CS376: Computer Vision: Assignment 3
Due: Oct. 31st, 11:59 PM

Instruction: 100 points in total.
Submission: See the end of this document for submission instructions.

1 Programming: Camera Calibration (50 points)

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 keypoints 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 $2 \times N$, where N is the number of keypoints. The 3D coordinates of the feature points should be stored in a Matlab $'Coord3d'$ matrix of dimension $3 \times N$. Include the following function for estimating the intrinsic camera parameter:

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

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

- (5 pts) Explain how you pick the 2D keypoints. 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 keypoints, and then scale it back. It is also recommended to ues Matlab's SURF feature detection to pick the strongest corner features for the 2D keypoints on the calibration image., and then snap the keypoints you picked onto these detected keypoints. The 'snapping' procedure could be simply performing nearest neighbor search.

- (5 pts) 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 keypoints.

  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.

- (20 pts) The first step is to estimate the matrix $\Pi = [KR, KT]$, 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, you can eliminate one by enforcing that each λ_i has to be positive in the following two constraints:

$$\lambda_i \boldsymbol{x}_i = \Pi \boldsymbol{X}_i, \tag{1}$$

  where $\boldsymbol{x}_i = (x_i^{2d}, y_i^{2d}, 1)^T \in \mathbb{R}^3$ is the homogeneous coordinate of the 2D pixel coordinate of the i-th keypoint, and $\boldsymbol{X}_i = (x_i^{3d}, y_i^{3d}, z_i^{3d}, 1)^T \in \mathbb{R}^4$ is the homogeneous coordinate of the 3D coordinate of the

i-th keypoint. You can use the sign of

$$\sum_i \frac{\boldsymbol{x}_i^T \Pi \boldsymbol{X}_i}{\boldsymbol{x}_i^T \boldsymbol{x}_i}.$$

- (15 pts) The second step is to estimate K, R and T from matrix Π. This step would involve QR decomposition as well as other operations. For QR decomposition, please refer to 'CS376_Lecture_15_note.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.

- (5 pts) Test your program with different configurations of 2D and 3D keypoints. Compare the resulting intrinsic and extrinsic parameters. Answer the question where to pick keypoints for robust estimation of intrinsic and extrinsic camera parameters.

- **Extra credit (5 pts).** Suppose you have marked 20 keypoints, 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 Π as a function of the perturbations of the coordinates.

- **Extra credit (5 pts).** 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.

2 Programming: Structure-From-Motion (50 points)

In this problem, we are interested in estimating the relative camera pose between two images of the same underlying 3D scene (SourceImage.jpg and TargetImage.jpg). Our goal is to leverage the intrinsic camera parameters estimated in Problem 1 Camera Calibration, turning this into a calibrated two-view structure-from-motion problem. We will use the so-called eight-point method. Similar to the previous problem, you are asked to mark feature correspondences between two images (you can do this manully or using an automatic algorithm). Note that in the required part, the marked correspondences are supposed to be compatible with each other. In the extra credit part, you may run RANSAC to pick consistent correspondences across a pair of images.

Again, please submit the two sets of feature correspondences you pick. The first set of feature points will be given by 'sCoord2D', and the second set of feature points will be given by 'tCoord2D'. Both sets of feature correspondences have $n \geq 8$ keypoints. The same as above, you can first mark the keypoints, and then snap them onto the detected corners.

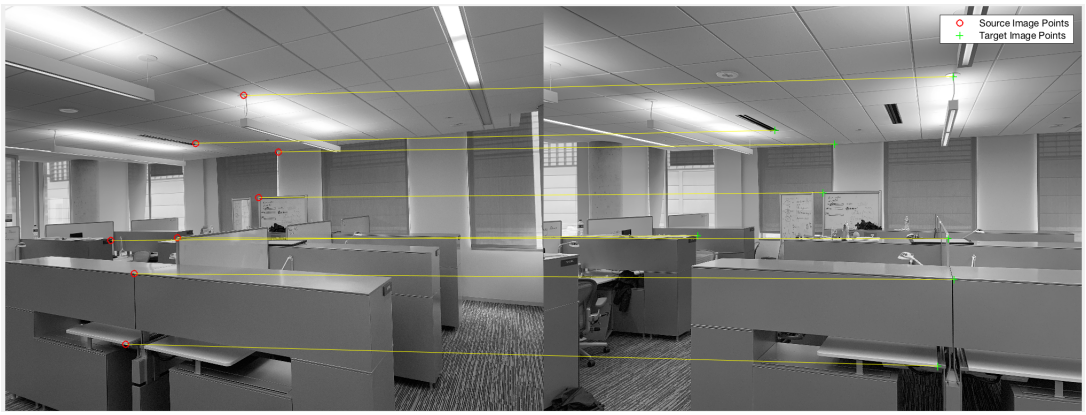
Now given the feature correspondences, and the intrinsic camera parameters K estimated in Problem 1. Our goal is to estimate the rigid transformation ($\overline{R} \in \mathbb{R}^{3\times3}, \overline{T} \in \mathbb{R}^3$) that relates the second image and the first image:

$$[\overline{R}, \overline{T}] = \text{relativepose(sCoord2D, tCoord2D, K)}$$

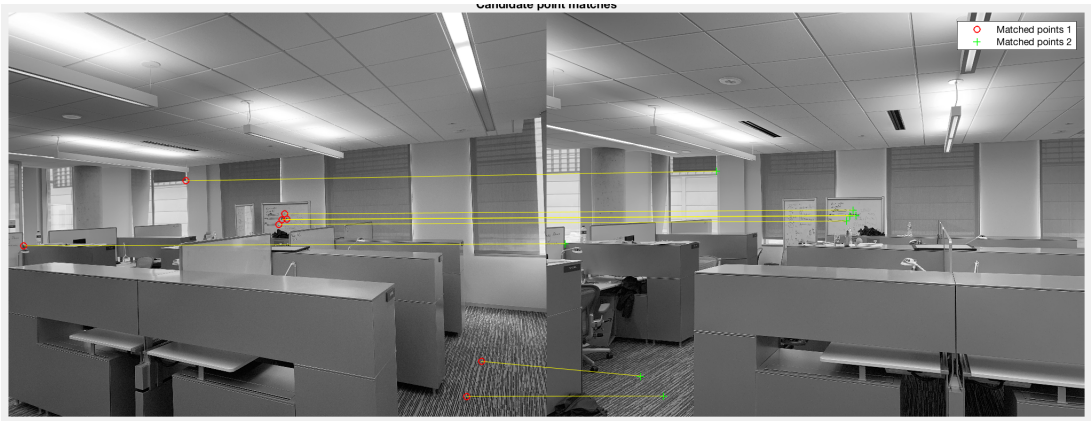
In your writeup analyze the following:

- (10 pts) 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 keypoints, 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.



- (20 pts) Use the provided intrinsic camera parameter matrix K and the corresponding keypoints 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.
- (15 pts) 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 keypoint 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.
- (5 pts) 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.
- **Extra credit (5 pts).** 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



Submission instructions:

Create a single **zip** file and submit on Canvas that includes

- Your well-commented code, including the files and functions named as specified above.
- A **PDF** writeup of your results with embedded figures where relevant.

Please do not include any saved matrices or images etc. within your zip file.