# CS543/ECE549 Assignment 3

**Name:** Hongbo Zheng

**NetId:** hongboz2

**Part 1: Homography estimation**

**A: Describe your solution, including any interesting parameters or implementation choices for feature extraction, putative matching, RANSAC, etc.**

First of all, both images are loaded and converted to grayscale. Then, the SIFT detector is used with the commands cv2.SIFT\_create() and sift.detectAndCompute to acquire the feature points in both images since it’s more robust than the Harris detector. After that, scipy.spatial.distance.cdist is used to compute the pairwise distance between the two descriptors. Finally, only the coordinates pairs that have the distance between them less than the distance threshold are kept, while the others are filtered out because their distances are larger than distance threshold.

Then, the RANSAC algorithm is run on the selected coordinates pairs returned by sift detector, which have distance less than distance threshold. The RANSAC algorithm is performed with 1000 iterations and a threshold of 1.0. Additionally, more iterations are also tried, but it does not show significant improvement. In each iteration of RANSAC, 4 pairs of points are randomly chosen from the filtered coordinates pairs. The matrix A is constructed according to the formula in the lecture with the 4 pairs of points. Then, matrix A is solved, and the singular vector with the smallest singular value is selected. The singular vector is then reshaped into 3x3 to construct the candidate Homography matrix H. After that, the matrix H is used to transform the key points into new coordinates. Then, the l2-norm is computed between the transformed coordinates with the original matched coordinates. If the l2-norm is less than the threshold, this pair is an inlier. The entire process is repeated with the number of iteration times, and the matrix H that has the maximum number of inlier is returned.

**B: For the image pair provided, report the number of homography inliers and the average residual for the inliers. Also, display the locations of inlier matches in both images.**

**C: Display the final result of your stitching.**



**Part 2: Shape from shading**

**A: Estimate the albedo and surface normals**

1. Insert the albedo image of your test image here:
2. What implementation choices did you make? How did it affect the quality and speed of your solution?
3. What are some artifacts and/or limitations of your implementation, and what are possible reasons for them?
4. Display the surface normal estimation images below:

**B: Compute Height Map**

1. For every subject, display the surface height map by integration. Select one subject, list height map images computed using different integration method and from different views; for other subjects, only from different views, using the method that you think performs best. When inserting results images into your report, you should resize/compress them appropriately to keep the file size manageable -- but make sure that the correctness and quality of your output can be clearly and easily judged.
2. Which integration method produces the best result and why?
3. Compare the average execution time (only on your selected subject, “average” here means you should repeat the execution for several times to reduce random error) with each integration method, and analyze the cause of what you’ve observed:

|  |  |
| --- | --- |
| Integration method | Execution time |
| random |  |
| average |  |
| row |  |
| column |  |

**C: Violation of the assumptions**

1. Discuss how the Yale Face data violate the assumptions of the shape-from-shading method covered in the slides.
2. Choose one subject and attempt to select a subset of all viewpoints that better match the assumptions of the method. Show your results for that subset.
3. Discuss whether you were able to get any improvement over a reconstruction computed from all the viewpoints.

**Part 3: Extra Credit**

Post any extra credit for parts 1 or 2 here. Don’t forget to include references, an explanation, and outputs to receive credit. Refer to the assignment for suggested outputs.

**Reference**

https://scikit-image.org/docs/stable/auto\_examples/registration/plot\_stitching.html#sphx-glr-auto-examples-registration-plot-stitching-py