CSci 4968 and 6270
Computational Vision,
Fall Semester, 2009-2010
Homework 1
Due: Monday, September 14, 2009, 5 pm

Homework Guidelines

Your homework submissions will be graded on the following criteria:

- Correctness of your solution,
- Quality of your output,
- Conciseness and clarity of your explanations,
- Clarity of your code, and
- Where appropriate, computational efficiency of your algorithms and your implementations.

Clarity of code includes the usual properties of good code: clear and easy-to-follow logic; concise, meaningful comments; good use of indentation, spacing, varable naming, and blank lines (don't insert a blank line after each statement!) to make the functions easy to read.

Please submit your solutions via email to `cvstewart@gmail.com`. Submit a *pdf file* containing your write-ups and image results, Matlab diaries as necessary, and your m-files as well. Do NOT submit Word files or Powerpoint files — if you use these tools, save the results to pdf. Combine everything into a zip file; if you want you can create subfolders for each problem. Include the phrase "HW 1 submission" on the subject line, so that I can automatically sort your email.

Problems

This is a "warm-up" assignment to get started working with images and with Matlab. Homeworks 2-5 will be harder and worth more points. Homeworks 6 and 7 will be worth fewer, since they will be assigned while you are working on your final projects.

1. (**10 points**) The Matlab image processing toolbox has a histogram equalization function called `histeq` (as well as a histogram function called `imhist`). Using this function, or the equivalent in a C++ toolkit, find an example image for which histogram equalization produces a very nice output image, and find an example image for which histogram equalization does a very poor job. These images must not be artificial. In fact, if you have access to a digital camera, you should take them yourself. (Remember to convert them to grayscale using `rgb2gray` before doing so.)

Submit the images before and after histogram equalization, as well the histograms of each image (before equalization). Include a concise explanation of why histogram equalization works well in the one case and poorly in the other.

2. (**20 points**) Write a Matlab function that takes two arguments as input — a grayscale image f and a floating point number in the interval $[0, 0.5)$ — and produces a grayscale image f' as a result. The function should compute the cdf of the image, and then find g_0 and g_1 such that

   - g_0 is the largest grayscale g such that $\mathrm{cdf}(g) \leq r$, and
   - g_1 is the smallest grayscale g such that $\mathrm{cdf}(g) \geq 1 - r$.

   The result image should be

$$f'(i,j) = \begin{cases} 0 & f(i,j) \leq g_0 \\ 255 & f(i,j) \geq g_1 \\ \frac{255(f(i,j)-g_0)}{g_1-g_0} & \text{otherwise.} \end{cases}$$

   Submit the source code, diary from running Matlab, and example results, including histograms and images.

3. (**20 points**) Write a function that computes the result of convolving an image with a Gaussian kernel of standard deviation σ. It should take the image and the value s (representing σ). It should form two one-dimensional kernels, each of size $2w + 1$ where $w = \lfloor 3\sigma \rfloor$. One kernel should be vertical and the other horizontal. The function should use function `imfilter` to do the actual work. As an intermediate result, the function should output the contents of the two kernels. Remember to convert the image to single or double precision before running through the filter.

   Submit the source code, diary from running Matlab, and example results.