School of Computer Science and Applied Mathematics
UNIVERSITY OF THE WITWATERSRAND, JOHANNESBURG

APPM4058A & COMS7238A: Digital Image Processing
Assignment 2

Hand-out date: April 29, 10:00
Hand-in date: May 24, 23:59

Instructions

- This is an individual assignment, adhere to the academic integrity. Plagiarism will result in 0%.
- The due date is strictly applied.
- Before the final hand in, submit your solution PDF file to Turnitin to produce a similarity check report. This report will be viewed by the lecturer only.
- Hand in the electronic files and source code on Sakai. See more instructions in Section **Hand-Ins**.
- You may use built in image processing functions, unless otherwise specified where image processing algorithms must be coded from scratch.
- Questions starting with "(**Lab**)" require implementations.
- The total marks available for this assignment is 46.

Objectives

- Understand and apply basic morphological algorithms for both binary and gray scale image processing.

Problems

1. **Hit-or-miss transform.**

   (a) Apply a hit-or-miss transform for the binary image I in Figure 1 using structuring elements given in Table 1. The origins of B_1 and B_2 are indicated in bold. Illustrate the output binary image in a table by simply write out the pixels with '1' values, and leave the pixels with '0' values empty. [6]

Table 1: Structuring elements for Question 1a

0	0	0
0	**1**	0

B_1

1	1	1
1	**0**	1

B_2

I $\quad$ I_o

Figure 1: Images for Question 1a

(b) How would you obtain the output image I_o in Figure 1 by applying hit-or-miss transform to image I in Figure 1? (Hint: what kind of SEs would you use?) [4]

2. **Connected component labelling.**

   (a) The algorithm for extracting connected components discussed in lecture `chap5_2` requires a pixel in a connected component known in order to extract the connected component. Suppose you are given a binary image with a number of connected components. Propose an automated process, i.e. no foreground pixels in a connected component is given, for extracting all connected components. [5]

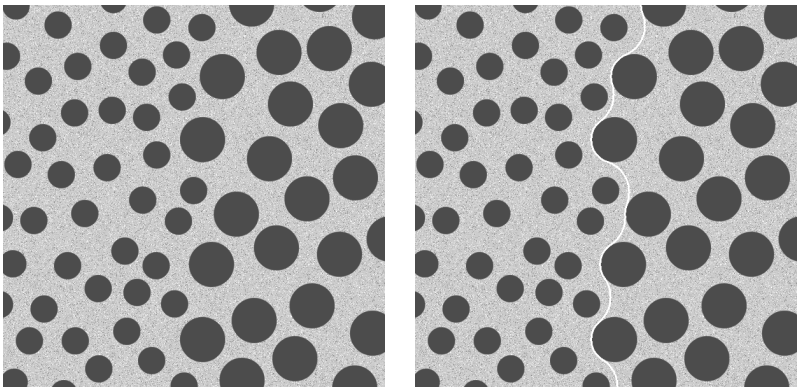
   (b) (**Lab**) Implement your proposed method for Question 2a and test it using images **DIP.tif** and **balls-with-reflections.tif**. Your output should include i) an image of labels starting from 0, where all the background pixels are labelled as 0, and all the connected components are labelled by the series of integers $1, 2, 3, \ldots$, respectively; ii) the number of connected components without including the background component. Display the label image in similar fashion shown in Figure 2. [8]

   (c) (**Lab**) Compare your results with those from Matlab or Python built-in functions for connected component labelling by displaying the labelled images side by side. [1]

Figure 2: Displaying the connected component labelling result.

3. (**Lab**) **Gray scale morphology.** Image **blobs.tif** contains two texture regions — circular blobs of smaller size on the left, and circular blobs of larger size on the right (see Figure 3, (a)). Find the boundary between the two regions using suitable morphological approaches we learned in class. The result should appear like the one in Figure 3, (b). [10]

4. (**Lab**) **Coin separation and counting.** Assume the circles in Image **coins.png** represent coins. Design an approach that could separate the coins, and count the coins through connected component labelling. For this, you can try the following steps: i) separate the coins using suitable morphological approach(es); ii) binarize the image using thresholding; iii) perform connected component labelling. Your final labelling could appear similar to the image in Figure 4. [12]



(a) (b)

Figure 3: (a) Original image; (b) Two texture regions are separated.

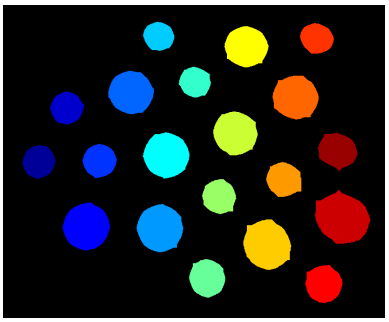


Figure 4: Coins separated and labelled.

Hand-Ins

Before your final hand-in, submit your solution PDF file in Turnitin to produce a similarity report. This report will be viewed by the lecturer only.

A compressed file (.tar.gz) named by your student number is to be submitted on Sakai. This file when extracted must include:

- Your solution PDF file, named as follows: "assignment_2_<student_no>.pdf" (e.g., "assignment_2_0000000.pdf"). Put solutions for both general and lab questions in the same document. In particular, for each **Lab** question (clearly marked by the question number) include the following:
  1. display the input images and output images with suitable sizes for view.
  2. discussions and comments on the techniques and results.
  3. source code list following your discussions.
- Your source code folder "src" containing separate python or matlab files for each problem. All code must be clearly commented on top of generating appropriate images and graphs included in your report when run.
- An output folder, containing output images for each problem which must be saved as "q<question_number>.png" (e.g., "q1.1.png").