Name : Vignesh Ravichandran Radhakrishnan
UID: 119144060
Directory ID: rr94

ENPM673 – Perception for Autonomous Robots
Project 2

Problem 1:

Pipeline:

1. The video is read and processed frame by frame.
2. The image/frame is first blurred using gaussian blur function
3. The edges are found using the Canny edge detector function
4. The x and y coordinates of the edges are obtained where the pixel value of the detected edges (image) from the canny is not 0.
5. Hough transform is performed on the edges to obtain the corners
    - For each value of x and y (where pixel values of edges > 0), the distance is calculated as:
      d = x * cos (theta) + y * sin (theta)
    - A dictionary is created to check the occurrence of (d,theta) where (d,theta) is the key and the occurrence of each pair of (d,theta) is stored as the value for it.
    - The Dictionary H{(d,theta) : value} is sorted with the key in the ascending order and the points (denoted by (d,theta)) with the most occurrence are obtained. (eg:. 8 points with the maximum occurrence)
    - Each point is then checked for their angles (available in their key) with the other points and the (x,y) coordinate is obtained using a defined function calculate_xy where the d, theta and indexes of theta are passed as parameters.
      I.e., x and y are obtained from 2 equations: (using matrix operations)
      Say : d1 = x * cos (theta1) + y * sin (theta1)
      d2 = x * cos (theta2) + y * sin (theta2)
    - These (x,y) points obtained from the above equation are the corners of the paper

- These corners are then grouped together using a function which calculates the distance between the points to classify the different corners. (say each corner is at least 20-30 pixels away or the corner points which are in the range between 20-30 are of the same corner)

6. Homography is then calculated for each of the corner points with reference to the world coordinates of the paper. {(0,0) , (0,w), (l,0), (l,w)}
   Where the l and w are the length and width of the paper.

$$\underbrace{\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1 x_1 & -x'_1 y_1 & -x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y'_1 x_1 & -y'_1 y_1 & -y'_1 \\ & & & & \vdots & & & & \\ x_n & y_n & 1 & 0 & 0 & 0 & -x'_n x_n & -x'_n y_n & -x'_n \\ 0 & 0 & 0 & x_n & y_n & 1 & -y'_n x_n & -y'_n y_n & -y'_n \end{bmatrix}}_{\substack{\mathbf{A} \\ 2n \times 9}} \underbrace{\begin{bmatrix} h_{00} \\ h_{01} \\ h_{02} \\ h_{10} \\ h_{11} \\ h_{12} \\ h_{20} \\ h_{21} \\ h_{22} \end{bmatrix}}_{\substack{\mathbf{h} \\ 9}} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}}_{\substack{\mathbf{0} \\ 2n}}$$

Here n = number of corner points
h = homography matrix
$\hat{h}$ = eigenvector of $A^T A$ with smallest eigenvalue

7. Camera Pose Estimation:
   The Camera pose is then calculated using the intrinsic matrix of the camera and homography matrix.

$$\boldsymbol{H} = \lambda \boldsymbol{K} [\boldsymbol{r}_1 \boldsymbol{r}_2 \boldsymbol{t}]$$

$$\boldsymbol{K}^{-1} \boldsymbol{H} = \lambda [\boldsymbol{r}_1 \boldsymbol{r}_2 \boldsymbol{t}]$$

Where k -> Intrinsic matrix of the camera (given in the problem)
H -> homography matrix
Lambda -> value to be found (r1 and r2 are unit vectors)

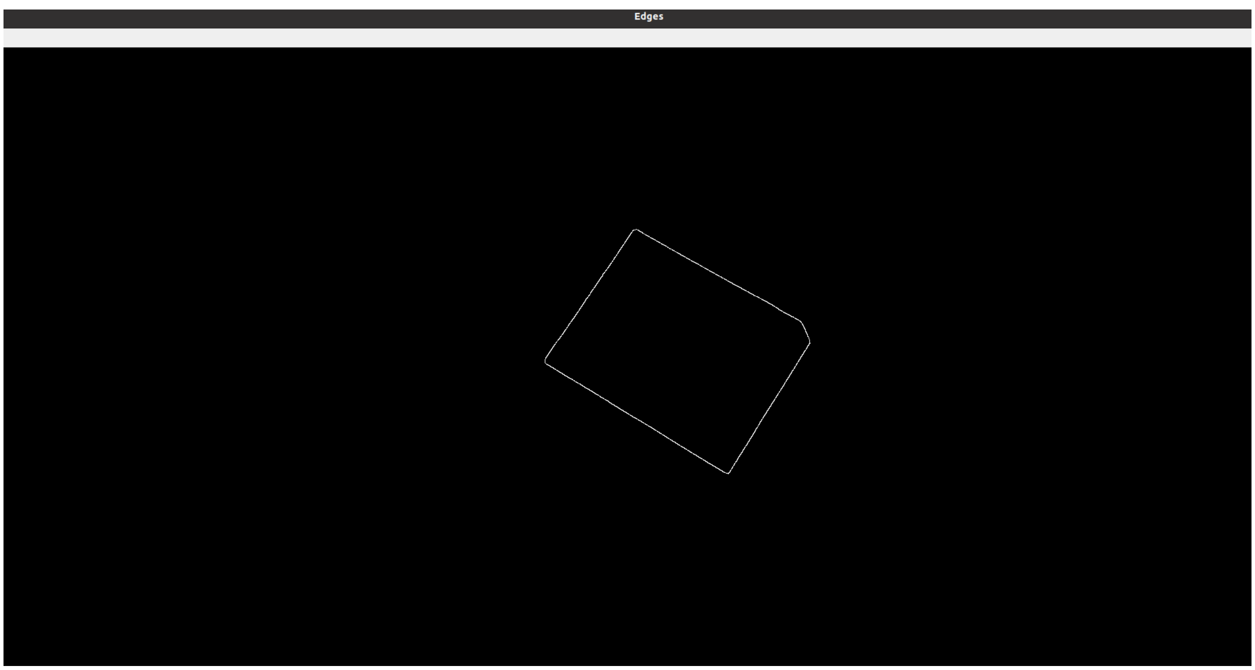
Camera pose, P is defined as:

$$\boldsymbol{P} = \boldsymbol{K}\left[\boldsymbol{r}_1 \quad \boldsymbol{r}_2 \quad (\boldsymbol{r}_1 \times \boldsymbol{r}_2) \quad \boldsymbol{t}\right]$$

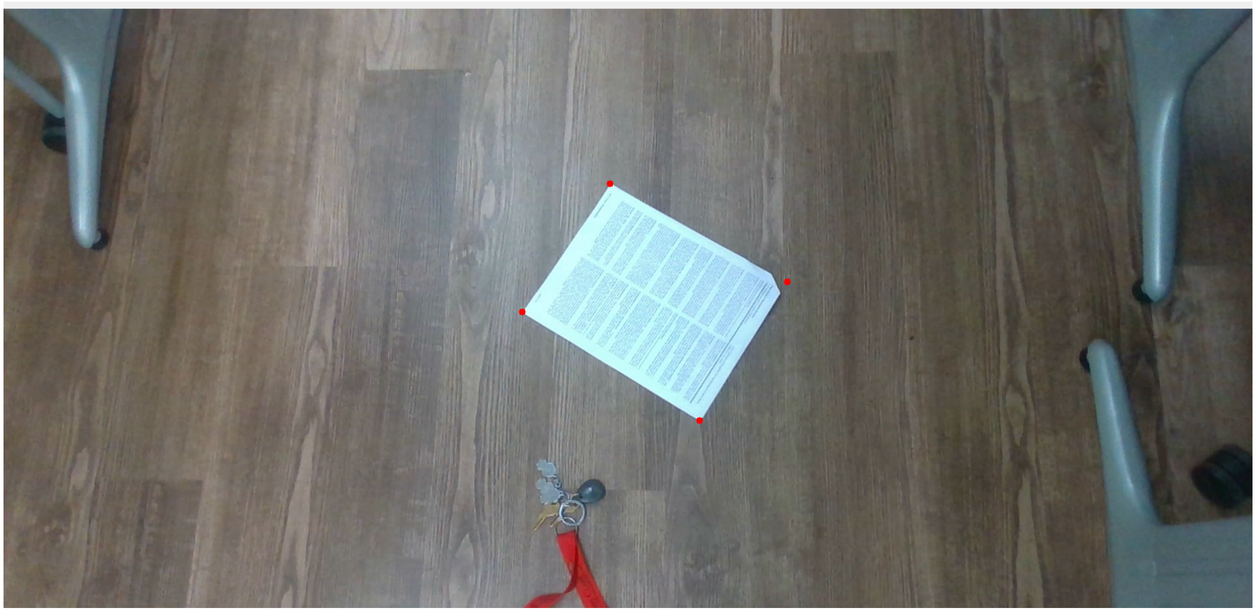
8. Finally the rotation matrix (r1,r2,r3) and the translation matrix (t) is plotted in a graph for each frame.

Results:

Edge Detection:



Corner points are drawn as a circle in the image at every corner point detected:



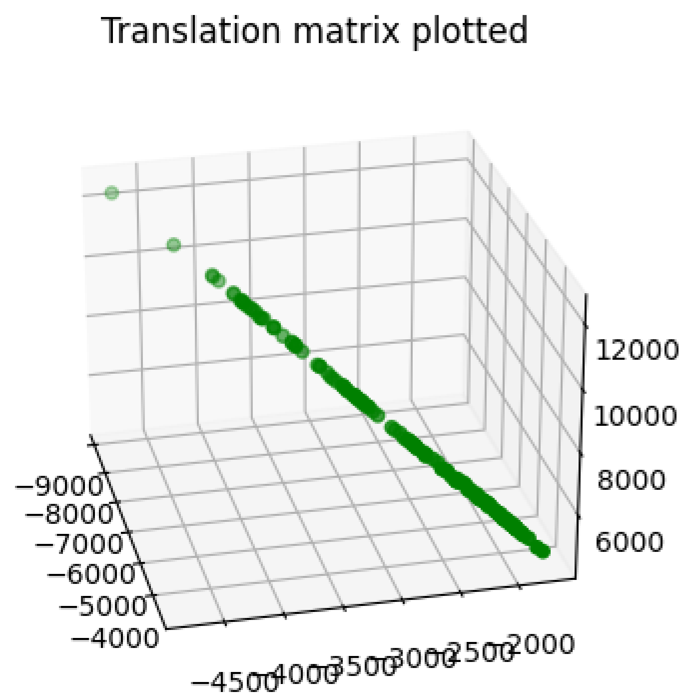
<u>Problem faced:</u>

1. Presence of noise when the edges were detected without performing the gaussian blur.
2. When only the successive d values (after sorting) are compared for grouping, then corner points were missed in some frames.

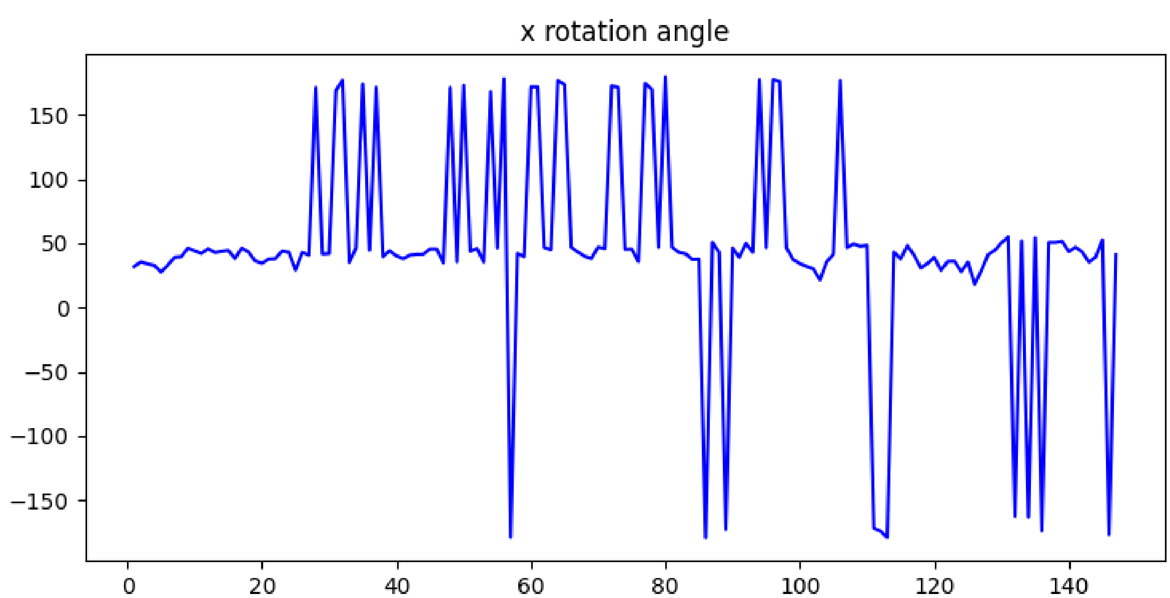
<u>Solutions to overcome the problem:</u>

1. Gaussian blur was performed on the image
2. Each d value is compared with all other values from the sorted dictionary (H(d,theta:val)) and then when the theta differs by 90 (say 85-95) degrees, then the points are considered orthogonal.

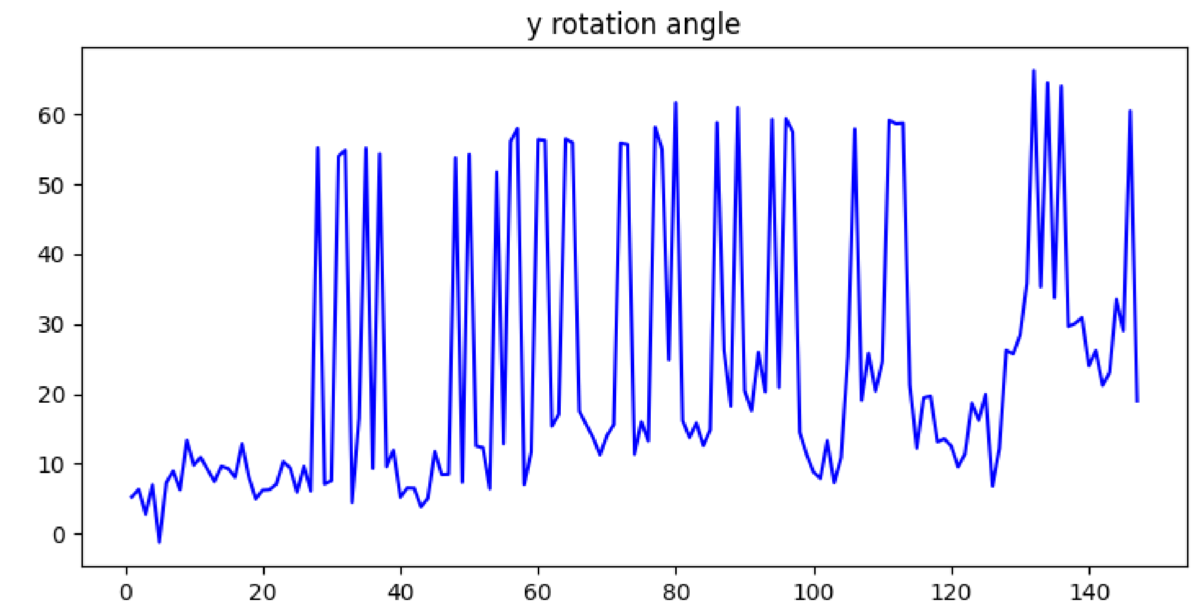
GRAPHS:



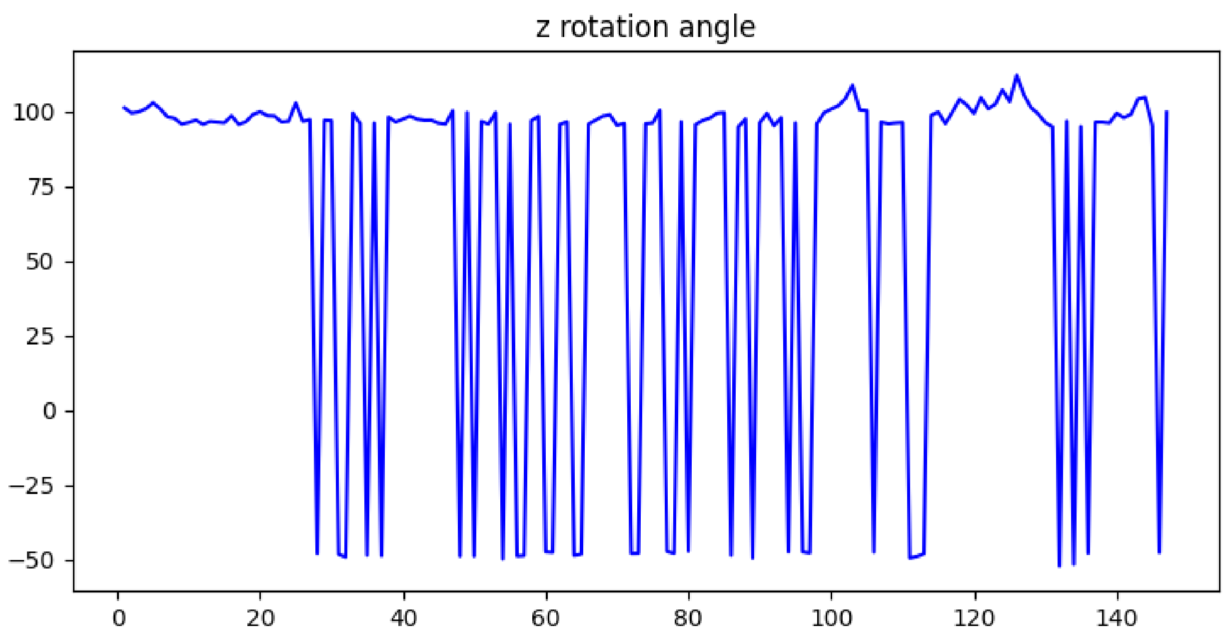
Translation matrix plotted
12000
10000
8000
6000
−9000
−8000
−7000
−6000
−5000
−4000
−4500
−4000
−3500
−3000
−2500
−2000



x rotation angle
150
100
50
0
−50
−100
−150
0
20
40
60
80
100
120
140



y rotation angle
60
50
40
30
20
10
0
0
20
40
60
80
100
120
140



z rotation angle
100
75
50
25
0
−25
−50
0
20
40
60
80
100
120
140

Problem 2:

Pipeline:

1. All the 4 images are read and resized accordingly (i.e., dimension of 640*480)
2. The keypoints and descriptors for each image are obtained using the sift inbuilt function detectAndCompute()
3. Brute force matching technique is used to find the matches between any 2 images by passing the descriptors of both the images.
   (Descriptors contain the information of the keypoints of each image in a 2d matrix of the order n * 128 where n is the number of keypoints)
4. In the matches obtained, the distance between any 2 matches are compared and the best match is appended into a list where the distance is minimum.
5. The coordinates (x,y) for both the images are then obtained from the best matches obtained and are appended to an array of points.
6. Homography between the 2 images is then performed on the obtained points (Same as problem 1)

$$\underbrace{\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1 x_1 & -x'_1 y_1 & -x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y'_1 x_1 & -y'_1 y_1 & -y'_1 \\ & & & & \vdots & & & & \\ x_n & y_n & 1 & 0 & 0 & 0 & -x'_n x_n & -x'_n y_n & -x'_n \\ 0 & 0 & 0 & x_n & y_n & 1 & -y'_n x_n & -y'_n y_n & -y'_n \end{bmatrix}}_{\substack{\mathbf{A} \\ 2n \times 9}} \underbrace{\begin{bmatrix} h_{00} \\ h_{01} \\ h_{02} \\ h_{10} \\ h_{11} \\ h_{12} \\ h_{20} \\ h_{21} \\ h_{22} \end{bmatrix}}_{\substack{\mathbf{h} \\ 9}} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}}_{\substack{\mathbf{0} \\ 2n}}$$

Here n = number of points

h = homography matrix

$\hat{h}$ = eigenvector of A^TA with smallest eigenvalue

7. The images are then combined using the computed homographies using the warp function.
8. Steps 1-7 are performed for all the 4 images where images are stitched in order, one next to each other.
9. The images are cropped accordingly since combining 2 images may result in a different dimension.

Problems faced:

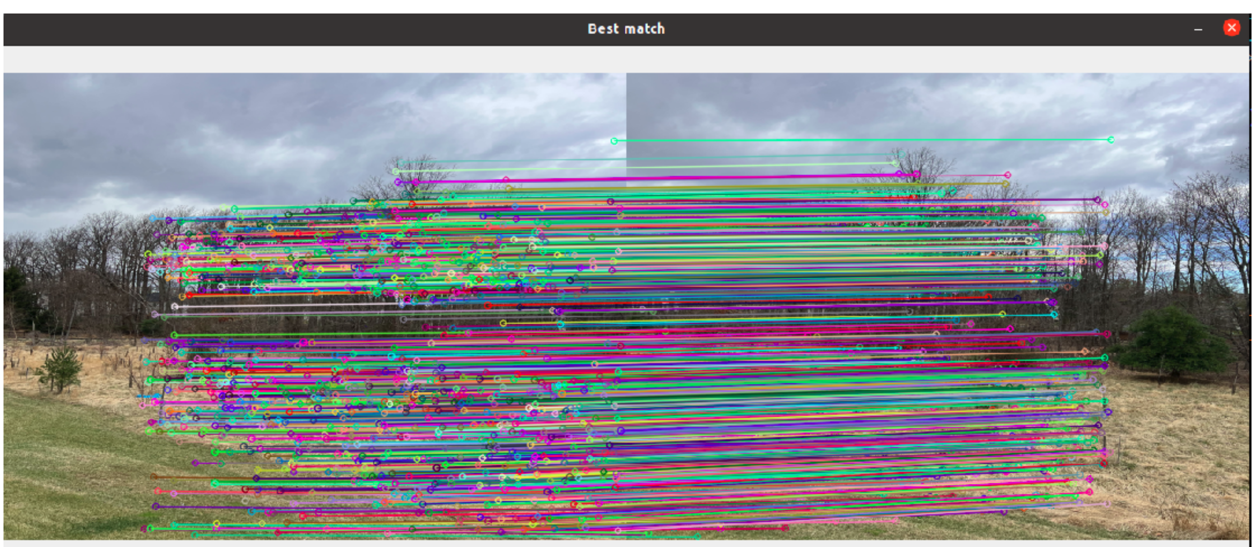
1. Size of warp image could not be identified since the region of overlap is unknown

Solutions for Problem:

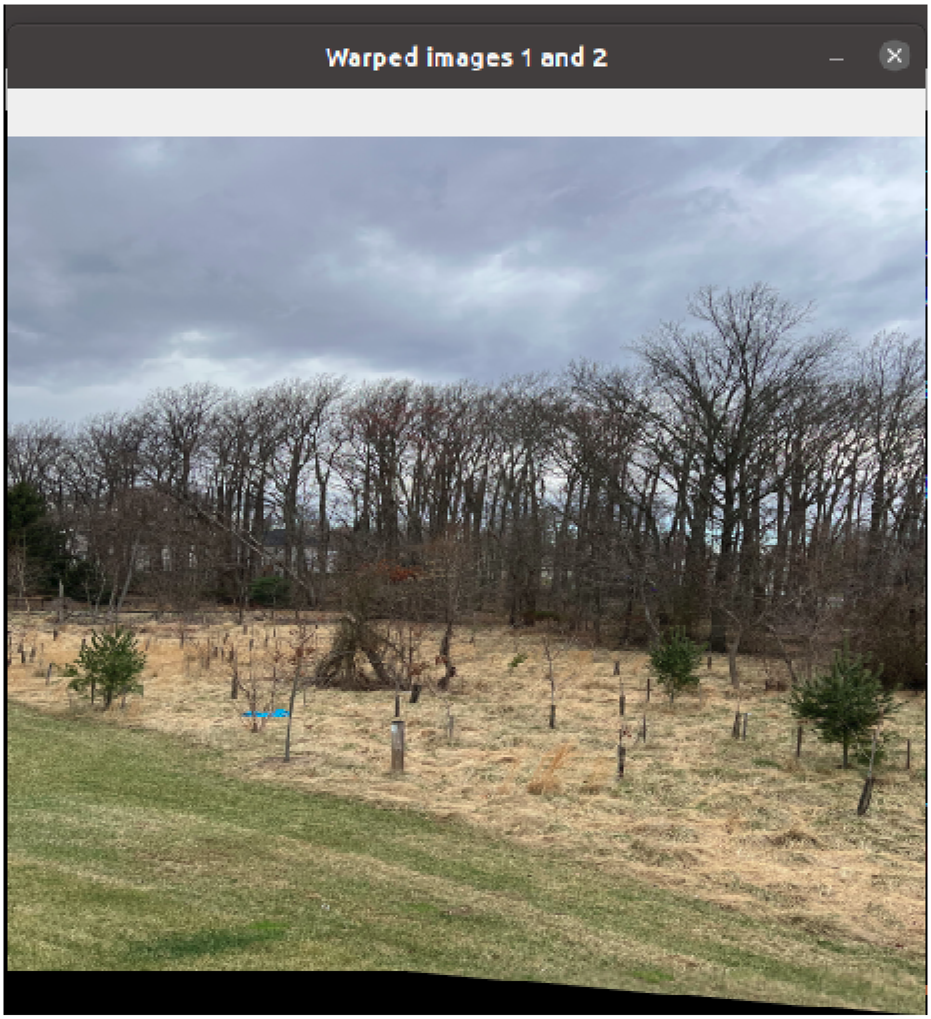
1. A higher dimension of warp image provided based on trial and error and then crop function is used to remove the unwanted region.

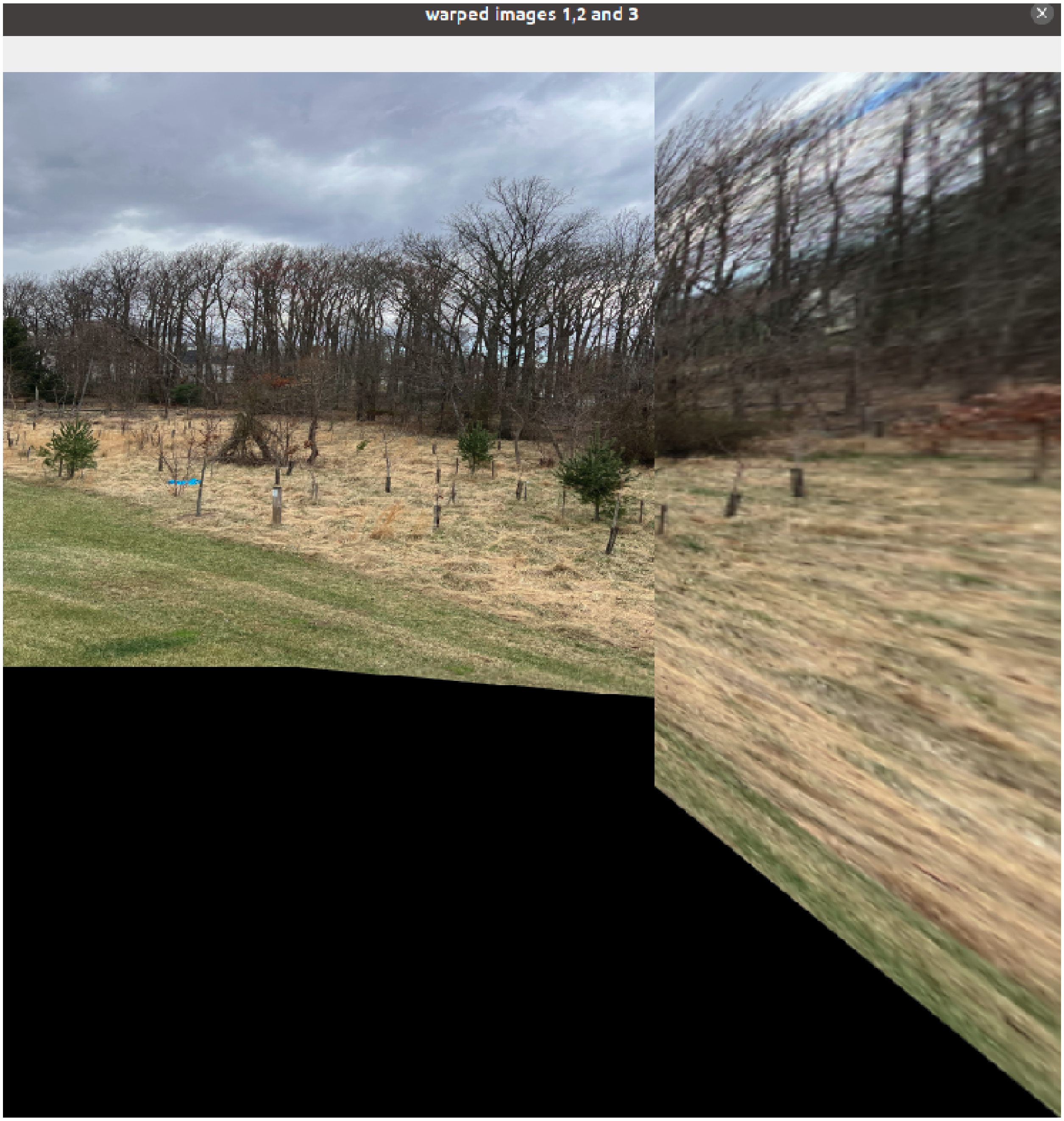
Results:

Matches from Image1 and Image2: (Sample of matching Keypoints using descriptors)



Combined Image:




warped images 1,2 and 3


warped all images