フーリエ変換 (p. 77)

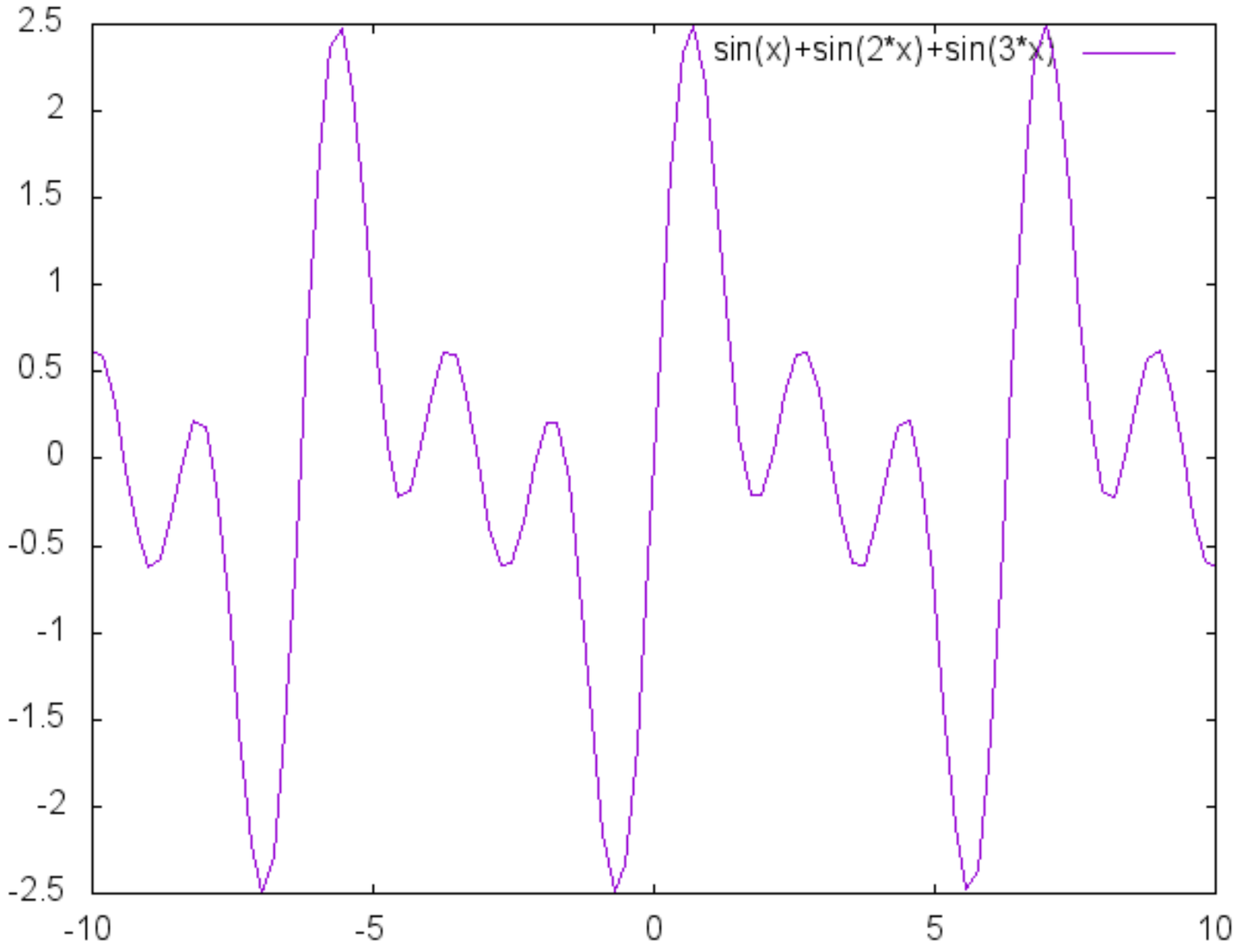
森本，塚田，澤野

本日の目標

- フーリエ変換
  - 特徴抽出・画像圧縮・ノイズ軽減に利用される

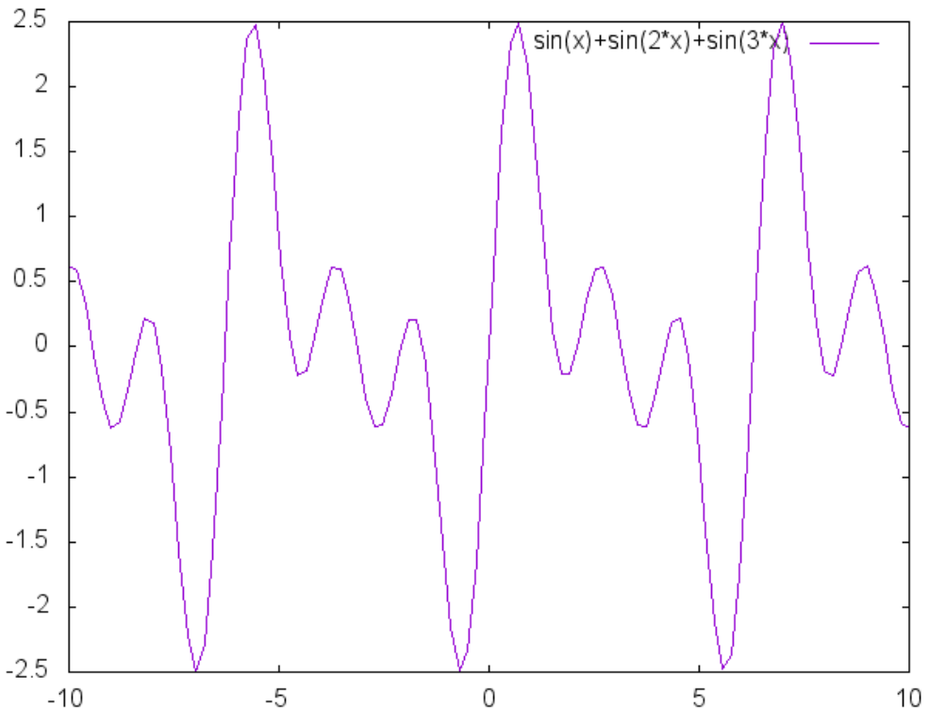
画像や音の性質

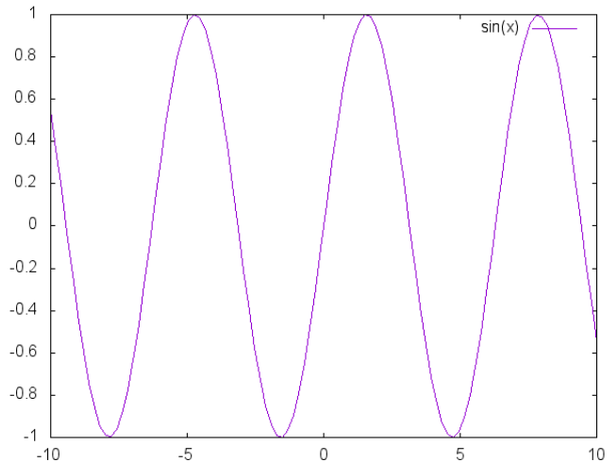
- 波の集合で表現できる



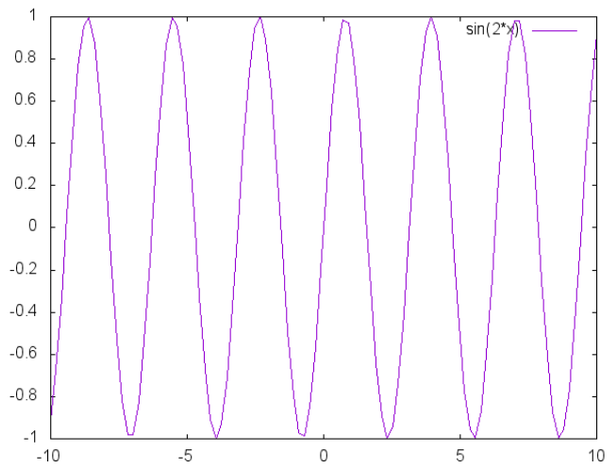
音声波形の例

波の分離・合成

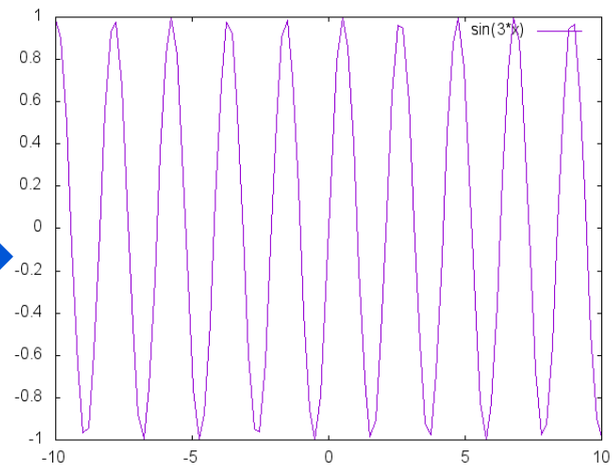




sin(x)

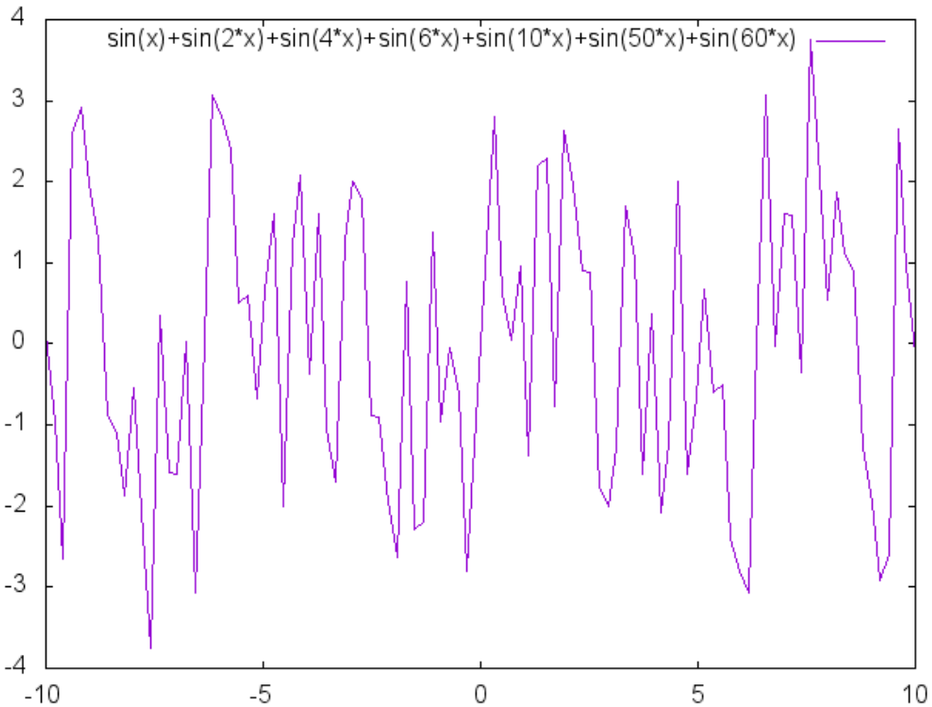


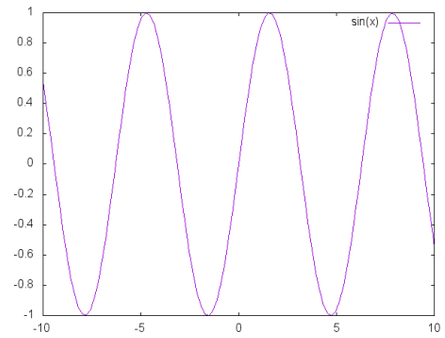
sin(2x)



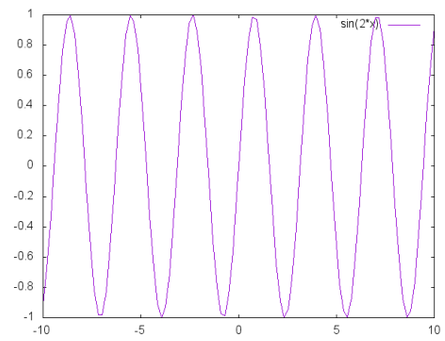
sin(3x)

複雑な音声でも

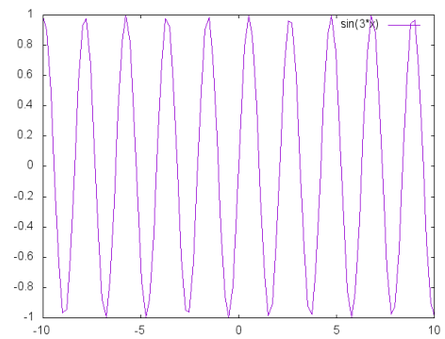




sin(x)



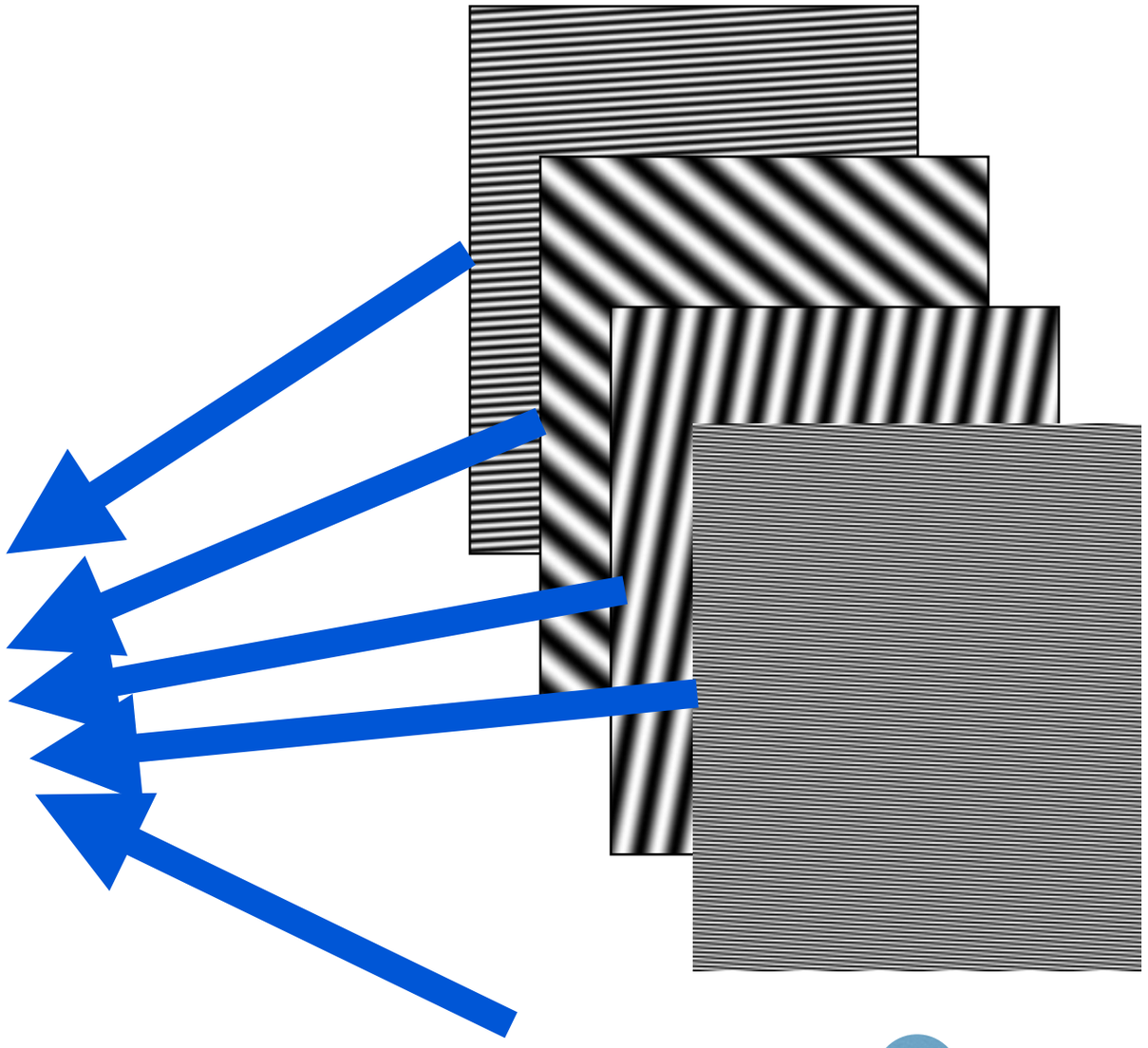
sin(2x)



sin(3x)

一次元周波数成分

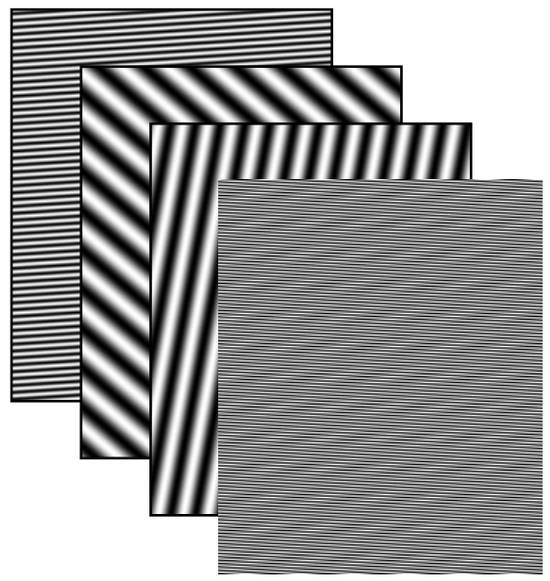
画像も同じ (縞の合成)



フーリエ変換とは

- 音声や画像を
  「**周波数領域内の表現に変換**」すること

フーリエ変換



フーリエ**逆**変換

画像　　周波数領域内の表現

一次元フーリエ変換 (p. 80)

- 関数 $f(t)$ のフーリエ変換

$$F(w) = \int_{-\infty}^{\infty} f(t)e^{-jwt}dt$$

  - $f(t)$: 時刻tにおける入力f
  - j: 虚数単位 $(j^2 = -1)$
  - $w\left(= 2\pi\frac{1}{T}\right)$: 角周波数，$T$: 周期

- フーリエ逆変換 (逆フーリエ変換)

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(w)e^{jwt}dw$$

オイラーの公式

- 自然対数の底 (ネイピア数): e
  - e^x もしくは exp(x) で計算される
  - ネイピア数 e = 2.71828…
- オイラーの公式

$$e^{j\theta} = \underbrace{\cos\theta}_{\text{実部}} + \underbrace{j\sin\theta}_{\text{虚部}}$$

  - j: 虚数単位
  - オイラーの等式 (数学における最も美しい定理)

$$e^{j\pi} = -1$$

二次元フーリエ変換

- フーリエ変換

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

- フーリエ逆変換

$$f(x,y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}F(u,v)e^{j2\pi(ux+vy)}\,du\,dv$$

フーリエ変換を利用した画像特徴

- 画像特徴には以下が利用される
  - 絶対値 (振幅スペクトル)

$$\left|F(u,v)\right| = \sqrt{\mathrm{Re}\left\{F(u,v)\right\}^2 + \mathrm{Im}\left\{F(u,v)\right\}^2}$$

  - 偏角 (位相スペクトル)

$$\arg\left\{F(u,v)\right\} = \tan^{-1}\frac{\mathrm{Re}\left\{F(u,v)\right\}}{\mathrm{Im}\left\{F(u,v)\right\}}$$

    - Re: 実部，Im: 虚部

離散フーリエ変換

- 連続空間

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

- 離散空間

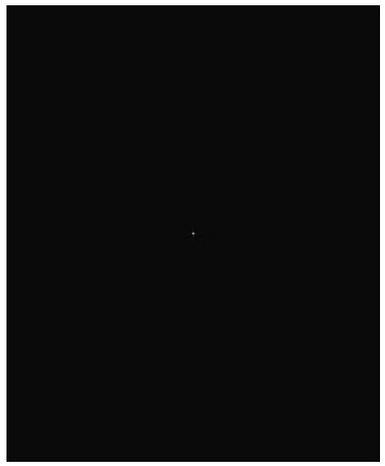
$$F(u,v) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\mathrm{e}^{-2j\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)}$$

考え方

実数領域

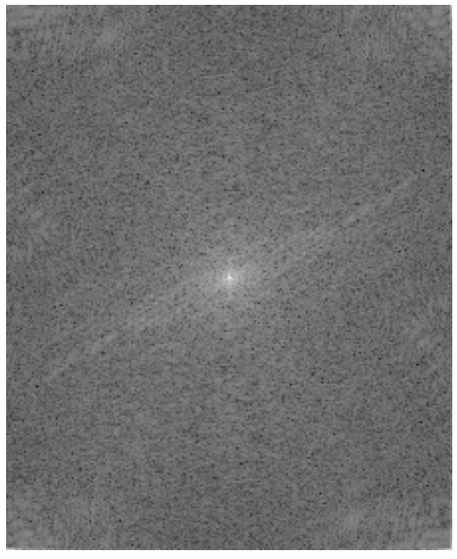
フーリエ変換

逆フーリエ変換

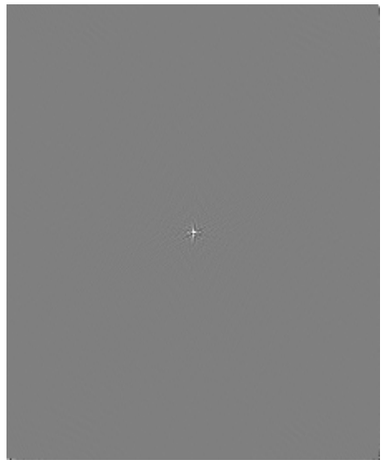


実数領域

二乗和



パワースペクトル画像

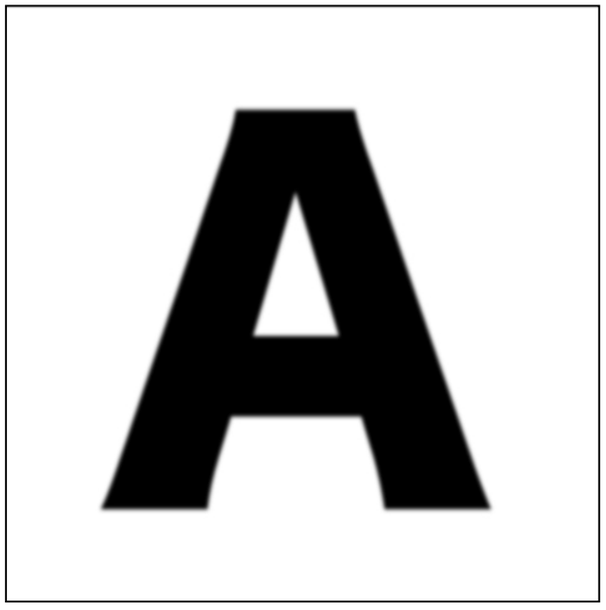


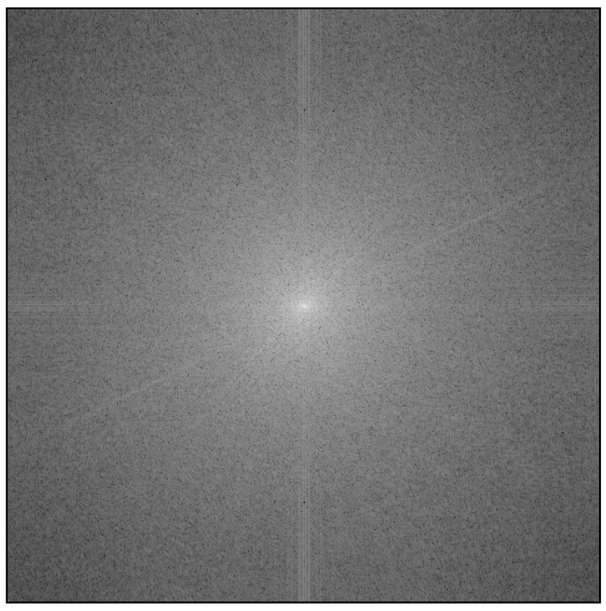
虚数領域

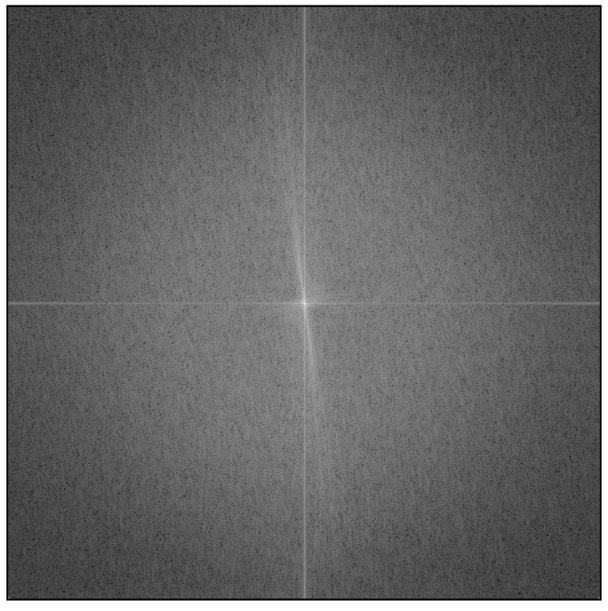
空間領域

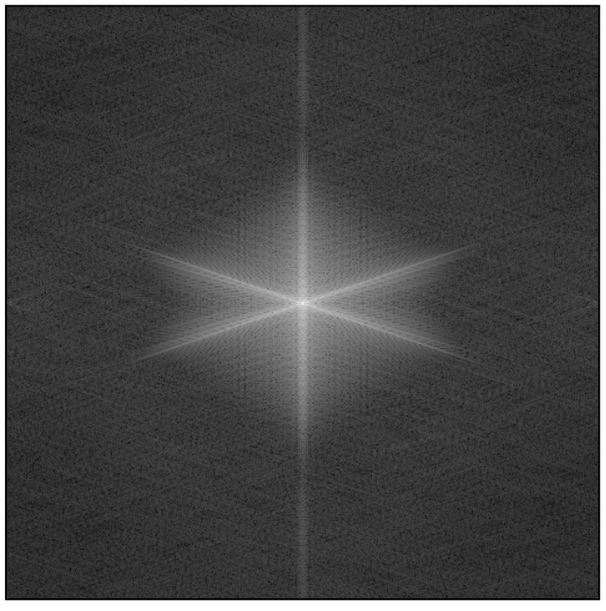
周波数領域

フーリエ変換の例









デモ

dftDemo.zipをダウンロードしてください

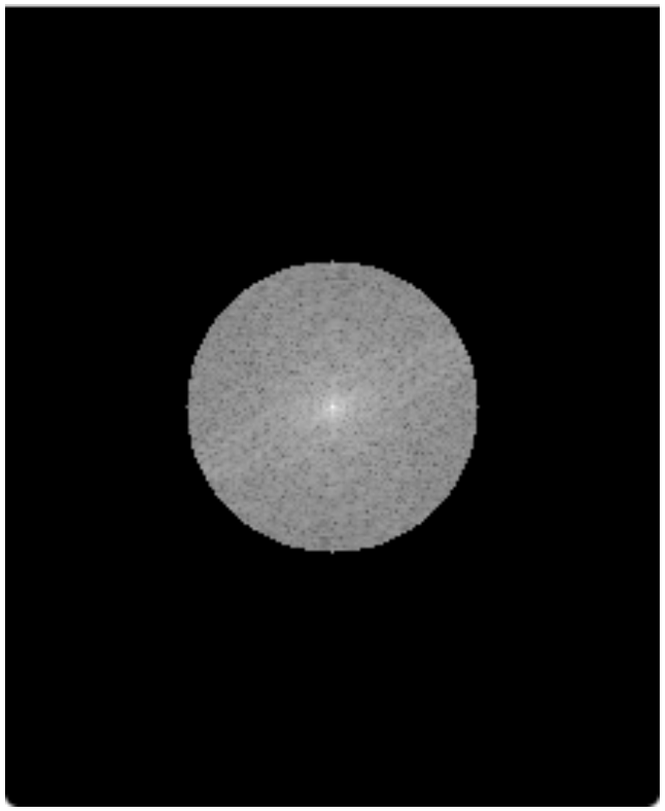
OpenCVの離散フーリエ変換の関数

```
void dft(const Mat& src, Mat& dst, int flags=0)
```

- src: 入力画像
- dst: 出力画像
- flags: 変換フラグ (省略の場合は0)
  - 0 (記載なしの場合): 離散フーリエ変換
  - DFT_INVERSE: 離散　フーリエ逆変換

ローパスフィルタ

- 低周波成分を残し，高周波成分の除去
  - 低周波成分: 色の変化があまりない領域
  - 高周波成分: 濃度値が急激に変化している領域

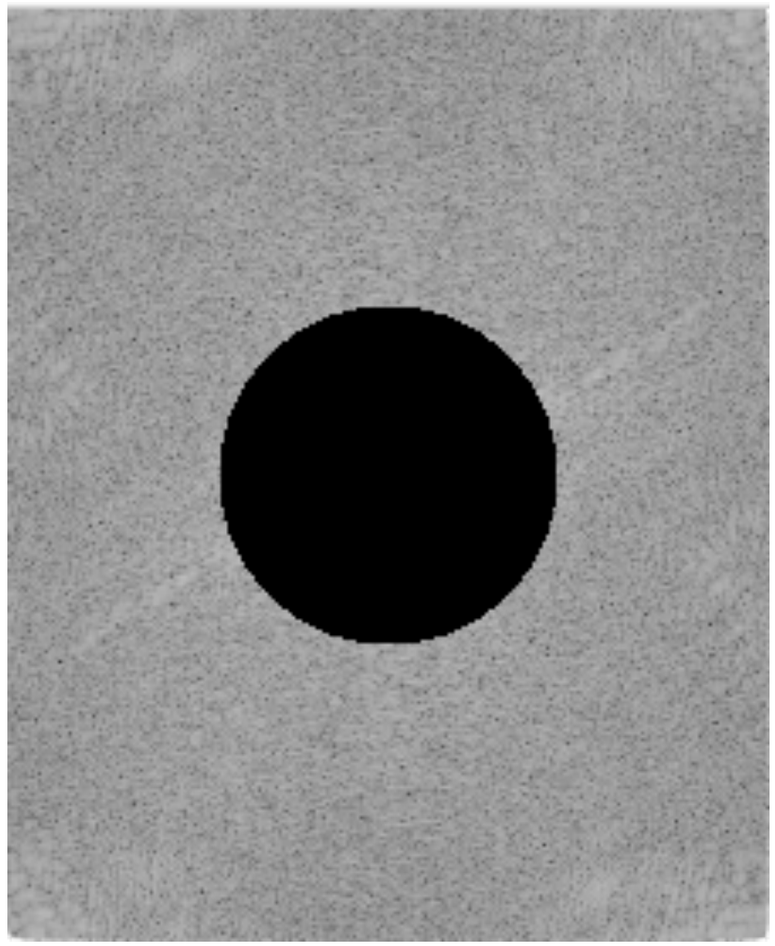


ローパスフィルタ

出力画像

ハイパスフィルタ

- 高周波成分を残し，低周波成分の除去

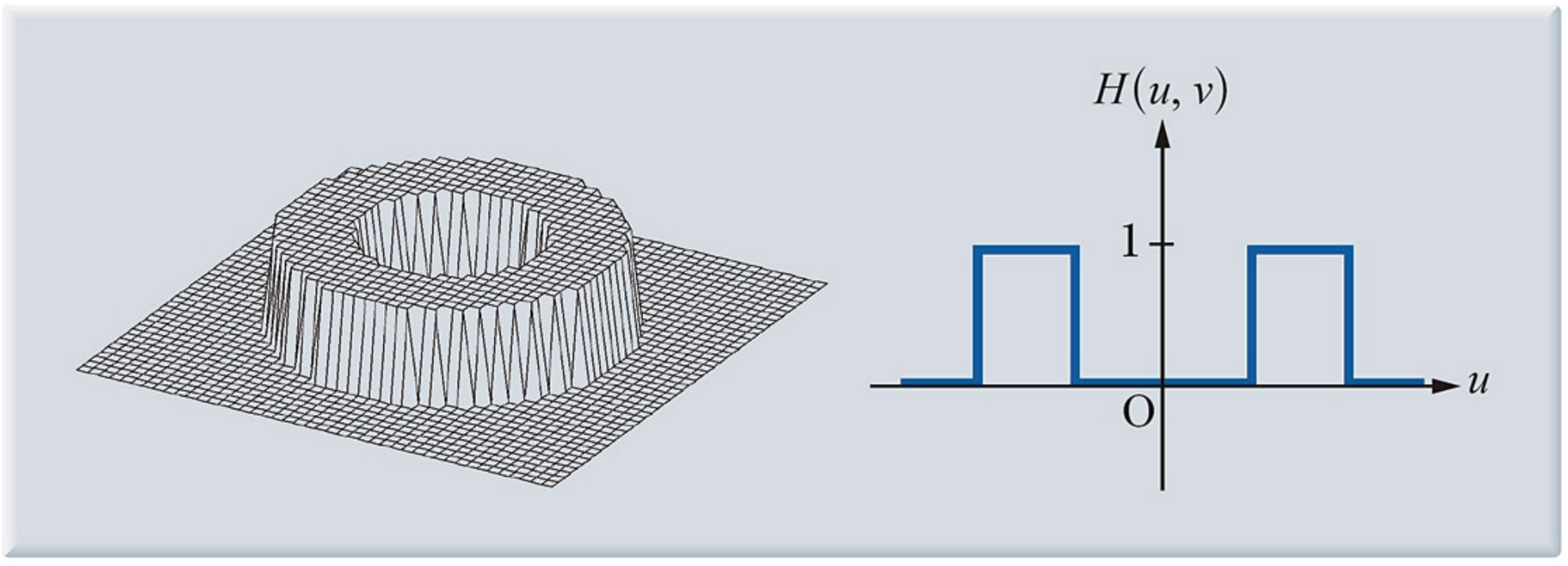


ハイパスフィルタ

出力画像

バンドパスフィルタ

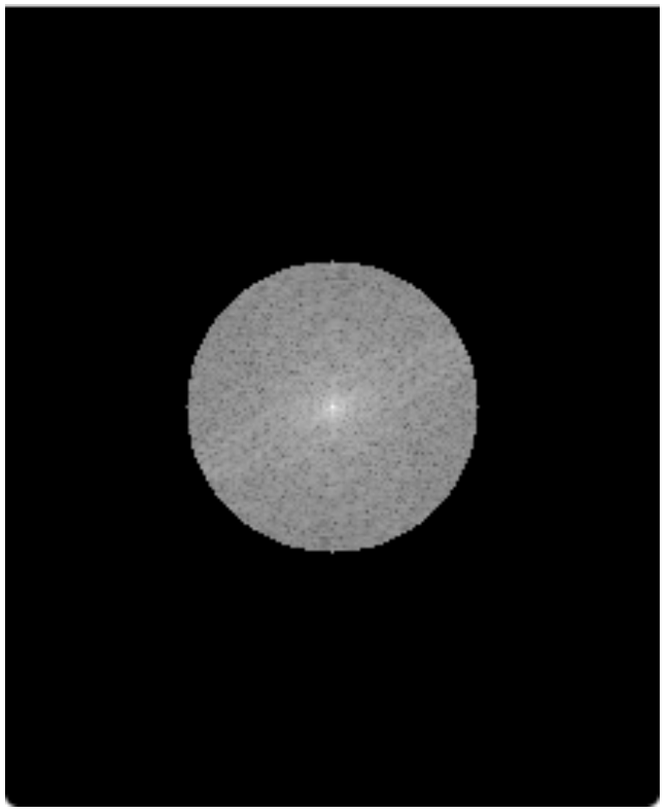
■図 6.16——バンドパスフィルタ



「ディジタル画像処理　改訂新版」2015年 ／公益財団法人 画像情報教育振興協会(CG-ARTS協会)

演習

- サンプルファイル (dft4student)のダウンロード
- ローパスフィルタの実現



ローパスフィルタ

出力画像

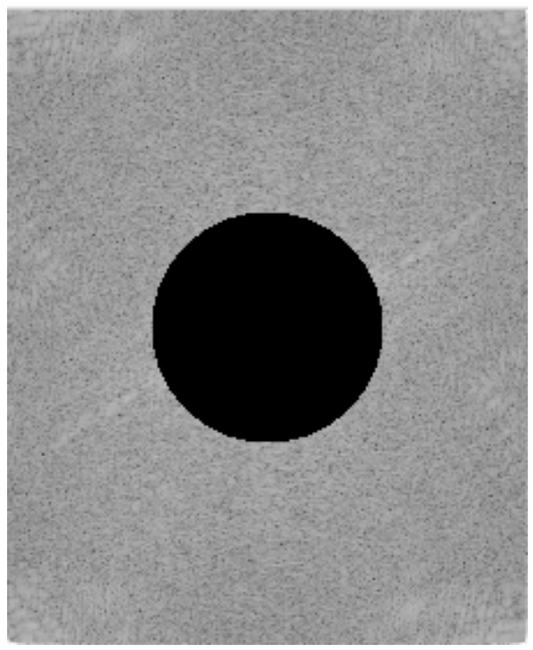
```
cv::Vec2d s; //色値
s[0] = s[1] = 0.0;
int cx = complexFT_mat.cols / 2;
int cy = complexFT_mat.rows / 2;

for (int y=0; y<complexFT_mat.rows; y++) {
    for (int x=0; x<complexFT_mat.cols; x++) {

        //距離の計算
        double dist = sqrt((cx-x)*(cx-x) + (cy-y)*(cy-y));
        if (dist > 50) { //ローパスフィルタ
            complexFT_mat.at<cv::Vec2d>(y, x) = s;
        }
    }
}
```

課題1

- ハイパスフィルタを実現せよ
- 距離は51以上で設定せよ
- 提出ファイル名:
  - 13_01_学籍番号.cpp
  - 13_01_学籍番号.jpg (tiffでもいい)

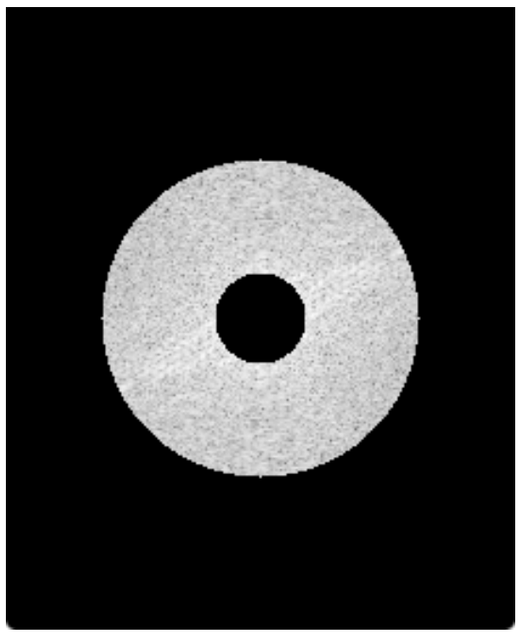


ハイパスフィルタ

出力画像

課題2

- バンドパスフィルタを実現せよ
- 21以上，70未満で実現せよ
- 提出ファイル名:
  - 13_02_学籍番号.cpp
  - 13_02_学籍番号.jpg (tiffでもいい)

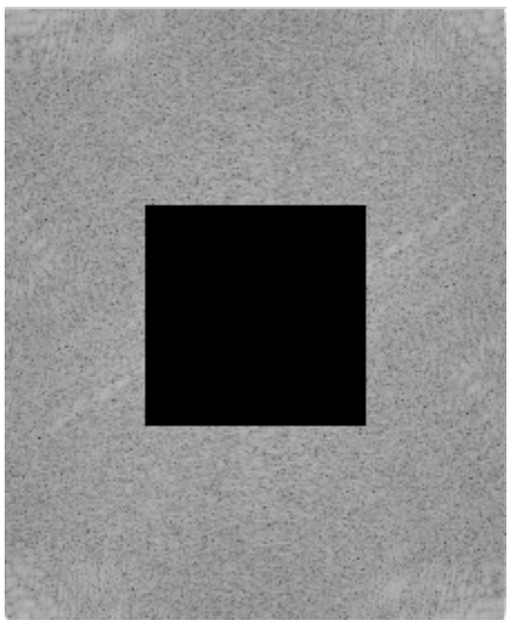


バンドパスフィルタ

出力画像

課題3

- 矩形のハイパスフィルタを実現せよ
  - 縦と横が100画素の正方形とする
- 提出ファイル名:
  - 13_03_学籍番号.cpp
  - 13_03_学籍番号.jpg (tiffでもいい)



ハイパスフィルタ

出力画像

チャレンジ課題

- 離散フーリエ変換を実装せよ
  - cv::dftを自作せよ
  - 演習で実施した他の関数はすべて使用して，cv::dftの部分だけを自作関数に差し替えること
- 提出ファイル名:
  - 13_challenge_学籍番号.cpp