



Università
Ca' Foscari
Venezia

Computer Vision

Lab Companion Guide

Filippo Bergamasco (filippo.bergamasco@unive.it)

<http://www.dais.unive.it/~bergamasco>

DAIS, Ca' Foscari University of Venice

Academic year 2017/2018



Università
Ca' Foscari
Venezia

Requisites

This companion guide is based on Linux Ubuntu 16.04 Xenial but should be easily adapted to OSX or Windows

What you need:

- CMake
- Git
- A C++ compiler toolchain (like gcc)



Università
Ca' Foscari
Venezia

Install some libraries

For a complete opencv build, including libraries to open video files and/or external webcams, install the following before building opencv:

```
$ sudo apt-get install git build-essential  
cmake curl liblapack-dev libblas-dev  
libboost-all-dev ffmpeg libavcodec-dev  
libswscale-dev libavformat-dev libavutil-dev  
libavresample-dev
```



Università
Ca' Foscari
Venezia

Obtaining and building OpenCV

1- Clone the OpenCV git repository from github and create a directory named “build” inside it

```
$ git clone --depth 1 https://github.com/opencv/opencv.git  
$ cd opencv  
$ mkdir build  
$ cd build
```

2- Generate makefile using cmake and compile

```
$ cmake ../ -DCMAKE_BUILD_TYPE="Release"  
$ make -j 2
```



Università
Ca' Foscari
Venezia

Obtaining and building OpenCV

When the building is completed, `opencv/build` directory will contain all the necessary files (libs and headers) to create new applications with OpenCV

Additionally, running

```
$ sudo make install
```

Will install OpenCV to default system locations



Università
Ca' Foscari
Venezia

Installing ocv in OSX

In OSX the easiest way to install OpenCV is by using homebrew:

https://brew.sh/index_it.html

```
$ brew install opencv
```

When linking your project, use the following opencv build directory:

```
/usr/local/Cellar/opencv/3.4.0_1/
```



Università
Ca' Foscari
Venezia

Starting a new project

1- Clone the project template to be used for your lab assignments and final project

```
$ git clone https://gitlab.com/fibe/lab_ocv_template.git
```

2- Enter the lab_ocv_template directory and create a directory named build and one named dist

```
$ cd lab_ocv_template
```

```
$ mkdir build
```

```
$ mkdir dist
```



Università
Ca' Foscari
Venezia

Starting a new project

3- Edit the file CMakeLists.txt and change the lines

```
set(CvLab_NAME "assignment1")  
set(CvLab_VERSION "0.1.0.${BUILD_NUMBER}")  
set(CvLab_BRIEF "${CV2017_NAME} is the first assignment  
of the computer vision course 2017")
```

according to your project name, version and
description

4- Edit the file src/main.cpp and/or add additional cpp
and h files inside src/



Università
Ca' Foscari
Venezia

Compile the project

1- Enter the build directory and invoke cmake

```
$ cd build
```

```
$ cmake ../ -DOpenCV_DIR=<YOUR OPENCV BUILD DIRECTORY>
```

2- Compile your project with “make”, copy all your files in the dist directory with “make install” and run the main executable with “make run”

```
$ make
```

```
$ make install
```

```
$ make run
```



Università
Ca' Foscari
Venezia

OpenCV Basics

OpenCV (3.4.0) library documentation:

<https://docs.opencv.org/3.4.0/>

Composed by different modules:

core

imgproc

calib3d

features2d

highgui

... and many others



Università
Ca' Foscari
Venezia

OpenCV Basics

Please read carefully the library documentation introduction to understand the basic concepts behind the OpenCV APIs

<http://docs.opencv.org/3.4.0/d1/dfb/intro.html>



Università
Ca' Foscari
Venezia

cv::Mat

The cv::Mat is probably the most important data type in OpenCV

http://docs.opencv.org/3.4.0/d3/d63/classcv_1_1Mat.html

The class Mat represents an n-dimensional dense numerical single-channel or multi-channel array. It can be used to store real or complex-valued vectors and matrices, grayscale or color images, voxel volumes, vector fields, and so on



Università
Ca' Foscari
Venezia

Loading/Saving images

The `imgcodecs` module contains two useful functions `cv::imread` and `cv::imwrite` to save and load images respectively

Imgcodecs documentation:

http://docs.opencv.org/3.4.0/d4/da8/group__imgcodecs.html

`imread` function:

http://docs.opencv.org/3.4.0/d4/da8/group__imgcodecs.html#ga288b8b3da0892bd651fce07b3bbd3a56

`imwrite` function:

http://docs.opencv.org/3.4.0/d4/da8/group__imgcodecs.html#gabbc7ef1aa2edfaa87772f1202d67e0ce



Università
Ca' Foscari
Venezia

Displaying images

Images can also be displayed interactively with the function `imshow` implemented in the `highgui` module

`highgui` documentation:

http://docs.opencv.org/3.4.0/d7/dfc/group__highgui.html

`imshow` function:

http://docs.opencv.org/3.4.0/d7/dfc/group__highgui.html#ga453d42fe4cb60e5723281a89973ee563

NOTE: The function should be followed by `cv::waitKey` function which displays the image for specified milliseconds. Otherwise, it won't display the image.



Università
Ca' Foscari
Venezia

Other useful functions

Linear system solver:

http://docs.opencv.org/3.4.0/d2/de8/group_core_array.html#ga12b43690dbd31fed96f213eefead2373

Color-space conversion:

http://docs.opencv.org/3.4.0/d7/d1b/group_imgproc_misc.html#ga397ae87e1288a81d2363b61574eb8cab

Capture video from webcam:

http://docs.opencv.org/3.4.0/d8/dfe/classcv_1_1VideoCapture.html