ECS 174 Spring 2020 (05/11) Problem Set #2
Bingwei Wang 914683636
Nan Chen 915218152

Part 1.

(1) We believe that the grouping algorithm graph-cuts, would be the most appropriate grouping algorithm to recover the model parameter hypotheses from the continuous vote space. The reason is that it links every pair of pixels and assigns an affinity weight for each edge.

Additionally, graph-cuts connect shapes with edges that have high affinity while mean shift and k-means does not, which suits those lines and circles defined by a set of boundary points in Hough Transform.

But here comes a problem which we find mean-shifts might have better performance. Graph-cuts use a pixel-based energy function to compute the similarity of neighboring pixels, while in a continuous space there will be an infinite number of points, each with an infinite number of neighbors, so this would not work. For mean-shift, it is non-parametric and does not segment the vote space because it takes in the number of clusters we want as input. The algorithm naturally discretizes the vote space, and we do not need discretization of the vote space in the mean-shift algorithm.

We do not choose k-means because in continuous vote space, it is hard(impossible) to select parameter k.

(2) If we use K-means to cluster with two graphs, we will have two center points which divides all the feature inputs into two clusters. Each center point will get half of the feature points and the center point will be positioned by calculating the least squared Euclidean distance. After that it will updates the new center of each cluster. The process will keep repeated until half of the data points are in one group and the other half is in the other group.

(3) TO BE CONTINUE. □

Part 2.

(a) The code is written in file **get_correspondences.m**. The function has two inputs of two images and the number of points the user wants to pick. The outputs are two 2 x N point matrices of selected points.

(b) The code is written in file **computeH.m**.

(c) The code is written in file **warpImage.m**.

(d) Images:

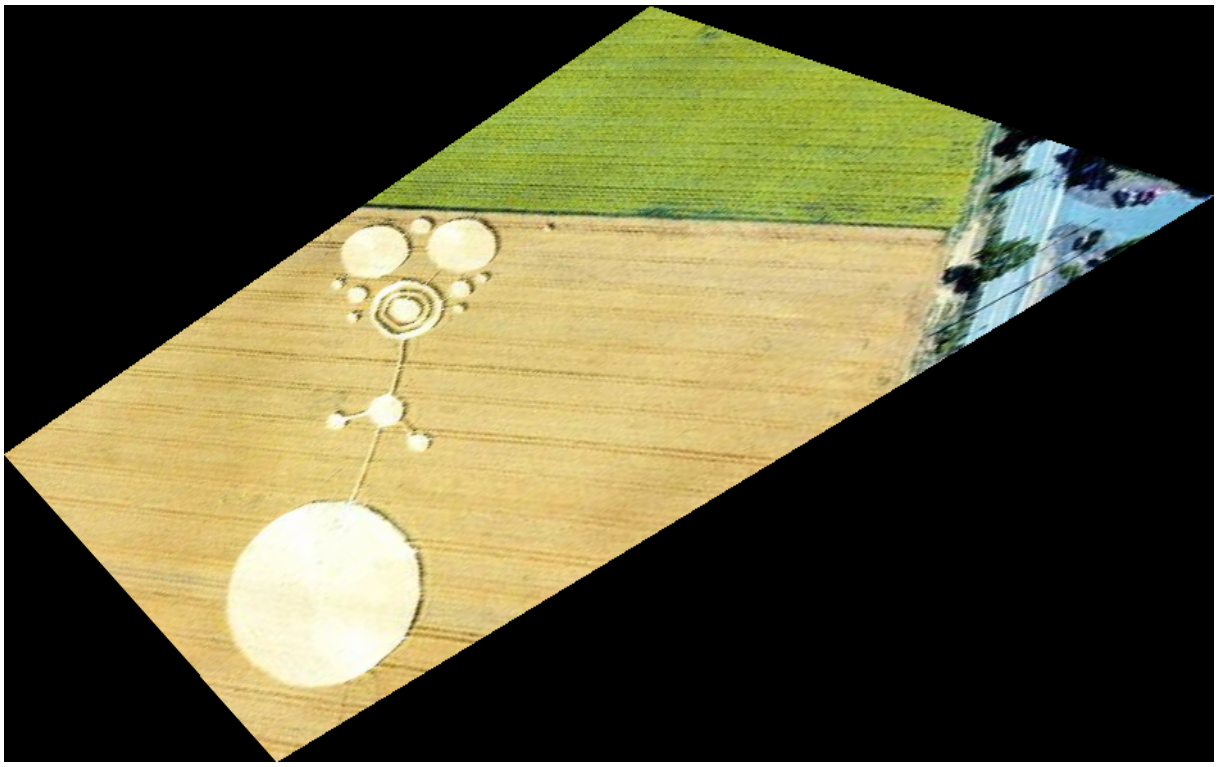

Figure 1: crop_warp

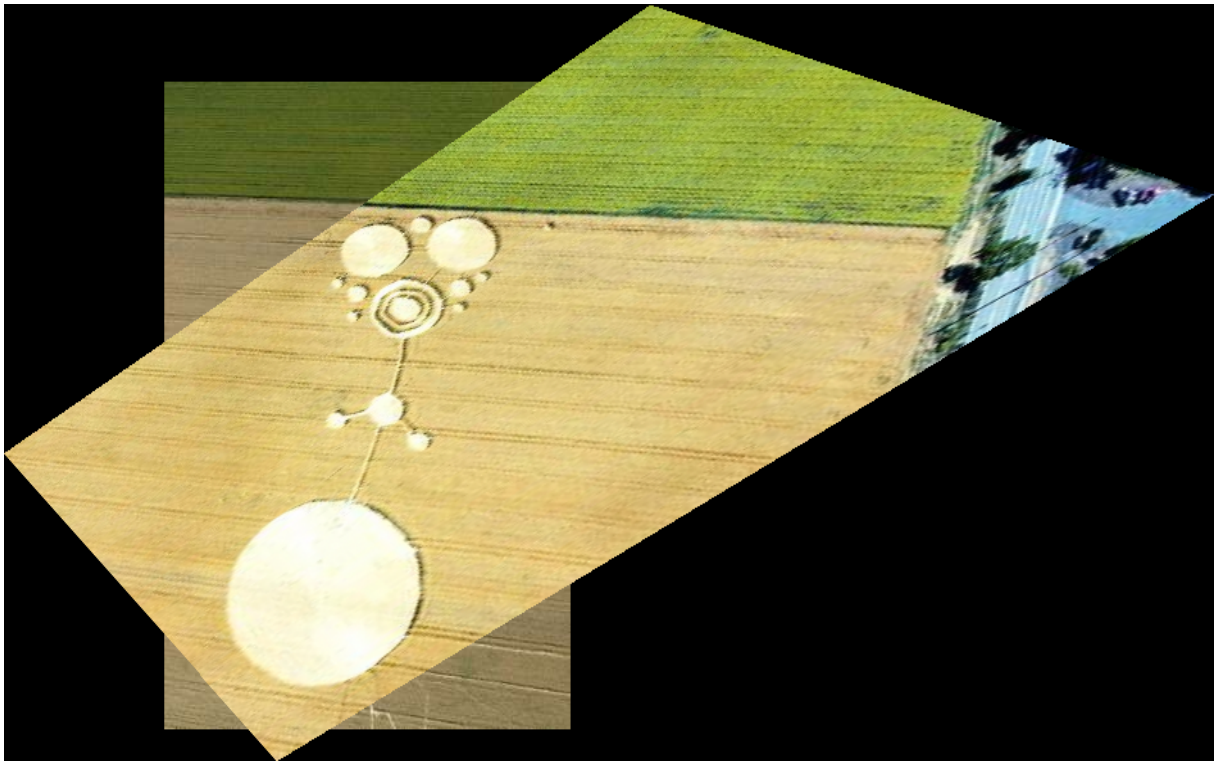

Figure 2: crop_merge

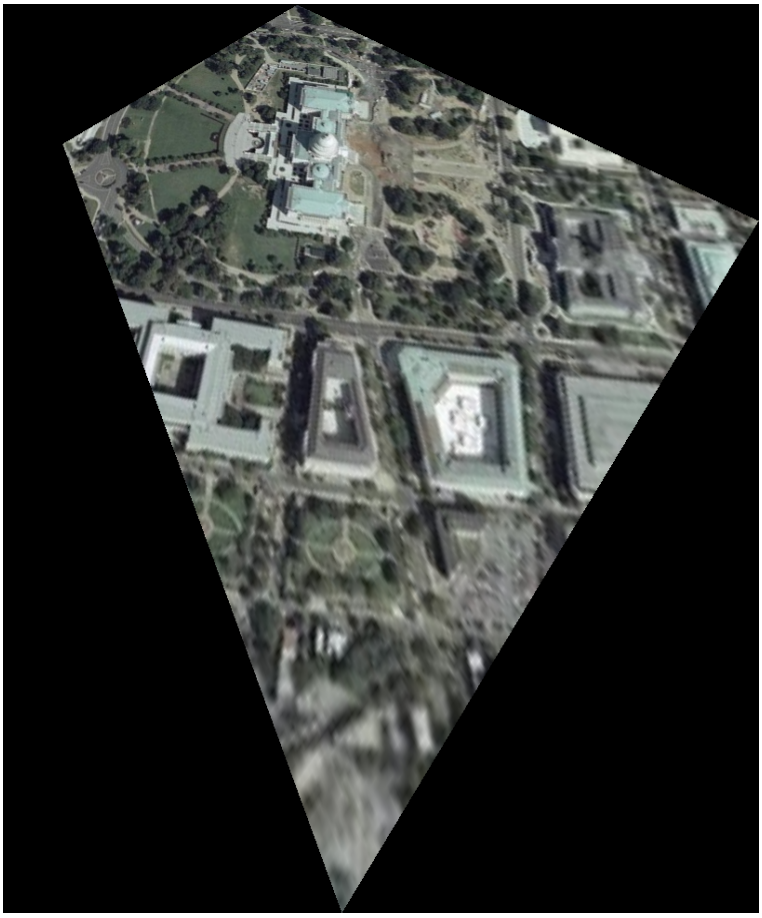

Figure 3: wdc_warp

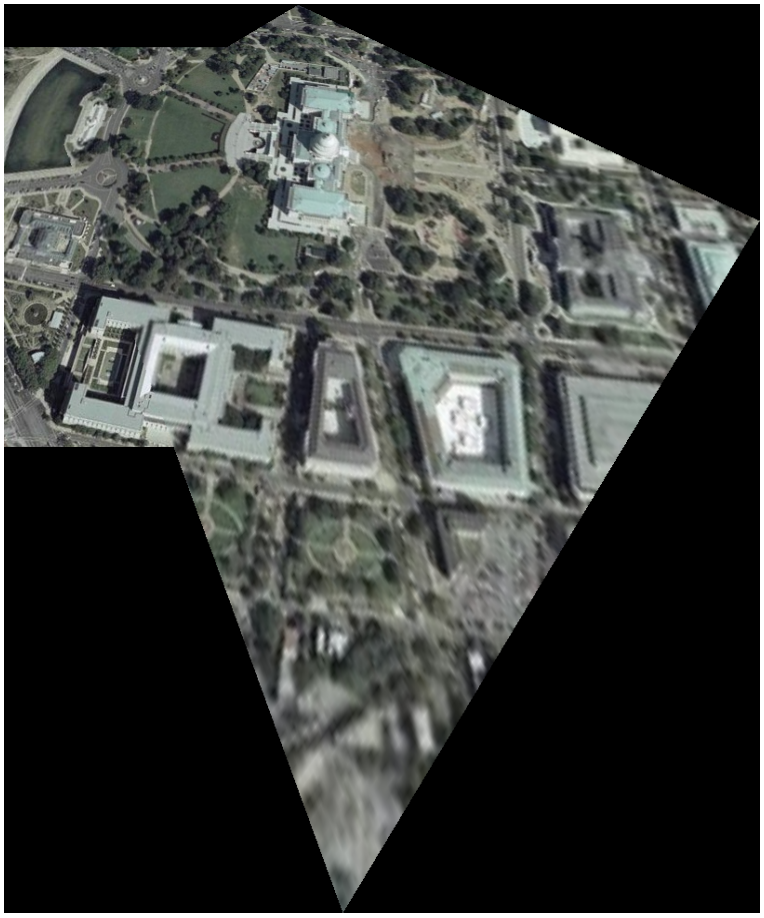

Figure 4: wdc_merge

The data points1 and points2 are saved in the file **points.m**.

(e) Images:

Figure 5: town_source_1

Figure 6: town_source_2

Figure 7: town_mosic

Source: https://www.bilibili.com/bangumi/play/ep264278.

Another example (can be omitted):

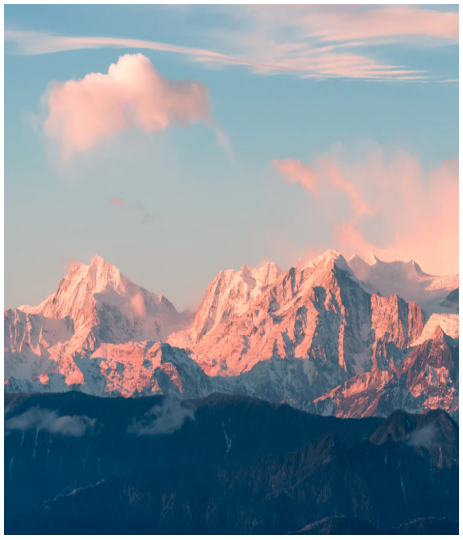

Figure 8: mountain_source_1

Figure 9: mountain_source_2

Figure 10: mountain_mosic

Source: https://unsplash.com/photos/Y8lCoTRgHPE.

(f) Images:

Figure 11: plate

Figure 12: car

Figure 13: car_mosic

Source: ScreenShots from Game Forza Horizon4.
Take the license plate picture, and mosic it on the car. □

Part 3.

(a) Images:

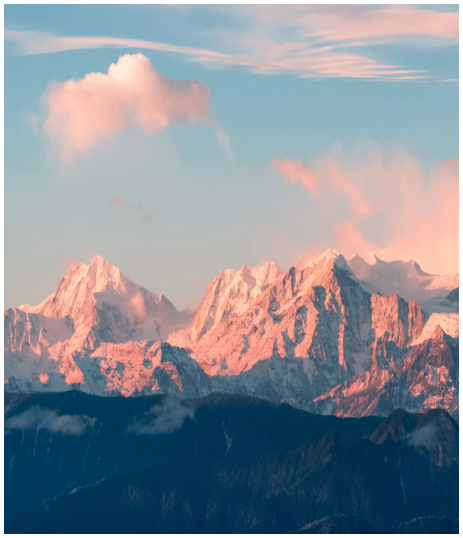

Figure 14: mountain_source_1

Figure 15: mountain_source_2

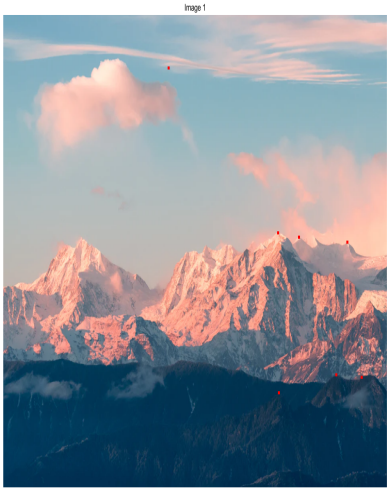

Figure 16: mountain_select_1

Figure 17: mountain_select_2

Figure 18: mountain_mosic_original

Figure 19: mountain_mosic_RANSAC

Source: https://unsplash.com/photos/Y8lCoTRgHPE.

Code: **RANSAC_script.m** and **RANSAC.m**.

Explanation: We implemented a new function **bestH = RANSAC(t1, t2)** to calculate the new H value using RANSAC algorithm. From Figure 16 and Figure 17, we can see there is one bad point selected. Thus, in Figure 18, the image is "broken" and no longer seems "stitch". In Figure 19, we use RANSAC method, and that can eliminate the bad point; therefore, gives us the correct stitch.

Implementation: The user can use the script **RANSAC_script.m** and change the image names to get results with RANSAC algorithm.

(b) Images:

Figure 20: tiles_source

Figure 21: tiles_fronto

Source: https://www.wayfair.com/home-improvement/pdp/msi-bianco-dolomite-marble-mosaic-tile-mvp4112.html?piid=49665325.

Code: **fronto_script.m** and **get_correspondences_fronto.m**.

Explanation: We revised the **get_correspondences()** function to only get the four points of one image. Then, we use these four points as the first input points to compute H, and we selected the four corners of the image as the second input points to compute H. Lastly, as usual, we use the image twice and H to get the results.

Implementation: The user can use the script **fronto_script.m** and change the image name to get results. Either warp result or merge result is okay. □