

The Hong Kong University of Science and Technology  
Department of Computer Science and Engineering  
MSBD5010 (Fall 2020)

## Assignment 2

Total = 100 marks

**Due: 11:55pm, Nov. 22, 2020**

Assignments must be submitted via Canvas

Late Policy: 10% reduction; only one day late is allowed, i.e., 11:55pm, Nov 23.

---

### Overview

The assignment is designed for investigating three typical segmentation methodologies, which are watershed segmentation algorithm, active contour model and Hough transform respectively. You cannot also use the built-in functions relating to the watershed algorithm, active contour model and circular Hough transform. All the work should be submitted via Canvas system.

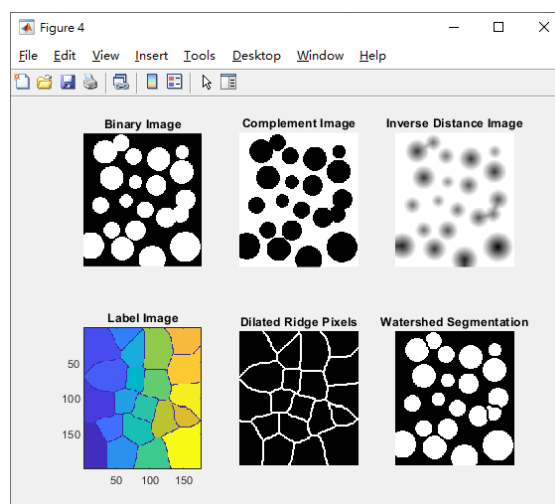
### Programming assignment specifics (100%)

#### Part 1: Watershed segmentation algorithm (30%)

You need to complete the implementation of the routine in *my\_watershed.m*. It leverages the Watershed segmentation algorithm to segment all the circles in the input image. You are not allowed to use the `watershed()`, `imdilate()`, or other related built-in functions for this question.

- Given an inverse distance map of the input image, compute the segmentation label map using Watershed segmentation algorithm defined in the lecture notes. Assign 0 for ridge/dam and unique integer in  $[1, \dots, n]$  for each region.
- Extract the ridge/dam pixels and set it to 1. Then, apply binary dilation to the ridge/dam map.
- Superimpose the image of dilated ridge/dam map and original image to separate distinct circles in the original image.

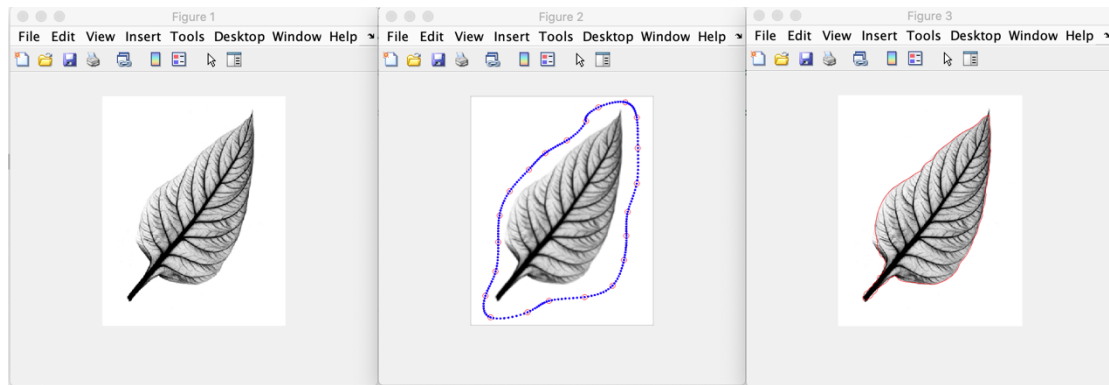
The sample output for this task is shown below:



### Part 2: Active contour model (50%)

You need to complete the implementation of the routine in *my\_active\_contour.m*. It leverages the snake contour model proposed by Kass etc. in 1988 to segment the leaf in the input image. You are not allowed to use the `conv()`, `gradient()`, `imfilter()`, `fspecial()`, `circshift()`, `interp2()`, or other related built-in functions for this question.

The sample output for this task is shown in Fig. 3:



Note:

- You can suppose the weight  $\alpha$  and  $\beta$  of the internal energy are uniform over the snake.
- The function for generating the initial contour has been provided (*get\_initial\_contour.m*). This function requires the user to left-click the mouse on the shown image to obtain the coordinates of control points for the initial contour. Then the initial contour will be generated automatically. Please right-click the mouse to stop the function. For the naïve snake model, the initial contour should be close to the object boundary, otherwise the final active contour will fail to converge to the desired location. The TAs will try their best to mark the control points as close as to the image boundary during grading.
- Please state the derivation process of the matrix A (Lecture note: image-segmentation-part-2). You can either write your statement in the provided comment area in the Matlab code, or write it down in a .txt file or .pdf file. (10%)**
- When updating the next locations of the contour, you are required to implement your own interpolation method. The nearest interpolation is not allowed to used. (10%)**
- For your reference: Kass, Michael, Andrew Witkin, and Demetri Terzopoulos. "Snakes: Active contour models." *International journal of computer vision* 1.4 (1988): 321-331.

### Part 3: Hough transform (20%)

You need to complete the implementation of the routine in *hough\_transform.m* that can be used to detect circles in a binary image with circular Hough transform. Given that the range of radii of the circles in the input image is [40, 80] pixels. You are not allow to use the `imfindcircles()` Matlab built-in function for this question.

- a) Find the edge of the input image. (5%)
- b) Perform the circular Hough Transform and create a variable 'Accumulator' to store the bin counts. (10%)
- c) Search for the highest count cells in the Hough accumulation array. (5%)

### **Sample run of the programming assignment**

The main routine of the assignment including displaying the final results is well written in the *msbd5010\_assign2.m* file. After completing all the functions, you are supposed to get figures on the screen when you run the command below in the MATLAB environment.

```
>> msbd5010_assign2;
```