



DISPLAYS - WS 2017/18

EXERCISE 1

Submission deadline: Thursday, 09.11.2017

- Language for all programming exercises is C++. Please upload the code and image results
- Eclipse is already installed for all WSI PCs (Sand 14, room C410). To change the IDEs perspective to C++ click on the upper right button *Open Perspective* and switch to *C/C++*
- The recommended OpenCV version is 3.1.0 since it's currently installed on our WSI PCs running on Ubuntu 16.04
- If you're working on your own machine, install OpenCV and an IDE of your choice. Make sure to change the projects library und include folders accordingly.
- The internet provides you with plenty of tutorials about OpenCV, e.g.
<http://opencvexamples.blogspot.com/>
<http://opencv-srf.blogspot.de/>
<http://proquest.tech.safaribooksonline.de/book/graphic-design/9781491937983>
https://docs.opencv.org/3.1.0/d9/df8/tutorial_root.html

1.1 Display an image with OpenCV

In this exercise, your task is to implement a small program to load and display an image using OpenCV. Start by creating a new project, according to the following tutorial:

http://docs.opencv.org/doc/tutorials/introduction/linux_eclipse/linux_eclipse.html

Remarks on tutorial:

7. Use the following example code:

```
1  #include <opencv2/opencv.hpp>
2  #include <opencv2/imgcodecs.hpp>
3  #include <stdio.h>
4
5  using namespace cv;
6
7  int main( int argc, char** argv )
8  {
9      Mat image;
10     image = imread( argv[1], 1 );
11
12     if( argc != 2 || !image.data )
13     {
14         printf( "No image data \n" );
15         return -1;
16     }
17
18     namedWindow( "Display Image", CV_WINDOW_AUTOSIZE );
19     imshow( "Display Image", image );
20
21     waitKey(0);
22
23     return 0;
24 }
```

- 8a. Use `/graphics/opt/opt_Ubuntu16.04/opencv-3.1.0/include` as include directory
- 8b. The **Library search path (-L)** is `/graphics/opt/opt_Ubuntu16.04/opencv-3.1.0/lib`
- 8b. The most common **Libraries(-l)** for OpenCV 3.1.0 are:
- `opencv_core`
 - `opencv_imgcodecs`
 - `opencv_imgproc`
 - `opencv_highgui`

Xa. Add the OpenCV library path to your environment variables. You can do this in Eclipse projectwise: Go to Run Configurations - C/C++ Application - [Name of your project] - Environment - Click on New...

Name: `LD_LIBRARY_PATH`

Value: `$LD_LIBRARY_PATH:/graphics/opt/opt_Ubuntu16.04/opencv-3.1.0/lib`

Xb. In case the method above fails, you can set the environment variable globally and unlimited. Open your terminal and type in:

```
echo 'export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/graphics/opt/opt_Ubuntu16.04/opencv-3.1.0/lib' >
~/.bashrc
```

After restarting Eclipse, the program will run successfully.

1.2 Gamma Correction and Absolute Difference Image

In this exercise, you will use OpenCV and C++ to do some basic image processing. The task will be computing the image gamma correction and the absolute difference image between the original image I_{in} and result image I_{out} expressed in Equation 1 and 2. y and x stand for the pixel indices for rows and columns.

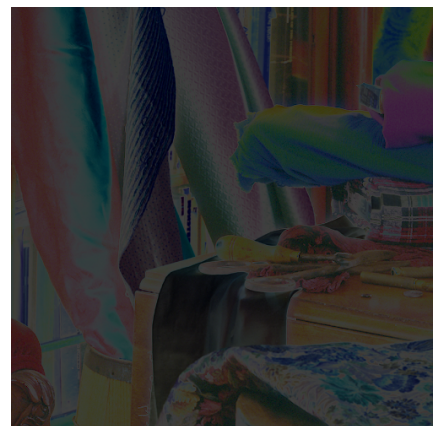
$$I_{out}(y, x) = I_{in}(y, x)^\gamma \quad (1)$$



(a) Original Image



(b) Gamma (0.5) Corrected Image



(c) Difference Image

Abbildung 1: Gamma Correction and Absolute Difference Image

$$I_{diff}(y, x) = |I_{out}(y, x) - I_{in}(y, x)| \quad (2)$$

Hints:

- `absdiff()` Computes the absolute difference between two arrays when they have the same size and type.
- The pixel intensity value of a color image with BGR color ordering, can be assessed by `Vec3b intensity = img.at<Vec3b>(y,x)[c];` where `c` stands for the color channels, blue – 0, green – 1 and red – 2.
- `pow()` raises every array element to a power.
- `imwrite()` saves an image to a specified file.