CS 485 - Homework 4, due April 12th, *Jana Košecká*

The code for this homework is available at http://cs.gmu.edu/∼kosecka/cs485/hw4/.

1. (2) Read the image of the house, convert the image to gray level and resize it as:

   ```
   im = imread('house1.jpg')
   imgray = rgb2gray(im);
   imsmall = imresize(imgray,0.25);
   figure; imagesc(imsmall); colormap gray; axis image;
   ```

   Use this image to test the `conv2` routine, which convolves the image with a 2D filter

   ```
   filter = ones(3,3);
   imresult1 = conv2( imsmall, filter, 'same');
   figure; imagesc(imresult1); colormap gray; axis image;
   ```

   Try the result with filters composed of 1's and vary the size of the filter (e.g. , 7×7, 9×9 or 11×11). Comment on the effect of changing the size of the filter.

2. (3) **Edge detection** Test the `edge` function in MATLAB and hand in images of edge maps under two different choices of threshold and report what the thresholds are. Use the 'canny' option to get the Canny edge detector. Do `help edge` to see how to use the function. Test the function `bwlabel` for computing the connected components of the edges obtained as an output of the edge detector. This is often useful intermediate representation used in variety of problems. Hand in the results of using these functions for at least two threshold settings of the edge function.

3. (5) **Corner detection** In this problem you will use Matlab implementation of the Harris corner detector which you can download from http://cs.gmu.edu/∼kosecka/cs485/hw4/harris corners.m. This includes comments on the input parameters and how to run the detector. Your task is to run this corner detector on sample images provided in same .../hw3/ directory, visualize the results and answer some questions below. Here is a set of Matlab commands demonstrating its use.

   ```
   >>  im = rgb2gray(imread('house1.jpg'));
   >>  corners = harris_corner(im, 7,1.5);
   >>  imshow(img); hold  on;
   >>  plot(corners(:,1), corners(:,2), 'ro');
   ```

   This reads the image from file house1.jpg, runs the corner detector using a gaussian kernel of width 7 pixels and standard deviation 1.5. The last three commands display the image and the detected corners superimposed on it.

   (a) For fixed parameter values, run the detector on house1.jpg and house1-rotated.jpg. The latter image is a rotated copy of the former. If we rotate the input image, do the detected corner position s rotate by the same amount ? Justify your answer based on your observations.

(b) If we scale down the input image, are all the detected corner positions scaled accordingly ? Yo u can test this experimentally by comparing the corner detection on the images house1.jpg, house1-2down.jpg, house1-4down.jpg. Each image in this sequence is half the size of its predecessor. Justify your answer based on your observations.

4. **Correspondences.**

   (a) Use the Harris feature corner code from the previous exercise, select the features in the first image and find the corresponding points in the second image. Carry out the experiments on the stereo pair of images. Implement the search for correspondences using SSD sum-of-squared-differences similarity measure (i.e. for each detected point returned by Harris corner detection find the closest point in SSD sense in image 2). Carry out experiments on the pair house1.jpg and house2.jpg and the pair house1.jpg and house1-rotated.jpg and house1-4down.jpg and house2.jpg.

   (b) Repeat the same exercise for SIFT detector and descriptor which you can download from www.vlfeat.org and hand again the two sets of correspondences for two sets of image pairs.

   Submit the SSD correspondences code and results of correspondences/matching algorithms and comment on the difference between the two strategies, which one is better for which pair.

   To visualize the result make a new image putting the two images side-by-side and connect the corresponding features by plotting lines originating in one view and finishing in another. This is done easiest in Matlab in 3 lines of code. Function `appendimages.m` to make a composite image is available on the web site.

5. (5) **Bayes Rule** Suppose you are a witness to a nighttime hit-and-run accident involving a taxi in Athens. All taxi cars in Athens are blue or green. You swear, under oath, that the taxi was blue. Extensive testing shows that, under the dim lighting conditions, discrimination between blue and green is 75% reliable. Is it possible to calculate the most likely color for the taxi? (Hint: distinguish carefully between the proposition that the taxi is blue and the proposition that the taxi appears blue.) What is your resulting estimate, given that 9 out of 10 Athenian taxis are green?