

コンピュータビジョン：第1回

岡谷貴之

この講義について

- 目的

- コンピュータビジョンの基礎を実際のプログラミングを通じて学ぶ
- （効用：python と OpenCV が使えるようになる）

- 進め方

- 理論を講義した後，実際のプログラムで動作を確認
- プログラムには python と opencv を使用
- 複数回のレポート課題（プログラミング）を課す予定

- 評価

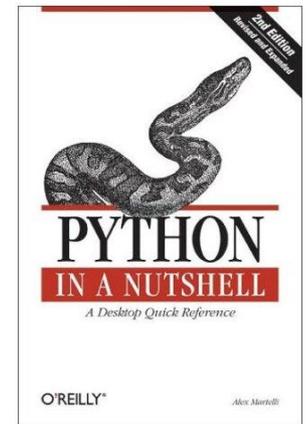
- レポート課題

- 質問

- okatani@vision.is.tohoku.ac.jp

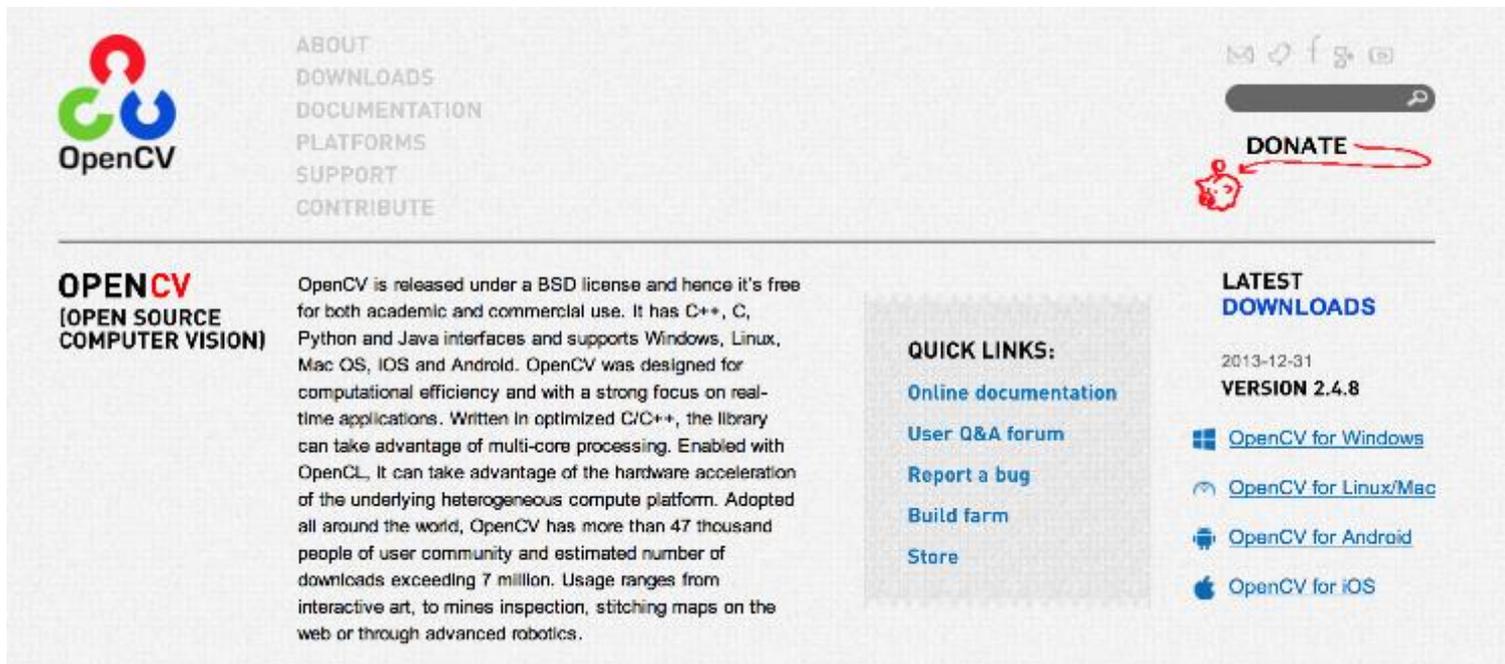
python

- 概要
 - 広く使用されている汎用のスクリプト言語
 - コードのリーダビリティが高くなるように言語が設計
 - Cなどの言語に比べて、より少ないコード行数でプログラムを表現できる（以上Wikipediaより）
- 数多くの利点
 - オープンソース
 - プログラム簡単
 - 様々なOSで利用可能
 - 豊富なライブラリ群があり，インストールが簡単
- 欠点
 - 実行速度を追求する場合には向かない



OpenCV

- **OpenCV**（オープンシーヴィ、英語: Open Source Computer Vision Library）とはインテルが開発・公開したオープンソースのコンピュータビジョン向けライブラリ。現在はWillow Garage（ウィロー・ガレージ）が開発を行っている。1999年にプロジェクト開始。現在に至る。（Wikipediaより）



OpenCV
(OPEN SOURCE COMPUTER VISION)

OpenCV is released under a BSD license and hence it's free for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, IOS and Android. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing. Enabled with OpenCL, it can take advantage of the hardware acceleration of the underlying heterogeneous compute platform. Adopted all around the world, OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 7 million. Usage ranges from interactive art, to mines inspection, stitching maps on the web or through advanced robotics.

QUICK LINKS:

- [Online documentation](#)
- [User Q&A forum](#)
- [Report a bug](#)
- [Build farm](#)
- [Store](#)

LATEST DOWNLOADS

2013-12-31
VERSION 2.4.8

- [OpenCV for Windows](#)
- [OpenCV for Linux/Mac](#)
- [OpenCV for Android](#)
- [OpenCV for iOS](#)

python + Open CV のデモ

- camtest.py
 - 注：カメラがついていないと実行できません
 - 適当なエディタで以下の中身のテキストファイルを作成
 - （あるいは講義ページからダウンロード）

```
import cv2

cv2.namedWindow('Camera')
cap = cv2.VideoCapture(0)

while 1:
    stat, image = cap.read()

    cv2.imshow('Camera', image)
    key = cv2.waitKey(10)

    if key == 0x1b: # ESC
        break
```

python + Open CV のデモ

- ビデオの再生 : vidtest.py

```
import cv2

cv2.namedWindow('Camera')
cap = cv2.VideoCapture('sample.avi')

while 1:
    stat, image = cap.read()

    if stat == False
        break

    cv2.imshow('Camera', image)
    key = cv2.waitKey(10)

    if key == 0x1b: # ESC
        break
```

python + Open CV のデモ

- shellからインタラクティブに実行も可能

```
>>> import cv2
>>>
image=cv2.imread('C:/Users/okatani/Dropbox/Lecture/CV/1st/lion.jpg')
>>> image.shape
(540, 960, 3)
>>> cv2.namedWindow('Image')
>>> cv2.imshow('Image', image)
>>> cv2.waitKey()
```

python + Open CV のデモ

- 遊んでみる: camtest.py に1行追加

```
import cv2

cv2.namedWindow('Camera')
cap = cv2.VideoCapture()

while 1:
    stat, image = cap.read()

    image = cv2.flip(image, 0) # flip the image upside-down

    cv2.imshow('Camera', image)
    key = cv2.waitKey(10)

    if key == 0x1b: # ESC
        break
```

python + Open CV のデモ

- 遊んでみる : 静止画とblending

```
import cv2

cv2.namedWindow('Camera')
cap = cv2.VideoCapture()

lion = cv2.imread('lion.jpg')
stat, image = cap.read()
print image.shape
lion = cv2.resize(lion, (image.shape[1], image.shape[0]))

while 1:
    stat, image = cap.read()

    image = cv2.addWeighted(image, 0.5, lion, 0.5, 0.0)

    cv2.imshow('Camera', image)
    key = cv2.waitKey(10)

    if key == 0x1b: # ESC
        break
```

python + Open CV のデモ

- 顔検出 : facedetect.py

```
import cv2

cascade = cv2.CascadeClassifier('haarcascade_frontalface_alt.xml')

cv2.namedWindow('Camera')
cap = cv2.VideoCapture(0)

while 1:

    stat, img = cap.read()
    rects = cascade.detectMultiScale(img, 1.3, 4)

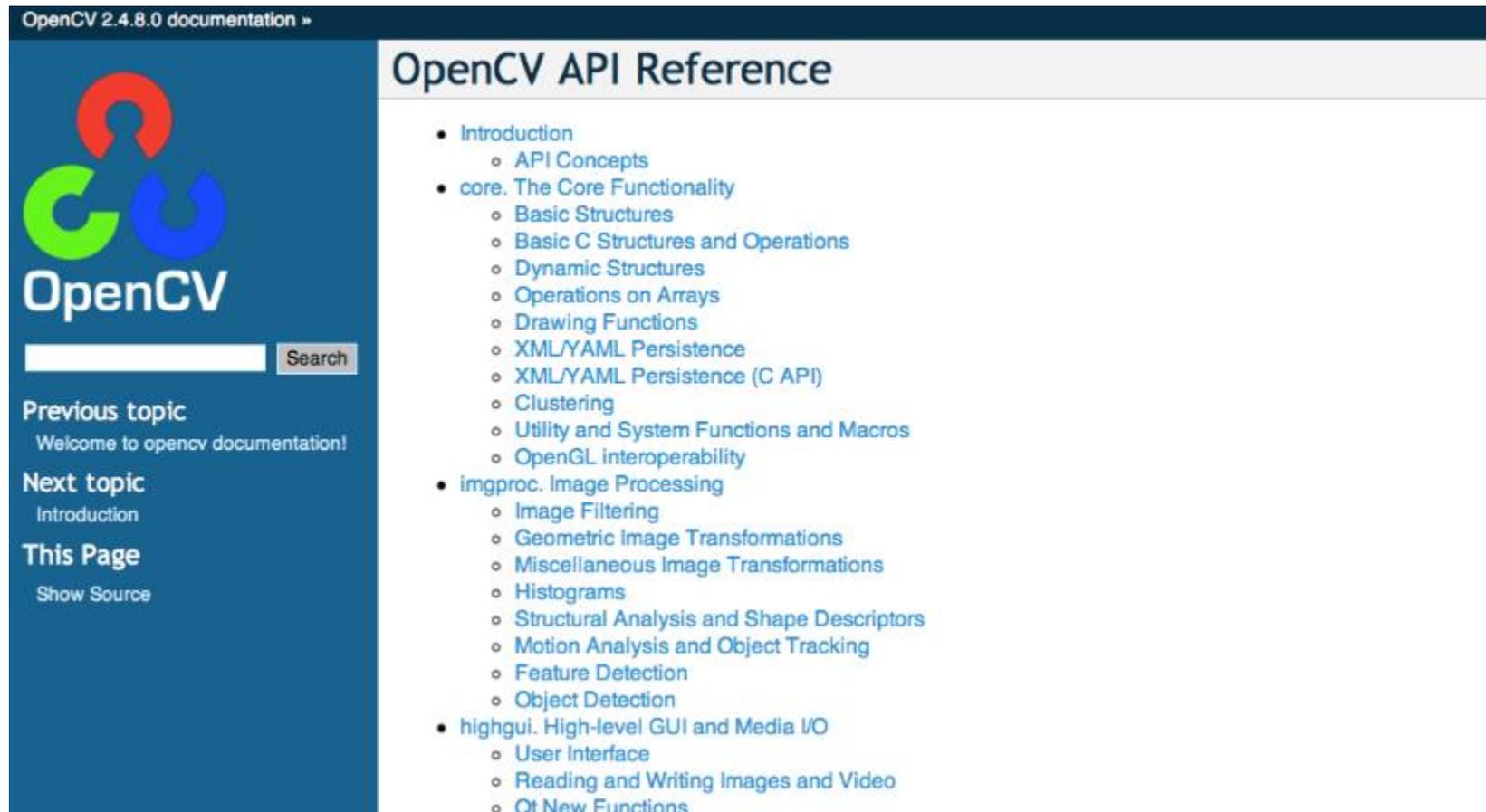
    if len(rects) != 0:
        rects[:, 2:] += rects[:, :2]
        for x1, y1, x2, y2 in rects:
            cv2.rectangle(img, (x1, y1), (x2, y2), (127, 255, 0), 2)

    cv2.imshow('camera', img)
    key = cv2.waitKey(10)

    if key == 0x1b:
        break
```

OpenCVのドキュメント

- <http://docs.opencv.org/modules/refman.html>



The screenshot displays the OpenCV 2.4.8.0 documentation website. The page title is "OpenCV API Reference". The left sidebar contains the OpenCV logo, a search bar, and navigation links for "Previous topic" (Welcome to opencv documentation!), "Next topic" (Introduction), and "This Page" (Show Source). The main content area lists the API Reference sections:

- [Introduction](#)
 - [API Concepts](#)
- [core. The Core Functionality](#)
 - [Basic Structures](#)
 - [Basic C Structures and Operations](#)
 - [Dynamic Structures](#)
 - [Operations on Arrays](#)
 - [Drawing Functions](#)
 - [XML/YAML Persistence](#)
 - [XML/YAML Persistence \(C API\)](#)
 - [Clustering](#)
 - [Utility and System Functions and Macros](#)
 - [OpenGL interoperability](#)
- [imgproc. Image Processing](#)
 - [Image Filtering](#)
 - [Geometric Image Transformations](#)
 - [Miscellaneous Image Transformations](#)
 - [Histograms](#)
 - [Structural Analysis and Shape Descriptors](#)
 - [Motion Analysis and Object Tracking](#)
 - [Feature Detection](#)
 - [Object Detection](#)
- [highgui. High-level GUI and Media I/O](#)
 - [User Interface](#)
 - [Reading and Writing Images and Video](#)
 - [Qt New Functions](#)

OpenCVのドキュメント

- このサイトにすべての関数の仕様が記載されている
- 例) ライブラリの関数 `addWeighted` の説明

addWeighted

Calculates the weighted sum of two arrays.

C++: `void addWeighted(InputArray src1, double alpha, InputArray src2, double beta, double gamma, OutputArray dst, int dtype=-1)`

Python: `cv2.addWeighted(src1, alpha, src2, beta, gamma[, dst[, dtype]])` → `dst`

C: `void cvAddWeighted(const CvArr* src1, double alpha, const CvArr* src2, double beta, double gamma, CvArr* dst)`

Python: `cv.AddWeighted(src1, alpha, src2, beta, gamma, dst)` → `None`

- Parameters:**
- **src1** – first input array.
 - **alpha** – weight of the first array elements.
 - **src2** – second input array of the same size and channel number as `src1`.
 - **beta** – weight of the second array elements.
 - **dst** – output array that has the same size and number of channels as the input arrays.
 - **gamma** – scalar added to each sum.
 - **dtype** – optional depth of the output array; when both input arrays have the same depth, `dtype` can be set to `-1`, which will be equivalent to `src1.depth()`.

The function `addWeighted` calculates the weighted sum of two arrays as follows:

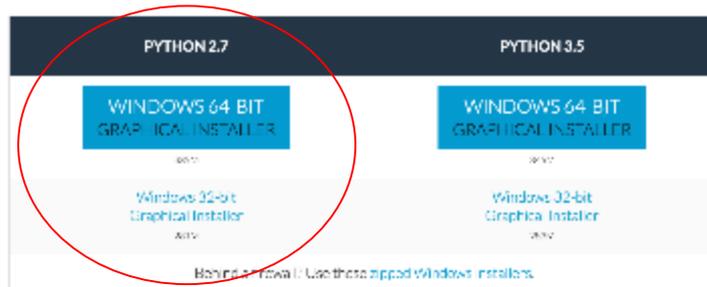
$$\text{dst}(\mathbf{I}) = \text{saturate}(\text{src1}(\mathbf{I}) * \text{alpha} + \text{src2}(\mathbf{I}) * \text{beta} + \text{gamma})$$

where \mathbf{I} is a multi-dimensional index of array elements. In case of multi-channel arrays, each channel is processed independently.

OpenCVとpythonのインストール (Windows7 64bit編)

- Anacondaと呼ばれるパッケージでpythonをインストールする。
(1) <https://www.continuum.io/downloads> ここからAnacondaをダウンロードする。

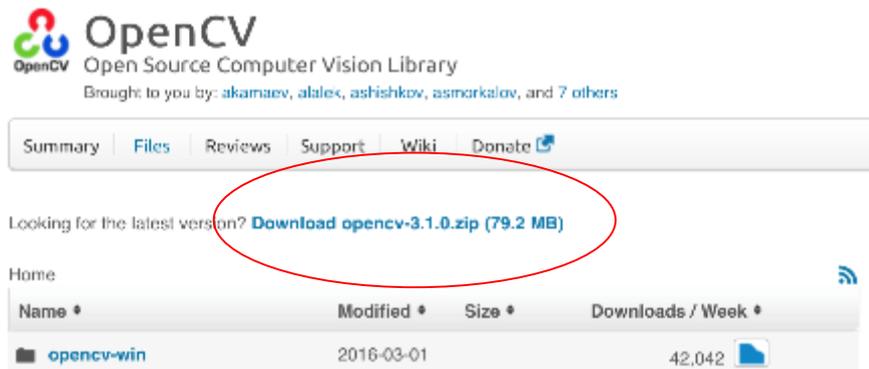
Anaconda for Windows



- (2) ダウンロードしたファイルを実行する。

- Opencv3をダウンロードしてインストールする。

<https://sourceforge.net/projects/opencvlibrary/files/> ここからOpencv3をダウンロードする。



OpenCVとpythonのインストール (Windows7 64bit編)

- パスを指定して、Opencv3をpythonでimportする。
 - (1) Copy "/opencv/build/python/2.7/cv2.pyd" to "/User/NAME/Anaconda2/Lib/site-packages/"
 - (2) pythonを開いてimportしてみる。

```
Python 2.7.11 [Anaconda 4.0.0 (32-bit)] (default, Mar  4 2016, 15:18:41) [MSC v.1500 32 bit (Intel)]
Type "copyright", "credits" or "license" for more information.

IPython 4.1.2 -- An enhanced Interactive Python.
?          -> Introduction and overview of IPython's features.
%quickref  -> Quick reference.
help       -> Python's own help system.
object?    -> Details about 'object', use 'object??' for extra details.

In [1]: import cv2

In [2]: print cv2.__version__
3.1.0

In [3]:
```

OpenCVとpythonのインストール(Ubuntu編)

- OpenCVのインストール：下記のページの説明どおり
 - http://docs.opencv.org/3.0-beta/doc/tutorials/introduction/linux_install/linux_install.html
 - (1) Required packagesをインストールする。
 - (2) <https://sourceforge.net/projects/opencvlibrary/> でOpencv3のソースコードをダウンロードする
 - (3) ターミナルを開いて、下記のどおり実行する

```
cd ~/opencv
mkdir release
cd release
cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local ..
```

(4)

```
make -j8 # -j8 runs 8 jobs in parallel
          # Change 8 to number of hardware threads
sudo make install
```

OpenCVとpythonのインストール(Ubuntu編)

- Pythonのインストール :

https://www.continuum.io/downloads#_unix ここでAnaconda-pythonをダウンロードしてインストールする。

Anaconda for Linux

PYTHON 2.7	PYTHON 3.5
LINUX 64-BIT 392M	LINUX 64-BIT 398M
Linux 32-bit 312M	Linux 32-bit 317M

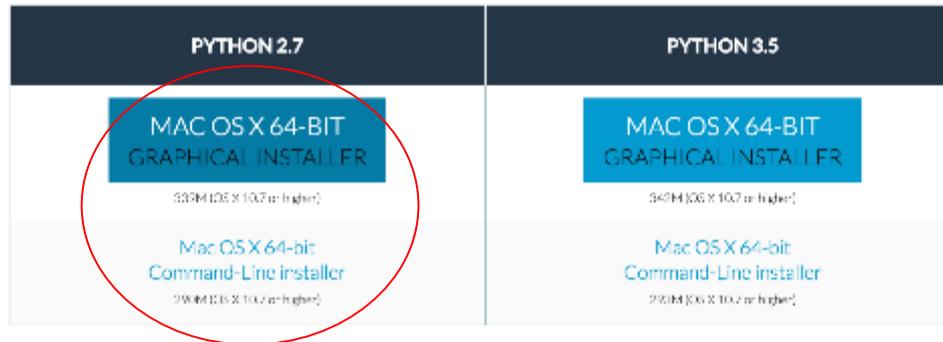
- ~/.bashrcでパスを設定する

```
export PYTHONPATH=/YOUR/OPENCV-PYTHON/INTERFACE/
```

OpenCVとpythonのインストール(Mac編)

- HOMEBREWの実装:
http://brew.sh/index_ja.html ここでhomebrewをダウンロードしてインストールする。
- Anaconda-pythonをインストールする:
<https://www.continuum.io/downloads#>

Anaconda for OS X



- OpenCV3 をインストールする:
ターミナルで“brew install homebrew/science/opencv3”を入力する

OpenCVとpythonのインストール(Mac編)

- パスの指定:
コピー "/local/Cellar/opencv3/3.1.0_3/lib/python2.7/site-packages/cv2.so" TO "~/anaconda/lib/python2.7/site-packages/"
- Import cv2してみる。

```
Python 2.7.11 |Anaconda 4.0.0 (x86_64)| (default, Dec 6 2015, 18:57:58)
[GCC 4.2.1 (Apple Inc. build 5577)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://anaconda.org
[>>> import cv2
[>>> print cv2.__version__
3.1.0
>>> ]
```

プログラムの実行

- 実行の仕方（ubuntuの場合）
 - Terminal を開いて， camtest.py のあるディレクトリで以下のよう
に打ち込む

```
user@ubuntu:~$ python camtest.py
```

- 実行の仕方（Windowsの場合）
 - “Python2.7” — “IDLE(Python GUI)” をクリックで実行
 - “File” メニューから “Open” を選び， camtest.py を選択
 - （あるいは同メニューから “New Window” を選んで開くエディタ
で camtest.py そのものを入力して作成してもよい
 - 開いたウィンドウの “Run” — “Run Module” を選ぶ（あるいは F5 を
押しても同じ）

次回までにすること

- 自分（の使える）PC上で、python と OpenCV が動作する環境を整える
- OSによってやり方が違う。分からなかったらまずは google検索。それでも分からなければ岡谷までメールを。