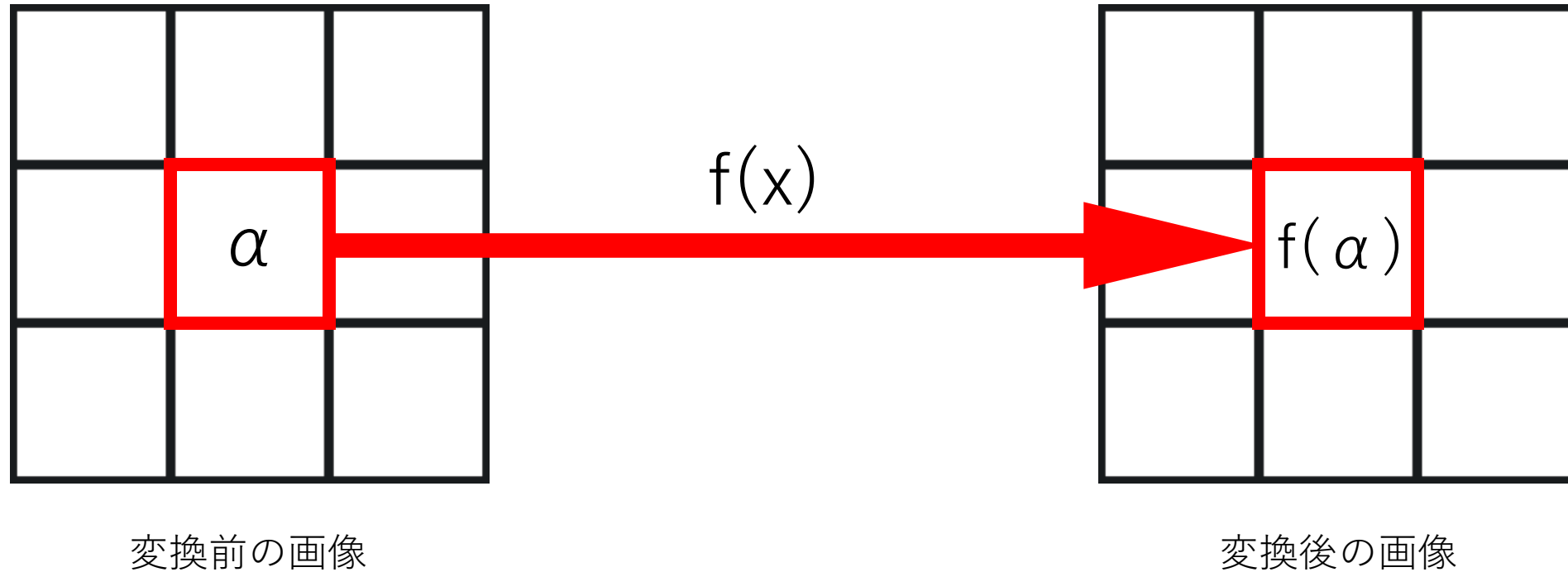


# メディアコンピューティング

第6回

# 復習

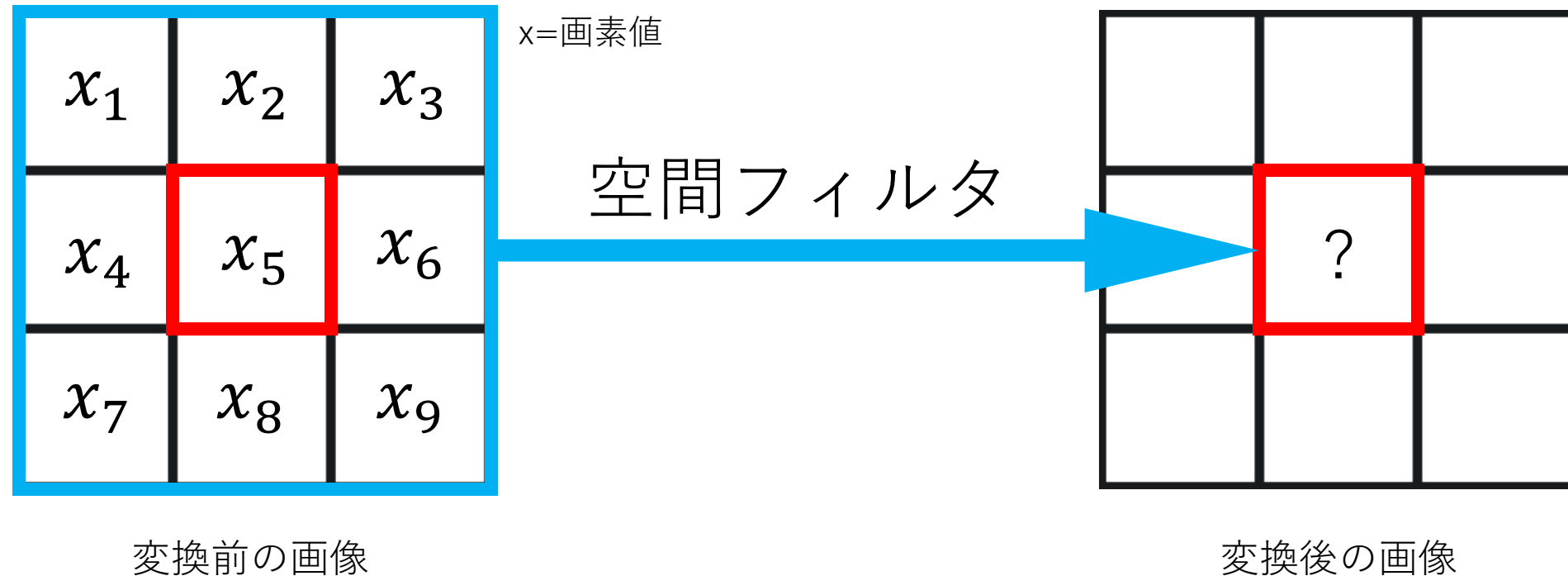
これまでは画素を任意の関数で画像(画素値)を変換していた



# 空間フィルタリング

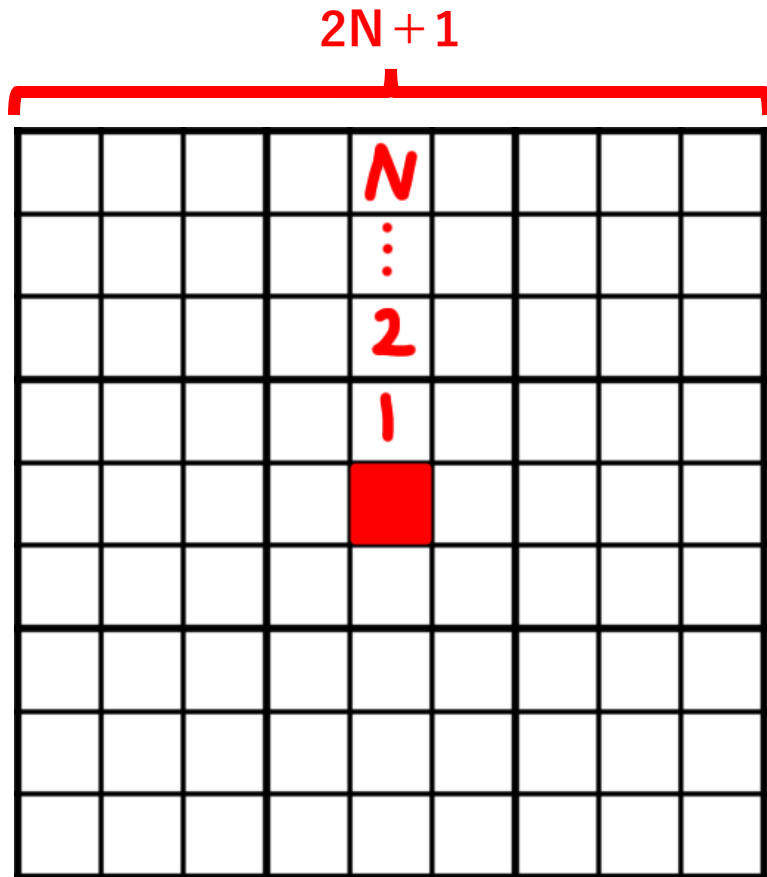
空間フィルタリングでは注目画素の周辺の画素も計算に入れて画像を変換

この時、計算領域や計算方法を指定するものを空間フィルタと呼ぶ



# 空間フィルタの掛け方①

空間フィルタを掛けるにはまずフィルタのサイズ決める



- フィルタは注目画素を中心とした正方形に適応
- 中心画素から辺までの距離を $N$ とすると一辺は $2N+1$
- つまり辺の長さは奇数で設定しなくてはならない

# 空間フィルタの掛け方②

空間フィルタの内，線形フィルタでは係数を指定する

$h_1$	$h_2$	$h_3$
$h_4$	$h_5$	$h_6$
$h_7$	$h_8$	$h_9$

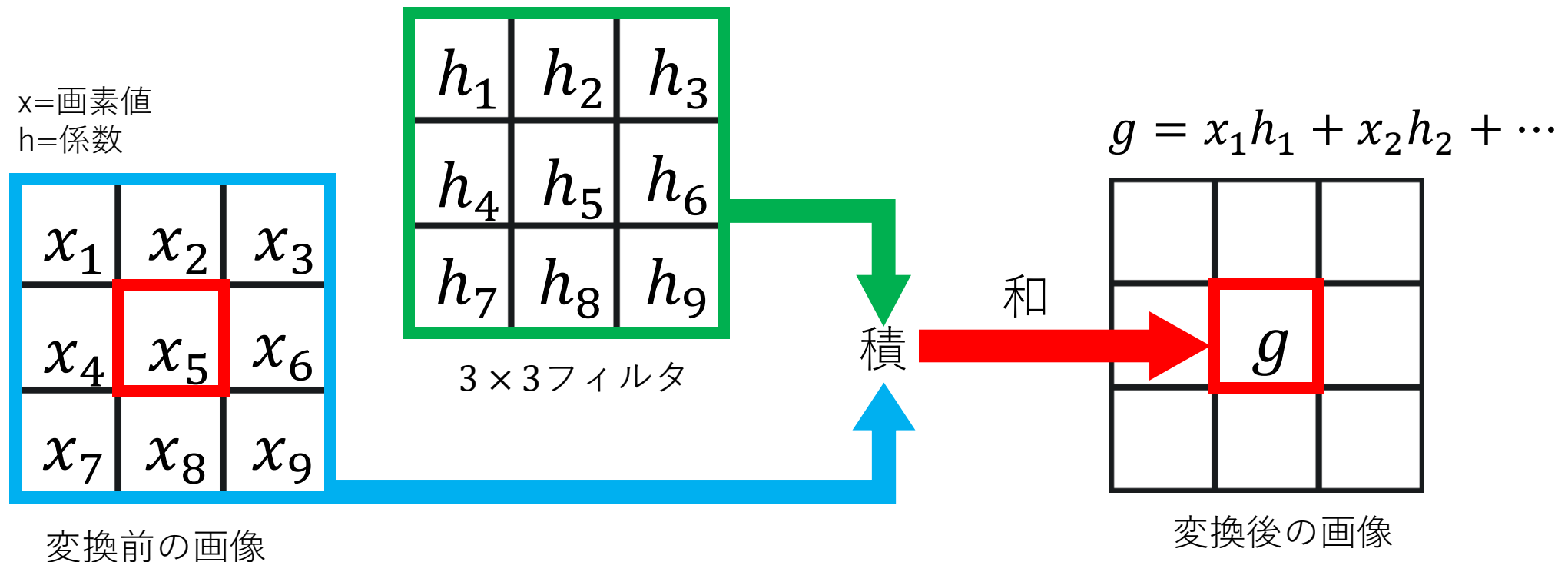
3×3フィルタ

- 各画素に任意の係数(h)を指定
- 係数の決め方で効果，名称が変わる
- 係数は2次元正方行列

# 空間フィルタの掛け方③

元画像の注目画素を基準に対応する画素値と空間フィルタの係数の積を取り  
その合計を変換後の画素値とする

また、この手順を踏まない空間フィルタを非線形フィルタと呼ぶ



線形フィルタ

# 平均化フィルタ

領域内の画素値の平均値を求める

3 x 3

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

より強い  
効果

 $\angle$  $6 \times 6$ [illegible]



# 重み付き平均化

## 加重平均化フィルタ

フィルタの原点に近いほど大きな重みをつけたフィルタ

1/16	2/16	1/16
2/16	4/16	2/16
1/16	2/16	1/16

3x3画素

## ガウシアンフィルタ

重みをガウス分布に近づけたもの

1/256	4/256	6/256	4/256	1/256
4/256	16/256	24/256	16/256	4/256
6/256	24/256	36/256	24/256	6/256
4/256	16/256	24/256	16/256	4/256
1/256	4/256	6/256	4/256	1/256

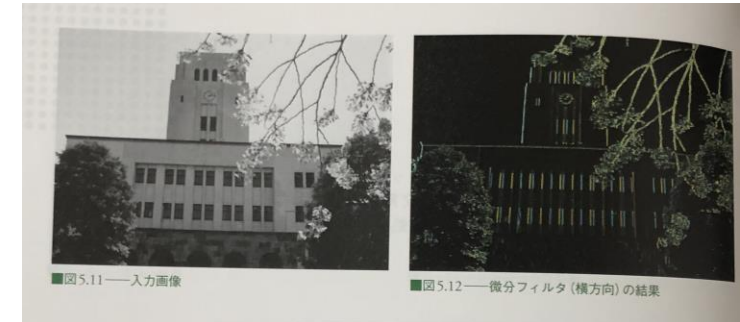
5x5画素

# エッジ抽出

画像中で明るさが急に変化するエッジ部分を取り出す方法

## 微分フィルタ

注目画素と隣接画素の濃度値の差分（微分を出力する）



縦方向

0	0	0
-1	1	0
0	0	0

左隣との差

0	0	0
0	-1	1
0	0	0

右隣との差

0	0	0
-1	0	1
0	0	0

左右差分の平均

0	1	0
0	-1	0
0	0	0

上との差

0	0	0
0	1	0
0	-1	0

下との差

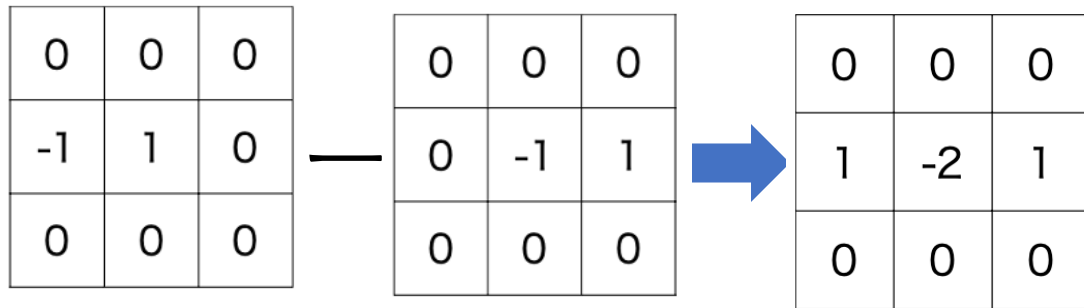
0	1	0
0	0	0
0	-1	0

上下差分の平均

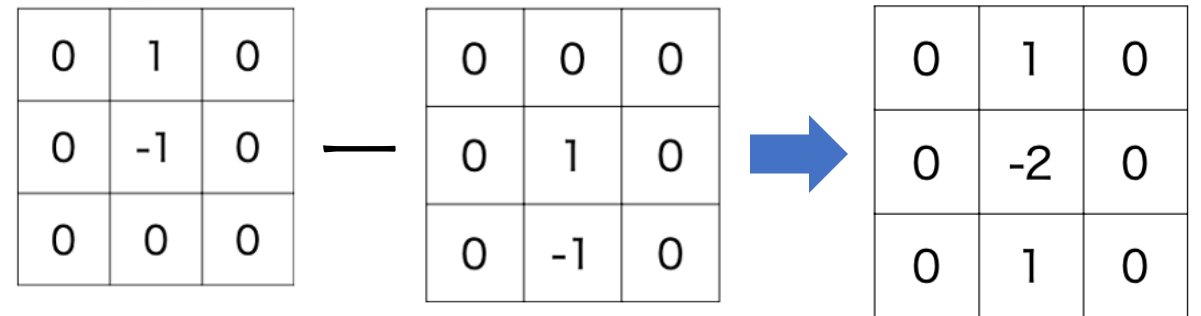
# 2次微分フィルター

フィルターの出力差を求めたもの

横方向の2次微分フィルター



縦方向の2次微分フィルター



# ラプラシアンフィルタ

方向に依存しないエッジ検出

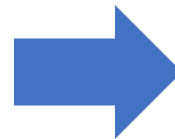
2次微分(縦)

0	0	0
1	-2	1
0	0	0

+

2次微分(横)

0	1	0
0	-2	0
0	1	0



0	1	0
1	-4	1
0	1	0



# 演習

前述の 4 つのフィルタをかけてみよう

- filter.cppをダウンロード
- サンプル画像は変えてもいいよ

# プログラム解説

線形フィルタ生成

```
float Fave[]={  
    1.0/9.0, 1.0/9.0, 1.0/9.0,  
    1.0/9.0, 1.0/9.0, 1.0/9.0,  
    1.0/9.0, 1.0/9.0, 1.0/9.0,  
};  
cv::Mat kernel(cv::Size(3,3),CV_32F,Fave);
```

cv::Mat インスタンス名(フィルタサイズ, フィルタの型, 1次元配列)

線形フィルタ適用関数

```
cv::filter2D(grayImage,aveImage,grayImage.depth(),kernel);
```

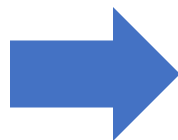
cv::filter2D(入力画像, 出力画像, 出力画像のビット深度, 線形フィルタ)

その他のフィルタ

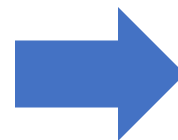
# メディアンフィルタ

- 非線形フィルタ
- `cv::medianBlur`
- 領域内の中央値を出力とする

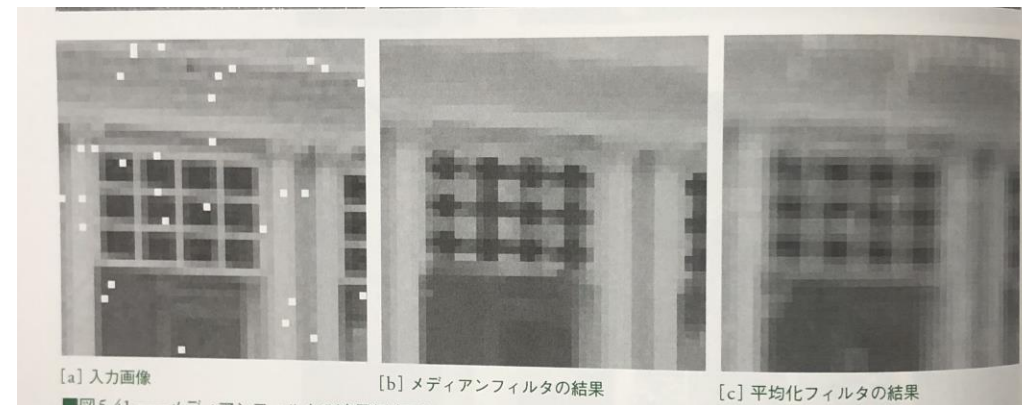
1	7	2
2	9	8
5	1	4



9
8
7
5
4
2
2
1
1



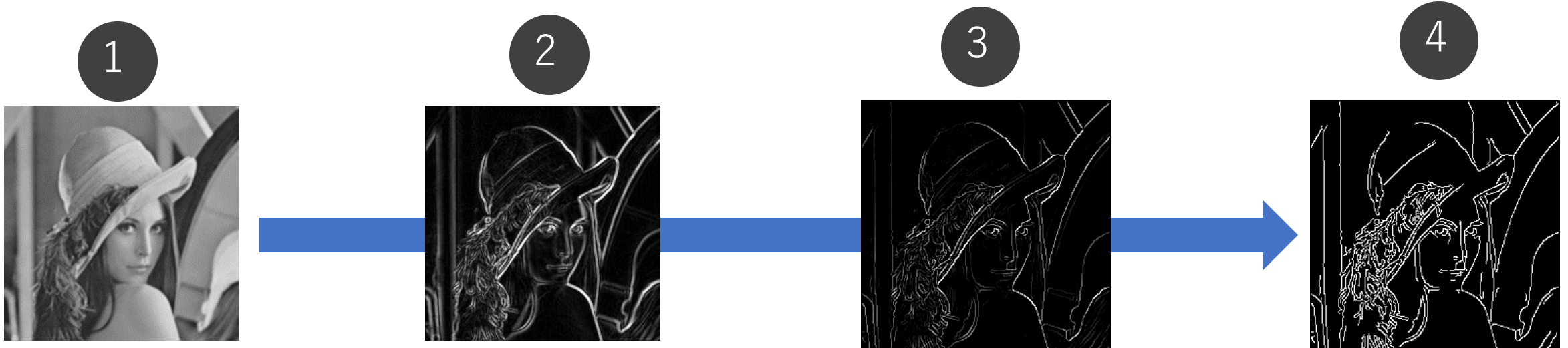
1	7	2
2	4	8
5	1	4





# CANNY法によるエッジ抽出

- cv::Canny
  1. 平滑化
  2. 輪郭抽出
  3. 非極大抑制
  4. ヒステリシス閾値処理



# 課題

- 動画に対しフィルターをかけてみよう
  - 動画は動画ファイルでもカメラ入力でも良い，お好みで
  - 2パターン以上は作る（ウィンドウ名は空間フィルタ名にしてね）
  - 一つはエッジ抽出とする
- 
- 元動画と変換後動画を並べた様子を10秒程度の動画で撮影し提出