Ethan Chiang 916154228
Kishan Patel 914983103

PS 2

ECS 174
May 21, 2019

I. Short Answer Problems

1. When using the Hough Transform, we often discretize the parameter space to collect votes in an accumulator array. Alternatively, suppose we maintain a continuous vote space. Which grouping algorithm (among k-means, mean-shift, or graph-cuts) would be appropriate to recover the model parameter hypotheses from the continuous vote space? Briefly describe and explain.

   A:

   - k-means: K-means algorithm is not suitable for the task, due to the fact that its inputs are data points and desired cluster number and outputs are the labeled set. However, for Hough transform the outcome should be the place where maximum vote (area, bin or point) is located.
   - graph-cuts: Graph-cut algorithm takes the feature space as input and divides that feature space into 2 by trying to break the weakest bounds connecting features in the feature space.
   - mean-shift: This algorithm takes a data set as input and converges to the center of mass. This algorithm can be useful when employed to find the maxima of hough space.

2. Consider Figure 2 below. Each small dot denotes an edge point extracted from an image. Say we are going to use k-means to cluster these points positions into k=2 groups. That is, we will run k-means where the feature inputs are the (x,y) coordinates of all the small dots. What is a likely clustering assignment that would result? Briefly explain your answer.

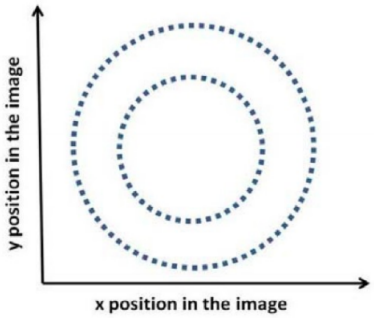


Figure 2: Edge points

A: If we run k-means, the points of the two concentric circles will be split cleanly in half. K-means doesnt do well with clusters within clusters. K-means attempts to guess

the center of clusters and tries to minimize the sum of squared differences among all the points near the supposed cluster centers. Since we pick k = 2 that means we are dividing the dots on the graph into 2 groups. With the k-mean algorithm is with effectively split the circles in the radius center, and the cluster center will be on both sides on the division. Like something below.

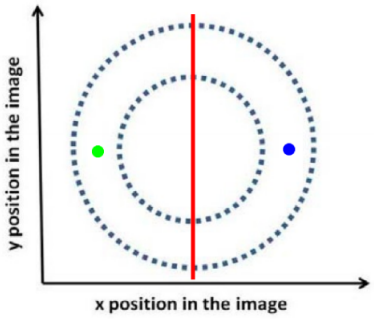


Figure 2: Edge points

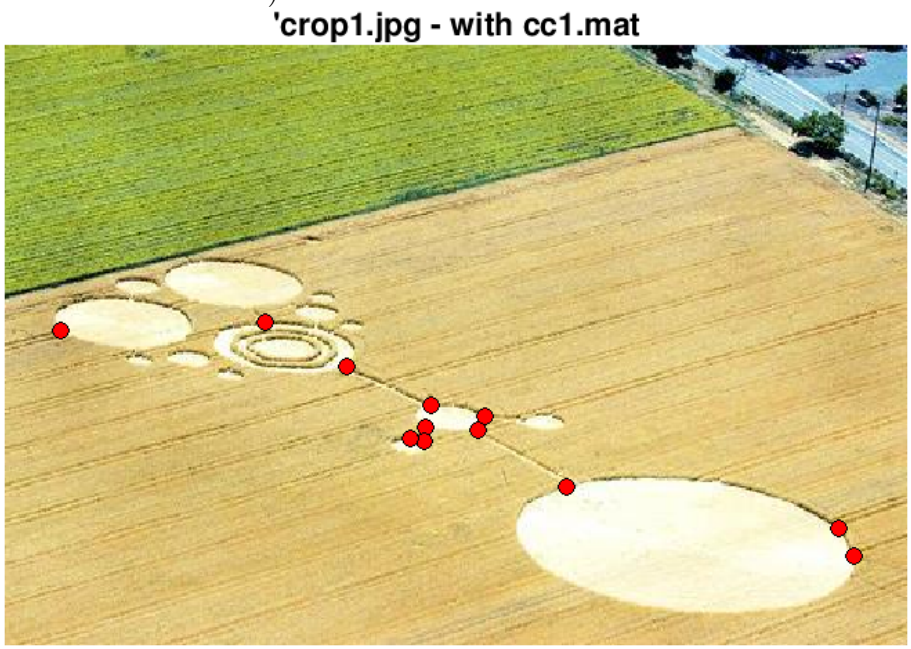
3. Suppose we have access to multiple foreground blobs within a binary image. Write pseudocode showing how to group the blobs according to the similarity of their area (# of pixels) into some specified number of groups. Define clearly any variables you introduce.

   A: [See Next Page]

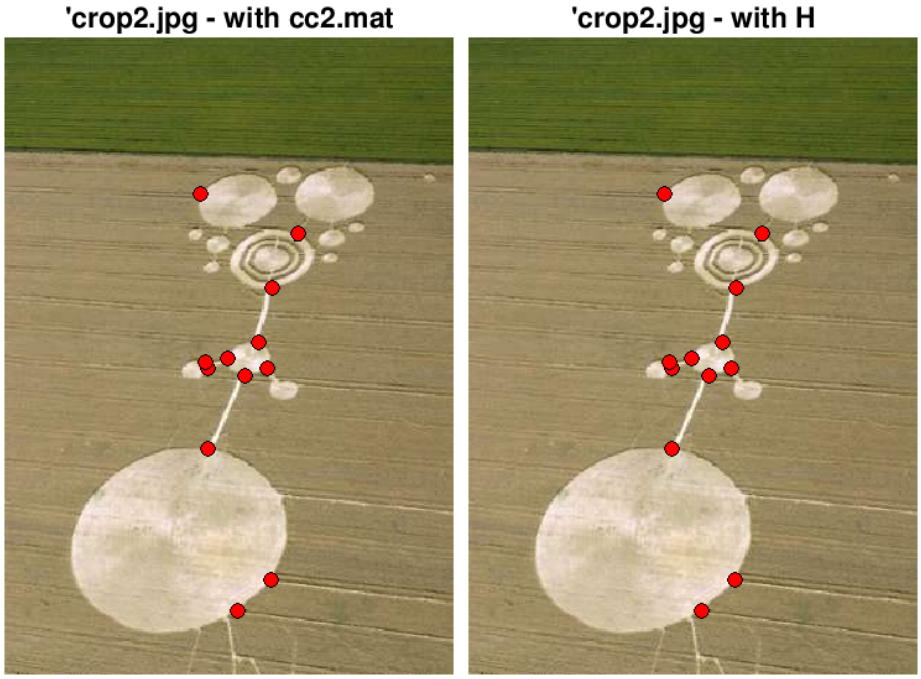
```
algorithm TwoPass(data)
  linked = []
  labels = structure with dimensions of data, initialized with the value of Background

  First Pass

  for row in data:
    for column in row:
      if data[row][column] is not Background

        neighbors = connected elements with the current element's value

        if neighbors is empty
          linked[NextLabel] = set containing NextLabel
          labels[row][column] = NextLabel
          NextLabel += 1
        else

          Find the smallest label

          L = neighbors labels
          labels[row][column] = min(L)
          for label in L
            linked[label] = union(linked[label], L)

  Second Pass

  for row in data
    for column in row
      if data[row][column] is not Background
        labels[row][column] = find(labels[row][column])

  return labels
```

II. Programming problem: content-aware image re-sizing

b)

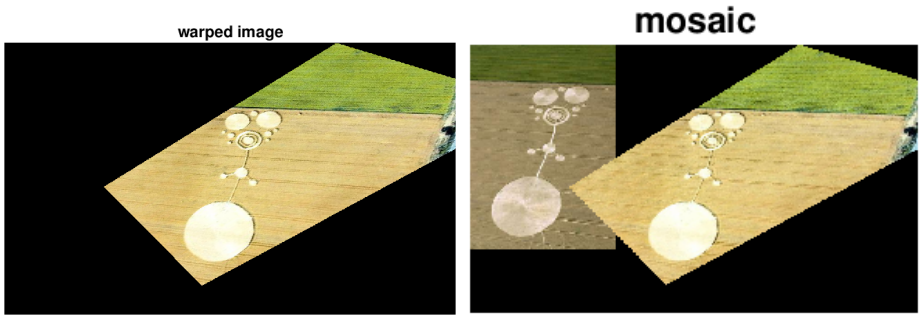


This is 'crop1.jpg' with corresponding points from cc1.mat



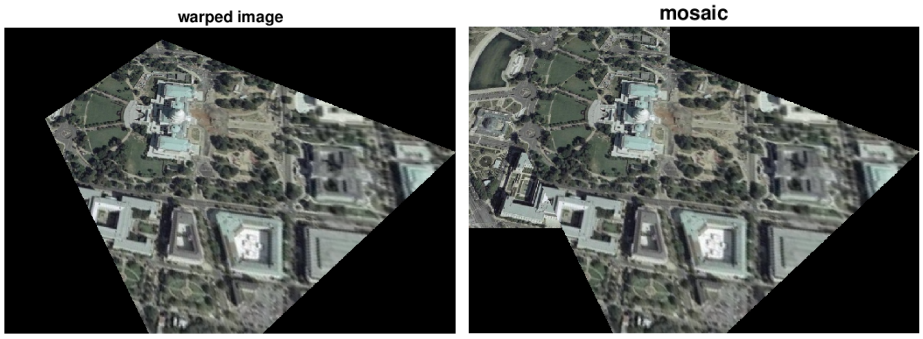
The left one is 'crop2.jpg' with corresponding points from cc2.mat and the right one is 'crop2.jpg' with corresponding points from computeH.m. You can see there is almost no visible difference

d)
pair 1:



The left is the image warped image, on the right is the mosaic of two image. The warped image is in the right orientation, however the image is shifted somehow. With provided with test in part c that our homography is computed correctly, we can only assume that in the inverse warp process something went wrong, or the bounding box is computed poorly.

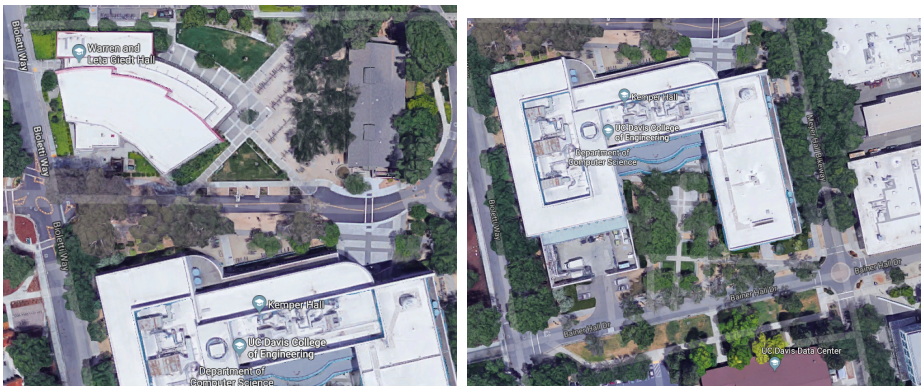
pair 2:



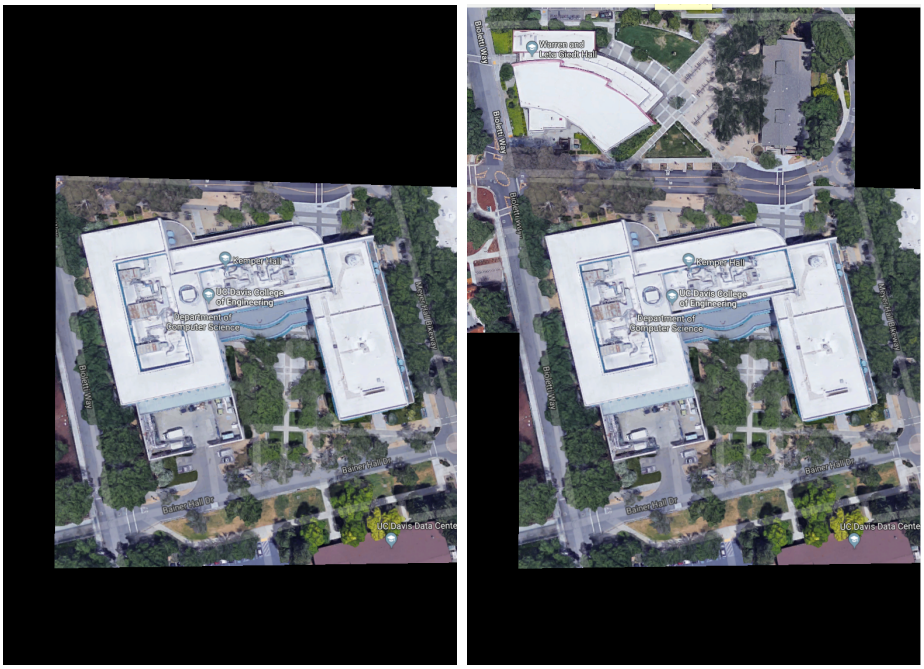
The left is the image warped image, on the right is the mosaic of two image. The extra black space was to ensure I can fit the two images no matter how them is stitched. The warped image is a little of in terms of the orientation, that might due to "bad" correspondences points. But the follow test shows that our code is performing well.

e)
In this part we screen shots the Google map images of the College of Engineering at

UCD. One image display half of the building, the other one display the full building from a view a little to the south

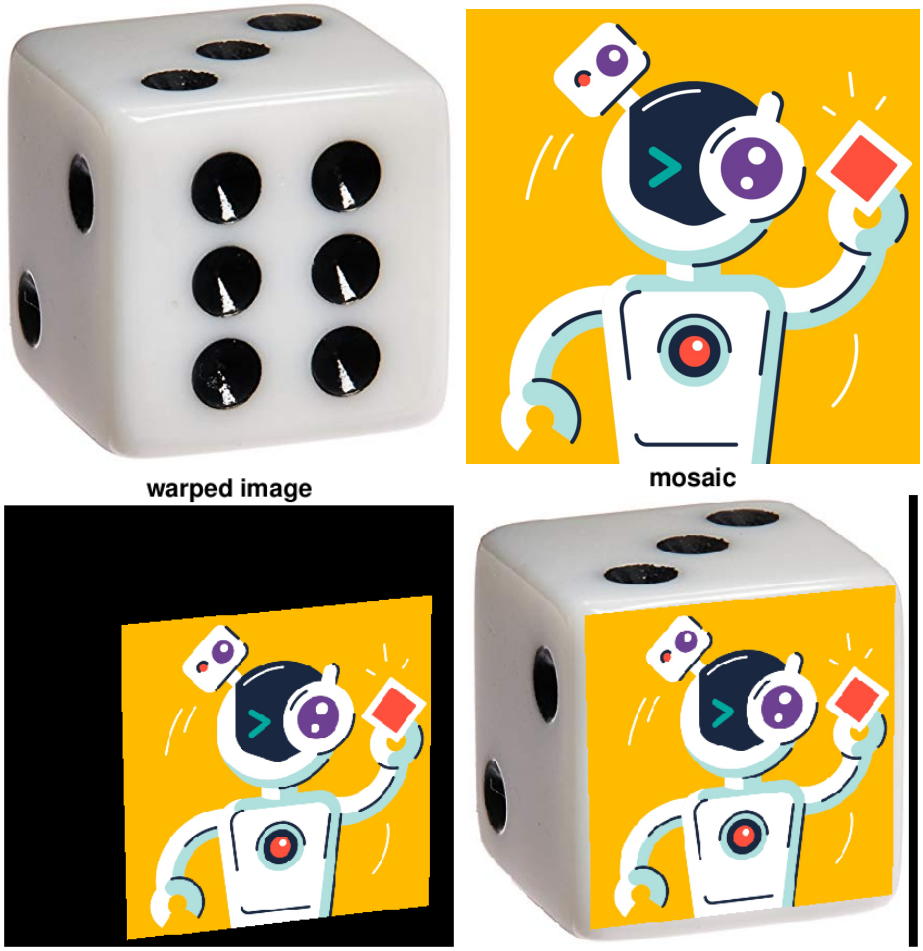


The result in display below. The warp is pretty good besides it is moved too to the south. However, the road on the left side is perfectly align.



f)

For this part, I want to frame the robot cartoon image onto the side of the dice. I pick the corners and center (5 points) as my correspondences points for both images. The results are shown below:


warped image
mosaic