EN2550 - Assignment 2 on Fitting and Alignment

Name: C. J. Kurukulasuriya
Index Number: 190337X
GitHub: https://github.com/chira99/image-processing-opencv-python.git

Question 01

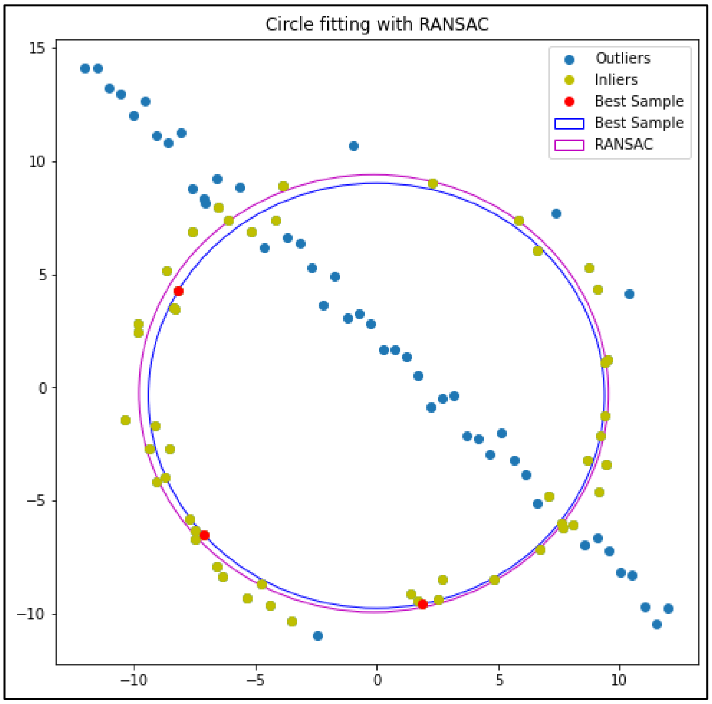
a) RANSAC algorithm for circle estimation is implemented as follows.

```
def RANSAC_circ(X):

    e = 0.5     # outlier ratio
    s = 3       # Number of points needed to create the estimated model
    p = 0.99    # probability that at least 1 sample is free from outliers
    t = 1.96 * 10/16   # treshold
    d = 50      # expected inlier count

    iters = int(np.ceil(np.log(1-p)/np.log(1-(1-e)**s)))

    best_inlier_count = 0
    best_samples = None
    best_fit_inliers = None

    for _ in range(iters):

        # Choose 3 distinct points from dataset
        [p1, p2, p3] = np.random.choice(len(X), size=3, replace=False)
        [p1, p2, p3] = X[p1, :], X[p2, :], X[p3, :]

        # Get circle through the 3 points
        f, g, r = getCircle(p1, p2, p3)

        if r == None:
            continue

        inlier_count, inliers = getInlierCount(f, g, r, X, t)

        if inlier_count > best_inlier_count:
            best_inlier_count = inlier_count
            best_fit_inliers = inliers
            best_samples = [p1, p2, p3]
            best_fit_circle = [f, g, r]

    if best_inlier_count < d:
        # Repeat RANSAC if no model found
        RANSAC_circ(X)

    ransac_circle = bestFitCircle(best_fit_inliers) # returns f,g,r

    return ransac_circle, best_fit_circle, best_samples, best_fit_inliers
```

Parameters of the algorithm:

- The minimum number of points needed to estimate the circle, **s = 3**
- A threshold of **t = 1.96*(r/16)** gives the required 95% probability of capturing all inliers since the dataset is corrupted by mean-zero variance-one gaussian noise. (r = 10)
- Consensus size, **d = 50**, since 50 points are inliers out of the given 100 dataset points.

b) The resulting circle fitting using RANSAC algorithm is as follows.

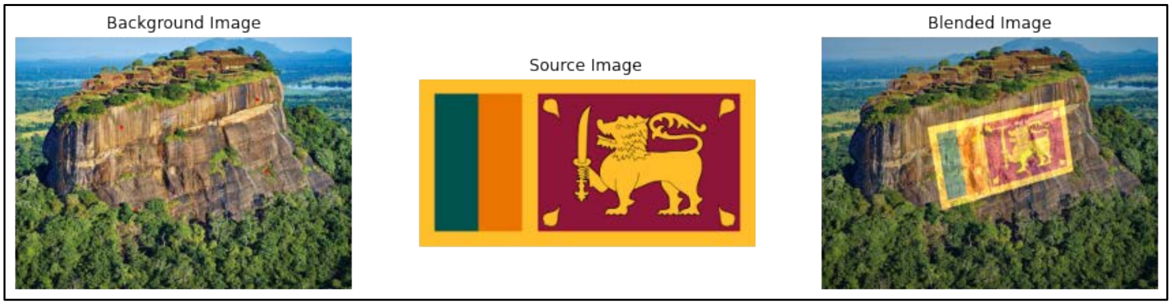


Question 02

1. Classical painting projected onto the wall of a display room.

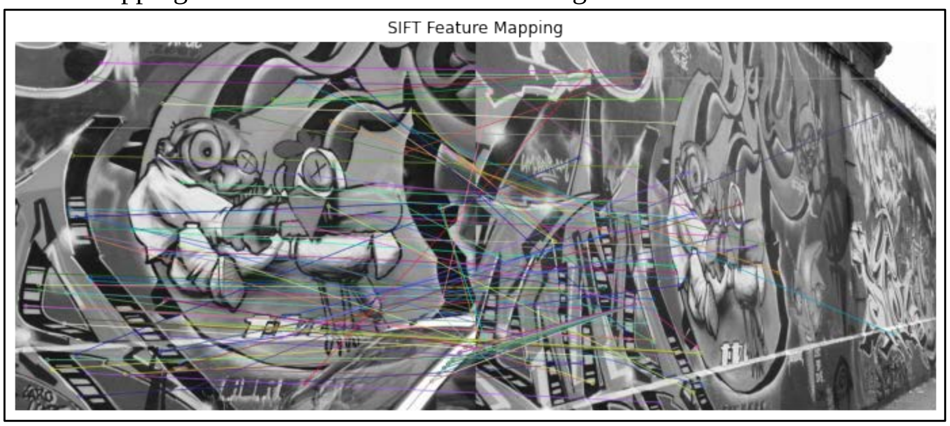
2. Movie poster on billboard display.

3. Sri Lankan flag projected/painted onto Sigiriya.



Question 03

a) The SIFT feature mapping carried out between the two images are as follows.

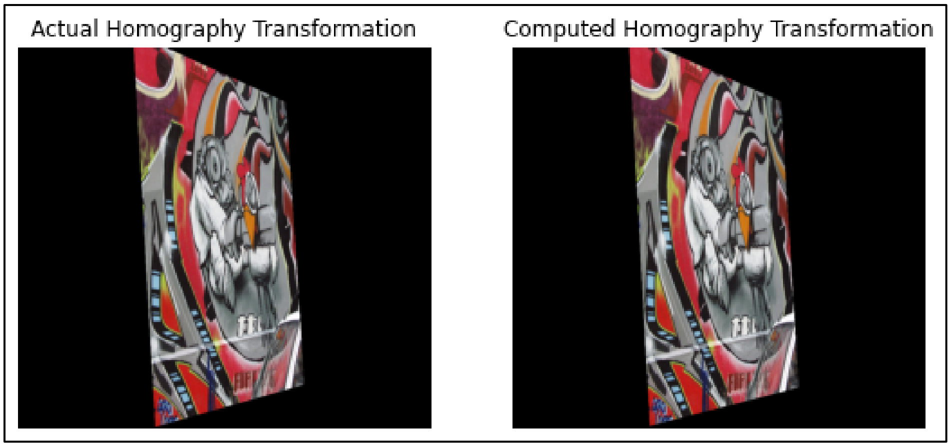


b) Due to the high perspective difference between image 1 and image 5 the number of “good” SIFT feature matches were insufficient to calculate a satisfactory homography transformation between them. Therefore, an intermediate homography was calculated using image 4 and by composition, a satisfactory homography was obtained for image 1 to image 5. The relevant code for calculating the homography is as follows.

```
def get_homography(X, Y):
    A = []
    zeros = np.array([0,0,0])

    # create matrix A
    for i in range(4):
        A.append(np.hstack((X[i, :], zeros, (-1*Y.T[0, i]*X[i,:]))))
        A.append(np.hstack((zeros, X[i, :], (-1*Y.T[1, i]*X[i,:]))))

    A = np.array(A).squeeze().astype(np.float64)

    # find the eigen vector H corresponding to the smallest eigen value
    eigen_values, eigen_vectors = np.linalg.eig(A.T @ A)
    col_idx = np.argmin(eigen_values)
    H = eigen_vectors[:, col_idx]

    # rearrange H to obtain the Homography transformation matrix
    H = H.reshape(3, -1)

    return H
```

```python
# Calculate homography in two parts
H1to4 = RANSAC("Images/graf/img1.ppm", "Images/graf/img4.ppm", 1, 20, 10000)
H4to5 = RANSAC("Images/graf/img4.ppm", "Images/graf/img5.ppm", 1, 20, 10000)
H1to5 = H4to5 @ H1to4
```

The transformations carried out using the computed homography and the actual homography have a high resemblance, as shown below



The two homography matrices are given below, and their Sum Squared Difference, SSD = 10.08.

- Actual Homography

$$\begin{pmatrix} 6.2544644e-01 & 5.7759174\mathrm{e}-02 & 2.2201217\mathrm{e}+02 \\ 2.2240536\mathrm{e}-01 & 1.1652147\mathrm{e}+00 & -2.5605611\mathrm{e}+01 \\ 4.9212545\mathrm{e}-04 & -3.6542424\mathrm{e}-05 & 1.0000000\mathrm{e}+00 \end{pmatrix}$$

- Computed Homography after accounting for λ difference

$$\begin{pmatrix} 6.13441589\mathrm{e}-01 & 5.42088817\mathrm{e}-02 & 2.23627892\mathrm{e}+02 \\ 2.19597334\mathrm{e}-01 & 1.15414254\mathrm{e}+00 & -2.28718545\mathrm{e}+01 \\ 4.73030560\mathrm{e}-04 & -4.23432171\mathrm{e}-05 & 1.00000000\mathrm{e}+00 \end{pmatrix}$$

c) The image 1 stitched onto image 5 using the homography calculated is as follows.