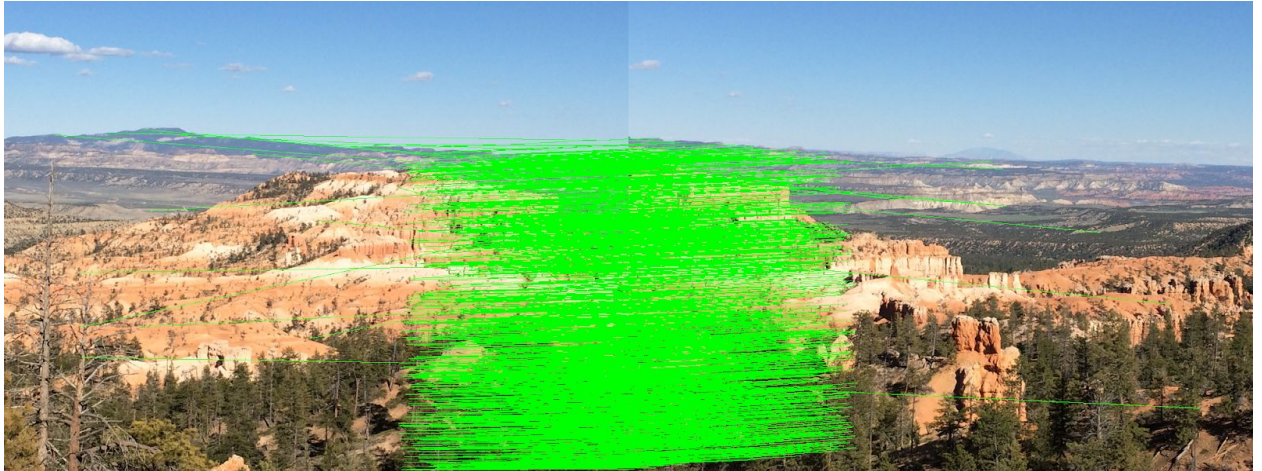


Report: Assignment - 2

Chris Andrew
2018701019

Panorama Stitching

- Our panorama stitching algorithm consists of four steps:
 1. Detect keypoints (DoG, Harris, etc.) and extract local invariant descriptors (SIFT, SURF, etc.) from the two input images.
 2. Match the descriptors between the two images.
 3. Use the RANSAC algorithm to estimate a homography matrix using our matched feature vectors.
 4. Apply a warping transformation using the homography matrix obtained from 3.
- **Challenges faced:**
 - Understanding perspective transform
 - Understanding Homography computation using RANSAC
 - Understanding the stitching method after homography is computed.
- **Procedure:**
 - For the first part, we use SIFT descriptors and keypoints to find the matching pair of points between the two images. The keypoints and descriptors for the two images are computed separately and then the nearest neighbour algorithm is used to match keypoints.
 - Thresholding is done to ensure that only those keypoints whose nearest neighbour is 0.75 times the distance of its second nearest neighbour is chosen. This criteria determined valid matches. An example of two images and their matched points are given below.[keypoints.py]



- Once we have the matched keypoints, we randomly sample 4 pairs to compute the homography. We then find the error of this homography by projecting the points using perspective projection and calculating the distance between the projected point and the actual point. This is done for both the base image and the projected image. We then measure the distance and label the point as an inlier or outlier (distance > 0.5). We repeat this process for 5000 iterations to get the best homography (least number of outliers). [homography.py]
- Once we have the homography, we use this to transform one image to the other images plane and then combining them. We first project the projection image to the base image perspective plane, and then we add the base image to the same plane. This is repeated for all images that need to be stitched together iteratively. Each time the new stitched image is treated as the base image for the next iteration. [stitch.py]

Image projected to perspective plane of base image:



Results for given images:



this week.



Own images:

