ENPM 673: Project 1

**Question 1:**

1. **AR Tag detection:**

The AR Tag is detected using Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT). FFT has a major advantage of easily detecting sharp edges or detect sharp edges depending on the use. Using a low pass filter, we allow only the high frequency edges to make note of. The reason we do this to detect only the major edges. This is gives us the edges on the AR Tag.

The way FFT works is it converts images into different set of frequency and then use this set to remove the background. At first, we use a FFT that converts the image into frequency domain, which can be seen in Fig. 2. Then we rearrange the frequency domain and remove the middle part of the domain, which contains low frequency. This gives us the output with only edges as shown. This is the contour found in the image and thus the AR tag is detected.

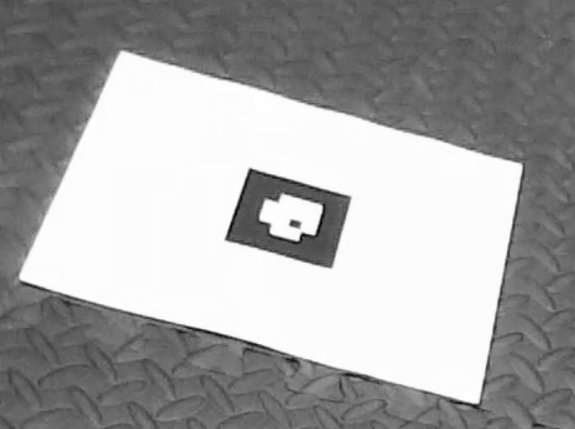


Fig. 1

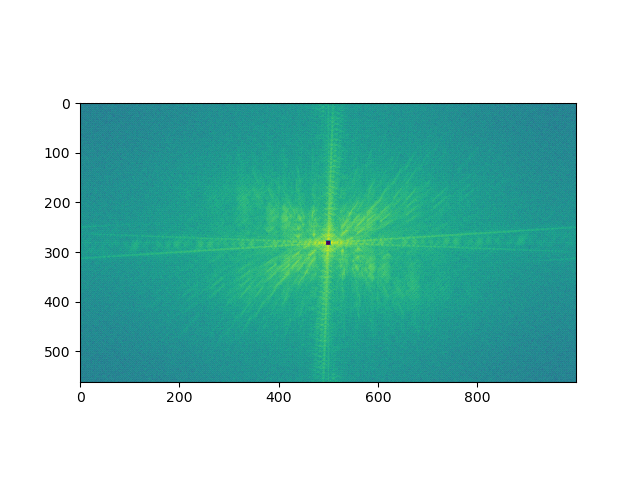


Fig. 2

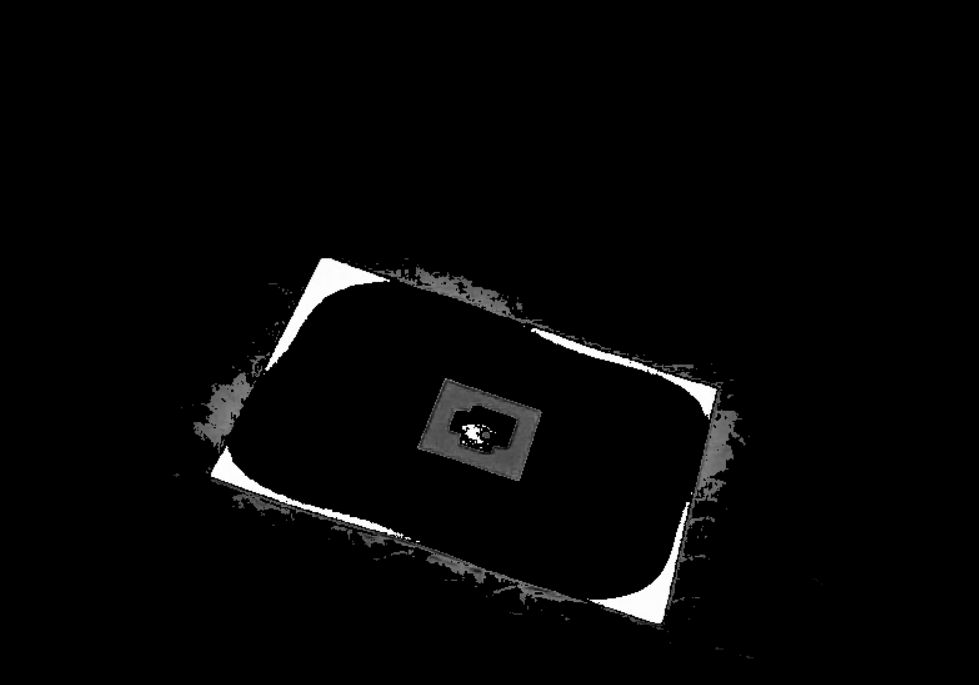


Fig. 3

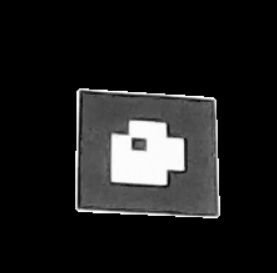


Fig. 4

1. **AR Tag decoding:**

This AR tag image is used to create an 8x8 array where ‘0’ represents black and ‘1’ represents white. This is done by sectioning the image into 8 parts. And then taking the nodes of the image in that region, we compare and check if the value is equal or greater than designated white value. And then declares the node as a white region.

The orientation of the tag is checked by checking the corner values of the inner 4x4 matrix and check in which the white block is present and assign the orientation as per the position

Taking these values into consideration, we decode the value of the image. We consider the inner 2x2 matrix for this. Required direction is from bottom left to top left.

**Question 2:**

1. **Testudo overlay:**

To transform the testudo image on the video frame, we fetch tag using the procedure used before. Then we perform homography and perspective transform. We then calculate the rotation using the procedure to find the orientation and using this value, we rotate the testudo image. Then we again calculate the homography between the image and frame tag location and perform perspective transformation:

Following is the procedure to calculate the homography matrix:

*x=HX------1*

*x:frame coordinates*

*X:image/testudo coordinates*

*H: Homography*

*x=src\_pts[:,0]*

*y=src\_pts[:,1]*

*xp=dest\_pts[:,0]*

*yp=dest\_pts[:,1]*

*A=np.zeros((8,9))*

*for i in range(4):*

*A[2\*i,0]=-x[i]*

*A[2\*i,1]=-y[i]*

*A[2\*i,2]= -1*

*A[2\*i+1,3]=-x[i]*

*A[2\*i+1,4]=-y[i]*

*A[2\*i+1,5]= -1*

*A[2\*i,6]=x[i]\*xp[i]*

*A[2\*i,7]=y[i]\*xp[i]*

*A[2\*i,8]= 1\*xp[i]*

*A[2\*i+1,6]=x[i]\*yp[i]*

*A[2\*i+1,7]=y[i]\*yp[i]*

*A[2\*i+1,8]= 1\*yp[i]*

*U,S,V=np.linalg.svd(A)*

*H=V[8].reshape((3,3))*

*#H=V[(np.argmin(S))].reshape((3,3))*

*H=(1/H[2,2])\*H*

Then we substitute this value in the equation 1 to get the frame coordinates for transforming the testudo image.



Fig. 5:

1. **Cube projection:**

For cube we multiply the homography matrix with the K matrix to find the transformation matrix. This matrix gives us ‘r1’, ‘r2’ and ‘T’. r1 and r2 constitute the first two columns of the rotation matrix. The third column is calculated by doing a cross-multiplication of these two columns. These values are also required for the transformation matrix. We then calculate λ. This is the scaling factor required for projection. We use these values to find the coordinates of cube on the frame. And then draw the cube.

Link for the video

testudo\_ar\_tag : <https://www.youtube.com/watch?v=HG2xJUl2tMU&ab_channel=abhilashmane>

AR\_cube:

<https://www.youtube.com/watch?v=0Uus9UQvn98&ab_channel=abhilashmane>