VINUNIVERSITY

Lab Assignment 03

COMP3040 Computer Vision **November 11, 2024**

Due: November 11, 2024, 5:05pm

Objective: The goal of this assignment is provide students with hands-on experience in working with RANSAC and Hough Transform

Evaluation Criteria:

- Correctness of the code.
- Proper comments and documentation.

Submission Instructions:

- Download **Lab03_code.zip** to your computer
- Write your Python code in either the **lab03.py or lab03.ipynb**
- Zip the folder and submit on Canvas

Note: Please install the following Python packages:

- numpy
- opencv-python (cv2)

1 Problem 1: Panorama Image Stitching (60 pts)

Information: The panorama stitching algorithm consists of four steps:

- Detect keypoints (DoG, Harris, etc.) and extract local invariant descriptors (SIFT, SURF, etc.) from the two input images.
- Match the descriptors between the two images.
- Use the RANSAC algorithm to estimate a homography matrix using our matched feature vectors.
- Apply a warping transformation using the homography matrix obtained from Step 3.

1.1 Stitching pairs of images

1. **Load both images and convert them to grayscale.** Detect keypoints and their descriptors in both images using the Scale-Invariant Feature Transform (SIFT) algorithm. Display both images in RGB color along with the keypoints.
2. **Match Keypoints.** From the detected keypoints, use the Brute-Force matcher in OpenCV to find the best matches between the two images. Filter the matches using the ratio test. Display the matches by drawing lines between the matching keypoints in both images.
3. **Estimate Homography.** Run RANSAC (Random Sample Consensus) to estimate a homography that maps one image onto the other. For the best-fit homography, print the number of inliers and outliers.
4. **Create the Panorama.** Create a new image with sufficient size to hold the panorama. Use `cv2.wrapPerspective` to composite the two images into the panorama.

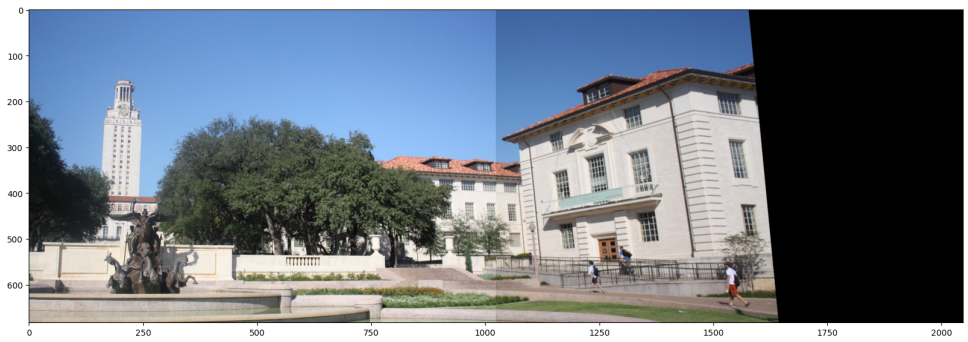

Figure 1: Input and Output of Problem 1.1.4

1.2 Stitching multiple images

Now that you have code for pairwise stitching working, the next step is to extend it to work on multiple images. You will be working with the provided data, consisting of three sequences of

three images each. For the "problem12" sequence, sample output can look as follows (although yours may be different if you choose a different order of transformations):

Figure 2: Output of Problem 1.2

For this part, you should experiment with and discover your own strategy for multi-image stitching. For full credit, the order of merging the images should be determined automatically. The basic idea is to first run RANSAC between every pair of images to determine the number of inliers to each transformation, use this information to determine which pair of images should be merged first, merge this pair, and proceed recursively.

2 Problem 2: Find Intersections with Hough Transformation (40 pts

1. Apply Hough Trasformation to detect all the lines appear in the input images.
2. Once the lines are detected, the next task is to find and highlight the intersections between any two lines. (Hint: As the lines detected using the Hough Transform might have small variations (due to noise or line thickness), it is crucial to filter out nearly identical lines before computing intersections). OpenCV is not allowed in this part.

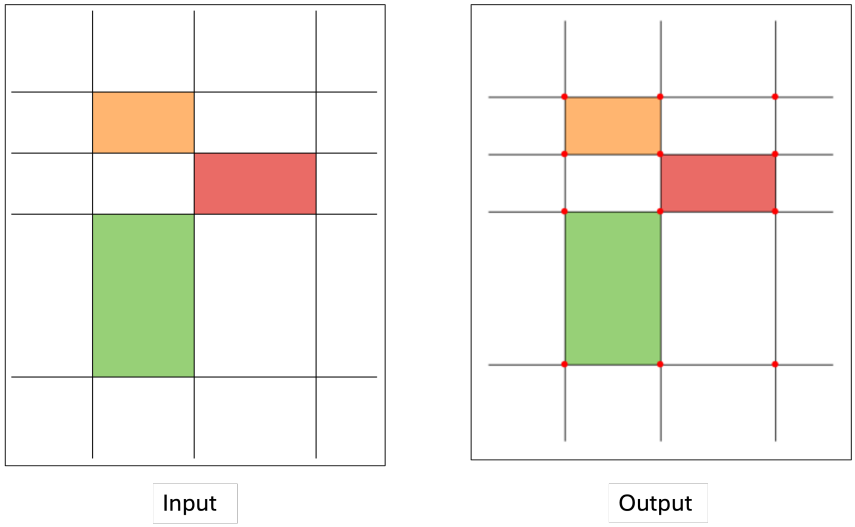

Figure 3: Input and Output of Problem 2.2