Perception for Autonomous Robots

Project 1

AR-Tag Detection and Augumented Reality Application

Pranali Desai (116182935)

Nikhil Mehra (116189941)

Sayan Brahma (116309165)

Contents

List of Figures

1 Introduction

1. This project focuses on detecting a custom AR Tag in different videos. Four videos are taken as input and AR tags have been detected from that.

2. Homography matrix has been found using a self created function, once the corners of the AR Tag has been found.

3. Homography matrix has been further used to get the AR-Tag upfront and calculate the projection matrix.

4. For bringing AR-Tag upfront we have created a new image, by assigning the necessary pixels from the frame.

5. Further this AR-Tag has been used to get its binary value (tag id value) and also its orientation.

6. Inverse of Homography has been used to place the Lena image on the video frames.

7. Projection matrix has been calculated using a self created function.

8. Projection matrix is further used to project points and lines in the space to draw the virtual cube.

2 AR-Tag Detetction

1. Initially Gaussion Blur of kernal 7x7 is applied to the frames.
2. Then the blurred frames are converted into grayscale.
3. A threshold is applied on the grayscale image from 240-255.
4. Contours are found using the findcontour function on the thresholded frame.
5. Using the arcLength and approxpolydp function the corners of all the contours are estimated.
6. To check the real corners compensation is applied by checking which set of corners have the black border inside them due to the AR-Tag. Area of the contour was also a parameter used for rejecting unwanted contours.
   (a) The pixels around the corner are looked up so as to determine if the corner is top left, top right, bottom left or bottom right.
   (b) The main concept used is that the inner pixels of a contour should be black(borders of AR-Tag).
   (c) Apart from the tag contours all other contours have been eliminated and the corners detected have been used for further processes.
7. Thus we get only one set of corners for each AR-Tag present in the video, and hence we have successfully extracted the AR-Tags from the video.

3 Converting Tag into Matrix

1. After Homography we have the tag id upfront i.e. extracted from the video input.
2. Once we have the upfront images of the AR-Tags, we divide it into images containing 64 equal squares.
3. We further convert those 64 equal squares into a 8x8 matrix by determining whether the square boxes contains maximum number of white pixels or black.

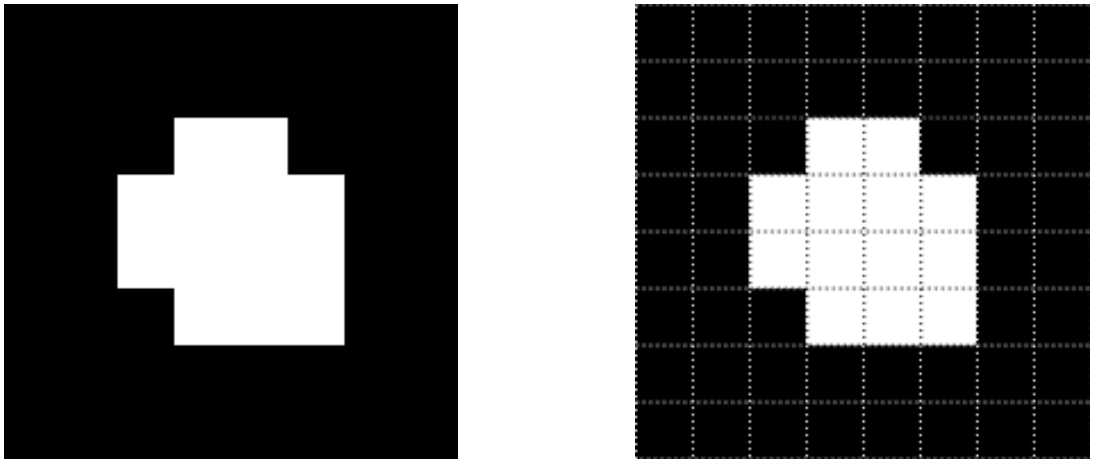

Figure 1: AR Tag - 8x8

4. Here the 1 represent white colour and 0 represents black colour.

4 Finding the Orientation and Binary Value

1. Once we have the 8x8 matrix, we try to find the position of the white box present in the corner of the 4x4 matrix, which will give us the orientation of the tag.

2. The position of the white box determines how much has the tag been rotated.

3. Once we have the angle of rotation we find the value of tag ID through the 2x2 matrix in between.

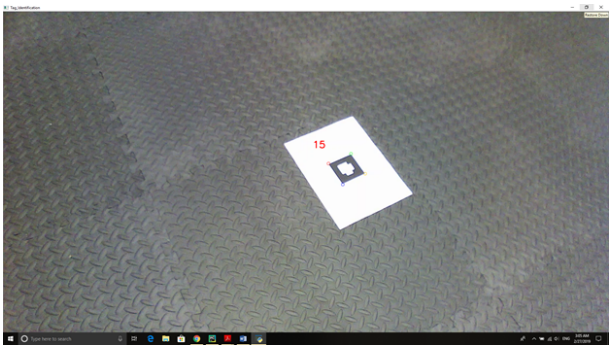

Figure 2: Printing Corners and Tag Id for Tag-0

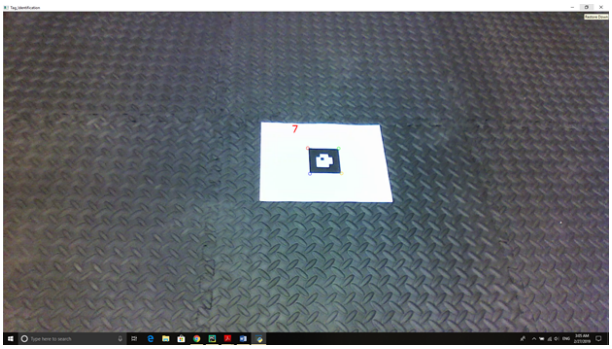

Figure 3: Printing Corners and Tag Id for Tag-1

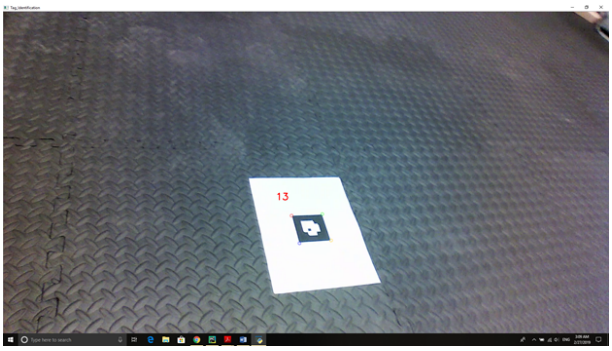

Figure 4: Printing Corners and Tag Id for Tag-2

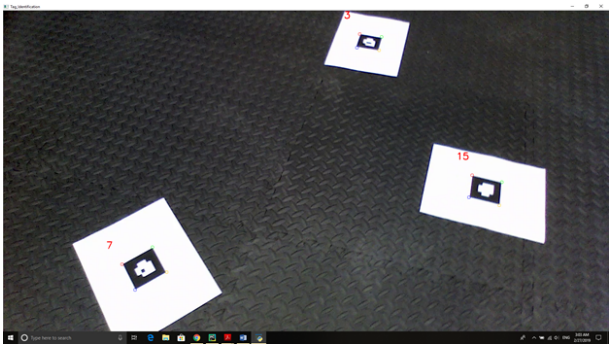

Figure 5: Printing Corners and Tag Id for multiple Tags

4. We get the tag in the actual rotation and check the inner 2x2 value for zeros and ones.

5 Placing Lena Image on the tag

1. The corners of the AR tag in the videos from section-1 are used in this section.

2. The corners are updated according to the orientation of the AR-Tag.

3. A homography function is defined to estimate the homographic matrix between the given template, image of Lena(world frame) and the coordinates of the corners of the AR tags(camera frame), based on the relations and equations provided in the supplementary reading file.

4. Using inverse homography the camera frame coordinates corresponding to the world frame could be determined. Hence applying inverse homograpgy to each pixel of the template, the corresponding pixel of the camera frame could be determined.

5. After finding the relation between the camera frame and world frame, the pixels of camera frame are simply replaced by the corresponding pixels of the world frame.

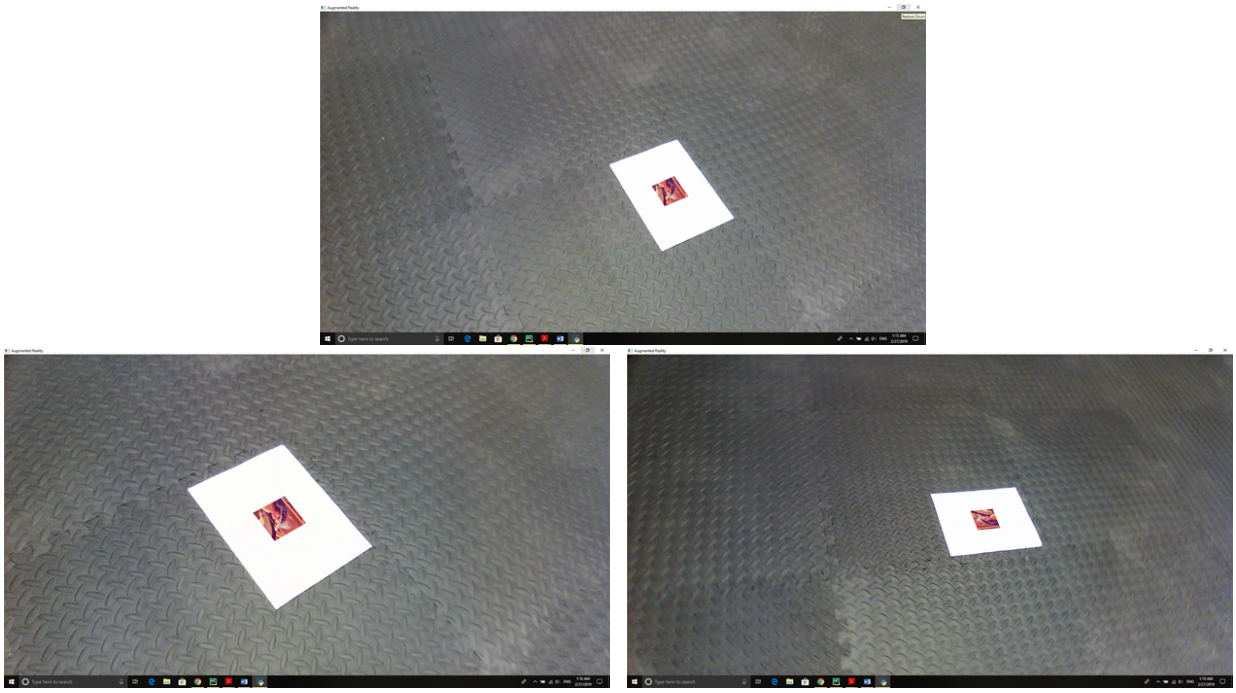

Figure 6: Placing Lena on Tag-0 with different frames of camera

Figure 7: Placing Lena on Tag-1 with different frames of camera

Figure 8: Placing Lena on the Tag-2

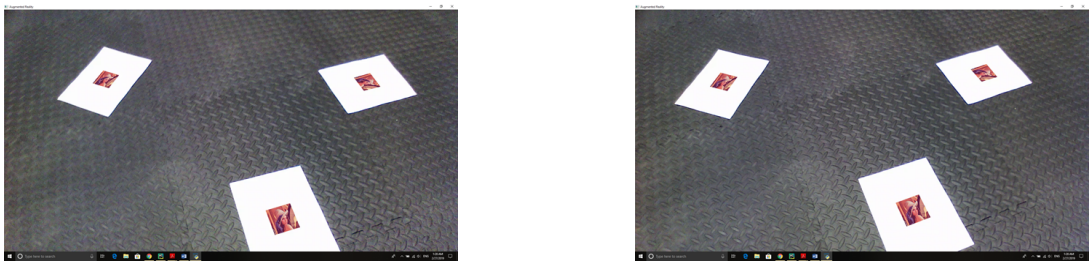

Figure 9: Placing Lena on Multiple Tags with different frames of camera

6. The orientation of the image/template matches to that of the AR tag which is clearly observed in the video since there is no flip or disorientation of the image with the changing orientation of the camera.

6 Placing a virtual box on the tag

1. The objective is to place a virtual cube on the AR tags in the video by the process of projecting a 3-D shape on to a 2-D image.

2. The cube consists of 8 corners and 12 corresponding edges joining them. Out of the 8 corners and 12 edges, 4 corners and edges lie on AR tag. These four corner points can be written as (0,0,0), (0,200,0),(200,0,0) and (200,200,0) in the world frame.

3. As the positive Z axis is assumed in the downward direction the other four corners could be considered as (0, 0,-200), (0, 200,-200), (200, 0,-200) and (200, 200,-200).

4. First, the homography between the AR tag from the video in the camera frame and the world frame is calculated. The corners of the AR tag are considered as camera frame and the corners (0, 0, 0), (0, 200, 0), (200, 0, 0) and (200, 200, 0) are considered as the world frame. In order to calculate homography a user-defined function homography is called.

5. Secondly, in order to project 3-D points from world frame to camera frame, we require a projection matrix. For calculating projection matrix, a user defined function projectionMatrix is created, which consists all necessary mathematical equations. projectionMatrix requires camera calibration matrix and homography matrix as its two arguments.

6. Then the 2-D camera coordinates are calculated using 3x4 projection matrix and 3-D world coordinates [(0,0,0), (0,200,0), (200,0,0), (200,200,0), (0, 0,-200), (0, 200,-200), (200, 0,-200) and (200, 200,-200)].

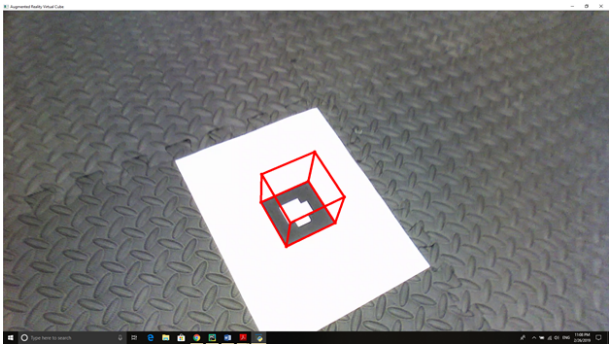

Figure 10: Placing a virtual box on the Tag-0

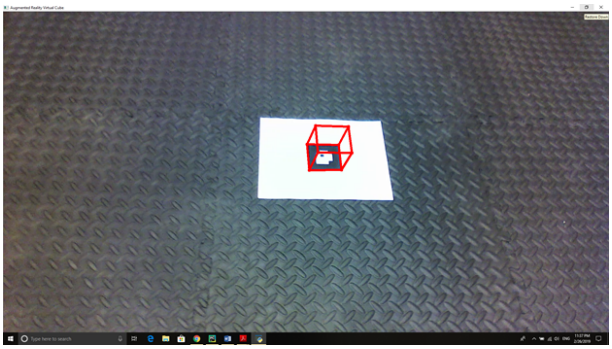

Figure 11: Placing a virtual box on the Tag-1

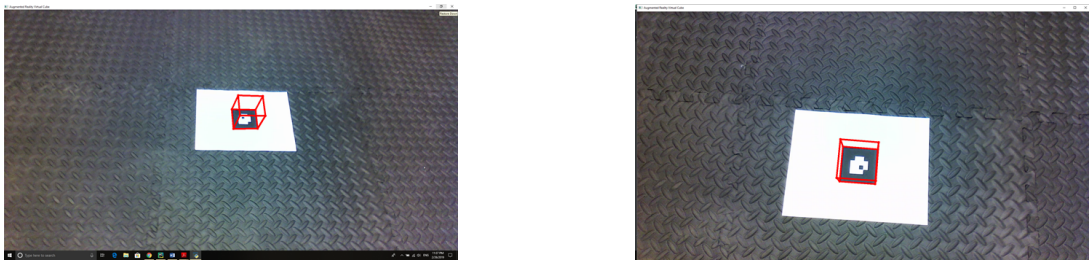

Figure 12: Placing a virtual box on the Tag-2 with different frames of camera

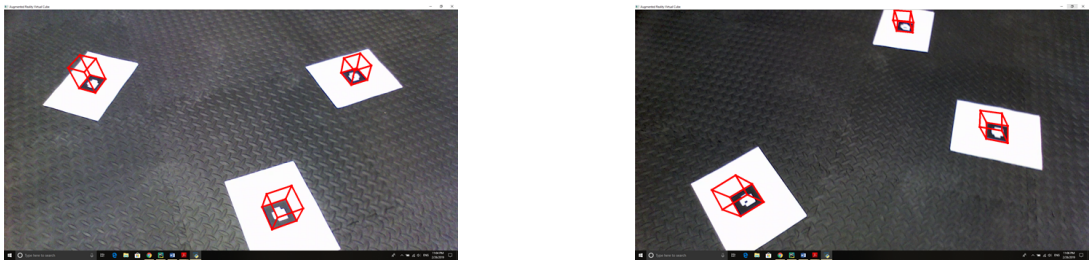

Figure 13: Placing a virtual box on the Multiple Tags with different frames of camera

7. The formulas and equations required for defining the homography and projectionMatrix functions are referred from the supplementary provided.

7 Results and Conclusions

1. All the three parts of question statement have been achieved.
2. We faced problems for finding the real corners but we were succesful after checking several aprroaches.
3. We also noticed that without using the warp perspective function the frames are quites slow to be processed since every single pixel has to be multiplied.
4. Screenshots of all the videos at different frames and camera angles have been attached.