

OpenCV について



- OpenCV とは
- OpenCV の Python プログラム例
- 配列
- 画像の B, G, R の 3成分
- OpenCV でカメラの表示

URL: <https://www.kkaneko.jp/cc/opencv/opencv1.pptx>

金子邦彦



OpenCV とは



- **コンピュータビジョンと機械学習**のライブラリ.
- 2500 以上のアルゴリズム.
- 顔認識、物体認識、人間の動きの分類、カメラの動きの追跡、オブジェクトの動きの追跡、3次元モデルの抽出、ステレオカメラからの3次元点群の生成、イメージスティッチング、類似画像の検索、赤目の除去、眼球運動の追跡、ARの機能など
- ライセンス: BSD ライセンス
- インタフェース: C++, Python, Java, MATLAB
- マシン: Windows, Linux, Mac OS, iOS, Android

<https://opencv.org/about.html>

コンピュータビジョンに関わるライブラリ類



- 顔検知、顔識別： Dlib など
- 一般物体認識： 種々の手法
- 画像： Mask R-CNN, YOLO など
- 文字認識： Tesseract, OpenALPR など

OpenCV の Python プログラム例 (画像表示)



```
import cv2
```

```
import numpy as np
```

```
v = cv2.VideoCapture(0)
```

```
r, f = v.read()
```

```
if ( r == False ):
```

```
    exit()
```

```
print( f.shape )
```

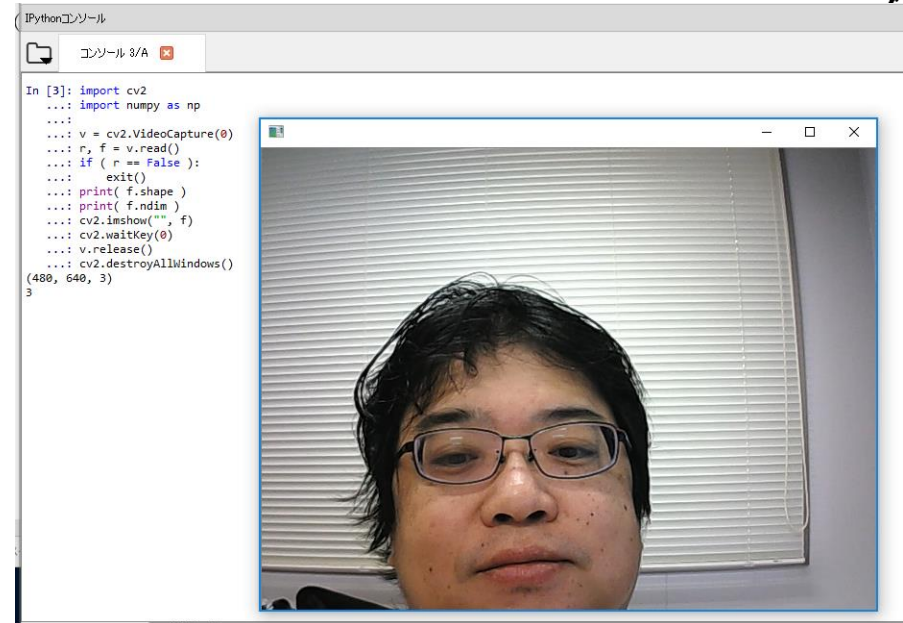
```
print( f.ndim )
```

```
cv2.imshow("", f)
```

```
cv2.waitKey(0)
```

```
v.release()
```

```
cv2.destroyAllWindows()
```



動作画面

v は cv2.VideoCapture 型の
オブジェクト

read : フレーム読み出し

release: 使用終了

f は配列

配列



- 要素の並び. 要素には添字がある.

0	1	2	3	4
8	5	4	1	3

1次元の配列 [8 5 4 1 3] の添字は、

0 1 2 3 4

配列の次元



配列は Python では次のように表示される.

1 次元 : [要素の並び]

2 次元 : [[要素の並び] ... [要素の並び]]

```
[8 5 4 1 3]
```

1 次元

```
[[ 1 2 3 4]
 [10 20 30 40]
 [100 200 300 400]]
```

2 次元

配列の形と次元の表示



```
In [2]: print(f)
[[[107 110 106]
  [ 98 101  97]
  [ 90  93  89]
  ...
  [ 93  87  85]
  [ 93  87  85]
  [ 94  88  86]]
 [[104 107 103]
  [ 96  99  95]
```

```
In [3]: f.shape
Out[3]: (480, 640, 3)
```

```
In [4]: f.ndim
Out[4]: 3
```

shape: 形の取得
ndim: 次元数の取得

3次元の配列 f を
`print(f)` で表示

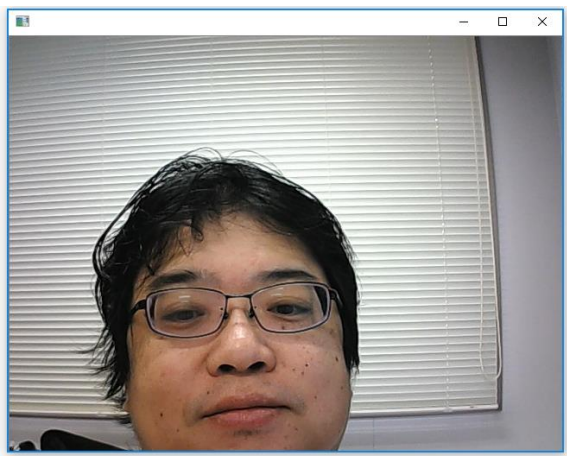
配列の形は

「 $480 \times 640 \times 3$ 」で
あることを確認

配列の次元は

「3」であることを
確認

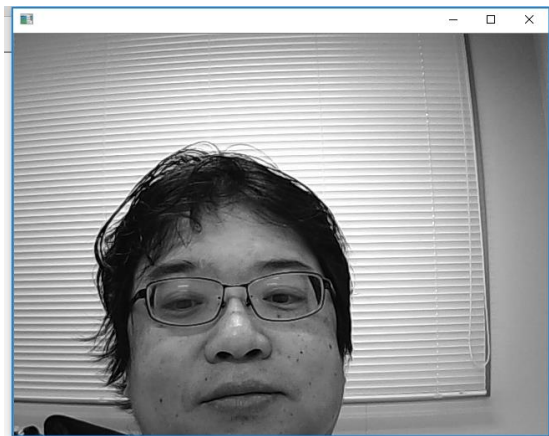
画像の B, G, R の 3成分



`cv2.imshow("", f[:, :, 0])`

`cv2.waitKey(0)`

B成分



`cv2.imshow("", f[:, :, 1])`

`cv2.waitKey(0)`

G成分



`cv2.imshow("", f[:, :, 2])`

`cv2.waitKey(0)`

R成分

OpenCV でカメラの表示



```
import cv2
```

```
import numpy as np
```

```
v = cv2.VideoCapture(0)
```

```
while(v.isOpened()):
```

```
    r, f = v.read()
```

```
    if ( r == False ):
```

```
        break
```

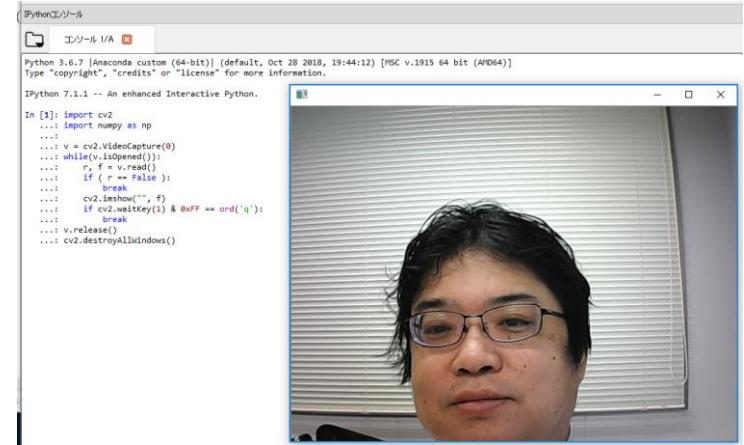
```
    cv2.imshow("", f)
```

```
    if cv2.waitKey(1) & 0xFF == ord('q'):
```

```
        break
```

```
v.release()
```

```
cv2.destroyAllWindows()
```



動作画面

ビデオ表示では
ループを行う