Project 2
CSE 473/573 (Summer 2019)
Due Date: July 16th, (During class hours)

1. Image Stitching [100 points]

The goal of this task is to experiment with image stitching methods. Given a set of photos, your program should be able to stitch them into a panoramic. There are no restrictions regarding the method you use to stitch photos into a panoramic photo. Please keep in mind that the best solution may require transformation of some of the images in 3D, not just a simple overlap and blending.

For this project, you can assume you will have at most 3 images that you need to stitch together and that the overlap of any two will be at least 20%. You will have to determine the spatial arrangement of the images automatically. While some of the most modern techniques may use a spherical projection of better panoramas, you are free to assume that basic 2D Planer transformations are sufficient.

To get the points, you can to take 3 photos that demonstrate something of value to you about UB life. This could be at an event, something on campus, a road trip, etc, but have something UB related in the scene. You must then take your program and stitch them into a panoramic photo. The overlapped region between any 2 of the photos you take should not exceed 50%.

1.2. Requirements

Your submission should meet the following requirements:

- The code and photos are stored in a single file whose extension is “zip”.
- The code you provide should be stored in a folder named “src”.
- The photos you take for the UB are named “ub1.jpg”, “ub2.jpg” and “ub3.jpg” and should be stored in a folder named “ubdata”.
- The program should read ALL jpg files in the data directory and stitch them together.
- The panoramic photo your program creates should be saved to the same directory as the data and be called “panorama.jpg”.

Finally, a report should be prepared where you need to provide detailed implementation plan, the output image and the reason for your approach.

1.3. Python Library

- Any Python Standard Library could be used.
- Any API provided by Numpy or Scipy could be used.
- Any API provided by OpenCV could be used, except “cv2.findHomography()” and APIs that have “stitch”, “Stitch”, “match” or “Match” in their names, e.g., “cv2.BFMatcher()” and “cv2.Stitcher.create()”.
- If you decide to use SIFT, please note it has been patented and it has been removed from OpenCV3, but it is included in OpenCV2.