Homework 3

Due Monday, February 18, 2013 (by 8:00 pm)

Notes: Please email me your solutions for these problems (in order) as a single Word or PDF document. If you do a problem on paper by hand, please scan it in and paste it into the document (although I would prefer it typed!).

1. (15 pts) On the course website are two images of a street intersection taken from a stationary camera, taken 5 seconds apart. Transform these images to "orthophotos"; *i.e.*, images taken from a viewpoint directly overhead, such that the scale is uniform. Here are some control points that have been measured in the scene:

Image (x,y)	Actual (x,y) in feet
154, 389	0, 0
453, 287	53.5, -1
263, 214	60.5, -54.5
20, 252	5, -56
253, 464	0, 17.7
316, 436	8, 17.7
227, 312	19.9, -14.6

Choose a scale of one pixel = 0.25 feet in the output image. How fast (miles per hour) is the person in the lower left walking (it's ok to measure the location of the person in the orthophotos by hand)?

2. (15 pts) The auto-correlation score is $E_{AC}(\Delta\mathbf{u}) = \sum_i w(\mathbf{x}_i)\left[\nabla I_0(\mathbf{x}_i)\cdot\Delta\mathbf{u}\right]^2$ (equation 4.5 in the textbook). Show that this equals $\Delta\mathbf{u}^T\mathbf{A}\Delta\mathbf{u}$, where matrix A is as given in equation 4.8.

3. (20 pts) Take the Matlab corner detector program developed in class and make the following changes:
   a. Instead of using a square region of size NxN to sum the gradient products, weight the values, using a Gaussian mask for w(x,y).
   b. Instead of using the interest point measure det(A)/trace(A), use the minimum eigenvalue of A[1]. This is the "Shi-Tomasi" approach.
   c. Instead of taking all interest points above a minimum threshold, take the 10 points with the highest scores.

Apply this program to find the top 10 corner points in the image "test000.jpg". Draw a rectangle around each of these points on the original image and label them.

[1] Do not use "for" loops to go through the image and call Matlab's "eig" function at every point. Instead, use the equation for the eigenvalue that you calculated in HW1, problem 3c. You should be able to do this without any "for" loops.

4. (15 pts) Run the OpenCV version of the Shi-Tomasi corner detector, which is implemented in the function "`goodFeaturesToTrack`". The use of the Shi-Tomasi corner detector is illustrated in a tutorial on the http://docs.opencv.org/ website.
   a. Apply the program to the image "test000.jpg" and find the top 10 corners (note - you may not get exactly the same corners as your Matlab program finds in the previous problem). Give the code you used and the resulting image.
   b. Apply the program to the image "cube1.jpg". See if you can find all the corners on the checkerboard pattern (you will have change the program to increase the maximum allowable number corners to find).

5. (15 pts) Consider the 3x3 template *w* as shown.

w

-1	0	-1
0	4	0
-1	0	-1

   a. Compute (by hand) the normalized cross correlation score of template *w* with image *f*, at the center position of *f*. You might want to check your answer using Matlab's normxcorr2 function.

f

1	0	1	0	1
1	2	1	1	1
0	2	6	0	0
1	1	3	2	1
1	0	1	0	1

   b. Give a different image *f* such that the normalized cross correlation score of template *w* with image *f*, at the center position of *f*, yields a score of -1. Assume that the image is the type unsigned 8-bit integer (ie, its values lie between 0 and 255).

6. (20 pts) Using normalized cross correlation, match the top 10 points from the corner detector program of problem 3, from image "test000.jpg" to their best matches in image "test012.jpg". You can use Matlab's "normxcorr2" function. Mark the best matches in the second image, and label them with their identifying index from the first image (i.e., "1", "2", etc).