CITS4402 Computer Vision

Project: Automatic calibration of a holographic acquisition rig

Due: 23 May 2023, 11:59pm (NO EXTENSIONS)

Students are free to use either Matlab or Python with OpenCV and scikit-image, provided that all deliverables are provided and presented/demonstrated in the assessment session.

Grouping

Students have to work in groups of three to complete the project. Students to for their groups and inform facilitator.

Timeline

Week 7 to Week 12: Work on your project. The facilitator will be available on Wednesdays (4 pm to 6 pm) for any query in room G.04, or can be contacted through email, nasir.ahmad@uwa.edu.au.

Week 12: Presentation and Demonstration of the project will be during lab timings on Wednesday and Thursday of Week 12. Timing for individual groups will be announced at a later stage.

23 May 2023, 11:59 pm: Final deadline to submit your project on LMS. One submission per group.

The project

Proper calibration is of utter importance for the use of vision algorithm. In this project, you will re-implement the automatic calibration procedure developed by one of the UWA PhD student for a holographic acquisition rig (Figure 1). The rig in question is made up of several cameras meant to reconstruct a subject in 3D from multiple points of view. To properly align the different images, the rig needs to be calibrated. To this end, an automatic procedure has been developed based on the recognition of hexagonal targets.

You will be re-implementing this procedure. You will come up with an interface to quickly change the hyperparameters of the algorithm and visualize the results. Your interface should be a graphical user interface (GUI) created with matlab appbuilder or with python qt, or a web app created with django. The code should be provided with all the relevant dependancies. If you decide to use python you need to specify any external library you used in a virtualenv defintion file. Along with the code, you should provide a ReadMe file describing how to use your project and your main results.

Figure 1: The holographic acquisition rig

Data and references

You will be provided with an image for each of the cameras in the rig (1 ZED stereo vision camera (camera 11) and 4 Real Sense Active vision cameras (cameras 71, 72,73 and 74), as well as the intrinsic parameters of the cameras as Json Files. The Real Sense cameras have one RGB sub-camera and two Infrared sub-cameras, while the ZED camera has two RGB sub cameras. For this project, use **only the images of the RGB** sub-cameras.

You will also be provided with a copy of the relevant sections of the Thesis of the PhD student who worked on that project, describing the algorithm for the automatic calibration in more details. For clarity and assessment purposes, the project is divided into the following sub-tasks.

Task 1 – The rough detection (30 pts)

The first step is to detect the candidate targets in the images. To this end, use the method described in section 1.2 of the attached thesis extracts.

You are free to use existing functions of matlab or python to conduct the connected components analysis. All the required measurements can be extracted from the binary images by using the blobanalysis class in the matlab vision toolbox [https://au.mathworks.com/help/vision/ref/vision.blobanalysis-system-object.html]. In python, you can use skimage.measure.label [https://scikit-image.org/docs/stable/api/skimage.measure.html#skimage.measure.label] and skimage.measure.regionprops from the skimage package [https://scikit-

image.org/docs/stable/api/skimage.measure.html#skimage.measure.regionprops] for the same purpose.

Your code should then display the input image, along with the approximate position of all targets' points (you can indicate the position of each point with squares drawn around them, by using a binary mask or by marking them with a dot). Each target should have six and only six points. As an image might contain multiple targets you can use different colours to differentiate the individual targets when you display the results.

Task 2 – Targets analysis and refinement (20 pts)

The second step in the project is to implement the analysis and refinement methods described in section 1.3 and 1.4 of the attached thesis extract.

Your code should then display the refined centroid position of each target's points. Each target should be labeled with the string corresponding to the target color arrangement, as in figure 2 of the thesis extract.

Task 3 – Cameras alignments (30 pts)

The last step, is to use the points you measured in the two previous steps to find the position of the targets and cameras in the rig, as described in section 1.5 of the attached thesis extract. You are free to use existing functions for the PnP problem, which are estworldpose for Matlab [https://www.mathworks.com/help/vision/ref/estworldpose.html] and solvePnP for OpenCV [https://docs.opencv.org/4.x/d5/d1f/calib3d_solvePnP.html]. It is your responsibility to implement the global optimization function based on what you have been presented in the lectures.

Your code should then display the 3D position of the targets' points and the cameras as displayed in figure 3 of the attached thesis extract. In matlab, you can do that with the plot3 function [https://www.mathworks.com/help/matlab/ref/plot3.html]. In python, you can use the mplot3d toolkit [https://matplotlib.org/stable/tutorials/toolkits/mplot3d.html].

Presentation and ReadME file (20 pts)

The rest of the points shall be awarded based on the group presentation/demonstration and quality of the ReadME file. The ReadME should give the procedure to run the whole project.

FAQS

Do I have to write a report?

No. The approach should be explained in the presentation/demonstration. ReadMe file (submitted with your code) should give the detailed procedure to run the project.

Presentations

The group will present their system and demonstrate that their calibration system is working properly. All members to present.

Submission Requirements

You are required to upload (on LMS) a folder containing your Matlab/Python code along with ReadME file and any other requirement. Marks will be based on your presentation and the submitted code.