ENPM673 Project1
Perception of Autonomous Robot
Detection of a custom AR Tag

By
B. Sai Praveen(DirID: spraveen)
Preyash Parikh(DirID: pparikh)
Ajinkya Parwekar(DirID: ajinkyap)

February 25, 2020

INTRODUCTION

The objective of this project is to detect a custom AR tag which is used for obtaining a point of reference in the real world applications such as augmented reality. The custom AR tag is given in a video. After detecting, the Lena image has to be placed on the AR tag. This project uses concepts like Homography and Computer Vision.

Implementation

After reading a video frame by frame, we performed tag detection and projection of Lena image on detected tag. Later, cube is projected using projected point coordinates.

1 Problem 1: Detection

We chose to implement $findcontour()$ function of OpenCv. There are many ways to have contours in different manner such as RETR LIST, RETR TREE. Since we were looking for child parent hierarchy, we chose to go ahead with RETR TREE. The reason of selecting child parent hierachy is we wanted to know which contour is inside which contour always. Before using this function, we applied Gaussian filtering of kernel size 7x7 and then converting this image to Grayscale. To find contours, thresholded image is required. Also, there are two ways for approximation method in $findcontour()$ and we have selected CHAIN APPROX SIMPLE.

After generating hierarchy list which have information of parent contour at 3^{rd} index column list we filter out contours which do not have a parent, as these would constitute of outermost contours which we don't need. To capture right contour we implemented an area check. The value of area check is decided by experimenting. For each frame, we draw these contours using red color to display the area selected or detected in that frame.

2 Problem 2: Tracking

2(a) - Superimposing an image onto the tag

After we find the 4 corners of the tag, we need to perform Homography estimation so as to perform image processing operations like superimposing an image over the tag.

We selected the coordinates of Artag in world frame to be 200×200. With the selected coordinates and corners of detected tag we calculate homography matrix between the world frame and camera frame. Since tag is present in 2D frame in world frame we dont need R3.

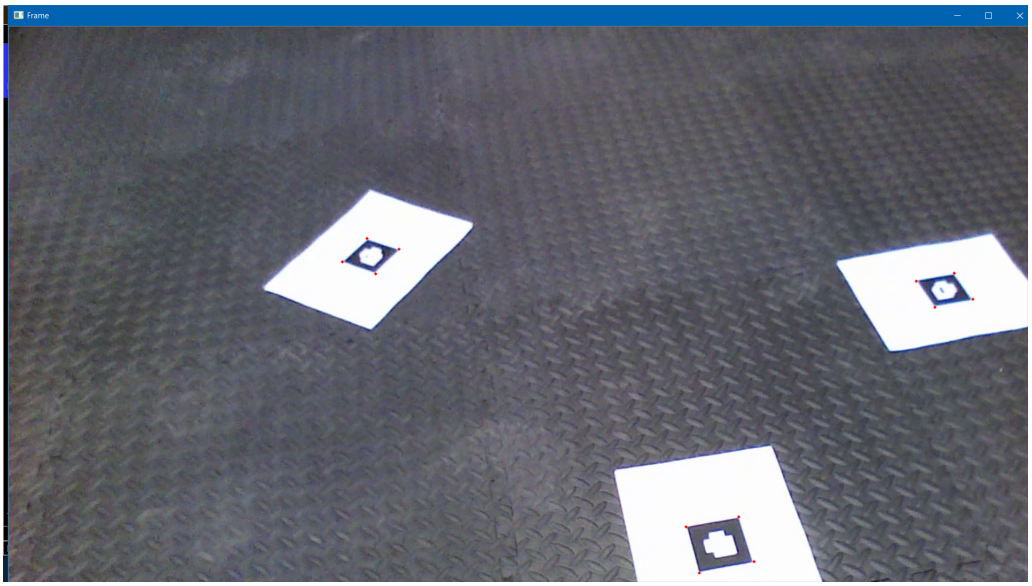

Figure 1: Detected Contours

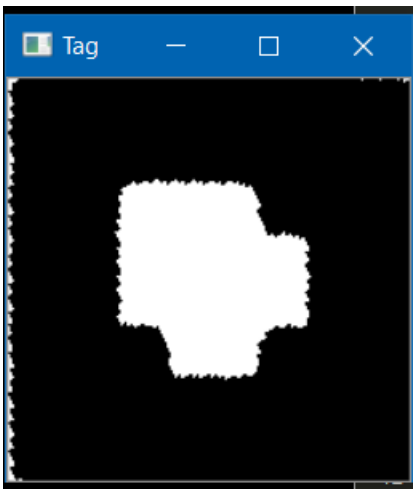

Figure 2: AR Tag Detected

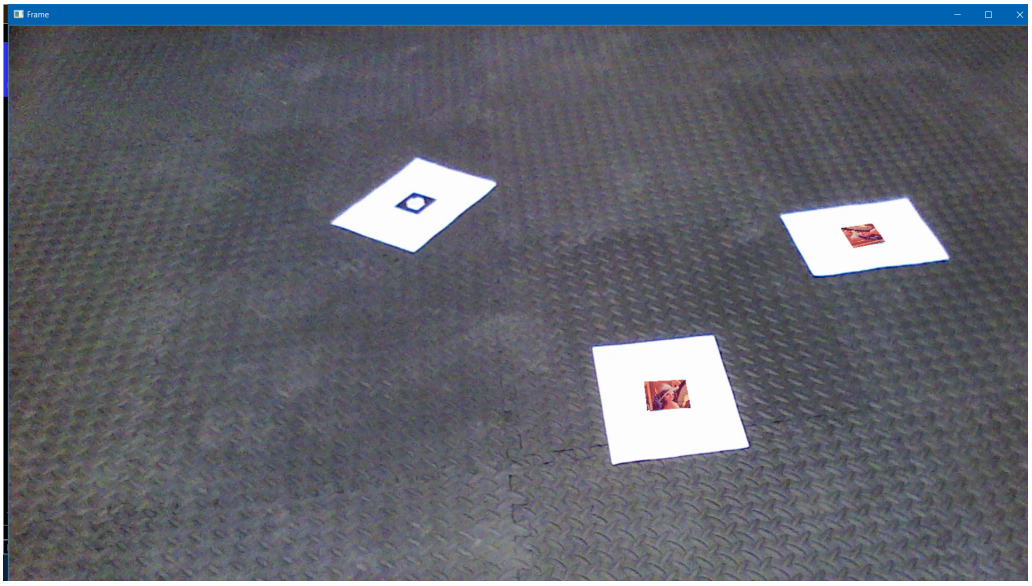

Figure 3: Image of Lena placed on detected AR Tags

Thus, homography matrix is given by [R1 R2 T]

Calculation for Homography

The homography is calculated by first defining a A matrix as :

Ah = $0_{8\times1}$

Now we decompose it via Singular Value Decomposition to construct separate U,S and V matrices.

The V matrix is then used to calculate the "h" vector

$$H = \frac{(V_{19}, ..., V_{99})^T}{V_{99}} \tag{1}$$

The obtained H matrix is then resized to generate the required Homography matrix.

2(b) - Placing a virtual cube on the tag

We first define the coordinates of a cube in the world frame and calculate the transformation matrix. The first step is to calculate a matrix which is the product of the homography matrix and camera matrix. The resulting matrix's first and second columns represent the first two columns of the rotation matrix (r1 and r2). We then calculate the r3 column of the rotation matrix by taking the cross product of r1 and r2. The last column is the required translation marix. For scaling the transformation matrix we use lambda. Lambda is scaling factor.

$$\lambda = \left(\frac{||k^-1 * h_1|| + ||k^-1 * h_2||}{2}\right)^{-1}$$

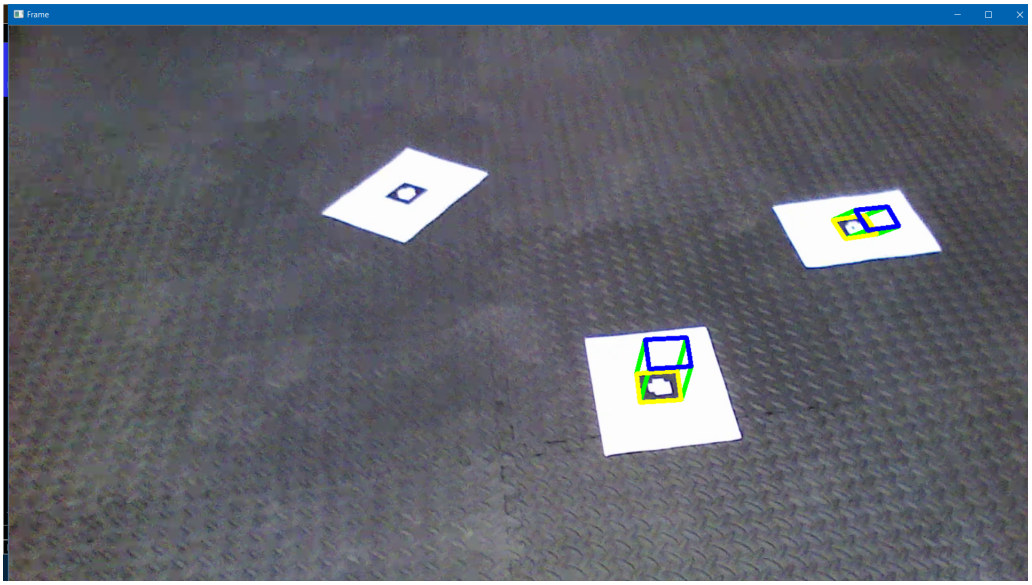

Figure 4: Virtual Cube placed on detected AR Tags

After using the data to calculate the projected coordinates, the data is fed into function which in turn is used to draw the cube on the tag. Two Pair of x and y coordinate is divided individually by the z coordinate and the integer value is fed into cv2.line function to draw the separate faces of a cube. The colour code can be varied into cv2.line function.

Challenges faced

1) Initially we used Harris detector and Shi Tomasi detector but we were not satisfied with the edge detection results. Thus, we had to find some another method for the same and we encountered "findcontour()" method.
2) By using findcontour(), after extracting child parent hierarchy we were left with lots of unrequired contours. To restrict them, we had to study which contour exactly needs to be restricted.
3) We removed unrequired contours from single tags but to remove unrequired contours from multiple tags we used concept of restricting contour area.
4) While having Artag of size 200×200 and if size of Lena image is increased, we got perfect output but the video speed gets very slow and Vice versa. As of now, we have selected the optimal values.

References

For Corners
1) https://docs.opencv.org/3.4.2/d4/d73/tutorial_py_contours_begin.html
2)https://docs.opencv.org/2.4/doc/tutorials/imgproc/shapedescriptors/find_contours/find_contours.html
3)https://docs.opencv.org/3.4/d9/d8b/tutorial_py_contours_hierarchy.html
4)https://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html

For Homography
1) https://docs.opencv.org/master/d9/dab/tutorial_homography.html
2)https://www.youtube.com/watch?v=MlaIWymLCD8&t=285s

For Lena Image
1) https://docs.opencv.org/3.2.0/d0/d86/tutorial_py_image_arithmetics.html
2) https://stackoverflow.com/questions/7589012/combining-two-images-with-opencv

For Placing Cube
1) https://www.geeksforgeeks.org/draw-geometric-shapes-images-using-opencv/
2) https://www.youtube.com/watch?v=wVPcXGGsVlM

Supplementary material based on Homography and Projection Matrix provided for course ENPM 673 at University of Maryland,College Park