Exercise 10 for Computer Vision WS19/20
Submission on 11.01.2019

January 6, 2020

1. **Key-point Localization:** Using the given set of 14 checkerboard images, each containing 7 × 10 key-points, detect and display them in each image using the following commands:

   - findChessboardCorners()
   - cornerSubPix()
   - drawChessboardCorners()

   (4 Points)

2. **Camera Calibration:** Assuming that the top-left key-point of the checkerboard lies at the origin and in z = 0 plane; such that the 3D coordinates of the top-left and the bottom-right key-points are (0, 0, 0) and (9, 6, 0). Compute and print on console the

   - cameraMatrix and distortionMatrix
   - rotation and translation matrices for each input image

   Use calibrateCamera() and **note** that the cameraMatrix is of type double and of size 3 × 3.
   (4 Points).

3. **Reprojection Error:** In each input image, compute the 3D coordinates of each key-point. Visualize the 2D location of each key-point (in green) and their 2D reprojection (in red) by overlaying it on the image. Also, compute and print the reprojection error (ϵ_x, ϵ_y) where

   $$\epsilon_i = \frac{1}{NK} \sum_{n=1}^{N} \sum_{k=1}^{K} |p_{nk}^i - q_{nk}^i|, \quad i \in \{x, y\} \tag{1}$$

   where N is the total number of images, K is the number of key-points, p_{nk}^x is the x-coordinate of the k^{th} key-point in the n^{th} image and q refers to the corresponding reprojected key-point.
   (2 Points)

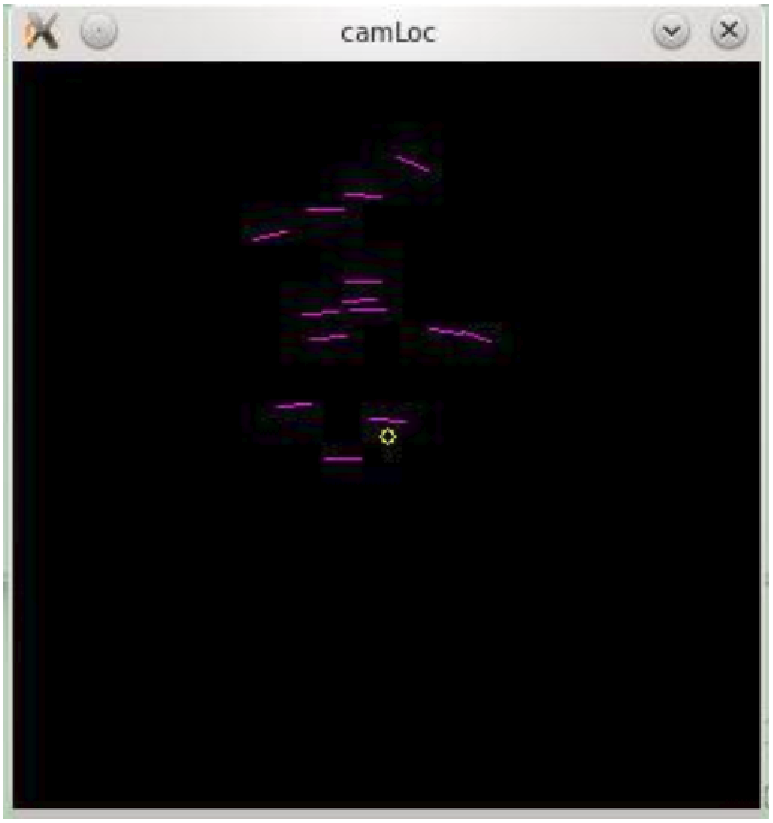

Figure 1: Relative positions and orientations of image planes w.r.t the camera coordinate system

4. **Image Undisortion:** Convert each image to grayscale and compensate for lens distortion using undistort(). Display the absolute difference between input-output pair of images.
   (2 Points)

5. **Chessboard Locations:** Display the top view of the position and orientation of image planes (by lines) relative to the camera coordinate system (origin shown by a circle) as shown in the figure 1. You may scale and translate coordinates for display purposes.
   (8 Points)