Assignment 2: Panoramic Stitching

If the camera center is fixed, or if the scene is planar, different images of the same scene are related by a homography. In this project, you will implement an algorithm to calculate the homography between two images. Several images of each scene will be provided. Your program should generate image mosaics according to these estimated homographies.

(100%)

The basic algorithm:

To estimate the homography between two images, we need to first identify corresponding points in these images. You can use a feature detector to identify feature points in both images, and find corresponding points by comparing their descriptors. Popular detectors include Harris, SIFT or Surf. For feature descriptors, we might use either the SIFT descriptor or the concatenated pixel values within a local window. (You are allowed to use any existing library for feature detector and descriptor.)

Note that if you are using DLT as the solution, you should normalize the pixel coordinates of these feature points, such that their pixel position is between -1 to 1 (you can use the statistics of the pixels positions or use the image dimension to perform the normalization). (Justifications of this normalization were discussed in the class.) We can then apply the RANSAC and DLT algorithms to obtain the largest set of inliers. Basically, the program will randomly select 4 corresponding pairs at a time to use the DLT algorithm to compute a homography. Then the other points are classified as inliers or outliers according to how good they fit to this homography. We need to decide the number of iterations according to our estimated ratio of outliers. Among all the results during random sampling, choose the largest set of inliers. We can then apply the DLT algorithm again to all these inliers.

Once the homography is estimated, we can use it to transfer images to align with a reference image to create mosaics. We can simply use linear blending to decide pixel values at overlapped regions. In other words, if a pixel is covered by two images, we can retrieve its values from both images and set it to the average value.

You are required to submit your source code through Carman system.

Please also submit a **pdf** or **doc** format report. **Your report cannot be longer than 4 pages.** Your report should show the input and output side by side for easy reference. Please zip everything together and submit only ONE zip file in the Carman system with your name as the file name [e.g. Firstname_Lastname.zip].

In the report, you are expected to discuss your findings through the experiment. For example, what brings troubles to homography fitting? What kind of data works best/worst? What do you think is the highlighted skills/unique tricks in your implementation? How the implemented algorithm can be improved?

Bonus [20%]

Comparison of different feature descriptors:

Different feature descriptors have different performance. They would generate matches with different percentage of outliers. This ratio will affect the number of iterations we need for the

RANSAC algorithm [discussed in the class how to determine the number of iterations]. We can use the smallest number of RANSAC iterations that ensures a correct result to compare these feature detectors/descriptors. Please compare the SIFT descriptor vs. descriptor formed by concatenated pixel values. To decide the number of iterations needed, we can choose a smallest number that ensures the RANSAC algorithm succeed in 50% percent chance of many independent tests, say 100 tests.