Computer Vision
Spring-2021
Assignment-1
Posted on: 26/01/2021
Due on: 23:59hrs, 03/02/2021

Guidelines

1. Follow the specified repository structure. `src` folder will contain the Jupyter notebooks used for the assignment. `images` folder will contain any images used for the questions.
2. Commit your work regularly to avoid losing progress. Make sure you run your Jupyter notebook before committing, to save all outputs.
3. The report should contain description of the problem, algorithms and results. It should be written in markdown, in the notebook itself.
4. Make sure that the assignment that you submit is your own work. **Any breach of this rule could result in serious actions including an F grade in the course**.
5. The experiments and report writing takes time. Start your work early and do not wait till the deadline.
6. You are not allowed to use inbuilt functions that directly solve the tasks assigned. Confirm with TAs regarding whether some function can be used, when in doubt.

Questions

1. **Direct Linear Transform**

    1. For the given image `calib-object.jpg` , use any 20-30 different points on different planes and perform the Direct Linear Transform (DLT) based calibration as discussed in class. Note that you need to manually estimate the image co-ordinates of the given world points and refer to `calib-object-legend.jpg` for world measurements.

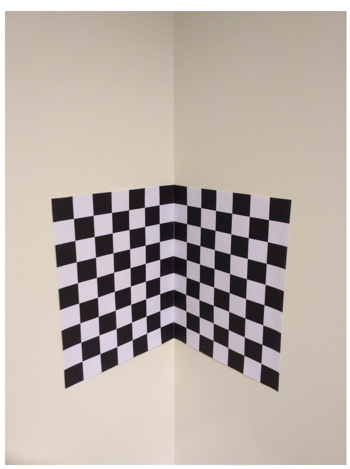

Figure 1: Calib object

Assume these two scales :

(a) Scale of each chessblock = 28x28mm

(b) Scale of each chessblock = 2800x2800mm

Perform these three experiments, report the **projection matrix** and **reconstruction error**, for both the above mentioned scales:

(a) Use original data points as input.
Perform reconstruction on the original data points. **(main experiment)**

(b) Use normalized data points as input.
Perform reconstruction on the normalized data points (without denormalizing the projection matrix obtained)

(c) Use normalized data points as input.
Perform reconstruction on the original data points (with the denormalized projection matrix).

Do the results vary between the above experiments? If yes, report how and why are the variations observed.

References for this problem:

(a) Use similarity transformation matrices for normalizing and denormalizing steps. Refer these Camera Calibration Notes. Slide 17-19 are relevant for this task.

(b) Refer Chapter 4 of "Multiple View Geometry in Computer Vision" by Richard Hartley and Andrew Zisserman (2nd Edition), uploaded on moodle. Section 4.4 talks about the importance of normalization of points.

2. Decompose the projection matrix obtained for the **main experiment** above and report:

   (a) **camera matrix (K)**

   (b) **rotation matrix (R)**

   (c) **projection center (C)**

   Notice that upon QR decomposition to obtain **K** and **R**, the obtained K matrix might have positive diagonal elements. This indicates a positive camera constant, while we usually assume it to be negative in theory as it becomes easier to handle signs. To fix this we use the following method:

$$K' = KR(z, \pi) \text{ and } R' = R(z, \pi)R$$

   Be sure to fix the matrices using this method. Show how the decomposition still holds true with these transformed matrices.

$$\begin{bmatrix} c & cs & x_H \\ 0 & c(1+m) & y_H \\ 0 & 0 & 1 \end{bmatrix}$$

Figure 2: **K** matrix parameters, **c** is the camera constant, **s** and **m** are the shear and scale parameters respectively.

3. (a) Implement the RANSAC based variant of the above calibration method and report your observations for **main experiment**.
   (b) RANSAC is said to be successful if in atleast 1 of the iterations, it selects only *inliers* (correct correspondences) from the input data points. Assume that your accuracy for annotating the points for the above experiment is 80%. What is the minimum number of iterations you will have to run RANSAC for getting probability of success greater than 95%?
   (c) Assuming the same accuracy, plot a curve of **Probability of success of RANSAC** vs **Number of iterations required**.

4. **BONUS**: Repeat the **main experiment** after correcting for radial distortion. Estimate the radial distortion parameters from the straight lines in the image. What do you observe regarding the resulting parameters?
   For this part you can use inbuilt cv2 functions.

2. **Zhang's method**

   1. Use checkerboard images `IMG5456.JPG - IMG5470.JPG` and perform camera calibration using Zhang's Method. **For this part you can use inbuilt cv2 functions**.
   2. Using the estimated camera parameters compute the image points and overlay a wireframe over the actual image of chessboard using straight lines between the computed points. Refer to `example-wireframe.png` for reference. What do you observe about the overlay?
      Note that do not use the image points found using the cv2.findChessboardCorners for wireframe overlay. Size of each square on checkerboard 29mmX29mm.

3. **DIY**
   Perform the above DLT using the an image taken by your camera. Use a calibration object for which you can measure the world co-ordinates.
   Try the RANSAC based variant as well, for the same, and compare results.