ENPM 673
Perception for Autonomous Robots

Project 1 Report

Akshay Bapat (UID: 116215336)
Patan Sanaulla Khan (UID: 116950985)
Kulbir Singh Ahluwalia (UID: 116836050)

26 February 2020

Problem 1: Tag Detection

Contours

A curve joining all the continuous points, with same color or intensity such that it can help analyze the shape of the object or to identify the object. This method is usually more efficient when used with images in grey scale. Therefore to obtain a better shape analysis the image is applied a threshold of value (240, 255, 0).

This method requires 3 parameters. The first parameter is the binary image which is processed with a threshold. The second parameter is cv.RETR_TREE and the third parameter is cv.CHAIN_APPROX_SIMPLE which help the method to generate the hierarchy of the contours in the image which is processed by the method. This value of the parameter help us generate the right image points to generate the homography matrix for the given image. This value can also be cv.CHAIN_APPROX_NONE which returns all the points that lie on the contour line.[1]

Listing 1: Finding the contours

```
gray = cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
ret, thresh = cv2.threshold(gray, 240, 255, 0)
contours, hierarchy = cv2.findContours(thresh, cv2.RETR_TREE,
                                       cv2.CHAIN_APPROX_SIMPLE)
```

Hierarchy

The parameter passed in the last function cv.RETR_TREE, can also be replaced with cv.RETR_LIST which help return the hierarchy of the different contours found in the image. Here the hierarchy is the order of the shapes or contours found within the contours. Hence such nested shapes are therefore placed with the values such that the tree or the list contains the hierarchy of the contours for all the shapes found in the image. We then select the images based on the needed hierarchy. [2]

RETR_TREE gives an array where all the contour's have the list of the the relation with the other contours in the image. It returns the most detailed

description of the the image contours.

Now to select the interested contour in the image of tags, we iterate the hierarchy values such that we find the contour that has no contour in same level, and has one child contour which is the shape of the tag.

Listing 2: sample hierarchy

```
array([[[ 7, -1,  1, -1],
        [-1, -1,  2,  0],
        [-1, -1,  3,  1],
        [-1, -1,  4,  2],
        [-1, -1,  5,  3],
        [ 6, -1, -1,  4],
        [-1,  5, -1,  4],
        [ 8,  0, -1, -1],
        [-1,  7, -1, -1]]])
```

Finding Corners

Now from the given set of values for the contour we generate lines parallel to the lines form the 2 points on the contour and find the angle of the corners. Once if the angle of the corners is identified as not less than 45 we use the formulations like Max of X+Y values and Min of X+Y and similarly Max of X-Y and Min of X-Y to find the corners.

Interesting Problem

If the angle of the inclination of the contour is found to be less than or equal to 45 we use the values of the Min and Max of X and Y to get the corners.

Problem 2(a) - Superimposing an image onto the tag

To superimpose the image onto the tag, we calculate the homography matrix using the corners and the contours. Once the homography matrix is found we need to warp the image using the homography matrix and superimposing it onto the output image.

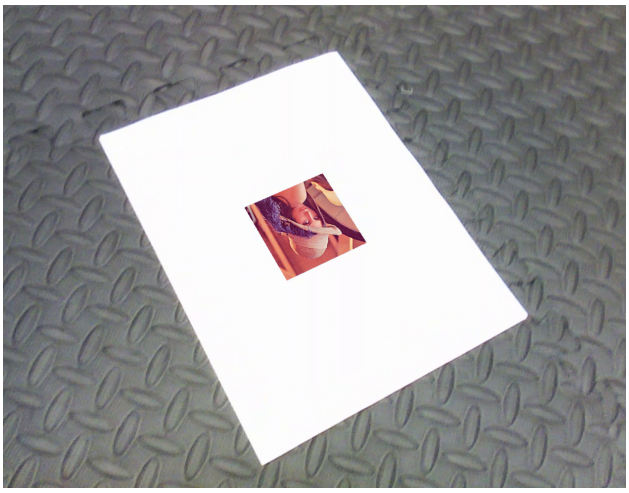

Figure 1: Lena superimposed on Tag 0

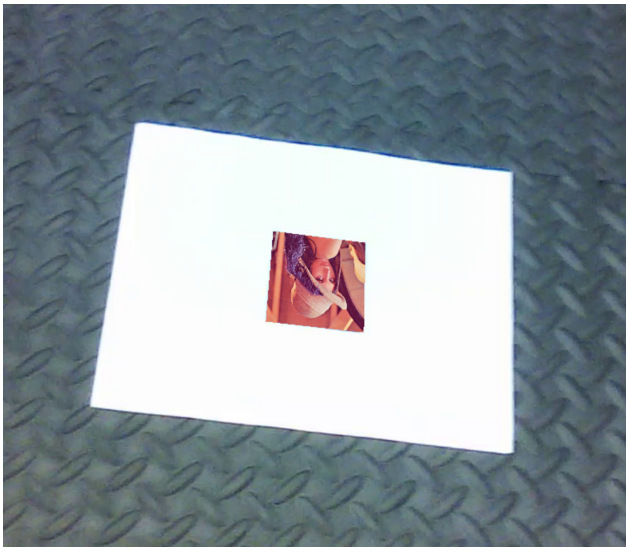

Figure 2: Lena superimposed on Tag 1

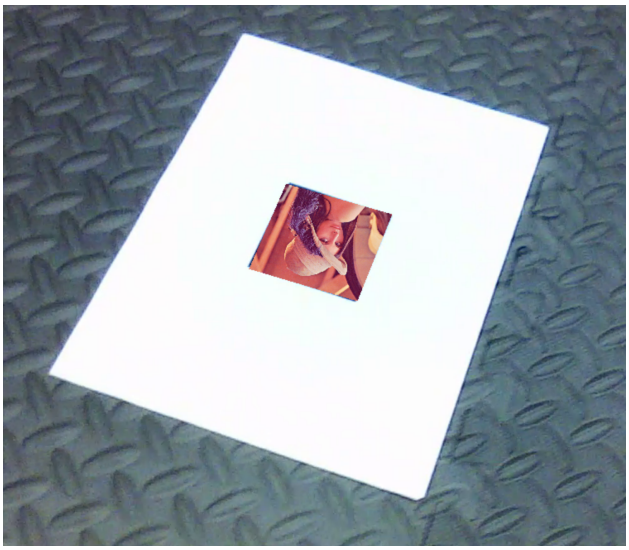

Figure 3: Lena superimposed on Tag 2

We have used the custom function **customWarp** to superimpose the image of Lena onto the AR tag.

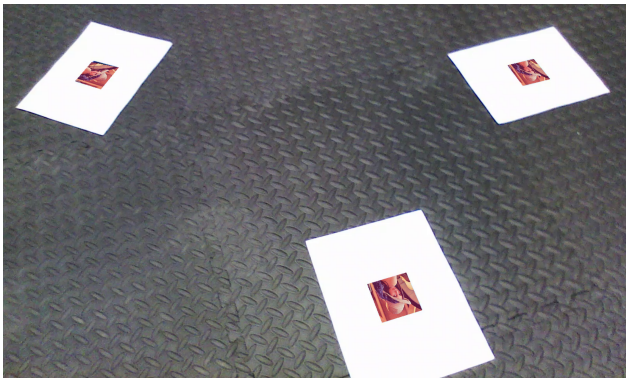

Figure 4: Lena superimposed on Tag 0, Tag 1 and Tag 2

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

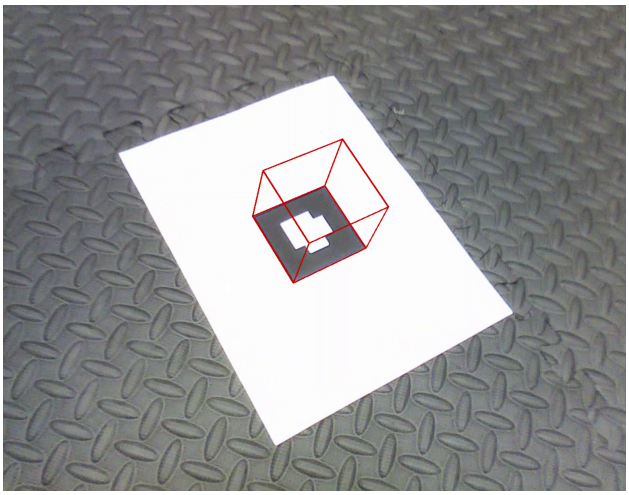

Figure 5: Cube superimposed on Tag 0

To place a cube on the tag, we first need to calculate the Homography matrix for the given points of the corner. The corners detected using contour and filtered are used as the source points and the final points are the points which are to be transformed into for the image to be in homography. Hence we find the homography matrix using the concept of SVD method where the

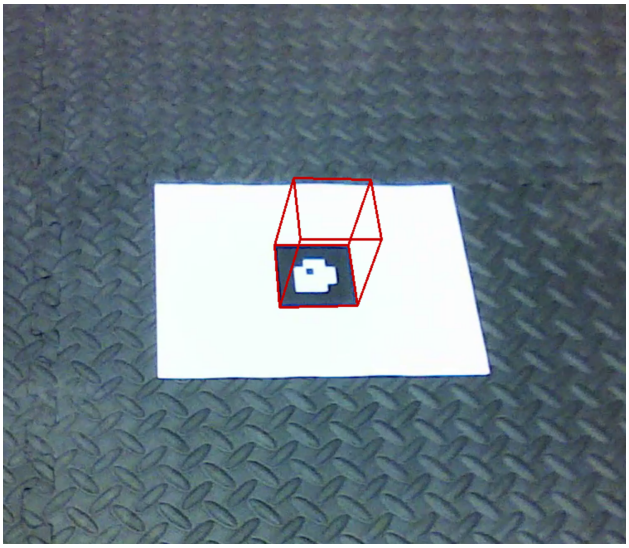

Figure 6: Cube superimposed on Tag 1

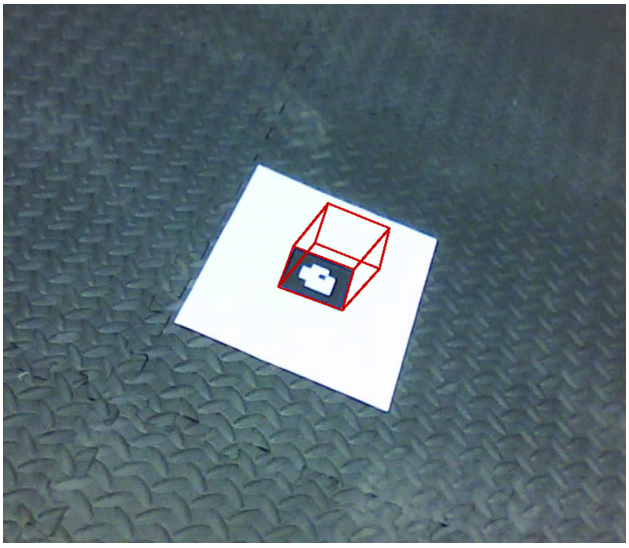

Figure 7: Cube superimposed on Tag 2

generated value of V is used. The homography matrix is 3X3.

Now this matrix along with the given K matrix, which holds the values of the intrinsic parameters given in the question is used to identify the new projection matrix for generating the cube.

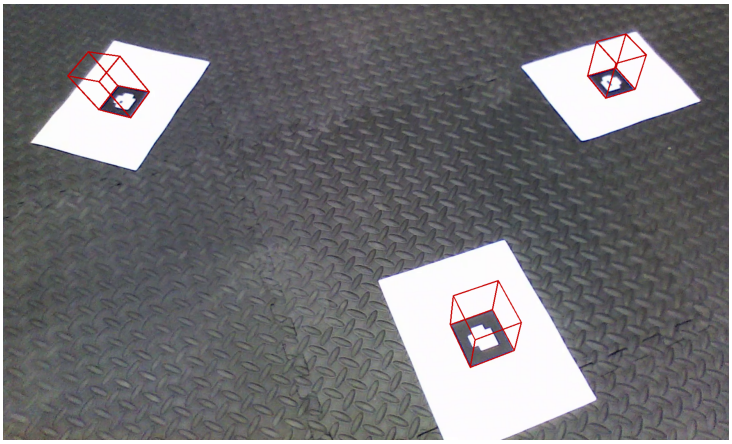

Figure 8: Cube superimposed on Tag 0, Tag 1 and Tag 2

0.0.1 Problems Faced

As the video was unstable, there are errors while estimating projection matrix and this results in unstable cubes and unstable orientation of the Lena.png image.

Bibliography

[1] Open Source Computer Vision,
https://docs.opencv.org/3.4/d4/d73/tutorial_py_contours_begin.html

[2] Open Source Computer Vision,
https://docs.opencv.org/3.4/d9/d8b/tutorial_py_contours_hierarchy.html