EE4704 CA2

Semester 1, AY2020/21

This assignment is to be done with Matlab. You may use built-in Matlab functions or write your own Matlab scripts. The code must be compatible with Matlab R2018.

A. Feature Measurement (10 marks)

1. The test image is test1.bmp (image *I*).
2. Input image is *I*. Implement the intermeans algorithm to calculate the threshold T_1 and use it to threshold *I*. The output image is *I*1. [intermeans.m]
3. Input image is *I*1. Calculate these features: perimeter, area, compactness, centroid, invariant moment ϕ_1. [features.m].
4. Note that:
   (a) You can check your implementation intermeans.m by testing it on image "letter.bmp". The calculated threshold value should be 87.
   (b) Perimeter and area – calculate using the method given in the notes or employ some other method that gives more accurate results.
   (c) Centroid – calculate from the moment values. Take the image origin to be at the bottom left corner, with the *x* axis pointing to the right.

B. Feature Invariance (10 marks)

1. The test image is test2.bmp (image *J*).
2. What do you think is the optimum threshold T_{opt} for segmenting the object accurately?
3. Obtain the intermeans threshold T_2 using intermeans.m.
4. Threshold *J* using T_2 and measure the features using features.m.
5. Threshold *J* with threshold T_{opt} and measure the features using features.m.
6. Compare the segmentation results obtained with T_2 and T_{opt}. Discuss the sensitivity of the measured feature values to the threshold values.

C Boundary Plot (5 marks)

1. Input image is the boundary image test3.bmp (image *K*).
2. Calculate the $r - \theta$ values and plot the graph. [rtheta.m]

D. Matlab Code

1. The same Matlab code is to be used for Sections A and B.
2. As part of the assessment process, your code may be tested on images other than test1.bmp, test2.bmp and test3.bmp to check that it is reasonably robust.
3. Follow the templates below to write your code as functions.

   intermeans.m

```
% To calculate the intermeans threshold;
% input is the gray level image 'test1.bmp'
% output is the threshold value T and the binary thresholded
image Iout.

function [T,Iout] = intermeans(Iin)
%
% put your code here
%
end
```

   features.m

```
% To compute the features;
% input is the binary thresholded image
% outputs are the feature values

function [P, A, C, xbar, ybar, phione] = features(Iin)
%
% put your code here
%
end
```

   rtheta.m

```
% To compute the r-theta plot;
% input is a boundary image 'test3.bmp'
% output is the array containing the r-theta value

function [r, theta] = rtheta(Iin)
%
% put your code here
%
end
```

4. In addition to the above, you will also have to provide scripts to test your functions.

run_A.m

```
%%%%% Section A %%%%%
% This m file is used to test your code for Section A
% Ensure that when you run this script file, the output images
are generated and displayed correctly

%--- 1. Display the thresholded image and the threshold
I = imread('./test1.bmp');
[T, IT] = intermeans(I);
imshow(IT) % display image IT
output = T % display the intermeans threshold

%--- 2. Display the measured feature values
[P, A, C, xbar, ybar, phione] = features(IT)
```

run_B.m

```
%%%%% Section B %%%%%
% This m file is used to test your code for Section B
% Ensure that when you run this script file, the output images
are generated and displayed correctly

%--- 1.
I = imread('./test2.bmp');
[T, IT] = intermeans(I);
imshow(IT) % display image IT
output = T % display the intermeans threshold

%--- 2
% display the measured feature values
[P, A, C, xbar, ybar, phione] = features(IT)

%--- 3
Iopt = I >= Topt; % threshold J with Topt
imshow(Iopt) % display image Iopt
% display the measured feature values
[P, A, C, xbar, ybar, phione] = features(Iopt)
```

run_C.m

```
%%%%% Section C %%%%%
% This m file is used to test your code for Section C
% Ensure that when you run this script file, the r-theta plot
is displayed correctly

%--- 1.
I = imread('./test3.bmp');
[r, theta] = rtheta(I); % calculate r and theta
% plot r-theta graph
```

E Report

1. Your report should focus on the results, observations, explanations and discussion. Relevant images should be included.

2. If you use any algorithms that are not from the lecture, you should explain how they work.

3. The GA for the assignment is Zhang Jianfeng (zhangjianfeng@u.nus.edu). You may consult him if you need any clarification on the assignment.

4. The softcopy of the report (pdf file) and the Matlab m-files are to be zipped and submitted to the "CA2-report" folder on the EE4704 module LumiNUS website, by 5 pm, 11 November.

5. The file is to be named as follows:
matric number_full name.zip (e.g., A010134J_Tan_Shu_King.zip).

6. ***The results and report must entirely be your own work. Plagiarism is a serious offence.***