Intro to Image Understanding (CSC420)
Assignment 1

Submission Deadline : September 27, 11.59pm, 2014

Max points: 10, max extra credit points: 5

1. [**1 point**] Write pseudo-code for computing convolution of the 2D image and a 2D filter.
2. (a) [**1 point**] Given a $n \times n$ image, I, and $m \times m$ filter, h, what is the computational cost of computing $h * I$ (the convolution)? What is the computational cost if h is a separable filter?
   (b) [**1 point**] If I first convolve an image with a Gaussian filter with $\sigma = 3$, and then convolve the output with a Gaussian with $\sigma = 4$, this gives an equivalent result as if I just convolve the image with a Gaussian with what σ?
   (c) [**1 point**] Convolve the attached lena.png image with a (2D) Gaussian filter with $\sigma = 1$ and visualize the result (display the result of the convolution). Time the convolution function in this case. You can use the Matlab's functions for convolution/correlation, and functions TIC and TOC for timing.
   (d) [**1 point**] Gaussian filter is separable. How can you use this fact to speed up convolution? Time the convolution function in the separable case, and compare to the time you got in the previous step. How would you make sure that the convolution output is the same as in the previous case?
3. (a) [**1 point**] Compute magnitude of gradients for the attached images waldo.png and template.png.
   (b) [**1 point**] Write a function that localizes the template (template.png) in the image waldo.png based on the magnitude of gradients. You can help yourself with the function available under Lecture 2 on class webpage (be careful, that function uses intensity values).
   (c) [**2 points**] Compute the image pyramid with 3 levels for the attached image waldo.png.
4. [**1 point**] Name two specific ways in which one could reduce the amount of fine, detailed edges that are detected with the Canny edge detector.
5. Exercise for **Extra Credit** [**5 points**] (this is an optional exercise): Implement seam carving:
   (a) Compute magnitude of gradients of an image
   (b) Find the connected path of pixels that has the smallest sum of gradients. A path is **valid** if it is connected (the neighboring points in the path are also neighboring pixels in the image), it starts in the first row of the image and in each step continues one row down. It finishes in the last row of the image.
   (c) Remove the pixels in the path from the image. This gives you a new image with one column less.
   (d) Remove a few paths with the lowest sum of gradients. Create examples with a few images.