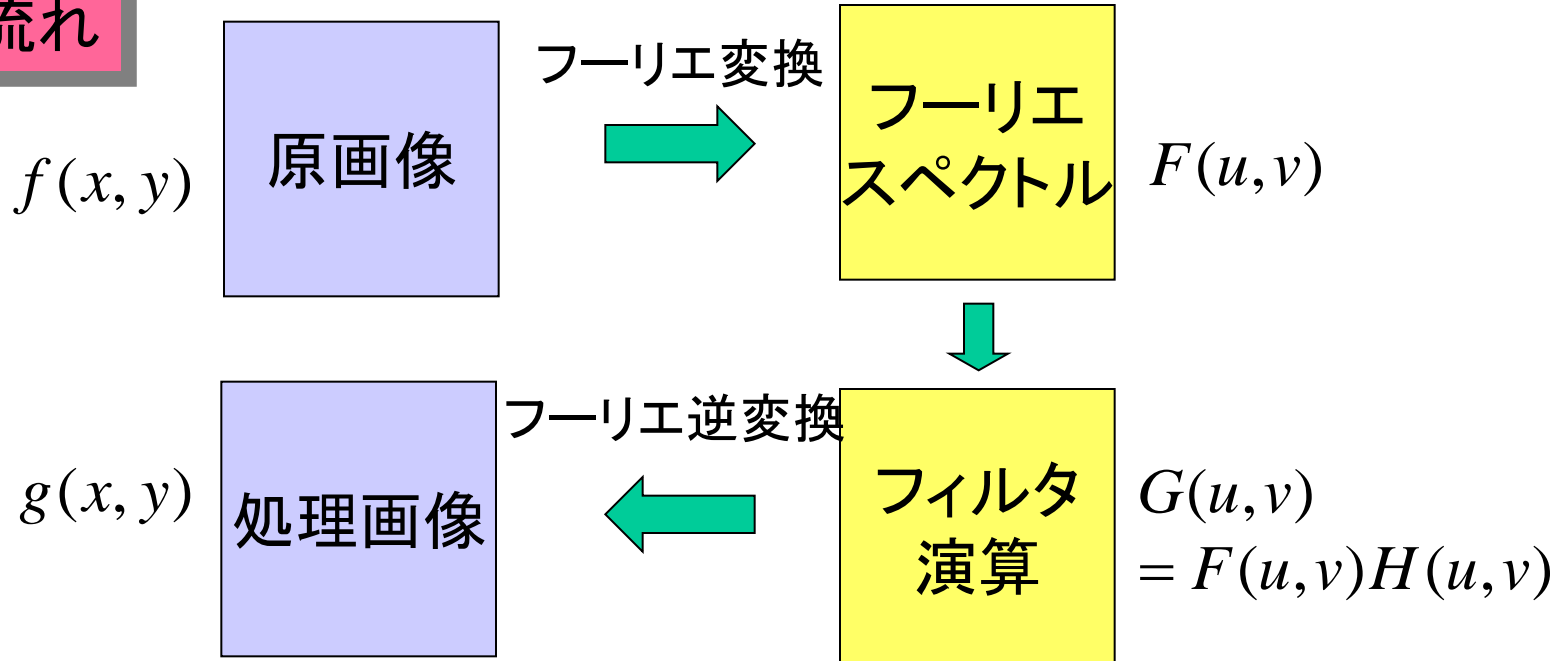
画像のフィルタリング処理

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

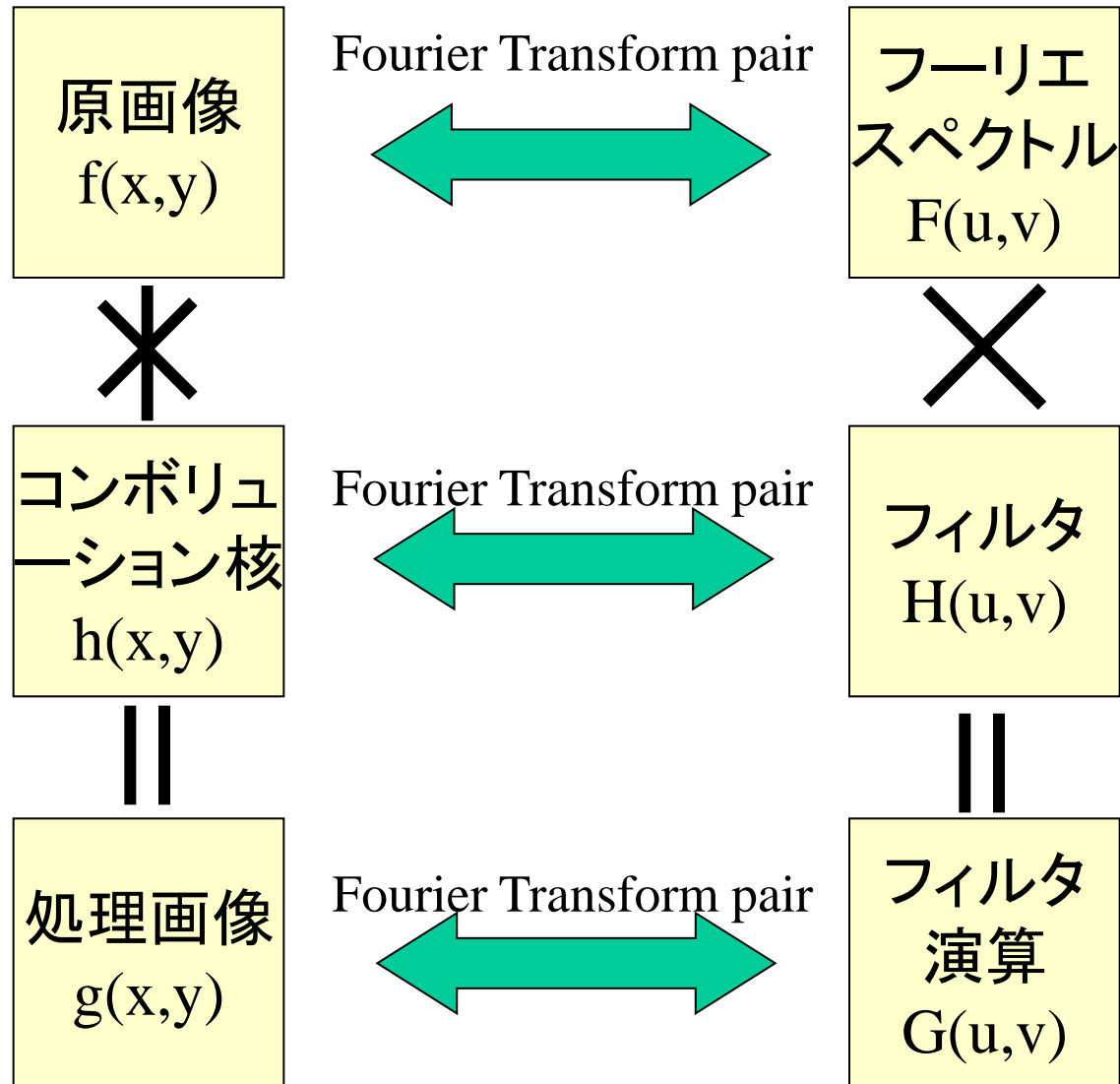
積

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

積

コンボリューション

処理の等価性

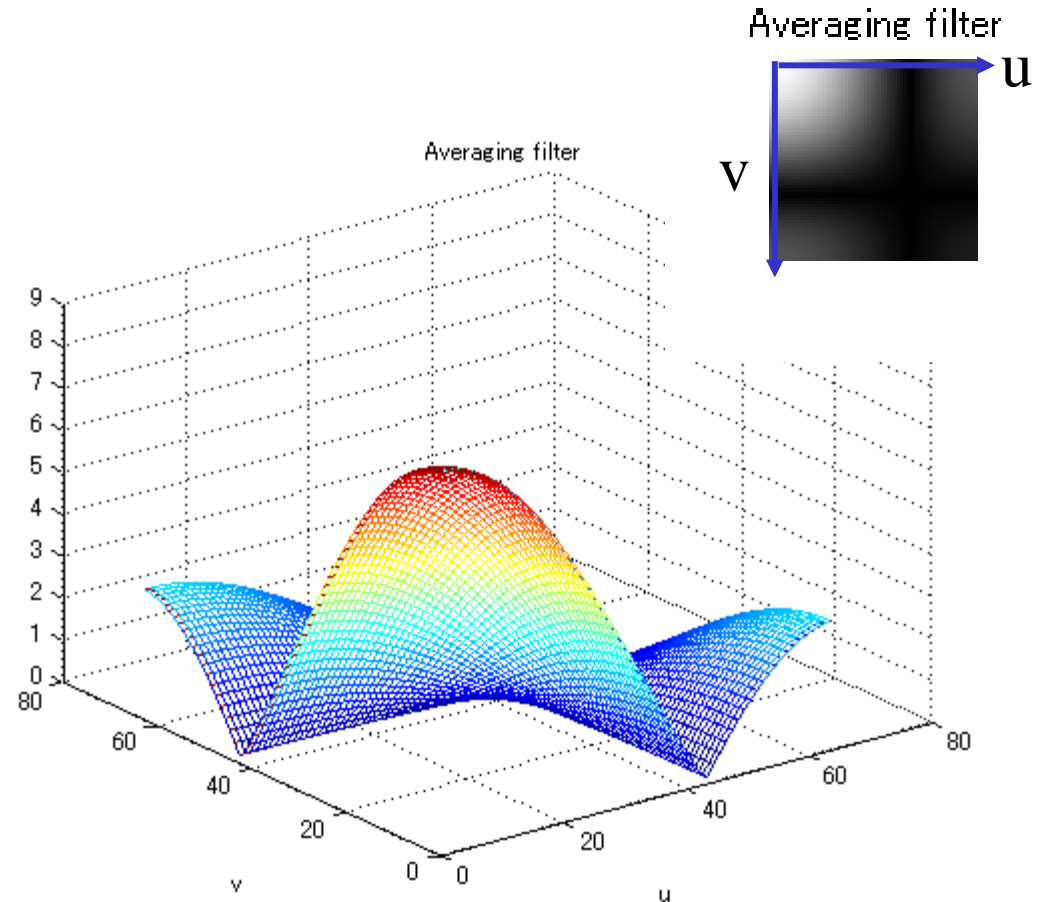
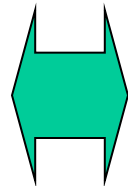


平滑化フィルタ

空間周波数フィルタ

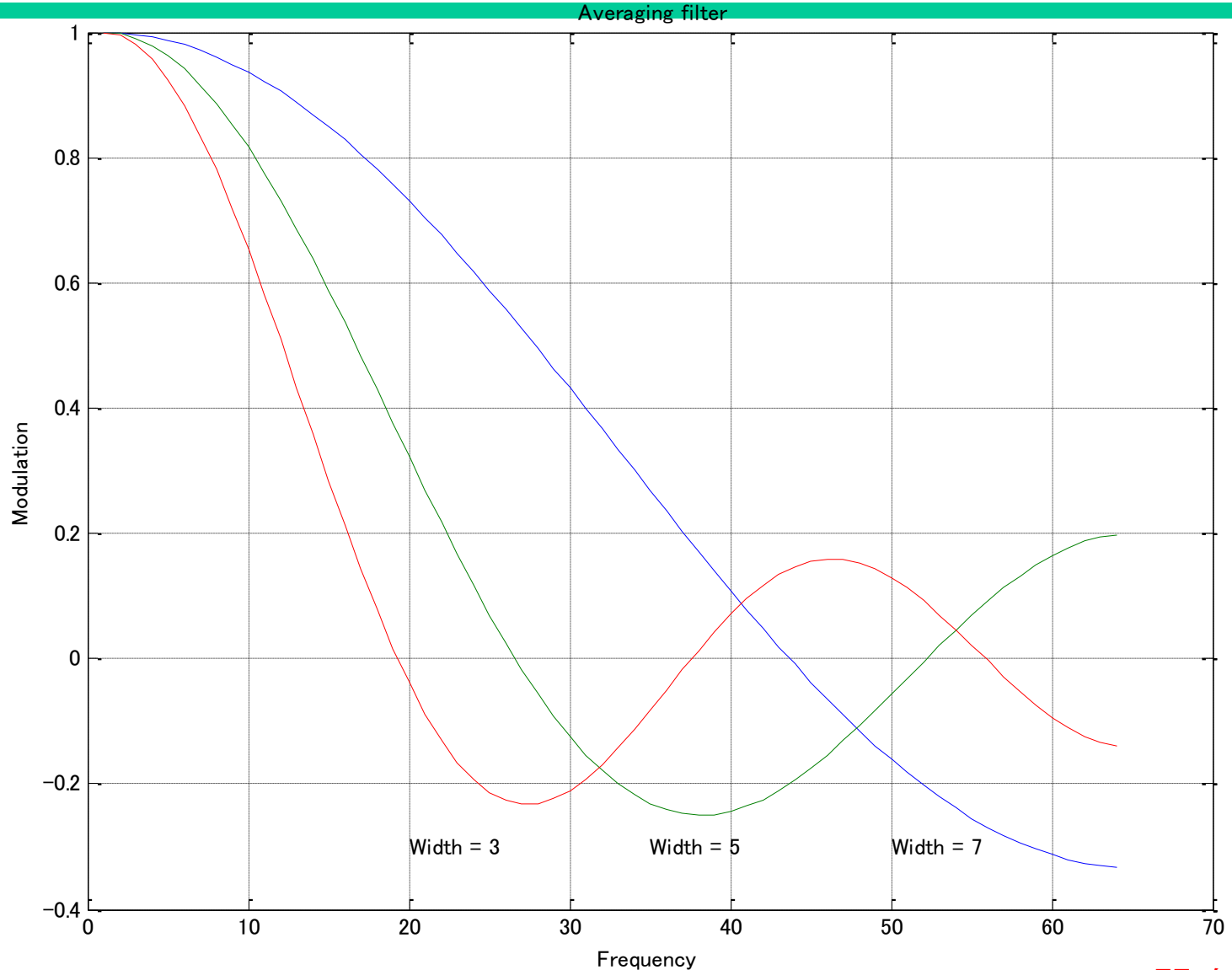
実空間でのフィルタ
(コンボリューション核)

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



(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

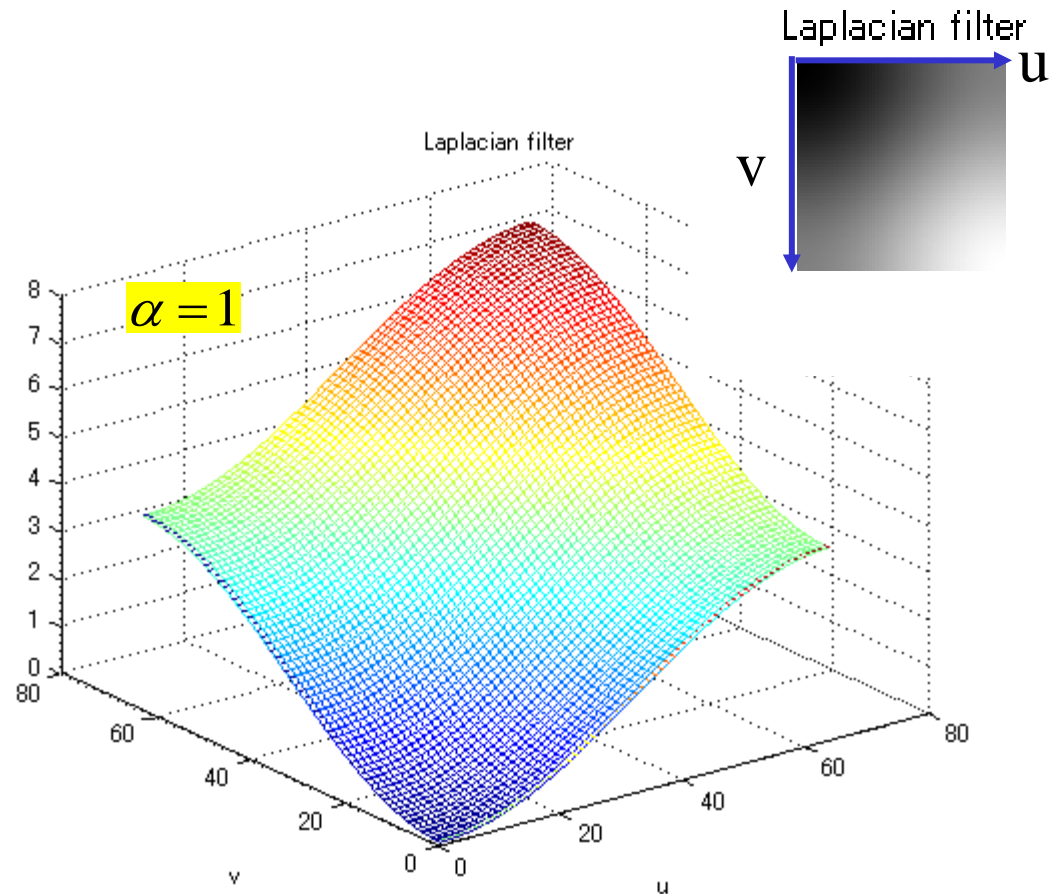
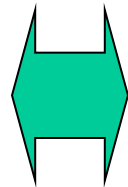
間違い！
3, 7が逆.

Laplacianフィルタ

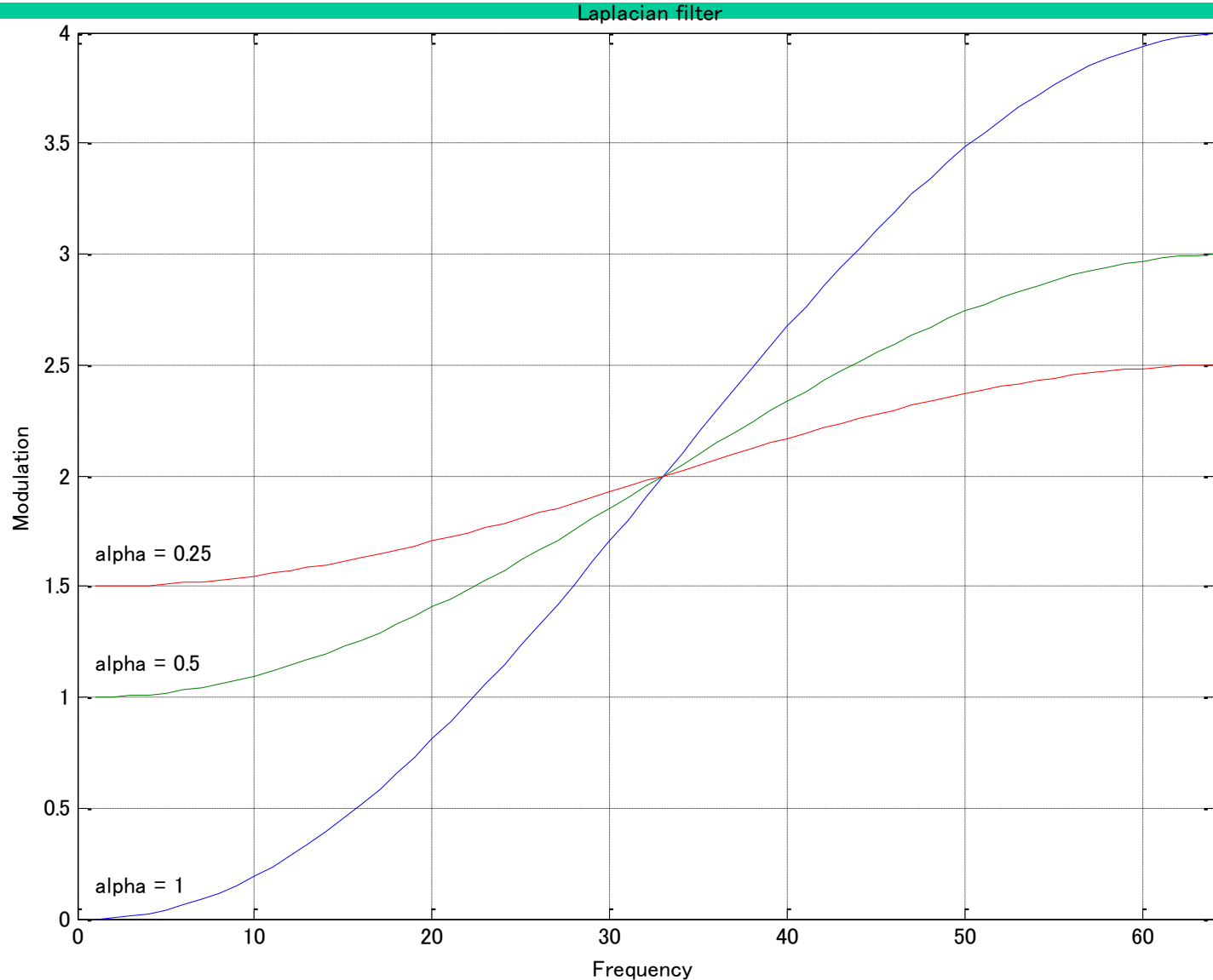
空間周波数フィルタ

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

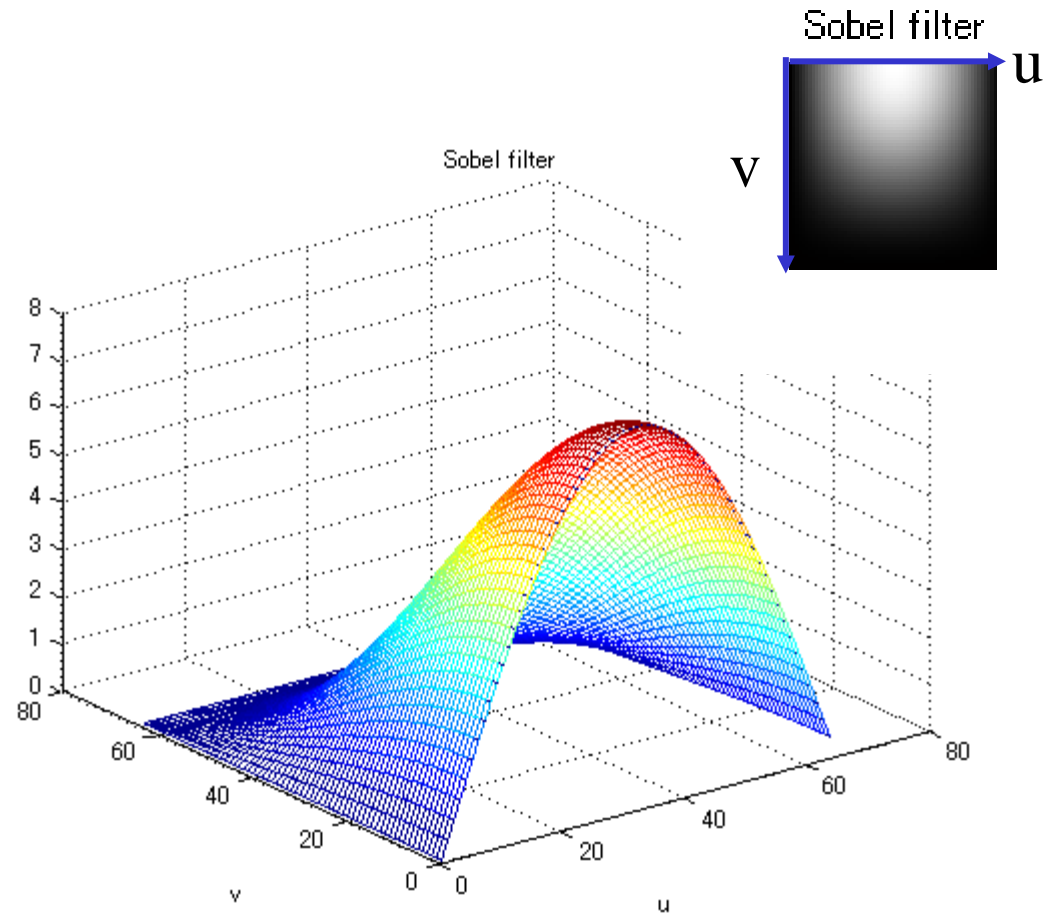
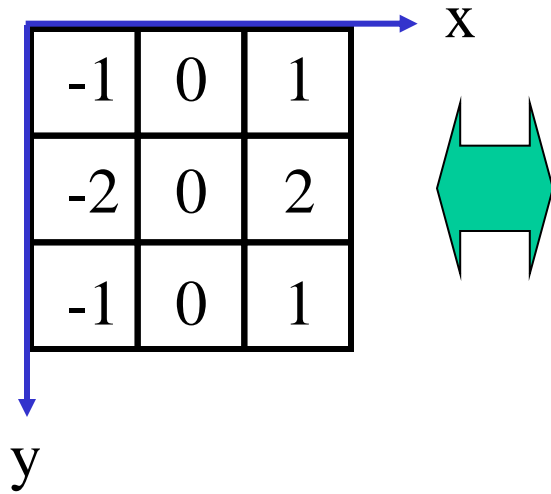


High pass filter

Sobel フィルタ

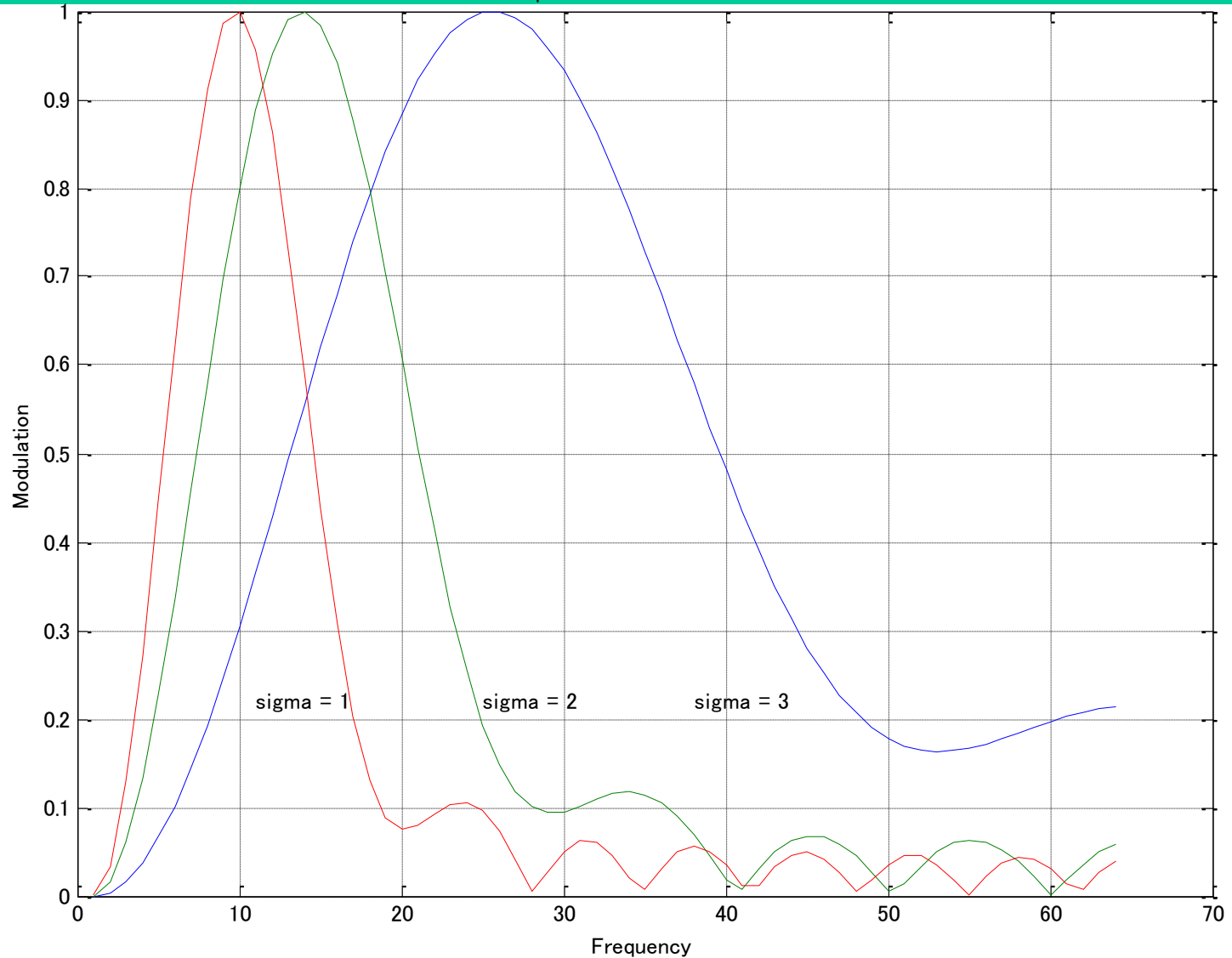
空間周波数フィルタ

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



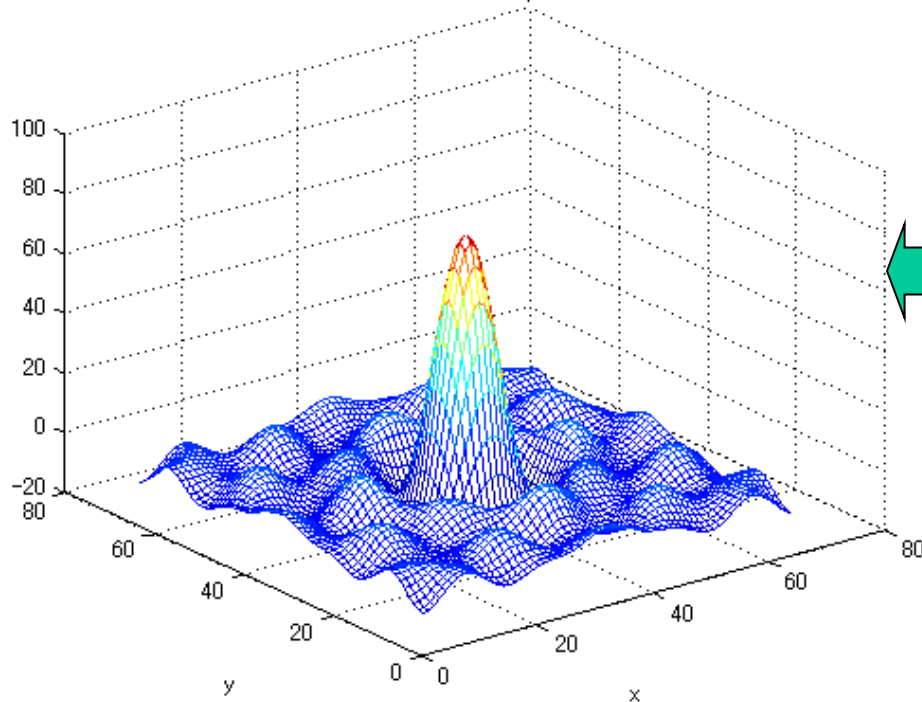
Band pass filter

間違い！
Sigmaの並びが逆.

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

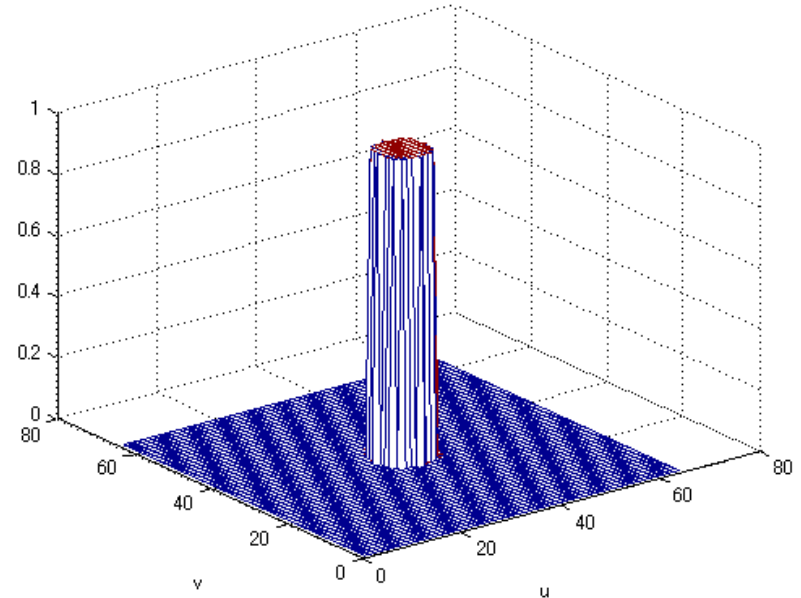
Convolution kernel in spatial domain



コンボリューション核

フーリエ空間

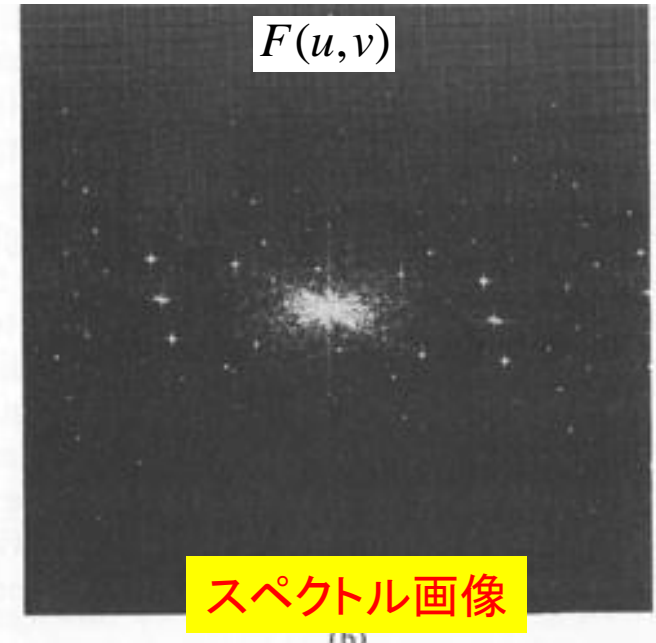
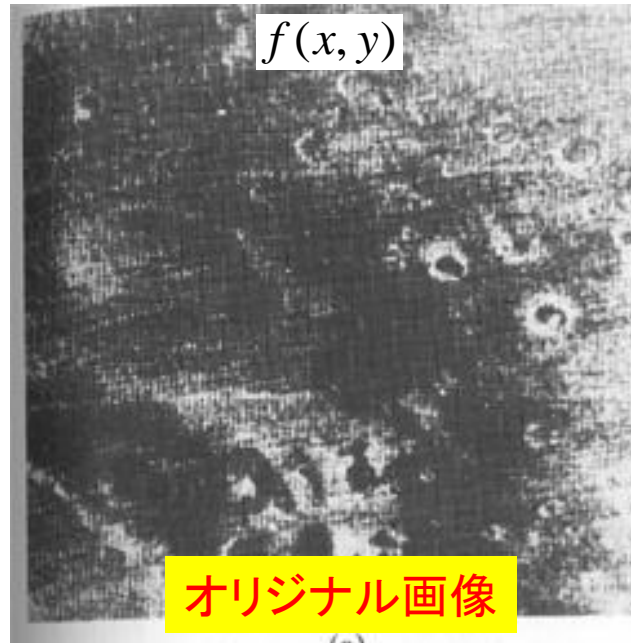
Filter in Fourier domain



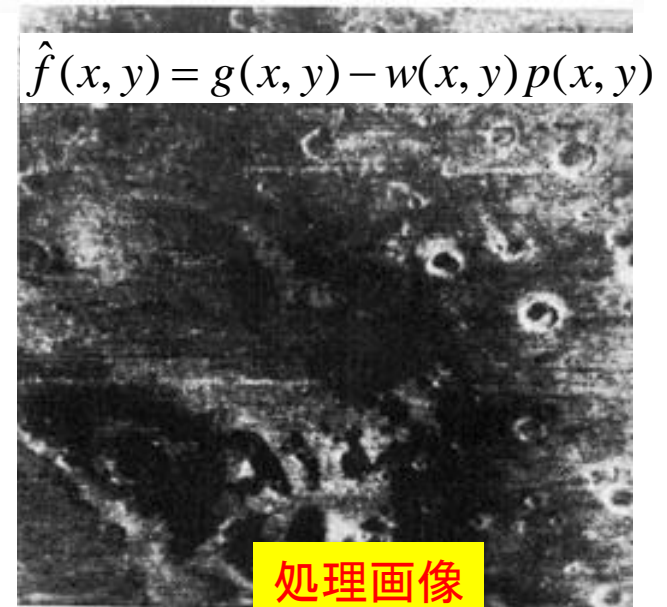
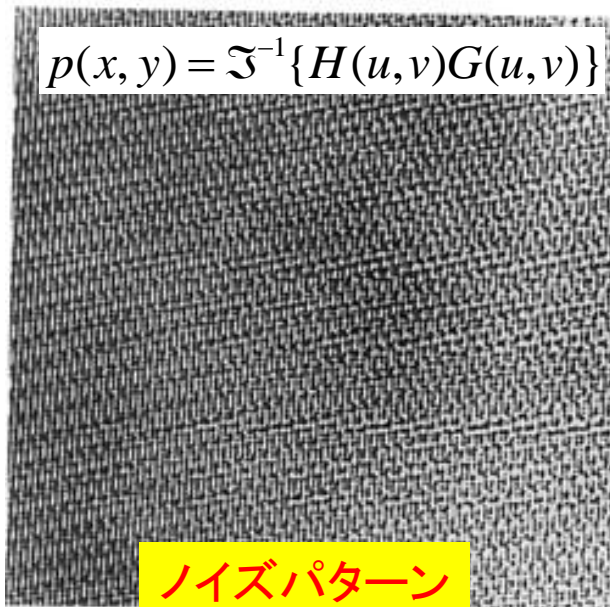
空間周波数フィルタ
Sharp-cut LPF

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.



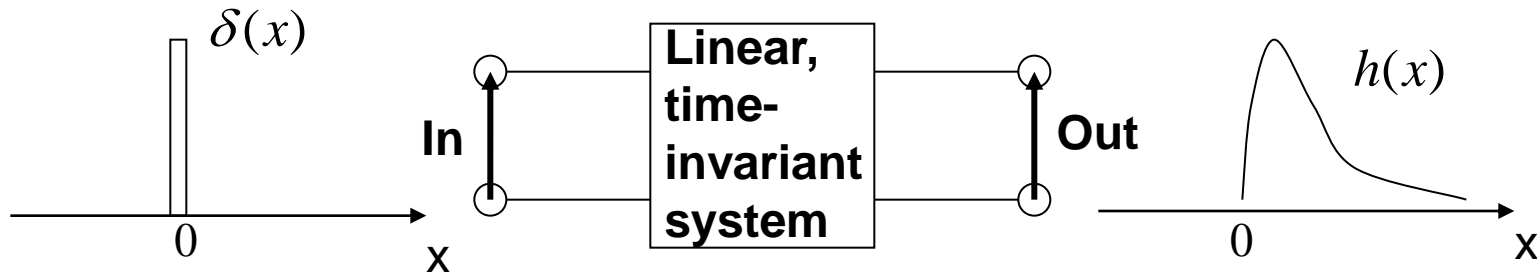
Digital Image Processing, R. C. Gonzalez and R. E. Woodsから引用

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

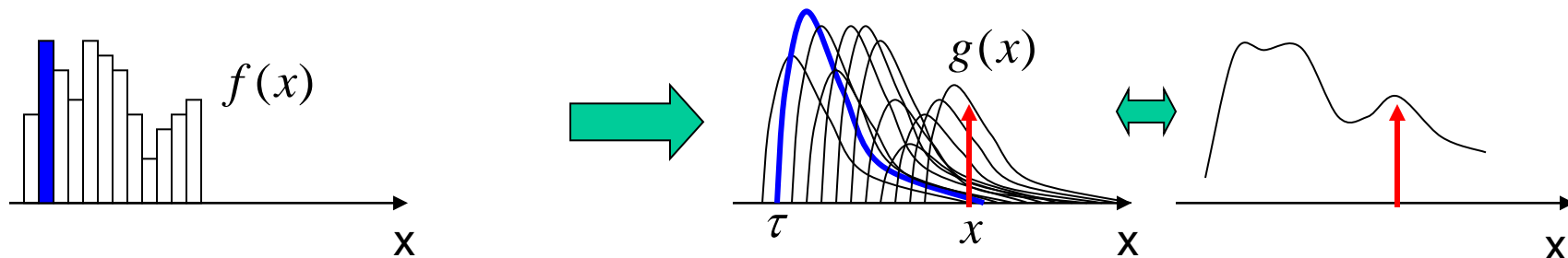


ディラックのデルタ関数
：インパルス関数

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



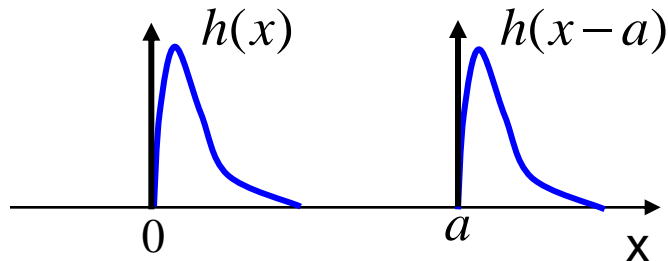
出力信号は入力信号と
インパルス応答との
コンボリューションで
表される。

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

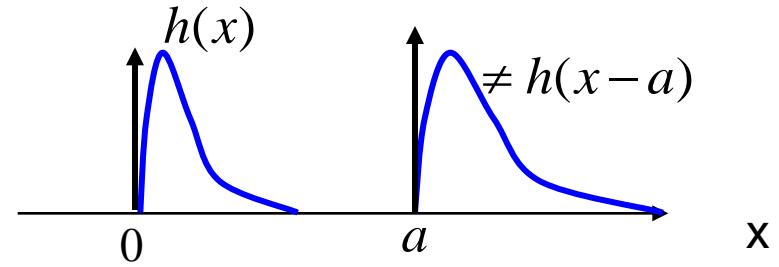
$$= h(x) * f(x)$$

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。

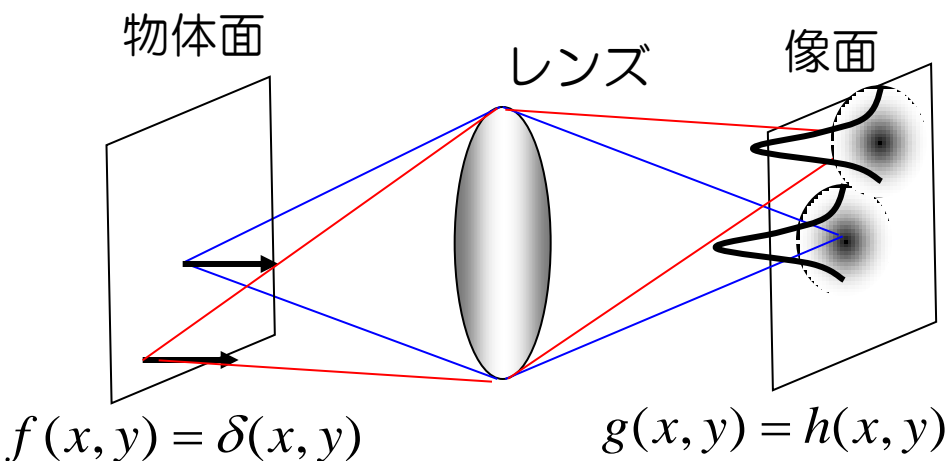


シフトインバリエント

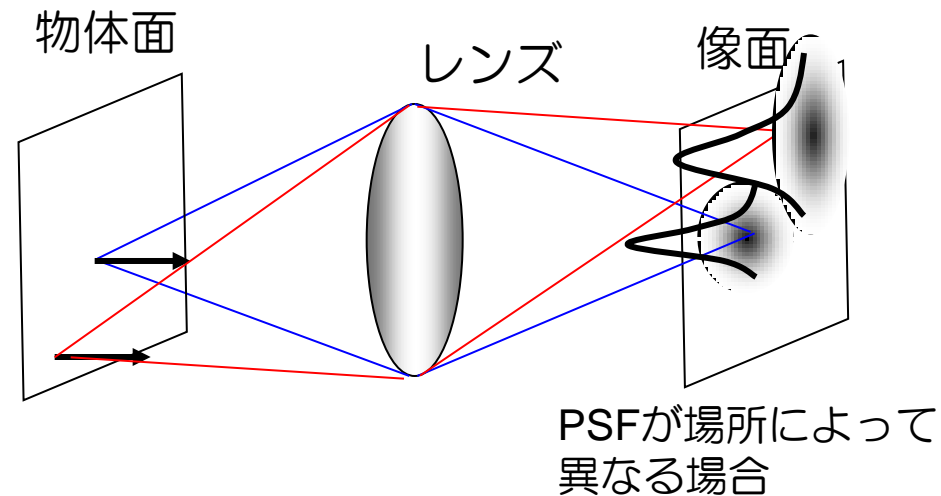


シフトバリエント

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

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。

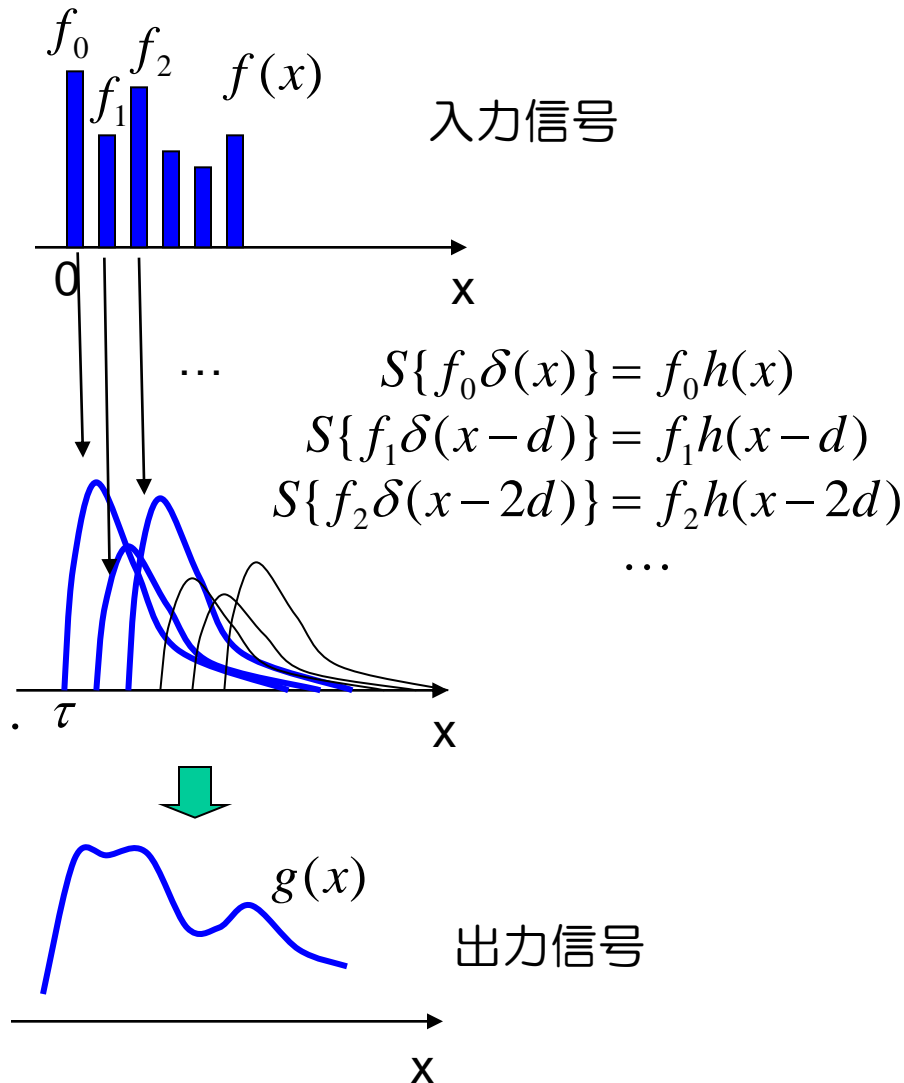
$$\begin{aligned} g(x) &= S\{a_1 f_1(x) + a_2 f_2(x)\} \\ &= a_1 S\{f_1(x)\} + a_2 S\{f_2(x)\} \end{aligned}$$

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：

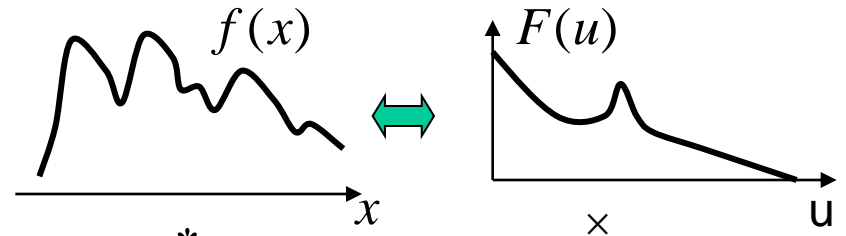
$$g(x) = f_0 h(x) + f_1 h(x-d) + f_2 h(x-2d) + \dots$$



周波数空間で考える

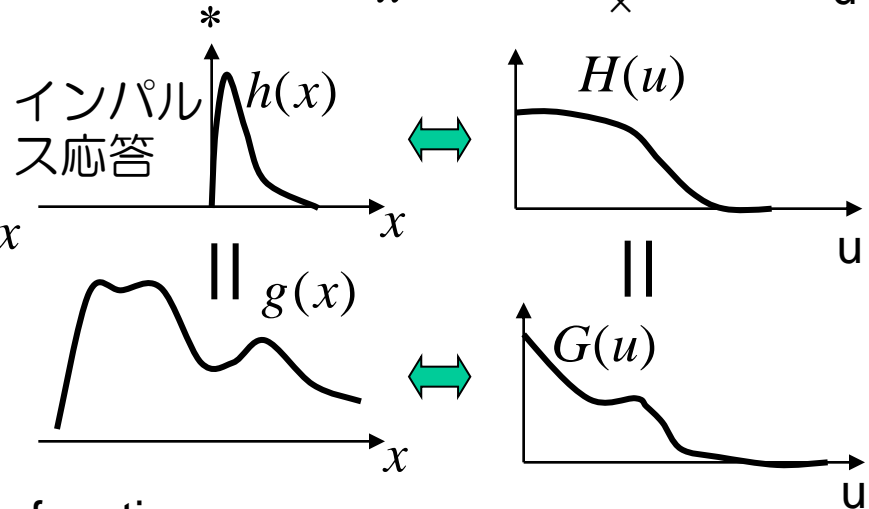
入力信号のスペクトル：

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



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

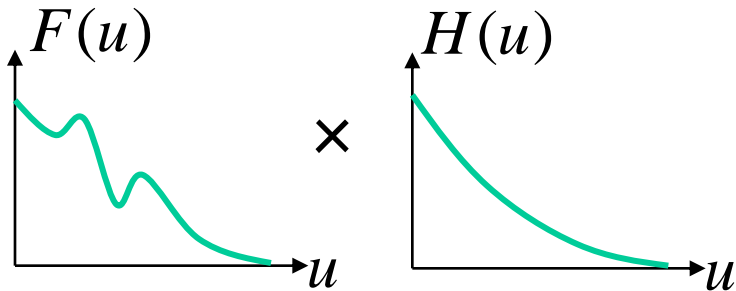
理想画像: $f(x, y)$

劣化の点像分布関数: $h(x, y)$

劣化画像:

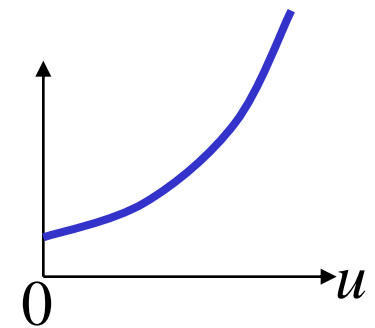
$$g(x, y) = f(x, y) * h(x, y) + n(x, y)$$

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



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

