CSC 589 Introduction to Computer Vision

Problem set 1 Basic Image Processing (total points: 80pts)

Due date: Feb 2nd, Monday, end of the day (late policy applies).

Instructions

Homework 1 is a warm-up homework containing several simple image-processing exercises. You must complete this homework set individually. Teamwork is not allowed for this project. You can discuss with your classmates and me, posting questions on Piazza (if you are stuck at a syntax problem), but you must comment on top of your code with whom you have discussed. Please do not copy codes from the Internet. It is easy to spot and will result in little grades. But you are welcome to finish this with either MATLAB or Python. A homework template .py file is included.

In this class, besides technical knowledge, we are also practicing writing clear and well-commented code, and also clear and easy to understand research report. Please take the writing up part of the project seriously. We are hoping toward the end of the semester, some of the project could lead to Python image tutorial that can be shared on the Internet and or independent research project.

Hand-in instructions:

Save your homework as Firstname_lastname_ps1.py. Please zip your code and your test images into a folder and upgrade to Blackboard. Please include all the test images you tend to run. The folder name MUST has the following structure:

firstname_lastname_ps1

You must make sure after I download your folder, I can automatically run your code (without fussing around with image paths, etc).

You are also expected to turn in a write up document as pdf (you can use word or Latex for equations). For each problem, briefly explain what you your algorithm you have used and the answers to some of the questions asked in the problem. This report should also include the image results you generated. Please include your test images, your output images. Try to use subplot to plot multiple figures in one panel.

Grading:

Final grades will be based on 70% correctness, 10% clarity and 20% write-up.
If your code can't compile, meaning it stopped running and report an error. You will get a 50% reduction automatically.

Homework that is failed to follow hand-in instructions will get 10% reduction in grades. Please notice the late submission policy in the syllabus.

Problem 1 (20pts). Warm up. Download the image folder for PS1 and choose two images and do the following: for each of these images

1.1 Load the image into your environment

1.2 Blur the image using Gaussian filter.

1.3 Display the result.

1.4 Compute the DFT (Discrete Fourier Transform) of the image. Please read Numpy FFT package. np.fft.fft2 and see here (http://docs.scipy.org/doc/numpy/reference/routines.fft.html)

1.5 Display the magnitude of the DFT .

Problem 2 (20 pts). Histogram equalization. Compute the gray level (luminance) histogram for an image and equalize it so that the tones look better (and the image is less sensitive to exposure settings). You might want to use the following steps:

2.1 Convert the color image to luminance.

2.2 Compute the histogram, the cumulative distribution, and the compensation transfer function (normalized CDF)

2.3 Try to increase the "punch" in the image by ensuring that a certain fraction of pixels (say 5%) are mapped to pure black and white.

2.4 Limit the local gain f'(I) in the transfer function. One way to do this is to limit

$$f(I) < \gamma I \; or \; f'^{(I)} < \gamma$$

while performing the accumulation, keeping any unaccumulated values "in reserve".

2.5 Compensate the luminance channel through the lookup table and regenerate the color image using color ratios (Chapter 2.116 in Szeliski)

Problem 3 (20pts). Padding for neighborhood operations. **Write down** the formulas for computing padded pixel values as a function of the original pixel values and the image width and height (M,N) for each of the padding modes shown in Figure 3.13 (Szeliski). For example, for replication (clamping), which repeat edge pixels indefinitely,

$$\check{f}(i,j) = f(k,l), \begin{matrix} k = \max(0, \min(M-1, i)), \\ l = \max(0, \min(N-1, j)), \end{matrix}$$

(Hint: you may want to use the min, max, mod, and absolute value operators in addition to the regular arithmetic operators)

Problem 4 (20pts) Separable filters. Implement convolution with a separable kernel (Figure 3.14, Page 102). The input should be a grayscale or color image along with the horizontal and vertical kernels. Please include examples of Gaussian filter, box filter, and Sobel filter. Make sure you support the padding mechanisms developed in Problem 3. You will need this functionality for some of the later exercise. Please specify the kernels you used and display the original image and the output images after each horizontal and vertical operation.