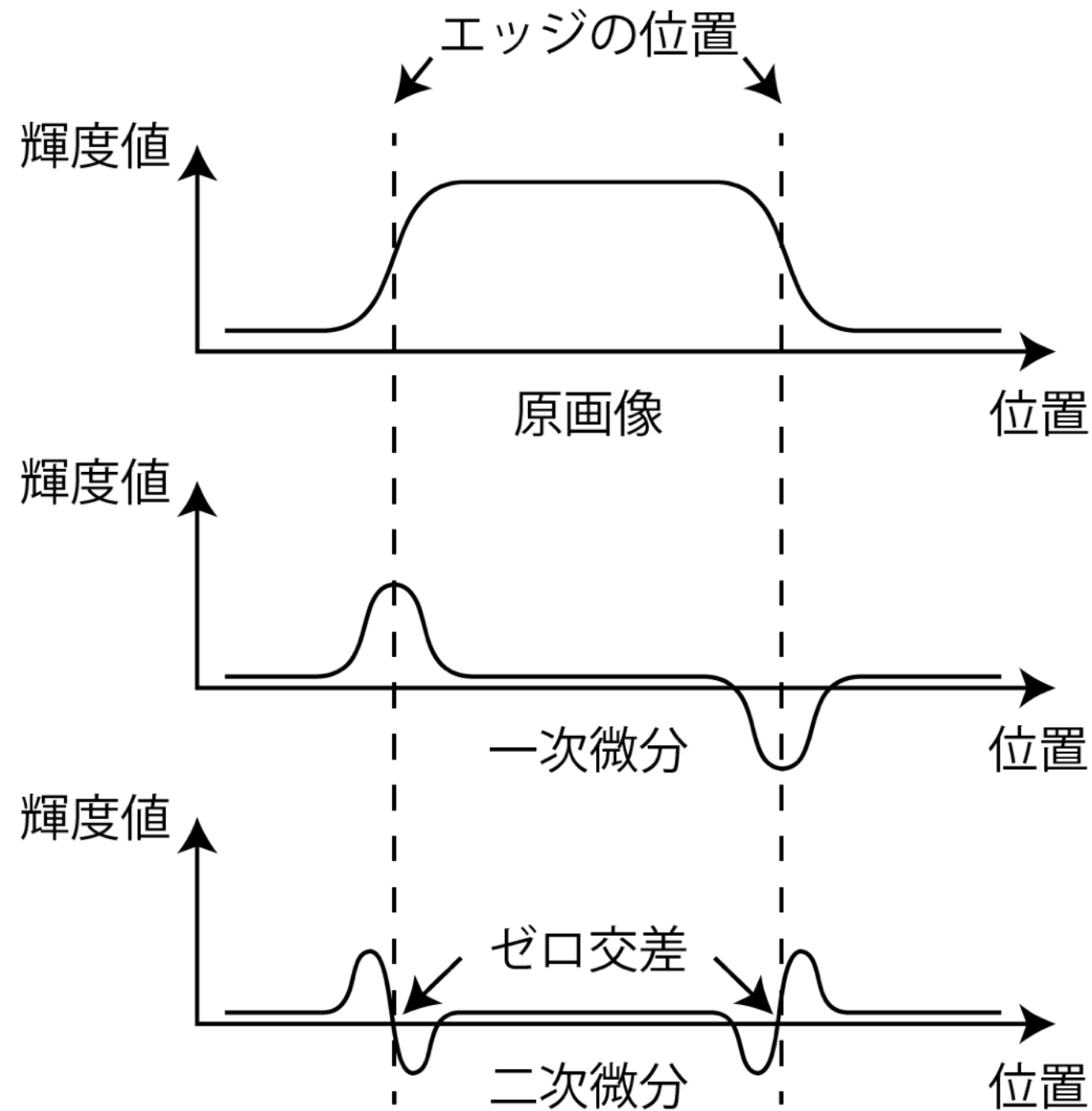
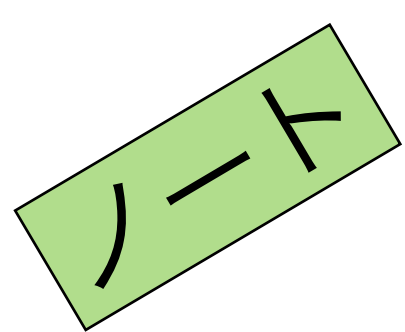


二次微分によるエッジ検出



ゼロ交差：値が＋から－へ変化する位置
(あるいは逆)



二次微分フィルタ

(教科書p.54)

- 微分（差分）フィルタの差分

0	0	0
0	-1	1
0	0	0

微分フィルタ

−

0	0	0
-1	1	0
0	0	0

微分フィルタ

=

0	0	0
1	-2	1
0	0	0

二次微分フィルタ
(横方向)

二次微分フィルタ
(縦方向)

0	1	0
0	-2	0
0	1	0

ノート

ラプラシアンフィルタ

- 関数 $f(x, y)$ のラプラシアンの定義 (教科書p.55)

$$\frac{\partial^2}{\partial x^2} f(x, y) + \frac{\partial^2}{\partial y^2} f(x, y)$$

0	0	0
1	-2	1
0	0	0

 +

0	1	0
0	-2	0
0	1	0

 =

0	1	0
1	-4	1
0	1	0

ラプラシアンフィルタ

方向に依存しないエッジが得られる

ラプラシアンフィルタ処理結果例



ソーベルフィルタ
(横) 処理後の画像



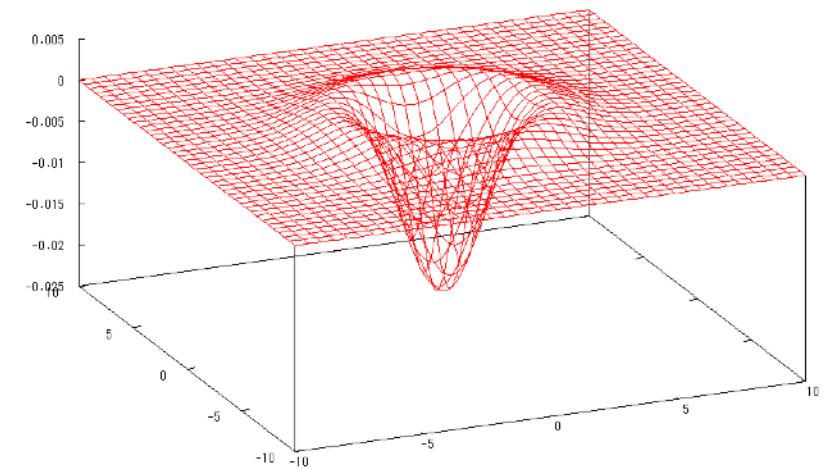
ラプラシアンフィルタ
処理後の画像

補足：二次微分フィルタの欠点

- ノイズに弱い（微分を繰り返すため）

➡ 平滑化してから二次微分を実施 する改善手法

- ガウシアンフィルタ＋ラプラシアンフィルタ
=LoGフィルタ
(Laplacian of Gaussian)



例

0	1	0
1	4	1
0	1	0

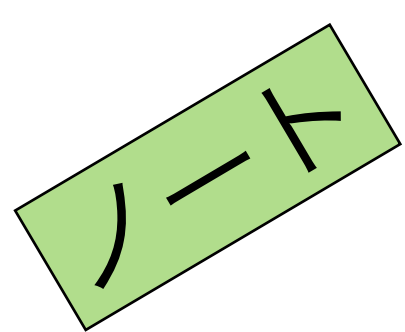
⊗

畳み込み

0	1	0
1	-4	1
0	1	0

=

		1		
	2	0	2	
1	0	-12	0	1
	2	0	2	
		1		



Cannyオペレータ

(教科書p.56)

- 現在エッジ検出でよく用いられる手法
 - ガウシアンフィルタによる平滑化
 - 一次微分により勾配を求める
 - 勾配の最大値を検出し細線化
 - 2つのしきい値を用いてエッジを判定

メモ

Cannyオペレータ関数

```
cv::Canny (gray_img, edge_img, double th1, double th2);
```

- gray_img: 入力画像 edge_img: 出力画像
- th1, th2: 勾配の大きさに対するしきい値
($th1 < th2$ とする)
 - ➡ th2: エッジ初期検出のためのしきい値
 - ➡ th1: このしきい値以上で、エッジに接続されている画素はエッジと見なす

演習: Cannyオペレータ

- プロジェクト名: canny
- 入力画像: apple_grayscale.jpg
- OpenCVのcannyオペレータを利用
- しきい値 : $th1=100$, $th2=200$
 - 通常 $th2$ は $th1$ の2～3倍

medianプログラムを参考に
canny.cppを作成しましょう

演習中 . . .



cannyプログラムのポイント

//2. 画像変数の宣言

```
cv::Mat dst_img; //結果画像
```

//3. Cannyオペレータ

```
cv::Canny(src_img, dst_img, 100, 200);
```

convertScaleAbsは不要 (canny内部で処理)

処理結果



入力画像



出力画像

(th1=100/th2=200)

しきい値を変更してみよう