Where are my sheep?

CS39440 Major Project Report

Author: Nathan Williams (naw21@aber.ac.uk)
Supervisor: Dr Frederic Labrosse (ffl@aber.ac.uk)

May 7, 2020
Version: 1.0 (Draft)

This report was submitted as partial fulfilment of a MEng degree in Software Engineering (G601)

Department of Computer Science
Aberystwyth University
Aberystwyth
Ceredigion
SY23 3DB
Wales, U.K.

Declaration of originality

I confirm that:

- This submission is my own work, except where clearly indicated.
- I understand that there are severe penalties for Unacceptable Academic Practice, which can lead to loss of marks or even the withholding of a degree.
- I have read the regulations on Unacceptable Academic Practice from the University's Academic Registry (AR) and the relevant sections of the current Student Handbook of the Department of Computer Science.
- In submitting this work I understand and agree to abide by the University's regulations governing these issues.

Name .....

Date .....

Consent to share this work

By including my name below, I hereby agree to this project's report and technical work being made available to other students and academic staff of the Aberystwyth Computer Science Department.

Name .....

Date .....

Acknowledgements

I am grateful to...

I'd like to thank...

Abstract

Include an abstract for your project. This should be approximately 300 words.

The abstract is an overview of the work you have done. Highlight the purpose of the work and the key outcomes of the work.

Glossary

- Templating

  Locating where in a image a template best matches.

- Thresholding

  Taking all values of an image above a certain value and maximising them and minimising anything below.

- colour band

  A collection of samples of a certain colour from an image.

- aerial image

  An image taken of from a height such as from a drone, helicopter, plane or satellite looking down on a landscape.

- samples

  a collection of a colour samples taken from different colour bands such as red,green,blue,rededge or near IR taken from a certain spot of an image.

Contents

List of Figures

List of Tables

Chapter 1

Background & Objectives

Initial brief:

> Research projects in collaboration with biologists and farmers look at automatically detecting sheep from aerial imagery, in particular multi-spectral images. This can also applied to other animals such as deer. This project looks into using these images to detect and count sheep. A variety of methods can be explored.

1.1 Background

The aim of this project is to investigate ways to locate and count the sheep in a set of aerial images of fields, some of which were taken on a red-edge camera to provide multi spectral images. At the start we will mainly look at white sheep, but sheep are not just white so we must investigate ways of identifying black and brown sheep also. I will also discuss how the techniques could be used to count and track other mammals such as deer. The purpose of identifying the animals from the aerial images is that it allows biologists to observe them without disturbing them such as attaching GPS trackers to them individually. The images were originally used by biologists to identify and track the flora but using the same data to track fauna would also advantageous.

This will lead to a discussion on how the techniques could be used to count and track other mammals such as deer. The purpose of identifying the animals from the aerial images is that it allows biologists to observe them without disturbing them such as attaching GPS trackers to them individually. The images were originally used by biologists to identify and track the flora but using the same data to track fauna would also advantageous.

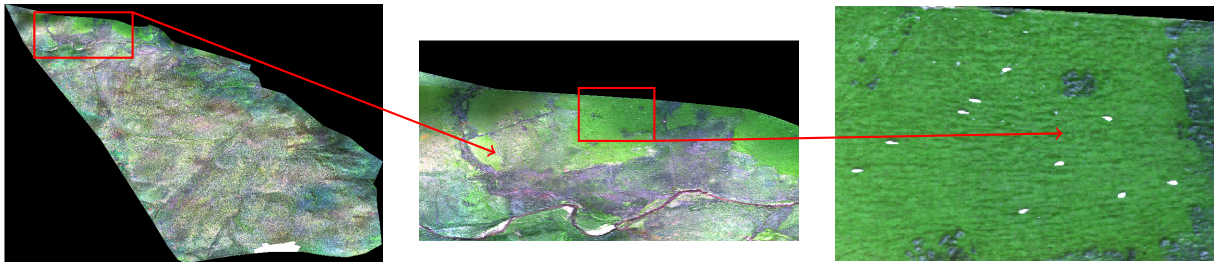


Figure 1.1: Manually locating the sheep

Locating and counting the sheep can obviously be done manually as seen in figure 1.1 but this would be tedious and time consuming as the images can be very large and the sheep few and far between with each sheep only covering around 20 pixels in size. As the images are at a resolution of approximately 8 cm per pixel if the camera is 120m above the landscape. [**?**]

this is the first method, I found some existing research about it.... This is the good things about the first method... This is the bad things about the first method.... on the whole the first method would be useful in this scenario..

this is the second method, I found some existing research on it This is the good things about the second method... This is the bad things about the second method.... on the whole the second method would be useful in this scenario..

this is the third method i found some existing research on it... This is the good things about the third method... This is the bad things about the third method.... on the whole the third method would be useful in this scenario..

This method would be the most superior method to use in comparison to the other methods because....I hope to investigate applying this to the project and proving/disproving my ideas....

1.1.1 rededge and near IR

red edge used for vegetation detection as there is a region of rapid change in reflectance just outside the red range of the electromagnetic spectrum.

1.2 Analysis

1.2.1 Input: images

1.2.1.1 Standard RGB images

Some of the digital images used in this project are stored as PNG and JPG, these images will usually contain 3 layers of pixels: one for each Red, Green and Blue. Each pixel

contains a brightness value for it's colour, normally between 0 and 255 for an 8-bit image. They are stored in a 3 dimensional array, with the first dimension being which layer of the image 0:red, 1:green and 2:blue then the next dimension is the y axis of the image and finally the x axis of the image, an example shape of an image could be: [3,500,400].

1.2.1.2 Red-edge Camera images

The other primary format used in this project are stored as geoTIFF files. These contain a set of metadata, such as a scale and GPS coordinates of the image, and a 2 dimensional array for the y and x axis of the image. In the array there is a set of colour samples, the number of samples can vary. In this case the project is using images from a RedEdge camera so we get 5 samples at each location. Each sample contains 12-bits of data, the first 3 samples being the standard blue, green, red, the remaining two are red edge and near-IR. This gives us extra data to work with.

Colour bands of a rededge camera:

- Blue

  475 nm center, 20 nm bandwidth
- Green

  560 nm center, 20 nm bandwidth
- Red

  668 nm center, 10 nm bandwidth
- rededge

  717 nm center, 10 nm bandwidth
- near-IR

  840 nm center, 40 nm bandwidth [**?**]

1.2.1.3 Opening and managing images

opencv only designed to handle standard RGB images so we need something else to handle the geoTIFF files.

An initial trial with PyLibTiff [**?**] which is wrapper for the libtiff library to Python using ctypes, resulted in issues with being able to read the geo tags in the files, and saving and transferring the data to the processed files.

Settled on using tifffle.py [**?**] as it was much simpler to use and had all the features I needed.

1.2.2 Processing

opencv pros and cons, limitations languages? python quick and easy scripting,

1.2.3 Output: Displaying results

powerful graphing with matplotlib flexible.

1.3 Research Method and Software Process

1.3.1 Agile Method

Scrumban lite

jira board roadmap backlog weekly sprints

Chapter 2

Experiment Methods

2.1 hypothesis

What is a good method to locate the sheep? Compare how accurate they are.

templating better than thresholding.

2.1.1 Thresholding

best colour band or combination of bands to use? best threshold value to use? look at colour bands together or seperately and combine results? Is it any good of a method what about none white animals?

The first method to be tried is using a threshold value to help us identify the sheep [**?**]. Taking one colour band or a merging a combination of bands together we can get a 2d array to work with.

In the example in fig.**??** you can see part of the blue band of an image appearing in grayscale on the left, where the brighter pixels of the image correspond to bigger value(255). The central plot shows this in a 3 dimensional graph. You can clearly see where the sheep stands out.

For the thresholding method we can take a value to threshold for example if the values are greater than 200 we set to 255 else to 0. Which produces something like the graph on the right. We can then use a simple search for the values of 255 and look at it's neighbouring high values to locate it's centre and estimate it's size.

2.1.2 template matching

best template shape? size? best colourband to use is it same as thresholding? Is it any good of a method what about none white animals?

The second method to be tried out is to use a template, either a sheep that we have

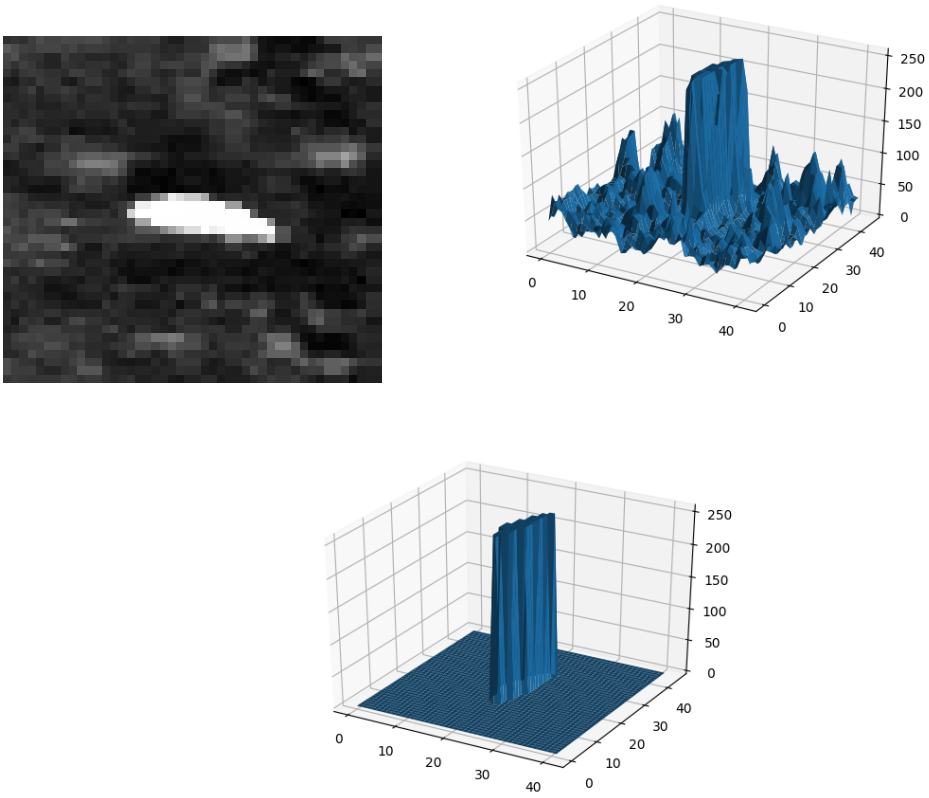
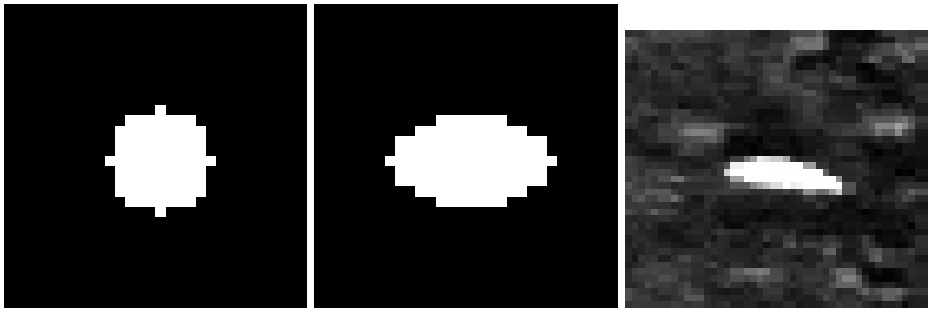


Figure 2.1: Caption

Figure 2.2: Template Examples



previously found or an artificial sheep shape to help match a more general sheep shape, ignoring rotation. See fig.2.2 for examples.

The template is then moved across the image and we calculate a metric and get a value to help measure how closely the area matches our template, an example result is shown in fig.4.2, you can see hot spots where the pattern matches most. We can then pick out the spots that have the highest value and these are likely our sheep.

2.1.3 other approaches considered

machine learning.

2.2 How measurements and comparisons of results are to be made

ROC graphs.

2.3 Support Tools

backend: Python3, opencv, matplotlib, numpy, tifffle. programming UI: object oriented python for ease of use in scripts to build graphs etc. demo UI: flask simple webserver, django too comlex for this task, bulma make it look nice with not much hasle, (discuus potential to package cross platform with electron).

design and implementation in Chapter 3

Chapter 3

Software Design, Implementation and Testing

3.1 Design

3.1.1 Overall Architecture

python modules, object oriented:

ImageManager to handle loading, saving and standardising images and running the methods. Discussed in detail in section 3.1.2

flask server to run UI and call ImageManager in user friendly way. Discussed in detail in section 3.1.3

finders, classes to implement the methods, flexible

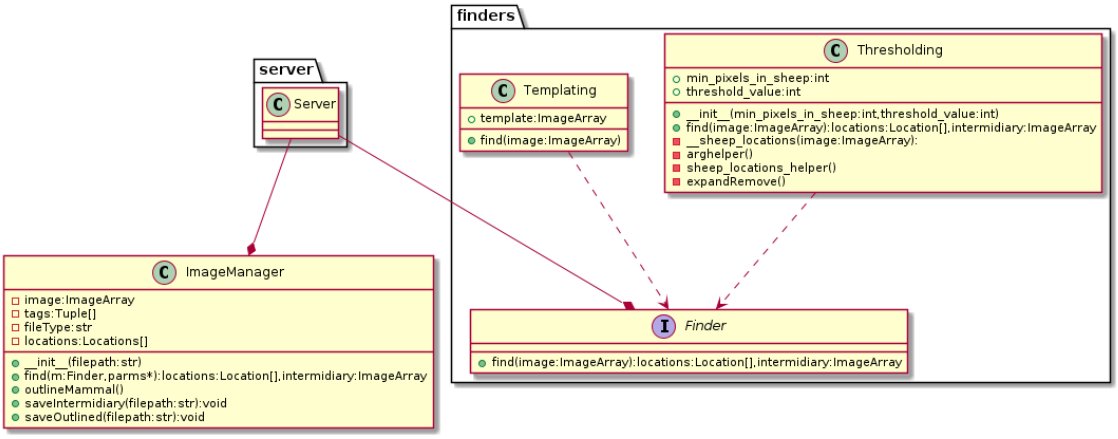


Figure 3.1: UML Class Diagram

3.1.2 Image Manager

3.1.2.1 Even more detail

3.1.3 User Interface

3.1.4 Finders

3.1.4.1 Templating

3.1.4.2 thresholding

3.2 Implementation

3.3 Testing

3.3.1 Overall Approach to Testing

Do little and often, while developing, write tests as I go, does this do what I want it to?

3.3.2 Automated Testing

Not using jenkis for automated testing on commit's alos python so nothing to build.

3.3.2.1 Unit Tests

done some of this but very little

3.3.2.2 User Interface Testing

not done this..

3.3.2.3 Stress Testing

Use biggest image I have available, with most complex set of parameters.

3.3.3 Integration Testing

3.3.4 User Testing

Chapter 4

Results and Conclusions

4.1 Thresholding Results

Manual adjustments of the threshold value is required on an image by image basis to minimise false positives and false negatives. The issue with this method though is the that it not only picks up the sheep it also picks up anything bright some examples in fig.**??**.

4.2 Templating Results

But there is have similar issues with the templating method as there was with thresholding, rocks match with the template. To overcome this the template can be adjusted, for example the template can be rotated or look at using different ways to calculate the metric. [**?**]

4.3 Method Comparison

4.3.1 Other methods discussion

4.4 Conclusion

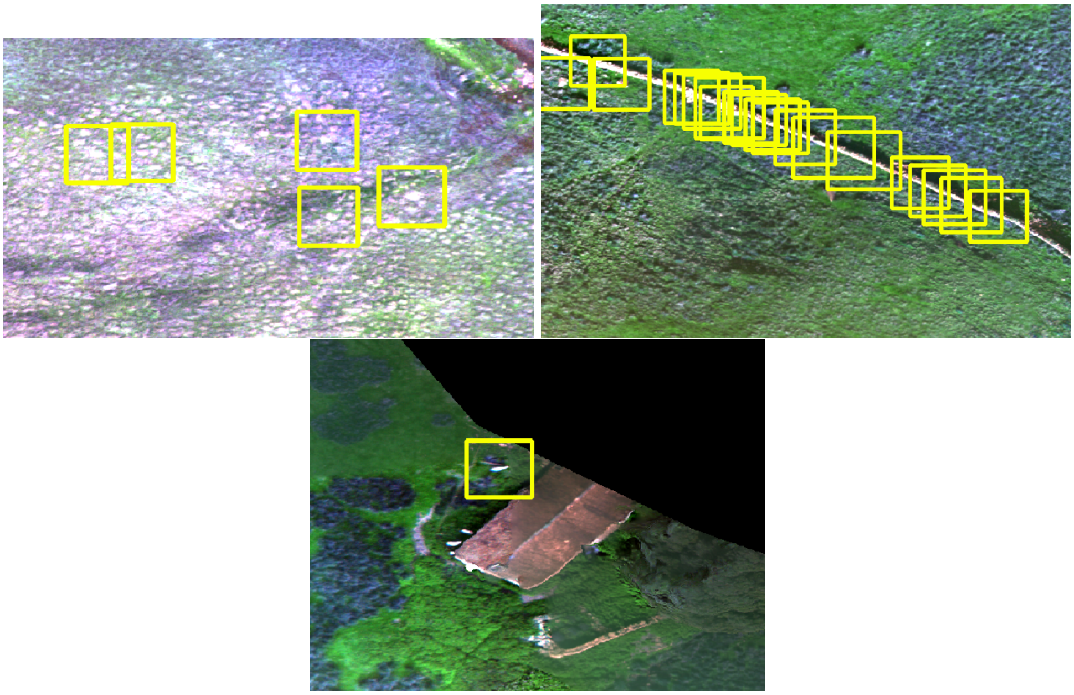
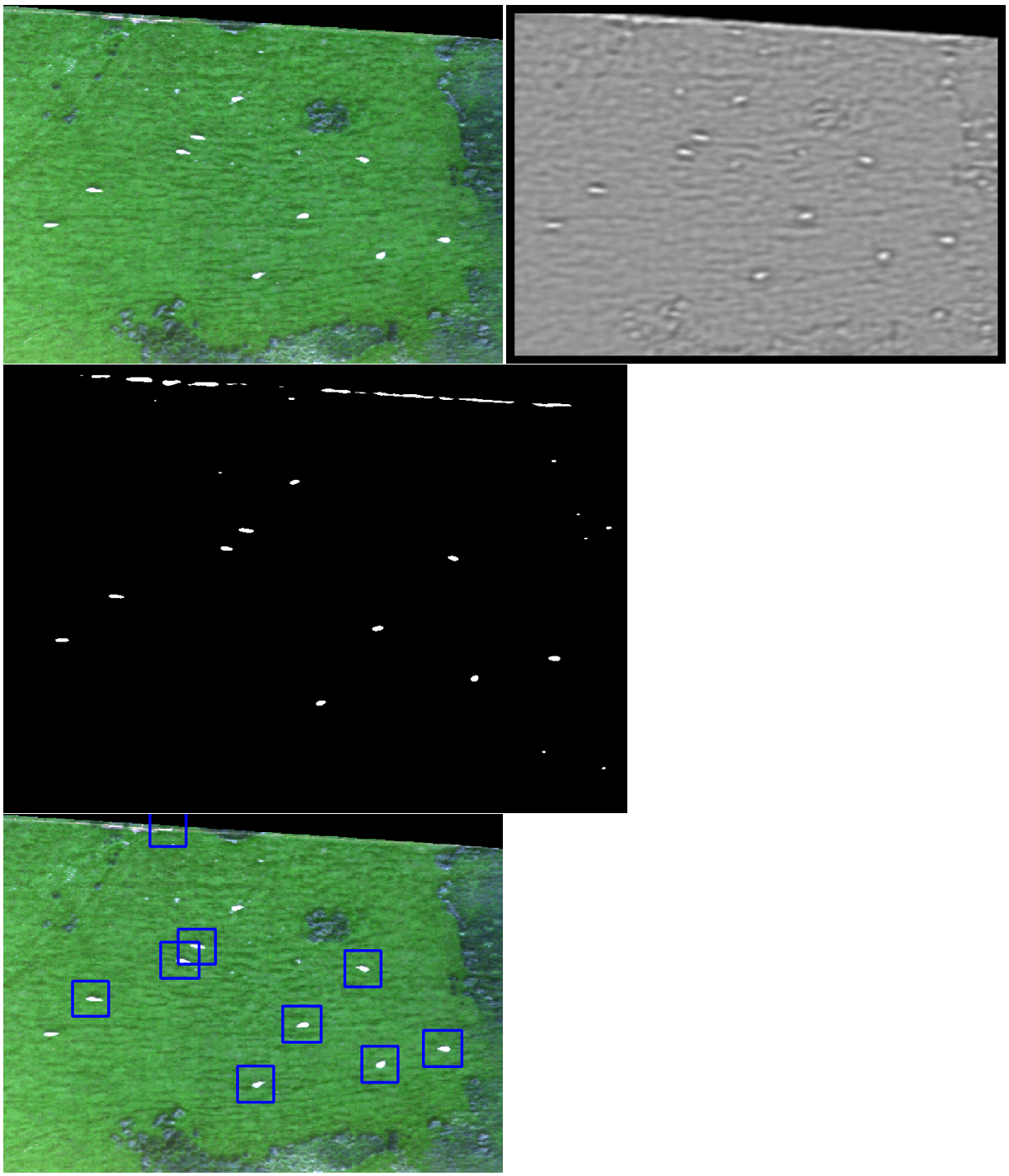


Figure 4.1: Caption

Figure 4.2: Templating Search



Chapter 5

Evaluation

5.1 Requirements

5.2 Design, Implementation and Tools

5.3 Project Aims

5.4 Self evaluation

What would I do differently? What did I do right? What did I do wrong?

Appendices

The appendices are for additional content that is useful to support the discussion in the report. It is material that is not necessarily needed in the body of the report, but its inclusion in the appendices makes it easy to access.

For example, if you have developed a Design Specification document as part of a plan-driven approach for the project, then it would be appropriate to include that document as an appendix. In the body of your report you would highlight the most interesting aspects of the design, referring your reader to the full specification for further detail.

If you have taken an agile approach to developing the project, then you may be less likely to have developed a full requirements specification. Perhaps you use stories to keep track of the functionality and the 'future conversations'. It might not be relevant to include all of those in the body of your report. Instead, you might include those in an appendix.

There is a balance to be struck between what is relevant to include in the body of your report and whether additional supporting evidence is appropriate in the appendices. Speak to your supervisor or the module coordinator if you have questions about this.

Appendix A

Third-Party Code and Libraries

Appendix B

Ethics Submission

This appendix includes a copy of the ethics submission for the project. After you have completed your Ethics submission, you will receive a PDF with a summary of the comments. That document should be embedded in this report, either as images, an embedded PDF or as copied text. The content should also include the Ethics Application Number that you receive.

Appendix C

Code Examples