

コンピュータビジョン：第1回

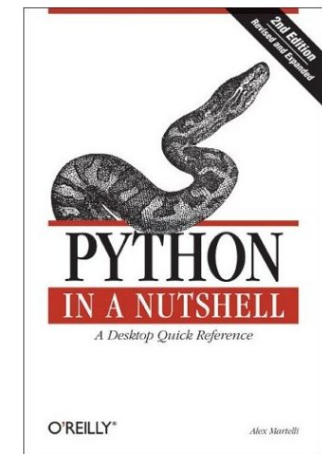
岡谷貴之

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パ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

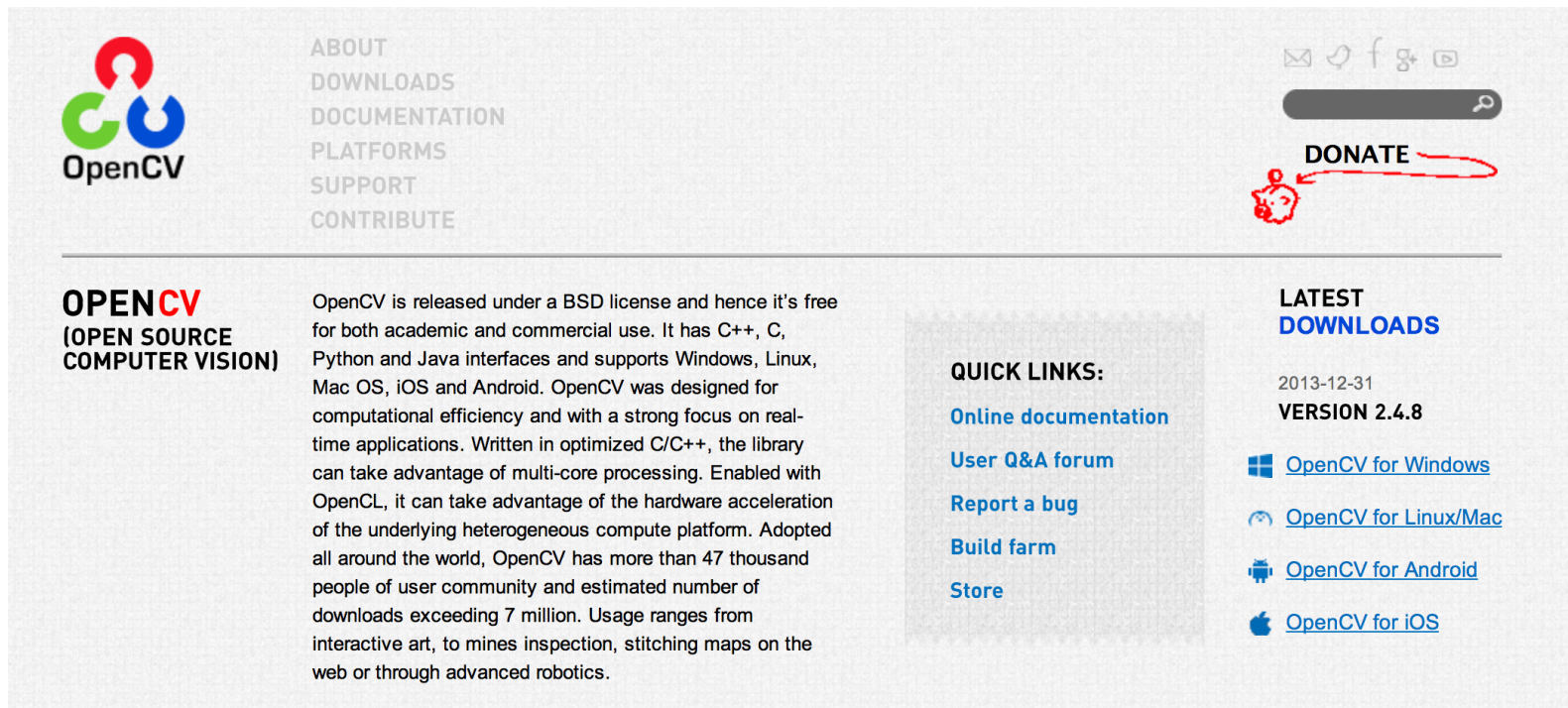
python

- 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）



The screenshot shows the OpenCV website homepage. At the top left is the OpenCV logo, which consists of three interlocking circles in red, green, and blue, with the text 'OpenCV' below it. To the right of the logo is a vertical menu with links: ABOUT, DOWNLOADS, DOCUMENTATION, PLATFORMS, SUPPORT, and CONTRIBUTE. In the top right corner, there are social media icons for email, GitHub, Facebook, Google+, and YouTube, followed by a search bar and a 'DONATE' button with a red arrow pointing to a small red robot icon. Below the header, the main content area is divided into three columns. The left column features the 'OPENCV (OPEN SOURCE COMPUTER VISION)' logo. The middle column contains a paragraph about OpenCV's BSD license, supported platforms (C++, C, Python, Java, Windows, Linux, Mac OS, iOS, Android), and its focus on real-time applications and hardware acceleration. The right column is titled 'LATEST DOWNLOADS' and lists the version '2013-12-31 VERSION 2.4.8' along with links for downloading OpenCV for Windows, Linux/Mac, Android, and iOS. A 'QUICK LINKS' section on the right side of the main content area provides links to online documentation, user Q&A forum, bug reporting, build farm, and the store.

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

cap = cv2.VideoCapture(0)

while 1:
    stat, image = cap.read()
    #image = cv2.resize(image, ((int)(image.shape[1]/2),
    (int)(image.shape[0]/2))

    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

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

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

- shellからインタラクティブに実行も可能

```
>>> import cv2
>>> image=cv2.imread('lion.jpg')
>>> image.shape
(540, 960, 3)
>>> 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

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

cap = cv2.VideoCapture()

lion = cv2.imread('lion.jpg')

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

    lion = cv2.resize(lion, (image.shape[1], image.shape[0]))
    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')

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

- <https://docs.opencv.org/3.1.0/>

Google Custom Search

 **OpenCV** 3.1.0
Open Source Computer Vision

[Main Page](#) [Related Pages](#) [Modules](#) [Namespaces](#) [Classes](#) [Files](#) [Examples](#) [Sphinx Documentation](#)

OpenCV modules

- [Introduction](#)
- [OpenCV Tutorials](#)
- [OpenCV-Python Tutorials](#)
- [Tutorials for contrib modules](#)
- [Frequently Asked Questions](#)
- [Bibliography](#)
- Main modules:
 - core. [Core functionality](#)
 - imgproc. [Image processing](#)
 - imgcodecs. [Image file reading and writing](#)
 - videoio. [Media I/O](#)
 - highgui. [High-level GUI](#)
 - video. [Video Analysis](#)
 - calib3d. [Camera Calibration and 3D Reconstruction](#)
 - features2d. [2D Features Framework](#)
 - objdetect. [Object Detection](#)
 - ml. [Machine Learning](#)
 - flann. [Clustering and Search in Multi-Dimensional Spaces](#)
 - photo. [Computational Photography](#)
 - stitching. [Images stitching](#)
 - cudaarithm. [Operations on Matrices](#)
 - cudabgsegm. [Background Segmentation](#)
 - cudacodec. [Video Encoding/Decoding](#)
 - cudafeatures2d. [Feature Detection and Description](#)
 - cudafilters. [Image Filtering](#)
 - cudaimgproc. [Image Processing](#)

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

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

Calculates the weighted sum of two arrays.

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

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

where I is a multi-dimensional index of array elements. In case of multi-channel arrays, each channel is processed independently. The function can be replaced with a matrix expression:

```
dst = src1*alpha + src2*beta + gamma;
```

Note

Saturation is not applied when the output array has the depth `CV_32S`. You may even get result of an incorrect sign in the case of overflow.

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.

gamma scalar added to each sum.

dst output array that has the same size and number of channels as the input arrays.

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()`.

See also

[add](#), [subtract](#), [scaleAdd](#), [Mat::convertTo](#)

python3 と opencv3のインストール

- Windows
 - anaconda(python3用)をインストール
 - opencv3をインストール
 - 参考サイト：
 - <https://www.youtube.com/watch?v=82Sx57q-5DQ>
 - <https://qiita.com/FukuharaYohei/items/5d49938ffd33d198f0c0>
- Mac
 - brewをインストール
 - pyenvをインストール
 - anaconda(python3用)をインストール
 - 私の環境では4.4.1使用
 - `conda install -c menpo opencv3`
 - 参考サイト：
 - <https://datumstudio.jp/blog/pythonによるopencvの導入macos>
 - <https://qiita.com/neet-AI/items/54f675dbcc1a0a513882>

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
- 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とTAまでメールを
 - okatani@vision.is.tohoku.ac.jp
 - xu@vision.is.tohoku.ac.jp