

Introduction to OpenCV

Satoshi Murashige

October 20, 2018

Mathematical Informatics Lab., NAIST

『詳解 OpenCV』

Table of Contents

1. 基本的な画像処理

- ・ 1 章：概要
- ・ 2 章：OpenCV 入門
- ・ 10 章：フィルタとコンボリューション

2. 物体検出

- ・ 12 章：画像解析
- ・ 13 章：ヒストグラムとテンプレートマッチング
- ・ 14 章：輪郭

3. 動画解析

- ・ 15 章：背景除去
- ・ 16 章：キーポイントと記述子
- ・ 17 章：トラッキング

4. 3次元復元

- ・ 18 章：カメラモデルとキャリブレーション
- ・ 19 章：射影変換と3次元ビジョン

サンプルコードについて

- ・ テキスト中のプログラムはすべて C++ で書かれている
- ・ 今回使用するコードはそれらを Python 向けに書き直したもの
 - ・ 全部は書き直していない 😊
 - ・ なるべく Pythonic なスタイルにするのと説明の都合により機能が異なることがある

Python を使用するメリット・デメリット

- ・ Python を使用するメリット 😊
 - ・ コンパイル不要なので Try & Error がしやすい
 - ・ Python 向けの強力な開発環境の恩恵を受けられる
 - ・ Python で書かれた他のモジュールとの組み合わせが容易になる
- ・ Python を使用するデメリット 😞
 - ・ 上手に書かないとめっちゃ遅くなる
 - ・ 上手に書いても最適化された C++ の速度には及ばない
(リアルタイム性が求められるタスクでなければ最低限の最適化で OK)

- ・ サンプルコードのリポジトリをダウンロードする
`https://github.com/eqs/opencv3-book-python.git`
- ・ 画像処理用の画像をダウンロードする
 - ・ `cd`コマンドで`images`に移動してから`sh get_images.sh`を実行

1 章：概要

aaa

2 章 : OpenCV 入門

初めてのプログラム-写真を表示する

```
import cv2

img = cv2.imread('../images/color/Lenna.bmp')
assert img is not None, 'Loading image is failed.'

cv2.namedWindow('Example 2-1', cv2.WINDOW_AUTOSIZE)
cv2.imshow('Example 2-1', img)

cv2.waitKey(0)
cv2.destroyAllWindows('Example 2-1')
```

2 つ目のプログラム-動画

動画やカメラの映像に対する処理を行う際は、このプログラムが雛型になる

```
import cv2

cv2.namedWindow('Example 2-3', cv2.WINDOW_AUTOSIZE)
cap = cv2.VideoCapture(0)
assert cap.isOpened(), 'Cannot open the video.'

while True:

    ok, frame = cap.read()

    if not ok:
        break

    cv2.imshow('Example 2-3', frame)

    key = cv2.waitKey(33)
    if key >= 0:
        break

cap.release()
cv2.destroyAllWindows()
```

OpenCV で画像を読み込み，Matplotlib で表示する： 色チャンネルの順番が逆になっているので注意

```
import cv2
from matplotlib import pyplot as plt

img_bgr = cv2.imread('../images/color/Lenna.bmp')

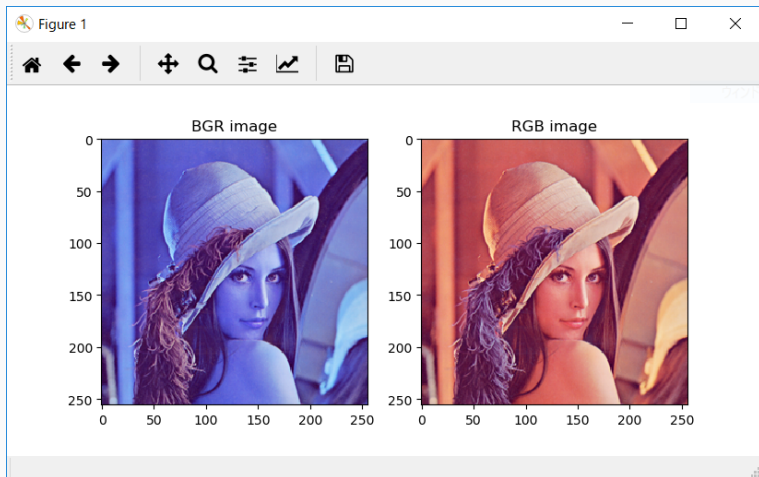
assert img_bgr is not None, 'Loading image is failed.'
assert len(img_bgr.shape) == 3, 'Loaded image is not color'

# 画像を BGR から RGB に変換する
img_rgb = cv2.cvtColor(img_bgr, cv2.COLOR_BGR2RGB)

# 変換前の画像を表示
plt.subplot(1, 2, 1), plt.imshow(img_bgr), plt.title('BGR image')
# 変換後の画像を表示
plt.subplot(1, 2, 2), plt.imshow(img_rgb), plt.title('RGB image')
plt.show()
```

実行結果

OpenCV はカラー画像を BGR 画像として読み込むが、Matplotlib は RGB 画像として解釈しようとするので表示が変になる



10 章：フィルタとコンボリユーション

- ・ 画像を色の値からなる「2次元配列」ではなく、「2変数関数」と解釈する
- ・ 画像フィルタリング：入力画像 $I(x, y)$ から新しい画像 $I'(x, y)$ を計算するアルゴリズム
 - ・ 例1：ある画像からぼけた画像を生成する
 - ・ 例2：ある画像を白と黒のみからなる画像に変換する

画像フィルタリングの内容はカーネルによって定義される

- 出力画像 $I'(x, y)$ の位置 (x, y) における画素値は入力画像中の位置 (x, y) 周辺の画素から計算される

$$I'(x, y) = \sum_{i,j \in \text{kernel}} k_{i,j} \cdot I(x + i, y + j)$$

- 上式中の $k_{i,j}$ を線形カーネル（フィルタ）と呼ぶ
- 画像に対してカーネル（線形，非線形問わず）を適用する操作をコンボリューションと呼ぶ

線形カーネルと適用のイメージ

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

線形カーネルによる画像フィルタリングの例：平滑化

