Mobile Robotics Homework 3
Hongzhen Xu

Notes: Because gtsam and apriltag libraries cannot be installed on my windows machine,

this homework is finished with the virtual box linux ubuntu os.

When you run the codes, please change the paths of the files in the code.

1 1

The corners data for each image is in 8 txt files named [corners IMG XXXX.txt].
The output images for each image with corners marked are in 8 jpg files named [output_image(IMG_XXXX).jpg].
All 8 images are detected with their corners.
But for IMG_3917.JPEG, Gaussian Blur is applied to reduce noise, and I increase contrast, for the purpose of detecting corners.
Therefore, this image is not suitable for calculating camera calibration because of loss of resolution and details.
Therefore, I use only 7 other images to calculate camera matrix.
The codes for problem 1 are in [problem1 part 1.py] and [problem1 part 2.py]
.
Specifically, the code for IMG_3917.JPEG is in [problem1 part 1 for 3917.py].
The estimated parameters of the camera matrix K and other outputs are in the txt file named [calibration estimated parameters of the camera matrix K.txt].

2 2

(a)
The camera projection function $\pi(K, X, P_i)$ maps a 3D point P_i in R^3 in the world frame to a 2D point u_i in R^2 in the image plane:
$u_i = \pi(K, X, P_i)$
where K is the camera matrix, and X describes the rotation and translation of the camera.
The goal is to minimize the reprojection error, which is the difference between the observed 2D points u and their projections π based on the estimated pose X.
The reprojection error for a single point is:
$re_i = u_i - \pi(K, X, P_i)$
The PnP problem can then be formulated as a nonlinear least-squares optimization problem:

optimal X=$argmin_X \sum_{i=1}^{N} ||u_i - \pi(K, X, P_i)||_2^2$
where X is the variable to optimize (the camera's pose).
The term $||u_i - \pi(K, X, P_i)||_2^2$ is the squared Euclidean distance (or L_2 norm) between the observed 2D point u_i and the predicted 2D point $\pi(K, X, P_i)$.
(b)
Tag 0 corners are recorded in the file [Tag0corners.txt].
Marked corners in the file [tag0corners.jpg].
Estimated and optimized pose of camera is in [camera pose.txt].
The codes for problem 2 are in [problem 2 part 1.py] and [problem2 part 2.py]
and [problem2 part 3.py] .