

Pt 2

by Paul Austin

FILE	3719_PAUL_AUSTIN_PT_2_116677_1299390341.DOCX		
TIME SUBMITTED	11-SEP-2018 02:08PM (UTC+0100)	WORD COUNT	2277
SUBMISSION ID	92231153	CHARACTER COUNT	15677

Name: Paul Austin

P Number: P148321

Module: Remote Sensing and Imaging

Assignment

Part 2

Lecturer: Dr Stephen Mosey

1 BRIEF

From the data captured during the practical session; you are required to process two images of the same object (before and after a particular event) using MATLAB to segment the images into regions of interest.

You will extract the images from the video files captured by the drone using the MATLAB script.

You should aim to extract the maximum number of valid whole features from the images by adjusting the filtering parameters used during the segmentation process.

Your report should include:

- a) A description of the methods you have used to segment the images.
- b) A discussion of the factors that will affect the accuracy of the machine vision processes you have employed.
- c) A discussion of how these two images could be used together to determine the impact the event has had on the targets in the images.

2 TABLE OF CONTENTS

3	List of Figures and Tables	3
3.1	List of Figures	3
3.2	List of Tables	3
4	Introduction	4
5	Method	5
5.1	Method	5
5.1.1	StrelSize.....	6
5.1.2	OpeningdiscSize	6
5.1.3	DilateSize.....	6
5.1.4	NumPixels.....	6
6	Accuracy.....	8
7	Impact	13
8	References	14
9	Bibliography	15
10	Appendix 1	16
10.1	MATLAB Script.....	16

3 LIST OF FIGURES AND TABLES

3.1 LIST OF FIGURES

Figure 1: Original Images imported into Matlab.	5
Figure 2: Original image example prior to dilation. Figure 3: Dilated image.	6
Figure 4: Segmentation of original image.	7
Figure 5: Segmentation of modified image.	7
Figure 6: Drone image prior to event taking place.	8
Figure 7: Drone image after event took place.	9

3.2 LIST OF TABLES

Table 1: Comparison of segmentation from the drone footage.	9
Table 2: Segmentation with StrelSize adjusted to 10.	10
Table 3: Segmentation with OpeningdiscSize adjusted to 6.	11
Table 4: Segmentation with dilateSize adjusted to 6.	11
Table 5: Segmentation with numPixels adjusted to 20.	12

4 INTRODUCTION

The human eye is capable of recognising the items that we see and translate them into meaningful data for our brains to process and translate. The question is, how can this process be automated to allow software to carry out visual comparison of items for us? This could potentially save analytical time in processing images captured along a variety of disciplines such as medical, military and forensic examination.

The natural way of doing this is to sub-divide the image into segments, based on colour, texture and depth of field.

This process is called digital Image Segmentation and can be defined as “the process of partitioning a digital image into multiple segments (pixels)”.¹

Within software, it is important that these segments should not overlap and have clearly defined boundaries. In addition, the software can use light intensity to differentiate between items in the foreground and background. This technique is particularly useful in detecting changes in digital imagery.²

The end result of image segmentation is a duplicated image from the original with defined segments segregated by a characteristic such as colour. These segments can then be used for comparison and / or reconstruction.

This technique has been used within this document to demonstrate its effectiveness in detecting changes within a digital image. Also demonstrated is the importance of accuracy with regard to the original images being used for the comparison and the impact that using images taken from different vantage points can have on such comparisons being made.

For the scope of this document, images have been made in a “before” and “after” state and analysed using MATLAB.

¹ “Computer Vision”, Shapiro. L, Stockman. G, New Jersey, Prentice-Hall, 2001, ISBN: 0-13-030796-3.

² Digital Image Analysis for Evidence: A MATLAB toolbox, Welford. S, Gibson. S, Payne. A, Digital Image Analysis and Evaluation (DIAnE): A Forensic Image Processing Tool using MATLAB. In: The 5th Cybercrime Forensics Education & Training Conference, 1st - 2nd September 2011, Canterbury Christchurch University, Canterbury, Kent.

5 METHOD

5.1 METHOD

The following process was adapted to demonstrate how image comparison can work.

To capture the original image, a camera was placed on a tripod. Using a timer so as not to cause any issue with movement during image capture the photograph was taken. Without moving the camera (or changing settings), an item was moved / added to the area and another photograph was taken in exactly the same way.



Original image.



Second image with modification.

Figure 1: Original Images imported into Matlab.

Using a pre-defined MATLAB script³ it has been possible to carry out a process of segmentation on both of these images. The MATLAB script carried out the following key processes on the images in order to complete segmentation:

- Import picture and lock aspect ratio.
- Re-format the picture into Black & White.
- Pass through a vertical "Y" filter.
- Pass through an "X" filter.
- Combine the images within the system.
- Mark the foreground markers, creating a flat maxima in the foreground and reconstruct the image.
- Set "close" the image.
- Calculate regional maxima to obtain new foreground markers.
- Carry out an erode to clean up the edges.
- Super impose onto original image.
- Create background markers.
- Combine these images and re-size.

³ See Appendix 1 - MATLAB Script.

The adjustable structuring elements within the script are:

- StrelSize provisionally set to 5.
- OpeningdiscSize provisionally set to 3.
- DilateSize provisionally set to 3.
- NumPixels provisionally set to 10.

5.1.1 StrelSize⁴

This is used as part of the MATLAB Strel command to create a structured element within a scaled image. The command allows for objects in the background that are smaller than the set “StrelSize” to be removed to allow for further analysis.

5.1.2 OpeningdiscSize⁵

Performs morphological opening on binary or intensity images.

5.1.3 DilateSize⁶

Allows for the grayscale and binary image to be diluted.

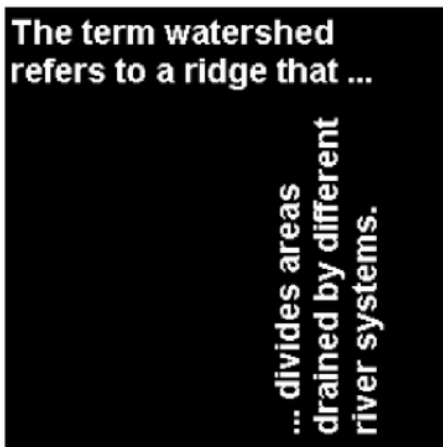


Figure 2: Original image example prior to dilation.



Figure 3: Dilated image.

5.1.4 NumPixels⁷

A pixel contains the value of your image at a location point within that image. The number of pixels held within the image show the level of detail to which the image is being displayed. Too little detail means that the image may not be sharp enough. Too much detail can result in the image being blurry.

⁴ MathWorks – Strel: https://uk.mathworks.com/help/images/ref/strel.html?s_tid=srchtitle – accessed 15:00 on 11th July 2018.

⁵ MathWorks – Opening: https://uk.mathworks.com/help/vision/ref/opening.html?s_tid=doc_ta – accessed 08:00 on 12th July 2018.

⁶ MathWorks – imdilate: <https://uk.mathworks.com/help/images/ref/imdilate.html> – accessed 09:00 on 12th July 2018.

⁷ MathWorks - how to code the total pixels number and white pixels and black area pixels number: <https://uk.mathworks.com/matlabcentral/answers/319328-how-to-code-the-total-pixels-number-and-white-pixels-and-black-area-pixels-number> - accessed 10:00 on 12th July 2018.

While the original images themselves are clear enough for a person to ascertain the main difference, the examples of the segmentation results provide evidence as to the advantages of using software for electronic comparison:

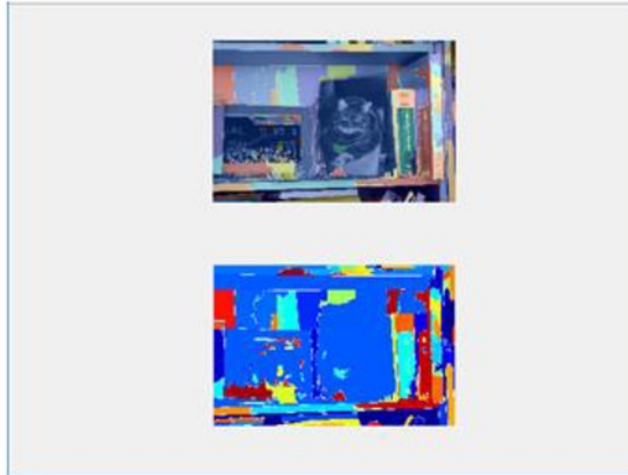


Figure 4: Segmentation of original image.

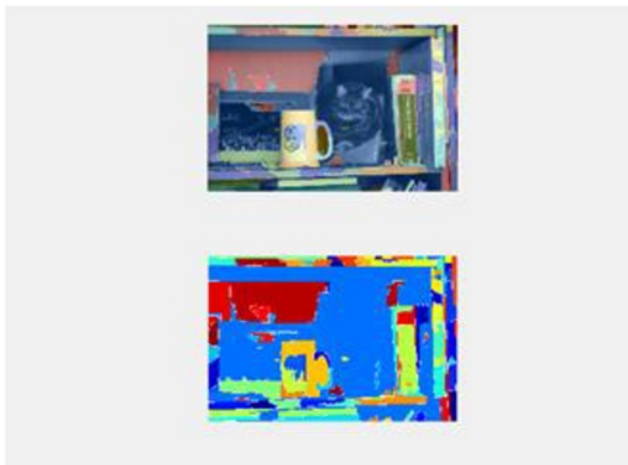


Figure 5: Segmentation of modified image.

Figure 5 clearly shows from the segmentation that a change has occurred. This shows the benefits of segmentation using software for analysis for more complex imagery.

6 ACCURACY

In order for an accurate comparison of images to be made it is important to consider the fact that the camera (or data capture device) needs to maintain its aspect ratio. As demonstrated above, to achieve this, the focus and location of the camera need to be consistent. Changes in this result in segmentation taking place but direct comparisons cannot be made.

The following images demonstrate the difficulty of image comparison when the image platform and data capture device are in different locations. These images were captured from video taken by a moving drone. The drone over flew an area and captured data both before and after an event. However, the drone was not at the same location or the same altitude when the before and after video was taken. Digital stills from the video could be taken to show a best fit comparison and these were then used to conduct segmentation.



Figure 6: Drone image prior to event taking place.



Figure 7: Drone image after event took place.

It is evident simply by looking at the pictures that the aspect ratio is different – but how does this affect the segmentation?

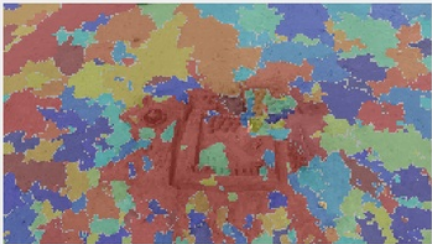
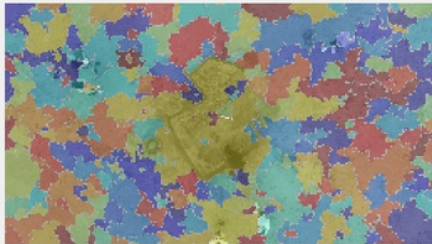
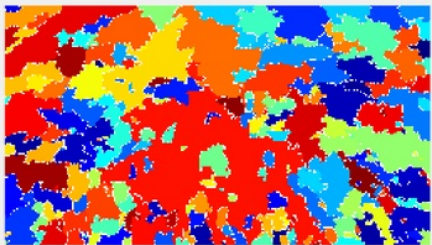
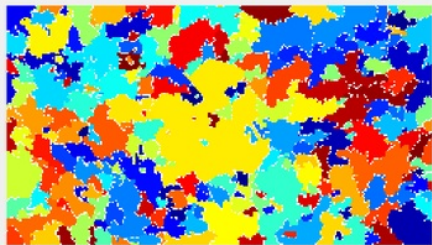
Before Event	After Event
	
	

Table 1: Comparison of segmentation from the drone footage.

While there are similarities within the segmented images, it can be seen that a direct comparison is difficult to achieve. While the software has been able to define boundary lines in both pictures, because the pictures were captured from different angles, these boundary lines for the pixilation process are substantially different.

Further editing within the segmentation process can be applied. The MATLAB script allows us to modify key aspects with the segmentation process. These are:

- StrelSize → above images set at 5.
- OpeningdiscSize → above images set at 3.
- dilateSize → above images set at 3.
- numPixels → above images set at 10.

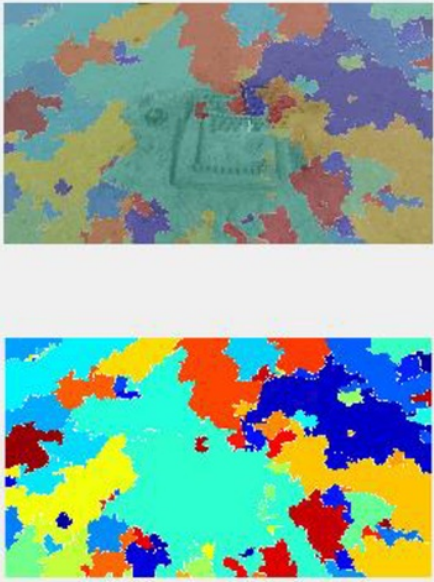
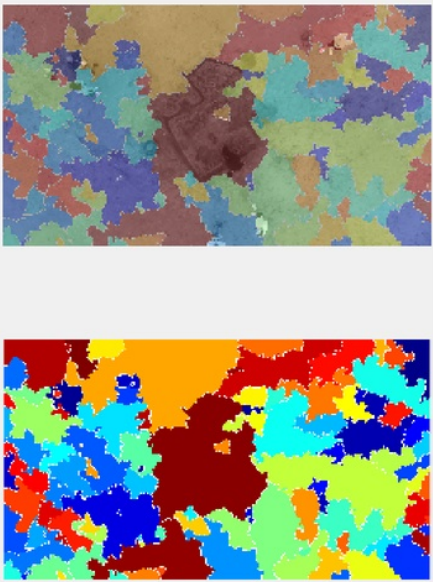
Before Event	After Event
	

Table 2: Segmentation with StrelSize adjusted to 10.

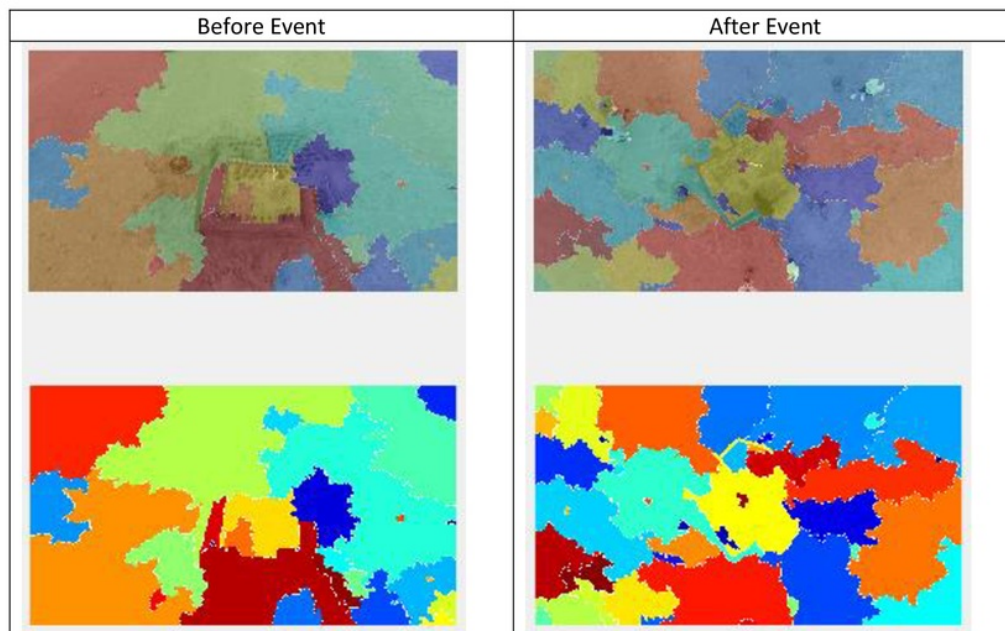


Table 3: Segmentation with OpeningdiscSize adjusted to 6.

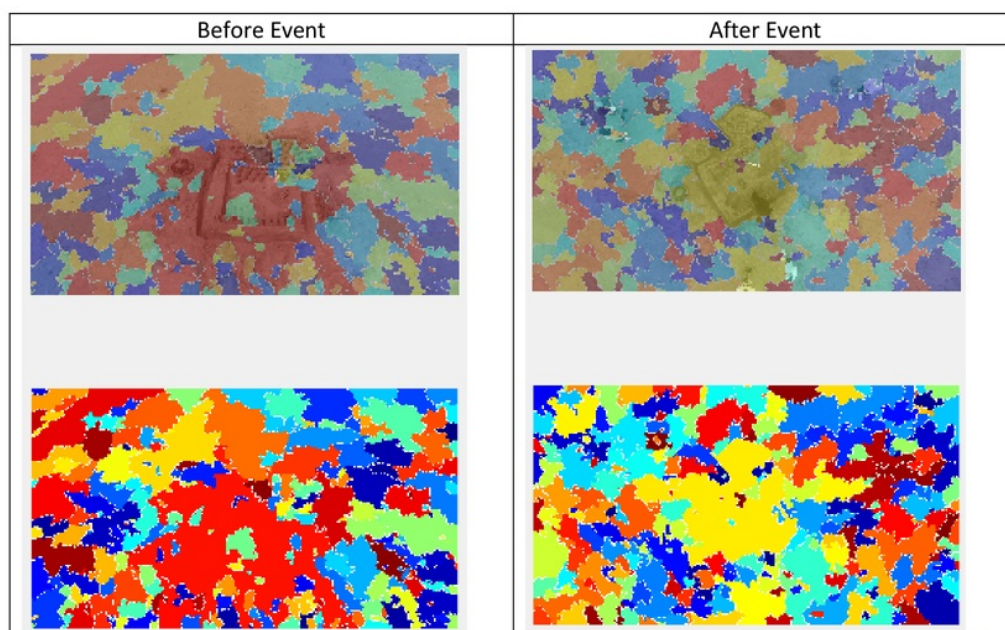


Table 4: Segmentation with dilateSize adjusted to 6.

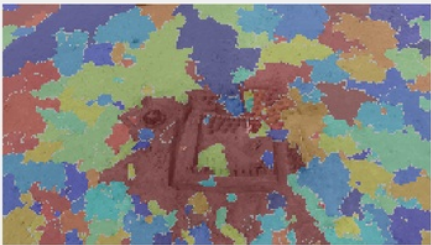
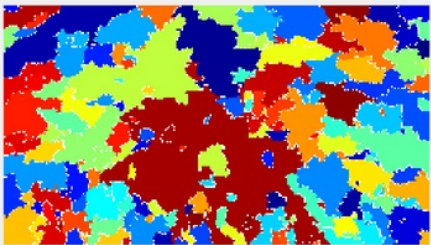
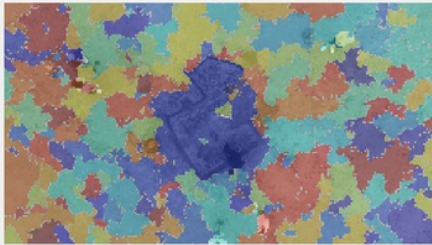
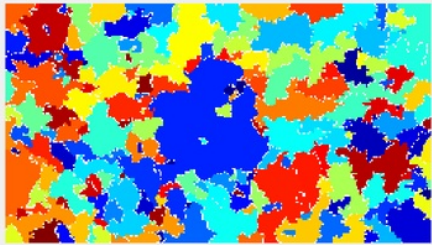
Before Event	After Event
 	 

Table 5: Segmentation with numPixels adjusted to 20.

7 IMPACT

As can be seen from the above comparisons, with the drone not being directly placed in the same position for the before and after photographs, no matter what segmentation parameter is used, comparison of the images is extremely difficult.

OpeningdiscSize provides perhaps the best adjustment for comparison with numPixels being the worst parameter to adjust for comparison.

The provision of the images provided can show in conclusion that for pixel by pixel comparison, the before and after images need to be identical. If these are not identical (as with the drone imagery), then it would be difficult for any form of digital comparison tool to provide a successful outcome. This is because during the comparison, the individual pixels within the image are compared for lightness and contrast. It is the differences in the pixels that show as different colours within the comparison.

The timing of capturing both sets of images for comparison is also important. For example, should the lighting of the subject contained within the image be different then the comparative result will be affected as the comparison will notice differences in the pixels contained within the image.

Returning to the original images used for comparison, as can be seen, the system of segmentation can be used to compare to images for differences. Should a neutral background be applied with these images overlaid, then digital comparison would be greatly enhanced.

8 REFERENCES

- [1] Remote Sensing - Models and Methods for Image Processing (3rd Edition), Schowengerdt. R, Elsevier, 2007, ISBN: 978-0-12-369407-2.
- [2] Digital Image Analysis for Evidence: A MATLAB toolbox, Welford. S, Gibson. S, Payne. A, Digital Image Analysis and Evaluation (DIAnE): A Forensic Image Processing Tool using MATLAB. In: The 5th Cybercrime Forensics Education & Training Conference, 1st - 2nd September 2011, Canterbury Christchurch University, Canterbury, Kent.
- [3] "Computer Vision", Shapiro. L, Stockman. G, New Jersey, Prentice-Hall, 2001, ISBN: 0-13-030796-3.
- [4] MathWorks – Strel: https://uk.mathworks.com/help/images/ref/strel.html?s_tid=srchtitle – accessed 15:00 on 11th July 2018.
- [5] MathWorks – Opening: https://uk.mathworks.com/help/vision/ref/opening.html?s_tid=doc_ta – accessed 08:00 on 12th July 2018.
- [6] MathWorks – imdilate: <https://uk.mathworks.com/help/images/ref/imdilate.html> - accessed 09:00 on 12th July 2018.
- [7] MathWorks - how to code the total pixels number and white pixels and black area pixels number: <https://uk.mathworks.com/matlabcentral/answers/319328-how-to-code-the-total-pixels-number-and-white-pixels-and-black-area-pixels-number> - accessed 10:00 on 12th July 2018.

9 BIBLIOGRAPHY

- [1] Datums and Map Projections for Remote Sensing, GIS and Surveying (2nd Edition), Iliffe. J, Lott. R, Whittles Publishing, 2008, ISBN: 978-1-904445-47-0.
- [2] Graphics and GUIs with MATLAB, Third Edition, Marchand. P., Holland. O, Chapman & Hall CRC, Boca Raton, 2003.
- [3] Essential MATLAB for Engineers and Scientists, Third Edition, Hahn. B, Valentine. D, Elsevier Ltd, Oxford, 2007.
- [4] Building Outline Extraction Using a Heuristic Approach Based on Generalization of Line Segments, Partovi. T, Bahmanyar. R, Krauß. T, Reinartz. P, IEEE: [https://elib.dlr.de/107332/1/final_manuscript%20\(2\).pdf](https://elib.dlr.de/107332/1/final_manuscript%20(2).pdf) – accessed 15:00 on 11th July 2018.
- [5] Background Removal Panel: http://mib.helsinki.fi/help/main/ug_panel_bg_removal.html - accessed 08:00 on 12th July 2018.
- [6] https://uk.mathworks.com/?s_tid=gn_logo – last accessed 10:00 on 12th July 2018.
- [7] MathWorks - how to extract the value pixel valu from image? <https://uk.mathworks.com/matlabcentral/answers/151779-how-to-extract-the-value-pixel-valu-from-image> - accessed 09:30 on 12th July 2018.
- [8] MathWorks - counting pixel of an image? <https://uk.mathworks.com/matlabcentral/answers/30501-counting-pixel-of-an-image> - accessed 09:45 on 12th July 2018.
- [9] MathWorks - how to extract the value pixel valu from image? <https://uk.mathworks.com/matlabcentral/answers/151779-how-to-extract-the-value-pixel-valu-from-image> - accessed 09:30 on 12th July 2018.

10APPENDIX 1

10.1MATLAB SCRIPT

```
StrelSize = 4;
OpeningdiscSize = 3;
dilateSize = 13;
numPixels = 30;

I = imread('Image_Name');
I = imresize(rgb2gray(I),0.25);
I = imadjust(I,stretchlim(I),[]);
--to lock aspect ratio
--makes picture go black and white.

--First image will have "Y" filter, 2nd image will have "X" filter....
--Pass through of vertical filter:
hy = fspecial('sobel');
hx = hy;
IY = imfilter (double(I), hy, 'replicate');

--Command for "X" filter;
IX = imfilter (double(I), hx, 'replicate');

--It is now possible to combine the "Y" and "X" filtered images as they are 90° out of phase....
gradmag = sqrt(IX.^2 + IY.^2);

--Now to mark the foreground markers (using opening by reconstruction then closing by
reconstruction to clean up the image – this creates flat maxima inside each object):
se = strel('disk', OpeningdiscSize);
lo=imopen(I,se);
--this applies our strel object to our original black and white image.

--It is possible to carry out further reconstruction:
le = imerode (I,se);
lobr = imreconstruct (le, I);

--Now it is possible to close the image...
loc = imclose (I,se);
-- and combine the results together:
lobrcbr = imreconstruct(imcomplement(lobrd),imcomplement(lobr));
lobrcbr = imcomplement (lobrcbr);
```

--It is now possible to calculate regional maxima from our images to obtain foreground markers....

```
fgm = imregionalmax(lobrcbr);
```

```
I2 = I;
```

```
I2(fgm)=255;
```

--carry out a close and an erode to clean up the edges....

```
se2 = strel (ones(StrelSize,StrelSize));
```

```
fgm2 = imclose (fgm,se2);
```

```
fgm3 = imerode (fgm2, se2);
```

--Superimpose back onto original image....

```
fgm4 = bwareaopen(fgm3,numPixels);
```

```
I3 = I;
```

```
I3(fgm4) = 255;
```

-- Now we need background markers....

```
bw = imbinarize(lobrcbr);
```

```
D = bwdist(bw);
```

```
DL = watershed(D);
```

```
bgm = DL == 0;
```

--Re-combine images.

```
gradmag2 = imimposemin(gradmag, bgm | fgm4);
```

```
L = watershed(gradmag2);
```

--And re-size.

```
I4 = I;
```

```
I4(imdilate(L == 0, ones(dilateSize, dilateSize)) | bgm | fgm4) = 255;
```

```
Lrgb = label2rgb(L, 'jet', 'w', 'shuffle');
```

--Now to show the final images:

```
figure
```

```
subplot(2,1,1);
```

```
imshow(I)
```

```
hold on
```

```
himage = imshow(Lrgb);
```

```
himage.AlphaData = 0.3;
```

```
%title('Lrgb superimposed transparently on original image')
```

```
subplot(2,1,2);
```

```
imshow(Lrgb);
```

FINAL GRADE

GENERAL COMMENTS

Instructor

59/100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAUL AUSTIN

58.95

RESEARCH-INFORMED LITERATURE



A very good range of reference material cited in your report

6.5

KNOWLEDGE AND UNDERSTANDING



knowledge is accurate but more information on the segmentation process would have been good

12.75

ANALYSIS



Evidence of logical thinking in your analysis. Good discussion on segmentation although I think you may have missed the point that after segmentation each segment within the image can be indexed and isolated for processing.

22.4

PRACTICAL APPLICATION AND DEP.



Very good application of the methods. It was nice to see an image with on change and no shift in

10.8

perspective to test the algorithm fairly.

SKILLS FOR PROFESSIONAL PRAC.



On the whole a well presented report with all the required information in a coherent and well structured manner.

6.5