

**CS/ECE 545 Digital Image Processing Homework 2,  
Spring 2016 (Due Fri. March 5th, by 11:59 PM)**

**Note:** Some of the problems below include questions for you to answer. Put your answers in a separate **Word or PDF document** with the file name **answers.(doc or pdf)** and submit it in a zip file that also your code. Use the new Plugin Template Java code from the site as the base for your plugins.  
Following the Instructions at end of document.

1      **Burger & Burge Exercise 5.7 (page 85) (25 points):** Using a histogram specification for adjusting *multiple* images (Section 5.6.4 of text), one could either use one typical image as the reference or compute an average reference histogram from a set of images. Implement the second approach in a plugin called **histogram\_spec\_averages.java** and test it on images on the class site. Discuss its possible advantages (or disadvantages). Be sure to indicate in your documentation which sets of images you tested your implementation on.

2      **Laplacian Filter (19 points):** Write an ImageJ plugin **Laplacian\_Filter\_with\_c.java** to sharpen an image by computing the Laplacian of the image and adding some amount  $c$  of that back

	$c/4$	
$c/4$	$1-c$	$c/4$
	$c/4$	

Your plugin should work on 8-bit grayscale images and accept positive and negative values of  $c$ . Test your program on some sample images from the class site and answer the following questions. Indicate in your documentation, which of the ImageJ samples you tested your program on.

**(2 points)** What value(s) of  $c$  give the best-looking results?

**(2 points)** What happens to the image when the parameter is too high or too low?

**3 Spatial Filtering (32 points):** In this assignment you will create a filter that will either blur or sharpen an image depending on the value of a user-specified variable. Declare this control variable in your program so that it can be modified before running your plugin. Since the functions you will implement have several variables, you should also declare appropriate variables in your code.

**(12 points)** Create a plugin called **Gaussian\_Blur** that implements a separable Gaussian filter. This class should have a settable member variable sigma and a kernel radius equal to three times sigma. Handle boundary pixels by the “extend” method. **Hint:** You may use the **makeGaussKernel1D** function from the book, but be sure to normalize your kernel!

**(12 points)** Create a plugin called **Blur\_Or\_Sharpener** that has a parameter  $w$  between -1 and 1 that performs the following:

$$I' = (1 + w) \cdot I - w \cdot I * G_{\sigma},$$

where  $I * G_{\sigma}$  is the application of your **Gaussian\_Blur** filter to  $I$ .

**Turn in: Gaussian\_Blur.java, Blur\_Or\_Sharpener.java**

**Answer the following questions**

Run your **Blur\_Or\_Sharpener** filter on the example images given. Try out several values for parameters  $\sigma$  and  $w$ .

- i. **(2 points)** What value of  $w$  gives the most blurring? Explain why
- ii. **(2 points)** What value of  $w$  gives the most sharpening? Explain why
- iii. **(2 points)** Run your filter on any image with the maximal amount of blurring (i.e. the  $w$  value from question 3.i and with  $\sigma = 10$ . Now try to reverse this by sharpening the image with the maximal amount of sharpening (the  $w$  from question 3.ii) and the same  $\sigma = 10$ . Did you recover the original image? Why or why not?
- iv. **(2 points)** Run your filter on the image **pattern.png** with  $w = 1$  and  $\sigma = 10$ . Describe the artifacts you see in the filtered image. Why does this happen?

**4 Combination Filtering (24):** In problem 3 of the homework your plugin uses a variable  $w$  to determine whether to sharpen or blur the image. This process could simply be integrated into the algorithm based on some feature of the image, such as an edge. Take your code from #3 and implement a conditional smoothing algorithm which only smooths the image if there is no edge present at the current pixel  $I(u,v)$ . Use the Sobel operator to find the edge strength and compare of a predetermined variable to determine whether to apply smoothing or not. The idea behind this algorithm is that it is to provide smoothing while preserving the edges of the image.

Create a plugin called “**Edge\_preserve\_blur.java**” that implements the above algorithm **(18 points)**. Run it on the **Peppers.jpg**, **Baboon.jpg**, **Cameraman.jpg**.

**Answer the following questions** (support your answer with justification):

- i. **(2 points)** Does this filter achieve the task of edge preserving filter (why or why not)?
- ii. **(2 points)** What can be done to improve the edge preserving properties of this algorithm?
- iii. **(2 points)** Would it help to provide a scale-space feature to the algorithm, like the Canny edge detection?

**Turn in – Edge\_preserve\_blur.java.**

Submit The Following:

1) All ImageJ plugins (documented with your name, date, email, and description – Use template provided on class site):

- histogram\_spec\_averages.java,
- Laplacian\_Filter\_with\_c.java,
- Gaussian\_Blur.java,
- Blur\_Or\_Sharpen.java,
- Edge\_preserve\_blur.java

2) Submit Answers to Questions (filename should be answers.doc or answers.pdf)

3) A separate file called README.txt (plain text file) that contains your name, email, date, a listing the files you are submitting, and any comments on the assignment (such as things to know when grading/running, any issues you had or things you did not finish).

All files need to be placed into ONE zip file named *yourfirstname\_yourlastname\_hw2.zip* using Turnin.

**DON'T EMAIL ME YOUR HOMEWORK. Also, TEST YOUR CODE IN THE ZOOLAB BEFORE SUBMITTING**