Radiometric Correction of MultiSpectral Cameras Using Photosensor Irradiance for Agronomy Applications

Nicholas S. Mitchell, The University of Western Ontario

Supervisor: Sabarinathan, Jayshri, The University of Western Ontario
A thesis submitted in partial fulfillment of the requirements for the Master of Engineering Science degree in Electrical and Computer Engineering
© Nicholas S. Mitchell 2020

Follow this and additional works at: https://ir.lib.uwo.ca/etd

Part of the Electrical and Electronics Commons, Electromagnetics and Photonics Commons, and the Signal Processing Commons

Recommended Citation
https://ir.lib.uwo.ca/etd/7539
Abstract

Cloud coverage has a significant impact on the reflectance maps generated by multispectral cameras mounted on UAV’s in small farm settings. The current approach is to calibrate the camera once before every flight and use the radiometric calibration map for the entire flight. In this work we have designed and built a downwelling and upwelling photosensor to work in sync with our custom multispectral camera. A dual reflectance panel implementation for the time dependent radiometric calibration of the multispectral camera is completed. The solar spectral irradiance curve is approximated using the current measurements from the downwelling photosensor and the ground truth curve using a spectrometer. A multilinear regression algorithm was used for this purpose. The solar spectral irradiance curves are used during image acquisition to modify the initial radiometric calibration mapping function. This has potential to improve the usage of the multispectral camera under a wide range of weather conditions.

Keywords: Relative radiometric calibration, precision agriculture, remote sensing, multispectral camera, downwelling photosensor irradiance, radiometric correction
Lay Summary

Using multispectral cameras in farms fields is a rapid and accurate method of gathering the data needed to optimize crop growth. Catching diseases spread early can be derived from the data obtained from flying a multispectral camera mounted on an unmanned aerial vehicle over a farm field. The work here aims to provide a solution to one of the current drawbacks with multispectral cameras, the inability to capture accurate image data with change ambient light conditions due to cloud coverage.

This thesis focused on building a dual photosensor instrument which can be matched with a wide range of multispectral cameras in the visual and near infrared spectrum. One of the photosensors called the downwelling photosensor measures the sky with its lens such that it can correct the images from the multispectral camera. The dual photosensor instrument is integrated with a custom multispectral camera so it is built an interfacing microprocessor, which then sends the incoming light irradiance data to the main camera processor.

The multispectral camera is initially relatively radiometrically calibrated before flight using a dual reflectance panel method implemented in this thesis work. This technique allows for increased reflectivity measurement accuracy when imaging with the multispectral camera. The algorithm used to detect both panels used two different object recognition algorithms: the first assumed that the panels lay on a dark surface and the other did not. This work enabled efficient time series measurements of the reflectance panels needed to correct for the cloud coverage effects. Finally, the photosensor data was used to correct for the changing incoming irradiance due to the clouds. The first step was to model the incoming irradiance using the obtained photosensor current data. The solar spectral irradiance was modeled, and the results were tailored to our custom multispectral camera. A radiometric calibration correction model was developed with information from a single channel of the camera, and the by sharing information across all channels to improve the reflectance accuracy of the output image. The correction model showed positive results in which images that were effect by clouds were successfully corrected for by the dual photosensor instrument.
Acknowledgments

I would like to acknowledge the funding received from the Ontario Centres of Excellence, NSERC and support from industry partner A&L Canada Laboratories. I would also like to acknowledge CMC Microsystems for access to their CAD tools. I would like to also acknowledge my supervisor Dr. Jayshri Sabarinathan for the constant support and positivity as well as Dr. Aref Bakhtazad who provided technical support as a research associate. I would also like to acknowledge Dr. Rangarajan Sudarsan who helped me with my thesis writing. I would also like to give a special thanks to my friends Denis Kikinov and Chantal Mason who listened and helped me through the tough times. I would like to thank my fellow lab mate Brett Poulsen for lending an ear to any ideas I had. I would also like to thank industry partner colleagues Himanshu Shrinkal, Bo Shan and Zahra Ataee for all their hard work to bring me up to speed on the project and Dr. Matthew Cross for providing a lending ear for whenever I felt overwhelmed.
Contents

Abstract i

Lay Summary ii

Acknowledgements iii

List of Figures viii

List of Tables xii

List of Appendices xiii

List of Abbreviations, Symbols, and Nomenclature xiv

1 Introduction 1
  1.1 Remote Sensing Imaging for Precision Agriculture Applications . . . . . . . . 1
  1.2 Motivation for Multispectral Imaging for Small Farm Applications . . . . . . . 3
  1.3 Limitations of Multispectral Cameras . . . . . . . . . . . . . . . . . . . . . . . 4
  1.4 Motivation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
  1.5 Objectives and Thesis Layout . . . . . . . . . . . . . . . . . . . . . . . . . . 6
  1.6 Thesis Contributions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

2 Literature Review 9
  2.1 Radiometric Calibration for Multispectral Cameras . . . . . . . . . . . . . . . 9
    2.1.1 Multispectral Camera Electronics . . . . . . . . . . . . . . . . . . . . . . 10
    2.1.2 Relative Radiometric Calibration . . . . . . . . . . . . . . . . . . . . . . 12
    2.1.3 Absolute Radiometric Calibration . . . . . . . . . . . . . . . . . . . . . . 12
2.2 Image Processing ........................................... 13
  2.2.1 Hardware Deficiency Corrections ............................ 13
      2.2.1.1 Pixel Vignetting .................................... 13
      2.2.1.2 Barrel Distortion .................................... 14
  2.2.2 Otsu Segmentation ....................................... 15
  2.2.3 Maximum Entropy Segmentation ............................. 17
  2.2.4 Sobel Edge Detection .................................... 18
  2.2.5 Canny Edge Detection .................................... 19
  2.2.6 Laplacian Edge Detection ................................. 21
  2.2.7 Laplacian of Gaussian .................................... 22
  2.2.8 Morphological Operators ................................. 23
  2.3 Statistical Metrics ......................................... 24
      2.3.0.1 Confusion Matrix and Rates ............................ 25
      2.3.0.2 Dice Coefficient ..................................... 25
      2.3.0.3 Volume Similarity ..................................... 26
      2.3.0.4 Hausdorff Distance .................................... 26
  2.4 Conclusion .................................................. 28

3 Dual Photosensor Instrument Design and Integration with Multispectral Camera 29
  3.1 Systems Design .............................................. 29
      3.1.1 Requirements ............................................ 30
      3.1.2 Constraints ............................................... 31
  3.2 Optical Design ............................................... 32
  3.3 Analog to Digital Circuit Design ............................. 34
      3.3.1 ADC Selection ............................................ 34
      3.3.2 MCDC04 Sampling and Conversion ....................... 35
  3.4 Digital Communication Design ................................ 37
      3.4.1 Qualcomm Snapdragon Q410 Processor .................. 37
3.4.1.1 Available Digital Interfaces ............................................. 37
3.4.2 Microcontroller Selection .................................................. 37
3.5 Software Coding and Integration with the Multispectral Camera ........ 38
  3.5.1 USB Communication ....................................................... 38
  3.5.2 I2C Communication ......................................................... 44
  3.5.3 Dynamic Range Adjustment .............................................. 50
3.6 Testing Results .................................................................... 52
  3.6.1 Hardware Pin Problems ................................................... 54
3.7 Conclusion ......................................................................... 54

4 Dual Reflectance Panel Implementation for Radiometric Calibration 57
  4.1 Dual Panel Motivation ......................................................... 57
  4.2 Requirements for Automatic Detection of the Dual Panels ............. 61
  4.3 Experiment Setup ............................................................... 62
  4.4 Image Processing Algorithm Implementation ............................ 64
    4.4.1 Controlled Background Design ....................................... 64
      4.4.1.1 Otsu Segmentation Vs. Maximum Entropy Segmentation .... 65
      4.4.1.2 Morphological Operators ......................................... 66
      4.4.1.3 Square Recognition ............................................... 67
      4.4.1.4 Dark Panel Recognition ........................................... 69
    4.4.2 Uncontrolled Background ............................................. 71
  4.5 Results ........................................................................... 74
  4.6 Conclusion ...................................................................... 79

5 Radiometric Map Correction Using Solar Spectral Irradiance Modeling 81
  5.1 Solar Spectral Irradiance Modeling ........................................ 82
    5.1.1 Experimental Setup and Assumptions ................................. 85
    5.1.2 Solar Spectral Irradiance Modeling and Results ................... 87
      5.1.2.1 Downsampling Measured Ground Truth Data .................. 89
      5.1.2.2 Polynomial Fit for the Solar Spectral Irradiance Ground Truth Curve ....................................................... 89
5.1.2.3 Multilinear Regression Model Using Photosensor Data ........................................... 90
5.1.2.4 Evaluating the Effectiveness of Our Solar Irradiation Curve Model .............................................. 91
5.2 Radiometric Correction Model for Cloud Coverage .............................................................. 95
  5.2.1 Linear Model for Radiometric Correction .............................................................................. 97
  5.2.2 Multilinear Model ............................................................................................................... 97
  5.2.3 Results and Discussion ...................................................................................................... 98
5.3 Conclusion ............................................................................................................................... 103

6 Conclusion .......................................................... 105
  6.1 Summary .............................................................................................................................. 105
  6.2 Future Work ......................................................................................................................... 107

Bibliography ............................................................................................................................... 108

A PCB Design for the Dual Photosensor Instrument ............................................................... 118
  A.1 Photosensor ......................................................................................................................... 118
    A.1.1 Downwelling and Upwelling Photosensor ........................................................................ 119
  A.2 Peripheral Board ................................................................................................................. 119

B Software and Analysis Details ............................................................................................. 121
  B.1 USB Packet Return Format for Buffer Contents ............................................................... 121
  B.2 Matlab Code for Ground Truth Generation of Reflectance Panels ................................. 122
  B.3 Violion Plots for Photosensor Input to Spectral Solar Estimation Model ......................... 127

Curriculum Vitae .......................................................... 128
List of Figures

1.1 Diseased leaves RGB Image vs. NDVI Map[1] ............................................. 3
1.2 Image Classification Results of Farm Field on 22/04/2005 and 28/05/2005[2] 4
1.3 Case Example of Farm field imaged with the Western/A&L Multispectral Cam-
  era vs Micasense Red Edge camera at 700nm band[3] ............................. 5

2.2 Architecture of a high speed CMOS image sensor[5] .............................. 11
2.3 Example of Pixel Vignetting due to Shadowing from Photodidode Cavity[6] . 14
2.4 Barrel Distortion Correction[7] ............................................................... 15
2.5 Otsu Segmentation Samples Working (1-3) and Failure(4) [8] ............. 16
2.6 Otsu Method Visualization[9] ................................................................. 17
2.7 Noisy Edge Detection Sample[10] ......................................................... 19
2.9 Morphological Operation Example[12] ................................................. 24
2.10 Confusion Matrix[13] ........................................................................ 25

3.1 Custom Multispectral Camera Filter Response[14] .................................. 31
3.2 Photosensor Optical Design ................................................................. 33
3.3 Downwelling Photosensor Solidworks Model ....................................... 33
3.4 Synchronous Measurement Mode Timing Diagram of the MCDC04[15] .... 36
3.5 Dual Photosensor Instrument System and Integration with the Multispectral
  Camera Block Diagram ........................................................................... 39
3.6 USB Transfer from Snapdragon Q410 to PIC18F45K50 Microprocessor Block
  Diagram .................................................................................................. 41
3.7 NACK Timing Diagram[16] ................................................................. 45
3.8 I2C Starting and Stopping Timing Diagram[16] ............................................. 46
3.10 I2C PIC18F45K50 to MCDC04 Block Diagram ........................................... 48
3.11 Dynamic Range Flow Chart ................................................................. 50
3.12 Downwelling Photosensor Validation Results ........................................... 52
3.13 Time Series Data for Upwelling and Downwelling Photosensors ............... 53
4.1 Reflectance Curves for Different Materials ............................................... 59
4.2 Reflectance Curves for the Spectralon Panels used for Radiometric Calibration . 60
4.3 Input images of the dual reflectance panels for 7 spectral bands of our multispectral camera .......................................................... 63
4.4 Image Processing Flowchart for Controlled Background ........................... 64
4.5 Histogram of Reflectance Panels ............................................................ 65
4.6 Otsu Segmentation results ..................................................................... 66
4.7 Edge Defects Along a Reflectance Panel .................................................. 67
4.8 Bright Panel Detected Output ................................................................. 68
4.9 Bright Square Removal Process .............................................................. 69
4.10 Dark Panel Otsu Segmentations .............................................................. 70
4.11 Dual Panel Segmentation Output Overlay ................................................ 71
4.12 Flow Chart for Uncontrolled Background Segmentation ........................... 71
4.13 Effects of Kernel Size Different on Output Image ..................................... 74
4.14 Results of the uncontrolled background segmentation process ............... 75
4.15 Example of Blurred Image Capture Excluded from Testing ...................... 77
5.1 Plots of the solar spectral energy curve[17] at sea level, outside the atmosphere and one obtained by treating the Sun as a black body at a temperature of 6000k. The absorption zones resulting from the different constituents in the earth’s atmosphere can be seen in the curve corresponding to the sea level. .............. 82
5.2 Plot of the Solar flux versus wavelength corresponding to different Ozone thickness values in the atmosphere taken from[17] .......................... 83
5.3 RGB images captured by Raspberry-pi imager at times t=0 and t=20 minutes in (a) and (b) with the plot of the solar spectral irradiance curve measured by the spectrometer at the same times in (c).

5.4 Schematic showing the setup used to collect data using the Flame spectrometer in (b) and in (a) a representation of the solar spectral irradiance curves obtained using this setup at two different times.

5.5 Flow chart showcasing the 4 different steps undertaken in the current study to model the solar spectral irradiance curve using the current values measured by the photosensor. The first two steps involve downsampling the measured ground truth values measured by the spectrometer and then obtaining a polynomial curve fit to the downsampled values.

5.6 Solar spectral irradiance curves with different scaling for the downsample ground truth (green), 6th order polynomial fit for the downsampled ground truth (orange) and the one obtained from photosensor current measurements via multilinear regression (blue) at four different times during a data acquisition run.

5.7 Solar spectral irradiance curves with the same scale for the downsample ground truth (green), 6th order polynomial fit for the downsampled ground truth (orange) and the one obtained from photosensor current measurements via multilinear regression (blue) at four different times during a data acquisition run.

5.8 Plots showing the probability density function computed for the reflectance data captured by the multispectral camera at different specific wavelength bands. The orange curve corresponds to before radiometric correction while the blue one is after using multilinear regression based radiometric correction. The known reflectance value of the panel is marked by the red dashed line.

5.9 Reflectance maps corresponding to the center wavelength of 542 nm at times 0 min in (a) and uncorrected one corresponding to \( t = 20 \) minutes in (b). The reflectance maps after enforcing correction obtained by using linear regression and multilinear regression methods to the \( t = 20 \) minutes data in (c) and (d) respectively.
List of Tables

3.1 Requirements .................................................. 30
3.2 Spectral Capability Comparison[in nm] for Western/A&L Camera and Multi-
spectral Photosensor ........................................... 32
3.3 Comparison of DDC118 and MCDC04 Photometry ADC ................. 34
3.4 Resolution of MCDC04 ............................................ 35
3.5 Device Descriptors ............................................... 42
3.6 USB Request Photosensor Packet .................................. 44

4.1 Table of Successful Segmentations for Otsu vs. Canny Edge Detector on a
Controlled Background ............................................. 76
4.2 Example Statistics for the 75% Reflection Panel .......................... 77
4.3 Segmentation Comparison Statistics to the Ground Truth ................. 78

5.1 RMSE Metric comparison between Downsampling ground truth and estimated
polynomials .......................................................... 90
5.2 $R^2$ Metric for Estimated Photosensor Polynomial Coefficients vs. Ground Truth
Estimated Polynomial Coefficients .................................. 94
5.3 RMSE Metric comparison between photosensor evaluated solar spectral irradi-
ance and the curve fit downsampling solar spectral irradiance or the downsam-
pled solar spectral irradiance [uW/cm$^2$/nm] ......................... 95
5.4 $R^2$ Metric for Linear or Multilinear Model Radiometric Coefficient Estimation . 99
5.5 RMSE Metric for Standard, Linear or Multilinear Model Reflectance Mapping
function given spetralon panel ground truth .......................... 100

B.1 PIC18F45K50 Diode Data Response Packet Format .......................... 121
List of Appendices

Appendix A PCB Design for the Dual Photosensor Instrument . . . . . . . . . . . . . . 118
Appendix B Software and Analysis Details . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 121
# List of Abbreviations, Symbols, and Nomenclature

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADC</td>
<td>Analog to Digital Converter</td>
</tr>
<tr>
<td>BLSP</td>
<td>Bus Access Manager Low Speed Peripheral</td>
</tr>
<tr>
<td>CCD</td>
<td>Charged Coupled Devices</td>
</tr>
<tr>
<td>CMOS</td>
<td>Complementary Metal-Oxide-Semiconductor</td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
</tr>
<tr>
<td>DN</td>
<td>Digital Number</td>
</tr>
<tr>
<td>DVI</td>
<td>Differential Vegetation Index</td>
</tr>
<tr>
<td>EEPROM</td>
<td>Electrically Erasable Programmable Read-Only Memory</td>
</tr>
<tr>
<td>EVI</td>
<td>Enhanced Vegetation Index</td>
</tr>
<tr>
<td>FFT</td>
<td>Fast Fourier Transform</td>
</tr>
<tr>
<td>FN</td>
<td>False Negative</td>
</tr>
<tr>
<td>FNR</td>
<td>False Negative Rate</td>
</tr>
<tr>
<td>FOV</td>
<td>Field of View</td>
</tr>
<tr>
<td>FP</td>
<td>False Positive</td>
</tr>
<tr>
<td>FPR</td>
<td>False Positive Rate</td>
</tr>
<tr>
<td>FWHM</td>
<td>Full Width Half Maximum</td>
</tr>
<tr>
<td>GPIO</td>
<td>General Purpose Input Output</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>HD</td>
<td>Hausdorff Distance</td>
</tr>
<tr>
<td>HID</td>
<td>Human Interface Device</td>
</tr>
<tr>
<td>IC</td>
<td>Integrated Circuit</td>
</tr>
<tr>
<td>LAI</td>
<td>Leaf Area Index</td>
</tr>
<tr>
<td>LoG</td>
<td>Laplacian of Gaussian</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>MIPI</td>
<td>Mobile Industry Processor Interface</td>
</tr>
<tr>
<td>NDVI</td>
<td>Normalized Differential Vegetation Index</td>
</tr>
<tr>
<td>NIR</td>
<td>Near Infrared</td>
</tr>
<tr>
<td>NRVI</td>
<td>Normalized Ratio Vegetation Index</td>
</tr>
<tr>
<td>OSR</td>
<td>Operational State Register</td>
</tr>
<tr>
<td>PCB</td>
<td>Printed Circuit Board</td>
</tr>
<tr>
<td>RGB</td>
<td>Red Blue Green</td>
</tr>
<tr>
<td>RMSE</td>
<td>Root Mean Square Error</td>
</tr>
<tr>
<td>RVI</td>
<td>Ratio Vegetation Index</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal to Noise Ratio</td>
</tr>
<tr>
<td>SP</td>
<td>Saturation Percentage</td>
</tr>
<tr>
<td>SPI</td>
<td>Serial Peripheral Interface</td>
</tr>
<tr>
<td>TN</td>
<td>True Negative</td>
</tr>
<tr>
<td>TNR</td>
<td>True Negative Rate</td>
</tr>
<tr>
<td>TP</td>
<td>True Positive</td>
</tr>
<tr>
<td>TPR</td>
<td>True Positive Rate</td>
</tr>
<tr>
<td>UART</td>
<td>Universal Asynchronous Receiver/Transmitter</td>
</tr>
<tr>
<td>UAV</td>
<td>Unmanned Aerial Vehicle</td>
</tr>
<tr>
<td>USB</td>
<td>Universal Serial Bus</td>
</tr>
<tr>
<td>VI</td>
<td>Vegetation Index</td>
</tr>
<tr>
<td>VIS</td>
<td>Visible Spectrum</td>
</tr>
<tr>
<td>VS</td>
<td>Volume Similarity</td>
</tr>
</tbody>
</table>
A  Binary Input Image into Morphology Operation
B  Binary Structuring Element
G  Image Gradient
\( G_{aus} \)  Gaussian Image kernel
H  Entropy
I  Image Intensity
\( \sigma \)  Variance
P  Probability
\( \rho \)  Reflectance
\( \lambda \)  Wavelength
\( \text{sr} \)  Steradian
u  Mean
\( w_i \)  Probability of class \( i \) occurring
\( \ominus \)  Erosion operator
\( \oplus \)  Dilation operator
\( \circ \)  Opening operator
\( \bullet \)  Closing operator
Chapter 1

Introduction

1.1 Remote Sensing Imaging for Precision Agriculture Applications

Spectral imaging remote sensing has become an important part of precision agriculture because it is a non-destructive method of providing rapid and accurate information about crops needed to maximize crop yield[18]. With a growing population on earth, there is a greater demand for food which emphasizes the need for crop optimization. Not only are crops being consumed quicker, there are also products that are being made out of crops such as biofuels, plastics and pharmaceuticals which is driving the need for greater crop protection and identification of possible points of failures in crops. While many methods have been developed to prevent disease spread through farms there is still more room for improvement that can be achieved with earlier detection using remote sensing imaging data[19].

An important factor that should be considered before going into any spectral remote sensing application is the required resolution of data, both spatial and spectral, needed to achieve the scientific models required accuracy. Satellite imagery has the common issue where each individually captured pixel might contain several types of vegetation within itself[20]. This would imply that any model derived from the low spatial resolution images would lead to false or misleading results. Satellite imagery is able to capture large areas of land in a small amount of time, which is the main benefit of this remote sensing technique. Being able to image large
swathes of land to create global models is a powerful tool that has helped scientists describe the earth's surface with a greater degree of accuracy than ever before. However there is a great need to capture spectral data variations within a plant itself. This implies that the need for high spatial resolution camera data cannot be satisfied by satellite imagery and will therefore be dominated by the following remote sensing instruments attached to unmanned aerial vehicles (UAV): multispectral cameras, hyperspectral cameras and Light Detection and Ranging (LiDAR) instruments. Multispectral image information is particularly important for small farm applications because it is able to provide high spatial resolution data for individual plants as well as the minimum spectral band information to satisfy various scientific models. Some of the more important analyses and models that can be derived from multispectral cameras specifically for agricultural purposes include [21]:

- Disease detection using Vegetation Index (VI) maps.
- Monitoring plant stress.
- Estimating quality of plant production.
- Estimating biochemical parameters such as levels of nitrogen, chlorophyll and CO₂.
- Plant production models.
- Canopy estimation.
- Estimating the Leaf Area Index (LAI).

LAI itself which is defined as half the total leaf area per unit ground surface area is a very important indicator used to estimate vegetation dynamics, crop monitoring or even climate change [22]. LAI can also be used as a key input parameter into other models such as the hydrology or biochemistry models [23]. While the LAI is an important end product for remote sensing in farm fields, there are also other widely use data products to be derived from surveying a farm field as well. A VI is a map produced from performing mathematical operations on different spectral image bands captured from a multispectral or hyperspectral cameras [24]. Some of the more common VI indices are Normalized Differential VI [NDVI], Ratio VI [RVI],
Normalized RVI [NRVI], Differential VI [DVI] and Enhanced VI [EVI]. Each of the VI have their own pros and cons, however they all commonly use the Near Infrared [NIR] and red spectral bands in their computation. This is because in general unhealthy plants show a difference in reflection in these spectral bands when compared to their healthy counterparts [25]. Figure 1.1 shows an example of an NDVI map, which uses the NIR and red spectral bands to generate the map, for a diseased plant. In the image and NDVI number of less than 0.3 denotes diseased section of the plant. Conversely NDVI number greater than 0.3 is classified as a healthy section of the plant [1].

Figure 1.1: Diseased leaves RGB Image vs. NDVI Map [1]

1.2 Motivation for Multispectral Imaging for Small Farm Applications

The remainder of the thesis will primarily focus on small farm applications, which require high spatial resolution data in order to differentiate between intra-plant reflectances at the leaf level. Being able to probe each leaf for sections of disease is critical to prevent the rapid spread of some diseases through crops. Figure 1.2 shows a typical example of the rapid spread of powdery mildew and leaf rust pathogens in a wheat farm merely 7 days apart [2]. For this application a multispectral camera is usually the instrument of choice since it is able to provide a sufficient spatial resolution for its datasets, while satisfying the minimum required spectral bands for LAI and VI models and it is able to produce these models within the a
limited time. Hyperspectral cameras, while being able to provide many more spectral bands than multispectral cameras, sacrifice its ability to provide sufficient spatial resolution for data without taking long periods of time for data capture [26, 27, 28]. Hyperspectral cameras also provide significantly larger datasets than multispectral cameras and have the added complexity of processing this information in a timely manner.

Figure 1.2: Image Classification Results of Farm Field on 22/04/2005 and 28/05/2005[2]

LIDAR instruments on the other hand are able to provide the highest level of spatial resolution among all the discussed remote sensing instruments, however they also have the downside of providing non-trivial amounts of data like the hyperspectral cameras[29] and lack the spectral information required for VI maps[30]. They are quite good at capturing information for LAI[30], however like the hyperspectral camera they also require extended flights in the air to capture entire farm fields. Since the multispectral cameras satisfy the requirements for small farm remote sensing models, they are commonly chosen over both LIDAR and hyperspectral cameras. The remainder of this thesis will focus on multispectral cameras, and particularly trying to improve upon the reliability of data captured by these instruments.

1.3 Limitations of Multispectral Cameras

Remote sensing using multispectral cameras have 2 clear limitations:
1.3. Limitations of Multispectral Cameras

- the unreliability to provide accurate reflectance data in changing radiation environments and

- the need to establish an initial radiometric calibration mapping function before flight.

It is likely that during a flight over a farm field particularly in Canada and similar climate regions in North America a cloud might cover the sun creating a scenario where the initial mapping function used for radiometric calibration taken at the beginning of flight would no longer be valid during some sections of the flight. Currently there are no methods used commercially to correct for this scenario in the data obtained. One such example is shown in figure 1.3, where the farm fields clearly have different results while capturing similar wavelengths. The mosaic on the left was captured at 708nm with a 8nm bandwidth while the mosaic on the right was captured with at 717nm with a 10nm bandwidth[3]. Both datasets where taken consecutively one after another, approximately 30 minutes apart flying the same path. It is noticeable that both mosaics contain mutual information with one another, however the image on the right contains much more variation that only possibly be due to the changing ambient light conditions.

Figure 1.3: Case Example of Farm field imaged with the Western/A&L Multispectral Camera vs Micasense Red Edge camera at 700nm band[3]
1.4 Motivation

The motivation for this thesis work is to develop the instrumentation, integrate it with a multispectral camera and develop methods required to correct for changing ambient light conditions for multispectral cameras in precision agriculture applications, ideally in real time. Improving the accuracy of the radiometric calibration mapping function as a function of time will improve the reliability of the reflectance datasets captured by the multispectral camera. Building an ambient light sensor instrument that can predict irradiance over the VIS-NIR regime is required since various multispectral cameras on the market boast different central wavelengths and bandwidths. Developing a sensor system which can estimate the irradiance over the VIS-NIR regime is also important for the instrument it is integrated with: the Western/A&L custom multispectral camera because it has the ability to change spectral filters, allowing it to be flexible for different remote sensing applications. This work will also lay the foundation for the novel instrument to fully radiometrically calibrate multispectral cameras as a function of time, and potentially remove the requirement for initial calibration on the ground before flight.

1.5 Objectives and Thesis Layout

The main objectives of the thesis are as follows:

- Build a dual photosensor system with multi-wavelength filters that can monitor the changing ambient light and integrate it with our custom multispectral camera.

- Develop the software codes that will be able to radiometrically calibrate the multispectral camera in-situ. This calibration will include a scalable object recognition program to detect targets of known reflectance.

- Develop a model to estimate ambient light irradiance from 400-900nm.

- Develop a model to calculate the relative radiometric calibration correction factor as a function of time for our multispectral camera.

The models were developed in this thesis only for one of the two photosensors in the integrated dual photo-sensor system. The upwelling sensor hardware was built and integrated
into the camera. However the combined models for using both sensors is beyond the scope of this thesis. Further details on the possible functions for the upwelling photosensor data can be found in the future works section.

Chapter 2 focuses on relevant literature that will setup what the radiometric calibration for multispectral cameras is, why it is so important and what typical methods are currently being used to perform radiometric calibration. Chapter 2 also introduces the background information necessary to understand the upwelling and downwelling photosensor designs, as well as the how the integration with the main custom multispectral camera unit was designed and implemented. Other information covered in chapter 2 includes the common hardware deficiencies observed by camera systems and how to correct for them, as well as some image processing techniques which were used to detect the dual reflectance panels needed to radiometrically calibrate the multispectral camera. Chapter 2 also details relevant statistical metrics for evaluating the estimation of the ambient light conditions and the reflectance correction factor captured by the multispectral camera and photosensor integrated system.

Chapter 3 describes the detailed work of the electrical and software design of the photosensors assembly, and how it was integrated with the android operating system that controlled the main camera unit.

Chapter 4 provides the background about the object recognition methods used to detect reflectance panels. This chapter also outlines the benefits of using multiple targets of reflectance for radiometric calibration. A brief overview of the tested object recognition methods is explored and the advantages and disadvantages of each method is also detailed.

Chapter 5 explains how the estimated ambient light irradiance spectrum between 400-900nm was modeled. This includes the time series measurements from the downwelling photosensor, an RGB rasperry pi camera pointing at the sky for contextual information and spectrometer to ground truth the photosensors. The downwelling photosensor was built to have the 180° FOV to view the entire sky during flight, which is the same as the spectrometer probe that was used to ground truth the readings. Chapter 5 also demonstrates how the photosensor irradiance readings are used to correct the reflectance maps captured by the camera. This was verified by having known targets of reflection in the scene while capturing a time tagged series of images with photosensor data as well. The modified reflectance images in the end were also
shown to be closer to the true value of reflectance for each target.

Finally, chapter 6 summarizes the work completed in the thesis and the suggested future work that can be done with this technology as well.

1.6 Thesis Contributions

All written materials, simulations, design, code and results unless explicitly stated were done by me over the course of the thesis.

Chapter 3 includes PCB and optical designs which was done by Dr. Bakhtazad. A large part of the electrical schematics for the photosensors unit were designed by me. The controlling microprocessor unit was chosen by me, and all code associated with the microcontroller was also generated by me. Circuit design particular for the interfacing peripheral board unit was discussed and between Dr. Bakhtazad and me, however a large majority of the circuitry for the USB connection was done by Dr. Bakhtazad. Any testing in the lab with the photosensor post manufacturing was conducted by me as well, however UAV tests were conducted by A&L personnel with me present. Software implementation of the android-photosensor link and interpretation of the results was done by me.

In chapter 4 the idea for the dual panel calibration was developed by Dr. Bakhtazad and implemented jointly. Suggestions relating to image processing in chapter 4 where received from several people however all coding implementations and interpretation of results were done by myself. All the chapters in this thesis involved the use or idea of a custom built multispectral camera. As part of my thesis, and the industry project which funded me, I helped debug hardware issues post manufacturing on the camera however I was not part of the original hardware design our custom multispectral camera. Software coding related to the photosensors was entirely written by myself, however integrated with the main camera application was facilitated collaboratively by myself and Himanshu Shrinkal in equal parts. GPS assembly issues due to low SNR and coding of the GPS section of the software while not technically under my thesis was part of my involvement inside the project as well.
Chapter 2

Literature Review

In this chapter the background and literature review of three key areas are presented. The first key area that is covered is radiometric calibration which directly contributes to the accuracy of reflectance data gathered by the multispectral camera. The second key area reviewed is the common image processing techniques used for object recognition and camera hardware deficiency correction. The final review section 2.3 presents some of the statistical methods and measures that are used to quantify the success of the image processing results as well as the radiometric correction and spectral irradiance modeling results.

2.1 Radiometric Calibration for Multispectral Cameras

The process to transform the raw image data to reflectance data is called radiometric calibration and is essential for capturing precise and accurate reflection data for various different crops. This radiometric calibration procedure is demonstrated in figure 2.1. First, the multispectral camera is secured to a UAV. Then an image of a known reflectance standard is taken for each spectral band of the multispectral camera. Using both the multispectral cameras readings and the known reflectance of the reflectance standard, the radiometric calibration coefficients are generated. The UAV with the radiometrically calibrated multispectral camera is then flown over a farm field. The images captured during the flight are then post processed with the radiometric calibration coefficients which are used to transform the image data into maps of reflectance.

The accuracy of the reflectance data taken from multispectral cameras is critical for ob-
taining graphical vegetation indices such as NDVI maps and obtaining LAI. It is important to understand how the camera image sensor itself captures data which is explored in section 2.1.1, as well as the image processing models that are used to obtain reflectance data which is explained in sections 2.1.2 and 2.1.3.

2.1.1 Multispectral Camera Electronics

The camera itself captures data in the form of bits, and when an array of bits are put together to form an image, each individual pixel value is referred to as a gray level or Digital Number [DN]. Each individual sensor has its own camera response function, which means that the camera DN will change under the same light conditions based on the exposure time and ISO, which is also commonly referred to as the analog gain, from the circuitry design [31, 32]. While both Complementary Metal-Oxide-Semiconductor [CMOS] and Charged Coupled Devices [CCD] image detectors are common, the remainder of the thesis will focus solely on CMOS based image systems because of their wide use in the industry. Each image sensor is composed of an array of photodiodes coupled with CMOS technology, which in combination provides a device which converts light into current. The low level analog current signal is then subsequently amplified. Each CMOS photodiode array is characterized by a spectral responsivity, which details the efficiency of light power in $W/cm^2$ to current in $A/cm^2$ for a given section of the electromagnetic spectrum.
2.1. Radiometric Calibration for Multispectral Cameras

The exposure time of a CMOS image sensor is defined by the length of time the CMOS image sensor is exposed to light during the image capture. By increasing the exposure time the incident light power is also increased due to the increased number of photons impinging on the exposed image sensor[5]. The image produced by the observing camera is typically well suited for still image capture, however the reliability of the produced image is dependent on the amount of movement in the scene which generates a blur kernel. Image blur is a direct result of when an camera’s image sensor exposure time being too long when coupled with either a moving capturing system or capturing an object that is moving at high velocity[33]. There is a large area of research in which blur kernels are estimated and the image is reconstructed in an approximation of what the scene would like like without motion. There is another method of increasing the brightness of an image without causing blur. This is by increasing the analog gain associated with the CMOS part of the technology. The CMOS section of the circuitry is responsible for the analog gain, which amplifies both the desired signal as well as any noise obtained by the photodiodes.[5] Increasing the gain across the amplifiers will result in a decrease in image quality, however it is often more desirable over a blurred image.

Since the camera’s DN (bit value stored by the analog to digital device) is a function of the exposure time and ISO, it is critical to generate a mapping function to correlate the camera’s DN with its corresponding reflectance value $\rho$. The two common approaches used to convert a
DN to a reflectance value are by absolute and relative radiometric calibration [34, 35].

### 2.1.2 Relative Radiometric Calibration

Relative radiometric correction is the process in which a mapping function is used to directly transform the camera’s obtained DN to a reflectance value [34, 36, 37, 38, 39]. Much research has gone into the best model to transform DN to reflectance values [40], however the majority of research papers use the empirical line method, which is shown in equation [2.1], as a baseline because of its simplicity and reliability. The equation [2.1] is used to find the constant $a$ and $b$, given a known values of $\rho$ and observed values of DN. The main advantages of relative radiometric calibration is that it is simpler and less costly to implement than absolute radiometric calibration. Materials needed to perform relative radiometric calibration include only the multispectral or hyperspectral camera and targets of known reflectance as a function of wavelength $\lambda$. Relative radiometric calibration is done before a multispectral camera is flown over a farm field. The process involves imaging targets with known constant reflectance and creating a baseline transform that can be applied to images captured during flight.

$$\rho = a \ast DN + b$$ (2.1)

### 2.1.3 Absolute Radiometric Calibration

Absolute radiometric calibration involves an extra step in addition to that done for relative radiometric calibration as described above. Here the DN of the camera is first transformed into units radiance in $Wm^{-2}sr^{-1}nm^{-1}$ [41, 42, 43, 44]. Once the transform to units of radiance has been established, as a function of camera gain and exposure, another transform can then be applied to map the units of radiance to reflectance values. The main benefit of absolute radiometric calibration is the disassociation of camera gain and exposure upon the reflectance map. This gives us the capability to change the gain and exposure of the multispectral camera dynamically during flight in order to prevent image saturation or underexposed images. The main disadvantage of absolute radiometric calibration is the extra equipment that is required to be used in a lab setting to accurately create the mapping function between DN and
radiance units. This requires an integration sphere or well characterized diffused light sources illuminating reference target panels[41]. Such equipment is expensive and the setup is difficult, therefore it cannot be done by the average consumer of the multispectral cameras. Other common issues with absolute radiometric calibration which are seen quite commonly with satellite spectral cameras are the temporal effects upon absolute radiometric calibration [44]. The need for re-calibration is often fulfilled by including extra instrumentation on satellites or by using vicarious (in-flight) methods[42], however such methods are not common for multispectral cameras designed for small farm applications. Some manufactures such as Micasense use absolute radiometric calibration, however for the remainder of the thesis only relative radiometric calibration will be used because our multispectral camera used for testing was calibrated using relative radiometric calibration for the reasons described above.

2.2 Image Processing

2.2.1 Hardware Deficiency Corrections

Upon capturing the image, it is necessary to apply some corrections to the image to account for several hardware deficiencies of a multispectral camera. These hardware deficiencies are a product of the image sensor manufacturing process therefore each image sensor will have to be tested and corrected for on an individual basis. While there a multitude of hardware corrections that could be applied to make an image more accurate, only the corrections that have a could have a significant impact on reflectance data have been including in the subsequent section.

2.2.1.1 Pixel Vignetting

The first important hardware deficiencies compensated for by software corrections is the phenomenon known as pixel vignetting. Pixel vignetting is introduced by the difference in incidence angle into the cavity a photodiode resides inside of. Oblique angles will obtain the largest amount of light, while pixels near the edge of the sensor will observe a shadowed region as demonstrated in Figure 2.3 [6]. In order to correct the vignetting, it is necessary to image a uniform surface that is also uniformly lit to create a ground truth. Once the ground truth has
been established, a vignetting correction factor $I_v$ can be obtained by fitting a parabolic model as shown in equation [2.2]. Variables $x$ and $y$ are the coordinates of the pixel to be corrected inside the image sensor, and coefficients $a_0 - a_n$ and $b_0 - b_n$ are found through optimization with the Root Mean Square Error [RMSE] of the ground truth and the input image.

$$I_v(x, y) = \frac{1}{2}(a_{x2} \times x^2 + a_{x1} \times x + a_{x0}) + \frac{1}{2}(a_{y2} \times y^2 + a_{y1} \times y + a_{y0})$$  \hspace{1cm} (2.2)$$

Figure 2.3: Example of Pixel Vignetting due to Shadowing from Photodidode Cavity[6]

\subsection{2.2.1.2 Barrel Distortion}

The other significant correction that is applied to the image is the compensation for barrel distortion. This is done by spatially shifting a pixel’s DN at coordinate $(x_d, y_d)$ to an undistorted location $(x_u, y_u)$. One of the most common approaches is to assume a radial distortion model, then fit the model using a least squares method with a known white and black checkerboard ground truth[45]. The radial component is calculated in equation [2.3], where $(x_d, y_d)$ are the Cartesian coordinates of the distorted image and $r_d$ is the radial position of the distorted image. The white checkerboard should be manufactured in a such a way that each square is a straight line. In the radial distortion model, the center distorted image is denoted as the origin $(x_d, y_d) = (0, 0)$. From the center of the distorted image, the radial distortion is applied given a
2.2. Image Processing

polynomial model shown in equation (2.4) to find the undistorted image gray level at \( r_u \). During the fitting process, coefficients \( a_1 \) and \( a_2 \) are found when the undistorted radial component \( r_u \) and \( r_d \) are known.

\[
\begin{align*}
    r_d^2 &= x_d^2 + y_d^2 \\
    r_u &= \frac{r_d}{1 + a_1 r_d^2 + a_2 r_d^4}
\end{align*}
\]  
(2.3)  
(2.4)

Figure 2.4: Barrel Distortion Correction[7]

2.2.2 Otsu Segmentation

Segmentation is the method in which you can categorize an image into groups of pixels based on mutual traits such as pixel location or gray level intensity. Otsu segmentation categorizations gray scale image histograms into binary partitions by maximizing the inter class variance of the pixel intensity[46, 47, 8]. Otsu segmentation is one of the most popular segmentation methods because of its simplicity and efficiency especially when a bimodal histogram is present. The ideal case is when a controlled smooth background is present, while the desired object is clearly distinct in pixel intensity from the background. Figure 2.5 illustrates three examples of bimodal histograms with the Otsu’s automatic threshold method, while the fourth example shows how Otsu segmentation might fail when two distinct peaks inside the image’s histogram are not present.

The Otsu method is a statistical based method of separating the image into 2 different categories. The foundation of the algorithm is based on calculating the probability of a gray level \( i \) [\( P_i \)] inside of an image by:
\[ P_i = \frac{n_i}{N} \]  \hspace{1cm} (2.5)

Where \( n_i \) is the pixels of intensity \( i \) and \( N \) is the total number of pixels.

Two image classes are defined with probabilities \( w_1 \) and \( w_2 \) of occurring when divided by threshold \( t \). These equations are shown below in 2.6a and 2.6b respectively.

\[ w_1 = \sum_{i=0}^{t} P_i \]  \hspace{1cm} (2.6a)

\[ w_2 = \sum_{i=t+1}^{L-1} P_i \]  \hspace{1cm} (2.6b)

Where \( L-1 \) is the maximum gray level that can be achieved in the image.

Each class has its own mean given by \( u_1 \) and \( u_2 \) respectively. The means are for each class are shown in equations 2.7a and 2.7b.

\[ u_1 = \frac{iP_i}{w_1} \]  \hspace{1cm} (2.7a)

\[ u_2 = \frac{iP_i}{w_2} \]  \hspace{1cm} (2.7b)
And the total mean of the image $u_T$ is given by the equation:

$$u_T = u_1w_1 + u_2w_2$$  \hspace{1cm} (2.8)

The inter class variance $\sigma^2_B$ is shown in equation (2.9), and the ideal threshold is chosen by finding the gray level $t$ at which the inter class variance is maximized shown in (2.10). The inter class variance can be maximized either through an exhaustive search if the bit depth is small enough, such as $N = 255$ bits, or through a hill climbing method. Figure 2.6 illustrates an exhaustive approach.

$$\sigma^2_B = w_1(u_1 - u_T)^2 + w_2(u_2 - u_T)^2$$  \hspace{1cm} (2.9)

$$t = \arg \max_{0 < t < L-1} \{ \sigma^2_B(t) \}$$  \hspace{1cm} (2.10)

Figure 2.6: Otsu Method Visualization[9]

### 2.2.3 Maximum Entropy Segmentation

Another common method of automatic segmentation based on histogram gray levels is the maximum entropy segmentation method[48]. Using the theory of maximum entropy, a threshold is devised by maximizing the amount of information that can be gained from each class in a binary classification. Maximum entropy tends to find an improved threshold value for segmentation if the histogram is not bimodal, which in cases where the background cannot be controlled is very likely. Similar to Otsu’s method, the probability and probability distributions
are used for the formulation of maximizing the entropy. Equation 2.11a and 2.11b shows the formulation of the entropy within classes 1 $H_1$ and 2 $H_2$:

$$H_1(i) = -\sum_{i=0}^{L} \frac{p_i \log_2 p_i}{w_1}$$

$$H_2(i) = -\sum_{i=t+1}^{L-1} \frac{p_i \log_2 p_i}{w_2}$$

The total entropy $H$, or ability to gain information, between both classes should be weighted equally and the total entropy is shown in equation 2.12.

$$H = H_1 + H_2$$

### 2.2.4 Sobel Edge Detection

Edge detection itself is the process of finding pixel intensity step changes inside of an image. An edge is defined by a sudden change from low intensity to high intensity inside of a image, which is a property that edge detection can take advantage of. Edge detection itself is the most useful when trying to sort a target object out from the background of an image. Sobel edge detection is one of the available first order derivative discrete approximation edge detection techniques[10, 49, 50]. The other common edge detection methods which involve taking the gradient upon the image intensities include; Roberts and Prewitt[11]. For the specific application discussed in this thesis, the Sobel edge operator was the preferred filter choice because of its ability to detect vertical and horizontal edges.

The Sobel operator is based on the gradient of the image as shown in equation [2.13], where $I$ the the pixel intensity at location $(x,y)$ and $\hat{x}$ and $\hat{y}$ are the unit vectors in direction $x$ and $y$ respectively.

$$\nabla I = \hat{x} \frac{\partial I(x,y)}{\partial x} + \hat{y} \frac{\partial I(x,y)}{\partial y}$$

Using a two point approximation to find the finite difference between two sections of an image, equations 2.14a and 2.14b show the resultant image gradient along the x axis $G_x$ or y-axis $G_y$ respectively when the smallest distance between pixels $(dx,dy) = 1$.

$$G_x = \frac{\partial I(x,y)}{\partial x} = \hat{x} \frac{I(x + dx,y) - I(x,y)}{dx} = \hat{x}I(x + 1,y) - I(x,y)$$

$$G_y = \frac{\partial I(x,y)}{\partial y} = \hat{y} \frac{I(x,y + dy) - I(x,y)}{dy} = \hat{y}I(x,y + 1) - I(x,y)$$
2.2. IMAGE PROCESSING

Sobel operators are used for detecting line in either the horizontal (x-axis) or vertical (y-axis), however it is common to want to have edge detection in both axes. This problem is easily solved by taking the magnitude of the gradients $G_x$ and $G_y$ as shown in equation (2.15).

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (2.15)$$

In order to use these discrete approximations upon an image, a convolutional filter is applied across the image. Equation 2.16 shows the Sobel operators, with a 3x3 kernel size, below are for the horizontal and vertical axis respectively. The Sobel operators are weighted with a slight bias towards the center of the convolutional kernel.

$$G_x = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, \quad G_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (2.18)$$

One of the largest disadvantages with using a gradient based method for edge detection is that noise inside the image is often amplified through this method as well. The common issue is that true edges are hard to differentiate from noise in the resultant image[11]. Figure 2.7 below shows the example of an image polluted with noise, and the effects upon the edge detected image.

![Image](a) Input Noisy Image  (b) Sobel Operator Output

Figure 2.7: Noisy Edge Detection Sample[10]

2.2.5 Canny Edge Detection

Expanding upon Sobel edge detection, Canny edge detection aims to tackle the problems observed with gradient edge detection: the effects of noise upon the image, thick edges and arti-
facts leftover in the image as well[50, 51]. In order to reduce the effects of the noise, averaging the area around the desired pixel is performed. In most cases a Gaussian smoothing filter is applied, however it would also be possible to apply a boxcar or median filter in order to reduce the effects of noise. This would be application dependent. The equation for a continuous 2-D isotropic Gaussian can be seen in equation (2.19)[52].

$$G_{aus}(x, y) = \frac{1}{\sqrt{2\pi}\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}}$$ (2.19)

In order to create a discrete filter, 2 important factors must be taken into account: the standard deviation for the Gaussian filter and the kernel size. Equation (2.20) shows an example of a discrete 5x5 kernel that approximates a 2-D Gaussian function with $\sigma = 1$[53]. It is important to normalize the kernel such that the integrity of the original bit depth of the image is not influenced by the convolutional kernel.

$$G_{aus} = \frac{1}{273} \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$ (2.20)

Following the Gaussian edge detection the magnitude of the Sobel operators is computed. The next step in the Canny edge detection technique, non-maxima suppression is applied to across the image as well. This is a localization technique where nearby pixels are compared in magnitude and low magnitude gray level values are suppressed to 0. This will reduce the thickness of the edges in the resultant image[11]. Once the edges have been thinned, a dual threshold method of binarizing the image is applied. If a pixel intensity value is below threshold $T_1$ it is considered not to be a edge. If the pixel intensity is above the threshold value $T_2$, it is considered to be a strong edge. If the pixel intensity is in between threshold $T_1$ and $T_2$, the pixel will be processed through a hysteresis method. In essence if the pixel in question is connected to a neighboring strong edge pixel it is accepted into the strong pixel category, else it is rejected. Figure 2.8 shows an example of an image that has been processed by a Sobel operator versus a Canny edge detector method.
2.2. Image Processing

2.2.6 Laplacian Edge Detection

While first order gradient operations perform well to get the general location of an edge inside the image, it requires edge thinning techniques such as Canny edge detection method in order to find the precise location of an edge. Second order approximations by the taking the Laplacian upon the original image produces much sharper edges, with the trade-off that the noise is amplified in an even more drastic fashion then with the first order approximations[54, 55, 11]. The formulation for the Laplacian of an image intensity I is shown in equation 2.21.

\[
\nabla^2 I(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}
\]

(2.21)

Since the Laplacian is not a vector, there is no need to take the magnitude like what is needed for the first order gradient methods. Similar to the Sobel operator, the Laplacian filter can be approximated using \((d_x, d_y) = 1\) with a finite difference approximation shown below. The derivation is shown in equations 2.22a through 2.22d for the partial derivatives of x, however similar methods could be used to obtain the answer for y. It should be noted that I’ and I” are the first and second derivatives of the image intensity I respectively.
Chapter 2. Literature Review

\[ I'(x) = I(x + 1) - I(x) \]  \hspace{1cm} (2.22a)
\[ I'(x + 1) = I(x + 2) - I(x + 1) \]  \hspace{1cm} (2.22b)
\[ I''(x) = \frac{\partial I'(x)}{\partial x} = I'(x + 1) - I'(x) \]  \hspace{1cm} (2.22c)
\[ I''(x) = I(x + 2) - 2I(x + 1) + I(x) \]  \hspace{1cm} (2.22d)

Given that coordinate \( y \) is constant.

Since the derivation has introduced a bias factor it is possible to shift it by applying a variable transform of \( x = x + 1 \) in order to the final equation 2.23.

\[ I''(x) = I(x + 1) - 2I(x) + I(x - 1) \]  \hspace{1cm} (2.23)

The final discrete filter are calculated by adding the results of the \( x \) and \( y \) partial derivatives which is shown in equation (2.21).

\[
\nabla^2 I(x, y) = \begin{bmatrix}
0 & 1 & 0 \\
1 & -4 & 1 \\
0 & 1 & 0 
\end{bmatrix}
\]

(2.24)

2.2.7 Laplacian of Gaussian

Similar to Canny edge detection method, in order to suppress the noise generated by taking the Laplacian upon an image, a Gaussian blurring filter is first applied to the image[11, 54, 56]. Unlike the Canny edge detection method, there is no need to thin the edges of the resultant image. It is however necessary to develop a threshold, which is commonly done by an edge crossing method. The continuous derivation for the Laplacian of Gaussian is shown in equations 2.25 and 2.26[56], given that the equation for the 2-D Gaussian was already explained in equation(2.19):

\[ \text{LoG}(x, y) = \nabla^2 G_{\text{aus}}(x, y) = \frac{\partial^2 G_{\text{aus}}}{\partial x^2} + \frac{\partial^2 G_{\text{aus}}}{\partial y^2} \]  \hspace{1cm} (2.25)
\[ \text{LoG}(x, y) = -\frac{1}{\pi \sigma^4} \left[ 1 - \frac{x^2 + y^2}{2\sigma^2} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}} \]  \hspace{1cm} (2.26)

In practicality rather than approximating the continuous Laplacian of Gaussian shown above, the usual implementation is to convolve the already found discrete Laplacian and Gaussian convolutional kernels before applying over the image. This would require less computation
than applying the Gaussian kernel to the image, follow by the Laplacian since convolution by nature is associative.

### 2.2.8 Morphological Operators

Morphological operators are some of the most useful mathematical tools that can be used in image processing especially for cleaning residual artifacts in an automatic object detection algorithm. More specifically the use of erosion and dilation are key to either decreasing or increasing edge thickness respectively[57, 58, 59]. The use of erosion followed by dilation with kernels of the same size and shape is called morphological opening. In contrast, morphological closing is the act of applying a dilation filter followed by a erosion filter[57, 58, 59]. Morphological closing and opening are by definition increasing and idempotent and are therefore called filters, while erosion and dilation simply morphological operators[60]. Idempotent is a property of mathematical operations in which it can be applied multiple times without changing the result after the initial application.

Erosion of binary images with a binary structuring element can be explained by a moving structuring element B over the binary image A. If B is completely contained inside of A’s pixels when being shifted, that is when the structuring element is a subset of A, the center of the subset of pixels being eroded is considered to be true. In all other conditions the pixel being eroded would be considered false and labeled as 0. After a dilation operation, the image should have less binary pixels in the class DN = 1, which can be thought of as edge thinning. The mathematical formulation for erosion is shown in equation 2.27[57], where $z$ is defined as the translation of B from the origin point to $z$ and $\ominus$ is the erosion operator.

$$A \ominus B = \{z | B_z \subseteq A\} \quad (2.27)$$

Dilation can be thought of as the opposite of erosion, where it will increase the number of pixels in class DN = 1 of an image. Dilation is performed by translating the structuring element B over a binary image A, and the pixel centered at translation $z$ over image A is considered to be class DN = 1 if the intersection between A and $B_z$ is a nonempty set[57]. The mathematical formulation for dilation is shown in equation 2.28[57], where $\oplus$ is the dilation operator.

$$A \oplus B = \{z | B_z \cap A \neq \emptyset\} \quad (2.28)$$
Subsequently, the mathematical formations for opening and closing are shown in equations 2.29a and 2.29b respectively, where \( \odot \) is the opening operator and \( \bullet \) is the closing operator.

\[
A \odot B = (A \ominus B) \oplus B \tag{2.29a}
\]

\[
A \bullet B = (A \oplus B) \ominus B \tag{2.29b}
\]

![Image of morphological operations](image)

Figure 2.9: Morphological Operation Example[12]

Not only is the kernel size important for the dilation and erosion operations, the formation of the binary component of the structuring element \( B \) also has a big influence on the output binary image. Common structuring elements include: box, cross, circle, line and diamond. The structuring element is chosen on a per application basis, as the target to be detected has the greatest influence on the composition of \( B \). Figure 2.9 an example of erosion and dilation structuring element upon a binary image.

### 2.3 Statistical Metrics

In order to evaluate the image processing pipelines developed in chapter 4, various statistical metrics are explained below. These metrics will be evaluated against a ground truth, which is also generated in chapter 4. Chapter 5 will similarly use some statistical metrics to quantify the effectiveness of the generated models for solar irradiance and relative radiometric correction coefficients against their respective measured ground truths.
2.3. Statistical Metrics

2.3.0.1 Confusion Matrix and Rates

When a ground truth is present, it is possible to compare predicted classification of pixels inside of an image. The confusion matrix is a compact method of displaying the following metrics: true negative [TN], true positive[TP], false negative[FN] and false positive[FP][13, 61]. True negative is the number correctly identified pixels for the classification DN = 0. Likewise the true positive is the number of correctly identified pixels of the classification DN = 1. It is also important to define the false negative and false positives, which are the number of pixels that have not been identified correctly for classifications DN = 0 and DN = 1 respectively. Figure 2.10 shows an example of the matrix that is formed from these statistical metrics.

![Confusion Matrix](image)

Figure 2.10: Confusion Matrix[13]

Likewise when analyzing the success of a classification, it is sometimes desirable to find the percentage of success or failure of each possible classification. This can be summarized with the true negative rate [TNR](recall), true positive rate[TPR](sensitivity), false positive rate[FPR] and the false negative rate[FNR]. The given metrics above are calculated using equations 2.30a through 2.30d[13, 61].

\[
TPR = \frac{TP}{TP + FN} \quad (2.30a)
\]

\[
TNR = \frac{TN}{TN + FP} \quad (2.30b)
\]

\[
FPR = 1 - TNR \quad (2.30c)
\]

\[
FNR = 1 - TPR \quad (2.30d)
\]

2.3.0.2 Dice Coefficient

The Dice coefficient is a metric which utilizes components of the confusion matrix into a single numerical value. In other words it is the measure of how similar two image segmentation results
will be in this case. This value is also known as the F-1 metric[61] and calculated with equation 2.31.

\[
\text{Dice} = \frac{2TP}{2TP + FP + FN}
\] (2.31)

The Dice coefficient is not the only metric used for this type of application, as the Jaccard coefficient is also commonly used. For the remainder of this thesis only the Dice coefficient will be used because the Jaccard coefficient can be derived from the Dice coefficient and vice versa[61]. They both give the same information in the end.

### 2.3.0.3 Volume Similarity

Volume similarity is a metric that can be used to compare how similar one volume is to another[61]. In the case that will be used for this thesis, it will be used to compare areas instead of volumes since the thesis mainly works in 2-D when comparing segmentation methods, however the same underlining principles and formula will be used. Volume similarity will give an idea on if a segmentation method will undershoot or overshoot the bounds of the target reflectance panel when compared to a ground truth. Volume similarity is calculated using equation 2.32.

\[
VS = 1 - \frac{|FN - FP|}{2TP + FP + FN}
\] (2.32)

### 2.3.0.4 Hausdorff Distance

The Hausdorff distance[HD] is a metric which is based upon the euclidean distance between two sets of points. The HD is calculated by finding the closest distance \([h]\) from a pixel in the class \(DN = 1\) of a segmented image to a TP pixel in the ground truth for every single pixel that is part of the class \(DN = 1\) in the segmented image which is shown in equation 2.33a. From this subset of distances, the Hausdorff distance is simply the largest one[61]. The Hausdorff distance is a good metric to quantify how far off a segmentation is from the ground truth in terms of a pixel distance. Equation 2.33b for calculating the HD given that A and B are both a finite set of points [61]:

\[
h(A, B) = \max_{a \in A} (\min_{b \in B} \|a - b\|)
\] (2.33a)
2.3. Statistical Metrics

\[ HD(A, B) = \max[h(A, B), h(B, A)] \]  \hspace{1cm} (2.33b)

### 2.3.1 Root Mean Square Error

One of the most common statistical metrics used is the root mean square error (RMSE), which is partially due to its ability to compare 2 or more models with error in the units of interest\[62\]. In such a case the units used in this thesis will be [\(\mu W/cm^2/nm\)] and % respectively. This unit of measurement is important in cases were the absolute value of error is a good metric or performance for the model. In particular finding the absolute value of error for solar irradiance modeling as well as error in perceived reflectance from an camera is particularly useful. The RMSE value is shown in eq (2.34) below, where \(y_{known}\) is the ground truth value, \(y_{predicted}\) is the predicted value and the dataset is made up of values between \(i=1\) and \(N\).

\[ RMS E = \left[ \frac{\sum_{i=1}^{N} (y_{known} - y_{predicted})^2}{N} \right]^{0.5} \]  \hspace{1cm} (2.34)

### 2.3.2 \(R^2\) Metric

The \(R^2\) metric is another statistical metric, also known as the square of the Pearson or correlation coefficient, which is used to evaluate the goodness of fit of a model. In particular the main difference between the RMSE and the \(R^2\) metric is that the \(R^2\) metric is a relative measure of goodness. This implies that it is a unitless entity which is bounded to be below 1. Equation (2.35) below shows the equation for the \(R^2\) metric where \(y_{known}\) is the ground truth value and \(y_{predicted}, y_{mean}\) is the mean value of the dataset. This implies that the further away from the mean of the dataset the known value is, the less penalized the output will be. The perfect output of this measure is \(R^2 = 1\), therefore any model that is below \(R^2 < 0.6\) would be considered a poor fit on the test dataset\[63\]. This model shouldn’t be used alone to assess the goodness of fit, however it is quite good when used in conjunction with the RMSE\[63\].

\[ R^2 = 1 - \frac{\sum(y_{known} - y_{predicted})^2}{\sum(y_{known} - y_{mean})^2} \]  \hspace{1cm} (2.35)
2.4 Conclusion

In this section background information was given for the following overarching topics: radiometric calibration, image processing and statistical modeling. Radiometric calibration covered an introduction into multispectral imaging, and how to change DN’s into reflectances via relative or absolute radiometric calibration. The image processing section explained two important hardware deficiencies in cameras and how to correct for them via post processing on images. This section also covered many basic or common image processing algorithms in order to complete automatic object recognition, as well as the method in order to compare multiple algorithms in order to find the most optimal for agricultural applications.
Chapter 3

Dual Photosensor Instrument Design and Integration with Multispectral Camera

In this chapter the design, fabrication, assembly, and testing of the dual photosensor instrument is presented. This custom sensor system is necessary to achieve the goal of the this thesis, to radiometrically correct datasets measured by a multispectral camera. Section 3.1 describes the dual photosensor instrument design consideration, requirements and constraints. Section 3.2 covers the optical design of the dual photosensor instrument. Section 3.3 covers the analog to digital design of the photosensors which includes the analog to digital converter selection and how they were integrated with the multispectral photosensor. Section 3.4 describes the digital circuit design in order to communicate the measured photosensor data to the main camera processor, which was completed through an interfacing microprocessor. Section 3.5 details the work done with the software coding which was necessary to integrate all the hardware components of the photosensors and custom multispectral camera unit together. The operation and preliminary results of the dual photosensors instrument were obtained and presented in section 3.6. Conclusions and discussion are finally presented in section 3.7.

3.1 Systems Design

Before we describe the detailed design of the dual photosensor instrument, it was important to clearly investigated the requirements and constraints imposed. It is important to note that this
design was not for a stand alone instrument but for a system that was meant to be integrated into our custom multispectral camera. The constraints come from components that were already pre-selected before this design was done and had to be incorporated as part of the design.

### 3.1.1 Requirements

The key goal of the project is to radiometrically correct images captured by our multispectral camera due to changing ambient light conditions due to cloud coverage. For this purpose a novel dual multispectral photosensor instrument was conceived. One photosensor - the downwelling photosensor collects ambient light from the sun impinging on the camera from the top at different wavelengths. The upwelling photosensor collects light reflected from the ground and impinging on the camera from the bottom at different wavelengths. The longterm goal of the project is to develop a model that incorporates both these photosensors. The dual photosensor instrument was also envisioned to enable the real time dynamic optimization of the gain and exposure of the camera to to best capture targets in a farm field. This imposed more restrictions on the design since both an upwelling and downwelling photosensor design must had to be incorporated. The requirements for this system design are summarized in table 3.1:

<table>
<thead>
<tr>
<th>Requirement #</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS-01</td>
<td>The photosensors must span the spectral range of the multispectral camera.</td>
</tr>
<tr>
<td>MS-02</td>
<td>The photosensors must be lightweight to integrate on a drone.</td>
</tr>
<tr>
<td>MS-03</td>
<td>The upwelling sensor must capture the same FOV as the multispectral camera.</td>
</tr>
<tr>
<td>MS-04</td>
<td>The downwelling sensor must capture the entire sky in it’s FOV.</td>
</tr>
<tr>
<td>MS-05</td>
<td>The photosensors must be supported via the multispectral camera’s power supply.</td>
</tr>
<tr>
<td>MS-06</td>
<td>The photosensors must be able to capture data at the same rate or faster than the multispectral camera.</td>
</tr>
</tbody>
</table>

Table 3.1: Requirements
3.1. Systems Design

Figure 3.1: Custom Multispectral Camera Filter Response[14]

Since the multispectral camera is critical to the design of the photosensor, it is important to outline the wavelengths that are able to be captured by the camera as well. The camera is able to capture wavelengths according to figure 3.1. Since the camera spans the 400-900 nm region, the photosensors should as well. Since the photosensors are expected to run on the same power supply and power distribution board as the multispectral camera, the camera is able to supply voltages from 12V to 1.8 V in various standard increments to the photosensor. It is also important to note that the FOV of the camera’s lenses are 47°. The camera has also been designed such that all 7 spectral images are able to be captured and saved within 1 second.

3.1.2 Constraints

The main hardware component constraints that cannot be changed are electrical or optoelectronic in nature:

- The processor for the camera was already chosen.
- The individual analog multispectral photosensor component was already chosen.

The main reason for these constraints were the multispectral camera design was already fixed and the photosensor units were purchased. The camera was chosen to be a Qualcomm Snapdragon Q410[64], and the analog multispectral photosensor was chosen as a PIXELTEQ VIS-NIR PixelSensor[65]. The pixelsensor is composed of 7 different photodiodes each with its own spectral coating that allows specific wavelength bands of light into the photodiodes.
There is also an 8th photodiode however it does not have a spectral filter over top, therefore it is not useful in this application and excluded in the spectral comparison. Table 3.2 shows the comparison for the center wavelengths and bandwidths of the camera [14] versus the photosensors.

<table>
<thead>
<tr>
<th>Band Number</th>
<th>Western/A&amp;L Multispectral Camera</th>
<th>Multispectral Photosensor</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Center Wavelength</td>
<td>FWHM Bandwidth</td>
</tr>
<tr>
<td>1</td>
<td>464</td>
<td>32</td>
</tr>
<tr>
<td>2</td>
<td>542</td>
<td>35</td>
</tr>
<tr>
<td>3</td>
<td>639</td>
<td>50</td>
</tr>
<tr>
<td>4</td>
<td>669</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>708</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>800</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>845</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 3.2: Spectral Capability Comparison [in nm] for Western/A&L Camera and Multispectral Photosensor

### 3.2 Optical Design

In the optical design, the goal is to make sure that the photosensors are receiving diffused light equally across the photodiode and to make sure that the FOV requirements are met as well. The upwelling photosensor was designed to have the same lens as the custom multispectral camera, a 47° FOV lens. The lens mount was also similar to the camera, an M12x05 holder which allowed to manual adjustment for the lenses with a focal length of 5.52 cm. The aperture of the lens is \( \frac{f}{2.4} \) which implies that the lens has a fixed aperture size. The downwelling sensor was designed with a 180° FOV lens in order to capture the entire sky. This lens was purchased from Sunex, and has a focal length of 1.96 mm. Similar to the previous lens, it is mounted via a M12x05 mount, however it has a slightly larger aperture of \( \frac{f}{2} \).

In order to diffuse the light in an equal manner across the photodiode array, a Lambertian
optical diffuser was implemented[66]. The Lambertian property of this filter makes it such that the spectral radiance transmitted through the diffuser and onto the photodiodes is the same regardless of incoming angle. This will make it so that the entire sky will be unbiased in terms of spectral irradiance readings at the surface of the multispectral photosensor. The incoming light is also diffused such that the light is scattered before impinging on the surface of the photodiodes. This will remove the spatial bias of the multispectral photosensors, such that equal optical power be spread onto the multispectral photosensor. Figure 3.2a and 3.2b show the upwelling and downwelling designs respectively without the lens mount.

Figure 3.2: Photosensor Optical Design

Figure 3.3 shows the 3-D model of the photosensor instrument design with the lens, the lens holder, the surrounding 3-D printed casing and the PCB designs inside as well.

Figure 3.3: Downwelling Photosensor Solidworks Model
Chapter 3. Dual Photosensor Instrument Design and Integration with Multispectral Camera

3.3 Analog to Digital Circuit Design

Since the output of the photodiode is an analog current, there is a need to change this into a format that a computer can comprehend and track. The common method of changing an analog electrical signal into a digital signal is by performing sampling followed by quantization. Sampling is the rate at which you can capture time signals and quantization is the estimation of the current or voltage into bits. The quantization process dictates the resolution of the instrument, which is important for the accuracy of data captured. An IC called an analog to digital converter[ADC] is able to do this process.

3.3.1 ADC Selection

Selecting the proper ADC for this application is crucial to collecting reliable and fast data to support the main camera unit. In this case 2 different ADC’s are compared: the MCDC04[15] and the DDC118[67]. While there were other available microcontroller at the time, an initial search was performed to find an ADC suitable for photometry applications, in which only these 2 were available for a somewhat reasonable price. Table 3.3 shows a comparison chart of important technical specifications between the two ADC’s.

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value for DDC118</th>
<th>Value for MCDC04</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Channels per Chip</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>Max Number for chips in subsystem</td>
<td>Unlimited</td>
<td>4</td>
</tr>
<tr>
<td>Communication protocol</td>
<td>SPI</td>
<td>I^2C</td>
</tr>
<tr>
<td>Power Consumption</td>
<td>13.5 mW/channel</td>
<td>Low power, Standby mode optional</td>
</tr>
<tr>
<td>Noise</td>
<td>5.2ppm of FSR</td>
<td>Low noise with 50/60Hz rejection</td>
</tr>
<tr>
<td>Resolution</td>
<td>20 bit</td>
<td>16 bit</td>
</tr>
<tr>
<td>Price(CAD)</td>
<td>5$ in batch of 1000</td>
<td>85$ per unit</td>
</tr>
</tbody>
</table>

Table 3.3: Comparison of DDC118 and MCDC04 Photometry ADC

In terms of sheer capability the DDC118 was a much better IC, however the MCDC04 required much less hardware overhead. The Serial Peripheral Interface [SPI] protocol for the
3.3. Analog to Digital Circuit Design

DDC118 required more wires between master and slave devices then when compared with the I²C protocol of the MCDC04, and the MCDC04 was a 16 pin integrated circuit [IC] as compared to a 64 pin IC. Since both the MCDC04 and DDC118 are able to satisfy the number of required channels, which is 16, neither had an advantage in this section. In the end the MCDC04 was selected as the component to be used because of the greatly reduced price and significantly lower hardware overhead in the PCB design.

3.3.2 MCDC04 Sampling and Conversion

The MCDC04 is implemented with a topology similar to a delta-sigma converter, in which the output in bits is a function of the ratio between the input current and a user selected reference current[15], as shown in (3.1).

\[ \text{OUT} = \frac{I_{\text{in}}}{I_{\text{ref}}} T_{\text{int}} f_{\text{clk}} \]  

(3.1)

The MCDC04 utilizes an integrator whose function is to find the average input current \( I_{\text{in}} \) at the end of a predefined integration time. One of the disadvantages of the MCDC04 is that the resolution of the reading changes as a function of the integration time used, as seen in Table 3.4. It also measures the amount of time passed as a function of the clock frequency for the device as seen in (3.2). It should also be noted that the MCDC04 only has a 16 bit memory storage capacity, and if a 20 bit reading is taken, the lowest 4 bits will be discarded. This effectively keeps the readings as a 16 bit resolution.

\[ N_{\text{clk}} = T_{\text{int}} f_{\text{clk}} \]  

(3.2)

<table>
<thead>
<tr>
<th>( N_{\text{clk}} )</th>
<th>1024</th>
<th>2048</th>
<th>4096</th>
<th>8129</th>
<th>16384</th>
<th>32768</th>
<th>65536</th>
<th>131072</th>
<th>262144</th>
<th>524288</th>
<th>1048576</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T_{\text{int}}[\text{ms}] )</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>64</td>
<td>128</td>
<td>256</td>
<td>512</td>
<td>1024</td>
</tr>
<tr>
<td>Resolution[bit]</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 3.4: Resolution of MCDC04

The resolution and reference current shown in eq(3.1) are controlled by the configuration register which can be written to by through the I²C communication line. The configuration register is able to hold 8 bits, in which the 4 most significant bits are for the reference current and the 4 least significant bits are used for the integration time.
The MCDC04 ADC is capable of 4 different sampling schemes: continuous measurements, command measurement, synchronous measurement and synchronous start and stop measurements. Since the measurements from the multispectral photosensors will be synced and controlled by the main camera unit, a continuous measurement schemes is not necessary. The synchronous start and stop measurement scheme allows control by the main camera for the integration time, however this was not implemented. The only difference between the command measurements mode and synchronous measurement mode is that they controlled the start of measurement via software or hardware respectively. In the end the synchronous measurement mode was implemented, therefore the start of measurement was controlled by a hardware signal. In terms of controlling the MCDC04, the SYN pin was critical. When the SYN pin was brought low signified it signified the start of a reading and subsequently resetting the pin to a high voltage for the next cycle. Once the readings for the MCDC04 are finished, the ready pin RDY is brought high such that the master of the $I^2C$ communication line is able to request the data from the ADC. Figure 3.4 shows the timing diagram for the synchronous mode of operation for the MCDC04.

![Figure 3.4: Synchronous Measurement Mode Timing Diagram of the MCDC04](image)
3.4 Digital Communication Design

3.4.1 Qualcomm Snapdragon Q410 Processor

The main processor for the Western/A&L multispectral camera is the Qualcomm Snapdragon Q410E processor[64, 3]. This processor was chosen because of its capability to interface with camera’s via a set of Mobile Industry Processor Interface [MIPI] data lines which enables image and video transmissions[68]. Hence finding a device that can interface between the Qualcomm processor and the MCDC04’s was necessary to collect the multispectral photosensor data.

3.4.1.1 Available Digital Interfaces

The Snapdragon Q410 Bus Access Manager Low Speed Peripheral [BLSP] has many supported interfaces which include $I^2C$, SPI and Universal Asynchronous Receiver/Transmitter [UART] as well as a high speed Universal Serial Bus [USB] 2.0 port. Even though the processor is able to directly interface with the 4 separate MCDC04 units, it was decided to implement a microcontroller to act as an interface unit between the ADC and the main processor. The main reason for this decision was based on cables lengths to place the dual photosensor instrument on the wings of a fixed wing drone and ease of testing initial prototypes with a computer via a USB 2.0 communication line. It was possible to find development boards for microcontrollers which would allow USB 2.0 communication and facilitate ease of testing without needing complete PCB’s.

3.4.2 Microcontroller Selection

The microcontroller that was selected in order to interface between the MCDC04 and the Q410E processor was the PIC18F45K50 microcontroller[69]. The microcontroller is able to communicate via USB 2.0 and $I^2C$, which is necessary for this application. The unit is also equipped with a set of General Purpose Input/Output [GPIO] pins which is necessary for the required SYN, and array of READY pins for the set of MCDC04’s. The microcontroller is a low power microcontroller which has flash cells for both the program memory and Elec-
3.5 Software Coding and Integration with the Multispectral Camera

Once all the electrical components had been selected, it was then possible to visualize how the system will interact with each other. Figure 3.5 provides the block diagram of the entire system, including serial protocols, critical IC’s and the optical hardware assembly. The main software interfaces explained include the USB V2.0 protocol as well as the I²C protocol. The USB interface was coded on the android system using JAVA and XML languages, and the USB interface on the PIC18F45K50 was coded using C. Supporting libraries from both android and Microchip for the snapdragon processor and microcontroller where used in order to establish USB communication. The PIC18F45K50 also used an I²C library offered by Microchip to communicate with the MCDC04. This section was also written using the C language.

3.5.1 USB Communication

The USB communication line provides the link between the Snapdragon Q410E processor and the PIC18F45K50 microcontroller. This implies that both systems must be able to initiate USB communication and then communicate with the host or device on the serial network. Figure 3.6 shows a block diagram which includes setting up the USB communication protocol, as well as the data acquisition sequence which has been synced with the trigger of the first image sensor of the multispectral camera. The host of the USB communication is the android processor,
software coding and integration with the multispectral camera

which means it manages the microcontroller device and processes the data acquired from the microcontroller since it does not have a native math library. The device on the network is the microcontroller and its responsibility is to properly identify itself to the host by providing USB descriptors on first contact, as well as acquiring the photosensor data via $I^2C$ communication when prompted.

The steps in order to establish USB communication go as follows: the host enumerates the devices on the USB network, devices respond with USB descriptors, the host decides on who to communicate with and claims the interface of the selected device[70, 71]. Once the interface has been claimed the transfer type must be selected from : controlled, isochronous, bulk and interrupt. Control transfer transfer is used by the host to initiate communications with devices on the network, and is therefore used during enumeration and when claiming interfaces for the
setup period. Interrupt data transfer is a short packet delivery transfer which is specified to happen at regular intervals. Bulk transfers are short packets that are not scheduled and have the potential to deliver large amounts of data and utilize the full bandwidth of the network. The downside of bulk transfers is that they are not guaranteed to deliver in a set period of time if the network is busy. Isochronous transfers have larger packets and is a larger packet protocol that is scheduled at regular intervals of time as well. Isochronous is also the only transfer type that does not do error detection through cyclic redundancy checking, which is not desirable. For the photosensor application it was decided that the camera trigger would signal the start the data acquisition cycle of the photosensors, so the non-scheduled bulk transfer method was selected to acquire data since the camera is not guaranteed to run on a periodic timer.
Figure 3.6: USB Transfer from Snapdragon Q410 to PIC18F45K50 Microprocessor Block Diagram
Starting with the beginning of the USB setup, enumeration first detects the capabilities of the device on the Vbus. Since the network was designed to be highspeed at 480Mbps transfer speed, a 5V line signal is present on the differential D+ line[71]. Once the host has identified that the device is highspeed, the network is reset and the USB descriptors are requested. The following USB descriptors are common and have been implemented for the network: device descriptor, configuration descriptor, interface descriptor, endpoint descriptor and the string descriptor.

The device descriptor is the first that is read by the host during enumeration, and lets the host know what the function of the device is, who made it, what product it is and how many configurations the device is able to support. For the case of the peripheral board manufactured table 3.5 shows the relevant values chosen for the device descriptor. An important note is that the USB VID hexadecimal F055 was chosen because it is commonly used for open-source hardware projects and is not registered to any company. The physical hardware of the USB is also different from standard cables and therefore is not compliant with computer USB ports. This eliminates the risk that any host other than the designed Q410E processor and custom cable assembly will be able to interface with this specific USB device. The special class code of 0x00 was also used, which implies that the function of this device will be explained inside of the interface descriptor.

<table>
<thead>
<tr>
<th>Device Descriptor Name</th>
<th>Hexadecimal Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>USB Spec type (2.0)</td>
<td>0200</td>
</tr>
<tr>
<td>Vendor ID</td>
<td>F055</td>
</tr>
<tr>
<td>Product ID</td>
<td>005F</td>
</tr>
<tr>
<td>Class Code</td>
<td>00</td>
</tr>
<tr>
<td>Device Release Number</td>
<td>0001</td>
</tr>
<tr>
<td>Number of Possible Configurations</td>
<td>01</td>
</tr>
</tbody>
</table>

Table 3.5: Device Descriptors

The configuration descriptor sets up the maximum number of bytes per packet, the maximum power consumption on the line and the number of interfaces on the device. The maximum
allowed bytes per packet was designated as decimal 64 which is compliant with the bulk transfer USB transfer protocol. The maximum power consumption on the line was chosen as 50 mA at 5V and the number of interfaces on the device is only 1.

The interface descriptor served two important functions: defining the endpoints available on the device and the class description that was not defined in the device descriptor. The endpoints of the USB protocol are the unidirectional hardware which receives or transmits data. This is typically implemented in the Central Processing Unit [CPU] registers or in dual-port memory[71]. For this specific implementation, 2 endpoints were defined, one for sending commands and one for photosensor data transfer. The class description was initially chosen as a Human Interface Device [HID], since initial prototyping was done manually through a development board. Since the HID class was already implemented, it was not changed when interface with the android system.

The endpoint descriptor designated the endpoint address, the size of the endpoint, the transfer type used at the access point[72]. The critical information chosen was that the endpoint will accept 64 bits at a time, using bulk transfer type and that it will expect a Nack or Ack, depending on if it is an input or output endpoint, every frame. This finishes the device descriptors, which provided sufficient information for the android host processor to communicate with the microcontroller device.

Once the device has been enumerated by the android host, and received its device descriptors an important choice must be made. Should the host begin communicating data with the enumerated device? A safety protocol is implemented by android in which it will prompt the user to accept the USB device before any data is allowed to be transferred[73]. If it was the first instance of this data transfer, when the device has first been connected to the android system, it will request permission to get data from the connected device. If permission has previously been obtained, and saved, the android host will wait until the first camera trigger to request data from the device. Once the trigger command has been issued, the host will also command the device to return data using the packet structure shown in table 3.6.

The device at this point will collect the photosensor data via the I²C line from the ADC, and respond back with a data packet. The structure of the data, is shown in table B.1 of appendix B. Since each array element in the buffer can contain only a single byte, and the data collected
from each photodiode is 2 bytes, the photodiode data is split into 2: the high byte of data and the low byte of data. The photodiode data is then reconstructed at the opposite side of the link. A read check was implemented for each ADC which is shown in buffers 37-40 as well. The reconstruction process is simply done by bit shifting the high byte by 4 to the left and adding the lower byte as per normal. Converting the binary photosensor data into a decimal value is the final step before processing the data can occur. The purpose of the read check is to make sure that the ADC was able to be accessed via the I2C line. If the particular ADC device is not accessible the read check will be returned as 0, else it will be returned as a 1. At this point the USB communication section has been explained in detail, and in order to complete the explanation of the system it is necessary to explain in detail the I2C communication protocol and how it was utilized to control all the ADC’s.

### Table 3.6: USB Request Photosensor Packet

<table>
<thead>
<tr>
<th>Buffer Contents</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffer[0] = 0</td>
<td>Report ID</td>
</tr>
<tr>
<td>Buffer[1] = 0x81</td>
<td>Command - Get Sensor Data</td>
</tr>
<tr>
<td>Buffer[2] = Not implemented</td>
<td>Sequence Number (not implemented)</td>
</tr>
<tr>
<td>Buffer[4-63] = Not implemented</td>
<td>Unused</td>
</tr>
</tbody>
</table>

#### 3.5.2 I2C Communication

In order to discuss the I2C communication between the PIC18F45K50 microcontroller and the MCDC04 ADC’s, it is first necessary to detail what I2C communication is. The I2C protocol is a standard bidirectional interface between the controlling device, the master, and the responding devices called the slaves[16, 69, 15]. The main overview of the I2C hardware is that it has one wire for sending data, commonly referred to as the SDA line, and one line for the clock, commonly referred to as the SCL line. The clock controls how quickly data can be sent down the line, therefore having a faster clock will enable faster data transmission. The clock itself is generated by the master, however it is important to make sure that the slave devices are able
3.5. **Software Coding and Integration with the Multispectral Camera**

To track the maximum clock frequency as well. In order to communicate along the I\(^2\)C bus the following procedure must be followed[16]:

1. The master must send a start condition.
2. The master must send the device address of the device it wants to communicate with.
3. The master will either send data or request data from the device.
4. The slave will send data if requested.
5. The master will send a stop condition to terminate the communication.

Once the command has been sent or data has received from the slave to master or from the master to slave, they must respond to the packet with an acknowledge bit [ACK] or a not acknowledge bit [NACK]. In order to acknowledge the last received byte, the receiving device must pull down the SDA line during a low period of the clock on the SCL line[16]. Likewise a NACK is inferred by the transmitter when the SDA line remains high during a low period of the clock after the last transmitted byte was sent. Figure 3.7 shows an example of a NACK’d packet.

![NACK Timing Diagram](image)

**Figure 3.7: NACK Timing Diagram[16]**

In order to send a start or stop condition from the to the slave device, the master must either pull the SDA line from high to low or from low to high when the SCL line is high respectively.
Figure 3.8 shows both the starting and stopping conditions that must be sent along the SDA line to the slave device.

Figure 3.8: I2C Starting and Stopping Timing Diagram[16]

In order to differentiate slave devices it is necessary that each slave have a unique identifier. In this case the identifier is commonly referred to as the device address. The master does not need an address since it is always controlling the communication line. With the I²C implementation it is not possible for slave devices to communicate with each other. For the specific MCDC04 slave devices, it is possible to set the slave address during the PCB layout on a hardware pin of the IC. The unique identifier has 5 preset bits, and 2 bits (A0 and A1) controlled by external pin layout which is shown in figure 3.9a. Since the maximum possible of ADC’s were employed, the hexadecimal addresses ranged from 0xE8 to 0xEE assuming the final bit was 0. During the addressing stage of the I²C the final bit will always be 0 because it controls the read/write properties of the link. If the master ends the command with a 0, it implies that it will continue to send data along the I²C line and the slave will only be allowed to acknowledge. Figure 3.9b also shows the physical layout for the MCDC04 IC, which shows pins A0 and A1, that control the device address, which are set high or low via a 3.3V line or a ground line.

The final section which needs to be mentioned before the detailed design of the MCDC04 analog to digital conversion is explained is that the final bit of every byte transmitted by the master should end with 0 or 1 depending on if it expects the slave to respond with data or not. Ending the byte with a 0 implies that the master will continue to give commands or data to the device. Ending with a 1 implies that the master expects the slave to transmit data back towards
Figure 3.10 below shows the overall flowchart for the communication along the I2C line. This diagram shows the code in a block diagram form that was written on the PIC18F45K50. The first task that is completed by the master is to start the clock along the SCL line to establish communication between the master and slave devices. This is accomplished using the I2C library provided by Microchip, and the clock frequency for the I2 line to 400KHz. This is accomplished by taking the internal clock frequency of 48MHz and dividing through by the equation shown in (3.3). The SSPADD value controlled by a internal register on the microcontroller and is set manually to 29 decimal or equivalently 0x1D in hexadecimal.

\[ f_{clk} = \frac{f_{internalclk}}{4(SSPAD + 1)} \]  

(3.3)
Figure 3.10: I2C PIC18F45K50 to MCDC04 Block Diagram
3.5. Software Coding and Integration with the Multispectral Camera

Once the clock line had been initiated, it was then possible to write to the MCDC04 devices separately in order to set the reference current and integration time for each of the respective analog to digital converter channels. In order to set these values, the device must be put into configuration mode. This is accomplished by writing to the 5th bit of the Operational State Register [OSR] a 0. It is also necessary to disable power down mode during this time by also writing a 1 to the 2nd bit of the OSR as well. Once put into the configuration state, it is possible to change the bits inside the configuration registers CREGL and CREGH. CREGL contains the bits for the reference current and integration time which is shown in table 3.4. It also contains the direction of measurement for the photodiodes, which in our case is measuring the anode of the photodiode. The CREGH register contains information pertinent to the digital divider which is not implemented and the measurement mode for the MCD04 device. The CREGH registers have the appropriate bits set such that the device is put into synchronous measurement mode. It is also important to note during this section the MCDC04 is explicitly told to use its internal clock for measurements which is on average 1.024 MHz frequency signal. Following all configuration settings have been written to the appropriate registers, the OSR register is returned to measurement mode with the possibility to enter a low power state to conserve power when not measuring.

With the slave devices now configured with the appropriate settings for measurement, a conversion cycle is triggered by bringing the SYN pin from high to low, then low to high with an appropriate wait interval of 1ms as shown in figure 3.4. This starts the conversion cycle, and the respective MCDC04 device will immediately set its ready pin RDY low to signify that it has not finished collecting its measurements. Once the conversion for the device has finished, the respective MCDC04 will set the RDY pin high and data transfer along the I² line then will occur. At this stage the master requests the data over the SDA line until all 4 channels have been read. Once all 4 channels have been read, the PIC18F45K50 will set its USB buffer read check to 1 such that the main processor unit will know that conversion was possible on that particular MCDC04 device.
3.5.3 Dynamic Range Adjustment

Provided the photosensor readings are now able to be acquired on the main Qualcomm Q410E processor with the ability to tune both the integration time and the reference current, a simple algorithm was formulated to adjust the microcontroller settings to prevent saturation or low dynamic range readings from the sensors. The general logic can be described in the flow chart shown in figure 3.11.

Figure 3.11: Dynamic Range Flow Chart

The data acquisition was designed such that only the reference current was changed due to the fact that the resolution of the reading was tied to the number of clock cycles which is in essence the integration time. Keeping the resolution of the instrument at the highest possible
setting, 16 bits per channel, was the design choice to maximize the accuracy of the readings. This was also done in consideration that the photosensors acquired data much quicker than the multispectral camera, therefore time was not the limiting factor for performance. Since the reference current was sole variable that could be changed in order to increase or decrease the digital readings received from the MCDC04, it can be seen from the previously seen equation (3.1) that increasing the reference current would decrease the digital output of the MCDC04. Likewise by decreasing the reference current it is possible to increase the digital output of the MCDC04. Since the CREGL register is only able to generate 4 reference currents, optimizing the readings at most took 3 cycles assuming constant light conditions.

From the upwelling or downwelling photosensor, a maximum and minimum reading from all the recorded spectral channels was generated. These values were used to optimize the photosensor readings by checking user defined upper and lower bounds in order to change the reference current. The upper and lower bounds were generated by trial and error as well as some experience from radiometrically calibrating the custom multispectral camera. In general an upper bound of 95% showed good results for the multispectral camera since it used the entire dynamic range. Likewise the 40% lower bound showed an acceptable dynamic range as well, since having very high reference currents could reduce the accuracy of the instrumentation as well. Keeping the reference as close as possible to the photosensor current being measured was the desired setting for the measurements. Likewise having saturated data was a much larger problem than then having undersaturated data so the upper bound readings were calculated first and given priority at the decision making process.

Following that the dynamic range had been tuned the digital values were converted to currents by taking the digital value, the reference current setting from the acquired data and the constant number of clock cycles, 65536, and finding the input current of the photodiode as shown in equation (3.4). This data is then saved for later processing along with the multispectral cameras image data.

\[
I_{in} = \frac{OUT \times I_{ref}}{N_{clk}}
\]  
(3.4)
3.6 Testing Results

Once the dual photosensor instrument was built and the software code successfully loaded onto the hardware, the functionality of the system was tested. Time series data was collected from the upwelling and downwelling photosensors. Figure 3.12a shows a full reading of the downwelling photosensor at time \( t=0 \)s during a time series data acquisition sequence. In this graph it is possible to see the wavelength dependence of the photosensor currents. At this point in time the photosensor currents are not converted to radiance values, which have been discussed in depth in chapter 5, however figure 3.12b shows a strong correlation from the normalized upwelling photosensor readings to a normalized spectrometer reading which was time synced for the measurement. One result which came out of the measurements was that the 850nm band consistently had readings which were higher than any of the other spectral bands. This can be explained by the increased bandwidth of the spectral filter over the photodiode, which is 30 nm compared to every other spectral filter being 20nm in width. As a consequence, the current generated from the photodiode over the 850nm band is quite large since the current is directly proportional to the optical power in watts that is delivered to the photodetector.

The other baseline testing which was performed included a time series measurement in order to check that the photosensors responded with data as a function of time. The photosensors at this stage performed as expected when operated on the ground. Figure 3.13 below shows the time series data capture for both the upwelling sensor, named "bot" in this case, as well as the
downwelling sensor, named "top" in this case. It is noticed that at very low current readings the spectral readings from the photosensor can also jitter quite a bit. This is an assumed issue due to noise on the channel which can be derived from a multitude of factors such as electromagnetic interference on the PCB trace, too high of a reference current or an unstable clock due to thermal fluctuations.

Figure 3.13: Time Series Data for Upwelling and Downwelling Photosensors
3.6.1 Hardware Pin Problems

During initial test flights of the multispectral camera integrated with the dual photosensor instrument presented an issue where a communication error would occur between the microcontroller and the MCDC04’s lying on the photosensor PCB’s. This occurred when some of the important hardware pins were bent during user assembly of the multispectral camera system, including the dual photosensors, on the field before flight. Figure 3.14a and 3.14b shows the bent pin issue which occurred on multiple units that caused either the SDA pin for the I2C to be left open or the power pin to be left hanging. In either case the photosensors would fail to communicate with the MCDC04 and the read checks would be left blank. This problem was discovered upon debugging after several test flights. This failure mode was particularly difficult to find as the errors occurred randomly for a while and depended on the handling of the multispectral camera system by the individual field user. This will be rectified in the next design when a connector with better trace guides will be placed on the PCB to interface with the cables. The temporary workaround solution used in the mean time was to bend the pins back into place and to carefully place the connectors before launching the camera system on a UAV for image data acquisition. After this fix, only a couple of test flights were performed to test for the dual photosensor instruments functionality and it was shown that the data from the photosensors were received properly.

3.7 Conclusion

A custom dual photosensor instrument consisting of a multispectral upwelling and downwelling photosensors was designed, built and integrated with our multispectral camera. The system was tested successfully and was able to provide the data needed to radiometrically correct the multispectral camera images obtained during UAV flight. A thorough investigation of the requirements was conducted before designing the dual photosensor instrument. The design included the selection of the appropriate photosensors, ADC and microcontroller required to interface between the main Qualcomm Snapdragon process on the Multispectral camera and the multispectral photosensors. The optical design included the integration of a lens system to satisfy the required FOV for the upwelling and downwelling photosensor. A Lambertian opti-
3.7. Conclusion

(a) Upwelling Photosensor Cable Assembly

(b) Bent Pin of Upwelling Photosensor Connector

cal diffuser is integrated into the optical system as well to uniformly distribute the light to the multispectral photosensors. The software design included both USB and I2C communication lines which enabled all the systems to interface together seamlessly to stream multispectral photosensor data to the camera during flight. Notable results from the photosensor showed that the dual photosensors did capture data, and that it was relatively close to a spectrometer readings as well. The data captured from the photosensors was still collected as raw current values instead of the irradiance values required for radiometric calibration of the multispectral camera as a model to do so was yet to be developed. This analysis will be covered in Chapter 5 and is currently done only in post-processing. In order to correct radiometrically calibrated image data from our multispectral camera, a radiometric calibration sequence must first be established. Chapter 4 will cover a novel dual panel automatic detection program which was
developed and enabled fast and accurate radiometric calibration of the multispectral camera in the field. This data was then used with the raw photosensor data to develop the necessary data processing algorithms to correct the reflectance data as a function of time. This work also enables the efficient collection of reflectance data by the multispectral camera for the time series analyses required in chapter 5. In the future, this model can be incorporated directly into the onboard multispectral camera software.
Chapter 4

Dual Reflectance Panel Implementation for Radiometric Calibration

In this chapter a dual reflectance panel based radiometric calibration procedure is implemented along with automatic detection of the panels for faster calibration in the field. An important aspect of spectral imaging is the radiometric calibration procedure typically performed before imaging data are captured. This is necessary to correlate the digital data or radiance data obtained by the multispectral camera to actual reflectance data. An algorithm to automatically detect the panels and identify the type of panel is implemented which is important for the radiometric corrections that are described in chapter 5.

4.1 Dual Panel Motivation

Before the start of a UAV flight over a farm field, an initial radiometric calibration is performed, typically using a single known reflectance panel. However there are some drawbacks with this technique particularly for vegetation imaging where there is large contrast between the lower VIS wavelength regime and the higher (VIS-NIR) wavelength regime. This leads to either saturated or very low contrast images in a data set. Here we have implemented a dual panel based calibration technique to solve this issue. Additionally it also takes a long time to manually run the calibration in the field and there are errors introduced if the user doesn’t image the panel properly. So the dual panel technique also requires an automatic detection
algorithm which has been implemented that enables faster and more accurate calibration.

Two separate Spectralon reflectance panels were used for calibration rather than a single reflectance panel[74]. The advantage of using a dual panel for calibration is that it will be able to optimize the dynamic range of the camera better than a single panel, which in turn will improve the accuracy of the measurement system. Figure 4.1a shows an example spectra of various materials expected to be found inside a farm field, and figure 4.1b shows some of the highest reflectance vegetation expected to be imaged by the multispectral camera[75].
4.1. Dual Panel Motivation

Figure 4.1: Reflectance Curves for Different Materials

An assumption was made that between the region of 500nm to 700nm a typical plant up to a 20% reflectance, and that from 700nm to 900nm the material could have up to a 75%
reflectance.[76, 77, 78] Figure 4.2a and 4.2b below show the reflectance of the dual Spectralon reflection panels purchased from Ocean Optics used to radiometrically calibrated the multispectral camera.

Figure 4.2: Reflectance Curves for the Spectralon Panels used for Radiometric Calibration

It is important to note that the reflectance of a plant is also heavily influenced by where it is in its current life cycle and its water content as well. Matching the estimated spectral regions to an appropriate reflectance value allowed for optimizing the dynamic range of the images generated by the camera. The typical method of calibrating the multispectral camera with a 99% reflectance panel allows imaging of the widest range of objects, however using
the dual panel method allows for a higher resolution reflectance map to be generated. This is seen in the equation (4.1) in which the minimum resolvable reflectance $\Delta \rho$ is calculated by the dividing the spectral reflectance of the panel $\rho_{panel}$ by the normalize digital reading $DN$ of the reflectance panel and the bit depth $N$ of the image sensor. By lowering the reflectance of the panel, it also lowers the minimum resolvable reflectance step, thereby increasing the accuracy of the instrument. Since the gain and exposure are set using a low reflectance panel in the VIS wavelength regime, extra care must be taken to ensure that the targets crops in the farm do not surpass the reflectance used in the radiometric calibration procedure.

$$\Delta \rho = \frac{\rho_{panel}}{DN(2^N - 1)}$$ (4.1)

The optimization of the imaging system is based upon changing the gain and exposure of the camera while also taking into consideration that too long of an exposure time during flight will cause image blur. However, increasing the analog gain of the system is detrimental to the SNR of the image, therefore a maximum exposure time of 2.5ms was defined such that a minimal pixel blur would be allowed. The maximum allowable exposure time was defined based upon the maximum height of flight, the FOV of the camera and speed of the drone.

### 4.2 Requirements for Automatic Detection of the Dual Panels

Having an automatic algorithm was a requirement because in the field one of the most common issues was being able to align the reflectance panels to the center of the LCD display for the camera. The root cause of this issue is the glare from sunlight during the day, which made the LCD panel almost unreadable, therefore the autonomous algorithm was placed as a high priority requirement to prevent human alignment error. Development of the algorithm was not done natively on the system because there was a shortage of custom cameras available for development at the time, therefore having a requirement which would allow for the algorithm to be ported over was also essential. The code itself was written in python, via a Jupyter Notebook integrated development environment. It was noted that every function that was implemented
could similarly be found using OpenCV or equivalent statistic focused libraries in C or Java. The other requirements were designed based on the assumption that the camera will be calibrated in a farm field, and to well calibrate the camera quicker. Equipment that was allowed for the design included small objects that can be fit inside of a truck, the camera itself and the reflectance panels.

The requirements of the image processing algorithms are defined below. Requirements are defined with a "must" if they are absolute requirements that have to be met and requirements with a "may" are not required but provide additional insight.

- The algorithm must be able to detect both reflectance panels under good exposure and gain settings.
- The algorithm must be autonomous.
- The algorithm may be able to request user confirmation at the end.
- The algorithm must be able to be ported to an android operating system.
- The algorithm must be faster than the current method of calibration (<5 minutes).
- The algorithm must be able to return the saturation percentage of the image acquired.
- The algorithm may be able to return other statistical data other than saturation percentage.
- The algorithm must be able to operate in a farm field.

4.3 Experiment Setup

The materials used for the test setup include; a custom-built multispectral camera[14], 2 Spectralon Lambertian reflectance panels with constant reflectance and a black Bristol board. The custom-built multispectral camera is composed of multiple sensor channels measuring 7 discrete wavelengths, each with different bandwidths respectively. Each discrete band of wavelength will produce a separate image with 1280 x 960-pixels and 12 bits of information per pixel. During the experiment, the 12 bits of information was reduced to 8 bits to comply with
some of the libraries used in python. Once a binary mask is generated for the reflectance panels in the image, the 12-bit version of the desired output parameters could be computed if desired, however the information was kept as an 8 bits value for the current model. Each image sensor on the camera is slightly separated, this will effectively add a spatial shift between each target in the image when reflectance panels are relatively close to the camera. This implies that each band will require its own segmentation to find the reflectance panel. Below in figure 4.3, is an example of 7 images captured by the camera, each labeled as with its own center wavelength.

![Image of input images of the dual reflectance panels for 7 spectral bands of our multispectral camera](image)

Figure 4.3: Input images of the dual reflectance panels for 7 spectral bands of our multispectral camera

The test setup for the experiments was devised such that the following parameters could be found for each reflectance panel; average gray level, maximum gray level, minimum gray level, standard deviation of gray level and saturation percentage (SP) which is defined in equation (4.2). The saturation percentage is calculated by finding the average DN with the reflectance panel and dividing by the maximum DN value possible with the camera.

\[
SP = \frac{\sum DN_{panel}}{(2^N - 1) \sum a_{panel}} \times 100\%
\]  

(4.2)

In order to test greater standard deviation values across the reflectance panel, a more directive beam of light was applied to the target setup, rather than a diffuse light source. This will introduce a light intensity gradient across the standard reflectance panel, which should be reflected inside the standard deviation of the resultant images. An approximation that the reflectance panels have a constant spectral reflectance is used because the spectral bandwidth for
each sensor is small compared to the rate of change of the actual spectral reflectance. It is also important to mention that the Lambertian property of the reflectance panels guarantee that the reflectance mapping would not change as a function of phase angle between the camera and the light source.

### 4.4 Image Processing Algorithm Implementation

#### 4.4.1 Controlled Background Design

A modular approach was taken to segment the reflectance panels in succession. Choosing the background that the reflectance panels lay on to be black enabled the first segmentation act similar to a bimodal distribution, in which the brightest square could be found. Following segmentation, pixels that were detected are then grouped into groups of objects based on proximity. These objects, called labels, are then parsed to find which ones are squares. Saving the brightest square as a mask, it is then subtracted from the original image such that the new distribution of DN in the image acts similarly to a bimodal distribution if the gray level DN = 0 is ignored. The square detection algorithm is then applied again to find the dark square and the original mask is added to the dark square mask. Figure 4.4 below shows the implementation of the algorithm in the form of a flow chart. It is important to note that the process must be ran twice in order to find 2 separate reflectance panels, as shown by the arrow returning from remove brightest square to Otsu segmentation once again.

![Image Processing Flowchart for Controlled Background](image)

**Figure 4.4:** Image Processing Flowchart for Controlled Background
4.4.1.1 Otsu Segmentation Vs. Maximum Entropy Segmentation

Otsu segmentation was determined to be the best method of segmentation based on the histogram of the image. Dr. N. Otsu originally put forth the algorithm for a threshold selection method based on the gray-level of histograms in 1979 based on the minimization of intra-class variance. The Otsu segmentation method is known to work well with histograms which showed bimodal peaks, however it also performed well for the bright and dark reflectance panel segmentations even though they showed a trimodal histogram. This occurs because during the first segmentation it considered the dark panel and the background to be part of class 1 and the bright panel to be part of class 2. During the second segmentation, with the bright panel removed, the Otsu segmentation was able to consider the background to be part of class 1 and the dark panel to be part of class 2. Figure 4.5 below shows an example gray level distribution of an image with the bright square and the bright square removed. It can be seen clearly that the image with the bright square is trimodal and the image without the bright square is also trimodal. Being able to detect the bright square relies upon the fact that the first segmentation threshold is somewhere between 100 and 200 and the second segmentation is somewhere between 40 and 55 given the current cameras optimized gain and exposure values.

![Figure 4.5: Histogram of Reflectance Panels](image)

While maximum entropy is known to perform better with multimodal histogram segmentations, in the case of the bright and dark square segmentations it proved unreliable because...
while it was able to detect the bright reflectance panel, it was only able to detect the dark reflectance panel about half of the time. One of the possible solutions for this was to perform contrast enhancement by either histogram equalization or to try a different segmentation algorithm. In consideration that Otsu’s segmentation had already shown favorable results, the maximum entropy segmentation was discarded.

Following the flow chart shown in 4.4 for the detection of the reflection panels, it is necessary to segment the original image using Otsu’s method. It was noticed that there were 2 possible outcomes to the segmentations; a single square was found without any artifacts or a single square with artifacts also present. Figure 4.6a and 4.6b demonstrate the results of the Otsu segmentation. While not all the images had a perfect segmentation, the output from the Otsu segmentation produces results which are good enough to find statistical measures on the reflectance panels. In general, the segmentation had thresholds typically between 20000-33000 with respect to the original bit depth of the data depending on the wavelength band being examined.

![Figure 4.6: Otsu Segmentation results](image)

**4.4.1.2 Morphological Operators**

Morphological erosion is the process of reducing the area of segmentation by passing a structuring element over a binary image. Applying this to the Otsu segmented images will prevent
the edges of the reflectance panels from entering the segmented regions of the resultant image. Following Otsu’s segmentation, a binary output was eroded by a structuring element of type sitk.sitkBox and a variable kernel radius. By trial and error, a kernel radius of 3 seemed to produce the best the most accurate results without losing too much information around the edges of the segmentation.

Following the erosion upon the resultant image, dilation was performed with a slightly lower kernel radius. The kernel utilized was a sitk.sitkBox kernel with an isotopic kernel radius of 2, which is slightly less than the original kernel radius to eliminate some of the blurring which occurs at the edge interface of the panel. Figure 4.7 shows a zoomed in version of a 75% reflectance panel, which clearly shows the inconsistent data along the edges of the panel which need to be removed for radiometric calibration.

![Figure 4.7: Edge Defects Along a Reflectance Panel](image)

### 4.4.1.3 Square Recognition

Since square detection is the basis of this algorithm it was necessary to use an external library scikit-image in order to group pixels together based on proximity into so called “labels”. This external library was also able to provide 2 important details about the labels; the area and the perimeter. Using the assumption that each panel is perfectly square, equation (4.3) was used to relate the area in pixels to the calculated area using the perimeter of the label. While the ideal area ratio is 1, an acceptable error of $\pm 15\%$ was allowed in order to decide whether the label was square, which can be seen in the bounded region in equation (4.4). The allowable error for the square was also found via trial and error, in which the image was manually observed to see if the algorithm operated in an acceptable range. An additional restriction was placed on each label such that it must have at least an area of 10000 pixels in order to prevent any small artifacts from being counted as a reflection panel. These small artifacts are caused due to noise
during the image capture or during the segmentation of the original image. If the resultant number of labels exceeds 5 after iterating through the area ratio and the minimum number of pixels for each label, only the 5 largest are taken into the identifying the brightest label stage.

\[
\text{Area}_{\text{ratio}} = \frac{\sum \text{DN}_{\text{Label}}}{(P_4)^2}
\]  

\[
0.85 \geq \text{Area}_{\text{ratio}} \leq 1.15
\]

The brightest square in the image is found using the formula shown in equation (4.2) to find the saturation percentage of all labels in the image, and then by taking the label with the largest saturation percentage. The saturation percentage is bounded between 0 and 100%, while the ideal saturation percentage is determined by which reflection panel is being used for the calculation and which wavelength is being captured. Providing the saturation percentage of the reflection panels in the image satisfies one of the key requirements that needs to be output of this program.

In order to visualize the results of the segmentation, the overlay function was used inside of sitk in order to show the binary segmented square detected image on top of the original image. Figure 4.8 below all 7 wavelength bands and their respective bright panel segmentations. The output overlay of the image sets where stored inside of the folder named BrightSquareOverlay for each respective band which could be transferred to a user for confirmation if desired.

![Figure 4.8: Bright Panel Detected Output](image)

After the bright panel detection, the following step was to generate an inverted binary image of the bright panel to multiple element wise with the original image. This would effectively remove the bright square from the original image. The flow of this process can be seen inside
of the flowchart seen in figure 4.9a below. The bright square removal process can also be visualized as shown in figure 4.9b.

![Bright Square Removal Flow Chart](image)

(a) Remove Bright Panel Flow Chart

![Example Removed Bright Panel Image](image)

(b) Example Removed Bright Panel Image

Figure 4.9: Bright Square Removal Process

### 4.4.1.4 Dark Panel Recognition

Having modified the original image, and therefore modifying the original image’s histogram, segmentation was performed once more in order to detect the second reflection panel. The threshold was once again chosen by the Otsu method however this time the segmentation produced artifacts around the edges of the bright reflection panel if the chosen dilation kernel was smaller than the erosion kernel. Figure 4.10 below shows the segmentation of the removed bright square image such that the dilatation kernel equal to erosion kernel used beforehand.
The process in order to find the brightest square in the image was unchanged in the following loop since the brightest square was now the dark reflectance panel. The results concluded that 200/203 images were able to have both squares detected. The failed panel detections were not the fault of the algorithm, as the input images themselves were flawed with either extreme blurring or the panels themselves were not in the image. That means the dual square detection algorithm had a 98.52% chance of succeeding, and even higher if the the 3 faulty input images were not taken into account. Once again, the following information was calculated for each dark square: saturation percentage, average gray level intensity, max gray level intensity, minimum gray level intensity and standard deviation of the gray level intensity. Figure 4.11 below shows the output binary segmentation of both squares overlaid in purple with the original image.
4.4.2 Uncontrolled Background

In the case that the background is not controlled certain assumptions made about the distribution of pixel intensities are no longer valid. There is no guarantee that the histogram of the image will be bimodal or trimodal to implement Otsu segmentation or maximum entropy segmentation. More specifically there is many possible scenarios in which the background intensity could be less, greater than or equal to the average pixel intensity of the target reflectance panels. In such cases, localized region of light differences would be the most reliable method of object detection, in essence edge detection-based methods. Figure 4.12 below shows the generalized approach to finding the reflectance panels given an unknown background.

Figure 4.12: Flow Chart for Uncontrolled Background Segmentation

There are four noticeable differences in the signal processing chain between the controlled and uncontrolled background:
• the smoothing stage
• the zero crossing edge detection filter
• the hysteresis thresholding
• and the hole filling filter.

Inside the smoothing stage, two different methods were tested, a Gaussian filter and a median filter. The main reasoning for using these filters was to average the image before the zero-crossing edge detection in order to reduce noise which was present in the input image which is then subsequently amplified in the output image. Fairly large image kernels were used for the smoothing stage to obtain a resultant segmentation with an acceptable level of noise. Through testing, the median filter was found to be less effective at noise reduction than the Gaussian filter when tested across an image dataset of 840 images. The best segmentation results with the least noise were done with a Gaussian filter with kernel size of 30, however the output was still not perfect and many morphological filters were employed to reduce the effect of noise afterwards.

Two different zero-crossing edge detection algorithms were tested inside of the dataset; Canny edge detection and Laplacian of Gaussian (LoG). Controlling the noise in the resultant image was one of the two main problems observed using the zero-crossing edge detection methods, with the other problem being un-linked edges in the output. In testing it was found that Canny edge detection produced less noise in the resultant image, which is expected since it is based upon the first derivative of the image. The LoG was more susceptible to noise which is also expected since it is based upon the second derivative, however it was still able to threshold the edges of the reflectance panel with a smaller Gaussian filter applied beforehand. This is expected because of the effects of the non-maximum suppressing techniques within the Canny edge detection algorithm, as well as the hysteresis thresholding. Non-maximum suppression within the canny edge detection algorithm is the process in which edges in the resultant image are thinned. This is achieved by comparing the resultant gradient magnitudes of pixel intensities within a set distance of each other, and keeping the largest value only. This has an effect of turning large edges in the output to thinner versions of themselves. Hysteresis thresholding is
an edge linking algorithm which can take the thinned edges of the non-maximum suppression algorithm and link them together to get a continuous boundary. This is completed by sorting the pixel intensity values into 3 separate categories based on a dual threshold; low values, medium values and high values. High values will always be kept and low values will always be set to zero, however the medium values will check to see if it is adjacent to a high value. If it is, it will be considered a high value and kept else it will be set to zero.

Both the non-maximum suppression and hysteresis thresholding techniques are used to remove the weak 1st order intensity differences, and to provide more confidence inside of the edges that were obtained. A challenge associated with this method was that while there was a high confidence in the edges that were observed, the rounded edges of the reflectance panels had difficulty being detected within the threshold. This led to the need for a large variance kernel size in the Gaussian filter in order to provide edges in the output with sufficiently large girth to be connected later by a morphological filter such as dilation. A hole filling algorithm was implemented once the edges of the reflectance panels were segmented. The geodesic morphological operation utilized failed under two different conditions; when the edge of the reflectance panels were not connected and when the edges of the segmentation touched the boundary of the image itself. Given these two failing conditions the following criteria was developed: the edges of the panel following canny edge detection must be at continuous and the panels must be roughly centered in the image.

Following the hole filling algorithm, an opening morphological filter was applied to the image to remove any background noise afterwards. Since the object trying to be detected was square, a box erosion and dilation filter were applied in sequence, similar the the controlled background. When trying to maximize the area across the reflectance panel, both the erosion and dilation kernels were chosen to be 15 in size. When trying to minimize the false positive error of the reflection panel a kernel size of 15 was chosen for the erosion filter and 10 for the dilation filter. This would be intentionally reducing the data used to calculate the reflectance map, exposure time and ISO gain in order to improve precision of the algorithm. The size of the kernel chosen was directly related to the number of pixels in the image, therefore if this algorithm was to be used for a different camera the kernel size would have to be optimized again. Figure 54.13 shows the differences in ideology of kernel sizes for the reflectance panel.
detection for this particular camera.

![Figure 4.13: Effects of Kernel Size Different on Output Image](image)

After the morphological operations pixels were grouped based on proximity and intensity into labels once again, and the two largest square labels where taken to be the dark and bright reflectance panel. An assumption was made that no other squares of equal or larger size would be found inside the image, such that a false positive or false negative would be seen. Once the squares were found the statistical measures were taken from the pixel values within the dark and bright panel labels and saved with a purple overlay to a folder. Figure 4.14a below show the entire process in full, from canny edge detection to overlay the overlay image that was saved. Figure 4.14b also shows an example of the same process one a non-uniform background inside of a lab to demonstrate the flexibility of the algorithm to work in different scenarios other than on a black background.

4.5 Results

In order to validate both the controlled and uncontrolled background cases, an active contour segmentation was taken as a ground truth. The disadvantages of the active contour method are
that it is easy to get stuck inside local minima of the target image and also it requires an initial
guess of where the target of interest resides inside the image. It also has long computation
times which makes it unsuitable for radiometric calibration purposes in the field however it’s
accuracy and precision is good for validating other techniques. Manually checking the output
segmentation of the active contour segmentation was required as well to ensure a proper ground
truth. The first dataset used for testing was 203 images across 7 monochromatic bands. Table
4.1 below shows the results in terms of how many panels where able to be detected. During
this experiment the background was a purely dark Bristol board for both the Otsu segmentation
and the Canny edge detection algorithms. It is clear to see that the Otsu segmentation method
boasted a higher number of successes, however upon inspection of the data set itself it can
be seen that 2 bands had blurry images from the lens not being properly focused. The blur
effect upon the images explains why the edge detection method perform as well as the Otsu
segmentations, as the edge itself was not properly defined. The active contour binary images
were generate using a Matlab script name “ActiveContourExample.m” in which the user ac-
tively modified the images using simple isotropic scaling and edge enhancement techniques
before an active contour was applied to each individual image. The code for this program can
be found in appendix B.

Table 4.2 shows the example statistics found across the 75% standard reflectance panel.
The statistics were found for the first image of the dataset across all 7 wavelength bands. From
the results it is possible to infer that when the light source is not diffuse and fairly directive, as
Table 4.1: Table of Successful Segmentations for Otsu vs. Canny Edge Detector on a Controlled Background

intended, and the standard deviation of the gray level across the reflectance panel is high when the image is not saturated, as seen in the 1st and 5-7th bands. This is validated by the light intensity gradient which can be seen across the reflectance panel upon visual inspection of the images as well. Another inference that can be extracted is that when the image is saturated, the gradient cannot be detected as the entire panel is a uniform gray level. The ideal case would be a low standard deviation value across the reflectance panel, indicating an ideal diffuse light source such as the sun, with a percentage of saturation below 100%. It is also noticeable from the data that the brighter reflection panel has a higher value of standard deviation than the darker reflectance panel, when the panel has not reached gray level saturation. This could imply that nonlinearities in the camera exist when the camera reaches high digital numbers, which is possible since the radiance power received by the camera is not linearly dependent upon the pixel intensity especially at nearing saturation of the DN for the image sensor.

In order to evaluate the segmentation with a quantifiable metric the following statistics were calculated for every image with the active contour used as the ground truth; Dice coefficient, False Negative Rate (FNR), False Positive Rate (FPR), True Negative Rate (TNR), True Positive Rate (TPR), volume similarity and Hausdorff distance. For each statistic, it was the most logical to take the average, maximum, minimum and standard deviation across all bands in order to compare. While calculating the segmentation statistics across all bands was the most logical method to prove that the proposed segmentations worked, it is also important to look at the images being taken. Going through the images after taking them it is clear that regardless of the camera being held still, the 800 nm and the 845 nm images are blurry. This is the case
4.5. Results

<table>
<thead>
<tr>
<th>Image Band</th>
<th>Average gray level</th>
<th>max</th>
<th>min</th>
<th>Standard deviation</th>
<th>Saturation Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>225.47</td>
<td>255</td>
<td>158</td>
<td>21.76</td>
<td>88.42</td>
</tr>
<tr>
<td>2</td>
<td>254.74</td>
<td>255</td>
<td>243</td>
<td>0.89</td>
<td>99.89</td>
</tr>
<tr>
<td>3</td>
<td>254.74</td>
<td>255</td>
<td>243</td>
<td>0.89</td>
<td>99.89</td>
</tr>
<tr>
<td>4</td>
<td>254.74</td>
<td>255</td>
<td>246</td>
<td>0.50</td>
<td>99.80</td>
</tr>
<tr>
<td>5</td>
<td>240.88</td>
<td>254</td>
<td>194</td>
<td>14.25</td>
<td>94.46</td>
</tr>
<tr>
<td>6</td>
<td>155.39</td>
<td>254</td>
<td>97</td>
<td>17.91</td>
<td>60.94</td>
</tr>
<tr>
<td>7</td>
<td>169.87</td>
<td>254</td>
<td>99</td>
<td>18.19</td>
<td>66.61</td>
</tr>
</tbody>
</table>

Table 4.2: Example Statistics for the 75% Reflection Panel

because the camera had suffered from defocusing after a test flight, and the lenses needed to be tightened and refocused. Figure 4.15 demonstrates an image which is slightly out of focus, however squares are still visible to the eye. These images within the dataset present issues when trying to validate the proposed segmentation as the squares are still able to be detected, however ground truths did not accurately represent the data since active contours try to find boundaries inside an image. Since the boundaries are blurred of course the ground truth would hold no value.

![Figure 4.15: Example of Blurred Image Capture Excluded from Testing](image)

Table 4.3 shows the analysis of the segmentation methods as compared to the ground truth method of active contours when using 144 out of 203 images. The reason 2 bands where excluded from the segmentation analysis was because of the blur in sets of images. It can
be seen from the data that both methods produced accurate segmentation results compared to the ground truth, as demonstrated by the dice coefficients, false positive rate (FPR) and true positive rate (TPR). The Otsu method provided a slightly better segmentation, as can be seen by the low Hausdorff distance. Both methods under-segment the panel, which is preferred such that the false positive rate is always 0 and the volume similarity being negative. This under-segmentation is the preferred case because the accuracy of the instrument depends on all pixel’s values used for the calibration, and any false positives will skew the resulting reflectance map.

<table>
<thead>
<tr>
<th>Method</th>
<th>Measure</th>
<th>Ave</th>
<th>Max</th>
<th>Min</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Otsu</td>
<td>Dice Coefficient</td>
<td>0.980</td>
<td>0.986</td>
<td>0.962</td>
<td>0.003</td>
</tr>
<tr>
<td>Canny</td>
<td></td>
<td>0.970</td>
<td>0.980</td>
<td>0.961</td>
<td>0.004</td>
</tr>
<tr>
<td>Otsu</td>
<td>False Positive Rate</td>
<td>0.000</td>
<td>0.015</td>
<td>0.000</td>
<td>0.001</td>
</tr>
<tr>
<td>Canny</td>
<td></td>
<td>0.000</td>
<td>0.002</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Otsu</td>
<td>True Positive Rate</td>
<td>0.962</td>
<td>0.973</td>
<td>0.940</td>
<td>0.006</td>
</tr>
<tr>
<td>Canny</td>
<td></td>
<td>0.942</td>
<td>0.961</td>
<td>0.924</td>
<td>0.008</td>
</tr>
<tr>
<td>Otsu</td>
<td>Volume Similarity</td>
<td>-0.038</td>
<td>-0.025</td>
<td>-0.051</td>
<td>0.006</td>
</tr>
<tr>
<td>Canny</td>
<td></td>
<td>-0.059</td>
<td>-0.039</td>
<td>-0.079</td>
<td>0.009</td>
</tr>
<tr>
<td>Otsu</td>
<td>Hausdorff Distance</td>
<td>6.005</td>
<td>17.263</td>
<td>4.472</td>
<td>1.571</td>
</tr>
<tr>
<td>Canny</td>
<td></td>
<td>8.796</td>
<td>14.000</td>
<td>7.810</td>
<td>1.054</td>
</tr>
</tbody>
</table>

Table 4.3: Segmentation Comparison Statistics to the Ground Truth

Analyzing table 4.3 more in depth, starting with the dice coefficient it can be seen that the segmentation at first glance can be considered to be a good representation of the region of interest (ROI). With an average value of 97% and 98% for Otsu and Canny respectively and a standard deviation with less than a percent, it stands to reason that it is a good representation of the 2 reflection panels. One of the goals when calibrating the instrument beforehand was to have as small of an FPR as possible, and with an average value of less than 0.00% and a maximum value of under 2% it can be said that both segmentation did a good job of not obtaining false positives. In the future it would be recommended to increase the size of the structuring element for the eroding morphological process in order to decrease the maximum value down
to 0% particularly for the Otsu segmentation method since it boasted higher maximal FPR. Upon visual inspection of the segmentations, it can be seen that the proposed solutions are always volume negative as compared to the ground truth, therefore the volume similarity in fact upholds what eyes can process with empirical data shown in table 4.3.

In terms of TPR and FNR, it can be seen that this was in fact the weakest component of the segmentation. The volume similarity reflects that it is an under-segmented image, which can also be seen with a relatively low TPR of 96.2% and 94.2% for Otsu and Canny respectively. This implies that the edges of the panel were not captured inside the segmented region, however this was also intentionally brought into the design by have unequal erosion and dilation kernel sizes. For future work on the Otsu segmentation, the structuring element should be increase to 6 rather than 3 in order to reduce the FPR. The final data set to analyze is the Hausdorff distance, which can be seen to be on average 6-9 pixels long. With respect to how large the image is, this a relatively small error, however it does validate that the segmentation is centered on all sides of the image, thus retaining high accuracy but loosing precision.

4.6 Conclusion

A dual reflectance panel implementation for the time dependent radiometric calibration of the multispectral camera has been achieved. An algorithm to automatically detect the panels and identify the type of panel is also implemented. The longer term goal of this work is to develop an algorithm which can be incorporated onboard the multispectral camera and eventually be used to improve the radiometric calibration of the multispectral camera dynamically during flight. Results from the implemented algorithms showed that the critical data which can be used to improve the multispectral camera and its calibration procedure was obtained. The algorithm was also able to obtain the saturation percentage, maximum gray levels, minimum gray levels and standard deviation from the reflectance panels. The saturation percentage of the reflectance panels was necessary to complete the modelling presented in chapter 5. The other statistical metrics obtained were also useful in identifying if the lightsource used for the radiometric calibration was sufficiently diffused.

It is important to note that the work presented in this chapter specifically used a directive
lightsource to test the robustness of the algorithms. However, the measurements presented in chapter 5 used the diffused light from the Sun in order to radiometrically calibrate the multispectral camera. This ensured that the reflectance data obtained from the Spectralon panels were reliable and also that the reflectance data obtained from the panels were verified using the supplementary statistical metrics above.

With the framework for obtaining the DN across the reflection panel now developed, chapter 5 will describe in more detail how these values were used to develop a radiometric calibration correction factor from the data obtained from the downwelling sensor. Using these software algorithms presented here, it was possible to extract the necessary DN from the reflectance panels and use them as inputs for the radiometric calibration correction model.
Chapter 5

Radiometric Map Correction Using Solar Spectral Irradiance Modeling

In this chapter, we focus on using the acquired data from the photosensor at different times during the data acquisition run to correct the mapping function that maps from the multispectral image DN to the surface reflectance map. In this work we are focused on correcting for the effects due to cloud cover using the photosensor data. In general, the photosensor bands need not match the spectral bands in the multispectral camera. The photosensor data collected at different wavelengths which relates to the solar spectral irradiance varies due to cloud cover. This effect also depends on the nature of the cloud cover, as well as the location and season of the year. Since this relation is not well established, we resort to modeling the solar spectral irradiance curve using the photosensor measurements and a ground truth measurement and using this to enforce our cloud cover induced radiometric map correction.

We are interested in fitting a function to model the continuous solar irradiance spectrum in the range between 400 nm and 900 nm. The reason for choosing this spectral range is because most commercial multispectral cameras operate inside the VIS-NIR range. Extending too far outside this range causes extrapolation errors when dealing with photosensor data. The radiometric calibration correction introduced in this study is not commonly used in literature which assumes that solar spectral irradiance is constant throughout the flight.

In section 5.1, we describe our effort to model the solar spectral irradiance using the data measured by the VIS-NIR spectrometer which is used as our ground truth data. In addition
to modeling the ground truth solar spectral irradiance using a polynomial function, we also developed a multilinear regression model to best estimate the solar spectral irradiance curve using the photosensor data.

5.1 Solar Spectral Irradiance Modeling

The solar spectral irradiance curve is of great importance to the solar power generation industry. Therefore a lot of work has been done on understanding how it is effected by location, season, aerosols in the atmosphere, time of day, cloud coverage and other factors. Figure 5.1 shows the 3 different curves which are the basis for all spectral irradiance or radiance modeling: the energy curve for a black body at temperature 6000K (which estimates the energy radiated by our sun), the energy curve of the solar energy after traveling from the sun to Earth’s atmosphere and the solar energy curve after traveling through Earth’s atmosphere at sea level.

Figure 5.1: Plots of the solar spectral energy curve[17] at sea level, outside the atmosphere and one obtained by treating the Sun as a black body at a temperature of 6000k. The absorption zones resulting from the different constituents in the earth’s atmosphere can be seen in the curve corresponding to the sea level.
In our case, the photosensor data is being measured roughly at sea level, and so we expect it to follow the solid black curve in figure 5.1. Within the solid black curve, it can be seen that the key absorption bands seen are due to: gaseous absorbers in the ozone (O₃), the oxygen in the atmosphere (O²) and the water vapor absorption(H₂O)[17]. The atmospheric absorption has a large trough at 760nm, while the water vapor absorption has its largest absorption channel between 850nm-1000nm. The effects of ozone absorption is seen in particular between 500-700nm, which is also known as the Chappuis band. The attenuation seen in the Chappuis band depends upon the thickness of the ozone layer as can be seen in figure 5.2, which demonstrates the effects of ozone layer thickness of 0-10cm upon the attenuation band. Given that the ozone layer is thicker towards Earth’s poles and thinner towards the equator, it is expected that the ground truth instrument sees a moderately high absorption feature due to ozone layer thickness over high latitude regions like in Canada. This implies that the solar spectral irradiance curve will be strongly dependent upon the latitude and longitude of the location where it is measured.

![Figure 5.2: Plot of the Solar flux versus wavelength corresponding to different Ozone thickness values in the atmosphere taken from[17]](image)

The effect of cloud coverage on the solar spectral irradiance curve is of importance in the context of this current work. It is important to note that multiple sources have detailed that not all types of clouds have the same effect upon the energy in the solar radiative spectrum[17, 79]. In particular it is noted that the height, thickness and pressure in the cloud cover effects the water vapor absorption [17] and hence the solar irradiance curve. It should also be noted that the solar absorption band due to cloud coverage can be quite broadband compared to what is
expected from a water vapor column of equivalent height[17]. The effect of cloud coverage on the solar spectral irradiance curve has not received much attention in the literature. In contrast the inverse problem of estimating the cloud cover based on the changes in the solar irradiation power has been studied[79].

The key information to acknowledge is that after the initial radiometric calibration of the multispectral camera, the cloud coverage is expected to have the largest impact upon the calibration mapping function. The effect of the cloud coverage upon the solar spectral irradiance curve measured by the spectrometer is shown in the figure 5.3. In the figure 5.3a and 5.3b we can see RGB raspberry-pi images taken twenty minutes apart, while in figure 5.3c shows the ground truth solar spectral irradiance curve at the same times. The shape and power of the solar spectral irradiance curve is clearly affected by the cloud coverage. This will have an effect on the DN number in the images captured by the multispectral camera which needs to be corrected. The transient nature of the cloud coverage ensure that not all images captured by the multispectral camera are equally effected by this cloud coverage. Therefore continuous time synchronous photosensor data over the time span of the multispectral image data acquisition is hence needed.
5.1. **Solar Spectral Irradiance Modeling**

Figure 5.3: RGB images captured by Raspberry-pi imager at times \( t = 0 \) and \( t = 20 \) minutes in (a) and (b) with the plot of the solar spectral irradiance curve measured by the spectrometer at the same times in (c).

### 5.1.1 Experimental Setup and Assumptions

In our study, the Flame VIS-NIR spectrometer coupled together with a cosine corrector measurement head was used for ground truth data acquisition[80]. The spectrometer took absolute irradiance measurements with the cosine corrector after being calibrated with a halogen light-source with a resolution of 0.1nm. The spectrometer was able to take readings between 350nm and 1000nm, which satisfied the range requirements. An optimal exposure time of 9.5 ms
was chosen for all experiments such that the system would never reach saturation. The cosine corrector was chosen such that it had a FOV of 180° so that it matches our downwelling photosensor. During experimental measurements both the photosensors and the spectrometer were synced to measure once every 5s. It was assumed that the start time for both the spectrometer and the photosensors were the same, however in practice it is possible that the two instruments could be off by a maximum 2.5 seconds.

Figure 5.4a shows examples of the spectrometer measured solar spectral irradiance in \( \mu W/cm^2/\text{nm} \) and figure 5.4b shows a schematic of the system setup. Figure 5.4a shows two solar spectral irradiance curves taken on different days which clearly show differences due to the ozone absorption in the 500-600nm Chappuis band, the absorption feature due to mixed gasses at 760nm and the start of the absorption band due to water vapor at roughly 870 nm.

![Spectrometer Irradiance Measurement Outside](image1)

![Spectrometer Measurement System](image2)

Figure 5.4: Schematic showing the setup used to collect data using the Flame spectrometer in (b) and in (a) a representation of the solar spectral irradiance curves obtained using this setup at two different times.

The subsequent datasets captured by the photosensor and spectrometer coupled system had the following assumptions or restrictions applied to them. In our study the datasets were captured between June 16th and June 26th, this corresponds to the crop growing season in
southwestern Ontario. The time of day of measurement was restricted between 10am and 2pm, which is corresponds to the typical time when drones flights are conducted. The measurements were taken over 1 hour, with data collected every 5s apart. This generates 720 measurements per day. A total of 7200 measurements were obtained over a period of 10 days. It is assumed that both photosensor and the ground truth spectrometer were pointed accurately at the same region of the sky with both having a FOV of 180°. As mentioned earlier data from both systems are assumed to be perfectly time synced.

5.1.2 Solar Spectral Irradiance Modeling and Results

The method we used to model the ground truth solar spectral irradiance curve is presented in this section. The proposed method is made of four separate tasks as outlined in figure 5.5. It consists of (1) downsampling the ground truth, (2) estimating a polynomial to fit the ground truth, (3) developing a multilinear regression model to relate the photosensor current data to the estimated polynomial fit coefficients, and evaluating the solar spectral irradiance curve using the computed regression coefficients, (4) evaluating the performance of the analysis by comparing the downsampled solar spectral irradiation curve with one obtained using the computed regression coefficients.
Figure 5.5: Flow chart showcasing the 4 different steps undertaken in the current study to model the solar spectral irradiance curve using the current values measured by the photosensor. The first two steps involve downsampling the measured ground truth values measured by the spectrometer and then obtaining a polynomial curve fit to the downsampled values.
5.1. Solar Spectral Irradiance Modeling

5.1.2.1 Downsampling Measured Ground Truth Data

A typical data collection by the ground truth spectrometer is with a resolution of 0.1\textit{nm} over the spectral range from 350\textit{nm} to 1000\textit{nm}. This implies that a single solar spectral irradiance curve will have 6500 data points. Downsampling the ground truth was done in order to reduce the time complexity involved generating the fit polynomial for a data set of this size.

The downsampling was conducted using the resample function in the SciPy python library. This utilizes a Fast Fourier Transform (FFT) in order to best preserve the shape of the waveform when we downsample. The size of the downsampled data set was chosen based on the following arguments. Most of the spectral filters that can be found on the market have at best a minimum spectral bandwidth of 5\textit{nm}. With 5\textit{nm} chosen, we get a downsampled data set of size 100, however it did not preserve the original waveform. Hence we chose 200 data points in our downsampled data set which corresponds to a 2.5\textit{nm} resolution which did preserve the original waveform satisfactorily. This was verified over several spectral irradiance curves with an eye test to make sure it kept the desired spectral features.

5.1.2.2 Polynomial Fit for the Solar Spectral Irradiance Ground Truth Curve

A family of polynomials were tried to curve fit the ground truth data. The polynomials were taken from order $n=3$ to $n=7$. The maximum order of the polynomial was restricted to 7 as it is equal to the number of inputs coming in from the photosensor in our setup. The polynomials were generated using a least squares optimization method in MATLAB®. The fit coefficients and intercepts were saved to an excel file and used in the subsequent analysis.

The irradiance models follow the generalized formula in equation (5.1) below.

\[
L = a_0 \lambda^n + a_1 \lambda^{n-1} + \ldots + a_n
\]  

(5.1)

where $\lambda$ is wavelength in \textit{nm}, $L$ is spectral irradiance in $\mu\text{W}/\text{cm}^2/\text{nm}$, and $n$ is the order of the polynomial fit. In order to evaluate the accuracy of the model, a Root Mean Squared Error [RMSE] was computed by evaluating difference between the polynomial fit function and the downsampled ground truth values. The RMSE values for different order of polynomials computed for a given downsampled curve is shown in table 5.1. It shows that increasing the order of the polynomial, reduces the RMSE value. This holds true except for the polynomials
of order 6 and 7. The general shape of the solar spectral irradiance curve shows that it starts with increasing slope and ends with a decreasing slope in the our region of interest. This is best captured by even ordered polynomials, unless overfitting occurs while using higher order polynomials.

<table>
<thead>
<tr>
<th>Polynomial Order</th>
<th>RMSE Value [uW/cm²/nm]</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>5.9093</td>
</tr>
<tr>
<td>4</td>
<td>5.7127</td>
</tr>
<tr>
<td>5</td>
<td>5.6721</td>
</tr>
<tr>
<td>6</td>
<td>5.4061</td>
</tr>
<tr>
<td>7</td>
<td>5.4021</td>
</tr>
</tbody>
</table>

Table 5.1: RMSE Metric comparison between Downsampled ground truth and estimated polynomials

5.1.2.3 Multilinear Regression Model Using Photosensor Data

Our objective was to model the solar spectral irradiance curve using 7 data points corresponding to different wavelengths measured by the photosensor. To achieve this we developed a multilinear regression model based on the assumption that the coefficients $a_0$ to $a_n$ in the polynomial fit function seen in the equation (5.1) could be derived as a linear function of the measured photosensor current values (measured in amperes). This assumption is supported by the fact that the photosensor currents generated in the photosensor are linearly related to the spectral solar optical power received at the photosensor.

The equation (5.2) shown below represents this linear relationship between the photosensor measured $I$ values and the coefficients in the fit function.

$$
\begin{pmatrix}
  a_0 \\
  a_1 \\
  \vdots \\
  a_n
\end{pmatrix}
= 
\begin{pmatrix}
  b_{0,0} & b_{0,1} & \cdots & b_{0,6} \\
  b_{1,0} & b_{1,1} & \cdots & b_{2,n} \\
  \vdots & \vdots & \ddots & \vdots \\
  b_{n,0} & b_{n,1} & \cdots & b_{n,6}
\end{pmatrix}
\begin{pmatrix}
  I_0 \\
  I_1 \\
  \vdots \\
  I_6
\end{pmatrix}
+ 
\begin{pmatrix}
  c_0 \\
  c_1 \\
  \vdots \\
  c_6
\end{pmatrix}
$$

(5.2)
5.1. Solar Spectral Irradiance Modeling

In the equation, \( b_{i,j} \) and \( c_i \) \((i\) goes from 0 to \( n - 1 \), where \( n \) is the order of the polynomial fit\) are the fit coefficients. For each value of \( n \) \((n = 3..7)\) a multilinear regression was performed to evaluate the coefficients. Using the computed regression coefficients from the photosensor data, the irradiance is evaluated over the spectral range of 400-900\(nm\) using equation 5.1.

5.1.2.4 Evaluating the Effectiveness of Our Solar Irradiation Curve Model

In order to determine the polynomial order which best fit the downsampled ground truth solar spectral irradiance curve and the one obtained using the photosensor values the following procedure was adapted. In our multilinear regression modeling the training-test split was 80/20 which meant 5760 datasets were used for the training and 1440 were used for evaluating the model. Appendix B also contains the violin plots which show the distribution for photosensor current inputs in amperes that were fed into the model. This clearly shows that our data was not biased, and that it contains equal amounts of datasets for cloudy and sunny weather respectively. It is important to note that we did not use a hyperparameter in our model because we used a residual sum of squares approach available in the sklearn linear_model.LinearRegression() function. Since the linear regression didn’t use a hyperparameter, only a training and testing dataset were used to evaluate the model.

After the model was trained, the computed polynomial coefficients were used to evaluate the irradiance \( L \) over the 400nm - 900nm range. In order to compare all the models the testing dataset was evaluated and the graphs were plotted against one another as shown in figure 5.6. Figure 5.6 compares the downsampled ground truth curve fit to the downsampled ground truth and the photosensor curve fit using the regression coefficients. From visual inspections we can notice that the key absorption bands (ozone, mixed gasses and water vapor) are not captured by the multilinear regression model as well as the curve fit downsampled data. This behavior is expected because we are using a single curve to span the entire spectral range. In general, the range of values along the y-axis seen in the figure 5.6 and figure 5.7 spans from \(10\mu W/cm^2/nm\) to a maximum of \(180\mu W/cm^2/nm\). This wide range implies that the photosensor should be able correctly capture a significant change in incoming solar irradiance. This is essential to correct the radiometric calibration map to account for changing cloud coverage.
Figure 5.6: Solar spectral irradiance curves with different scaling for the downsample ground truth (green), 6th order polynomial fit for the downsampled ground truth (orange) and the one obtained from photosensor current measurements via multilinear regression (blue) at four different times during a data acquisition run.
5.1. Solar Spectral Irradiance Modeling

Figure 5.7: Solar spectral irradiance curves with the same scale for the downsample ground truth (green), 6th order polynomial fit for the downsampled ground truth (orange) and the one obtained from photosensor current measurements via multilinear regression (blue) at four different times during a data acquisition run.

We chose the $R^2$ value to compare the photosensor derived polynomial coefficient values with the coefficient values obtained while fitting the downsampled ground truth curve. This comparison is shown in table 5.2 for polynomial orders from three to seven. From table 5.2 we can see that polynomials fit with orders of $n=3,4,6$ performed better than polynomials of
orders $n=5$ and 7. This is likely due to the fact that most of the solar spectral irradiance curves between 400-900nm start with an increasing slope and end with a decreasing slope, which is better approximated by polynomials of even order rather than odd. The polynomial order $n=3$ is the only exception to this in which the $R^2$ value is higher than all the rest, however this order of polynomial is also a very crude approximation of the original solar spectral irradiance curve which is not captured by the $R^2$ metric.

<table>
<thead>
<tr>
<th>Polynomial Order</th>
<th>$R^2$ Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.9595</td>
</tr>
<tr>
<td>4</td>
<td>0.9015</td>
</tr>
<tr>
<td>5</td>
<td>0.8548</td>
</tr>
<tr>
<td>6</td>
<td>0.9035</td>
</tr>
<tr>
<td>7</td>
<td>0.5693</td>
</tr>
</tbody>
</table>

Table 5.2: $R^2$ Metric for Estimated Photosensor Polynomial Coefficients vs. Ground Truth Estimated Polynomial Coefficients

Similarly, the RMSE value was computed by comparing the downsampled ground truth [original curve with 200 points and not the polynomial fit curve] and the constructed solar spectral irradiance model using for the photosensor measurements. Likewise the RMSE was computed by comparing the original downsampled fit curve and the constructed solar spectral irradiance model using for the photosensor measurement. These results are summed up in table 5.3. It shows that the photosensor current based model was a closer approximation to the downsampled curve fit rather than to the downsampled ground truth. This is not surprising since the photosensor irradiance model was trained with the downsampled curve fit polynomial coefficients rather than the original downsampled data. This produced an increased error in the photosensor irradiance output plots which was expected due to the basis assumed in our original model design.

It can be seen that higher order polynomials are able to estimate the solar spectral irradiance with a higher degree of accuracy. In order to select a polynomial order to proceed to the final processing step the following factors were considered:
5.2 Radiometric Correction Model for Cloud Coverage

Table 5.3: RMSE Metric comparison between photosensor evaluated solar spectral irradiance and the curve fit downsampled solar spectral irradiance or the downsampled solar spectral irradiance [uW/cm²/nm]

<table>
<thead>
<tr>
<th>Polynomial Order</th>
<th>RMSE To Curve Fit Ground Truth</th>
<th>RMSE To Downsampled Ground Truth</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>5.2696</td>
<td>7.9177</td>
</tr>
<tr>
<td>4</td>
<td>5.2738</td>
<td>7.7749</td>
</tr>
<tr>
<td>5</td>
<td>5.2781</td>
<td>7.7481</td>
</tr>
<tr>
<td>6</td>
<td>5.2919</td>
<td>7.5651</td>
</tr>
<tr>
<td>7</td>
<td>5.2928</td>
<td>7.5629</td>
</tr>
</tbody>
</table>

- The ability for the photosensor currents to estimate the curve fit polynomial, which is defined by the greatest R² metric.

- The ability for the photosensor currents to estimate the downsampled solar spectral irradiance model, which is defined by the lowest RMSE value.

A polynomial of order 6 was chosen to proceed to the next stage, since it was able to best approximate the downsampled spectral irradiance [not the fit curve] while maintaining a highest R² value in estimating the downsampled curve fit polynomial. It was also chosen since it was of an even order polynomial as well since the spectral irradiance plots also behave approximately like an even order polynomial.

5.2 Radiometric Correction Model for Cloud Coverage

With the ability to model the solar spectral irradiation curve established in the previous section, we were ready to develop a way to correct the radiometric calibration map to account for cloud coverage. In our experiment we imaged dual Spectralon reflectance panels using our multispectral camera for a period of 1 hour each day during the course of our study. In parallel, synchronized photosensor data was obtained. The reflectance of the panel is known and remains unchanged during the course of the measurements. This implies that in the equation, $\rho(\lambda) = DN(\lambda) \alpha(\lambda)$ which express the radiometric mapping. We know the $\rho$ value as well as the
DN value measured by the multispectral camera. The $\rho$ value remains constant while the DN value will change depending upon the incoming solar irradiance. With DN and $\rho$ both known in the equation, we can find the correct radiometric calibration map coefficient $\alpha$ to map each image during our measurements. However, in reality if we were imaging in a farm, the $\rho$ values will not be known as before like in our setup with the reflectance panels, hence we cannot find the correct radiometric mapping coefficient. The approach we are going to develop is based on the idea that we can estimate the correct radiometric mapping coefficient at each instant of time using the initial radiometric calibration coefficient and the solar spectral irradiance data evaluated from the photosensor measurements. This is possible in our setup because we know that $\rho$ remains constant regardless of cloud coverage. Since the DN (when under set gain and exposure conditions) is related to the incoming irradiance to the scene, a change in incoming irradiance due to cloud coverage will result in a change in DN which is not reflected by the original mapping function.

We will use a linear regression model and a multilinear regression model which fits the correct radiometric mapping coefficient at each time instant (known value in our case) with the initial radiometric mapping coefficient at time $t=0s$ and the solar spectral irradiance data computed using the measured photosensors currents and the multilinear regression coefficient values computed earlier. The choice for the linear model is motivated by the assumption that each radiometric mapping coefficient is affected only by its own spectral channel, whereas the multilinear model assumes that the radiometric mapping coefficient is affected by all the spectral channels.

In order to train the radiometric correction model, time series image data was captured with the multispectral camera focused on both the reflectance panels along with the time synced photosensor measurements. The model was trained across a week of data, in which 720 datasets were collected per day which implies that 5040 datasets were taken in total. In order to train the subsequent linear and multilinear models, the dataset was split 80/20 which gave 4032 datasets for the training and 1008 for the testing section. All of the days on which a datasets were captured had noticeable cloud coverage was observed during an otherwise sunny day. This was necessary in order to keep the distribution of sunny versus cloudy data within the dataset even. The algorithm was trained with a mixture of days between July and August.
5.2. Radiometric Correction Model for Cloud Coverage

5.2.1 Linear Model for Radiometric Correction

The linear model was created off the assumption that the spectral reflectance mapping coefficient was now dependent upon time which can be seen in equation (5.3) given below:

\[ \rho(\lambda) = DN(\lambda) \alpha(\lambda, t) \]  \hspace{1cm} (5.3)

Further expanding upon the generalized time dependent equation, we develop a linear model for the correction of the time dependent mapping coefficient \( \alpha(\lambda, t) \) given by the following equation (5.4):

\[ \alpha(\lambda, t) = a_0 \alpha(\lambda, t = 0s) + a_1 \left( \sum_{\lambda_1}^{\lambda_2} L_t(\lambda) - \sum_{\lambda_1}^{\lambda_2} L_{t=0s}(\lambda) \right) \]  \hspace{1cm} (5.4)

In the above equation, the initial radiometric calibration coefficient is represented by \( \alpha(\lambda, t = 0s) \). \( \sum_{\lambda_1}^{\lambda_2} L_t(\lambda) \) and \( \sum_{\lambda_1}^{\lambda_2} L_{t=0s}(\lambda) \) are the sum which represent the evaluated irradiance at time zero and \( t \) over the different camera spectral channels.

5.2.2 Multilinear Model

In contrast to the linear model, the multilinear model assumes the mapping coefficient corresponding to each wavelength is dependent on its value at other wavelengths. In this model all information from the beginning mapping coefficients for all bands of the multispectral camera and the difference in irradiance sums over the spectral channel at each time are shared. This was motivated by upon the fact that when a cloud passed over the target scene, the entire irradiance plot shifted downwards as seen in figure 5.3c. This implies that some information contained in other close wavelengths could be used to estimate changes in input irradiance. Equation (5.5) shows the difference in the sum of irradiance gathered by the downwelling photosensor as a function of the specific multispectral cameras center bandwidth.

\[ L_{\text{difference}}(\lambda_{c_n}) = \sum_{\lambda_1}^{\lambda_2} L_t(\lambda) - \sum_{\lambda_1}^{\lambda_2} L_{t=0s}(\lambda), \]  \hspace{1cm} (5.5)

where \( \lambda_1 \) and \( \lambda_1 \) are the bandwidths of the spectral filter for the multispectral camera of center wavelength \( \lambda_{c_n} \).
Equation 5.6 shows the multilinear model which was created using both the initial mapping coefficients and the difference between the irradiance sums.

\[
\begin{pmatrix}
\alpha_{\lambda_0}(t) \\
\alpha_{\lambda_1}(t) \\
\vdots \\
\alpha_{\lambda_n}(t)
\end{pmatrix} = 
\begin{pmatrix}
a_{0,0} & a_{0,1} & \cdots & a_{0,n} \\
a_{1,0} & a_{1,1} & \cdots & a_{1,n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{n,0} & a_{n,1} & \cdots & a_{n,n}
\end{pmatrix}
\begin{pmatrix}
\alpha_{\lambda_0}(t = 0) \\
\alpha_{\lambda_1}(t = 0) \\
\vdots \\
\alpha_{\lambda_n}(t = 0)
\end{pmatrix} + 
\begin{pmatrix}
b_{0,0} & b_{0,1} & \cdots & b_{0,n} \\
b_{1,0} & b_{1,1} & \cdots & b_{1,n} \\
\vdots & \vdots & \ddots & \vdots \\
b_{n,0} & b_{n,1} & \cdots & b_{n,n}
\end{pmatrix}
\begin{pmatrix}
L_{\text{difference}}(\lambda_{c_0}) \\
L_{\text{difference}}(\lambda_{c_1}) \\
\vdots \\
L_{\text{difference}}(\lambda_{c_n})
\end{pmatrix}
\] (5.6)

5.2.3 Results and Discussion

In order to quantify the performance of both the linear model and the multilinear model the RMSE and the $R^2$ metric values were used. In particular, the $R^2$ value was used to compare the closeness of the known correct radiometric mapping coefficient values to the ones derived from the model. The RMSE values were used to compare the closeness of the final reflectance maps to the known reflectance of the Spectralon panels. Table 5.4 below shows the $R^2$ metric for the time dependent mapping estimation across all bands for both the linear and multilinear radiometric correction models. It is clear to see that there is a stark difference between the performance within the $R^2$ metric when comparing the linear model to the multilinear model. In particular, the ability of the multilinear model to estimate the ground truth mapping coefficient is much better since it is using information from all wavelength bands. This could be because the cloud coverage had an equal effect upon the solar spectral irradiance curve across a broad range of wavelengths. This effect depends upon the type of cloud coverage might not hold true for all weather conditions. However, having this in the model allows for us to handle all types of cloud coverages.

It is also possible to see within the respective models that there is a gradual decrease in accuracy when going from the VIS wavelength regime to the NIR wavelength regime. There could be 2 possible contributing factors to this:

- The original photosensor data had many bands in the VIS wavelength regime and not as many inside the NIR wavelength regime. This could fixed in the redesign of the photosensor.
5.2. Radiometric Correction Model for Cloud Coverage

<table>
<thead>
<tr>
<th>Multispectral Camera Wavelength[nm]</th>
<th>R² Value Linear Model</th>
<th>R² Value Multilinear Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>464</td>
<td>0.6900</td>
<td>0.8240</td>
</tr>
<tr>
<td>542</td>
<td>0.7906</td>
<td>0.8996</td>
</tr>
<tr>
<td>639</td>
<td>0.6708</td>
<td>0.9002</td>
</tr>
<tr>
<td>669</td>
<td>0.5511</td>
<td>0.8884</td>
</tr>
<tr>
<td>708</td>
<td>-0.9725</td>
<td>0.5212</td>
</tr>
<tr>
<td>800</td>
<td>-0.1234</td>
<td>0.7328</td>
</tr>
<tr>
<td>845</td>
<td>-1.098</td>
<td>0.6208</td>
</tr>
</tbody>
</table>

Table 5.4: R² Metric for Linear or Multilinear Model Radiometric Coefficient Estimation

- This could be because during the data collection reflection from artifacts around the panel could make the reflectance maps unreliable. This effect was minimized by retaking datasets which had this issue of unwanted reflection from surrounding materials being visible on the Spectralon panels. While this effect was reduced, it is still possible to see a small intensity gradient across the reflection panels. Creating an alternative test setup in which the Spectralon panels are hanging would minimize this effect further.

Following the estimation of the time dependent mapping coefficient, the reflection across each desired reflectance panel per camera wavelength was calculated. The reflectance of each panel was computed for the time dependent cases: using the linear and multilinear models and using the original mapping coefficient and the respective RMSE’s were calculated as well. Below table 5.5 shows the RMSE for all reflectance maps generated when compared to the Spectralon reflectance panel ground truths. It is possible to see from the results that the RMSE of both the linear model and the multilinear model with time information outperform the original mapping function which assumes constant illumination throughout the test dataset. As expected from the time dependent reflection correction factor analysis, the multilinear model also outperformed the linear model by a significant degree in most bands.

It is possible to visualize the generated reflectance map values for the original mapping, the linear time series mapping and the multilinear time series mapping. Figure 5.8 below shows the probability density functions which compare the multilinear vs the original mapping data. The
true reflectance of the panel is shown by the red vertical dashed line which is the value that the multispectral camera should report in ideal conditions. It is clearly seen that from the original dataset, there is a sharpening of the predicted reflectance range while using the multilinear time dependent mapping coefficient. From the probability distribution functions ranges, it is possible to see that there is a correlation between the lower RMSE values seen in table 5.5 and the graphs shown in figure 5.8.

<table>
<thead>
<tr>
<th>Camera Wavelength[nm]</th>
<th>Reflectance Panel</th>
<th>RMSE Standard</th>
<th>RMSE Linear</th>
<th>RMSE Multilinear</th>
</tr>
</thead>
<tbody>
<tr>
<td>464</td>
<td>17.18</td>
<td>1.74</td>
<td>1.42</td>
<td>1.03</td>
</tr>
<tr>
<td>542</td>
<td>17.67</td>
<td>4.21</td>
<td>2.05</td>
<td>1.55</td>
</tr>
<tr>
<td>639</td>
<td>18.24</td>
<td>2.69</td>
<td>2.60</td>
<td>1.53</td>
</tr>
<tr>
<td>669</td>
<td>18.24</td>
<td>3.13</td>
<td>3.12</td>
<td>1.61</td>
</tr>
<tr>
<td>708</td>
<td>73.03</td>
<td>4.33</td>
<td>4.32</td>
<td>1.95</td>
</tr>
<tr>
<td>800</td>
<td>74.14</td>
<td>4.13</td>
<td>3.88</td>
<td>1.88</td>
</tr>
<tr>
<td>845</td>
<td>74.96</td>
<td>5.73</td>
<td>5.72</td>
<td>2.45</td>
</tr>
</tbody>
</table>

Table 5.5: RMSE Metric for Standard, Linear or Multilinear Model Reflectance Mapping function given spetralon panel ground truth
5.2. Radiometric Correction Model for Cloud Coverage

Figure 5.8: Plots showing the probability density function computed for the reflectance data captured by the multispectral camera at different specific wavelength bands. The orange curve corresponds to before radiometric correction while the blue one is after using multilinear regression based radiometric correction. The known reflectance value of the panel is marked by the red dashed line.

In order to verify that the mapping function operated correctly, the multispectral reflectance
map was generated for an image within the 542nm band. Referring back to the raspberry pi images shown in figure 5.3a,5.3b and their respective solar spectral plots shown in 5.3c, the reflectance maps corresponding to the times shown in those figures were obtained. The resultant reflectance maps are shown in figure 5.9 below. Figure 5.9 shows the visual difference between having a time dependent correction factor versus using the original mapping function evaluated at time $t=0$. Some details which are important to note however is that some information between the original image and the corrected image are lost. In particular the shadowed object on the brick wall behind the reflection panels are not able to be recovered with this correction factor.
Figure 5.9: Reflectance maps corresponding to the center wavelength of 542 nm at times 0 min in (a) and uncorrected one corresponding to $t = 20$ minutes in (b). The reflectance maps after enforcing correction obtained by using linear regression and multilinear regression methods to the $t = 20$ minutes data in (c) and (d) respectively.

5.3 Conclusion

In conclusion in this chapter, the main objective of correcting the radiometric mapping function of the multispectral camera arising from changing cloud coverage was achieved. In order to do so the downwelling photosensor was first used to estimate the solar spectral irradiance, which was then used to modify the radiometric calibration map in each image taken by the multi-
spectral camera at each instant during the experiment. The solar spectral irradiance ground truth was captured by a spectrometer coupled with a cosine corrector measurement probe. This data was downsampled and fit by a of varying degree of polynomial with orders from $n=3$ to $n=7$. While this polynomial was not able to capture certain spectral features such as the ozone effects upon the Chappuis band, it was able to broadly represent the solar spectral irradiance curve with enough accuracy for our needs. Once the polynomial fit coefficients were obtained for the solar spectral irradiance curve, a multilinear regression method was used to estimate the polynomial coefficients using the photosensor data. Comparing both methods, it was evident that a curve generated from the downsampled ground truth data performed better than the polynomial derived from the photosensors current data, however they both retained most of the features present in the original downsampled ground truth. Considering the downwelling photosensor is only a lightweight low power unit, using it to approximate the solar spectral irradiance to sufficient accuracy and using it to correct radiometric maps has huge implications. In particular this offers a lightweight and low power option to correct many multispectral cameras for cloud coverage effects when their spectral bandwidths lies within the 400-900nm range.

The second part of this chapter consisted of developing a linear and multilinear time series model which were used to radiometrically correct the reflectance data captured by a custom multispectral camera under varying cloud conditions. In particular the multilinear model outperformed both the linear time series model and the original mapping model. Being able to calibrate images with the downwelling photosensor with minimal effort solves one of the largest problems with multispectral cameras in which they are not guaranteed to observe accurate reflectance data inside of changing illumination conditions. Testing of the developed algorithms on real farm data could not be conducted because of restrictions placed during the COVID lock downs. We see no reason why the developed methods would not work with real farm data.
Chapter 6

Conclusion

6.1 Summary

The work done in this thesis presents a dual photosensor system and a model to radiometrically correct multispectral image data as a function of time. The results provided by the thesis has a significant impact on one of the greatest challenges with multispectral cameras in small farms currently which is the inability to provide accurate reflectance data once the incoming illumination changes due to cloud coverage. With the work done in the thesis, multispectral cameras will be able to provide more reliable reflectance data. This could have large implications on the post processed parameters such as the LAI and NDVI’s which are used in other models as well.

Chapter 2 introduced relevant background material for the work throughout this thesis. In particular it covered multispectral cameras, their respective radiometric calibration models, image processing techniques for fixing hardware deficiencies of cameras, image object recognition techniques and identifying relevant statistical metrics for assessing the work done in the thesis.

Chapter 3 was focused on the photosensor design and assembly. It can mainly be split into 3 sections covering optical design, electrical design and software design. The optical design covered information such as lens choice, optical filtering choice and the multispectral photosensors which were used as the core for the upwelling and downwelling photosensor designs. The electrical analog to digital circuit design used in the photosensor was presented. It is
needed to convert the analog data taken by the multispectral photosensor and transform it into a signal which the main camera processor could understand. Finally the digital communication design involved the communication between the camera processor and a chosen microprocessor which handled all the incoming information from the multispectral photosensors. By the end of this chapter a working pair of photosensor assemblies were built and tested.

Chapter 4 introduced a method in order to radiometrically calibrate a multispectral camera in the field. The main goal was to be able to detect the novel dual panel method of calibrating the custom multispectral camera in a way which was faster than the current manual calibration procedure before flight. This also provided an algorithm which was later used to get reflectance data from Spectralon panels as a function of time. The main two methods which were compared were the background controlled and the uncontrolled background segmentation algorithms. The controlled background algorithm assumed that a uniform black background was present inside the calibration image while the uncontrolled background assumed otherwise. In general the controlled background algorithm was more accurate and consistent however the uncontrolled background algorithm was more flexible and required less setup beforehand.

Chapter 5 took the downwelling photosensor and the data from a spectrometer to obtain a model for a solar spectral irradiance curve which can be constructed from photosensor outputs alone. The solar spectral irradiance model was fit with a polynomial of order $n=6$, which showed a strong representation of the general solar spectral irradiance ground truth. It does not capture all the features like the ozone absorption band at 500-600nm or the mixed gas absorption band present at 760nm. However being able to model a general representation of the solar spectral irradiance meant that the photosensor system could be able to radiometrically calibrated any camera within 400-900nm. The second part of chapter 5 took the solar spectral irradiance plots generated by the photosensors and used them to radiometrically correct the custom multispectral camera for cloud coverage effects using a linear and multilinear regression time series method. The multilinear method always outperformed the linear method. It was then proven that by a correctly approximating the incoming solar spectral irradiance into the system, it was possible to correct the initial radiometric calibration mapping function as a function of time for cloud coverage.
6.2 Future Work

The immediate future work that we envision arising from the current completed work are listed below:

- Testing the developed protocol to correct for cloud coverage in a farm field setting and evaluating its effectiveness. This could not be completed because of COVID restrictions active in the current times.

- Revisit the assumption made in chapter 5 that the entire solar spectral irradiance has to be fit with a single polynomial. Dividing the entire spectral range into multiple blocks and using a different polynomial fit function for each will improve the accuracy of the solar spectral irradiance curve representation. The potential for this modification in improving the radiometric mapping correction is worth exploring.

- Revisit the test setup used during chapter 5 to reduce the effects of stray light upon the reflectance panels. This could involve hanging the reflectance panels instead of laying them on the ground like it is in the current setup.

- Change the custom multispectral camera from a relative radiometric calibration framework to one that uses absolute radiometric calibration. Using an absolute radiometric calibration framework for the multispectral camera with the irradiance data from the downwelling sensor would enable us to move away from the need to perform initial radiometric calibration every time before flight.

- Use the integrated upwelling photosensor already present in the current tested camera as a gain and exposure optimizing instrument. This could not be tested in the current study which focused on correcting for cloud effects. This upwelling photosensor could also potentially be used in radiometric calibration or correction.
Bibliography


Appendix A

PCB Design for the Dual Photosensor Instrument

A.1 Photosensor

The photosensor PCB design is an analog to digital design with 4 layers and special care was needed to be taken around isolating the analog traces to prevent crosstalk. This was also recommended inside the MCDC04 datasheet. Since both digital ground and analog grounds are present on the PCB, it was recommended to attach them via a 0 ohm resistor close to the ADC, which has been done as well. With a trace size of 10 mm, each trace with the Pixelteq VIS-NIR photodiode is insulated with a ground layer to prevent any crosstalk between the traces. The ADC MCDC04 is powered with a 3.3V line, as well as grounded. Both these traces were designed with a 17 mm line in order to account for the higher current expected into the ADC. Header J1 was used as a port to be attached to the peripheral board.

The main difference between the two photosensor boards is that the MCDC04 has hardware pins associated with the $I^2C$ slave address. It is necessary to encode each with the 4 possible binary options since 2 pins are used to differentiate between slave devices. The downwelling sensor is encoded with 11 and 00, while the upwelling sensor is encoded with a 01 and 10 for the slave address.
A.1.1 Downwelling and Upwelling Photosensor

(a) Downwelling Photosensor PCB Bottom layer 3D View

(b) Downwelling Photosensor PCB Top layer 3D View

(c) Upwelling Photosensor PCB Bottom layer 3D View

(d) Upwelling Photosensor PCB Top layer 3D View

Figure A.1: Spectrometer Measurements

A.2 Peripheral Board

The peripheral board is a 4 layer purely digital design in nature for the PIC18F45K50 microcontroller interface. There is other components present on this board that was used as a human interaction device for the camera. These extra section include a power on button, manual trigger button and volume down button which is used for flashing a system image to the
Snapdragon Q410 processor. There was an extra complexity as well since there is 3 different USB devices able to connect to the board. A USB hub IC was incorporated into the device to ensure that all the USB device were able to communicate with one another.

The PIC18F45K50 microcontroller is attached to an LED for a visual indication once the microcontroller has been programmed correctly. It is attach to 3 separate headers: USB 3.0 and 2 in order to connect to the photosensor boards. While the USB 3.0 hardware is being used, the serial interface is USB 2.0 since the hardware is backwards compatible.
Appendix B

Software and Analysis Details

B.1 USB Packet Return Format for Buffer Contents

<table>
<thead>
<tr>
<th>Buffer Contents</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffer[0]</td>
<td>Report ID</td>
</tr>
<tr>
<td>Buffer[1]</td>
<td>Command Echo</td>
</tr>
<tr>
<td>Buffer[2]</td>
<td>Sequence Number Echo</td>
</tr>
<tr>
<td>Buffer[3]</td>
<td>Sensor Firmware Revision</td>
</tr>
<tr>
<td>Buffer[4]</td>
<td>Configuration Register Echo</td>
</tr>
<tr>
<td>Buffer[5-36]</td>
<td>Diode Data</td>
</tr>
<tr>
<td>Buffer[37]</td>
<td>diode 1-4 read check</td>
</tr>
<tr>
<td>Buffer[38]</td>
<td>diode 5-8 read check</td>
</tr>
<tr>
<td>Buffer[39]</td>
<td>diode 9-12 read check</td>
</tr>
<tr>
<td>Buffer[40]</td>
<td>diode 13-16 read check</td>
</tr>
</tbody>
</table>

Table B.1: PIC18F45K50 Diode Data Response Packet Format
B.2 Matlab Code for Ground Truth Generation of Reflectance Panels

```matlab
start = 0;
stop = 28;

mkdir("tif3/464nm/ActiveContour");
mkdir("tif3/542nm/ActiveContour");
mkdir("tif3/639nm/ActiveContour");
mkdir("tif3/669nm/ActiveContour");
mkdir("tif3/708nm/ActiveContour");
mkdir("tif3/800nm/ActiveContour");
mkdir("tif3/845nm/ActiveContour");

% for the 464nm set the contrast is low so the image intensity has been
% scale by 3 to get better ground truth data
for i = start:stop
    image = imread("tif3/464nm/"+num2str(i)+"_464nm.tif");
    mask = ones(size(image));
    bwContour1 = activecontour(image*3,mask,3000);
    openedimage = imopen(bwContour1,ones(5,5));
    imwrite(openedimage,"tif3/464nm/ActiveContour/"+
        num2str(i)+"_464nm.tif");
end

% Apply it to the 542 image set as well
for i = 0:16
```

image = imread("tif3/542nm/"+num2str(i)+"_542nm.tif");
mask = ones(size(image));
bwContour1 = activecontour(image*1.5,mask,3000);
openedimage = imopen(bwContour1,ones(5,5));
imwrite(openedimage,"tif3/542nm/ActiveContour/"+...
num2str(i)+"_542nm.tif");
end

for i = 16:stop
image = imread("tif3/542nm/"+num2str(i)+"_542nm.tif");
mask = ones(size(image));
bwContour1 = activecontour(image*1.5,mask,3000);
openedimage = imopen(bwContour1,ones(5,5));
imwrite(openedimage,"tif3/542nm/ActiveContour/"+...
(i)+"_542nm.tif");
end

% apply to the 639 nm as well
for i = start:16
image = imread("tif3/639nm/"+num2str(i)+"_639nm.tif");
mask = ones(size(image));
bwContour1 = activecontour(image,mask,3000);
openedimage = imopen(bwContour1,ones(5,5));
imwrite(openedimage,"tif3/639nm/ActiveContour/"+...
num2str(i)+"_639nm.tif");
end

for i = 16:stop
image = imread("tif3/639nm/"+num2str(i)+"_639nm.tif");
mask = ones(size(image));
bwContour1 = activecontour(image*1.5, mask, 3000);
openedimage = imopen(bwContour1, ones(5, 5));
imwrite(openedimage, "tif3/639nm/ActiveContour/" + num2str(i)+"_639nm.tif");
end

% apply to 669 nm
for i = start:stop
image = imread("tif3/669nm/" + num2str(i)+"_669nm.tif");
mask = ones(size(image));
bwContour1 = activecontour(image*1.5, mask, 3000);
openedimage = imopen(bwContour1, ones(5, 5));
imwrite(openedimage, "tif3/669nm/ActiveContour/" + num2str(i)+"_669nm.tif");
end

% required the use of a sharpening filter to enhance edges
for i = start:stop
image = imread("tif3/708nm/" + num2str(i)+"_708nm.tif");
mask = ones(size(image));
bwContour1 = activecontour(imsharpen(image, 'Radius', 2, 'Amount', 3)*2, mask, 3000);
openedimage = imopen(bwContour1, ones(5, 5));
imwrite(openedimage, "tif3/708nm/ActiveContour/" + num2str(i)... +"_708nm.tif");
i
end

% one picture needed special inputs to work properly
for i = 24:24
B.2. MATLAB Code for Ground Truth Generation of Reflectance Panels

```matlab
image = imread("tif3/708nm/\"+num2str(i)+\"708nm.tif\";
mask = ones(size(image));
bwContour1 = activecontour(imsharpen(image*1.05,'Radius',5...,'Amount',10)*1.5,mask,5000);
openedimage = imopen(bwContour1,ones(5,5));
imwrite(openedimage,"tif3/708nm/ActiveContour/\"+num2str(i)...+\"_708nm.tif\";
i
im = cat(3, openedimage*2^16, image, image);
imshow(im)
end

% Required higher iterations than normal to come to a good enough
% conclusion

for i = start:stop
image = imread("tif3/800nm/\"+num2str(i)+\"800nm.tif\";
mask = ones(size(image));
bwContour1 = activecontour(image*3,mask,5000);
openedimage = imopen(bwContour1,ones(5,5));
imwrite(openedimage,"tif3/800nm/ActiveContour/\"...
+num2str(i)+\"_800nm.tif\";
end

% Required higher iterations than normal to come to a good enough
% conclusion

for i = start:stop
image = imread("tif3/845nm/\"+num2str(i)+\"845nm.tif\";
mask = ones(size(image));
```

bwContour1 = activecontour(image*3, mask, 5000);
openedimage = imopen(bwContour1, ones(5,5));
imwrite(openedimage, "tif3/845nm/ActiveContour/" + ...
num2str(i) + "_845nm.tif");
end
B.3 Violion Plots for Photosensor Input to Spectral Solar Estimation Model

Figure B.1: Violin Plots for Spectral Photosensor Inputs
# Curriculum Vitae

<table>
<thead>
<tr>
<th><strong>Name:</strong></th>
<th>Nicholas Mitchell</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Post-Secondary Education and Degrees:</strong></td>
<td>The University of Western Ontario</td>
</tr>
<tr>
<td></td>
<td>London, ON</td>
</tr>
<tr>
<td><strong>Degrees:</strong></td>
<td>2014 - 2018 B.E.Sc., Electrical Engineering</td>
</tr>
<tr>
<td><strong>Honours and Awards:</strong></td>
<td>Dean’s Honour List</td>
</tr>
<tr>
<td><strong>Awards:</strong></td>
<td>2014 - 2018</td>
</tr>
<tr>
<td></td>
<td>NSERC USRA</td>
</tr>
<tr>
<td></td>
<td>2017</td>
</tr>
<tr>
<td><strong>Experience:</strong></td>
<td>Canadian CubeSat Project - Ukpik 1</td>
</tr>
<tr>
<td></td>
<td>Communication and Electronics Subsystem Team Lead</td>
</tr>
<tr>
<td></td>
<td>The University of Western Ontario</td>
</tr>
<tr>
<td></td>
<td>2017-2020</td>
</tr>
<tr>
<td></td>
<td>Research Assistant</td>
</tr>
<tr>
<td></td>
<td>2018</td>
</tr>
<tr>
<td></td>
<td>The University of Western Ontario</td>
</tr>
<tr>
<td></td>
<td>Teaching Assistant</td>
</tr>
<tr>
<td></td>
<td>2018 - 2020</td>
</tr>
<tr>
<td></td>
<td>The University of Western Ontario</td>
</tr>
</tbody>
</table>
Publications:

Patents: