Orion Tran

Toby Lee

1. When using the Hough Transform, the grouping algorithm graph-cuts, would be the most appropriate to recover the model parameter hypotheses from the continuous vote space. The reason why graph-cuts would be most appropriate is because it links every pair of pixels and assigns an affinity weight for each edge. Also, we could break graphs into segments. We can delete links that have low affinity and keep the segments that have high affinity. In a Hough Transform, we would like to detect arbitrary shapes and define its shapes by its boundary points and reference point. It is used mainly for lines and circles that are defined by a set of boundary points. This is exactly what graph-cuts are good at. Graph cuts are able to connect those shapes with edges that have high affinity while mean shift and k-means doesn't. Mean shift is not used to assume shape in clusters. It is used mainly for features such as color, gradients and texture. Similarly, k-means is used for color and texture not for detecting shapes, which makes graph-cuts the best candidate.
2. From using K-means clustering with 2 groups, after we run it the clustering assignment will have two center points that will divide all the feature inputs into two groups. Each center point will get exactly half of the feature points and the center point will be positioned exactly where half of the feature points are closer to one center and the other half of the feature points are close to the other center point. That is because in the beginning of the algorithm, each center point will be randomly placed. After that it will find the centroid of the points it owns, then it will move to that position. It will repeat that process until half of the data points are in one group and the other half is in the other group.
3. Variables:

   outsidepixel: boundary pixel of blob

   position: sum of the position

   centerofblob: the center pixel of the blob.

   radius:(outsidepixel – centerofblob)

   area: (radius)^2 * pi

   circularity:comparison of blobs based on their area

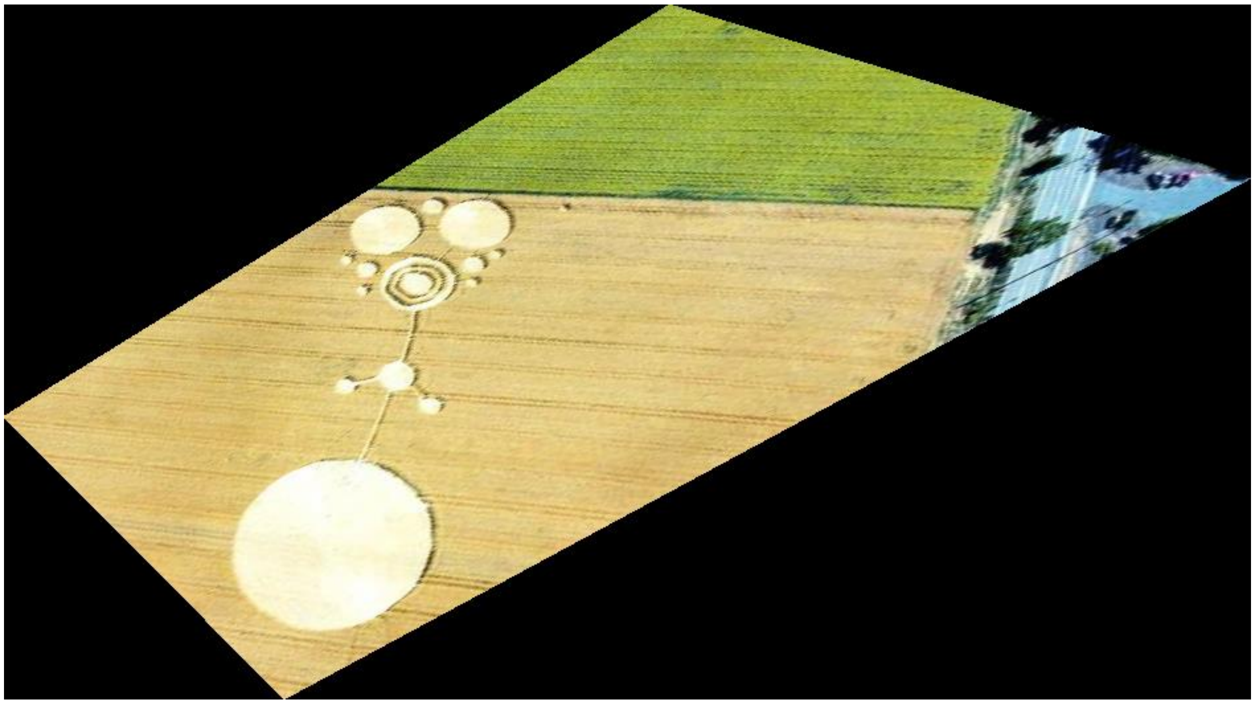
   1. Find the center of each blob
   2. Get the radius of each blob
   3. Calculate the area of each blob from the radius.
   4. Group blobs in their similarity in area. Using kmeans

Functions:

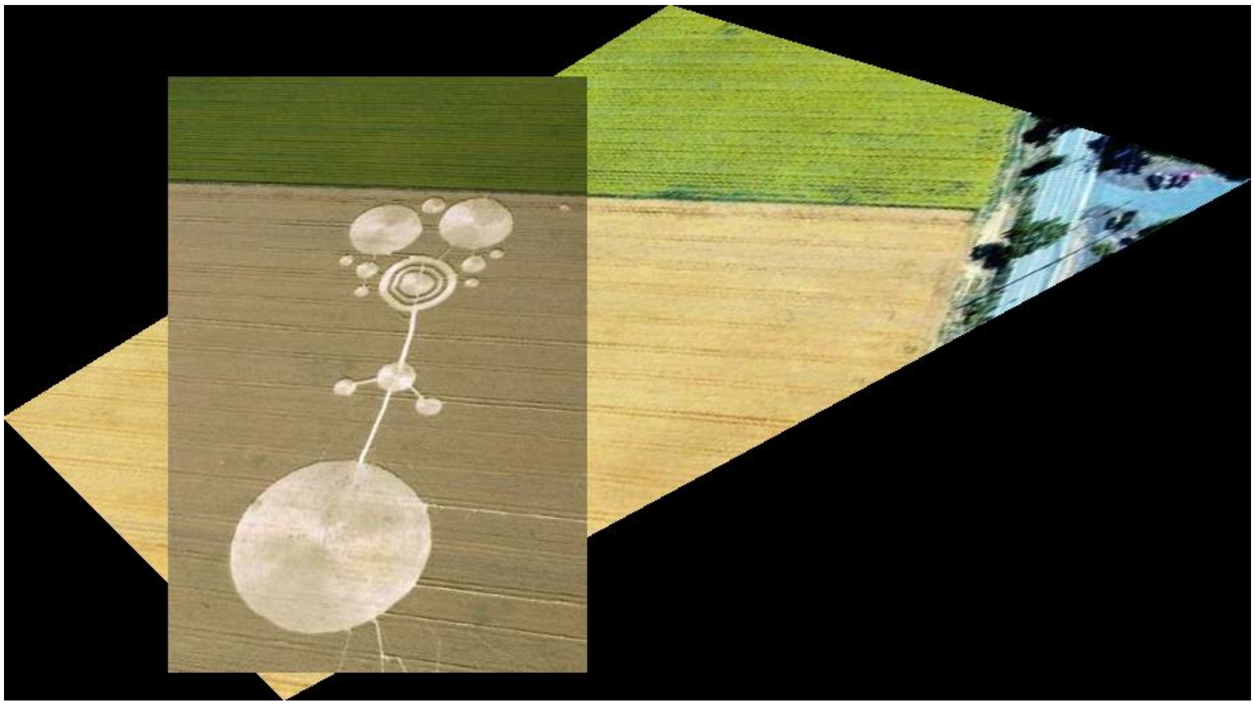
```
Findcenter(blob)
for each pixel
    position += position(pixel)
```

```
        centerofblob = position/size(blob)
        return centerofblob


area(blob)
radius = outsidepixel – centerofblob
for each blob
        area = pi*radius^2
        circularity = area/ size(blob)
return circularity


kmeans(circularity[], k)
        return kmeans = circularity []
```

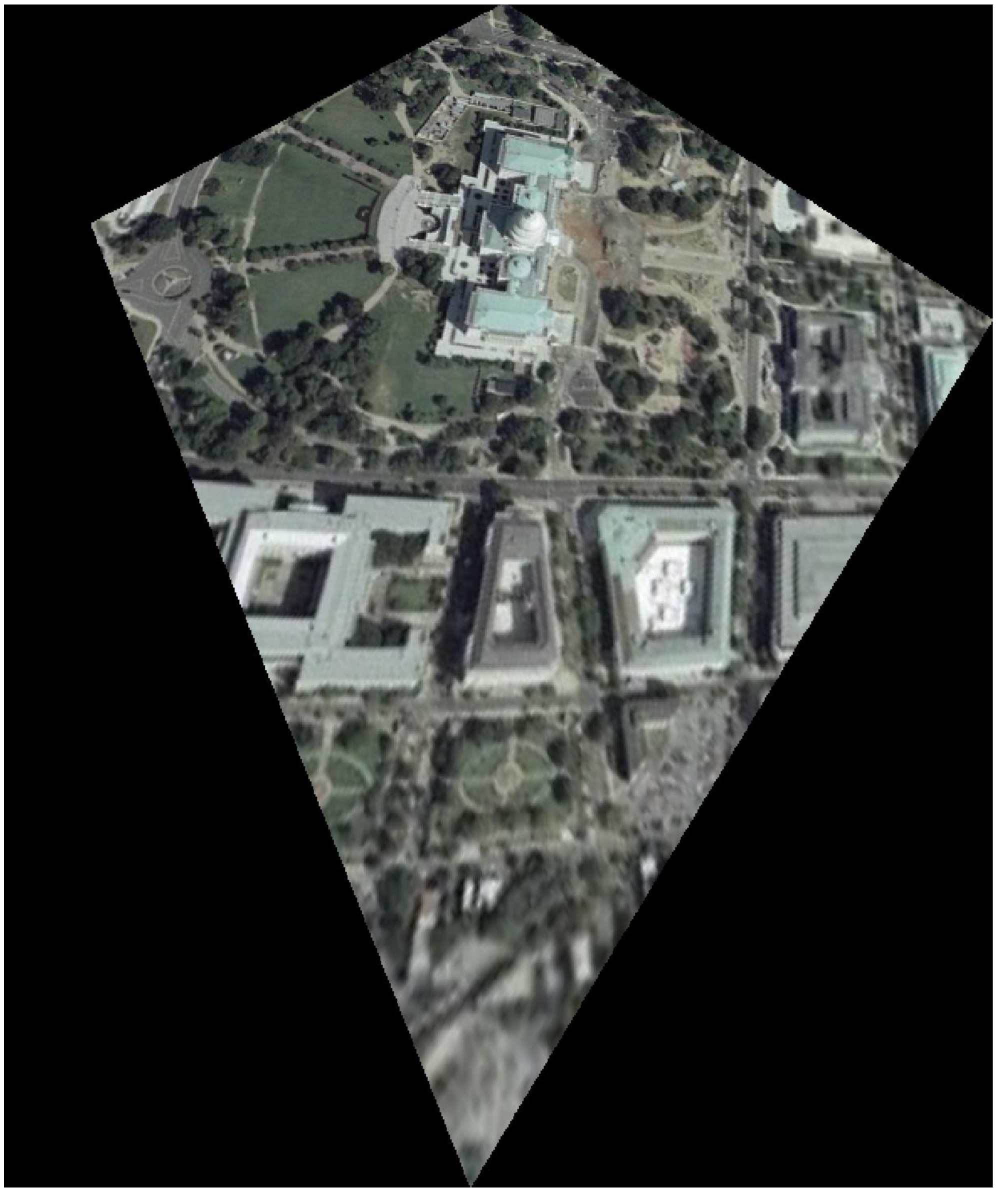
2d. Crop Warped



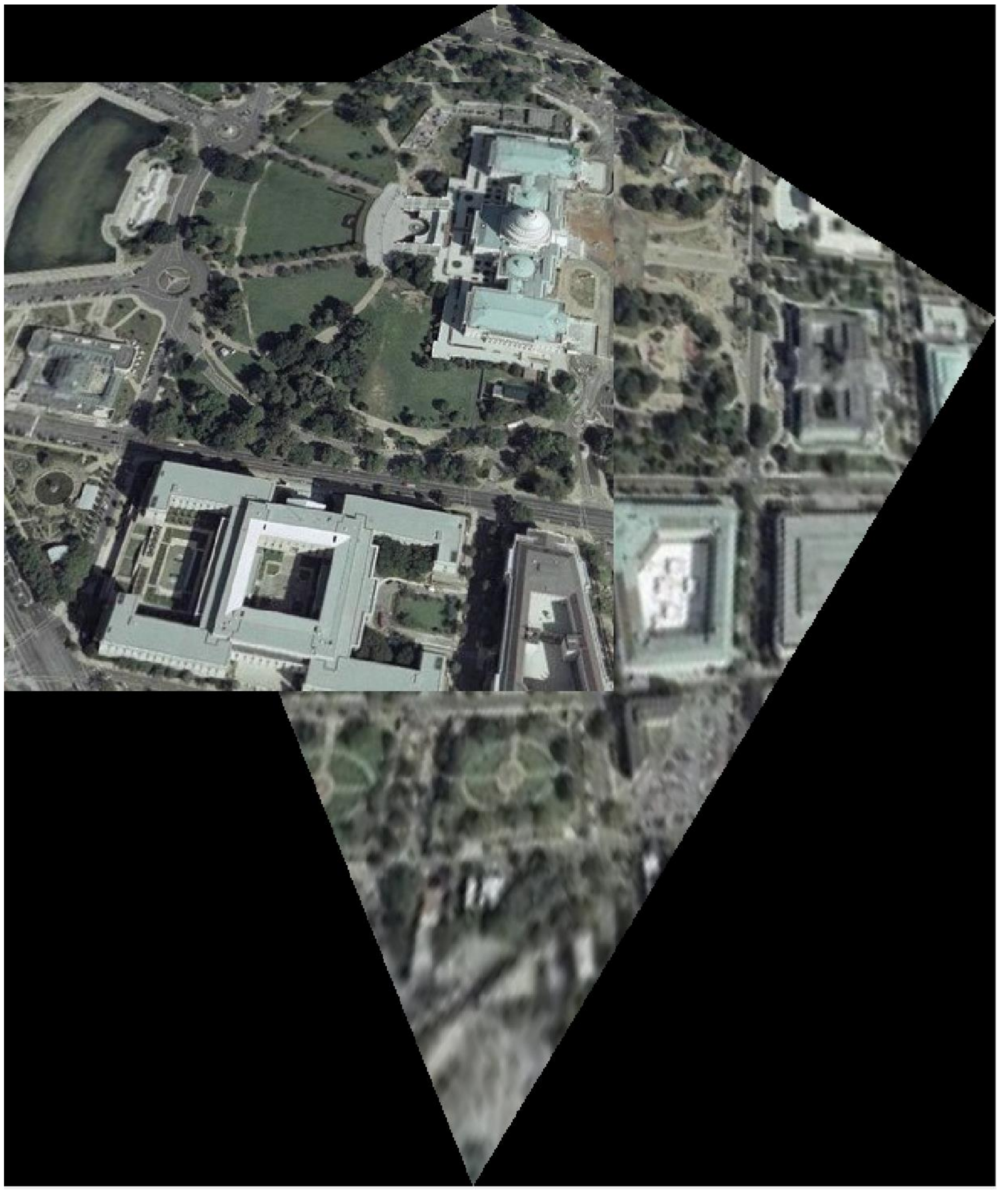
2d. Crop Mosaic



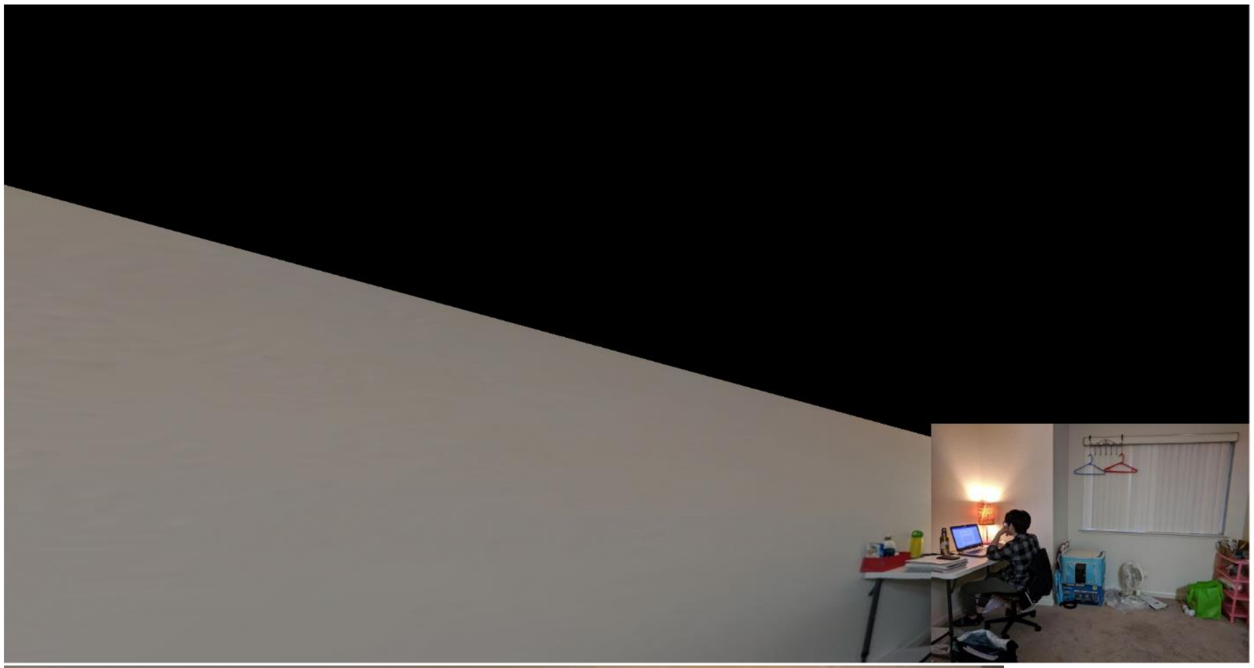
WDC Warp



WDC Mosaic



2e. Mosaic



2f.