Problem 1. Image Alignment with RANSAC:

- *Compare the parameter settings in SIFT feature and RANSAC and discuss the result.*
  - RANSAC (apply Homography):
    - random selection of matching points, comput the homography(num_iterations = 1000)

```
A = np.zeros((coordinate*2, 9))

for i in range(coordinate):

    x1 = -points[i]
    y1 = corres_points[i, 0]

    x2 = points[i]
    y2 = corres_points[i, 1]

    A[i*2,3:6]  = x1
    A[i*2,6:]   =  x2 * y2

    A[i*2+1,:3] =  x2
    A[i*2+1,6:] = x1 * y1
```

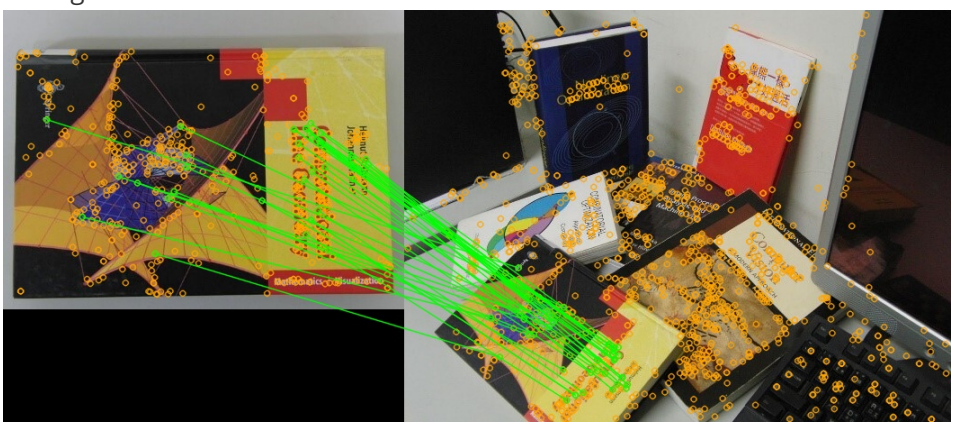
    - perform singular value decomposition(SVD) on the matrix H

```
U, S, V = np.linalg.svd(A, full_matrices=True)
f = V[-1, :]
Fundamental_matrix = f.reshape(3, 3)
```

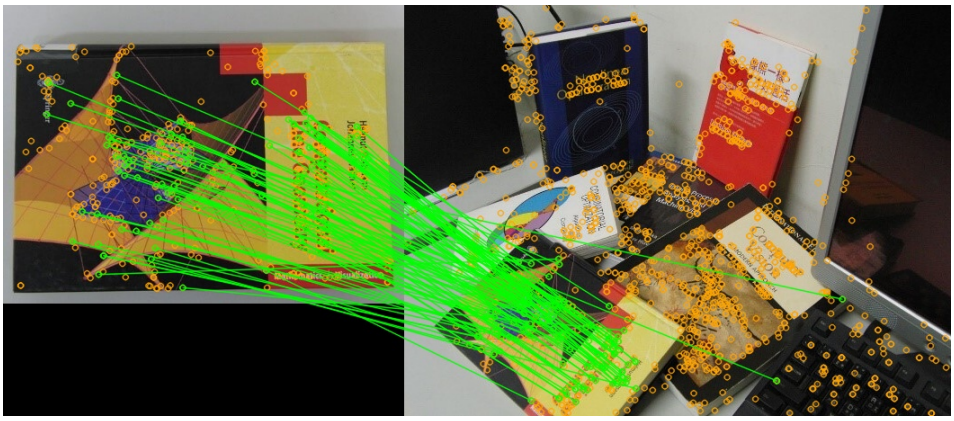
    - Find the inliers of homography transformation, retain the most inliers.

```
if len(inliers) < len(inlier_idx):
    inliers = inlier_idx
```

  - Discussion:
    - Both SIFT and RANSAC have set threshold parameters, but RANSAC needs to set one more **iterations** parameter.
      - The figure below shows a threshold value of 0.5



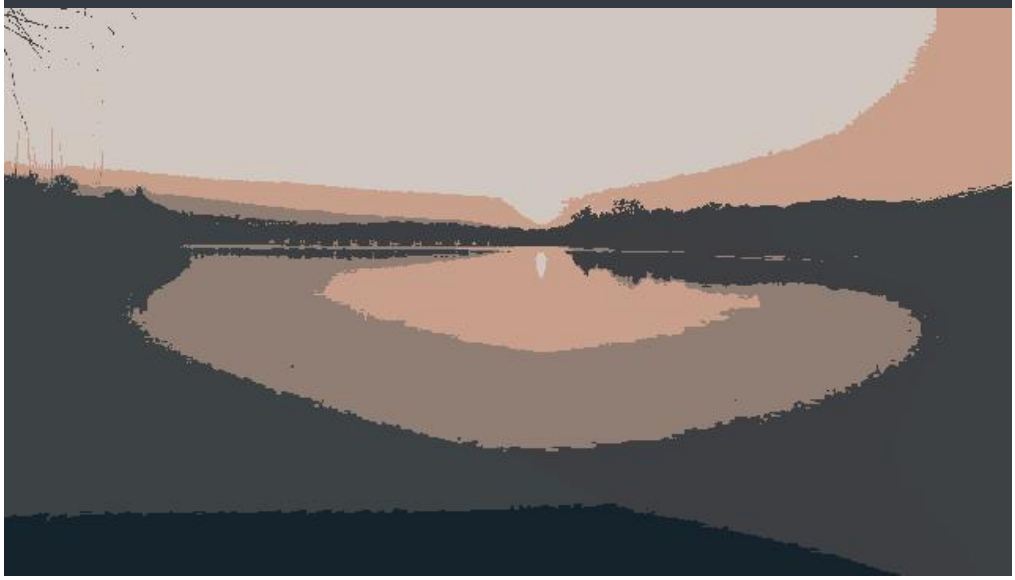
- The figure below shows a threshold value of 0.8



- The advantage of using RANSAC is that it can robustly estimate the model parameters, but it is necessary to manually calculate the number of iterations of the parameters.**If the upper limit of the number of iterations is set, the results obtained may not be the best results**.

Problem 2. Image segmentation

- *K-means(Please discuss the difference between the results for different K's)*
  - Different **k** indicates the number of classifications with k clusters. The following above picture shows the segmentation on K=5 and the following below picture shows the segmentation on K=7. **The larger the K, the more image details can be preserved.**



- *Please discuss the difference between (A:K-means) and (B: K-means++)*
  - **The difference between k means and k means ++ is the initial cluster center**.
  - k means is to randomly select K cluster centers

```
# randomly choose K center_clusters
idx_random = np.random.choice(M, K_clusters, replace=False)
center_clusters = pixels[idx_random]
```

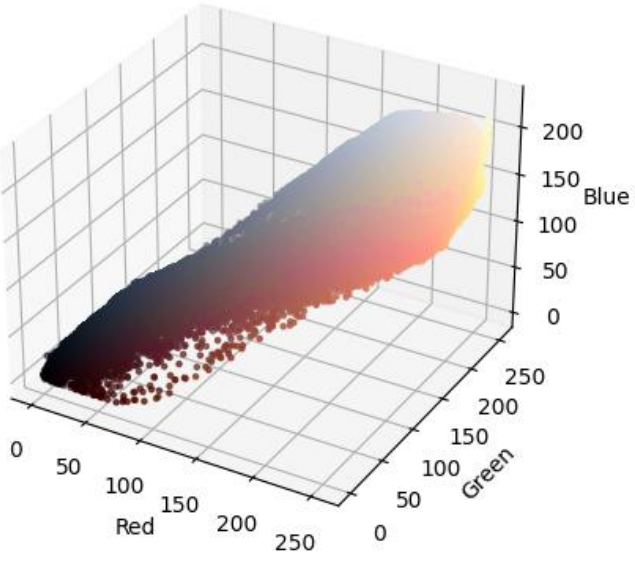
  - k means ++ first randomly selects one cluster center, and then based on the distance between the pixel point and the cluster centesr , the one with the largest distance is taken as the next cluster center until K cluster centers are selected.

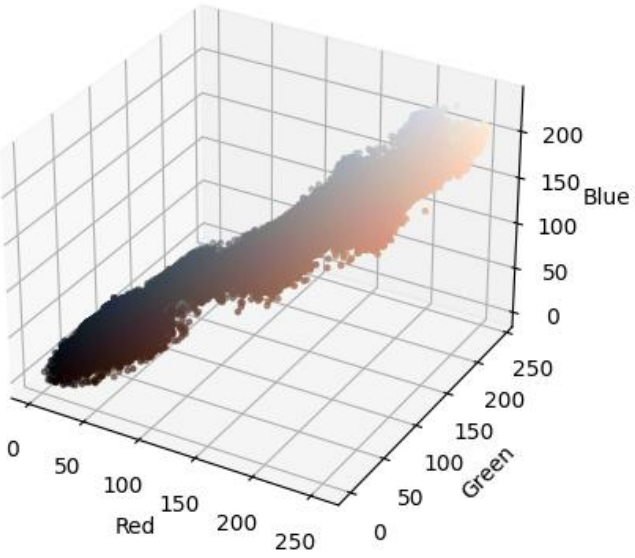
```
# randomly choose one center_clusters
idx_random = np.random.choice(M, 1, replace=False)
center_clusters = pixels[idx_random]

# choose the next center_clusters
for j in range(K_clusters-1):
    distanceList = []
    for i in range(pixels.shape[0]):
        distance = sum(np.linalg.norm(pixels[i] -
center_clusters,axis=1))
        distanceList.append(distance)

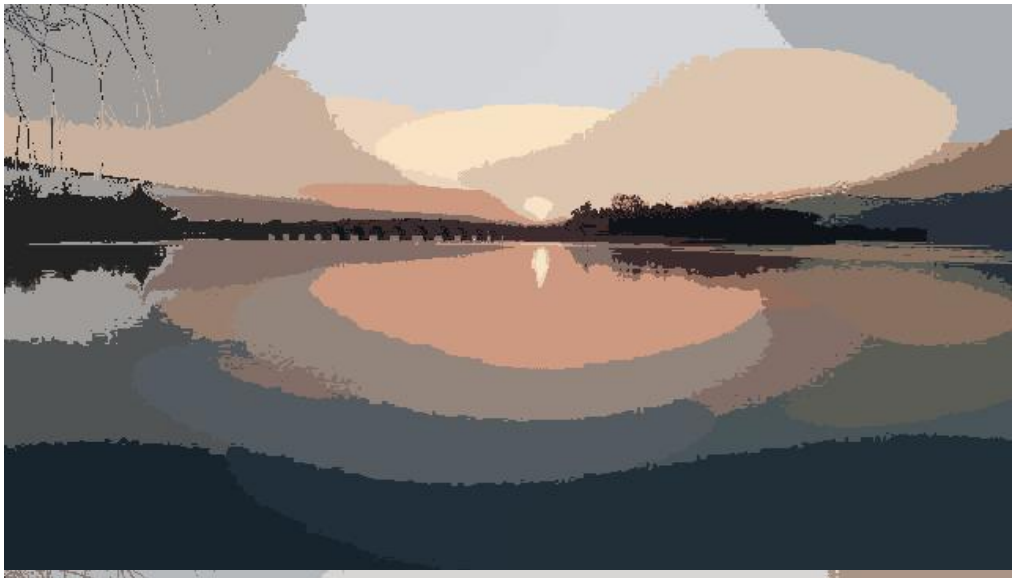
    max_distance = max(distanceList)
    idx_random = np.append(idx_random, distanceList.index(max_distance))
    center_clusters = pixels[idx_random]
```

- *Show the pixel distributions in the RGB feature space before and after applying mean-shift.*
  - The following above picture shows the RGB feature space before applying mean-shift and the following below picture shows the RGB feature space before and after applying mean-shift.



- 

- *Combine the color and spatial information into the kernel for mean shift segmentation and find the optimal parameters for the best segmentation result.*
  - 
- *Mean-shift(Discuss the segmentation results for different bandwidth parameters)*
  - The bandwidth of Means-shift is a value between 0 and 1, and is represented by the rgbspace of Uniform Kernel. The following above picture shows the segmentation on bandwidth=0.3 and the following below picture shows the segmentation on bandwidth=0.5. **The larger the bandwidth, the less likely it is to preserve image details.**

- 
- 

- *Compare the segmentation results by using K-means and mean-shift algorithms and their computational cost.*
    - **Mean-shift does not need to specify the number of clusters**, and can be used for clusters of any shape.However, **K-means relies on a given random cluster centers**, and random initialization will generate different aggregation effects, affects the entire segmentation effect.
    - Do k-means, k-means++ and mean-shift on 2-image respectively, and calculate the number of iterations needed to reach convergence. It can be found that the **convergence speed is mean-shift > k-means++ > k-means**.
    - ```
      PS D:\Desktop\NTHU_Computer_Vision\Homework3\HW3_111062594\2> python .\HW3-2.py
      K-means convergencetimes : 63
      K-means-plus convergencetimes : 28
      Mean-shift convergencetimes : 12
      ```
        - ps. the number of parameter clusters of kmean and k-means++ is set to 5, and the parameter bandwidth of meanshift is set to 0.5 as the experimental parameter.
      ```