

Weekly Exercise 1

Dr. Csaba Domokos and Lingni Ma
Technische Universität München, Computer Vision Group
April 12, 2016

Programming

General Information. In this lecture, you are free to choose your preferred programming language, e.g., C/C++, Python, Matlab. However, please note that, we mainly provide support for C++ and Python. We suggest you to use OpenCV library for image processing for both C++ and Python. If you program with Python, you will also need to get familiar with the numpy package. The OpenCV documentation can be found here: <http://docs.opencv.org/3.0-beta/modules/refman.html> . The numpy documentation can be found here: <http://docs.scipy.org/doc/numpy/reference/> . We also suggest to compile C++ programs with cmake. We usually provide you the CMakeLists.txt file. Therefore, you just need to run

```
cmake <dir> // generate Makefile, <dir> contains your CMakeLists.txt
make // generate executable
```

Exercise 1. Load, modify and save an image. This exercise is intended to get you familiar with the programming interface to process images. Your task is to load a color image, **write your own function** to convert the image to grayscale with the following formular,

$$g = 0.299 \times R + 0.587 \times G + 0.114 \times B$$

and then save the grayscale image. To check the results, you can compare your obtained grayscale image with the result from the build-in function. Below is a simple example of the C++ code.

Code 1: Example C++ code to read, modify and save image.

```
#include <iostream>
#include <opencv2/opencv.hpp>
#include <opencv2/imgproc/imgproc.hpp>
#include <opencv2/highgui/highgui.hpp>

using namespace cv;

cv::Mat
my_grayscale_conversion(const cv::Mat& color)
{
    /* fill your implementation */
    cv::Mat gray;
```

```
    return gray;
}

int
main (int argc, char** argv)
{
    Mat image = imread("bird.jpg", IMREAD_COLOR);    // Load

    /* Here fill in your implementation to convert to grayscale image.
     * Note that openCV read image in BGR order. */
    Mat mygray = my_grayscale_conversion(image);

    Mat refgray;
    cvtColor(image, refgray, COLOR_BGR2GRAY);        // Build-in function

    imshow("img", image);                            // Display
    imshow("mygray", mygray);
    imshow("gray", refgray);
    waitKey(0);

    imwrite("grayscale.png", refgray);              // Save

    return 0;
}
```
