

Part 2 Remote Sensing & Imaging

by Ralph Whittaker

FILE	5091_RALPH_WHITTAKER_PART_2_REMOTE_SENSING__IMAGING_116677_1793751976.DOCX		
TIME SUBMITTED	28-JUN-2018 06:49PM (UTC+0100)	WORD COUNT	3028
SUBMISSION ID	90998321	CHARACTER COUNT	17475



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

Remote Sensing & Imaging

PART 2

by

Ralph Whittaker

Student ID: P148322

Module: **M6X00945**

Lecturer: Dr Nik Whitehead & Dr Stephen Mosey

BEng (Hons) Test Range Engineering

June 2018

TABLE OF CONTENTS

SECTION	PAGE
Title Page	1
Table of Contents	2
1. Introduction	3
2. Image Segmentation	3-11
3. Factors that affect the accuracy of watershed segmentation	11-12
4. Determining the impact of an event	12
References	13
Bibliography	13

Table of Figures

- Fig.1 Original image, before destruction (Frame from 18.MOV at 05:10)
- Fig.2 Original image, after destruction (Frame from 19.MOV at 00:47)
- Fig.3 Greyscale image, before destruction
- Fig.4 Greyscale image, after destruction
- Fig.5 Segmentation function code
- Fig.6 Image foreground markers of the area before destruction
- Fig.7 Image foreground markers of the area after destruction
- Fig.8 Segmentation function code
- Fig.9 Image background markers of the area before destruction
- Fig.10 Image background markers of the area after destruction
- Fig.11 Image showing colour watershed matrix labelling of area before destruction
- Fig.12 Image showing colour watershed matrix labelling of area after destruction
- Fig.13 Image showing transparent pseudo-colour labelling matrix of area before destruction
- Fig.14 Image showing transparent pseudo-colour labelling matrix of area after destruction

1. Introduction

Using video footage captured from a drone flying over a sandcastle fort at the beach, the following report documents the computerised process involved in segmenting two still digital images of the area, before and after destruction, to identify changes to regions of interest. The methods used for digital image segmentation and processing are discussed, including the factors that affect the accuracy of the computerised processes used. The report will conclude with a discussion of how the two images could be used together to determine the impact the event has had on the targets in the images.

2. Image segmentation

Digital image segmentation can be used to divide digital images into meaningful structures for computerised image analysis. Similar to the human mind and vision, computerised algorithms can be performed on digital images to define their features (size, shape, or colour). Computer software, in various platforms, can use the defined pattern data to identify objects within images or videos and programmed conditional decisions can be made. Object recognition is a key output of machine programmed algorithms and is a vital computer vision technique.

"When humans look at a photograph or watch a video, we can readily spot people, objects, scenes and visual details. The goal is to teach a computer to do what comes naturally to humans: to gain a level of understanding of what an image contains" [1].

This can be achieved through object recognition without the need for storing thousands of comparable images.

The main use of segmentation is to partition an image into regions of either grey level, red-green-blue (rgb), different textures, depths or motion to enable digital image analysts to understand either object identification, for making object measurements, or for identifying changes to topography over time. Each pixel within an image is assigned a label, meaning that pixels with the same identification have common visual characteristics.

From viewing the separate video footage taken by the drone, before and after destruction of the sandcastles (using remote controlled tanks), it was possible to identify two images (single frames) of the same features, from the same perspective, (Fig.1 and Fig.2), ready for use by the script in the software MATLAB.



Fig.1 Original image, before destruction (Frame from 18.MOV at 05:10)



Fig.2 Original image, after destruction (Frame from 19.MOV at 00:47)

For this task, marker-controlled watershed segmentation was used to separate the objects in the images. The watershed transform “finds catchment basins and watershed ridge lines in an image by treating it as a surface where light pixels are high and dark pixels are low” [2].

“Morphological watersheds provide a complementary approach to the segmentation of objects. It is especially useful for segmenting objects that are touching one another” [3].

The software MATALAB was used to perform the task and the following describes the steps of the process applied to the separate images by entering code into the command window.

After loading the video files (18.MOV and 19.MOV), a predefined grab video frame function and a predefined (computed) segmentation function into Matlab, the video frames of selected instances could be read by Matlab and the script was ready for segmenting the images. Upon opening the *GrabVideoFrame* function, the image frame details were input, on separate occasions, for the image to be worked on. Commands were written in the command window to obtain and view these individual frames from the video footage at specific time intervals (47 and 310 seconds).

```
figure; imshow(GrabVideoFrame('DJI_0019.MOV',47));
```

```
figure; imshow(GrabVideoFrame('DJI_0018.MOV',310));
```

The images could have been rotated at this stage using image rotation code, but it was not necessary for this task since the targets were viewed from vertical perspective.

Next, the images were read as rgb (pixel colours) and then converted to greyscale and displayed (Fig.3 and 4) using the following code;

```
image(rgb); (displays rgb image)
```

```
I = rgb2gray(rgb); (turns image from rgb to greyscale)
```

```
imshow(I); (displays image)
```

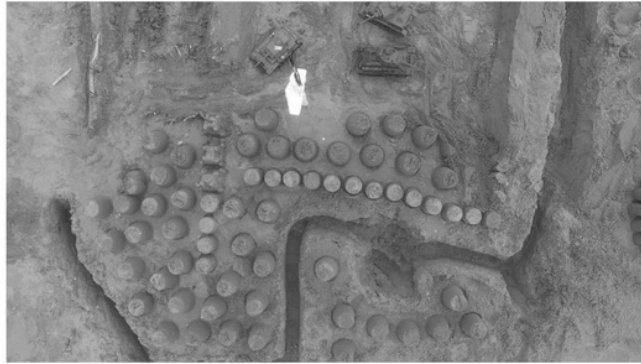


Fig.3 Greyscale image, before destruction



Fig.4 Greyscale image, after destruction

Viewing the greyscale image as a topological surface, allows watershed ridge lines and catchment basins to be created, where values of $f(X, Y)$ correspond to heights. This can be understood as water collecting in one of the two catchment basins to flood the basins from below (holes) or as water falling on the watershed ridge line (one pixel thick dam which separates the two basins), equally flowing into either of the catchment basins. Flooding continues until the entire image is segmented into separate catchment basins, divided by the watershed ridge lines. The algorithm plots the location of the catchment basins and ridge lines in the image. The image (f) has positive integer values at different regions and zero at the watershed ridge lines. A distance transform, conducted on a binary image, coordinates the distance from every pixel to the nearest nonzero valued pixel. Therefore, the algorithm converts the original image into a new image with catchment basins representing the objects to be identified.

Using the *strel* function within the segmentation function code, image parameters can be tweaked to remove certain elements, such as noise, by changing the *strel* radius size (disk shape structuring element) or opening disc size (number of line structuring elements used to approximate the disk shape). At the centre of a structuring element is the origin pixel which identifies the pixel image being processed.

"A strel object represents a flat morphological structuring element, which....is a binary valued neighbourhood, either 2D or multidimensional, in which true pixels are included in the morphological computation, and the false pixels are not" [4].

Using this morphological transformation within the segmentation function, the background features of the image were subtracted, removing all features smaller than the structuring element. This removes the noise from the image and creates a flat disc shape structuring element.

Any features smaller than the disk size of the structural element size, set in the segmentation function, can be removed by changing the disk and strel size in the segmentation function (Fig.5).

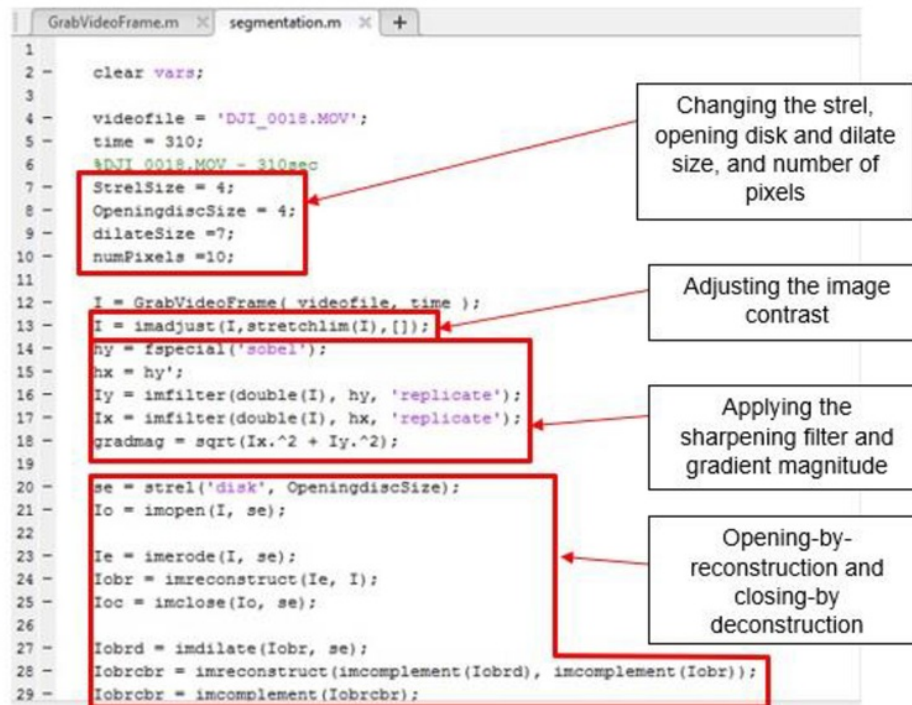


Fig.5 Segmentation function code

Once the noise had been removed, the image was adjusted by stretching the contrast (intensity values) of the image (normalisation) to span a desired range of values. This enhances the contrast to the full range of pixel values that the image (histogram) allows (Fig.5).

Using a *Sobel edge mask* and an *imfilter* (Fig.5), a filter was applied to the image in both X and Y axis, this plotted the X and Y sharpe edges separately. This function uses arithmetic to compute the gradient magnitude (*gradmag*) so that it is high at the borders of objects and mostly low inside the objects. The two were then layered on top of each other.

The foreground objects were then marked in the images by locating groups of pixels inside each of the foreground objects. Using *opening-by-reconstruction* (erosion followed by morphological reconstruction) and *closing-by deconstruction* (*imopen*, *imerode*, *imreconstruct*, *imclose*, Fig.5) the image was *cleaned-up*. This process removed dark spots in the images.

The functions *imdilate* and *imreconstruct* were used and the image inputs and outputs of *imreconstruct* were complimented. This removed small blemishes from the images without affecting the overall shapes of the objects.

The regional maxima (*lobrcbr*) was calculated to produce good foreground markers (Fig.8). The following code was used in the function, after the *fgm* function, to display the isolated foreground markers (Fig.6 and 7);

Figure;
Imshow(fgm);



Fig.6 Image foreground markers of the area before destruction



Fig.7 Image foreground markers of the area after destruction

Above, the foreground markers were superimposed onto the original image. The edges of the markers were then cleaned up by changing the strel size at the top of the function and using a closing followed by an erosion.

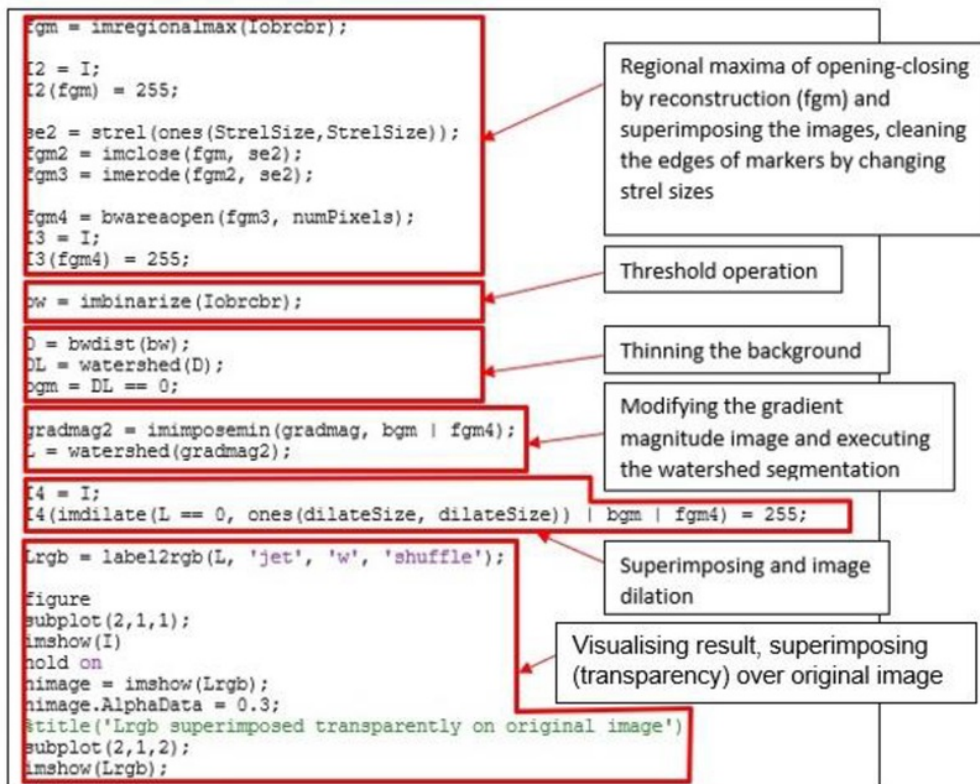


Fig.8 Segmentation function code

To mark the background in the cleaned-up image, background markers (Fig.9 and 10) were created using a thresholding operation (dark pixels belonged to the background). The following code was inserted after the *bgm* function, to display the background markers created by this part of the segmentation function;

Figure;
imshow(bgm);

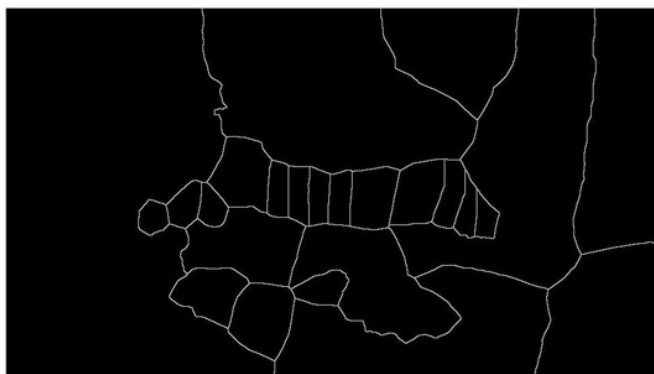


Fig.9 Image background markers of the area before destruction

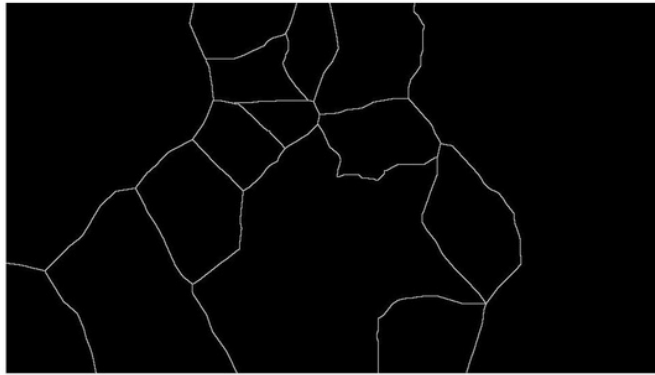


Fig.10 Image background markers of the area after destruction

To prevent the background markers being too close to the edges of the objects being segmented, the background was thinned using the “*skeleton by influences zones or SKIZ, of the foreground of bw...by computing the watershed transform of the distance transformed of bw and looking for the watershed ridge lines (DL == 0) of the result*” [1].

So that the image had regional minima only at the foreground and background marker pixels, the function *imimposemin* was used to modify the gradient magnitude image. The watershed-based segmentation was then performed.

Finally, the result was visualised by using image dilation (*imdilate*) and superimposing the foreground markers, background markers and segmented object boundaries onto the original image. Dilation was adjusted to make the object boundaries (where $L == 0$) stand out. A better visualisation showing how foreground and background markers affect the result was created using a coloured watershed label (Lrgb) matrix (Fig.11 and 12). Here the *watershed* and *bwlabel* were converted to true colour images using *label2rgb*.

Fig.13 and 14 shows how transparency was used to superimpose a pseudo-colour label matrix on top of the original intensity image. These images made it easier for interpretation.

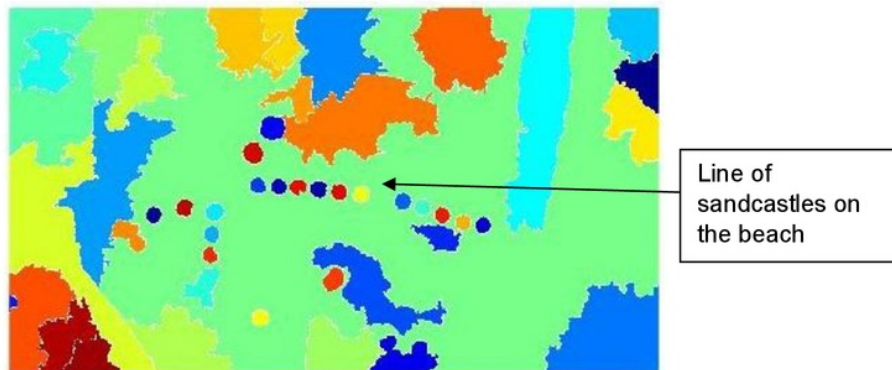


Fig.11 Image showing colour watershed matrix labelling of area before destruction

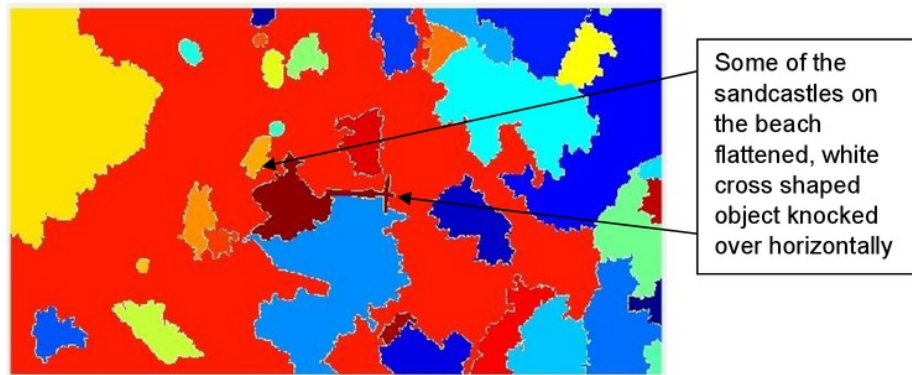


Fig.12 Image showing colour watershed matrix labelling of area after destruction

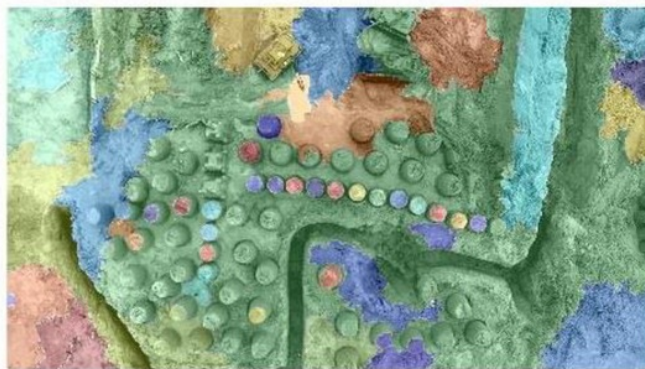


Fig.13 Image showing transparent pseudo-colour labelling matrix of area before destruction

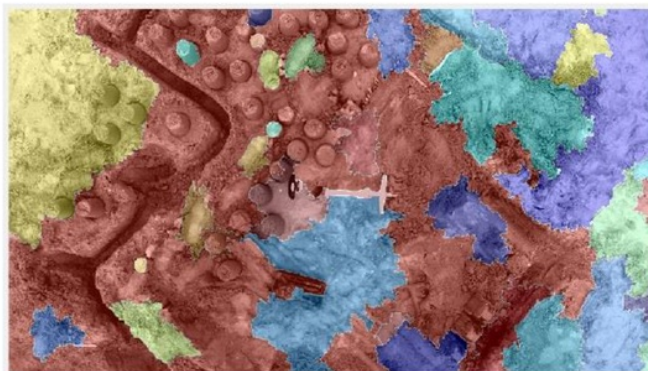


Fig.14 Image showing transparent pseudo-colour labelling matrix of area after destruction

The *strelsize*, *OpeningdiscSize*, *dilatesize* and *numPixels* (number of pixels) were adjusted through trial and error (Fig.5) to create the most distinct regional shapes for object recognition and for identifying the maximum number of valid whole features within the images.

The final filter settings for the before and after images are as follows;

StrelSize = 4

OpeningdiscSize = 4

dilateSize = 7

numPixels = 10

The destruction of the individual sandcastles can clearly be seen from the difference in the individual segment shapes visualised on the separate images. The process has successfully partially segmented the images to reveal the damage done to the sandcastles by the remote-control tanks. Since the task used images from the beach, the sandcastles made from sand were naturally in an environment surrounded by sand (of the same texture, lighting and shading) of which was completely disturbed during the making of the sandcastles. Therefore, it proved difficult to find filter values to isolate the sandcastle from their (same material) background and multiple elevated surface shaped (light and shaded) ground regions.

3. Factors that affect the accuracy of watershed segmentation

The factors that affect the accuracy of the machine vision processes used include the resolution (pixel array) that the input images were recorded in and the original sharpness (acutance) or contrast (along edge of objects) within the image. The higher the resolution and contrast of features within an image, the better result of segmentation. The environment in which the target objects are within also affects the ability of computing the visual and segmental differences within an image. If the background textures and contrasts are very similar to the objects under investigation, it becomes very difficult to distinguish between the two. The lighting and shading present within an original image plays a vital role in providing the true and false edges required for image processing. If the image is too bright or too dark (environment lit poorly or highly unevenly), the segmentation result will not be as accurate as preferred by the user or platform in which the data is to be used. Spatial arrangement of the objects within an image also plays an important role in gaining a successful segmentation.

If objects are very close together or even touching, it becomes difficult to differentiate between separate objects and this can cause problems in systems that use image recognition software for object detection.

For comparison of two images after an event has occurred, it is important to ensure that the pictures are taken from the same perspective, usually vertically above a scene. The slightly different overhead perspectives taken by the drone of the same area before and after destruction, had a slight negative affect on the post segmentation image analysis. Ideally the images (for topography analysis) needed to have been taken from the exact same height, perspective and orientation to make segmentation and analysis easier and more accurate.

It is important that segmentation respects object boundaries and doesn't break up the objects of interest. The filtering values used in the segmentation function have an impact on segmentation and it is important to choose the correct strel shape and size during segmentation, otherwise object (outlines) can become lost due to over filtering the image. Not enough filtering, and segmentation doesn't separate the image regions enough for object isolation and recognition.

It was found during this task that using an image with fewer larger objects (by selecting close-up frames) of the area and objects in question, the segmentation results were better than when a wider image from higher altitude was used. This made it easier to choose the ideal filter parameters without losing too many whole targets (sandcastles) from the image. Filtering was also made easier since the targets were all the same shape. Filtering an image containing targets of varying shapes can prove difficult to retain all the boundaries and regions of objects.

4. Determining the impact of an event

The two images, of before and after, could be used to determine the impact that the event had on the targets (sandcastles) by comparing their segmented regions and visually spotting the difference between the two. Furthermore, the images could now be compared using an image difference (*imdiffer*) function from the Matlab image processing toolbox and the changes to the area could be presented. This could allow for shape and spatial calculations to be made using mathematical algorithms, classifying the destructive impact that the event had on the spread (displacement) of the sandcastles. By comparing the pixel categories (of high sand and low or flat sand), a percentage error between the two images could be obtained. A calculated area of flattened sandcastles, could then be given, representing the level of destruction. Simply visualising the resulting segments of such destructive events can have a meaningful impact on memory, learning and the ongoing perception of an area or objects. The way that images can be computed and fragmented, allows for information about a scene or target to be identified in a more detailed and revealing way that might not be recognised by initial human visual interpretation. Image data can also be stored for ongoing future differential comparison.

After segmentation, the images could be further processed in Matlab using a connecting component algorithm, to label or mark the segment regions (1, 2, 3, ... n) to identify the individual regions that represent similar clusters of pixels. Objects can be recognised using pattern recognition techniques. A feature extraction technique can be applied to each segment separately to provide a feature vector for each segment. If the segmented images are binary images (pixel values that represent a segmented object), a connected component analysis could be performed to obtain an object representation that can be used in other applications.

 **Boarder** analysis could be performed on edges to complete and filter images, providing the exact body data of the objects, filling out to the boarder, to be obtained. A classifier could then be used on the images to classify the segments into separate categories. Images could then be used for object extraction or object detection using object identification software. This could allow for removal of the targets or background or for interactive object identification systems. Examples of platforms that utilise object recognition and object detection include driverless cars, security scanners, disease identification, industrial inspection, robotic vision and for building maps. A topographical use of the segmentation technique could be for comparing historical satellite images of the melting polar ice caps. By subtracting one image from another, a resultant image could be formed to represent the impact climate change is having on the polar region by depicting the total amount of ice (or thickness levels) that has melted over time.

References

- [1] MathWorks, What Is Object Recognition, 2018 [Online] Available: <https://uk.mathworks.com/solutions/deep-learning/object-recognition.html> [Accessed: 24/06/18]
- [2] MathWorks, Marker-Controlled Watershed Segmentation, 2018 [Online] Available: <https://uk.mathworks.com/help/images/examples/marker-controlled-watershed-segmentation.html?prodcode=IP&language=en> [Accessed: 24/06/18]
- [3] Sukhdeep Kaur, Manjit Sandhu, Jaipreet Kaur, Analysis of Various Image Segmentation Techniques Using MATLAB, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 6, Issue 4, April 2016, pg.90 [Online PDF] Available: http://ijarcsse.com/Before_August_2017/docs/papers/Volume_6/4_April2016/V6I4-0145.pdf [Accessed: 24/06/18]
- [4] MathWorks, strel, description, 2018 [Online] Available: <https://uk.mathworks.com/help/images/ref/strel.html> [Accessed: 25/06/18]

Bibliography

- [1] Robert Fisher, Simon Perkins, Ashley Walker, Erik Wolfart, Contrast Stretching. Image Processing Learning Resources, 2004 [Online] Available: <https://homepages.inf.ed.ac.uk/rbf/HIPR2/stretch.htm>
- [2] Sukhdeep Kaur, Manjit Sandhu, Jaipreet Kaur, Analysis of Various Image Segmentation Techniques Using MATLAB, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 6, Issue 4, April 2016, pg. 90-94[Online PDF] Available: http://ijarcsse.com/Before_August_2017/docs/papers/Volume_6/4_April2016/V6I4-0145.pdf [Accessed: 24/06/18]
- [3] ResearchGate, How can we find the objects from segmented images? 2015 [Online] Available: https://www.researchgate.net/post/How_can_we_find_the_objects_from_segmented_images [Accessed: 24/06/18]

Part 2 Remote Sensing & Imaging

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

Instructor

58/100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12



Comment 1

Border?

PAGE 13

RALPH WHITTAKER

57.60

RESEARCH-INFORMED LITERATURE



Although you have several references, it would have been nice to see more sources used.

4.8

KNOWLEDGE AND UNDERSTANDING



You have demonstrated a good level of knowledge of the methods employed. Some of your descriptions could have been expanded slightly.

13.75

ANALYSIS



Very good discussion on the factors affecting the success of segmentation.

24.8

PRACTICAL APPLICATION AND DEP.



Very good application of the segmentation methods taught. Nice selection of images.

9.75

SKILLS FOR PROFESSIONAL PRAC.

Mostly well written, but could have been structured



better.

4.5