Xie Zhou 912143385
ECS 174
05/22/2017

PS2

Short Answer Problems

1) Yes, the resulting representation is sensitive to orientation. Horizontal edge filters can detect horizontal patterns (at the corresponding scale); vertical edge filters can detect vertical patterns, etc. The orientations of a filter determine what kind of pattern in the picture it will give a higher score on.

For example, for this texture, horizontal edge filters would give higher scores than vertical edge filters because there are more horizontal edges.

2) The resulting 2 clusters would contain equal number of dots. Each cluster would have half of the "dot circle". (So, the cutting boundary would be a line going through the center of the 2 circles.) It's not a good result for spherical clusters.
Explanation:
First, suppose we pick the 2 initial centroids, p1, p2. Denote l1 as the line connecting p1, p2. Denote the resulting cluster boundary l2, where l2 is perpendicular to l1 and goes through the midpoint of l1.
Second, we pick the new centroids from each group. The previous centroid of the larger cluster would shift father away from the center of the circles; the previous centroid of the cluster w/ less points would move closer to the center of the circles.
Eventually, the cutting boundary would go through the center of the circles. Therefore, the two clusters would each have one side of the initial group of dots.

3) Mean-shift algorithm is the most appropriate. In the discrete voting space, we look for "grids" with the higher number of votes. Similarly, in the continuous space, we want to look for the local peaks. By the mean-shift algorithm with 2-dimensional feature space, we can find multiple modes. These modes are the local maxima/center of masses of each of the search windows. Therefore, with these local maxima, we would be able to recover the parameters in the image space.

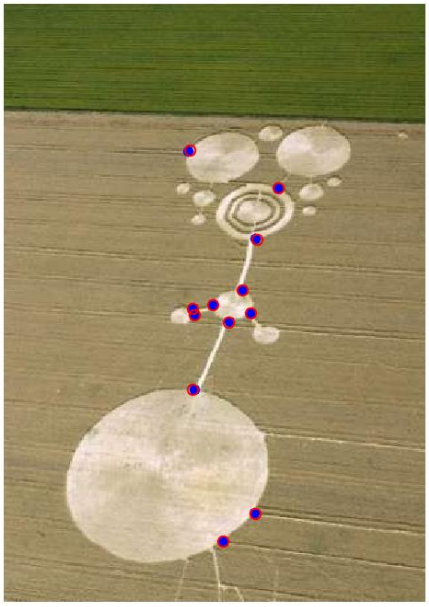
Programming

a) [p1_points, p2_points] = getPoints(im1, im2)
This function returns user selected points from the input image and the reference image.

b) H = compute(t1, t2)
This function returns a 3x3 homography matrix.

verify_H.m
This script loads “crop1.jpg”, “crop2.jpg”, uses the pre-selected points, p1, p2 to generate the H matrix, and plots the mapped points and original points p2 onto the reference image.

In the reference image below, the blue filled circles are the pre-selected points: p2; the red hollow circles are the mapped points after the transformation.

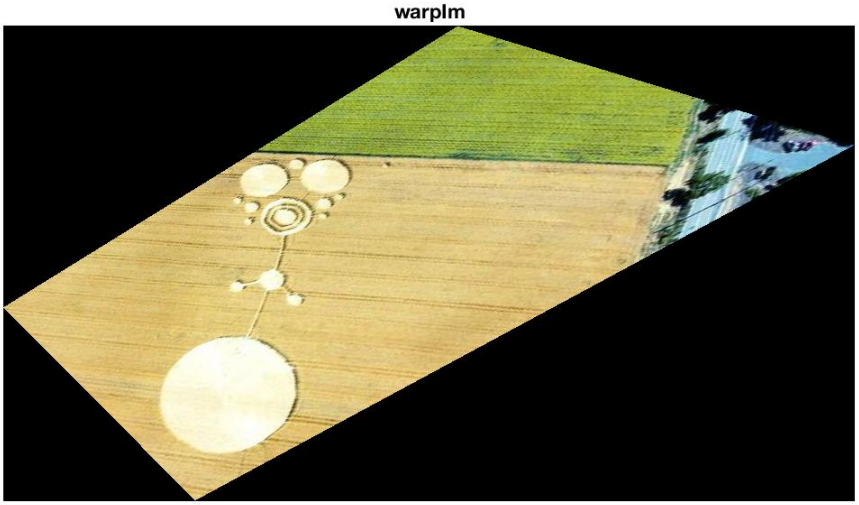


p2b_verify.jpg

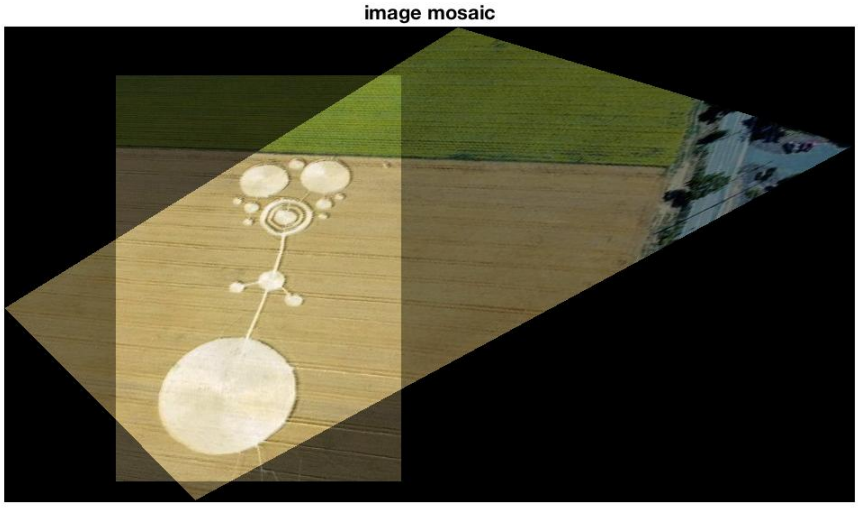
c) [warpIm, mergIm] = warpIm(inputIm, refIm, H)
This function returns a warped image of the input image in the reference image frame, and a merged image of the warped + reference image.

d) Pair 1:
p2_d_pair1.m
This script loads crop1.jpg and crop2.jpg and the loads two given set of points cc1.mat, cc2.mat.

crop_warp.jpg:



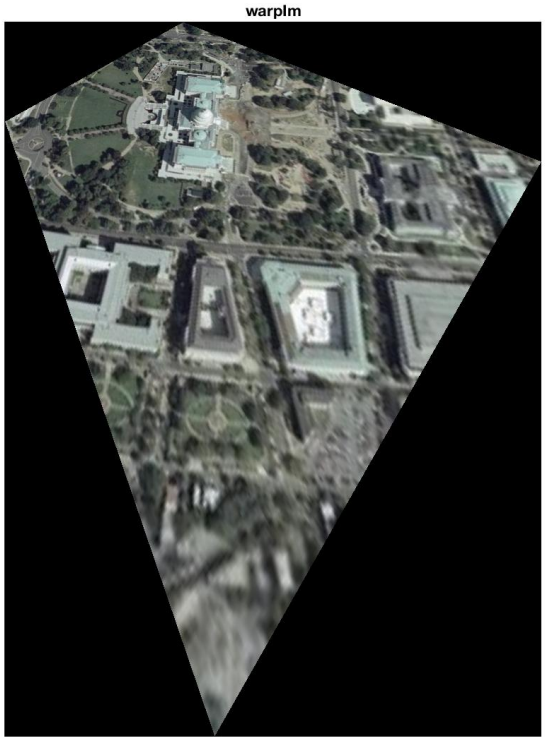
crop_merge.jpg:



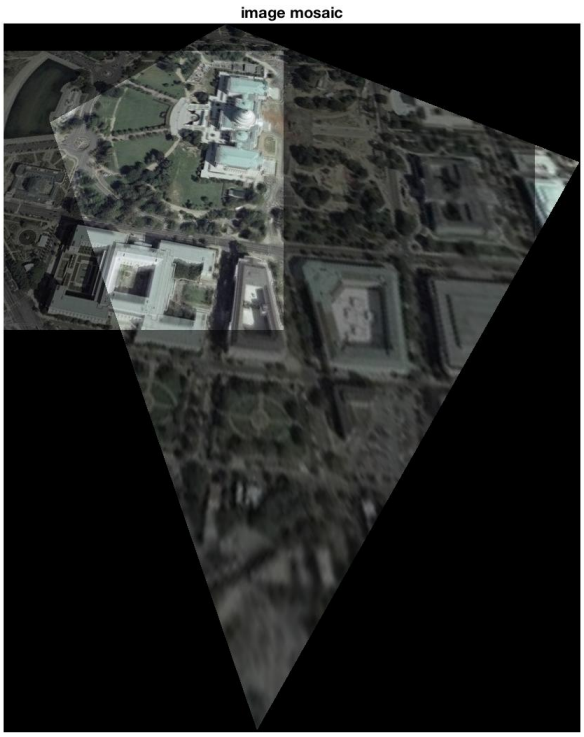
Pair 2:

p2_d_pair2.m
This script loads wdc1.jpg and wdc2.jpg and loads 2 sets of pre-selected points (10 points) stored in points.m.

wdc_warp.jpg:



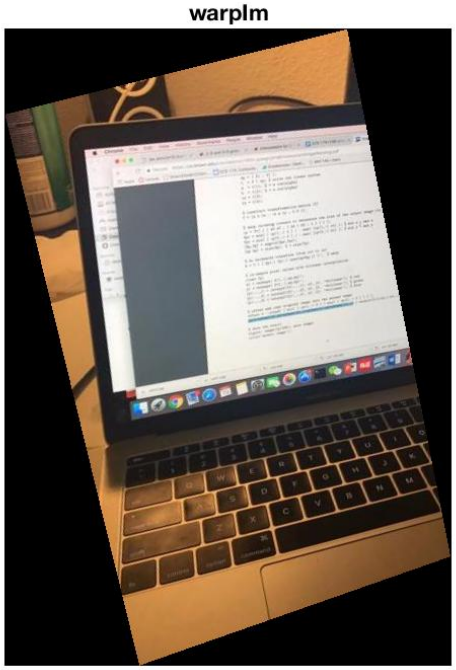
wdc_merge.jpg:



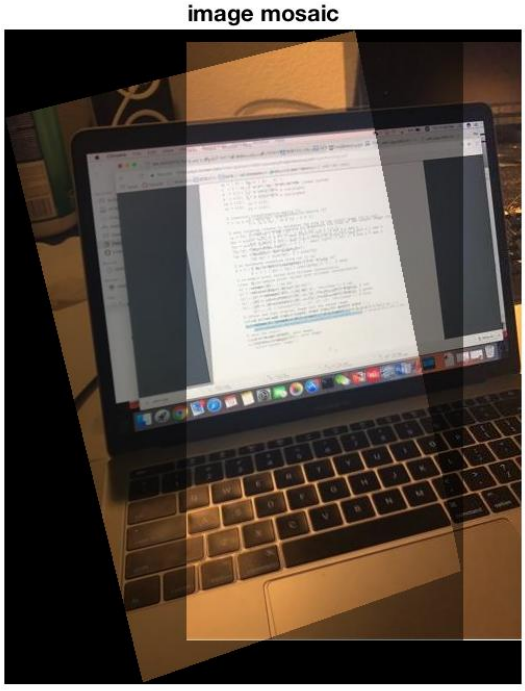
e) test_own.m

This script loads laptop1.jpg and laptop2.jpg and loads 2 sets of pre-selected points (10 points) stored in own_points.m.

laptop_warp.jpg:



laptop_merge.jpg:



f) replace.m

This script loads cola.jpg and board.jpg and loads 2 sets of pre-selected points (4 edge points in cola image; 4 frame points in board image) in replace_points.m.

replace_warp.jpg:

warpIm

replace_merge.jpg:

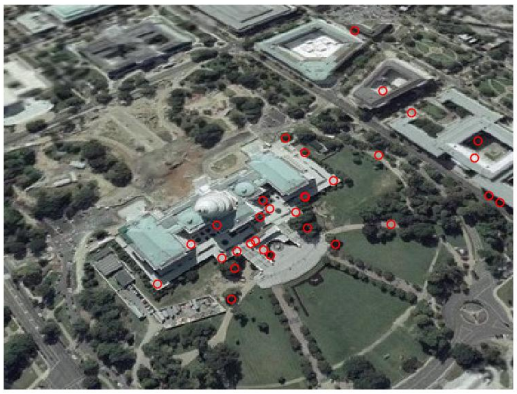
image mosaic

Extra credit

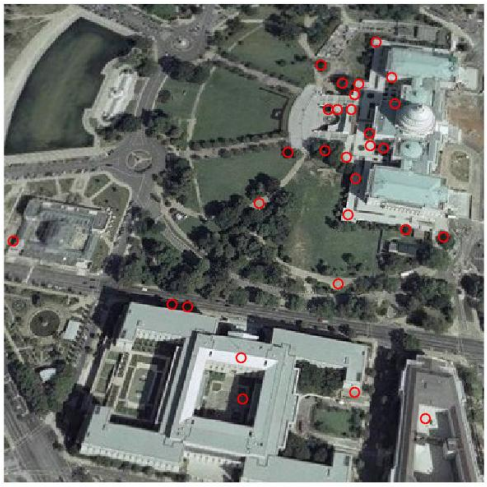
a) p3_a.m

This script uses the matlab SIFT library to accomplish automatic point detection. vl_sift and vl_ubcmatch functions were used.

To select the appropriate number of "good" local points in each image, I set the 'peak threshold' parameter for vl_sift function to be 12. Compare to the number of selected points generated when the threshold was 0, the number decreased from 168 to 35.



p3_a_inputpoints.jpg



p3_a_refpoints.jpg

There are 2 pairs of points in the images that obviously does not match. (One overlapping point in the middle, and one at top right in the first image.) Other 33 pairs of points pair very well. There are other ways of stressing the number of points such as ranking the scores, and selecting the top corresponding highly scored matches of pairs (vl_ubcmatch).

b) Didn't do this part.

c) p3_c.m

Reads squaretile_input.jpg and loads p3_c_points.mat (4 pre-selected points)

The output square image is 200x200.

p3_c_origional.jpg

p3_square.jpg