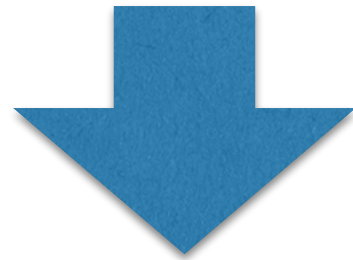


# エッジ検出処理における注意事項

- 微分処理では以下の点に留意する必要がある
  - 負の値も出現する
  - 値の変化が小さいとわかりにくい



- **表示のための**解決案
  - 絶対値を取る
  - 適切な倍率で値を強調する
  - 二値化（後日の講義）

# 演習：微分フィルタ (1/3)

- filter2Dプロジェクトを使って  
微分フィルタを演習しましょう
  - 先週作成したfilter2Dプロジェクトを修正します
  - 修正箇所は以下の通り
    - ◎ フィルタの値（微分オペレータにする）
    - ◎ フィルタの型 倍精度浮動小数点（CV\_64F）にする
    - ◎ 絶対値および適当な倍率での強調を行う

# 演習：微分フィルタ (2/3)

- まずは 微分フィルタ (横方向)

//3. フィルタの宣言と設計 (入力と正規化)

//フィルタa

```
double filter_a[] = { 0., 0., 0.,  
                     -1., 0., 1.,  
                     0., 0., 0.};
```

微分フィルタ  
(横方向)

修正

////配列をフィルタ行列kernelに変換

```
cv::Mat kernel = cv::Mat(3, 3, CV_64F, filter_a);
```

//正規化 (正規化しないと画素値が255を超えるため)

```
// cv::normalize(kernel, kernel, 1.0, 0.0,  
cv::NORM_L1);
```



コメントアウト (後で倍率変換するので)

# 演習：微分オペレータ (3/3)

//4. フィルタの計算

cv::Mat tmp\_img; // 一時的に格納する画像を用意



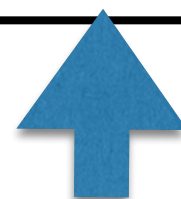
追加



修正

cv::filter2D(src\_img, tmp\_img, CV\_64F, kernel);

cv::convertScaleAbs(tmp\_img, dst\_img);



追加

注：dst\_imgは

cv::Mat dst\_img;

と定義されていること（メモリ確保しない）

演習中 . . .





# 微分フィルタ処理結果例



入力画像



微分フィルタ  
処理後の画像

メモ

# convertScaleAbs関数

- 絶対値をとり適切な倍率で値を変換する

OpenCV関数

```
cv::convertScaleAbs(入力画像, 出力画像);
```

- 8bit unsigned 出力画像  
(グレースケール画像) に変換される

# 演習：微分フィルタ（縦方向）

- 微分フィルタ（縦方向）もやってみましょう

//3. フィルタの宣言と設計（入力と正規化）

//フィルタh

```
double filter_a[] = { 0., -1., 0.,  
                     0., 0., 0.,  
                     0., 1., 0.};
```

微分フィルタ  
（縦方向）



////配列をフィルタ行列kernelに変換

```
cv::Mat kernel = cv::Mat(3, 3, CV_64F, filter_a);
```

//正規化（正規化しないと画素値が255を超えるため）

```
// cv::normalize(kernel, kernel, 1.0, 0.0,  
cv::NORM_L1);
```



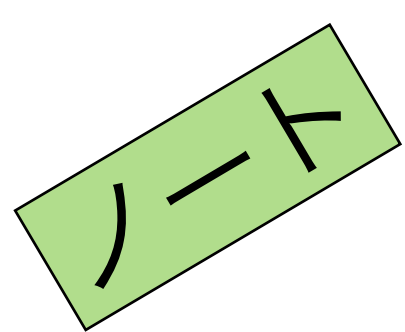
# 微分フィルタ（縦方向） 処理結果例



入力画像



微分フィルタ（縦方向）  
処理後の画像



# 画素値の勾配 (gradient)

- 横方向の差分

$$\Delta_x f(x, y) = \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x}$$

- 縦方向の差分

$$\Delta_y f(x, y) = \frac{f(x, y + \Delta y) - f(x, y)}{\Delta y}$$



- 勾配

$$(\Delta_x f(x, y), \Delta_y f(x, y))$$

- 勾配の大きさ

$$\sqrt{(\Delta_x f(x, y))^2 + (\Delta_y f(x, y))^2}$$

- 勾配方向

$$\tan^{-1} \frac{\Delta_y f(x, y)}{\Delta_x f(x, y)}$$



# 勾配の大きさ（エッジ強度）



入力画像



横・縦方向フィルタから  
勾配の大きさを求めた画像