CSCI-677: Advanced Computer Vision

USC

USC Viterbi
School of Engineering

OpenCV
Introduction and Tutorial

Jiyang Gao

Sep. 5, 2017

Fall 2017

Outline

- Introduction
- Installation
- Examples

Introduction

- Open Source Computer Vision Library
  - Library of programming functions
    - Aimed at real time computer vision, launched in 1999
  - BSD license
    - Free for both commercial and research
  - Cross-platform
    - Windows, Linux/Mac, Android, iOS, etc.
  - Latest version: OpenCV 3.3
    - Release on 08/03/2017

Introduction

- More than 2500 optimized algorithms
  - Image Processing
    - Matrix operation, format conversion, filtering, …
  - Computer Vision
    - Calibration, segmentation, recognition, tracking, …
  - Machine Learning
    - SVM, classification, regression, …
    - Deep learning: not for training, but for deploying, ResNet, Googlenet, Alexnet...
  - Basic human computer interaction

Introduction

- OpenCV has C++, C, **Python**, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS.

- http://opencv.org/about.html
- http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_tutorials.html
- https://www.learnopencv.com/
- http://www.pyimagesearch.com/

Installation

Use OpenCV in Python on Mac OS / Ubuntu / Windows:

(complicated way: http://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/)

Step 1: Install Anaconda or Miniconda

(https://docs.continuum.io/anaconda/install/,
https://conda.io/docs/user-guide/install/index.html)

Step 2: Create a Conda environment

```
conda create -n xxx
source activate xxx
```

Step 3: Install OpenCV with conda

```
#Install opencv 3
#https://anaconda.org/menpo/opencv3
$ conda install -c menpo opencv3
#install opencv 2
#https://anaconda.org/menpo/opencv
$ conda install -c menpo opencv
```

Examples

Read and show images:

```
1   import numpy as np      # import numpy
2   import cv2              # import OpenCV
3
4   # load the color image
5   img_color = cv2.imread('./files/car.jpg')
6
7   # convert it to gray one
8   img_gray = cv2.cvtColor(img_color, cv2.COLOR_BGR2GRAY)
9
10  # display two images
11  cv2.imshow('Color Image', img_color)
12  cv2.imshow('Gray Image', img_gray)
13
14  # write the gray one to the disk
15  cv2.imwrite('car_gray.png', img_gray)
16
17  # press any key in the image window to exit
18  cv2.waitKey(0)
19
20  # destroys all the windows we created
21  cv2.destroyAllWindows()
```

Examples

Read and show images:

http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_gui/py_image_display/py_image_display.html#py-display-image
http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_gui/py_video_display/py_video_display.html#display-video

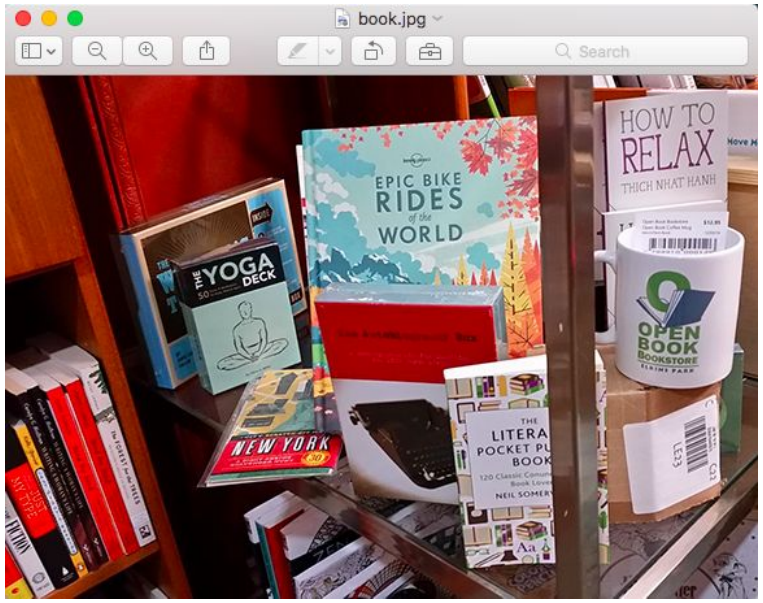
Examples

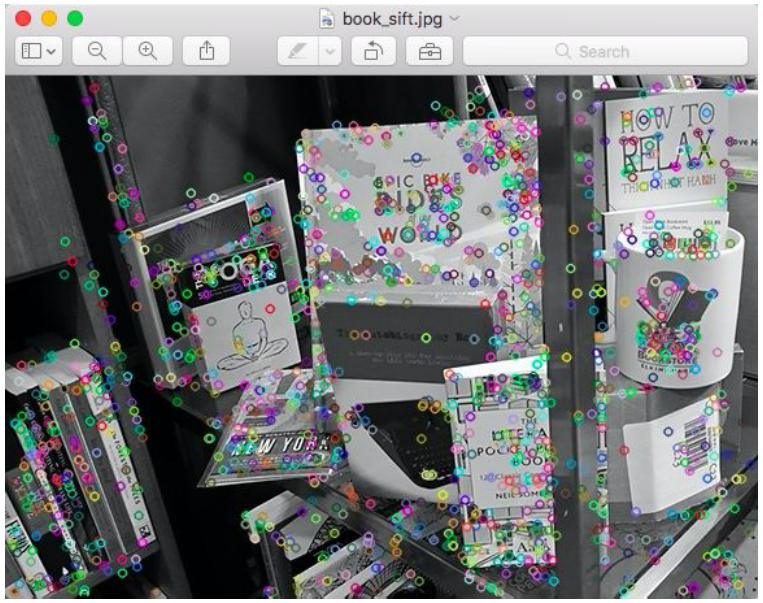
Compute SIFT features:

```python
import numpy as np      # import numpy
import cv2              # import OpenCV


# get the verion
cv2_ver = cv2.__version__.split('.')[0]

# read the image
img = cv2.imread('./files/book.jpg')

# conver it to gray
gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)

# create object
if cv2_ver == '3':
    sift = cv2.xfeatures2d.SIFT_create()
elif cv2_ver == '2':
    sift = cv2.SIFT()
else:
    print "Wrong OpenCV Version!"

# detect the key points
kp = sift.detect(gray, None)

# draw key points
img = cv2.drawKeypoints(gray, kp, img)

# write the image with key points to disk
cv2.imwrite('book_sift.jpg', img)
```

Examples

Compute SIFT features:





http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_table_of_contents_feature2d/py_table_of_contents_feature2d.html#py-table-of-content-feature2d