Homework 3: Edge Detection

Instructions: Be sure to electronically submit your answers in either ASCII text or PDF format for the written part and as Java source files for the coding part. Please make sure your name appears at the top of the written part. You may discuss the concepts with your classmates, but do the coding and write up the answers entirely on your own.

Java Coding

The goal of this assignment is to implement the Laplacian edge detection method that we discussed in class. You will want to use the separable convolution routines that you already coded in the previous assignment. You will just have to call these routines with different kernels (Gaussian and 1st and 2nd derivatives of Gaussian). If you do use a helper class, be sure to turn in the source code for it as well (resubmit it even if you also turned it in for the last assignment). As in the previous assignment, your filters should all work with floating point images as input. Also, all convolutions should be separable, use wrapping on the boundary, and all filters that use Gaussians should have a $\sigma$ (sigma) parameter that you can set.

- (10 pts) Create a plugin called \texttt{Gradient\_Magnitude} that computes gradient magnitude of an image. This filter should use the DOG filters we discussed in class.
- (20 pts) Create a plugin called \texttt{Laplacian\_} that computes the Laplacian of an image. Use Gaussian 2nd derivative filters.
- (20 pts) Write a plugin filter called \texttt{Zero\_Crossings} that calculates the zero crossings in an image. It should set a pixel to 1 if it is a zero crossing and 0 otherwise. Use the “neighbor-checking” algorithm we covered in class.
- (20 pts) Put it all together! Use the three filters above to do edge detection as discussed in class. Now you should have a threshold parameter to apply a threshold to the gradient magnitude. Follow this by an \texttt{AND} operation with the result of the Laplacian zero-crossing image. Your resulting images should be binary, with 1 on the edges and 0 elsewhere. Call your plugin filter \texttt{Laplacian\_Edge\_Detection}.

Turn in: \texttt{Gradient\_Magnitude.java, Laplacian\_.java, Zero\_Crossings.java, Laplacian\_Edge\_Detection.java}

Written Part

In this part all questions involve applying your filters to the image \texttt{hoos\_cow.tif}.

1. (5 pts) Apply your gradient magnitude filter with $\sigma = 1$. Describe what you see in the background (behind the cow). Now apply your gradient magnitude filter with $\sigma = 8$, and describe the background. Why are they different?
2. (5 pts) Apply your Laplacian filter with $\sigma = 4$. Describe the transition at the edge on the top of the cow’s head (as you move from the background down into the cow’s head). Compare this to the transition at edge on the top of the cow’s ears. Why are the transitions different?

3. (10 pts) Apply your Laplacian edge detection with a fixed threshold $= 10$, but with increasing values for $\sigma = 1, 4, 8, 16$. Describe the problems you see as $\sigma$ gets larger. Now for the same values $\sigma = 1, 4, 8, 16$ try to change the threshold as you change $\sigma$ so that you do not have this problem. How did you have to change the threshold?

4. (10 pts) Imagine that you were writing a computer vision algorithm that had to recognize text on any signs that it found in an image. Find good values for $\sigma$ and the threshold that will outline the edges in the text but will not get many “false” edges in the background. What values did you use? Try this also for the images sign1.tif and sign2.tif. Did you have to change the parameters to get good results? What values did you use?

**Turn in:** A txt or pdf file with written answers to the above questions.
Challenge Problem

(10 pts Extra Credit) In this problem you will try to figure out how to make an “ink drawing” of an image using image processing algorithms that we’ve learned in this class. Here is an example of the effect we are looking for:

Original Image:

“Ink Drawing” Output:
www.eng.utah.edu/~cs4640/homeworks/hw3/obama-sketch.png

The rules are:

- You can only use ImageJ. Any routines from the menus or plugins you’ve written for homeworks are ok.
- You must use one of the filters that you implement in this assignment.
- Turn in the list of steps (and specify any parameters) that you used at the end of your written answers.

A couple of hints: I used a total of four operations, and the last one was just to change the windowing (contrast/brightness) of the result. You will get credit if you get the basic operations that are needed. You might end up with one more or one less operation to get a good effect.