CS 346 Project 2: Homography matrix computation, Image warping and Mosaic (10 points)
Due April 24 2014, 9:30AM, submitted via Blackboard

This project will include homography matrix computation, image warping and image mosaic.

(0) Use your camera to take a few images in the campus for this project. Keep in mind that a good image pair should contain a predominantly planar scene structure, or else be taken from a camera that is only rotating (e.g., panning), with very little translation.

(1) Read in two images. Manually select corresponding points from the two images (codes provided). Compute the homography matrix using the Pseudo-inverse method (h33=1) and Singular Value Decomposition method ($\|h\| = 1$). Compare your resultant homography matrices using the two methods. Try different images and different sets of corresponding points on two images, summarize your observation and understanding of homography matrix computation.

(2) Based on the computed homography matrix, use forward and backward warping to warp one image to the coordinate of another image. Try the "for-loop" method we discussed in the class, and try different interpolation methods (nearest and bilinear) in the backward warping. Then, try Matlab's interp2() function. Summarize your observations.

(3) Based on the homography matrix and your warping method, create a mosaic image to combine two images. The following is an example.

Upload running code and a written report to Blackboard by the due date/time. Half of your grade will be based on the written report discussing your program, design decisions, and experimental observations, and half on the program itself and output it produces. The report should contain:

a) Brief summary of what you think the project was about (what was the task; what were you trying to achieve),

b) Brief outline of the algorithmic approach (e.g., enumerate your step or a flowchart showing the flow of control and subroutine structure of your code),

c) Pictures of intermediate and final results that convince me that the program does what you think it does.

d) Any design decisions you had to make, for example whether using preconditioning on the homegraphy matrix estimation or not. Be sure to document any additional features you added to increase robustness or generality of your codes.

e) Experimental observations. What do you observe about the behavior of your program when you run it? Does it seem to work the way you think it should? Play around a little with different setting to see what happens.

f) If you work in a team, a description of what each group member contributed to the final program or report, to discourage slackerism!!!

Bonus points (up to 5 points)

You can choose any of the following additional tasks to earn partial or full bonus points.

(1) (2 points) Capture three or more images in the MST campus and create a large mosaic image on these images using your codes.

(2) (2 points) Use your own creativity and imagination to create some cool image warp and mosaic applications.

(3) (5 points) Automatically find correspondence points for the Homography matrix computation (Lecture 14):

(a) Detect corners in source and destination images;

(b) Extract intensity patches around corners in source and destination images;

(c) Use RANSAC algorithm to find the inlier set of correspondence points by matching patches. Note, NCC is used as the metric to compare patches, two directional matching is used to determine if two patches are matched, a gating region can be used to reduce the search range if the displacement between the two images is known to be within the gating region.

(d) Use the inlier set of correspondence point s to compute the Homography matrix. Compare the results with those by manually selected correspondence points. Finish your image warping and mosaic.