**EN2550 – Fundamentals of Image Processing and Machine vision**

**Assignment 02**

**Name**: B.W.J.C. Kumara

**Index Number**: 190328V

**Question 01**

1. Estimate the circle using the RANSAC algorithm (must be coded on your own).

RANSAC Parameters:

* s : Minimum number of points needed to fit on to a circle is 3, Therefore s is taken as 3
* t : to capture all the inliers t should be 1.96 , since sigma =1
* p = 0.99
* N = Number of iterations of the hypothesize and verify loop, calculated using the Equation so that p = 0.99

1. #RANSAC Algorithm
2. def RANSAC(X):
3. s = 3  # Minimum number of points needed to fit to a circle is 3
4. t = 1.96 # to capture 95% of all inliers
5. d = 50
6. p = 0.99
7. N = int(np.ceil(np.log(1-p)/np.log(1-(1-0.5)\*\*s)))
8. max\_inliers = 0;
9. for i in range(N):
10. c1,c2,c3 = X[np.random.randint(0,100)],X[np.random.randint(0,100)],X[np.random.randint(0,100)]
11. radius, center = CircleCenter(c1,c2,c3)
12. inliers, outliers, inlier\_count = CheckPoints(X,radius,center,t)
13. if max\_inliers<inlier\_count:
14. max\_inliers = inlier\_count
15. best\_sample = np.array([list(c1),list(c2),list(c3)])
16. best\_inliers = np.array(inliers)
17. best\_outliers = np.array(outliers)
18. best\_radius = radius
19. best\_center = center
21. fig, ax = plt.subplots(figsize = (12,12),facecolor = "white")
23. # Circle to fit the best 3 samples
24. best\_circle = plt.Circle((best\_center[0],best\_center[1]),best\_radius,color = 'blue',fill = False,label = "Best Circle")
26. # Calculate the best circle to fit all the inliers
27. xr,yr ,rr,k = cf.least\_squares\_circle(best\_inliers)
28. RANSAC\_circle = plt.Circle((xr,yr),rr,color = 'm',fill = False,label = "RANSAC")
30. # Plotting inliers, Outliers and the Best 3 samples
31. ax.scatter(best\_outliers[:,1],best\_outliers[:,0], c= 'blue',label = "Outliers")
32. ax.scatter(best\_inliers[:,1],best\_inliers[:,0],c = 'green',label = "Inliers")
33. ax.scatter(best\_sample[:,1],best\_sample[:,0],c = 'red',label = "Best Sample")
35. #Plotting two circles
36. ax.add\_patch(best\_circle)
37. ax.add\_patch(RANSAC\_circle)
38. ax.legend(loc = "upper right")
39. ax.set\_title("RANSAC Circle fitting")
40. plt.show()

(b) Show in the same plot, the point set, the circle estimated from the sample leading to the best estimate, this sample of three points, inliers, and the best-fit circle. See Figure 1 for an example.

Chart

Description automatically generated with medium confidence

**Question 02**

**Graphical user interface, website

Description automatically generated**

02)Movie image projected onto a poster

03)Movie image projected to a billboard

01)British flag projected onto a building

**Question 03**

1. Compute and match SIFT features between the two images

A map of a city

Description automatically generated with low confidence

1. Compute the homography using your own code within RANSAC and compare with the homography given in the dataset.

Since sift features between image 1 and image 5 were not enough, homography between all the images had to be calculated using RANSAC.

1. Stitched Image

A picture containing text, graffiti, colorful

Description automatically generated