

ARIZONA STATE UNIVERSITY
SCHOOL OF ELECTRICAL, COMPUTER, AND ENERGY ENGINEERING

EEE 508 Spring 2017
Hands-On Assignment #1

Assigned: 19 January 2017

Due Date: 2 February 2017

Reading: Read Chapters 1, 2, 3, 4 (Working with Video section), 6 (up to and including Canny) in the OpenCV book.

EXERCISE 1.1:

(10 points) OpenCV is an open-source C library for image processing. You can find a lot of guides and help on the web (opencv.org, sourceforge.net, and other). Download and install the latest stable version of OpenCV 3.2 from the link <http://opencv.org/downloads.html> as described in the installation guide posted on Blackboard under Assignments.

For exercises requiring video display, below are links with useful instructions for playing a video from a file or camera:

http://docs.opencv.org/2.4/modules/highgui/doc/reading_and_writing_images_and_video.html

http://docs.opencv.org/3.0-beta/modules/videoio/doc/reading_and_writing_video.html

If the video does not play correctly, you may not have the appropriate video codec installed on your computer. The following link provides useful instructions in this regard:

<http://docs.opencv.org/doc/tutorials/highgui/video-write/video-write.html>

Also, refer to the posted Code under Assignment on Blackboard for capturing a video from camera and writing it to a file.

Perform Exercises 3 and 4 in Chapter 2 of the OpenCV book. Submit on Blackboard under Assignments, a zipped folder called VideoCaptureDisplay_yourlastname.zip, where yourlastname is replaced by your actual last name. The zipped folder should contain the codes for Exercises 3 and 4 and a Readme file including step-by-step instructions for compiling and running the codes.

EXERCISE 1.2:

(5 points) Refer to Chapter 6 of the OpenCV book.

- (a) Download the 256×256 image **bird.jpg** from the Course Web Page under Weekly Schedule by clicking on IMAGES.
- (b) Load and display the selected image as described in Chapter 2 of the OpenCV book.
- (c) Detect the edges of the image using the Laplacian operator `cvLaplace()`. Display the resulting Laplacian edge map.
- (d) Detect the edges of the image using the Sobel operator `cvSobel()`. Display the resulting Sobel edge map.
- (e) Detect the edges of the image using the Canny edge detector `cvCanny()`. Display the resulting Canny edge map.

Submit the implemented code, a readme file containing compilation and running instructions, the input image, and the resulting edge map images as a zipped folder called yourname_edge_detect.zip under Assignments in Blackboard.