EE7150 Computer Projects VIII & IX

Boundary Extraction (Chapter 9) & Segmentation (Chapter 10)

Due Date: November 29, 2021

1. **Edge Detection Combined with Smoothing and Thresholding**
   (a) Use the spatial filtering approach to compute the Sobel gradient using the masks in Fig. 10.14 [both Editions]. The program should implement Eq. (10-26) of 4th Ed. [Eq. (10.2-20) of 3rd Ed.], and have the option of outputting a binary image by comparing each gradient point against a specified threshold, *T*.
   (b) Download Fig. 2.39(c) of 4th Ed [Fig. 2.35(c) of 3rd Ed.] for this part. By combining smoothing with a 3 x 3 mask and the program from 1(a), process the kidney image and produce a binary image that isolates (segments) the large blood vessel in the center of the image. This will require repeated trials of smoothing and choices of *T*. Examining the histogram (Project 2) of the gradient image before it is thresholded will help in selecting a value for *T*.

2. **Basic Global Thresholding**
   (a) Write a program to implement basic global thresholding in which the threshold is estimated automatically using the procedure discussed in Section 10.3 [both editions]. The output of your program should be a segmented (binary) image.
   (b) Download the fingerprint image in Fig. 10.35(a) of 4th Ed. [Fig. 10.38(a) of 3rd Ed.] and apply your program to it. The result should match that in Example 10.13 of 4th Ed. [Example 10.15 of 3rd Ed.].

3. **Optimum Thresholding**
   (a) Implement Otsu's optimum thresholding algorithm (Section 10.3), [or you may use matlab built-in function].
   (b) Download Fig. 10.36(a) of 4th Ed. [Fig. 10.39(a) of 3rd Ed.] and use Otsu's algorithm to reproduce the results in parts (c) and (d) of the figure.

4. **Boundary Extraction using Morphological and Other Set Operations**
   (a) Write a computer program capable of performing binary dilation and erosion with an arbitrary structuring element of size 3 x 3 that can be specified by the user [or you may use matlab built-in functions].
   (b) Write a computer program for performing set intersection, differencing, and complementation (See Section 2.6).
   (c) Use the codes in part (a) and (b) to implement morphological boundary extraction given in Eq. (9-18) of 4th Ed. [Eq. (9.5-1) of 3rd Ed].
   (d) Download Fig. 9.16(a) of 4th Ed. [Fig. 9.14(a) of 3rd Ed] and extract its boundary.

5. **Connected Components**
   (a) Use your results from Part-4 to write a computer program capable of extracting (and counting) the connected components from a binary image.
   (b) Download Fig. 9.20(a) of 4th Ed [Fig. 9.18(a) of 3rd Ed.] and approximate the results in Example 9.7.

Project Report Guidelines

Page 1. Cover Page. Typed or printed neatly.

Project title
Project number
Course number
Student's name
Date due
Date handed in

Abstract (not to exceed 1/2 page) **Page 2.** Technical discussion. One to two pages (max). This section should include the techniques used and the principal equations (if any) implemented.
Page 3 (or 4). Discussion of results. One to two pages (max). A discussion of results should include major findings in terms of the project objectives, and make clear reference to any images generated.

Results. Includes all the images generated in the project. Number images individually so they can be referenced in the preceding discussions. Include titles, labels and legends in the figures, as appropriate.

Appendix. Program listings. Includes listings of all programs written by the student. Standard routines and other material obtained from other sources should be acknowledged by name, but their listings need not be included.

A note on program implementation: The objective of the computer programs used in the projects is to teach the student how to manipulate images. There are numerous packages that perform some of the functions required to implement the projects. However, the use of "canned" routines as the only method to implement an entire project is discouraged. For example, if a student is using MATLAB and the Image Processing Toolbox, a balanced approach is to use MATLAB's programming environment to write M functions to implement the projects, using some of MATLAB's own functions in the process. A good example is the implementation of the 2-D Fourier Fast Transform. The student should use the MATLAB function that computes the 2-D FFT directly, but write functions for operations such as centering the transform, multiplying it by a filter function, and obtaining the spectrum.