

Homework Assignment # 1

DUE: 5:00pm, Sunday Feb 17th (Electronic turnin required)



You are to implement a panorama stitching program in this assignment. The input images will be taken by a single camera with no change of any intrinsic camera parameters. The camera trajectory is always a rotation about the person holding the camera, either in a simple panning motion (left to right or right to left) or a “saw-tooth” pattern (panning + up and down motion). This implies that you should always match and stitch adjacent images in an alphabetic or numerical order.

Sample test images can be found in <http://www.cs.ucsb.edu/~cs281b/testimages/prog1/> (or follow the local image archive link from the class web page). There are two types of sequences: those depicting a complete, 360° rotation and those depicting less than a 360° rotation. Make sure that your program works on images in that directory, at least for those depicting less than 360° motion. You are welcome to use your own images for testing.

Your program should take a single argument, the name of the directory that holds the discrete snapshots. Your program should then stitch all the snapshots in the directory together and output the panorama image. You can use any programming language you feel comfortable with.

BONUS: If you implement some elaborate features, e.g., ability to handle 360° panorama and random input sequences, color blending and smoothing in stitching, etc., you should turn in your program with your own images (if necessary) to demonstrate the advanced capabilities. Please provide a README file to explain how to compile and run your program and what I should be looking out for.

NOTE: You are allowed to use packages like OpenCV for your assignment, but your own work must be more than 10-line codes that just call a stitching function from OpenCV. I do not mind (in fact, it is encouraged), that you rely on existing packages for sophisticated processing, e.g., SIFT, SVD, etc. but you should at least attempt to implement some basic routines, e.g., solving homography equations with RANSAC, executing forward (or backward) pixel mapping and splatting, and performing color normalization and nonlinear optimization (if you implement those functions). If in doubt, please check with me.

Also, it is highly encouraged that you try your programs out on your own images. It is thus strongly recommended that you include with your programs your own image sets and stitched results.