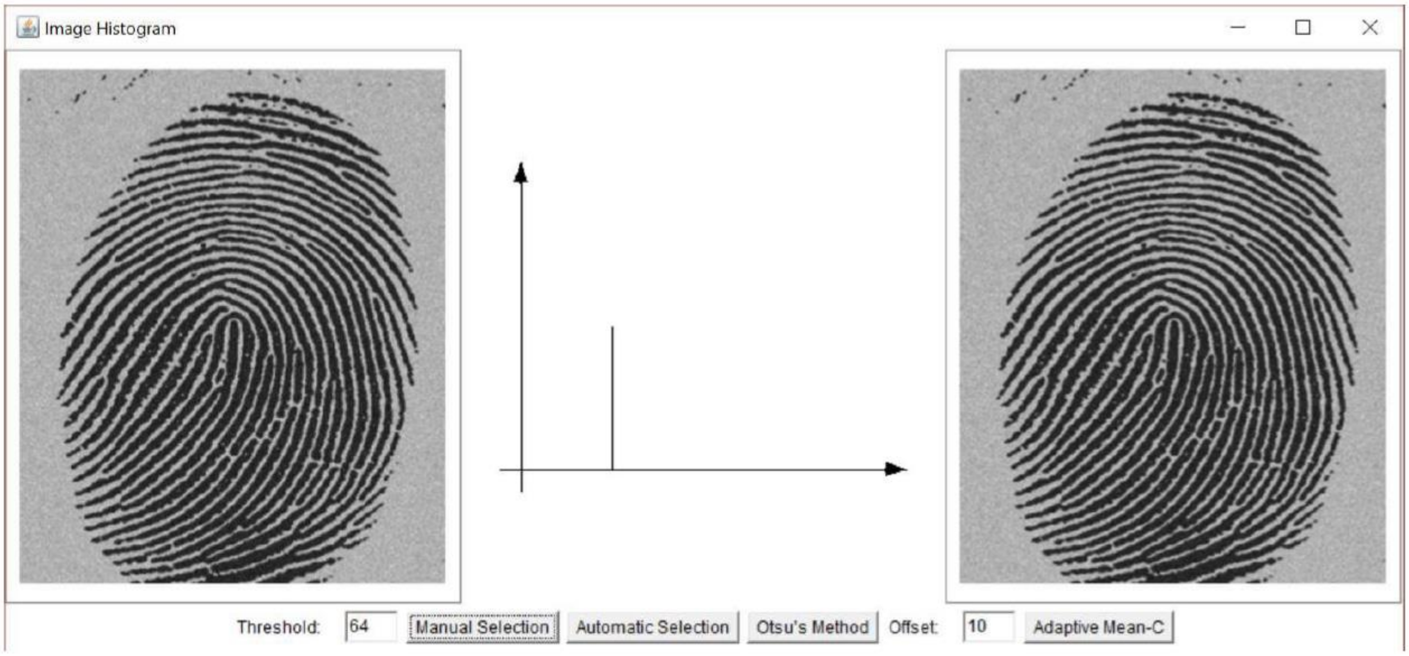
Image Threshold

COMP 3301 — Assignment 3 Due: November 3, 2020 (Tuesday) 11:59 PM
Group Self-Enrolment Expiry Date: Oct 21, 2020 @ 12pm

Objectives:
In this assignment you will implement different image threshold approaches. The goal is to give you a further understanding on these methods and allow you to compare their effects.

Your Task:
Given an input image, your first need to display its histogram (or histograms, if it is a color image) as you did in your previous assignments, prior to pressing any button.
Then, your task is to implement the following four thresholding approaches: manual thresholding using user-specified threshold value, automatic thresholding based on image histogram, Otsu's thresholding method, and adaptive thresholding with 7×7 window size and user-specified offset value. Upon selection of any of the first three global thresholding approaches, you are also required to display the threshold value used (or produced) in the histogram as a vertical bar, to highlight this value.
When the input is a color image, you need to perform thresholding to different color channels independently. This means that, for each of the last three automatic thresholding approaches, the threshold values used for the three color channels could be different. This also implies that for color images, there might be three bars of different colors, since the threshold for the color channels could be different.



Output of the skeletal program

Getting Started:
A skeletal program is supplied to get you started, which you are required to use as the basis of your solution. To run the program, put the test image in the same folder. The program opens a window that contains three panels, displays the test image in two of the panels, and plots a coordinate in the middle panel. As a demo, the program

also displays the user-specified threshold value when the “Manual Selection” button is pressed.

Grading:

Your program will be tested and graded using the standard Java environment (not Enterprise). The grade will be based on your program’s functionality (whether or not it works under different settings), as well as the efficiency of your implementation. The weights for different components are as follows:

- Result of global thresholding using user-specified threshold value 20%
- The correctness of threshold values identified under the automatic selection method 20%
- The correctness of threshold values identified using Otsu’s method 30%
- Result of adaptive thresholding using Mean-C approach (using the value from the Offset parameter) 30%

Some foreseeable deductions are:

- Missing to update the threshold bar: -5%
- Failure to handle color images properly: -20%
- Failure to include helper source files: -10%
- If the documentation mentioned below is missing or incomplete, anywhere from 10% to 20% will be deducted.
- If essential source files (.java) are missing, anywhere from 50% to 100% will be deducted, depending on how much is missing.

What and How to Hand in:

You are handing in the source of your program (source code is essential), as well as any data files required for running your program. Your source code must contain sufficient internal documentation to facilitate grading. This includes names & student numbers of all contributing members of the group, a brief description of each member’s individual contributions, an explanation of what the programs do, and a listing of known bugs and features and things that are not working, if any. This can be included either in the dropbox comments window, and/or in an additional text file.

Send in your source program through the Direct2Learn’s Dropbox as a single .zip file. The submitted file must follow this naming convention: A3_GroupX_Lastname1_Lastname2_Lastname3.zip.

Verifying your submission:

To ensure your application is considered as submitted complete by the deadline, you should be careful to make sure each of your submissions has been properly done. Once you have placed your assignment into the Assignment’s Dropbox, click on the link that indicates our submission has been done, download it, open it (preferably on a different computer), revise the contents, test that the program you have submitted actually is the one intended (not the source files, for example), and if it does not correspond to what you want the TA to mark, you must resubmit again, until you are satisfied, that the submission has all that you need the marker to see. Please note that the TA will only mark the last submission with a timestamp that is prior to the indicated deadline.