University of Stavanger

Faculty of Science and Technology

Stavanger, December 20, 2024

ELE610 Applied Robot Technology, V-2025

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 IDS camera

The Imaging Development Systems GmbH (IDS) simple μEye XS 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. We now want to add some color processing features. You can find some (technical) information on the μEye XS camera from the IDS XS page ↗.

3.1 Use IDS programs

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. If you have not examined the options in this program, now is the time. 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 μEye Cockpit program, and try the many options available in this program. Try these.

Karl Skretting, Department of Electrical Engineering and Computer Science (IDE), University of Stavanger (UiS), N-4036 Stavanger. E-mail: karl.skretting@uis.no.

Then, 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. Is there any difference between options and results? What are the important values? Save the images and include them in the report.

3.2 Use IDS 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. The two first point from the list below are basically the same as points from assignment 2. You should solve as many as possible of the points from the list below. You may name your solution for this assignment `appImageViewer3X.py` where `X` is your group name (desk label). You may start this by copying from your previous solution, `appImageViewer2X.py`, and for some hints you may look at `appImageViewer3.py`.

a. 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. Note that this is done in the example file `appImageViewer3.py` under `Dice` menu, and you may copy code from this file and use `clsHoughCirclesDialog.py`, as long as you understand what is done.

b. Add a feature (an action) to Dice menu that find the number of eyes in a dice in the captured image. The results may be printed to standard output or shown on image. Start simple with only one dice in the image, and color does not matter. Example:
`Dice shows 3 eyes.`

c. Find the color of the dice. This can be done in many ways, perhaps consider pixels just outside the radius for each circle that locate the black dot (dice eye). You should consider a table with color names and values, and allow for a small range around each color.

d. Print results for an image with one dice. Start simple with only one dice in the image. Example:
`Yellow dice shows 3 eyes.`

e. Print results for an image with two or more dices. Example:
`Yellow dice shows 3 eyes.`
`Pink dice shows 3 eyes.`

f. Add a feature (an action) to Camera menu that display a dialog window where you can change (at least one) camera option. The dialog should be considerable simpler than the dialog in the μEye Cockpit program. This link ↗ in IDS software documentation goes directly to where the setting camera parameters is described and will be helpful.

   Note that the corresponding automatic feature must be turned of to activate the (manually) change option, i.e. to make it actually work.

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)

Some hints for the solution, especially for color processing, are in `appImageViewer3.py` ↗, but this example has not (yet) any suggestions for solution for the video part of this exercise.

The IDS example Python files for Embedded Vision Kit using OpenCV and Qt4 in IDS techtip ↗ has a solution for video capture. An older(?) version of these files were modified to Qt5[1] by myself, 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?

[1] The version on the IDS web-page per June 2022 is newer, but still dated several years ago, that the one I once started from. The current files on IDS web-page may be a better start. The newer files are prepared for Qt5.