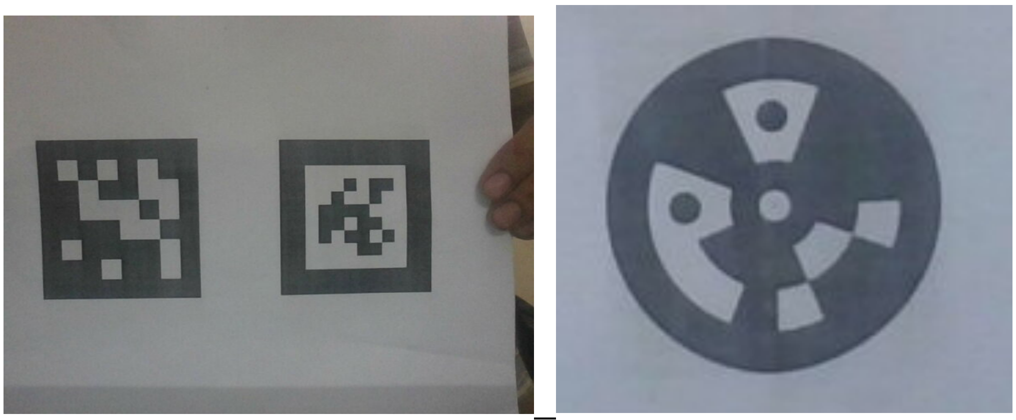
Computer Vision: Assignment 3

By
Lakshay Saggi: 2017CS50412
Siddhant Mago: 2017CS50419

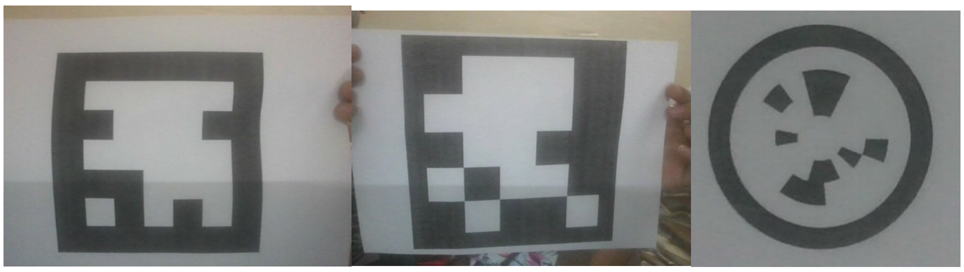
1. We did camera calibration using a video, we held a chessboard at different positions, to get the calibration matrices for our respective webcams. Our code can also calibrate using photos, however, we found calibrating using a camera easier as well as more accurate.
   The reprojection error calculated from video was around 0.19, while from the photos was 0.5.

2. We used several different visual markers to get as stable results as possible. The markers on which we obtained the best results were:
   - Best markers:



These 3 markers produced the best results, but we also tried some markers which had too many features similar to each other, which prevented us from being able to detect them separately which lead to problems in part 4.

   - Some not so good markers



These markers were being detected individually but when used together in part 4, they didn't produce the best results because we used feature detection which was not able to distinguish between these kinds of similar markers.

3. We used the ORB feature detector for detecting the marker. We also tried other feature detectors but some like SIFT, were not fast enough while some like Fast FREAK did not produce accurate results. After detecting the features, we computed a homography between the actual frame and the photo of the marker taken previously. We also know that:

   Projection matrix (P) = A H
   Where A: camera calibration matrix
   H: homography

   So, we can get the pose/projection matrix by multiplying A with H in each frame. In the video shown (in the output videos), we have projected a fox on top of the marker.
   Link to videos:
   https://drive.google.com/open?id=1DNLv2k0eQ5_I6Qm4r20yeoyRlY1MDk44

4. In this part, we detected two markers separately, found the center points of both the markers in 2-D, and then projected them using their respective projection matrices in the 3-D space. We then moved our object (fox in this case), from the center point of the source marker to the destination marker.
   Moving the fox required some thinking, we progressively increased the fox's coordinates and then took its projection on the frame. In our case, we give an input back-ground image which contains the two markers, and our fox moves on this image.

   Link to our two videos is:
   https://drive.google.com/open?id=1DNLv2k0eQ5_I6Qm4r20yeoyRlY1MDk44

   Dealing with perpendicular markers or markers at an angle is not easy as they are not easily detected using ORB, so we took photos of the marker at approximately 50^o, which allowed us to detect markers even at an angle. It could be further extended to taking a photo of markers at different angles like 15, 30, 45 etc. which leads to easy detection of the markers.

5. We have reflected the fox only (:P), in this part. Reflection is carried out only in the 2-D plane. We have assumed an imaginary vertical line passing through the center of the markers about which reflection takes place. The green line represents the net. A player cannot hit the fox from the opponent's side of the net.

   We have also progressively increased the speed of the fox as the game progresses. We have given, an initial velocity in the x and y direction, we then reverse the speed in the x- direction on each collision with the center of the marker. It could be extended, to allow us to play ping pong in all possible directions, however, for that we require more sophisticated marker detection logic.

   Link to videos:
   https://drive.google.com/open?id=1DNLv2k0eQ5_I6Qm4r20yeoyRlY1MDk44