ENPM673 – Perception for Autonomous Robots

Project 3

Arunava Basu, Madhu NC, Ritvik Oruganti, Samer Charifa

Due date: 12th April 2023, 11:59PM

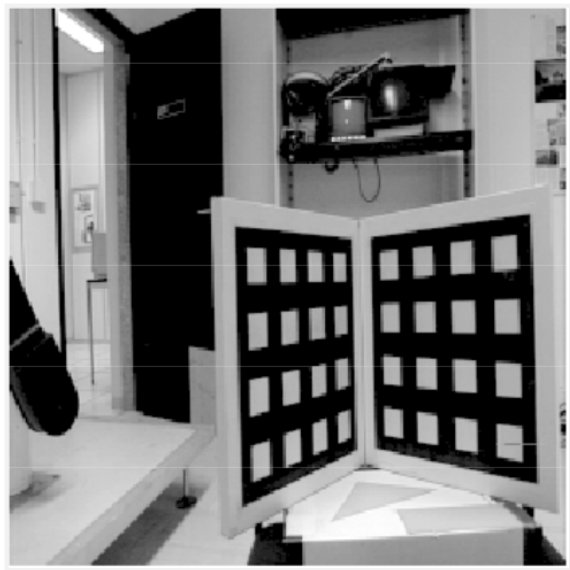
Submission guidelines:

- This homework is to be done and submitted individually.
- Your submission on ELMS/Canvas must be a **zip file & a pdf file**, following the naming convention
- YourDirectoryID_proj3.zip & YourDirectoryID_proj3.pdf. If your email ID is abc@umd.edu or abc@terpmail.umd.edu, then your Directory ID is abc. Remember, this is your directory ID and NOT your UID.
- Please submit only the python script(s) you used to compute the results, the PDF report you generate for the project and a detailed README.md file which includes the steps to run your code and any non-standard libraries used. The zip file should contain only the source code and related files. **The report should not be inside the zip file**.
- Include sample outputs in your report.
- For each section of the homework, explain briefly what you did, and describe any interesting problems you encountered and/or solutions you implemented.

Problem 1 [60 Pts]:

Calibrate the camera (Find the intrinsic matrix K),

For this question, you are **NOT** allowed to use any in-built function that solves the question for you.



1. What is the minimum number matching points to solve this mathematically?
2. What is the pipeline or the block diagram that needs to be done in order to calibrate this camera given the image above.
3. First write down the mathematical formation for your answer including steps that need to be done to find the intrinsic matrix K.
4. Write a program that will calibrate the camera given the point correspondences from world to image. Use the data provided below. Please note you are only allowed to use numpy for this question. No marks will be given if you use any other library/tool.

Image points		**World Points**		
x	**y**	**X**	**Y**	**Z**
757	213	0	0	0
758	415	0	3	0
758	686	0	7	0
759	966	0	11	0
1190	172	7	1	0
329	1041	0	11	7
1204	850	7	9	0
340	159	0	1	7

Problem 2 [40 Pts]:

In this problem, you will perform camera calibration using the concepts you have learned in class. Assuming a **pinhole camera model** and **ignoring radial distortion**, we will be relying on a calibration target (**checkerboard** in our case) to estimate the camera parameters. The calibration target used can be found here.

This was printed on an A4 paper and the size of each square is **21.5 mm**. Note that the Y axis has an odd number of squares and X axis has an even number of squares. It is a general practice to neglect the outer squares (extreme squares on each side and in both directions).
Thirteen images taken from a Google Pixel XL phone with focus locked can be downloaded from here which you will use to calibrate.

For this question, you are allowed to use **any in-built** function.

- Find the checkerboard corners using any corner detection method (inbuilt OpenCV functions such as `findChessboardCorners` are allowed).
- Use these corners to estimate the Projection matrix P.
- Decompose the P matrix into the Translation, Rotation and Intrinsic matrices using the Gram–Schmidt process and compute the reprojection error for each image.

Note:

- Your code should display:
  - The images showing the extracted corners.
  - The reprojection error for each image.
- Any other necessary intermediate results.