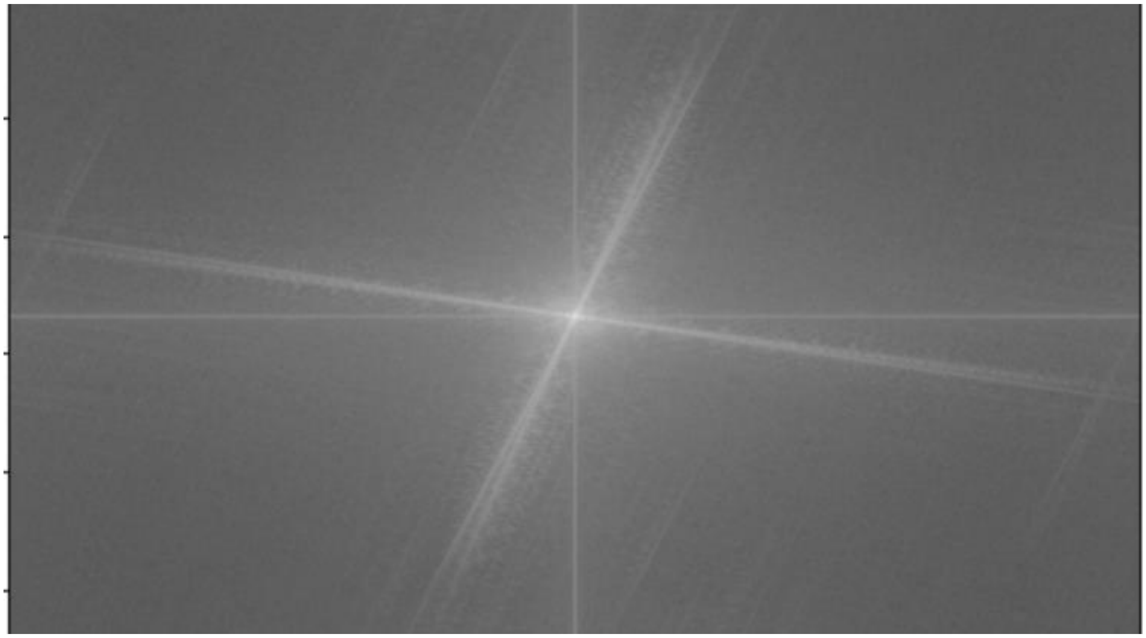
ENPM 673 PROJECT 1

Problem 1 – Detection

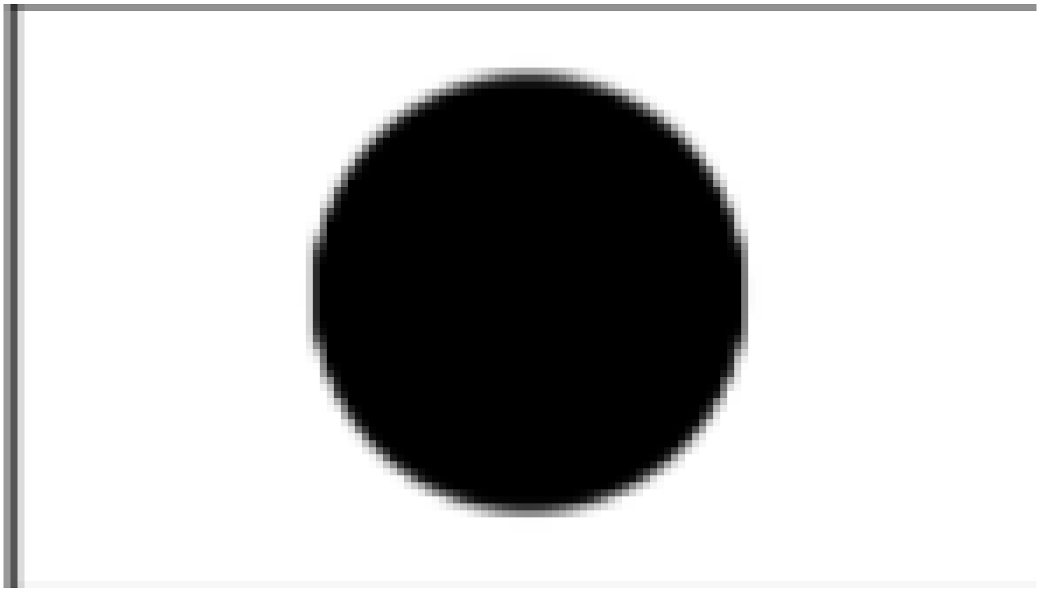
a) AR Code detection:

To detect the AR Code FFT was used. We know that the edge of an image has higher frequency signals. Hence, we used FFT to differentiate the AR code from background image and to do so following steps were used:

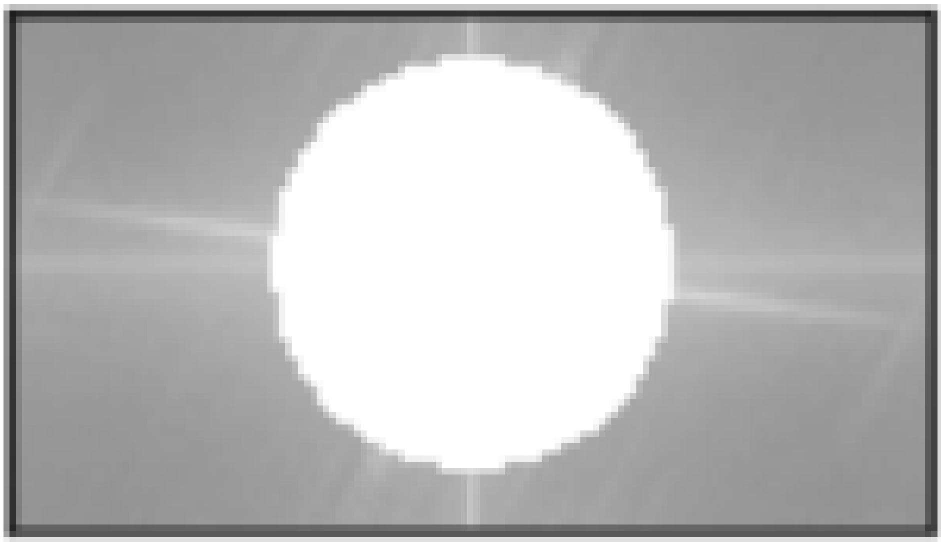
1. Convert the image into grayscale and compute FFT

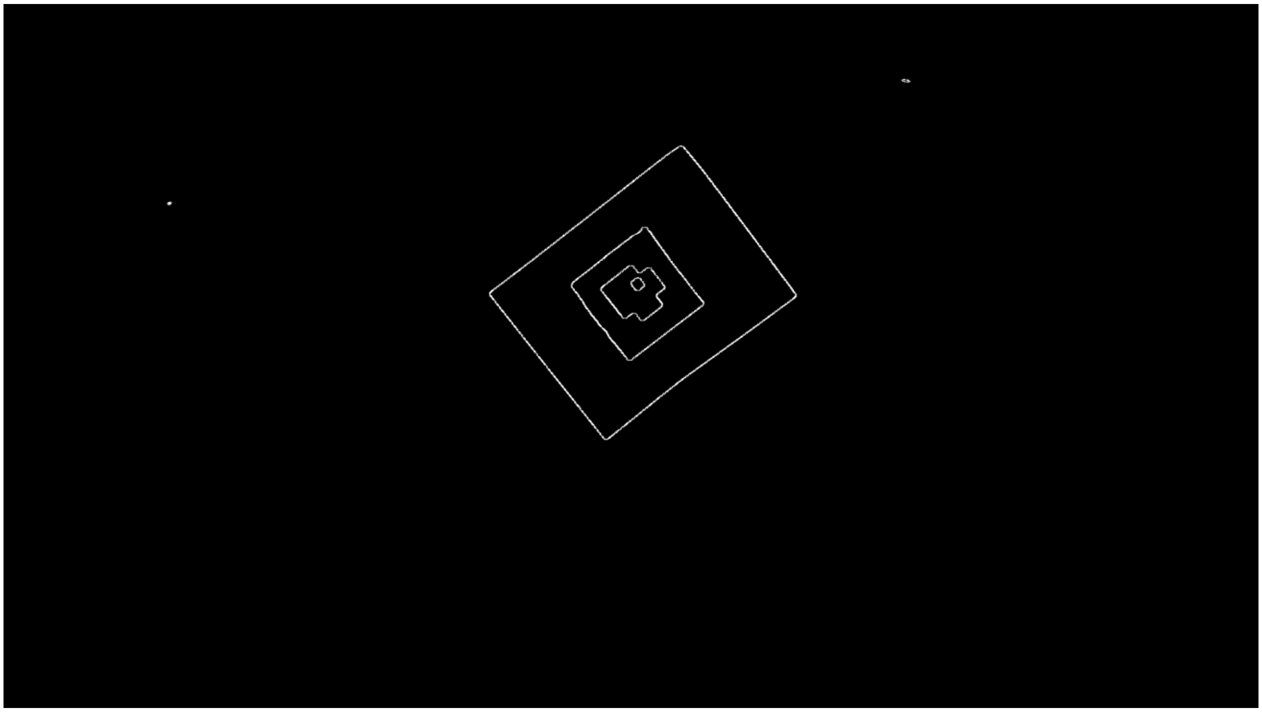


2. Create a Mask, a high pass filter mask was created. It was created by making a black circle at center and keeping the rest as white so that only high frequencies get captured which comprises of the change in intensity at the edges. The mask was multiplied with the image to filter out the low-frequency signals

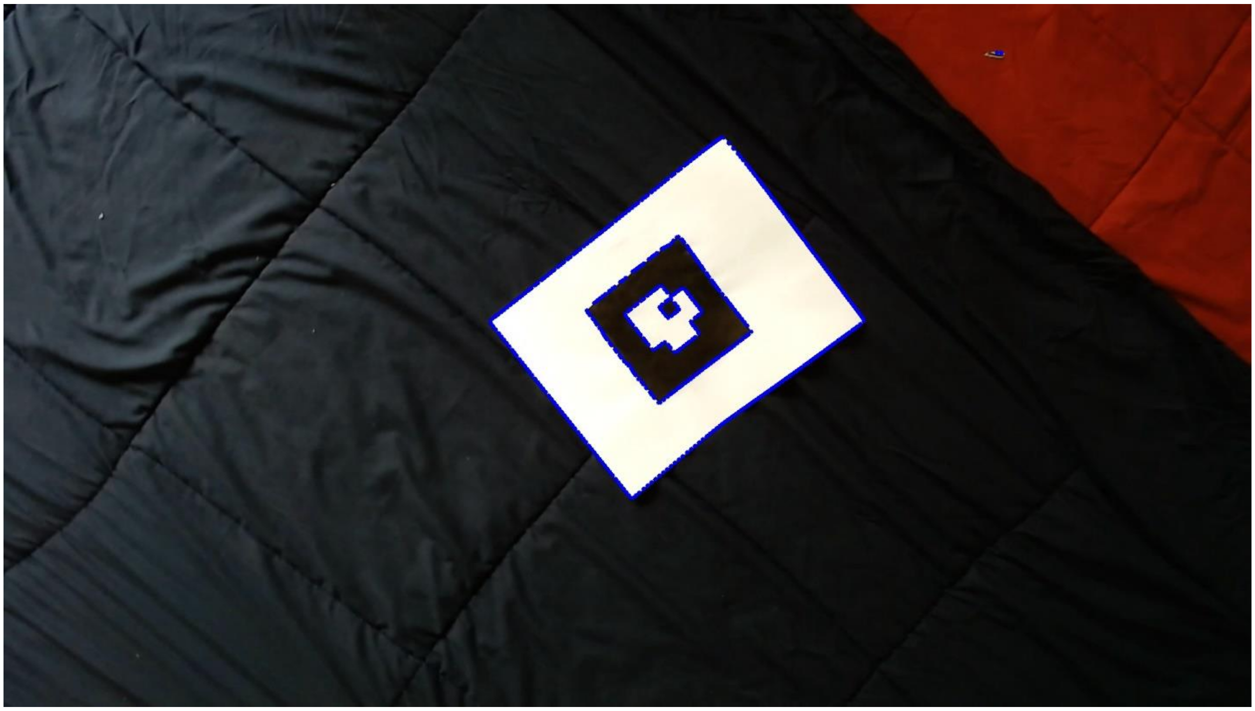


3. Compute the inverse FFT to get the image with edges and apply thresholding for better results.

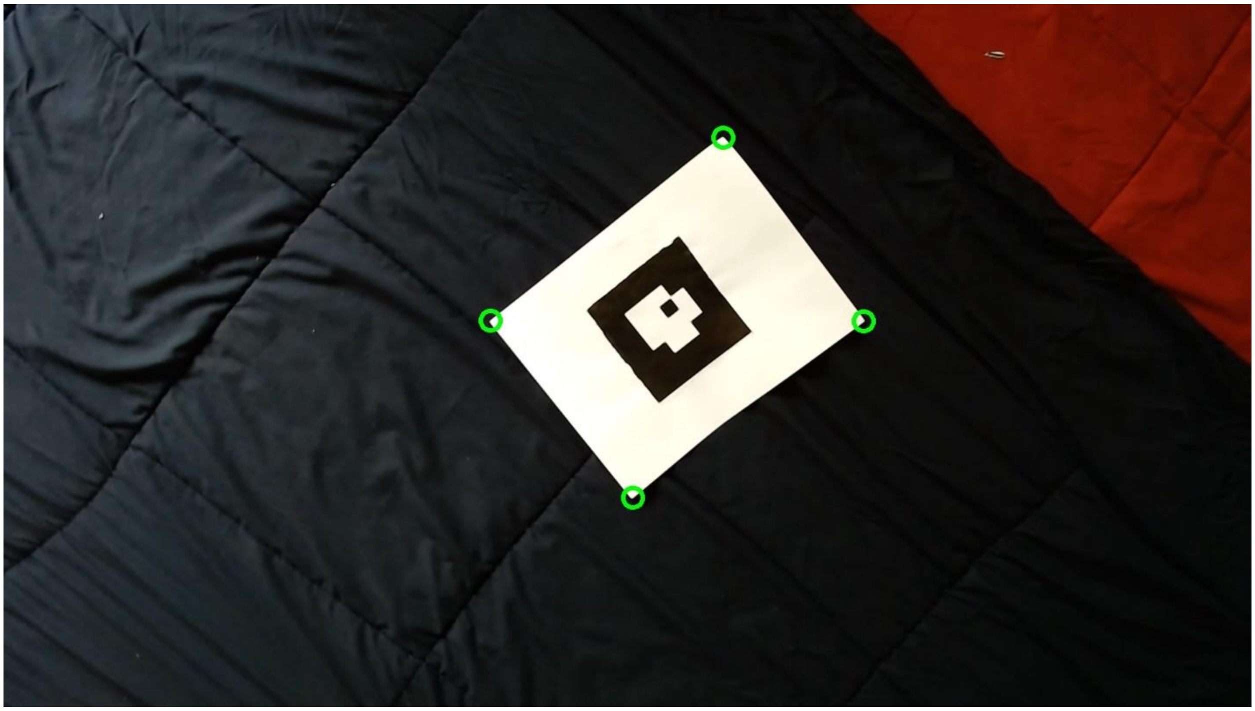




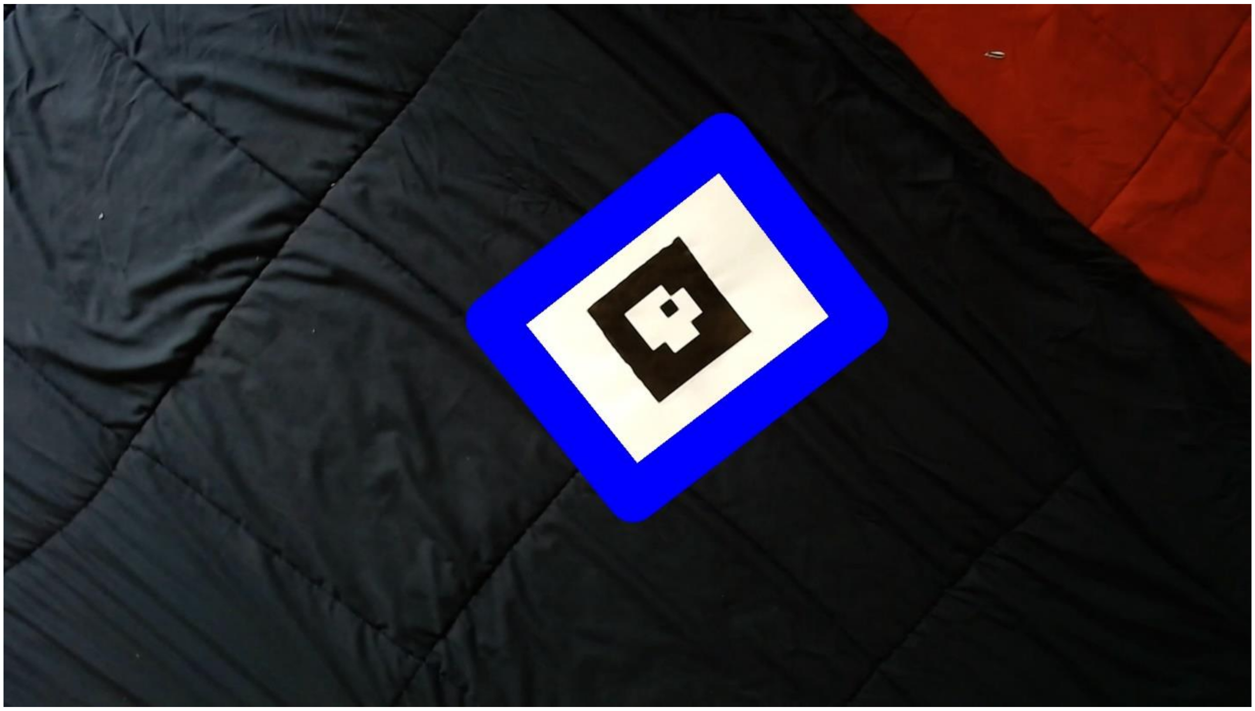
4. Apply Shi Tomasi corner detection **cv.goodFeaturesToTrack()** to detect the all possible corners.

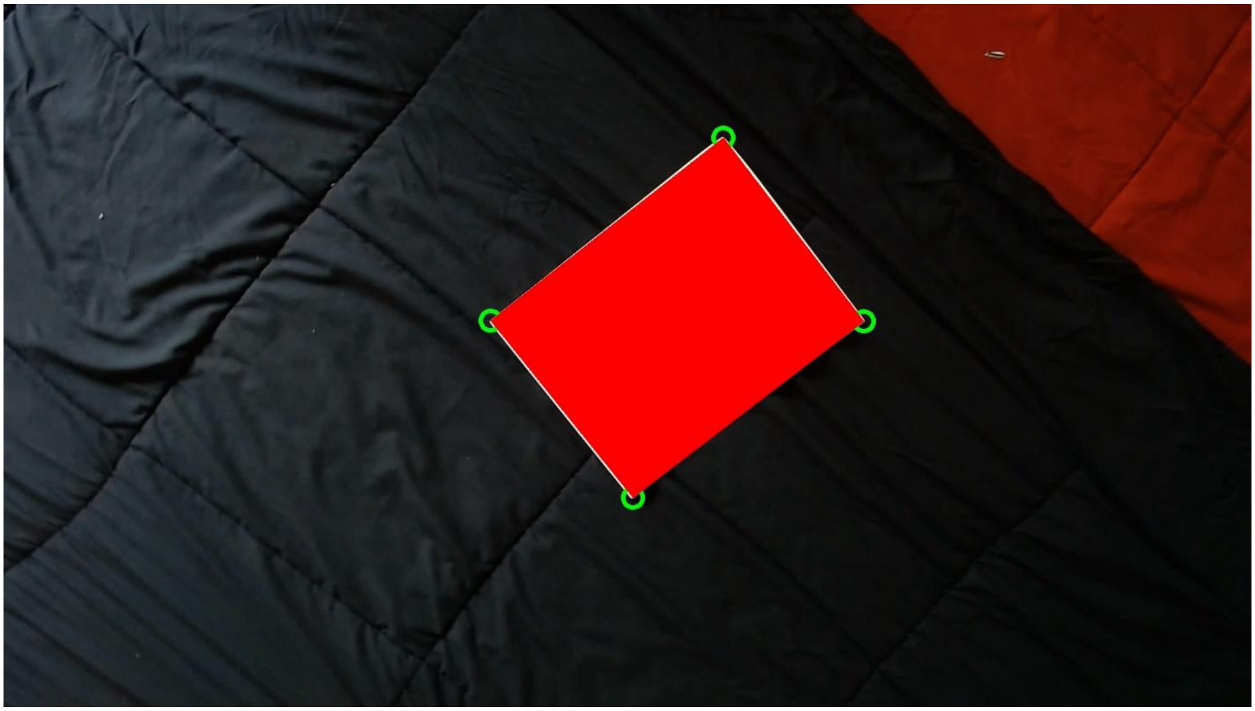


5. After getting the corners in the image there were some outliers The outliers were removed by using the z score from stats module from scipy library.
6. Once we get all the corner points the extreme corners were found out.



7. Find the next 4 extreme corners available which satisfies both the mask. The 4 corner points of AR Code are detected. There are two mask blue color and red color to check that the inner corners are not mismatched, and they lie inside the page.

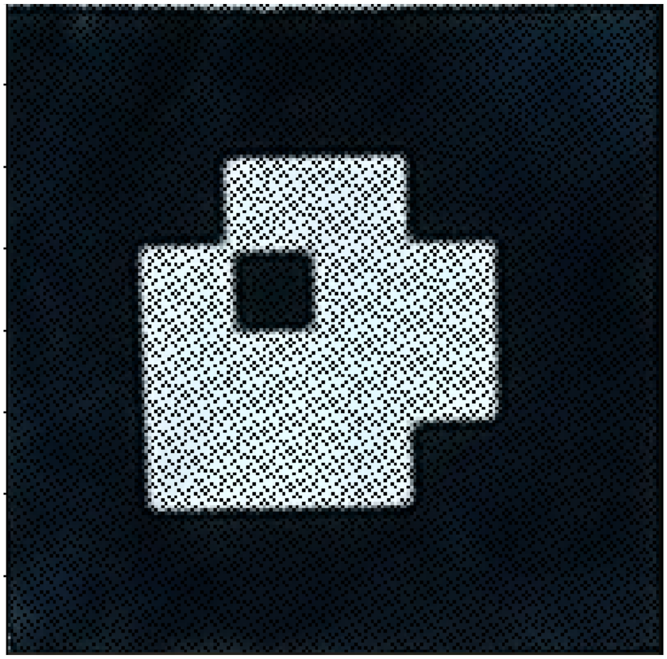




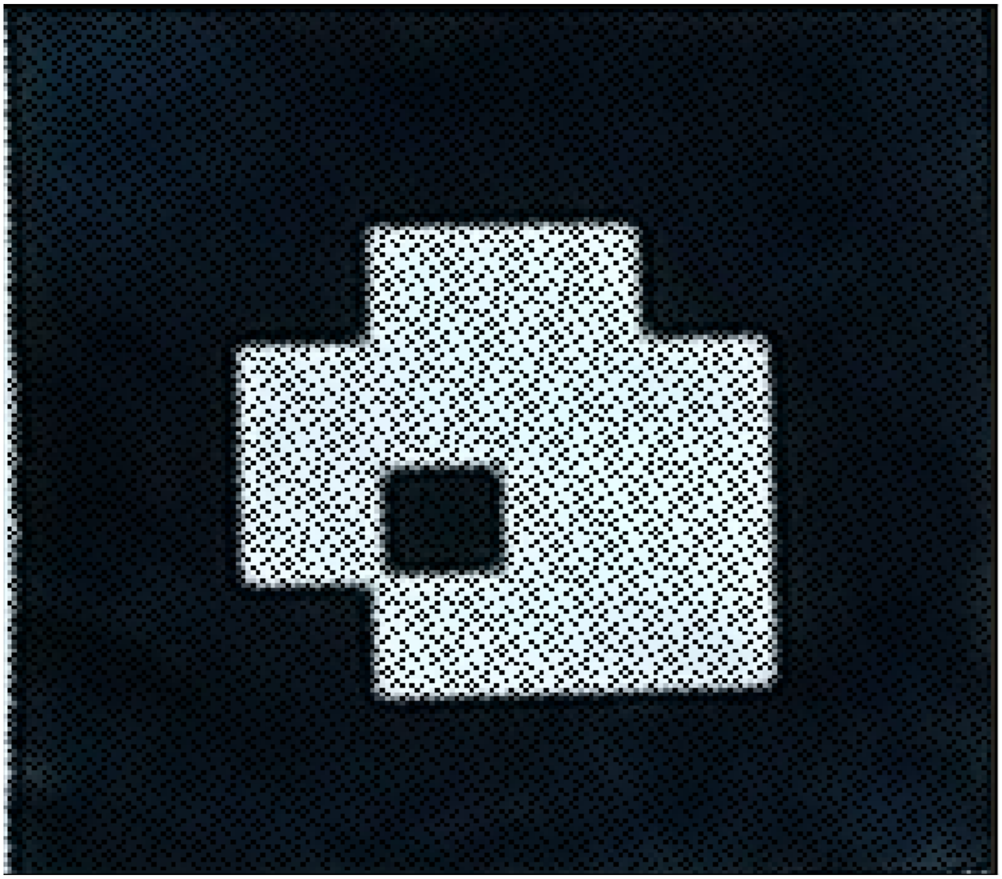
b) Decode custom AR tag:

Decoding of the AR tag is done in the following way:

1. The 4 corner points of the tag were found out in the last step
2. Using the custom made Homography and warpPerspective function the AR code is cropped out of the Image in a 200x200grid.



3. Then I divided the grid into 8x8 by taking 25x25 pixel grids
4. Check which corner grid has more than half pixel's white pixel and rotate the image to align it in such a way so that the AR code is upright i.e. the bottom right grid is white



5. Out of all the cells of the grid the 4 centered cells determine the tag id in clockwise direction such that the leftmost cell having value 1 the next cell to right 2 and cell below 4 and the left cell to it has 8 value.

Problem 2 – Tracking

a) Superimposing image onto Tag:

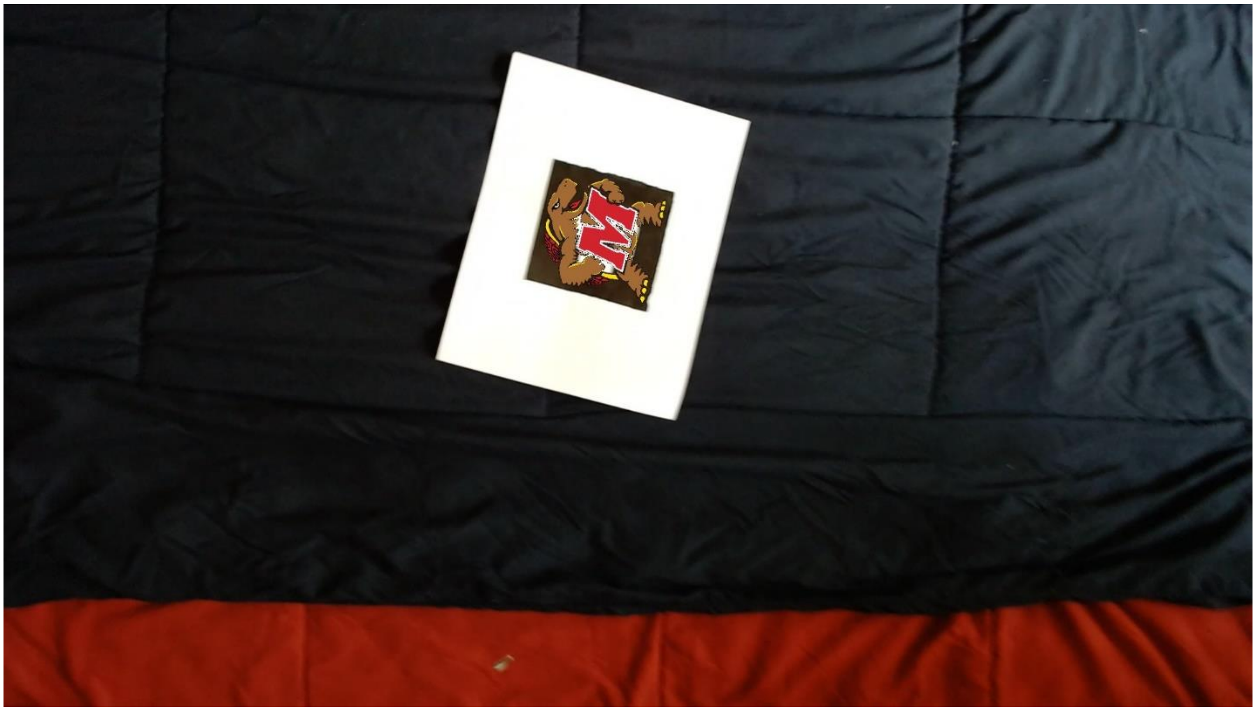
Once we fetched the corner points of the April tag and decoded the tag id and the correct orientation of the tag we follow the following steps to superimpose the testudo image on the April tag

1. Find out the height and the width of the testudo image and create a rectangle such that the end points of the rectangle are (0,0), (0,h), (w,h), (w,0) where w and h are the width and height of the testudo image.
2. Find out the Homography matrix using the corner points of the April tag and the rectangle created using the shape of the testudo image. The homography is created using the following equations and finding the SVD(Singular Value Decomposition).The resultant homography matrix is a 3x3 matrix

$$A = \begin{bmatrix} -x1 & -y1 & -1 & 0 & 0 & 0 & x1*xp1 & y1*xp1 & xp1 \\ 0 & 0 & 0 & -x1 & -y1 & -1 & x1*yp1 & y1*yp1 & yp1 \\ -x2 & -y2 & -1 & 0 & 0 & 0 & x2*xp2 & y2*xp2 & xp2 \\ 0 & 0 & 0 & -x2 & -y2 & -1 & x2*yp2 & y2*yp2 & yp2 \\ -x3 & -y3 & -1 & 0 & 0 & 0 & x3*xp3 & y3*xp3 & xp3 \\ 0 & 0 & 0 & -x3 & -y3 & -1 & x3*yp3 & y3*yp3 & yp3 \\ -x4 & -y4 & -1 & 0 & 0 & 0 & x4*xp4 & y4*xp4 & xp4 \\ 0 & 0 & 0 & -x4 & -y4 & -1 & x4*yp4 & y4*yp4 & yp4 \end{bmatrix}, x = \begin{bmatrix} H_{11} \\ H_{12} \\ H_{13} \\ H_{21} \\ H_{22} \\ H_{23} \\ H_{31} \\ H_{32} \\ H_{33} \end{bmatrix}$$

3. Once the homography is calculated we warp the original testudo image using the Homography matrix calculated in the previous step. The warped image is created by the vector multiplication between the Homography matrix and the coordinate points of the testudo image

4. The warped testudo image is stitched with the April tag. To stitch the image, we calculated the bitwise not mask of the grayscale testudo image and applied the bitwise and operation on the mask and the other image. Finally, this image was added with the testudo image to get the overlay of the testudo image on the tag. By overlaying the testudo image, it got superimposed on the April tag.



b) Placing a virtual cube onto Tag:

In the previous step we superimposed the testudo image on the April tag such that it matches the correct orientation of the tag. To place a virtual cube on the tag the following steps were followed

1. To place a virtual cube, it requires 8 points, 4 points in the top plane and another 4 points in the bottom plane.
2. To find out the projection matrix P the following steps were followed. The homography matrix was calculated. K is the camera intrinsic matrix. The projection matrix transforms the point from planar marker to the camera plane.
3. 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 the cube on the frame. And then draw the cube

$$\begin{aligned}
\boldsymbol{x}^{(c)} &= \boldsymbol{P}\boldsymbol{x}^{(w)} \\
\boldsymbol{x}^{(c)} &= \boldsymbol{K}[\boldsymbol{r_1}, \boldsymbol{r_2}, \boldsymbol{r_3}, \boldsymbol{t}][x^{(w)}, y^{(w)}, 0, 1]^T \\
\boldsymbol{x}^{(c)} &= \boldsymbol{K}[\boldsymbol{r_1}, \boldsymbol{r_2}, \boldsymbol{t}][x^{(w)}, y^{(w)}, 1]^T \\
\boldsymbol{x}^{(c)} &= \boldsymbol{H}[x^{(w)}, y^{(w)}, 1]^T,
\end{aligned}$$

$$[\boldsymbol{R} \mid \boldsymbol{t}] = [\boldsymbol{r_1}, \boldsymbol{r_2}, \boldsymbol{r_3}, \boldsymbol{t}]$$

$$\begin{aligned}
\lambda\boldsymbol{H} &= \boldsymbol{K}\widetilde{\boldsymbol{B}} \\
\widetilde{\boldsymbol{B}} &= \lambda\boldsymbol{K}^{-1}\boldsymbol{H} \\
\boldsymbol{B} &= \lambda\widetilde{\boldsymbol{B}}(-1)^{|\widetilde{\boldsymbol{B}}|<0},
\end{aligned}$$

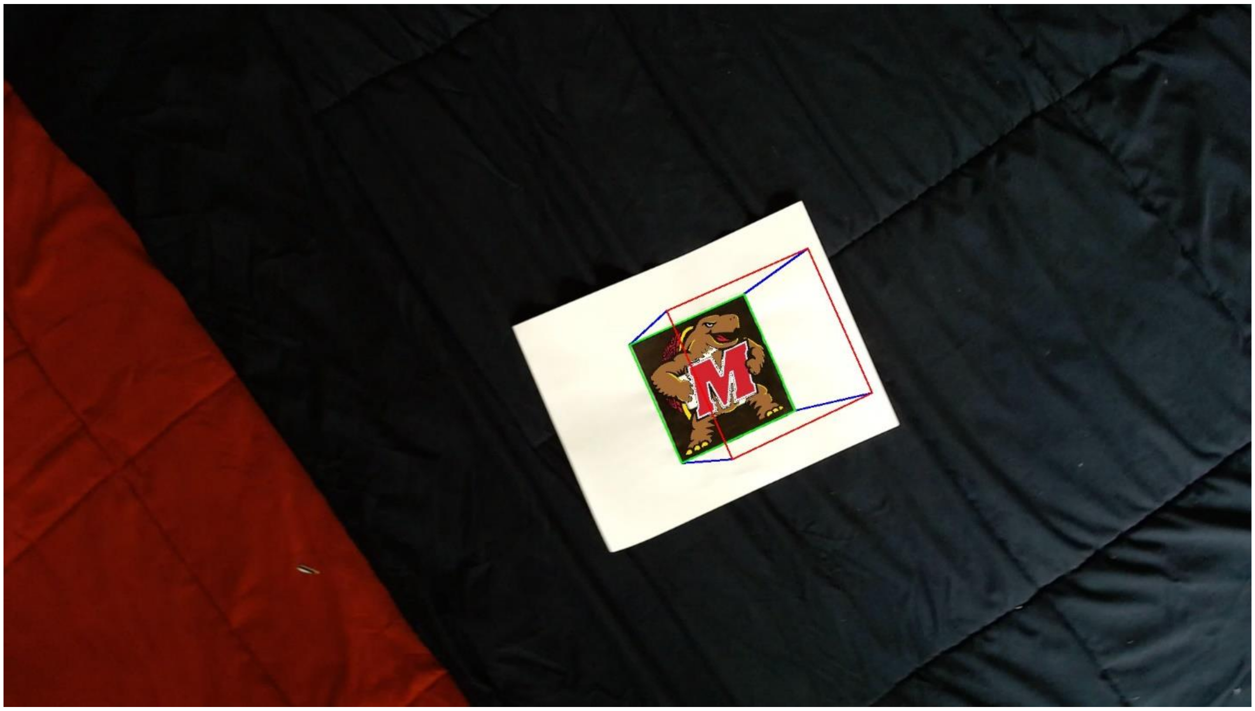
$$\lambda = \left(\frac{||\boldsymbol{K}^{-1}\boldsymbol{h}_1|| + ||\boldsymbol{K}^{-1}\boldsymbol{h}_2||}{2}\right)^{-1}$$

$$\boldsymbol{B} = [\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3].$$

$$\boldsymbol{r}_1 = \lambda\boldsymbol{b}_1 , \boldsymbol{r}_2 = \lambda\boldsymbol{b}_2 , \boldsymbol{r}_3 = \boldsymbol{r}_1 \times \boldsymbol{r}_2 , \boldsymbol{t} = \lambda\boldsymbol{b}_3.$$

$$\boldsymbol{P} = \boldsymbol{K}[\boldsymbol{R} \mid \boldsymbol{t}].$$

4. The 8 points were multiplied with the projection matrix to project those points in 3d space. The cube was constructed by joining the new projected 8 points.



Video Link :
https://drive.google.com/file/d/1OVsdZsXXpN9GXJRHbzzS515YzWen4l_T/view?usp=sharing