CSE527 Homework 1

Due Thurs Sept 25, 2014

In this homework you will work on basic image processing using Matlab. Be sure your Matlab has Image Processing Toolbox. (Graduate Lab Machines have it) Download the image files (hw1_images.tgz) from Blackboard into your directory and then load image files into Matlab with the imread command (make sure to set your Matlab's current directory to the directory containing the file). There are seven images in total. The lena-noise.bmp is the same as the lena.bmp but corrupted with Gaussian noise. The barbara_noise.bmp is similarly corrupted barbara.bmp, and mandrill_noise.bmp is corrupted mandrill.bmp.

If you would like to install Matlab on your personal machine, you can get a license for free via Softweb. See this page for more information and instructions.

For the basic usage of Matlab, see: Matlab Tutorial (Univ. Utah), Matlab User's Guide (2MB PDF), Matlab Primer (200K postscript; 25 pages). You can also look at the product help contents from the Matlab's help menu.

TIP: In Matlab, you can type "help *functionname*?to display usage for that function. ?(Example: >> help imread)

Problems

1. {15 pts} Write a function in Matlab that takes two arguments, a width parameter, w, and a variance parameter, s, and returns a 2D array containing a Gaussian kernel of the desired dimension and variance. The peak of the Gaussian should be in the center of the array. Make sure to normalize the kernel such that the sum of all the elements in the array is 1. Use this function and the Matlab's conv2 routine to convolve the lena and lena_noise arrays with a 5 by5 Gaussian kernel with sigma of 1. Repeat with a 11 by 11 Gaussian kernel with a sigma of 3. Comment on the difference between the two images. Try the same thing to barbara and barbara_noise, and to mandrill and mandrill_noise. Turn in the resulting images produced from the noisy data only, along with your Matlab code.

2. {15 pts} The Gaussian kernel is separable, which means that convolution with a 2D Gaussian can be accomplished by convolving the image with 2 1D Gaussian, one in the x direction and another in the y direction. Repeat the 2 convolutions you did in question 1 using this scheme. You can still use conv2 to convolve the images with each of the 1D kernels. Verify that you get the same results that you did in question 1 by computing the maximum difference between the arrays produced with the two methods. Turn in your Matlab code. (You don't have to turn in the images for this part)

3. {15 pts} Implement in Matlab the Sobel edge detector and apply it to the input image building.bmp. Then apply an appropriate threshold to the edge strength values to select what you think are the most important edges. Turn in the Matlab code and the resulting image.

4. {5 pts} Convolve an 11 by 11 Gaussian of sigma = 1 with the discrete approximation to the Laplacian kernel (i.e., [0 1 0; 1 -4 1; 0 1 0] or [1 1 1; 1 -8 1; 1 1 1]). Plot this 2D function using mesh function. Turn in this 3D plot. Do you see why this is referred to as the Mexican hat filter?

5. {15 pts} Implement in Matlab an edge detector based on locating the zero crossings of the Laplacian. More specifically, write a routine that first convolves the input image with the Mexican hat kernel you produced in question 4 then searches the output array for locations where the value changes from positive

to negative between neighboring pixels and the magnitude of the change is greater than some specified threshold. Test your program on the building image and find a threshold value which seems to yield acceptable results.

6. {35 pts} Implement in Matlab the first two steps of the Canny edge extraction algorithm: CANNY_ENHANCER(Gaussian filtering & finding magnitude and orientation of gradient) and NONMAX_SUPPRESSION. Apply these steps to all of the images then apply an appropriate threshold to the edge strength values to select what you think are the most important edges. Compare the outputs from the original images and the noised ones. Also compare the outputs for images between the zero crossing and Canny detectors. Turn in the results along with the Matlab code.

7. {15 pts extra credit} Implement in Matlab the third step of the Canny algorithm HYSTERESIS_THRESH. Select two appropriate thresholds for the building image. Turn in the results along with the Matlab code.

Submission

Submission will be via Blackboard. You will submit the following files for each question. Please submit a compressed file which includes all of them.

1. Matlab code and resulting images
2. Matlab code
3. Matlab code and a resulting image
4. Matlab code and a 3D plot
5. Matlab code
6. Matlab code and resulting images
7. Matlab code and resulting images

Plus, a report file containing your answer or description for each question. (PDF format preferred) Please clarify which files are for which question (by writing in the report, organizing the folders, etc.) You may include images as figures in the report document instead of independent files.