Computer Vision Lab 1
OpenCV Introduction

David Van Hamme, Sanne Roegiers

February 8, 2019

1 What is OpenCV

OpenCV is an open source computer vision library originally developed by Intel. The library can be used free of charge for all purposes under a BSD license. All important basic image processing functionality is present: reading and writing of video and images, filter operations, color transformations, feature detection, segmentation, stereo vision, calibration, ...

OpenCV is a cross-platform and supports Linux, Windows and OS X. The library has bindings for Python, C++ and C.

You can use OpenCV in the development environment that you choose yourself. For example, if you work in C++, you can use Eclipse or NetBeans. Windows users who already have Visual Studio installed can also continue to use that. For Python you can use a text editor of your choice and the command line, or an IDE like Spyder.

2 Installation

You can find out how to install OpenCV for the language and the operating system of your choice in the links on Minerva.

For those with limited programming experience, we recommend using Python under the Ubuntu operating system. In Ubuntu you can easily install the Python package manager " pip " via the command line:

```
sudo apt install python-pip
```

Then you can install the entire OpenCV library with

```
sudo pip install opencv-contrib-python
```

For your convenience you can also install the Spyder IDE:

```
sudo apt install spyder
```

3 Documentation

You can find the documentation on `http://docs.opencv.org/trunk/index.html`. When functions or data structures are suggested in these lab assignments, look them up in the documentation so that you understand what they are and how to use them. You will also find sample programs and tutorials here.

4 Exercise 1

Write a simple program that realizes the functionality below. You can test the program on **clouds.png** (download from minerva).

- read a PNG image whose filename is given on the command line;
- display this image on the screen until a key is pressed;
- convert the image to a grayscale image and display it on the screen until a key is pressed;
- threshold the image at 50% of the maximum intensity for this data type and display on the screen until a key is pressed;
- save the grayscale image and the threshold image as PNGs.

Functions that are useful: **imread, imwrite, namedWindow, imshow, waitKey, cvtColor, threshold**.

5 Report

Write a short report about how you solved the exercises. Include in this report your input and output images. Describe every new function that you used. Explain the basic algorithm and purpose of the functions and clarify the parameters and how you selected them. Upload your report in the form of **LabX_name.pdf** to the dropbox on Minerva.