Cover page

933 WORDS

5/5/2020

Image Processing

Item 1. (40%)

Mohab Magdy

ID: 20008429. OWA: HFYMA2.

Introduction

This module was so attractive for me even before reading or going through any lecture notes; that's why I have chosen it, because as we all know it is an optional module.

Image processing is really a crucial field and very outstanding, people always used to be visual creatures, we rely on our vision more than anything. Not only do we look at things to identify and classify them in detail, but also, we can scan for differences, and obtain an overall rough feeling for a scene with a quick glance in milliseconds.

The coursework was not that easy, but it was interesting watching yourself building, developing a code that can be used to play with different images and to extract the worms with different techniques.

The best part of this coursework is that I have learned how to develop myself and to do not only depend on the lecture notes, labs, and the lecturers.

While doing the coursework I have used a lot of different functions that have helped me a lot in obtaining the results.

From my perspective, we can divide them to 2 parts, basic and advanced.

Let me start with the basic.

- **Function number 1: imread ()**

This function takes the rgb values of all the pixels in the coloured worm image and puts them all into a matrix A. This matrix A is now a Matlab variable, and we can perform different matrix operations on it.

In general, the imread function reads the pixel values from an image file and returns a matrix of all the pixel values.

- **Function number 2: rgb2gray ()**

It converts the true colour RGB image to a grayscale image A_gray.

In general, this function converts RGB images to grayscale image by eliminating the hue and saturation information.

- **Function number 3: subplot (x, y, z)**

This function plots all the images I would like to print on the screen.

X: This is the number of rows.

Y: This is the number of columns.

Z: This is the position we choose to print the photo and it follows that order, starting from the top left going right by the first row.

Example:

1	**2**	**3**
4	**5**	**6**
7	**8**	**9**

We can here start with the advanced functions.

- **Function number 1: edge (sobel) and identifying the threshold**

This is a very important function.

It has 2 parts:

1. To find edges, we use the edge function. This function searches for places in the image where we have intensity changes.

   EDGE returns a binary image containing 1's where edges are found and 0's elsewhere.

   This method is therefore less likely than the others to be affected by noise, and more likely to detect true weak edges, that’s why it was very important for me to use it.

2. After EDGE does the Sobel filter, and then thresholds it. The threshold that it picked is returned in threshOut.

This function was so helpful in the code.

- **Function number 2: strel ()**

STREL function represents a flat morphological structuring element, which is an essential part of morphological dilation and erosion operations.

I have used the strel (‘line’) that creates a linear structuring element that is symmetric with respect to the neighbourhood center.

It has 2 arguments which is length and angle.

I have used 2 angles with the same length, angle 0 and angle 90 to perform 2 perpendicular lines.

- **Function number 3: imdilate ()**

As we know, erosion can be used for 3 purposes:

1. Can repair breaks.
2. Can repair intrusions.
3. Enlarges objects.

I have used this function to dilate the binary image, using both structuring element 90 and 0 degrees.

- **Function number 4: imfill ()**

Here, we fill the holes using keyword ‘holes’, because after dilation there were some holes that were not filled.

- **Function number 5: imerode ()**

As we know, erosion can be used for 3 purposes:

1. Can split apart joined objects.
2. Can strip away extrusions.
3. Can shrink objects.

I have used it 2 times to smooth the object as much as I can with the help of the keyword ‘diamond’ in the strel function.

- **Function number 6: imopen ()**

This is a function that only accepts grey scale images

That function simple definition is an erosion followed by a dilation.

I have used it for a lot of things, first to have a clear picture of the worm alone, and second to have an approximate calculation of the area by having fewer white parts (that has 1 pixel).

- **Function number 7: labeloverlay ()**

This function takes 2 arguments, both variables are images. The function takes the 2 images and overlay them to see the worm coloured and clear.

- **Function number 8: bwarea () and bwperim ()**

These 2 functions are so relatable, but they are different on the screen after applying them.

Bwarea calculates the area in pixels, it calculates the object pixels which means it counts the 1’s (in the binary images of course ‘to have 0’s and 1’s only).

Bwperim get only the boundaries and show how much space occupied by the object.

Conclusion

These are all the functions that I have used with this code to have the result I have reached till now.

I know they aren’t perfect 100%, but I did my best to get these results in that form.

To conclude, Image processing is one of the most enjoyable modules I have taken so far, and I would recommend it for any student to choose it and get the benefit out of it.