Rensselaer Polytechnic Institute
Department of Electrical, Computer, and Systems Engineering
ECSE 4540: Introduction to Image Processing, Spring 2017

Homework #3: due Monday, Feb. 12^{th}, at the beginning of class. Show all work for full credit!
Submit your work as a single PDF on Gradescope, labeling each page with a problem number.

For this homework, you'll need the images in the file `http://www.ecse.rpi.edu/~rjradke/4540/hw3.zip`.

1. (25 points.) Consider the image `walk.png`. Load this into Matlab and convert it to a double.

   (a) (5 points.) Use `filter2` to apply the Sobel operator $\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$ to the image. First, visualize the result using `imshow(result,[])` and interpret what you see. Remember, the strongest positive responses will be mapped to white, the strongest negative responses will be mapped to black, and responses close to zero will be mapped to neutral gray.

   (b) (10 points.) Now, create a binary image containing only those pixels whose response is greater than 200 in absolute value. Which pixels are highlighted? Turn in a plot with your homework. There are some vertical edges that aren't picked up at this threshold, like the nearly white-on-white edge of the front woman's coat. How low do you have to make the threshold in order to get a visible response in this area? What happens to the rest of the image when you do so?

   (c) (10 points.) Repeat the experiment for the second Sobel operator $\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$. Again, interpret what you see in various regions of the image and describe what pixels are highlighted if you threshold the pixels greater than 200 in absolute value.

2. (20 points.) Consider the image `fuzzycat.png`. First, view it normally with `imshow(...,[])`. Note that the image is slightly blurry.

   (a) (5 points.) First, filter the image using `ones(5)/25`. How does the image compare with the original? What details are different?

   (b) (15 points.) Using integer values of k ranging from 1 to 5, implement equations (3-64) and (3-65) in the book to apply unsharp masking ($k = 1$) and highboost filtering ($k > 1$) to the image. Explain what you see as k increases. How does the filtered image improve? Note that you should clip the results to [0, 255] before display. Include the $k = 1$ and $k = 5$ pictures in your homework. Discuss specific regions in the image for full credit.

3. (20 points.) Consider the image `boat.png`. This image has grainy noise added to it.

   (a) (8 points.) First, filter the image using the low-pass filter `ones(5)/25`. How does the image compare with the original? How and where is the noise visibly reduced?

   (b) (12 points.) Next, filter the image using a 5×5 median filter. How does the image compare with the original? How does it compare with the image in part (a)? Which image seems perceptually better? Pay particular attention to high-contrast edges, flat areas, and areas of fine detail. A good way to compare and contrast the two images is with a flicker animation using commands like `while 1, figure(1), pause(.5), figure(2), pause(.5), end`.

4. (10 points.) Suppose we filter an image twice: we first apply the box filter `ones(3)/9`, and then filter the resulting image with the 3×3 Laplacian filter (the one with -4 in the middle element).

   (a) (5 points.) Determine (manually) a single 2D filter that will result in the same output using a single pass. Show your work.

   (b) (5 points.) Would your result in (a) be different if we applied the filters in the reverse order? You should be able to answer this without explicitly computing the result; what is the signal processing basis for your answer?

5. (25 points.) Consider the image `board.png`.

   (a) (5 points.) First, filter the image with a 19×19 Gaussian filter with standard deviation 3 (which you can create using `fspecial`). Describe what you see.

   (b) (5 points.) Now create your own function called `gaussv` that returns a 1-D Gaussian low-pass filter as a column vector. The inputs to your function should be `N`, the length of the filter, and `sigma`, the standard deviation of the Gaussian. The filter should be scaled so its elements sum to 1.

   (c) (5 points.) Use your `gaussv` function with arguments $N = 19$ and $\sigma = 3$ and apply it to the input image. Describe what you see compared to part (a) and explain why the filter has this effect.

   (d) (5 points.) Use your `gaussv` function with arguments $N = 31$ and $\sigma = 5$ and apply it to the input image. Describe what you see compared to part (c) and explain why the filter has this effect.

   (e) (5 points.) Use your `gaussv` function with arguments $N = 75$ and $\sigma = 5$ and apply it to the input image. This should look almost exactly the same as part (d); explain why.