

# IE 590 – Homework III

Naveen Madapana – 0029093466

Code: [link is available here](#)

## Q1. Stitching two images

Attributes:

Features: *SURF*

Filename: *hw3\_q1.py*

Results:



Figure 1. Original Images to stitch

```
73
74 (kp1, desc1) = compute_surf(frame1)
75 (kp2, desc2) = compute_surf(frame2)
76 M = match_kps(kp1, desc1, kp2, desc2, knn_ratio = 0.75, ransac_reproj_thresh = 4.0)
77 if M is None:
78     print('Less than 4 points to find homography')
79
80 # Warping perspective to stitch the images together
81 (good_matches, H, status_flags, error) = M
82 print 'No. of inliers: ', np.sum(status_flags.flatten())
83 print 'No. of outliers: ', np.sum(status_flags.flatten() == 0)
84 print 'Average residual error: ', error
85 final_shape = (frame1.shape[1] + frame2.shape[1], max(frame1.shape[0], frame2.shape[0]))
86 result = cv.warpPerspective(frame1, H, final_shape)
87 cv.imshow('partial result', result)
88 result[0:frame2.shape[0], 0:frame2.shape[1]] = np.uint8(0.5*(np.float32(result[0:frame2.shape[0], 0:frame2.shape[1]]) +
89
90 # Show matches
91 matched_image = draw_matches(frame1, kp1, frame2, kp2, good_matches, status_flags)
92
93 cv.imshow('Matches', matched_image)
94 cv.imshow('Final Result', result)
95
96 No. of inliers: 476
97 No. of outliers: 102
98 Average residual error: 0.604967573025
99 [Finished in 69.8s]
```

Figure 2. Code and results

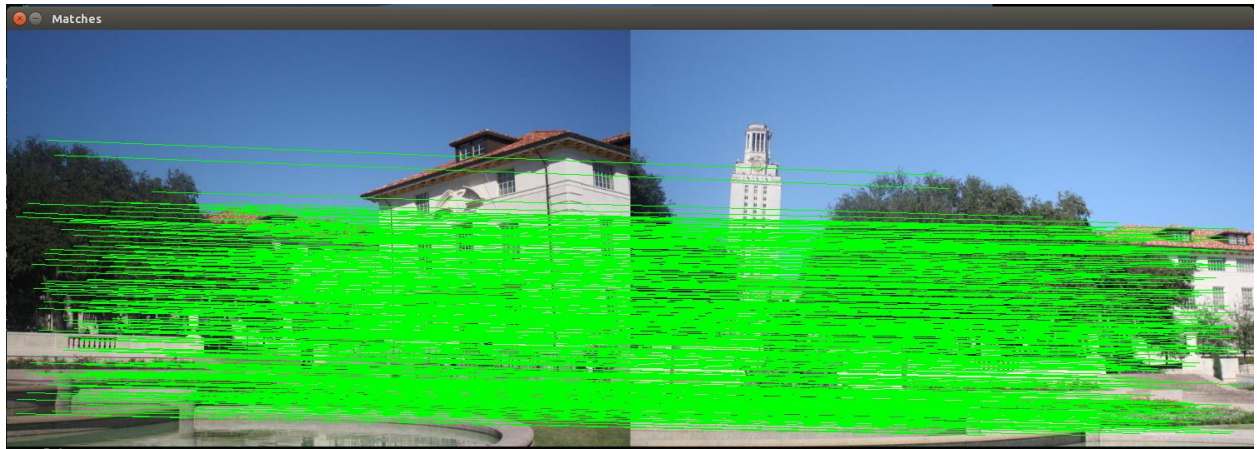


Figure 3. Drawing the key points using SURF feature descriptor



Figure 4. Rotating the Image using the estimated homography

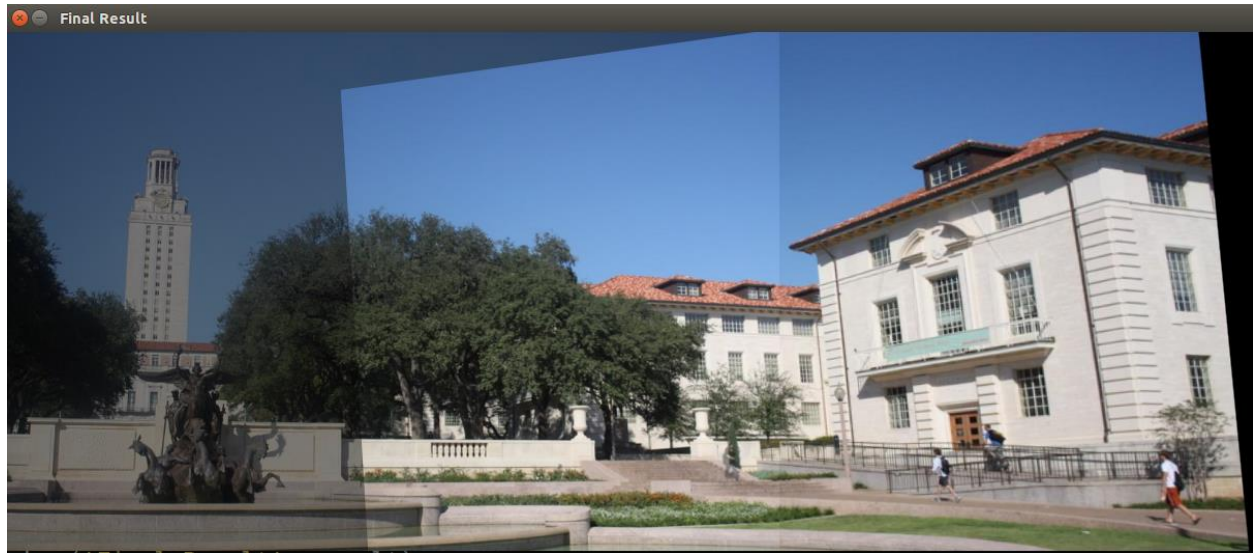


Figure 5. Final result of stitching of images

## Q2. Comparing old and new images

Attributes:

Features: *SURF*

Old Image: [Link](#)

Results in the form of [Video](#)

Results:





Figure 6. Old image of PMU







Figure 7. New images of PMU

```

Matches
hw3_q1.py  hw3_q2.py
1 import numpy as np
2 import cv2 as cv
3
4 left_image_path = '../Images//pmu_old.jpg'
5 right_image_path = '../Images//pmu_new.jpg'
6
7 bfm = cv.BFMatcher()
8 surf = cv.xfeatures2d.SURF_create() #nothing
9 knn_thresh = 0.75 # 0.75
10 ransac_reproj_thresh = 4.0 #4.0
11
12 # Record a video
13 cap = cv.VideoCapture(0)
14 fourcc = cv.VideoWriter_fourcc(*'XVID')
15 out = cv.VideoWriter('output.avi',fourcc, 20.0, (500,500))
16
17 def compute_surf(frame):
18     gray = cv.cvtColor(frame, cv.COLOR_BGR2GRAY)
19     (kps, desc) = surf.detectAndCompute(gray, None)
20     kps = np.float32([kp.pt for kp in kps])
21     return (kps, desc)
22
No. of inliers: 12
No. of outliers: 34
Average residual error: 1.04794532928

```

Figure 8. Code and Result



Figure 9. Showing new image on top of the old image