



University of
Stavanger

Faculty of Science
and Technology

Stavanger, August 25, 2021

ELE610 Robot Technology, autumn 2021

Image Acquisition assignment 3

For this assignment each group should write a brief report (pdf-file). Answer the questions and include figures and images as appropriate. You may answer in Norwegian or English, or a mix. The intention here is that you should do as much as you are able to within the time limit for each assignment, which is 15-20 hours. A report containing a table showing time used, for each student of the group, will normally be accepted even if all tasks are not done. If all tasks are done, the time report does not need to be included.

3 Image acquisition using μ Eye camera

The Imaging Development Systems GmbH (IDS) simple μ Eye XS or the more advanced μ Eye CP camera can be used in this assignment. The task is to capture an image, or a sequence of images (video), and to do some simple image processing on the image in Python. We will also continue to develop the simple image viewer and image processing framework using Python and Qt.

You can find technical information on the μ Eye CP camera [here](#), or from the [IDS page](#). The camera is mounted on the “UiS Image Acquisition Test Rig” and can be connected to the computer using the USB3 (!) interface, you may look at [IDS paper on USB 3](#). There is also a trigger connected to the camera.

3.1 Use IDS programs

The same camera driver as in assignment 2 should be used here. Attach camera to USB-gate on the PC and start “IDS Camera Manager”. The attached camera should be visible in “Camera list” on the top of the program window. The buttons in the middle of the window will display general information and specific camera information. You may double-click on the line showing the

camera to start the μ Eye Cockpit program. You can now try the different alternatives. Try some of these, in particular investigate the different options that can be adjusted for this camera. Note that more options can be used for the advanced CP camera than for the more simple XS camera. Eventually, use the “Optimal Colors” button and capture and display image and video. The scene should include some of the colored dices that should be on the camera rig table.

Find a good set of options to use for the dark blue dice, and another set of options to use for the yellow dice. What are the important options? Save the two images and include them in the report.

3.2 Use μ Eye camera and Python

This section continues the Python section in assignment 2. Here you should finish some, or perhaps all, of the tasks from assignment 2 but now using the CP camera. You may also include points from the list below, and note that you don’t need to start from the top of the list. You may name your solution for this assignment `appImageViewer3X.py` where X is your group name (desk label).

- a. Check the Camera menu from `appImageViewer2X.py` for the CP camera, update the Camera On and Camera Off actions if needed, and make the program aware of which camera that is connected.
- b. Update the Image Capture action on Camera menu if needed.
- c. Add a feature (an action) that print information on the CP camera, or XS camera if that is what is connected. There should (probably) be some differences on what information is printed depending on camera type.
- d. Add a feature (an action) to Dice menu that display a dialog window where you select parameters to the `cv2.HoughCircles()` function, try this, and show the effects.
- e. Add a feature (an action) to Dice menu that find the number of eyes in each dice in the captured image. The results may be printed to standard output or shown on image.
- f. Add a feature (an action) to Camera menu that display a dialog window where you can change camera options. This dialog should be considerable simpler than the dialog in the μ Eye Cockpit program.
- g. The final task here is to add yet another action to the Dice menu that capture video and continuously finds and shows (print) the number of eyes in each dice in the video scene. (I was not able to do this, but perhaps you are, anyway stop when time budget is used)

The preliminary sketch for solution to this exercise is `appImageViewer3.py`, but this example has not (yet) suggestions for solution for the video part of this exercise.

The IDS example Python files for Qt4 [IDS techtip](#) has a solution for video capture. When these files were modified to Qt5, as in

- `pyueye_example_main.py`,
- `pyueye_example_gui.py`,
- `pyueye_example_camera.py` and
- `pyueye_example_utils.py`.

the video features still works. But I was not able to deeply understand the way threading and memory control are done in these example files, thus the Camera menu in `appImageViewer2.py` does not include video capture functionality. It will probably also be a challenge for most of the students to solve this, but perhaps the more talented, and programming experienced, of you are able to do this?