

情報デザイン専攻

# 画像情報処理論及び演習II

## -後期課題の復習- レポート4・5ヒント

第12回講義  
水曜日 1限  
教室6218

吉澤 信  
shin@riken.jp, 非常勤講師  
大妻女子大学 社会情報学部

独立行政法人  
理化学研究所

Shin Yoshizawa: shin@riken.jp

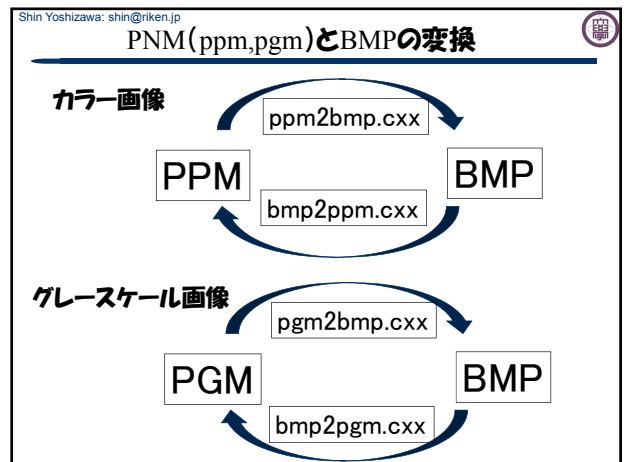
# レポート4のヒント

Shin Yoshizawa: shin@riken.jp

## 演習:レポート1の内容その1

[www.riken.jp/brict/Yoshizawa/Lectures/index.html](http://www.riken.jp/brict/Yoshizawa/Lectures/index.html)  
[www.riken.jp/brict/Yoshizawa/Lectures/Ex07.zip](http://www.riken.jp/brict/Yoshizawa/Lectures/Ex07.zip)

- ✓ Q1: bmp画像をppm画像へ変換するプログラムを作成.
- ✓ Q2: ppm画像をbmp画像へ変換するプログラムを作成.
- ✓ Q3: bmp画像をpgm画像へ変換するプログラムを作成.
- ✓ Q4: pgm画像をbmp画像へ変換するプログラムを作成.
- ✓ Q5, Q6: 講義のスライドをよく読む or 自分で調べる  
Lec15.pdf、Lec16.pdfを参照.

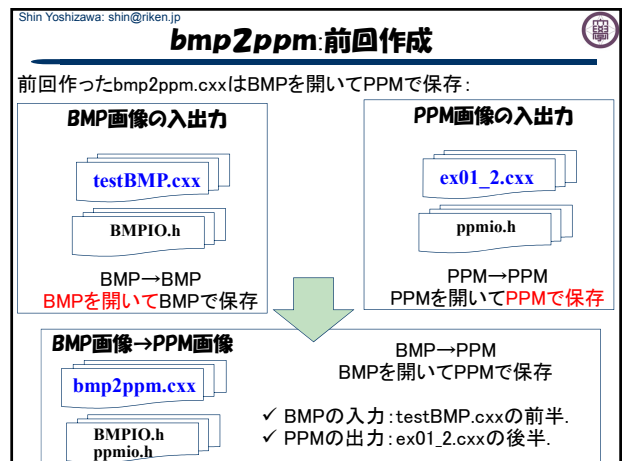


Shin Yoshizawa: shin@riken.jp

## PNM(ppm,pgm)とBMPの変換2

ソースコード作成→コンパイル→実行

- ✓ コンパイル:
  - g++ -o bmp2ppm bmp2ppm.cxx
  - g++ -o ppm2bmp ppm2bmp.cxx
  - g++ -o bmp2pgm bmp2pgm.cxx
  - g++ -o pgm2bmp pgm2bmp.cxx
- ✓ 実行:「./実行ファイル 入力画像 出力画像」
  - ./bmp2ppm lena.bmp lena.ppm
  - ./ppm2bmp lena.ppm lena\_ppm2bmp.bmp
  - ./bmp2pgm lena.bmp lena.pgm
  - ./pgm2bmp lena.pgm lena\_pgm2bmp.bmp



Shin Yoshizawa: shin@riken.jp

## ppm2bmp

ppm2bmp.cxxはPPMを開いてBMPで保存 : bmp2ppmの逆.

**BMP画像の入出力**

testBMP.cxx  
BMPIO.h

BMP→BMP  
BMPを開いてBMPで保存

**PPM画像の入出力**

ex01\_2.cxx  
ppmio.h

PPM→PPM  
PPMを開いてPPMで保存

**PPM画像→BMP画像**

ppm2bmp.cxx  
BMPIO.h  
ppmio.h

**PPM→BMP**  
PPMを開いてBMPで保存

✓ BMPの出力 : testBMP.cxxの後半.  
✓ PPMの入力 : ex01\_2.cxxの前半.

Shin Yoshizawa: shin@riken.jp

## bmp2pgm

bmp2pgm.cxxはBMPを開いてPGMで保存 : Gray=(R+G+B)/3

**BMP画像の入出力**

testBMP.cxx  
BMPIO.h

BMP→BMP  
BMPを開いてBMPで保存

**PGM画像の入出力**

ex01.cxx  
pgmio.h

PGM→PGM  
PGMを開いてPGMで保存

**BMP画像→PGM画像**

bmp2pgm.cxx  
BMPIO.h  
pgmio.h

**BMP→PGM**  
BMPを開いてPGMで保存

✓ BMPの入力 : testBMP.cxxの前半.  
✓ PGMの出力 : ex01.cxxの後半.

Shin Yoshizawa: shin@riken.jp

## pgm2bmp

pgm2bmp.cxxはPGMを開いてBMPで保存 : saveBMP(Image\*, char\*)

**BMP画像の入出力**

testBMP.cxx  
BMPIO.h

BMP→BMP  
BMPを開いてBMPで保存

**PGM画像の入出力**

ex01.cxx  
pgmio.h

PGM→PGM  
PGMを開いてPGMで保存

**PGM画像→BMP画像**

pgm2bmp.cxx  
BMPIO.h  
pgmio.h

**PGM→BMP**  
PGMを開いてBMPで保存

✓ BMPの出力 : testBMP.cxxの後半.  
✓ PGMの入力 : ex01.cxxの前半.

Shin Yoshizawa: shin@riken.jp

## カラーからグレースケールへの変換

[www.riken.jp/briect/Yoshizawa/Lectures/Ex07.zip](http://www.riken.jp/briect/Yoshizawa/Lectures/Ex07.zip)

- カラー画像(ppm)を読み込んでR,G,Bの平均値を輝度値とするグレースケール画像(pgm)を保存するプログラムを作成.

Shin Yoshizawa: shin@riken.jp

## 演習:レポート1の内容その2

[www.riken.jp/briect/Yoshizawa/Lectures/index.html](http://www.riken.jp/briect/Yoshizawa/Lectures/index.html)  
[www.riken.jp/briect/Yoshizawa/Lectures/Ex08.zip](http://www.riken.jp/briect/Yoshizawa/Lectures/Ex08.zip)

- ✓ Q7 : Ex08.zipのtestDCT.cxxを編集して円状に高周波をゼロにするローパスフィルタのプログラムを作成.
- ✓ Q8 : Q7プログラムにて16x16のブロックで半径1,2,3,4,8で実行した結果画像(下記の様な逆変換後の画像も)を載せよ.

半径1
半径2
半径4
半径8
半径16

Shin Yoshizawa: shin@riken.jp

## 周波数フィルタ

- ✓ testDCT.cxxを編集して円状に高周波をゼロにするローパスフィルタを作ってみましょう!
- ✓ 16x16のブロックで半径1,2,3,4,8で実行してみてください.
- ✓ ヒント : testDCT.cxxは四角に低周波を残しているの、円状にするだけ.

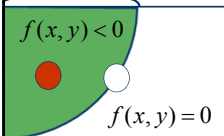
Shin Yoshizawa: shin@riken.jp

### 円の式

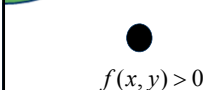
✓ 半径 $r$ の円の式は、
$$x^2 + y^2 = r^2$$

$r$  つまり、座標 $(x,y)$ が上記式を満たす点の集合が円.


$f(x,y) < 0$



$f(x,y) = 0$



$f(x,y) > 0$



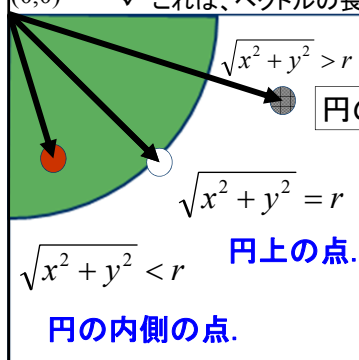
$f(x,y) = x^2 + y^2 - r^2$   
とすると $(x,y)$ は、

- $f(x,y) = 0$  円上の点.
- $f(x,y) > 0$  円の外側の点.
- $f(x,y) < 0$  円の内側の点.

Shin Yoshizawa: shin@riken.jp

### 円の式

✓ これは、ベクトルの長さで考えても同じ:



$\sqrt{x^2 + y^2} > r$  円の外側の点.

$\sqrt{x^2 + y^2} = r$  円上の点.

$\sqrt{x^2 + y^2} < r$  円の内側の点.

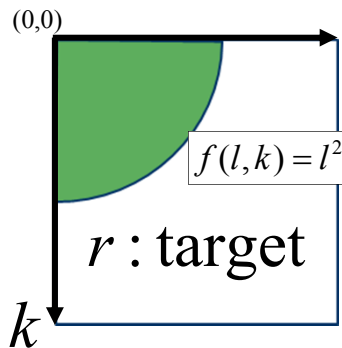
$f(x,y) = x^2 + y^2 - r^2$   
とすると $(x,y)$ は、

- $f(x,y) = 0$  円上の点.
- $f(x,y) > 0$  円の外側の点.
- $f(x,y) < 0$  円の内側の点.

Shin Yoshizawa: shin@riken.jp

### 円の式

✓ プログラム内では、



$f(l,k) = l^2 + k^2 - \text{target}^2$

$r : \text{target}$

$f(x,y) = x^2 + y^2 - r^2$   
とすると $(x,y)$ は、

- $f(x,y) = 0$  円上の点.
- $f(x,y) > 0$  円の外側の点.
- $f(x,y) < 0$  円の内側の点.

Shin Yoshizawa: shin@riken.jp

### 正解例

✓ 32x32のブロックで実行した場合:



Shin Yoshizawa: shin@riken.jp

### 正解例2

✓ 32x32のブロックで実行した場合:



✓ Ex08中のSeikai.zip内にStrasborug2.bmpでの正解画像が入っています.

Shin Yoshizawa: shin@riken.jp

### 演習:レポート1の内容その3

[www.riken.jp/briect/Yoshizawa/Lectures/index.html](http://www.riken.jp/briect/Yoshizawa/Lectures/index.html)  
[www.riken.jp/briect/Yoshizawa/Lectures/Ex09.zip](http://www.riken.jp/briect/Yoshizawa/Lectures/Ex09.zip)

- ✓ Q9: 実行し違いを考察 Lec16.pdfの演習15-1.
- ✓ Q10: 高域強調フィルタを実行するプログラムを作成し提出せよ、実行結果も載せる事(Lec16.pdfの演習17-3).

Shin Yoshizawa: shin@riken.jp

## 高域強調フィルタ

✓ ハイパスフィルタから作る事が出来る.

$$H_{h-emph}(u, v) = 1 + kH_{high}(u, v)$$

**エッジ強調!**

OGS-ARTS協会

(a) 入力画像 (b) 高域強調 (r=1) (c) 高域強調 (r=3)

(a) のフーリエスペクトル (b) のフーリエスペクトル (c) のフーリエスペクトル

Shin Yoshizawa: shin@riken.jp

## 演習17-3: 高域強調フィルタ

✓ 高域強調フィルタを作ってみよう!

✓ ヒント:

- Gaussianフィルタ(Ex09.zip中testGauss\_FDCT.cxx)を改造する.
- フィルタのカーネルは周波数領域で

$$H_{h-emph}(u, v) = 1 + kH_{high}(u, v) = 1 + k(1 - H_{low}(u, v))$$

- ここでローパスフィルタのカーネル  $H_{low}(u, v)$  はプログラム内ではgauss->img[i][j]というガウス関数の画像を用いる.
- フィルタ後は正規化なし(輝度値が0以下は0、255以上は255).
- ローパスフィルタのガウス関数のパラメータ(標準偏差)と上の強調フィルタのパラメータk二つのパラメータをプログラムの引数で与える.
- testGauss\_FDCT.cxxは  $H_{low}(u, v)$  だけのフィルタ.

Shin Yoshizawa: shin@riken.jp

## レポート5のヒント

Shin Yoshizawa: shin@riken.jp

## 演習: レポート2の内容

[www.riken.jp/brict/Yoshizawa/Lectures/index.html](http://www.riken.jp/brict/Yoshizawa/Lectures/index.html)  
[www.riken.jp/brict/Yoshizawa/Lectures/Ex10.zip](http://www.riken.jp/brict/Yoshizawa/Lectures/Ex10.zip)  
[www.riken.jp/brict/Yoshizawa/Lectures/Ex11.zip](http://www.riken.jp/brict/Yoshizawa/Lectures/Ex11.zip)  
[www.riken.jp/brict/Yoshizawa/Lectures/Ex12.zip](http://www.riken.jp/brict/Yoshizawa/Lectures/Ex12.zip)

✓ Q1: 平均化フィルタを作成(Lec17.pdfの演習17-1).  
 ✓ Q2: Gaussianフィルタを作成(Lec17.pdfの演習17-2).  
 ✓ Q3: 勾配強度計算を作成(Lec18.pdfの演習18-1).  
 ✓ Q4: 拡散方程式の平滑化を作成(Lec18.pdfの演習18-2).  
 ✓ Q5: Bilateralフィルタを作成(Lec19.pdfの演習19-2).

Shin Yoshizawa: shin@riken.jp

## 演習17-1

✓ 平均化フィルタの作成: Ex10.zip内のAverageFilter.cxxの中にあるコメントに従って平均化フィルタを作成しましょう.  
 ✓ カーネル画像(局所Window)のサイズをr=1, 5, 10で実行してみましょう.

**カーネル画像(局所Window)のサイズ: (2r+1)x(2r+1)**

11x11: r=5      21x21: r=10      61x61: r=30

Shin Yoshizawa: shin@riken.jp

## 演習17-2

✓ 正規化Gaussianフィルタの作成: Ex10.zip内のGaussianFilter.cxxの中にあるコメントに従って正規化Gaussianフィルタを作成しましょう.  
 ✓ パラメータをr=5, sigma=2.5、及びr=10, sigma=5.0で実行してみましょう.

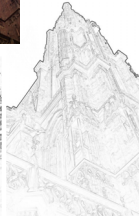
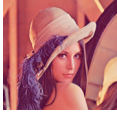
**ガウス関数の標準偏差パラメータ: sigma**

21x21: r=10, sigma=5.0      41x41: r=20, sigma=10.0      121x121: r=60, sigma=30.0

## 演習18-1



- ✓ エッジ強度画像の作成: Ex11.zip内のEdgeMagnitude.cxxの中にあるコメントに従ってソーベル作用素を使った勾配強度画像を生成するプログラムを作成しましょう。



## 演習18-2



- ✓ 拡散方程式による平滑化フィルタの作成: Ex11.zip内のDiffusion.cxxの中にあるコメントに従って拡散方程式による平滑化フィルタを作成しましょう。
- ✓ 微小時間 $\epsilon=0.25$ で繰り返し10,20,30,40,50,100で実行してみましょう。

繰り返し5回

繰り返し10回

繰り返し50回

繰り返し100回



繰り返し5回

10回

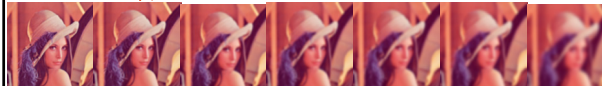
20回

30回

40回

50回

100回



## 演習19-2



- ✓ Bilateralフィルタの作成: Ex12.zip内ColorBilateralFilter.cxxの中にあるコメントに従ってフィルタを作成しましょう。
- ✓ Spatial Kernelの標準偏差 $\sigma=10.0$ 、25.0、Intensity Kernelの標準偏差 $h=1.0 \times$  (画像全体のグレースケールの標準偏差)を0.1、0.2、0.5と、畳み込み半径10で実行してみましょう(6種類)。

 $\sigma=10.0$   $h=0.15$  $\sigma=10.0$   $h=0.5$ 