CS 436 / CS5310 Computer Vision Spring 2016-17

Assignment 2: 2D Affine Warping

Wednesday, Feb. 22, 2017

Due *electronically on LMS* at 11:55PM on Mond**ay, Mar 6th, 2017**

Grade Scale: CS 5310 – 120 points + 20 bonus // CS 436 – 80 points + 40 bonus

You may not use any functions of the image processing toolbox in this implementation, except imread and imwrite to read/write images, or the ones that are explicitly mentioned below. If in doubt, please discuss.

1. [60 points] Write a MATLAB function which will take as input the name of a gray scale image and eight projective parameters a_1 to a_8. The output of the program should be another image file which is of the same size as the input file, but has the original image warped according to the eight parameters. You must use bilinear interpolation as discussed in class. Any portion of the output image which falls outside the bounds of the input image may be cropped.

   Your function must run from the command line as `>> output = imwarp(im, H)`, where `im` is a gray scale input image, and `H` is a 3x3 projective transformation matrix. You may have additional optional inputs after these two parameters if you want to implement more features.

   a. Using your function, generate several different types of image transformations.

   b. Figure out a transformation through trial and error such that transformed ‘mecca06.pgm’ looks identical (or as close as you can get) to ‘mecca06t.pgm’.

   c. Extend the warping function (implemented in part 1) for color images. Try on ‘mecca06.ppm’ and report your results.

2. [20 points] Modify the above program so that the complete transformed image is always visible, i.e. there is no cropping of the transformed image at the corners.

[Additional task: Required for CS5310 students, extra credit of up to 20 points for CS436 students]

3. [40 points] Recovering the projective transformation from control points using least squares (or pseudo inverse) approach:

   Here the input to the program will be two images (e.g. ‘mecca06.pgm’ and ‘mecca06t.pgm’) and a set of (at least three) control points (also called correspondences) between these images. The correspondences may be specified through a graphical interface by clicking (you can use the `getpts` function), or may simply be read from a text file. The projective transformation between ‘mecca06.pgm’ and ‘mecca06t.pgm’ needs to be recovered first. After that, mecca06.pgm can be transformed by the recovered projective parameters. Then, the transformed image should look identical to mecca06t.pgm. Display transformed image and mecca06t.pgm together to highlight differences if any. Subtracting the two, taking the square and summing over the whole image can compute the error between them. Comment on the error image.

[Bonus question: Not required, but worth 20 additional points]

4. Implement the `imwarp` function from Question 1 without using any FOR loops. *Hint:* Use the `meshgrid` function to generate x and y coordinate matrices and then try to apply the transformation in one go, rather than separately for every pixel. Since MATLAB is an interpreter, this will work much faster.