CS 370 Computer Imaging

FALL 2014

Programming Assignment - 2
Image Operations & Spatial Filtering (Due : Week 7, midnight)

Objective : In this project, you will add functionality to the GUI from Assignment 1 to perform the following image operations:
Addition, subtraction, product, negative, Power (Gamma) Transform, Log Transform, Histogram Equilization, Smoothing filters and Sharpening filters, Connected component labeling.

Implementation details:

Your application should support the following functionality:

- Storing the output of the operations to the disk (PPM format).
- Applying more than one operator to the input image. In other words, once an operation has been applied to an image, the next operation should be applied to the resulting output image and not to the original image. This process implements the so-called “cascade of image operations.”
- For operations that involve multiple images, the corresponding input images may be of different dimensions. There are multiple ways to solve this problem: first resize both images to a common size and then apply the operation. This will invariably lead to accuracy issues. How can you define a “common” size for the two images? It could be - the window size, or the larger image size.
- Provide the facility to view a histogram of an image using appropriate windowing controls / canvas. In case of Histogram equilization, the user should be able to view the histograms of the source and result images, and compare the results of the operation.

Grade Distribution :

Part A : Image Operations (10 points total)

The following operations should be implemented:

- *Addition, Subtraction, Product* (3 points)

The user should have the capability of choosing two image files required for the operation. The result of the operation should be displayed and stored in a separate file (also specified by the user). Values beyond the valid range [0,L-1] should be appropriately processed (renormalize or clamp). **Input images must be resized to the “common” size before applying the operation.**

- *Image Negative* (1 point) - Perform image negative operation on a PPM file. DO NOT make assumptions regarding the maximum value (L) stored in the file. This value should be read/ inferred from the file data.

- *Log Transform, Power (Gamma) transform* (3 points each)

The user should have the ability to specify the 'c' and 'gamma' values for these operations. For the above operations, ensure that the values in the image can be displayed faithfully.

Part B : Histogram Equilization (10 points)

Perform histogram equilization operation on an input image. The process should be completely automated without any user driven input.

Part C : Smoothing and Sharpening Filters (10 points)

Implement the Gaussian smoothing filter. The user should input the value of the kernel size N and the standard deviation (sigma). You should then construct a NxN digital filter by sampling the 2D continuous function at appropriate locations and using the discrete mask to perform spatial filtering. Assume suitable behavior (signal repeat, zero padding) at the boundaries. Justify your choice. (5 points)

Implement edge detection using the Sobel operators. The output is the addition of the outputs of the individual horizontal and vertical filters. (2 points)

Implement Unsharp-masking operation. The user should input the value of scaling factor 'k.' Use the current properties of the gaussian blur filter (see above) to implement the blur. (3 points)

Part D : Connected Component Labeling (10 points possible)

Implement the connected component labeling algorithm as we have discussed in class. The algorithm is a two-pass algorithm. See the Wikipedia entry for details. The description of the classic algorithm assumes that the input image is a binary image. How would you extend it to support an 8-bit image as the ones in the test data set? **Your implementation MUST support 8-bit images**.

Points will be awarded for implementing the algorithm using different topological operators as below:

(a) 4-connected topology - 5 points
(b) 8-connected topology - 7 points
(c) m-connected topology - 10 points

Part E : Answer the following questions (10 points)

Provide your answers in a README.[doc/pdf] (see below). Use images from your program to demonstrate a point if necessary.

1. Compare and contrast the two approaches for resizing images (resizing to a common size, vs resizing to the larger image size) when performing an image operation that involves more than one image. Which approach is suitable for a medical imaging application like an MRI? Which approach is suitable for a simulation application like finite element modeling on the GPU?

2. We believe that the histogram equilization process always gives a better-contrast image than the original. We also believe that it is a non-destructive process i.e. it will not reduce the contrast in an already "good-enough" image. Show that the preceding statements are FALSE with a counter example. Use a 2x2 image to demonstrate your answer.

Part F : Extra credit I (10 points)

Implement a 'better' connected component labeling algorithm than the naive two-pass version. Compare the space and time complexities of the original (Part D) and your own implementation (Part F) in detail and justify why your algorithm would be a better alternative to the classical version. Use the README file to state your answers.

Part G : Extra credit II (20 points)

We are going to extend the functionality of the command-line by adding the ability to supply operations via command line. At the end of the command execution, display the contents of the output file as the "post-processed" output for display. The general format of the command line for operations is

"<operation name> <parameters> -i <input file(s)> -o <output file>".

A description of the operations is given below:

Operation	Parameters	Description
add		Perform the addition of the intensities of the two images and write output to the specified file.
`add -i beetle-13.ppm apple-20.ppm -o output.ppm` Add contents of files beetle-13.ppm and apple-20.ppm (assuming they are of the same size) and write the output to the file output.ppm		
sub		Subtract second image from first image and write output to the specified file.
`sub -i beetle-13.ppm apple-20.ppm -o output.ppm` Subtract contents of file apple-20.ppm from beetle-13.ppm (assuming they are of the same size) and write the output to the file output.ppm		
mul		Perform element-wise multiplication of the two images and write output to the specified file.

Operation	Parameters	Description
`mul -i beetle-13.ppm mask-1.ppm -o output.ppm` Multiply contents of file beetle-13.ppm with mask-1.ppm (assuming they are of the same size) and write the output to the file output.ppm		
inv		Perform inversion operation (negative) on the current image and save output to the specified file.
`inv -i beetle-13.ppm -o output.ppm` Perform inversion (negative transform) on the file beetle-13.ppm and write the output to the file output.ppm		
log	-c <constant> -b <base>	Perform log transform to the specified base and save output to the specified file.
`log -c 1.25 -b 15 -i beetle-13.ppm -o output.ppm` Perform log transform to base 15 (with multiplicative constant c = 1.25) on the file beetle-13.ppm and write the output to the file output.ppm		
pow	-c <constant> -gamma <exponent value>	Perform gamma transform on the input file and save output to the specified file.
`pow -c 2.0 -gamma 0.35 -i beetle-13.ppm -o output.ppm` Perform gamma / exponential transform with c=2.0 and gamma=0.35on the file beetle-13.ppm and write the output to the file output.ppm		
gblur	-N <kernel size> -sigma <floating point value>	Implement a gaussian blur with kernel size NxN and standard deviation sigma. Save output to the specified file.
`gblur -N 7 -sigma 10 -i beetle-13.ppm -o output.ppm` Construct a gaussian blur filter of size 7x7 with standard deviation = 10. Next, convolve this filter with the input image beetle-13.ppm and write the output to the file output.ppm		
sobel		Perform the sobel filtering on input image and save output to the specified file.
`sobel -i beetle-13.ppm -o output.ppm` Convolve the input file beetle-13.ppm with horizontal and vertical sobel filters and write the output to the file output.ppm		

For all of the above examples, the contents of the file output.ppm should be displayed in the output panel after the command has been executed.

Q: How will you implement unsharp masking using the above commands? Try it and include your steps and output of each step in the README file. What is the next "logical" extension of our application GUI after this functionality is implemented?

Note that for all appropriate operations, images of different size must be resized to a common size before performing the operation.

Deliverable : For submitting your assignment, create an archive that is named as follows

CS370_PP2_*<digipen_login_id>*.**zip**

Include the following in the archive:

(a) ./src/ - complete source code of the application including any project files that you create (do NOT include the .ncb/.sdf file if compiling on Windows). Include the suitable project/ Make files to compile the project.

(b) ./doc/README file - a short description of your solution to the programming project. This should include a description of your data structures for storing images and performing Image Operations. Include description of extra credit if you have attempted it. In addition, provide examples of sample runs of your code on some of the test images. You may submit the file as a Word document, OpenOffice document, or PDF file.

Failure to submit the deliverable in the required folder structure will result in a ZERO assignment grade.

Late Submission : Refer to the course outline for the late submission guidelines.

Grade Sheet for Programming Project 2 (CS370/570)

Item	Points Possible	Points Awarded	Comments
Part A			
Addition	1		
Subtraction	1		
Product	1		
Negative	1		
Log Transform	3		
Power (Gamma) Transform	3		
Part B			
Histogram Equilization	10		
Part C			
Gaussian Filter	5		
Sobel Operator	2		
Unsharp Masking	3		
Part D			
Connected Component Labeling	10		
Part E			
Question 1	5		
Question 2	5		
Part F			
Extra Credit I - 1 Pass CC	10		
Extra Credit II - Cmd Line	20		
Penalties, if any			
TOTAL GRADE	**50**		

