University of
Stavanger

Faculty of Science
and Technology

Stavanger, September 9, 2021

ELE610 Robot Technology, autumn 2021

Image Acquisition assignment 4

For this assignment each group should write a brief report (pdf-file). Answer the questions and include figures and images as appropriate. 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 for each student. 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.

4 A rotating object in the scene

The Imaging Development Systems GmbH (IDS) μEye CP camera should be used in this assignment to capture an image of a scene containing a rotating object. The object here, on the UiS Image Acquisition Test Rig, is a disk faced directly towards the camera. The disk is driven by a motor and the speed can be adjusted. Using Python and image processing (OpenCV) the task is to find and show the angular speed of the disk (rotations per minute, rpm) based on one image, or a sequence of images (video). On the new (the smaller one) camera rig it should be possible to read out the true speed.

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 camera here. 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.

For this assignment it can be a good idea to work towards a final solution from the start. Thus read the complete assignment before you start programming, and work towards your own improved image viewer program, if you are group 1a an appropriate name for your program is `appImageViewerGr1a.py` and this program should extend `appImageViewer.py`.

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

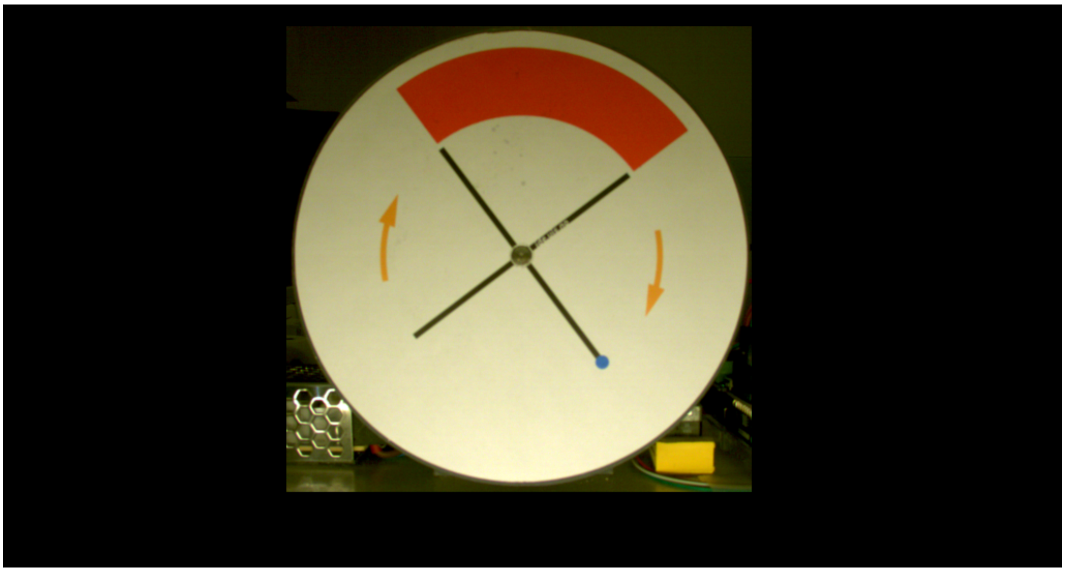

Figure 1: Image of disk on UiS camera rig. The disk is not moving and disk is in “trigger” position, i.e. the pin on the backside of the disk is positioned in the trigger optical switch, obstructing it. The image is size 2048x1088 (down sampled to 1024x544), but camera property size is set to 900x900 (down sampled to 450x450), which make the outside pixels black (but still in the image).

4.1 Angle of disk in image

We could use at least three different approaches to measure the angular speed of the disk.

i) If you know the exposure time the speed can be calculated based on how blurred the moving parts of the image is.

ii) If a video sequence of images exists and you know the frame rate the speed can be calculated based on how much an object has moved from one frame to the next

iii) If a trigger is available, which trigger the image capture when the object at a fixed position, and there is a small, fixed and known, delay from trigger time to exposure time, the speed can be calculated based on how much the object has moved from trigger position.

For all these tasks we need to locate the disk in the image and to find the angle that the disk has rotated, relative to an initial position. Case i) (how blurred?) is difficult (inaccurate), it can be used in combination with case iii) above. Thus below the first steps towards getting speed based on angle of disk

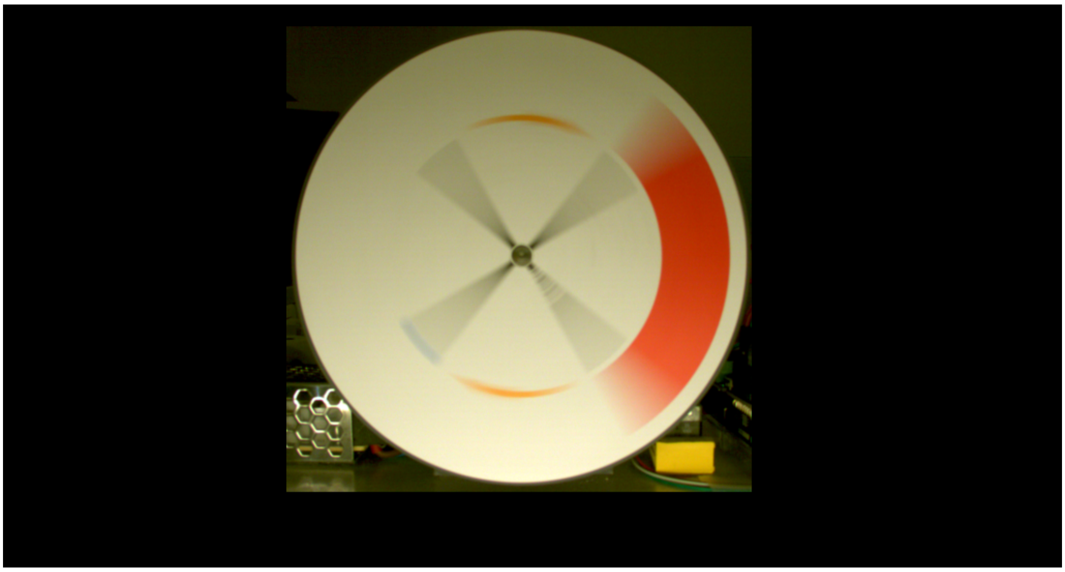

Figure 2: Image of disk on UiS camera rig. The disk is rotating and image is taken at random time. The image is size 2048x1088 (down sampled to 1024x544), but camera property size is set to 900x900 (down sampled to 450x450), which make the outside pixels black (but still in the image).

should be done. If you find it difficult to start you may look at my suggestion `appImageViewer3.py`.

a. Let the disk be still in "trigger" position, as in Fig. 1. Use the IDS μEye Cockpit program to adjust the camera options and capture an image.

b. Crop the image so that it contains only the disk. The crop function in `appImageViewer1.py` can be used as it is (even thought it is not good). Find out how it works and use it on the image from previous point. You may set all irrelevant pixels to black, perhaps this must wait to after next point, i.e. all pixels with a distance to the center larger than the disk radius is set to 0. You may also make the image gray scale and store it as a JPG image, which take much smaller space than the BMP image (perhaps) used when IDS μEye Cockpit program stored the image.

c. Find the center of the disk (pixel in image). Since shortcut C is taken (Clear image) you may use shortcut M (for Mid point). Several ways to do this can be considered, explain your method well before you try to implement it. The method in `appImageViewer3.py` is rigid, it may be improved by adding a dialog, similar to the dialog classed used in `appImageViewer1.py`. It may also be that other methods can solve this problem just as well.

d. Find the angle of the disk. Do it manually first by measuring on the screen or on a paper, then make a Python function that does this. This can be the angle of the blue dot relative to clock background in Fig. 1 it is between four and five, or it can be the center of the red patch which here is just after twelve. The important thing is that we get a reference. Explain the definition you use for the reference angle and how you find it.

e. Let the disk be rotating slowly and capture a new image, as in Fig. 2. Find the angle for this image. Give the answer in degrees between 0 and 360.

4.2 Camera in trigger mode

Let the disk rotate with approximately half of maximum speed. Use the IDS μEye Cockpit program to set camera option to trigger mode.

a. Let the trigger delay time be 10 ms. Capture single images and observe that the angle of the disk will be the same for each image in this set. What is the angle of the disk in these images?

b. Adjust the trigger delay time to 50 ms and capture a new set of images. What is the angle of the disk in these images?

c. What is the difference for the angle between the second set of images and the first set of images? What is the delay time difference?

d. Explain how you can use the two numbers from previous point to calculate disk rotation speed in rpm (rotations per minute). Do the calculations, and find the answer.

e. What angle would you expect the disk to have if the delay time were 0 ms?

f. Increase the rotation speed of the disk, keep the trigger delay time to 50 ms and capture a new image. What is the disk angle now? And the disk rotation speed?

4.3 Python program

This section continues the Python section in assignment 3. You continue to develop your best version of `appImageViewer.py`.

a. In assignment 3 you should have made three actions on the Camera menu of the program; open/initialize the camera, capture an image, and close the camera. Make sure that these three features (actions) works as intended and include (copy) them to your program. Alternatively, you may try the similar functions made in `appImageViewer2.py`.

b. Add a feature (an action) to the program that find the position of the disk center in the image, as done before. The results may also be printed to standard output.

c. Add a feature (an action) to the program that find the angle of the disk as done before. The results may be printed to standard output.

d. Add a feature (an action) to Camera menu that set the trigger on and that also set the trigger delay time to 50 ms. You don't need to use a dialog window here.

e. Modify the action in the program that find the angle of the disk to also find the rotation speed. Both results should now be printed to standard output.

f. Add an action to the program that check that trigger is on, capture an image using the trigger and calculate the rotation speed. The action could be on the program menu and perhaps have shortcut Ctrl+R.

g. Now, if you still have time left, you can clean up the program, make sure that all the features works as intended. It would be nice if the program detect by itself if a IDS camera is attached and which camera it is, and adjust the user interface accordingly.

h. If you still have time left, you can add an action to the program that start the camera in video capture mode, frame rate could be 1-3 fps. The action may also start a dialog window that display disk rotation speed, perhaps both by displaying a number and a slide bar. The dialog window should also have a Stop button that stop the video and close the window.