The Hong Kong University of Science and Technology
Department of Computer Science and Engineering
COMP4421 (Fall 2019)

Assignment 2

Total = 100 marks
Due: 11:55pm, Nov. 15, 2019
Assignments must be submitted via Canvas
Late Policy: 10% reduction; only one day late is allowed, i.e., 11:55pm, Nov. 16.

Overview

Topics: Connected Components and Hough Transform

In this assignment, you need to finish three programming questions. The first one is about connected components, while the rest two questions are about Hough transform.
Please **paste your results of your programs in a PDF file** and submit it together with all relevant codes and result images in a folder. **Please follow the structure of "Sample" folder to place your codes and result images (otherwise marks will be deducted)**. Then you should rename the "Sample" folder by your student ID and the programming language you use and compress it into a zip file. For example, if your student ID is "12345678" and you choose to use Matlab, then your folder should be "12345678_matlab".

This assignment should be submitted via the Canvas system on or before the due date.

1. Programming Tasks

Write programs to finish the following tasks.

1.1 Pre-requirement

1.1.1 Input: The input images are in the folder "input_imgs" (/Sample/input_imgs). For Question 3, you should test all the input images and paste the result images in the PDF file. Note that we will have some other test images for grading.

1.1.2 Language: Matlab/Octave.

1.1.3 Functions to be used: In this assignment, you can use the built-in function for filtering and edge detection. However, you cannot use the built-in function relating to the connected components and Hough transform.

Question 1 Connected Components (20%)

You need to complete the implementation of the routine in connected_component.m that can be used to search for all connected components with connectivity equals to 8 (A 3x3 structuring element with all 1) on the input image. You are not allow to use the Matlab built-in functions *bwlabel()*, *bwlabeln()* and *bwconncomp()* for this question. In this part, you need to make sure the running time of your program will be less than 3 minutes, otherwise marks will be deducted.

a) Search for all connected components with connectivity equals to 8 and output the results to the command window in the following format: (15%)

"There are total {number of regions} region(s):
Region 1, no. of pixels = {number of pixels}
Region 2, no. of pixels = {number of pixels}
…
"

b) Find the largest connected component (in terms of number of pixels) on the input image. (5%)

Question 2 Hough Transform for Synthetic Cycles (30%)

You need to complete the implementation of the routine in hough_transform_syn.m that can be used to detect circles in a binary image with circular Hough transform. The radius of the circles in the input image is 50 pixels. You are not allow to use the Matlab built-in function *imfindcircles()* for this question.

a) Find the edge of the input image; (5%)
b) Perform the circular Hough transform and create a variable named *Accumulator* to store the bin counts; (20%)
c) Search for the highest count cells in the Hough accumulator array. (5%)

Question 3 Hough Transform for Practical Segments (50%)

For each input image, there is a piece of target A4 paper. In Question 3, your task is to detect four sides of the target A4 paper by Hough transform and then compute 4 corners of the target. You should mark the lines and the corners in your result images. Here, an alternative framework for this part is provided:

i) Pre-processing: de-noise and edge extraction. You can try all the operators based on what you have learned in the previous lectures, for example, linear filtering or morphological image processing;
ii) Hough transform. Please think about how to collect the target sides only since you will see there are some other unwanted edges in the input images.
iii) Compute the corner points based on the 4 line functions.

The function prototype is given in the hough_transform.m. You need draw the lines and corner points in the images and print the line functions in the command window. A sample output is provided as follows,

```
The function of the line 0 is: y = 0.487717 * x + 2572.95
The function of the line 1 is: y = 0.600834 * x + 362.285
The function of the line 2 is: y = -1.96262 * x + 8195.12
The function of the line 3 is: y = -1.73239 * x + 2749.09
The intersection points are: (2294.45, 3691.99) (79.3389, 2611.64) (3055.58, 2198.18) (1022.96, 976.916)
```