CSC 205 - SPRING 2016
2D GRAPHICS AND IMAGE PROCESSING
ASSIGNMENT 4
UNIVERSITY OF VICTORIA

Due: Tuesday, March 15th, 2016 by noon.

1 Assignment Overview

The assignment involves implementing four image filtering operations: two point operations to manipulate histograms and two linear filters. All of the operations will process 8 bit grayscale images encoded as PNG files. References to the Burger and Burge textbook in the specification below refer to Volume 1 of *Principles of Digital Image Processing* by Burger and Burge.

You may implement your solution in any language that can be run (using non-proprietary software tools) in ECS 354, ECS 242 or ECS 266, but you are encouraged to write code which is not platform-dependent. A simple image processor template has been provided in C++, Java and Python. As with the previous assignments, you are not required to use any of the starter code in your final submission.

If you use a language which relies on platform-specific libraries, you may find that it is easier to complete the entire assignment on a lab machine, instead of working on it on your own machine, since it can be difficult to resolve library version conflicts in some cases. The sample code for C++ and Python has been tested on the ECS 354 machines and a recent Debian-based linux distribution. Talk to your instructor if you need help compiling or running the sample code on your own machine. The compile/build/run process for the starter code will be covered in the lectures and labs.

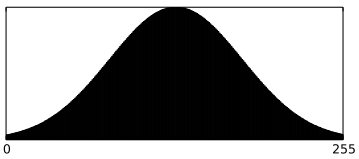
1.1 Histogram Manipulation

The normal (or Gaussian) distribution with mean μ and standard deviation σ is a continuous probability distribution for real numbers, and its probability density function is given by

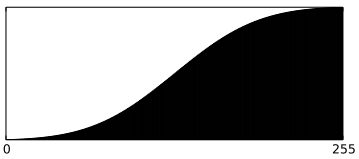
$$P(X \leq x) = f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

The PDF $f(x)$ of the normal distribution produces the familiar 'bell curve' centred at the point $x = \mu$, and the parameter σ influences the width of the bell curve (larger values of σ produce a wider bell). For a given value of μ, the maximum value of $f(x)$ is attained at the point $x = \mu$.

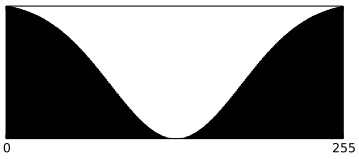
In Figure 4.15 (page 77) of Burger and Burge, the normal distribution is used experimentally as a reference histogram for histogram equalization. Your task is to write image processors to match histograms against the normal distribution $f(x)$, and to match histograms against the function $g(x) = f(\mu) - f(x)$, which produces an 'inverse bell curve'. For both functions, you should use $\mu = 128$ and $\sigma = 50$ by default. The figures below show the two reference distributions and their cumulative histograms.



(a) Reference histogram for $f(x)$



(b) Cumulative histogram for $f(x)$



(c) Reference histogram for $g(x)$

(d) Cumulative histogram for $g(x)$

Figure 1: Reference histograms and cumulative histograms for the normal distribution $f(x)$ and the inverse bell curve $g(x)$

You should submit two separate programs, one for each reference distribution. Ensure that your programs are easy to differentiate during your assignment demo.

1.2 Gaussian Blur

The Gaussian blur filter is based on the two dimensional normal distribution centred at $x, y = 0$. The coefficients of the filter can be generated by the function

$$G_\sigma(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$$

which is Equation 5.12 (page 109) of Burger and Burge. This function differs from the normal distribution itself in the lack of a multiplicative constant (which is eliminated to ensure the resulting filter is normalized).

Write an image processing program which applies a 5×5 Gaussian blur filter to the input image. You will receive extra marks (see Section 3) if you implement two versions: one which applies the 5×5 filter directly, and one which separates the filter into 5×1 and 1×5 filters and applies those in succession. Separability of Gaussian filters is discussed on page 114 of Burger and Burge.

1.3 Edge Sharpening

Difference filters, including the Laplace filter, can be used for edge detection in images. Once the locations of edges are known, manipulations can be targeted to pixels on edges to produce sharpening effects. Section 6.6.1 (page 147) of Burger and Burge discusses the use of a Laplace filter for edge sharpening. In particular, the book suggests

$$H^L = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

as a 3×3 Laplace filter[1], and specifies the equation

$$I' = I - w(H^L * I)$$

as a model for edge sharpening with the Laplace filter, where w is a strength parameter (which may be adjusted depending on the application).

Write an image processing program which performs edge sharpening with a Laplace filter. You are not required to use the Laplace filter above (the book provides several other suggestions). You will receive extra marks (see Section 3) if you implement a second version which performs 'Laplacian-of-Gaussian' (LoG) edge sharpening, specified by the equation

$$I' = I - w(H^L * G * I)$$

where G is a Gaussian blur. The LoG filter is often used instead of a plain Laplace filter, since the Laplace filter is highly sensitive to noise in the image, which can be mitigated by the blur filter.

2 Externally-Sourced Code

You are permitted to use appropriately licensed code from an external source, if full attribution is given (including the name of the original author, the source from which the code was obtained and an indication of the terms of the license). Note that using copyrighted material without an appropriate license (such as the GPL, LGPL, BSD License or MIT License) is not permitted. Short fragments of code found on public websites (such as StackOverflow) may be used without an explicit license (but with the usual attribution). If you embed externally sourced code in your own code files, add a citation in your code file and include a README file detailing the sources of all code used. If you use entire, unmodified source files from an external source, a README file is not necessary as long as the file in question contains complete authorship and licensing information. If you submit the work of others without proper attribution, it will be considered plagiarism. Additionally, for assignments in CSC 205, the following two caveats apply to externally sourced code.

- You will not receive any marks for the work of others. That is, if you use externally sourced code to implement the core objectives of the assignment, you will only be marked on the parts of the code that you personally wrote. Therefore, you should only use externally sourced code in a supplementary capacity (such as including a third-party hash table implementation which your code uses to store data, or a third party Gaussian elimination package to solve matrix equations). If you have any doubts about what is considered supplementary, contact your instructor.
- You may not use externally sourced code from another CSC 205 student, or share your code with another CSC 205 student (whether they intend to use it or not), without explicit permission from the instructor.

---

1. There is no definitive $n \times n$ Laplace filter, since the Laplace operator is not defined for discrete functions, so any discrete Laplace filter is an approximation.

3 Evaluation

Submit all of your code electronically through the Assignments tab on conneX. Your submission should include a complete set of files needed to compile, run and demonstrate your implementation's functionality for marking. Your implementation will be marked out of 80 during an interactive demo with an instructor. You may be expected to explain some aspects of your code to the evaluator. Demos must be scheduled in advance (through an electronic system available on conneX). If you do not schedule a demo time, or if you do not attend your scheduled demo, you will receive a mark of zero.

You may receive up to 90 marks on this assignment, but the final assignment mark will be taken out of 80 (so any marks you receive beyond 80 will be treated as bonus).

The marks are distributed among the components of the assignment as follows.

Marks	**Feature**
16	The program for histogram matching against the function $f(x)$ given in Section 1.1 functions correctly on a variety of test images.
16	The program for histogram matching against the function $g(x)$ given in Section 1.1 functions correctly on a variety of test images.
18	The 5×5 Gaussian blur filter functions correctly on a variety of test inputs.
6	A second Gaussian blur filter which uses separate 5×1 and 1×5 filters to perform the blur is implemented and functions correctly.
2	The value of σ for the Gaussian blur filter can be changed at run time by the user (for example, by providing it as a command line parameter).
18	The Laplace edge sharpening program functions correctly on a variety of test inputs.
6	An additional Leplacian-of-Gaussian edge sharpening program is implemented and functions correctly.
2	The value of w in the edge sharpening equation can be changed at run time by the user (for example, by providing it as a command line parameter).
8	For all of the filtering programs, out-of-bounds pixels are handled with reflection or smearing (instead of assuming all out-of-bounds pixels have a constant value).

Additionally, you may receive extra marks (with the total overall mark capped at 90) for other features of your own design, if you have received permission from the instructor before the due date.

Ensure that all code files needed to compile and run your code in an ECS lab are submitted. Only the files that you submit through conneX will be marked. The best way to make sure your submission is correct is to download it from conneX after submitting and test it. You are not permitted to revise your submission after the due date, and late submissions will not be accepted, so you should ensure that you have submitted the correct version of your code before the due date. conneX will allow you to change your submission before the due date if you notice a mistake. After submitting your assignment, conneX will automatically send you a confirmation email. If you do not receive such an email, your submission was not received. If you have problems with the submission process, send an email to the instructor **before** the due date.