

Projective Transformation and Panorama
Stitching
CS698U- Computer Vision

Vaibhav Nagar (14785)
Email: vaibhavn@iitk.ac.in

April 19, 2017

1. Projective Transformation using DLT

Four corner points are selected from two images (with two different perspective views) and homography matrix is calculated using Direct Linear Transform algorithm (DLT) by mapping these four points from one image to another. To warp the image using Homography, warpPerspective method is used from openCV package of python.

Results of five different images whose perspective distortion is corrected:

- Image1:



Figure 1: From



Figure 2: To



Figure 3: Projection

- Image2:



Figure 4: From

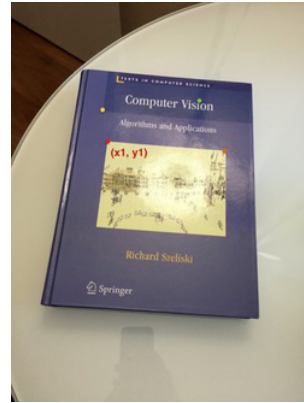


Figure 5: To

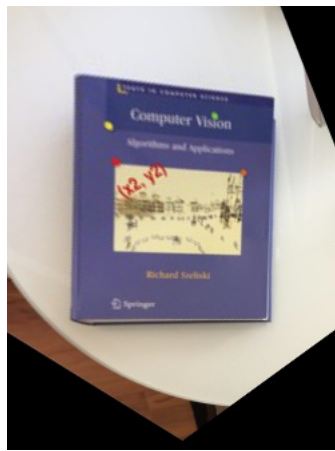


Figure 6: Projection

- Image3:



Figure 7: From



Figure 8: To



Figure 9: Projection

- Image4:



Figure 10: From

Figure 11: To



Figure 12: Projection

- Image5:



Figure 13: From



Figure 14: To



Figure 15: Projection

2. Panorama Stitching

- **Interest Points:** Interest points of an image are located using SIFT of python openCV package. Then these interest points sharing same descriptions are matched between two images using FLANN (Fast Library for Approximate Nearest Neighbor) based matcher and from KnnMatch method two nearest neighbors are computed which is then filtered out by applying ratio test on those two neighbors.

A interest point is discarded if ratio of distance between its two nearest neighbors is greater than 0.7. This helps in getting rid of around 90% of false matches.

- **Ransac Algorithm:** Randomly four interest points with their match are selected from putative matches and homography is calculated. Error is calculated by projecting an image using homography and L2 distance is computed. Number of inliers are compared with threshold inliers (initially set to 0) based on threshold distance. Ransac algorithm is run for n number of iterations (default 20) and homography of an iteration whose number of inliers are maximum is recorded.
- **Stitching:** Images are projected using homography and then stitched by calculating final dimension with offset of panorama from first image's corner point and then pixels of second image is pasted on it.

- Panorama1:



Figure 16: 1



Figure 17: 2



Figure 18: 3

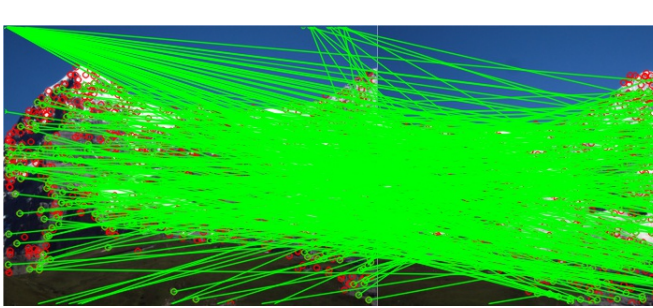


Figure 19: matches 1-2

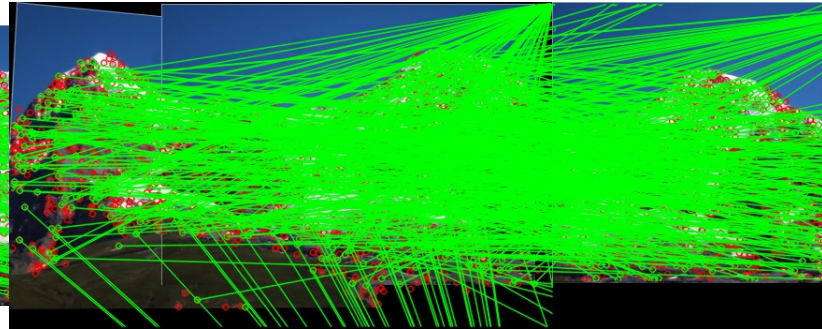


Figure 20: matches 2-3

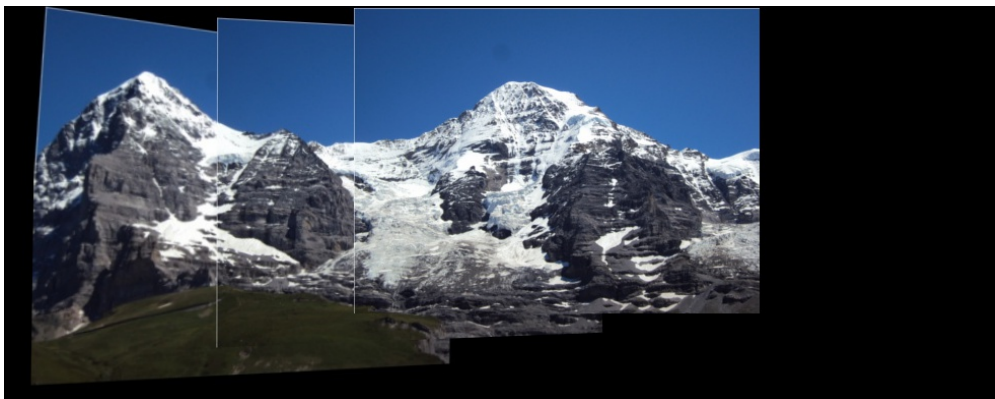


Figure 21: Panorama

- Panorama2:



Figure 22: 1



Figure 23: 2

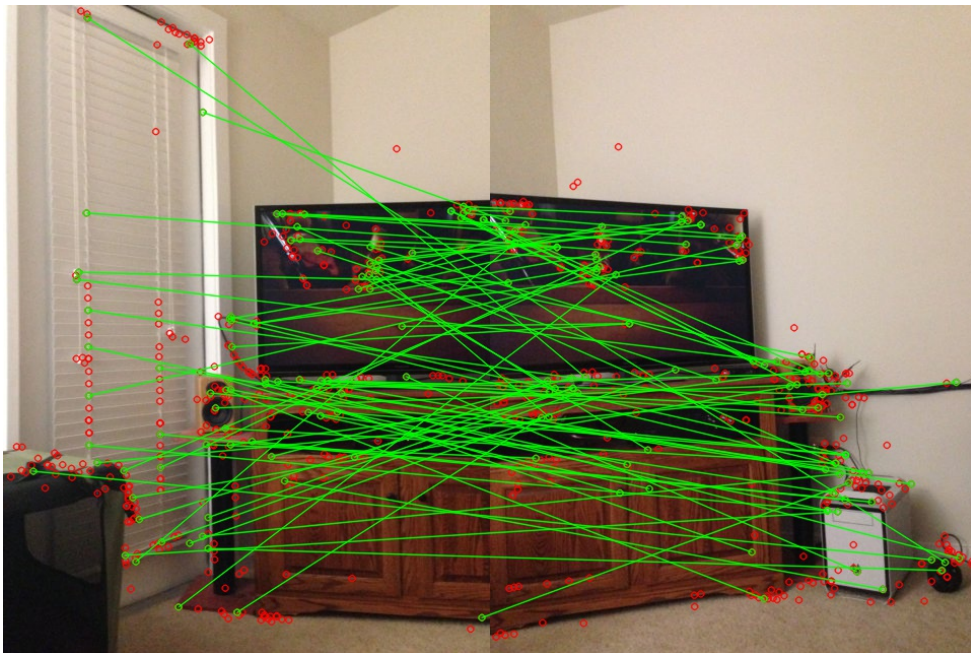


Figure 24: matches 1-2



Figure 25: Panorama

- Panorama3:



Figure 26: 1

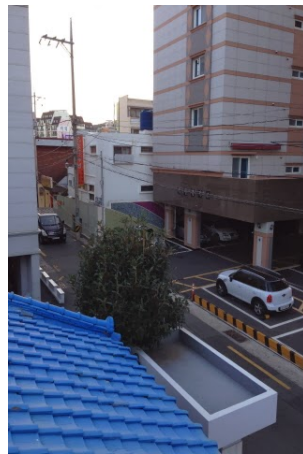


Figure 27: 2

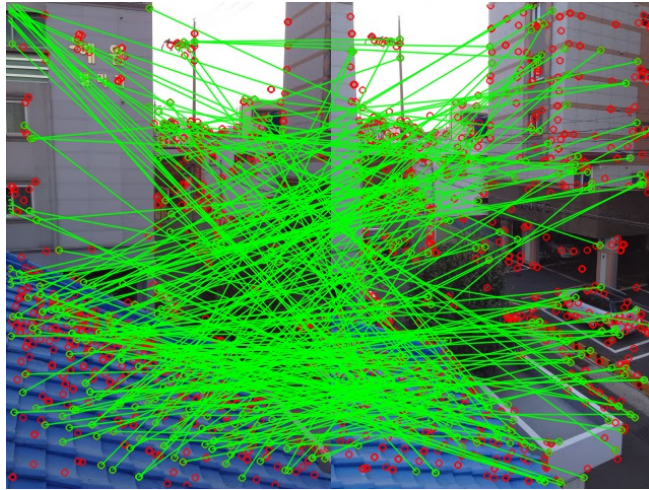


Figure 28: matches 1-2



Figure 29: Panorama

- Panorama4:



Figure 30: 1



Figure 31: 2

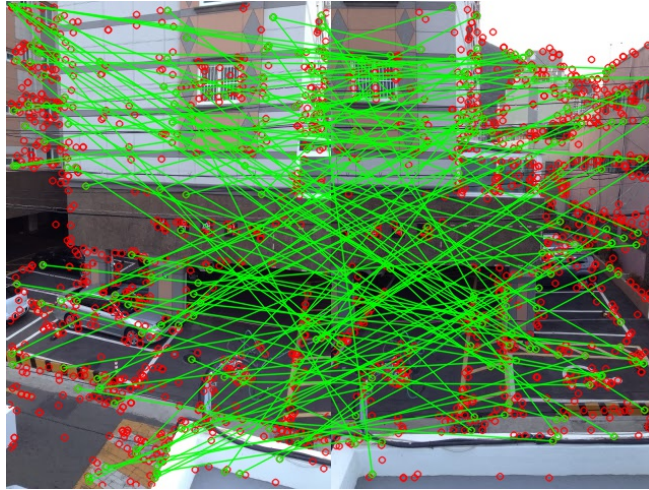


Figure 32: matches 1-2

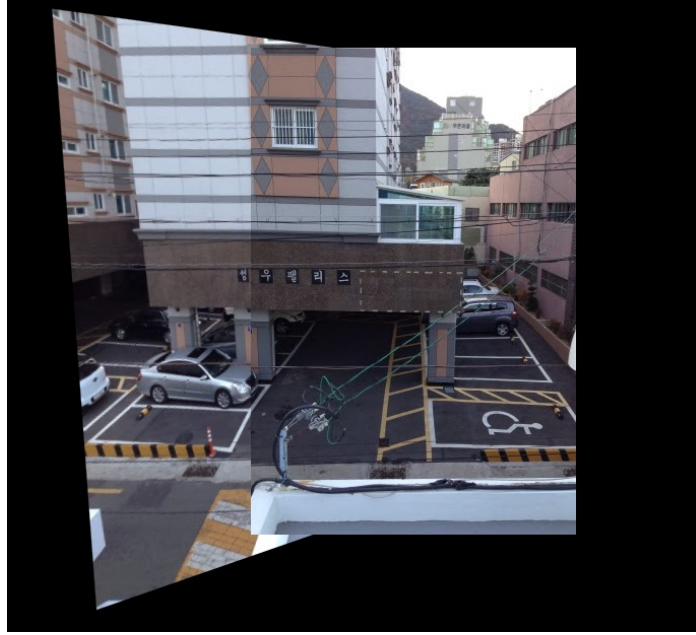


Figure 33: Panorama

- Panorama5:



Figure 34: 1

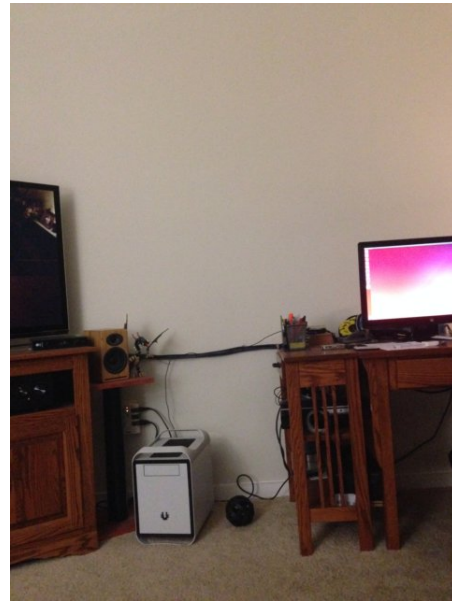


Figure 35: 2



Figure 36: matches 1-2

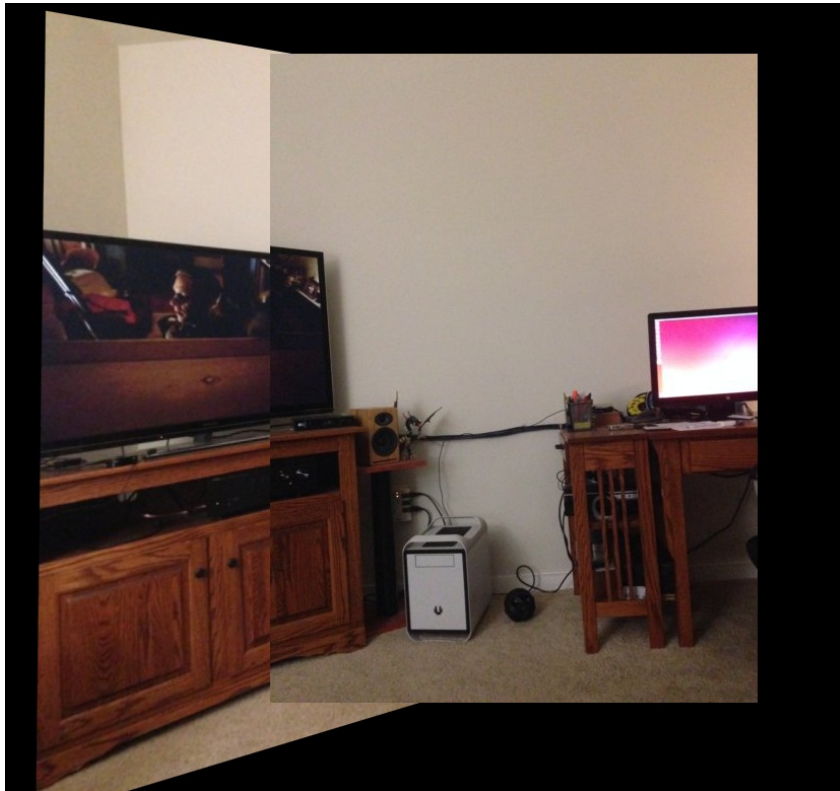


Figure 37: Panorama