1 Programming: Camera Calibration

In this problem, we are interested in solving the camera calibration problem, which determines the intrinsic parameters of a camera by using an image of a rig (Calibration.jpg). We will keep this question open-minded where we do not specify the key points on the rig. Instead, you are required to pick the key points by yourself. Include the 2D pixel coordinates and 3D coordinates of the feature points you picked. The 2D pixel coordinates should be stored in a Matlab matrix 'Coord2d' of dimension 2×N, where N is the number of key points. The 3D coordinates of the feature points should be stored in a Matlab 'Coord3d' matrix of dimension 3×N. Include the following function for estimating the intrinsic camera parameter:

$$[\mathrm{K}] = \mathrm{cameracali}(\mathrm{Coord2d}, \mathrm{Coord3d})$$

where 'Coord2d' and 'Coord3d' are the 2D and 3D coordinates of the key points you picked. $\mathrm{K} \in \mathrm{R}^{3\times3}$ is an upper right matrix that encodes the intrinsic camera parameters.
In your writeup analyze the following:

1. *Explain how you pick the 2D key points. You can use the function 'ginput ' to pick these feature-points. Note that depending on the resolution of your screen, you may have to rescale your image, pick the key points, and then scale it back. It is also recommended to use Matlab's SURF feature detectionto pick the strongest corner features for the 2D keypoints on the calibration image, and then snap the key points you picked onto these detected key points. The 'snapping' procedure could be simply performing nearest neighbor search.*
   Answer...

2. *Explain how you compute the corresponding 3D keypoints. To this end, it is recommended that you pick a 3D coordinate system, which is usually aligned with the axses of the cube box, and which the origin is located at one of the corners of the cube. Then you can count the x, y, and z coordinates of the key points.*

   Hint: You can compute the corresponding 3d points by assigning xyz axis to the checkerboard box and counting the number of units for the coordinates that each of the point lies on the box's coordinate space. I used the box corner closest to the camera as the origin.
   Answer...

3. *The first step is to estimate the matrix $\prod$ = [KR, KT], where $K \in \mathbb{R}^{3\times3}$ is the upright matrix that encodes intrinsic camera parameters, and $R \in$ SO(3) and $T \in \mathbb{R}^3$ encode the extrinsic camera parameters. Note that you will get two matrices from the smallest eigenvector computation, ou can eliminate one by enforcing that each λ_i has to be positive in the following two constraints:*

$$\lambda_i \mathrm{x}_i = \prod \mathrm{X}_i,$$

where $x_i = (x_i^{2d},\ y_i^{2d},\ 1)^T \in \mathbb{R}^3$ *is the homogenous coordinate of the 2D pixel coordinate of the i-th key point, and* $X_i = (x_i^{3d},\ y_i^{3d}, z_i^{3d},\ 1)^T \in \mathbb{R}^4$ *is the homogenous coordinate of the i-th key point. You can use the sign of*

$$\sum_i \frac{x_i^T \prod X_i}{x_i^T x_i}.$$

4. *The second step is to estimate K, R, and T from matrix* $\prod$*. This step would involve QR decomposition as well as other operations. For QR decomposition, please refer to 'CS376Lecture15note.pdf' on canvas. Compute the determinant of your estimated R. You should put your estimated K, R, T, and det(R) in the PDF writeup in order to earn credits.*
   Answer...

5. *Test your program with different configurations of 2D and 3D key points. Compare the resulting intrinsic and extrinsic parameters. Answer the question where to pick key points for robust estimation of intrinsic and extrinsic camera parameters.*
   Answer...

6. **Extra Credit** *Suppose you have marked 20 key points, and your goal is to select 10 of them to estimate the camera parameters. Write a program to output the indices of these 10 keypoints. Hint: You will need to consider a statistical model for the pixel and 3D coordinates of the feature points. Then you need to write out the matrix* $\prod$ *as a function of the perturbations of the coordinates.*
   Answer...

7. **Extra Credit** *So far we have talked about using points for image calibration. Propose a strategy that uses lines for calibration, i.e., correspondences between lines in the input image and 3D lines.*
   Answer...

2 Programming: Structure-From-Motion

- *Please mention how to pick the feature correspondences. You need to provide two visualizations, one for the source image, and another for the target image. You can either color-code the corresponding key points, or you can directly draw correspondences across the input images.*

  —Features that are clearly visible in both images with matching similarities around them. They should be spaced out and in differing depths.

  Answer...

- *Use the provided intrinsic camera parameter matrix K and the corresponding key points to solve for the essential matrix E. Note that you will have two potential solutions, where one is the negation of the other.*

  —Done with estimation.

  Answer…

- *Extract the rotation* $\overline{R}$ *and the translation* $\overline{T}$ *from each resulting essential matrix E. Use the sign of the induced depth for each key point to eliminate implausible essential matrices and the associated rotations and translations. Compute the determinant of* $\overline{R}$*. You should put* $\overline{R}$*,* $\overline{T}$*, and* $det(\overline{R})$ *in the PDF writeup in order to earn credits.*
  Answer…

- *Please play with five sets of correspondences and compare the resulting relative transformations. Discuss which set of feature correspondences lead to potentially more accurate relative pose estimations.*
  Answer…

- **Extra Credit** *Instead of using manually marked feature correspondences, please run RANSAC to extract consistent correspondences across the two input images.*

  —Done and commented out of the way. Proof below

  Answer…