



Atmospheric Correction and Anomaly Detection of Remotely Sensed Hyperspectral Imagery

Andrew Thomas Young

In the fulfilment of the requirement for the degree of
Doctor of Philosophy

Centre for Excellence in Signal and Image Processing
Department of Electronic and Electrical Engineering
University of Strathclyde, Glasgow

October 16, 2017

Declaration

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

The copyright of this thesis belongs to the author under the terms of the United Kingdom Copyright Acts as qualified by University of Strathclyde Regulation 3.50. Due acknowledgement must always be made of the use of any material contained in, or derived from, this thesis.

Andrew Young
October 16, 2017

Acknowledgements

I would like to thank a number of people for their support and guidance throughout my PhD.

Firstly, I would like to thank Professor Stephen Marshall for supervising me over the last three and a half years. From my initial meeting to discuss the PhD to sorting out contracts issues, through to my viva he has always been available for a meeting or simply by email, whenever I needed help, or just needed to talk.

My second supervisor Dr Alison Gray, she has always been there to discuss my research, help generate ideas and guided me through submitting my first two journal papers.

Next, I would like to thank my initial industrial supervisor Dr Ainsley Killey from the now closed BAE Systems Advanced Technology Centre (ATC). He provided the building blocks for my PhD, and it was his vision that inspired me to take this path in my career. As well I would like to thank everyone that was involved in my work from the ATC, every one of you helped me through the initial stages of my PhD.

When Ainsley retired at the end of 2014, Dr Henry White from BAE Systems Military Air & Information stepped in to take up where Ainsley left. While we may not have met as much as we would have liked, your technical expertise has been a huge help.

Another major influence on my work was Dr Paul Murray. Having worked with him on a three-month summer placement right before my PhD began, he was the one to first introduce me to the world of image processing. During my studies, I have worked on a number of projects with Paul and he has always been there to support me with his technical expertise and friendship.

I would also like to thank all the staff and students at the University of Strathclyde that I have had the privilege of working with over the last few years.

My research would not have been possible without funding from BAE Systems and the Engineering and Physical Sciences Research Council, and I hope that my research outcomes have returned this effort.

Finally, on a personal note, I would like to say thank you to all my friends and family for their support and understanding, with loving gratitude to Rebecca for her support from the distance.

Abstract

The work presented in this thesis focuses on improving the anomaly detection process for remotely sensed hyperspectral imagery. This process is split up into three main sections; data reduction, atmospheric correction and anomaly detection.

The final stage of this anomaly detection process is the actual anomaly detection algorithm. The initial contribution looks at developing a new type of anomaly detection algorithm based on the Percentage Occupancy Hit-or-Miss Transform. Also, a process for trying to improve the existing Mahalanobis Distance technique for hyperspectral data is explained. Both techniques are then tested on two aerial hyperspectral images, and the results are compared with an established technique the Sequential Maximum Angle Convex Cone algorithm.

One of the preprocessing steps of the anomaly detection process is the atmospheric correction phase. In this thesis an interface is developed in MATLAB for the atmospheric modelling software MODTRAN, this interface is then used to find the key parameters that have the most effect on the atmospheric models produced. Having determined the key parameters of a MODTRAN atmospheric model, the models are then used to atmospherically correct eight hyperspectral images; four visible to near-infrared and four short wave infrared hyperspectral images. Two scene based approaches for atmospheric correction are also proposed that use known spectra extracted from the scene to produce an atmospheric transform. All three techniques are then evaluated against existing scene-based approaches, namely Internal Average Relative Reflectance and Dark Object Subtraction.

The final contribution focuses on the data reduction phase, images of a wind turbine blade with simulated erosion were taken using a near-infrared hyperspectral camera. By

analysing the images produced it was possible to determine the optimal bands necessary to detect each type of erosion. The greyscale images produced for the optimal bands were then compared with standard RGB camera imaging to determine if any more detail was shown in the hyperspectral images. Also by imaging the blade at varying light levels, it was possible to determine when this technique breaks down, however by performing some post-processing of the new data using a polynomial surface subtraction to flatten the images it was again possible to extract additional information from the hyperspectral images.

Contents

List of Figures	xi
List of Tables	xvii
Nomenclature	xix
1 Introduction	1
1.1 Organisation of Thesis	3
1.2 Original Contributions	6
2 Background and Related Work	8
2.1 Introduction	8
2.2 What is Hyperspectral Imaging?	9
2.2.1 Types of Hyperspectral Imaging	10
2.2.2 Multispectral vs Hyperspectral vs Ultraspectral	12
2.3 Mixing Model	13
2.3.1 Linear Mixing Model	14
2.3.2 Non-Linear Mixing Model	14
2.4 Data Reduction	15
2.4.1 Band Reduction	16
2.4.2 Pixel Reduction	17
2.5 Atmospheric Correction	19
2.5.1 Scene Based Approaches	20
2.5.2 Radiative Transfer Approaches	21

2.5.3	Statistical Scene Based Approach	21
2.6	Spectral Unmixing Algorithms	22
2.7	Anomaly Detection Algorithms	23
2.8	Practical Applications	24
2.8.1	Defence and Security	24
2.8.2	Precision Agriculture	25
2.8.3	Environmental Monitoring	26
2.9	Summary	27
3	Anomaly Detection in Aerial Hyperspectral Imagery	28
3.1	Introduction	28
3.2	Image Acquisition	29
3.3	Anomaly Detection	30
3.4	Mahalanobis Distance	31
3.4.1	2D vs 3D vs nD Cases	32
3.4.2	Find Centre of Data	34
3.4.3	Centre of Data Results	35
3.5	Modified Hough Transform Approach	36
3.6	Hit or Miss Transform Approach	39
3.7	Percentage Occupancy Hit or Miss Transform	42
3.7.1	Band Selection	44
3.8	Results and Analysis	46
3.8.1	Speed Test	49
3.9	Summary	49
4	Atmospheric Modelling Using MODTRAN for VNIR Imagery	51
4.1	Introduction	51
4.2	Hyperspectral Data	53
4.2.1	Details	53
4.2.2	MODTRAN Atmospheric Profile Description	55
4.2.3	MODTRAN Setup	56

4.3	MODTRAN 5.2.1 MATLAB User Interface	57
4.3.1	LTN File Explained	57
4.3.2	Format	58
4.3.3	Reading a Card	59
4.3.4	Reading an LTN File	60
4.3.5	Writing an LTN File	60
4.4	Batch Mode	61
4.4.1	Generate Batch of LTNs	61
4.5	User Input	63
4.5.1	Manual Input	64
4.5.2	Configuration File Input	64
4.5.3	GUI Input	65
4.6	Saving the Output Data	65
4.6.1	Read MODTRAN Output Files	65
4.7	Analysis	67
4.7.1	Initial Tests	67
4.7.2	MODTRAN Key Parameters	68
4.8	Results	79
4.9	Summary	79
5	Atmospheric Correction Using MODTRAN Radiative Transfer Mod-	
	els	81
5.1	Introduction	81
5.2	Hyperspectral Images	83
5.3	Atmospheric Correction	85
5.3.1	Scene-Based Approach	85
5.3.2	Radiative Transfer Approach	86
5.4	Atmospheric Effects	88
5.4.1	Visible and Near-infrared (VNIR)	89
5.4.2	Short Wave Infrared (SWIR)	90
5.5	Analysis	91

5.5.1	MODTRAN Atmospheric Modelling	91
5.5.2	Predict Atmospheric Transform from Known Spectra	92
5.6	Results	93
5.6.1	Spectral Angle Mapper (SAM)	93
5.6.2	Comparison of Techniques	94
5.7	Summary	96
6	Erosion Detection in Wind Turbines Using Hyperspectral Imaging	98
6.1	Introduction	98
6.2	Wind Turbine Blades	99
6.3	Methodology	102
6.3.1	Image Acquisition	102
6.3.2	Initial Imaging	102
6.3.3	Introduction of Damage Types	104
6.3.4	Further Imaging	105
6.4	Results and Analysis	107
6.4.1	Band Selection	107
6.4.2	Classification	107
6.4.3	Image Flattening	109
6.5	Summary	111
7	Conclusion and Further Work	114
7.1	Conclusion	114
7.2	Future Work	117
A	MODTRAN Variables	119
B	MODTRAN Card Formats	123
C	MODTRAN GUI	132
C.1	Read Card 1	132
C.2	Read Card General	133

C.3	Read LTN File	134
C.4	Write Card 1	134
C.5	Write LTN File	135
C.6	Generate Multiple LTN Files	138
C.7	Running MODTRAN	139
C.8	Manual Input	141
C.9	Configuration File Input	142
C.10	Read Output Files	143
D	MODTRAN Output Variables	145
E	MODTRAN Output Results	146
F	Atmospheric Correction Equation Derivation	148
	References	150
	Publications by Author	173
	Publications Under Consideration	175

List of Figures

1.1	Comparison between RGB and HS images [1]	1
1.2	Spectral processing flow chart [11]	2
1.3	Anomaly detection process	3
2.1	Anomaly detection flow chart	9
2.2	Composition of a hyperspectral image [11]	10
2.3	Electromagnetic spectrum	11
2.4	Number of bands for multispectral, hyperspectral, and ultraspectral	13
2.5	The measured radiance of a mixed pixel is the weighted sum of fractional abundances of each material present in the pixel in the LMM	14
2.6	The measured radiance of a mixed pixel is a non-linear mixture based in the multiple scattering effects of each material present in the pixel	15
2.7	Vegetation vs. non-vegetation NDVI	17
2.8	NDVI (green) and low albedo (blue) classification for various hyperspectral images	18
2.9	Pixel purity index 2D example	23
2.10	Detection of aircraft over San Diego, CA, USA [86]	25
2.11	Classification of various Japanese crop types [91]	26
2.12	Vegetation classification using NDVI [6]	26
3.1	False-colour representations of hyperspectral images (636nm, 555nm 460nm respectively for RGB)	30
3.2	Scatter plot for both hyperspectral images	31

3.3	Random scatter plots with outliers highlighted	32
3.4	Random 3D scatter plots with outliers highlighted	32
3.5	Plot of Mahalanobis distance for all pixels, includes line indicating standard deviation (Green) and variance (Red)	33
3.6	Outliers produced using peaks from Mahalanobis distance plot	33
3.7	Random scatter plots with mean (blue), median (green), interquartile mean (purple) and one-step Tukey's bi-weight (red)	35
3.8	Cartesian to polar coordinates	36
3.9	Scatter plot angles diagram	37
3.10	Span of angles for Moll Harris 18-5 1129 dataset	38
3.11	Histogram for each set of angles	38
3.12	Histogram for each set of angles using log scale	39
3.13	Local maxima for each histogram	39
3.14	Erosion, Dilation and HMT Diagrams	40
3.15	Scatter plot divided into cells	40
3.16	Moll Harris scatter plot for bands 8 vs 23 (493nm vs 578nm)	41
3.17	(a-d) Foreground, (e-h) Background structuring elements	41
3.18	Hit or Miss Transform output for each set of structuring elements	42
3.19	Combined Hit or Miss Transform output	42
3.20	Scatter plot divided into cells	43
3.21	Four states used to detect outliers; blue is foreground, red is background	43
3.22	Ratios for each condition	44
3.23	Left 5x5 blocks, Middle 10x10 blocks, Right 20x20 blocks	44
3.24	NDVI plot. Low value indicates non-vegetative pixel, high value indicates vegetative surface	46
3.25	Target locations	46
3.26	SMACC results	47
3.27	Mahalanobis distance results	47
3.28	Ground truth	48
3.29	Results for each algorithm for every cell size	49

3.30 Speed test for each algorithm on both datasets	49
4.1 Atmospheric effects on aerial hyperspectral imagery	51
4.2 Viewing and solar zenith angle	52
4.3 False-colour representations of hyperspectral images (636nm, 555nm 460nm respectively for RGB)	54
4.4 Initial set-up for Moll Harris data cube	56
4.5 LTN file for initial set-up	57
4.6 LTN file with important variables highlighted	57
4.7 Card 1 format from MODTRAN documentation [73]	58
4.8 Example of Card 1 string	59
4.9 Pseudocode for reading Card 1	60
4.10 Function for reading complete LTN file	60
4.11 Function for writing LTN file	61
4.12 LTN file containing multiple runs	62
4.13 Function for writing multiple LTN files	62
4.14 Complete process for each type of user input	63
4.15 Manual input function running in MATLAB command window	64
4.16 Configuration file for Moll Harris dataset	64
4.17 MATLAB GUI	65
4.18 MODOUT2 or FL7 file structure	66
4.19 Function for writing LTN file	66
4.20 MAT file contents	67
4.21 Wavelength vs total radiance for various materials	68
4.22 MODTRAN initial output with no variation	69
4.23 MODTRAN output by varying Day of Year. 1st Jan (Black), 1st Mar (Navy), 31st Mar (Light Green), 30th Apr (Blue), 30th May (Red), 29th Jun (Green), 29th Jul (Pink), 28th Aug (Yellow), 27th Sep (Dark Green), 27th Oct (Lilac), 26th Nov (Cyan), 26th Dec (Dark Cyan)	70

4.24	MODTRAN output by varying Time. 12am (Blue), 2am (Dark Green), 4am (Cyan), 6am (Dark Cyan), 8am (Navy), 10am (Red), 12pm (Green), 2pm (Black), 4pm (Pink), 6pm (Yellow), 8pm (Lilac), 10pm (Brown) . . .	71
4.25	MODTRAN output by varying Latitude. -90° (Black), -67.5° (Green), -45° (Red), -22.5° (Blue), 0° (Pink), 22.5° (Yellow), 45° (Dark Green), 67.5° (Lilac), and 90° (Brown)	71
4.26	MODTRAN output by varying Longitude. 0°W (Blue), 40°W (Lilac), 80°W (Brown), 120°W (Red), 160°W (Green), 200°W (Black), 240°W (Pink), 280°W (Yellow), 320°W (Dark Green)	73
4.27	MODTRAN output by varying altitude of observer. 0.2km (Blue), 0.5km (Red), 2km (Black), 10km (Green), 23km (Pink), 50km (Yellow)	74
4.28	MODTRAN output by varying IHAZE. No Aerosol (Blue), Rural - 23km (Red), Rural - 5km (Green), Maritime - 23km (Black), Urban - 5km (Pink), Tropospheric - 50km (Yellow), Fog1 - 0.2km (Dark Green), Fog2 - 0.5km (Lilac)	75
4.29	MODTRAN output by varying zenith angle. 135° (Blue), 150° (Red), 165° (Green), 180° (Black), 195° (Pink), 210° (Yellow), 225° (Dark Green)	76
4.30	MODTRAN output by varying MODEL. Tropical (Blue), MLS (Red), MLW (Green), SAS (Black), SAW (Pink), 1976 US (Yellow)	76
4.31	MODTRAN output by varying MODEL. Tropical (Blue), MLS (Red), MLW (Green), SAS (Black), SAW (Pink), 1976 US (Yellow)	77
4.32	MODTRAN output by varying H2OSTR. 0 g/cm ² (Blue), 1 g/cm ² (Red), 2 g/cm ² (Green), 3 g/cm ² (Black), 4 g/cm ² (Pink), 5 g/cm ² Yellow . . .	78
4.33	MODTRAN output by varying CO2MX. 300ppmv (Blue), 350ppmv (Red), 400ppmv (Green), 450ppmv (Black), 500ppmv (Pink)	78
5.1	Simulated transmittance spectra of atmospheric aerosols using Atmo- spheric Removal Program (ATREM) [136]	82
5.2	Atmospheric scattering	82
5.3	RGB image and ground truth for each selected ROI for VNIR imagery (450 - 1000 nm)	84

5.4	False colour image and ground truth for each selected ROI for SWIR imagery (0.95 - 2.5 μm)	84
5.5	Atmospheric correction process	86
5.6	Comparison between lab-based spectra and spectra measured in field for both VNIR and SWIR hyperspectral imaging	89
5.7	Spectral profile of each material on the same days for VNIR images. Green Perspex (Blue), Orange Perspex (Red), White Perspex (Green), Grey Ceramic (Purple), Green Ceramic (Cyan), Beige Carpet Tile (Black), Green Carpet Tile (Pink)	90
5.8	Spectral profile of each material on the same days for SWIR images. Green Perspex (Blue), Orange Perspex (Red), White Perspex (Green), Grey Ceramic (Purple), Green Ceramic (Cyan), Beige Carpet Tile (Black), Green Carpet Tile (Pink)	90
5.9	MODTRAN predictions	91
5.10	Material classification for 15th August VNIR image	92
5.11	Atmospheric transform calculation steps for VNIR imagery	93
5.12	Diagram for precision and recall for true and false negatives and positives	94
6.1	Main elements of a wind turbine blade [167]	99
6.2	Examples of damage types identified on a blade structure tested to failure by a large compressive load [171]	100
6.3	Images taken with HD digital camera using different methods (Source: Offshore Renewable Energy Catapult, 2015)	101
6.4	Hyperspectral camera set-up	102
6.5	RGB images of initial defects	103
6.6	Initial hyperspectral image with existing defects highlighted	103
6.7	Damage types	104
6.8	Hyperspectral image of turbine blade after defects have been added . . .	105
6.9	Hyperspectral images at different illumination levels	106
6.10	Hyperspectral images at different illumination levels	106
6.11	Average contrast between eroded and not eroded surface	107

6.12	1000 Lux	108
6.13	500 Lux	108
6.14	100 Lux	108
6.15	25 Lux	109
6.16	Leading edge surface estimation	110
6.17	1000 Lux after surface subtraction	110
6.18	500 Lux after surface subtraction	111
6.19	100 Lux after surface subtraction	111
6.20	25 Lux after surface subtraction	111
6.21	High-resolution images taken with Nikon D5500 camera with various zoom levels	112
C.1	Function for reading card 1	132
C.2	Function for reading general card	133
C.3	Function for reading complete LTN file	134
C.4	Function for writing card 1	134
C.5	Function for writing LTN file	137
C.6	Code used to create multiple LTN files	138
C.7	Code used to run multiple LTN files	140
C.8	Manual input function	141
C.9	Configuration file input function	142
C.10	Function to run LTN files in MODRTAN	144
E.1	Output results for initial set-up	147

List of Tables

2.1	Examples hyperspectral cameras	12
2.2	Types of atmospheric correction algorithms	20
3.1	Dataset details	30
3.2	SMACC and MD results for both datasets	47
4.1	Dataset details	54
4.2	Seasonal latitude surface temperature profile [140]	54
4.3	Atmospheric profile description at 1 km [141]	55
4.4	Profile independent variables [141]	55
4.5	Card 1 variables and format	59
4.6	MODTRAN settings	68
4.7	MODTRAN initial simulation settings	69
4.8	MODTRAN simulation settings	72
4.9	List of MODTRAN essential and non-essential parameters for low altitude VNIR imagery	79
5.1	Presence of objects in images at the 4 dates	84
5.2	Description of output file values	88
5.3	Important parameters for each VNIR image	91
5.4	Recall results for VNIR imagery	95
5.5	Percentage of targets found for VNIR imagery	95
5.6	Recall results for SWIR imagery	96
5.7	Percentage of targets found for SWIR imagery	96

6.1	Typical damage to load-carrying structural members of a wind turbine blade [171]	101
6.2	Erosion classification scheme (Source: Offshore Renewable Energy Cate- pult, 2015)	113
A.1	MODTRAN variable descriptions and options [176]	122
B.1	Format for MODTRAN Card 1 [176]	125
B.2	Format for MODTRAN Card 2 [176]	128
B.3	Format for MODTRAN Card 3 [176]	130
B.4	Format for MODTRAN Card 4 [176]	131
B.5	Format for MODTRAN Card 5 [176]	131
D.1	Description of output file values	145

Nomenclature

Roman Symbols

\vec{a}	Reference spectrum in Spectral Angle Mapper
$abs()$	Absolute value operator in computer science
AI	Corresponding abundance image for selected endmember in Sequential Maximum Angle Convex Cone
\vec{b}	Target spectrum in Spectral Angle Mapper
c	Tuning constant in One-Step Tukey's Biweight
E_k	Endmember in the linear mixing model
EM	Spectral profile of the endmember selected in Sequential Maximum Angle Convex Cone
f	Frequency or wavenumber
F_0	Solar flux at the top of the atmosphere
h_0	Original data set in Sequential Maximum Angle Convex Cone
h_1	Updated data set in Sequential Maximum Angle Convex Cone
$H(X)$	Shannon Entropy of the image X
$H(X, Y)$	Joint Entropy between images X and Y
k	Number of endmembers in the scene
L_{obs}	Radiance measured at the observer
m	Median pixel vector in One-Step Tukey's Biweight
$median()$	Median operator in computer science
$MD(x)$	Mahalanobis distance for the pixel vector x

$MI(X, Y)$	Mutual information for the images X and Y
n	Number of pixels in cell in Percentage Occupancy Hit or Miss Transform
N	Number of pixels in the image
p_{ij}	Coefficients of the basis functions for the i th and j th term
$poly55()$	Polynomial surface model in computer science
r	Distance from the origin in Hough Transform
R	Ratio produced by POHMT
t_d	Downward reflected solar scattering
t_u	Upward reflected solar scattering
v	Band value in One-Step Tukey's Biweight
w	Vector representing the system noise in the linear mixing model
x	Hyperspectral pixel vector
x_{IQM}	Interquartile mean pixel vector

Greek Symbols

α_k	Fractional abundance of each corresponding endmember
Γ	Estimated covariance of the background
θ	Angle between line and x-axis in Hough Transform
λ	Wavelength
μ	Mean pixel vector
μ_0	Cosine of the solar zenith angle
ρ	Surface reflectance

Acronyms / Abbreviations

2D	Two-Dimensional
3D	Three-Dimensional
ACE	Adaptive Coherence Estimator
ACORN	Atmosphere CORrection Now
AI	Abundance Image
APS	Average Pixel Spectrum

ATCOR	Atmospheric and Topographic Correction
ATREM	Atmospheric Removal Program
AVIRIS	Airborne Visible / Infrared Imaging Spectrometer
BAE	British Aerospace Engineering
BBODY_T[K]	Blackbody Temperature in MODTRAN output file
BG	Background
CCA	Convex Cone Analysis
CDOM	Coloured Dissolved Organic Matter
DEPTH	Optical Depth in MODTRAN output file
DIR_EM	Emissivity in MODTRAN output file
DOS	Dark Object Subtraction
DRCT_RFLT	Direct Ground Reflected Radiance in MODTRAN output file
DSTL	Defence Science and Technology Laboratory
EM	Endmember
ENVI	Environment for Visualizing Images
ESAM	Extended Spectral Angle Mapper
FG	Foreground
FLAASH	Fast Line-of-sight Atmospheric Analysis of Hypercubes
FN	False Negatives
FREQ	Wavenumber in MODTRAN output file
GMT	Greenwich Mean Time
GPS	Global Positioning System
GRND_RFLT	Total Ground Reflected Radiance in MODTRAN output file
GUI	Graphical User Interface
GUIDE	Graphical User Interface Development Environment
HATCH	High-accuracy ATmospheric Correction for Hyperspectral Data
HD	High Definition
HMT	Hit or Miss Transform

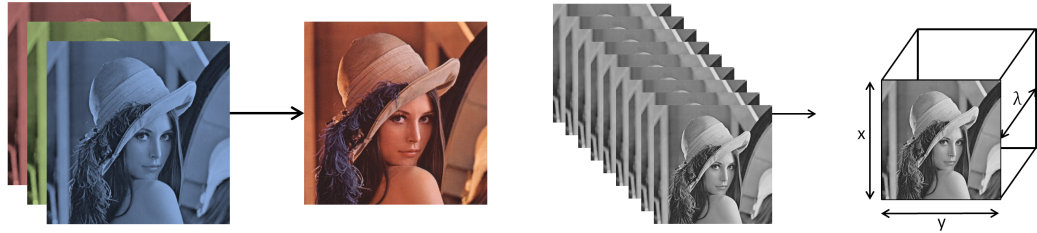
HSI	HyperSpectral Imaging
IARR	Internal Average Relative Reflectance
kNN	K-Nearest Neighbour
KS	Known Spectra
LMM	Linear mixing model
LWIR	Long Wave Infrared
MATLAB	Matrix Laboratory
MaxD	Maximum Distance
MD	Mahalanobis Distance
MH	Moll Harris
MI	Mutual Information
MLS	Mid-Latitude Summer atmospheric model
MLW	Mid-Latitude Winter atmospheric model
MODTRAN	MODerate resolution atmospheric TRANsmission
MSAM	Modified Spectral Angle Mapper
MWIR	Mid Wave Infrared
nD	N-Dimensional
NDVI	Normalised Difference Vegetation Index
NIR	Near-Infrared
OP7	Operation 7
PCA	Principal Component Analysis
PD	Porton Down
POHMT	Percentage Occupancy Hit or Miss Transform
PPI	Pixel Purity Index
PTH_THRML	Path Thermal Radiance in MODTRAN output file
QUAC	Quick Atmospheric Correction
REF_SOL	Reflected Solar Radiance in MODTRAN output file
RGB	Red, Green, and Blue
ROAVs	Remotely Operated Aerial Vehicles
ROI	Region of Interest

RXD	Reed-Xiaoli Detector
SAM	Spectral Angle Mapper
SAS	Sub-Arctic Summer atmospheric model
SAW	Sub-Arctic Winter atmospheric model
SCA	Spectral Correlation Angle
SD	Standard Deviation
SE	Structuring Element
SGA	Spectral Gradient Angle
SING_SCAT	Single Scatter Radiance in MODTRAN output file
SMACC	Sequential Maximum Angle Convex Cone
SNR	Signal-to-Noise Ratio
SOL_SCAT	Solar Scatter Radiance in MODTRAN output file
SOL@OBS	Solar Radiance at Observer in MODTRAN output file
SURF_EMIS	Surface Emission Radiance in MODTRAN output file
SWIR	Short Wave Infrared
T	Tropical atmospheric model
THRML_SCT	Thermal Scatter in MODTRAN output file
TOA_SUN	Top of Atmosphere (TOA) Sun in MODTRAN output file
TOT_TRANS	Transmission Total in MODTRAN output file
TOTAL_RAD	Total Radiance in MODTRAN output file
TP	True Positives
UAV	Unmanned Aerial Vehicle
UDRC	University Defence Research Collaboration
US	United States atmospheric model
USGS	United States Geological Survey
VCA	Vertex component analysis
VIS	Visible
VNIR	Visible and Near-Infrared

Chapter 1

Introduction

A standard RGB camera uses three bands each relating to a separate wavelength, red, green and blue (approx. 650, 510 and 475nm respectively). At each of these wavelengths, the camera measures the light intensity at that specific wavelength to produce a greyscale image. These three greyscale images are combined to produce the colour image taken by the camera (see Fig. 1.1). In a hyperspectral image, however, there are tens or even hundreds of bands all across the spectrum. These need not be limited to the visible spectrum; there are hyperspectral cameras currently available that can measure light in both the ultra-violet and infra-red regions of the spectrum.



Lena standard test image. Adapted from ©Playboy Magazine, 1972

Fig. 1.1: Comparison between RGB and HS images [1]

Therefore this type of imaging provides both spatial and spectral information and can be exploited in many different areas, e.g. defence and security [2, 3], precision agriculture for vegetation and crop monitoring [4, 5], environmental monitoring [6, 7] and food and drink quality control [8–10]. Aerial hyperspectral imagery for the purpose of remote sensing is a rapidly growing research area [11, 12]. There are three main areas

that aerial hyperspectral imaging is used for, these include Anomaly Detection, Target Recognition and Background Characterization [11]. One of the main applications for this technology is for the purpose of anomaly detection [2]. A generally used technique for detecting anomalies in a scene, when no prior information about the anomaly is known, is by looking at the distinct spectral features of the objects under surveillance and trying to differentiate them from surrounding materials. For the case of a camouflaged vehicle, the vehicle is deliberately designed to blend into the background grass and trees in the visible spectrum. However, by looking at spectral features in the near-infrared spectrum, such as the chlorophyll edge [13], the vehicle can be revealed. Due to the volume of data, capturing and subsequent processing of the data can be time-consuming. For many practical applications where cost is a constraint, it is necessary to determine the optimal number of bands that can be used to produce a satisfactory result.

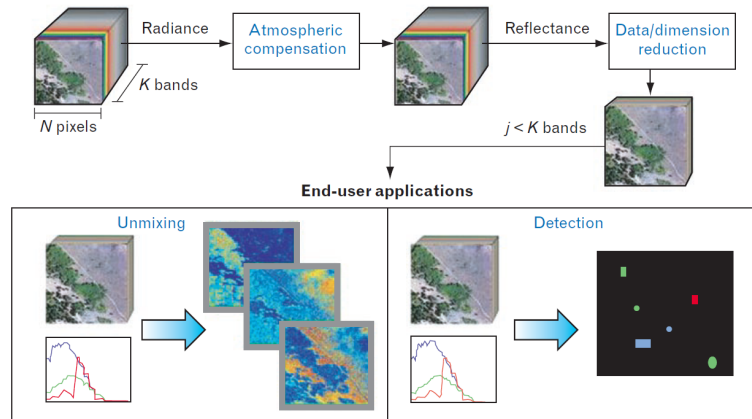


Fig. 1.2: Spectral processing flow chart [11]

Fig. 1.2 shows the spectral processing flow chart proposed by Shaw and Burke [11]. This process consists of three main steps Atmospheric Compensation, Data Reduction and Unmixing or Detection. The main aim of this thesis is the improvement of the anomaly detection process, therefore based on this spectral processing flow chart an anomaly detection process was proposed, see Fig. 1.3. Throughout this thesis each of these stages, is explored individually. The data reduction stage is explored when looking at erosion detection in wind turbine blades, various techniques are looked into

to determine the optimal set of bands to use that best show each type of erosion. The process of atmospheric correction is also explored first by using the atmospheric modelling software MODTRAN to determine the key parameters necessary to create an accurate atmospheric model for a hyperspectral image, this information is then used to atmospherically correct several visible to near-infrared (VNIR) and short wave infrared (SWIR) hyperspectral images. Finally the anomaly detection step is addressed through a new type of anomaly detection technique based on the percentage occupancy hit-or-miss transform (POHMT) [14] to find outliers in scatter plots created from hyperspectral images.

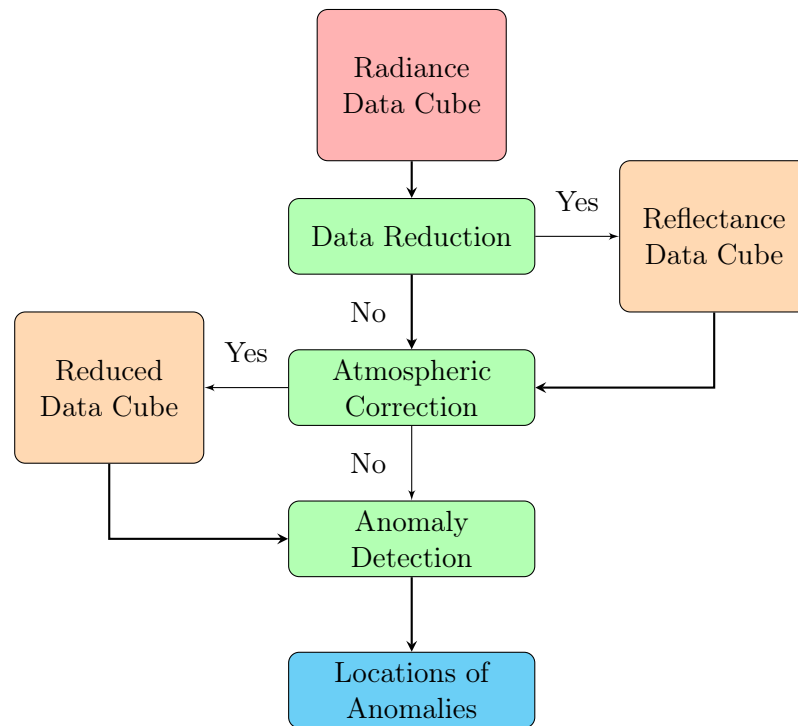


Fig. 1.3: Anomaly detection process

1.1 Organisation of Thesis

The remainder of the thesis is organised as follows:

- **Chapter 2:** This chapter is split into four main sections, the first section discusses hyperspectral imaging, what it actually is, how it varies from traditional types of

imaging, the various methods of acquiring it and what the format of the data looks like. The next section looks at some of the current anomaly detection techniques, which are split up into two main types: supervised and unsupervised. Following from this, the various atmospheric effects present in all aerial hyperspectral data are discussed, and what some of the current processes are for removal of these effects. Finally, this chapter ends with a summary of the practical applications that hyperspectral imagery has and the various fields it is used in.

- **Chapter 3:** In this chapter, various techniques are explored to find anomalies in visible to near-infrared hyperspectral images using scatter plots. All the techniques explored are unsupervised anomaly detection techniques which include the use of the Mahalanobis Distance, a modified Hough Transform, the standard Hit or Miss Transform and finally the Percentage Occupancy Hit or Miss Transform. The best of these techniques are then compared with traditional anomaly detection techniques using two hyperspectral data cubes with several targets located in each.
- **Chapter 4:** Atmospheric correction is an important step in the anomaly detection process, to accomplish this atmospheric correction it is necessary to create an accurate atmospheric model relating to the time, location and weather conditions of each image. This chapter first develops an interface for MODTRAN 5.2.1 in MATLAB, that can create a look-up table of atmospheric models which can be used for quick atmospheric correction of hyperspectral data rather than having to generate a new atmospheric model for each dataset. The interface is then used to explore the various parameters built into MODTRAN to determine which of these have the main effects on the atmospheric model.
- **Chapter 5:** In the previous chapter an interface for creating MODTRAN look-up tables was developed for MATLAB. Using these look-up tables a simplistic approach to atmospheric correction is developed by estimating the solar scattering contribution for each image, removing this from the image and then scaling the image based on an estimation of what a perfectly reflective surface would reflect to

produce a reflectance hyperspectral image. Another technique is also proposed by using known spectra from the scene. This works on the assumption that there will always be grass or tarmac in the images and uses spectra from the United States Geological Survey to estimate the atmospheric transform required to convert the radiance domain spectra to reflectance domain spectra. Both these techniques allow for the quick atmospheric correction of eight hyperspectral images (four visible to near-infrared and four short wave infrared) each of which contains targets placed throughout the image that have also been measured in a lab to determine their pure spectra. The results are then compared to more established scene-based techniques for atmospheric correction, i.e. Internal Average Relative Reflectance and Dark Object Subtraction. To compare the methods, a ground truth for the image was created to provide a quantitative value for recall and percentage of targets found.

- **Chapter 6:** There are various applications for hyperspectral imaging, which include remote sensing for security and defence, precision agriculture for vegetation and crop monitoring, and food, drink, and pharmaceuticals quality control. However, for the purpose of condition monitoring and erosion detection in wind turbine blades, the use of hyperspectral imaging is a relatively untouched area. Currently, this task is performed by rope-access technicians who visually inspect the wind turbine blades, remotely operated aerial vehicles with mounted RGB cameras, or long range cameras. In this chapter, with the use of a near-infrared hyperspectral camera, a new method for leading edge erosion detection is proposed. Realistic defects were added to a section of wind turbine blade and four hyperspectral images were captured in the laboratory, each at varying light levels.
- **Chapter 7:** The final chapter presents some concluding remarks on each chapter, along with possible ideas and suggestions for future work.

1.2 Original Contributions

It is believed that the following novel contributions have been made:

- Chapter 3 presents an improved technique for unsupervised anomaly detection for aerial hyperspectral imagery using the Percentage Occupancy Hit or Miss Transform (POHMT), through extensive testing of other techniques to find outliers in scatter plots it was shown that this technique was the best out of all explored. This technique is then compared with two established techniques, i.e. the Sequential Maximum Angle Convex Cone and the Mahalanobis Distance. It was found that using certain parameters with a pair of randomly selected bands the POHMT provided an improvement over the anomaly detection techniques tested.
- Chapter 4, presents a complete interface for the modelling of atmospheric effects present in aerial hyperspectral imagery based on the MODTRAN 5.2.1 atmospheric modelling software in MATLAB. Investigations are then undertaken to find the key parameters necessary to create an accurate atmospheric model. The results from this chapter show the parameters that have the most effect on the atmospheric model are the time of day, the day of the year, the latitude, the height of the observer and the aerosol model. Parameters such as the atmospheric profile, carbon dioxide (CO_2) mixing ratio, longitude and viewing zenith angle all have very negligible effects and can essentially be ignored in the atmospheric modelling calculations.
- Chapter 5, uses the interface developed in the previous chapter to propose a radiative transfer based atmospheric correction technique also two novel techniques that use known spectra from the scene to predict the atmospheric transform necessary to remove the atmospheric effects are proposed. The results show that for the visible to near-infrared imagery (400 - 1000nm) the new technique based on the MODTRAN atmospheric modelling code outperforms all the current scene based approaches tested and for the short wave infrared images (950 - 2500nm) the techniques using known spectra produce the best results.

- Chapter 6 presents a methodology for precise erosion detection in wind turbine blades using a short wave infrared hyperspectral camera. By the selection of optimal bands, it is possible to extract information relating to the depth of each level of erosion far beyond what is currently possible with standard imaging under ideal conditions. When the illumination conditions are varied from 1000 Lux to 25 Lux, a lot of the information is lost. However, by using a polynomial surface subtraction it is possible to again extract information relating to the depth of the erosion in all but the darkest of the images.

Chapter 2

Background and Related Work

2.1 Introduction

In the previous chapter, it was briefly discussed what hyperspectral imaging (HSI) is and what some of its many applications are. A process that can be followed when trying to detect anomalies in a hyperspectral image involves three main steps (see Fig. 2.1). These three steps are, the atmospheric correction step followed by a data reduction step where any preprocessing is performed on the data, e.g. bad bands are removed, geo-correction or noise removal. The final step is the anomaly detection step, this is where the data is processed using a specific algorithm and the output is a list of pixels in the image where an anomaly occurs.

The first section of this chapter explains in more detail what exactly HSI is. Firstly a comparison is made between traditional RGB cameras, spectroscopy and HSI, then the various regions of the electromagnetic spectrum that HSI cameras operate in, the subtleties between multispectral, hyperspectral and ultraspectral are discussed. In the next section, the issues relating to the inadequate spatial resolution on the ground causing mixed pixels is introduced, and the various models used to explain this behaviour. Following this the process of data reduction for the large volume of hyperspectral data by using either one of two methods; a band or pixel reduction approach is discussed. The next section discusses the effects the atmosphere has on the data and how there are many methods currently used to remove these effects using either a scene-based

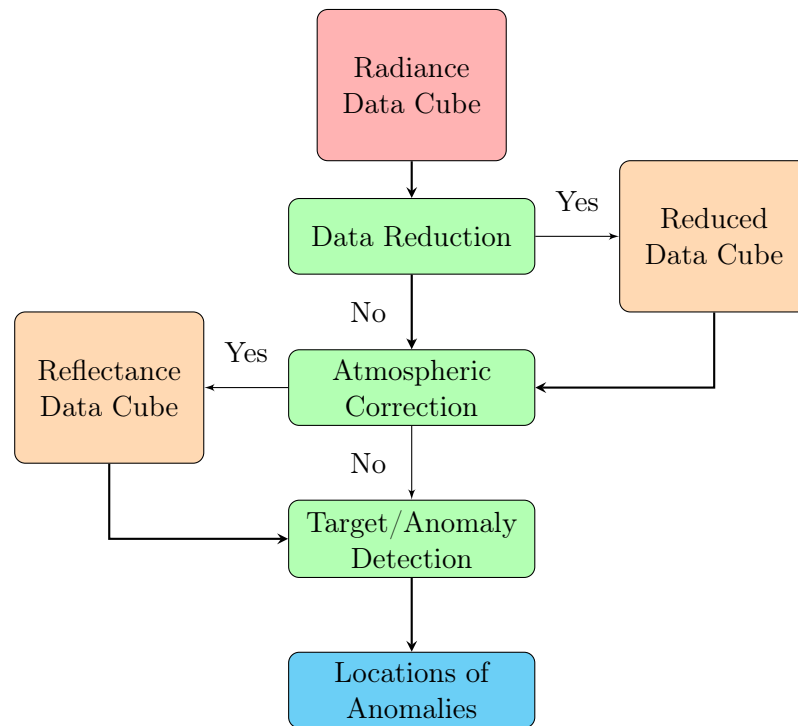


Fig. 2.1: Anomaly detection flow chart

or radiative transfer based approach. Spectral unmixing and anomaly detection is discussed in the next section, the difference between these two approaches is explained and the various currently used techniques are discussed. The final section gives an overview of some of the practical applications of aerial hyperspectral imagery for defence and security, precision agriculture, hydrology and environmental monitoring.

2.2 What is Hyperspectral Imaging?

Standard RGB cameras have a good spatial resolution but have a very limited spectral resolution by being limited to only three wavelengths, red, green, and blue (approximately 650, 510 and 475nm respectively). Hyperspectral imaging can provide both a good spatial and spectral resolution. The most commonly used technique for image acquisition is called the push broom method, this works by capturing a line of pixels at every wavelength, as the plane or unmanned aerial vehicle (UAV) moves along its flight path this builds up the image to produce a 3D hypercube, see Fig. 2.2 and for each

pixel in the scene a spectrum can be retrieved relating to the material covered by that pixel.

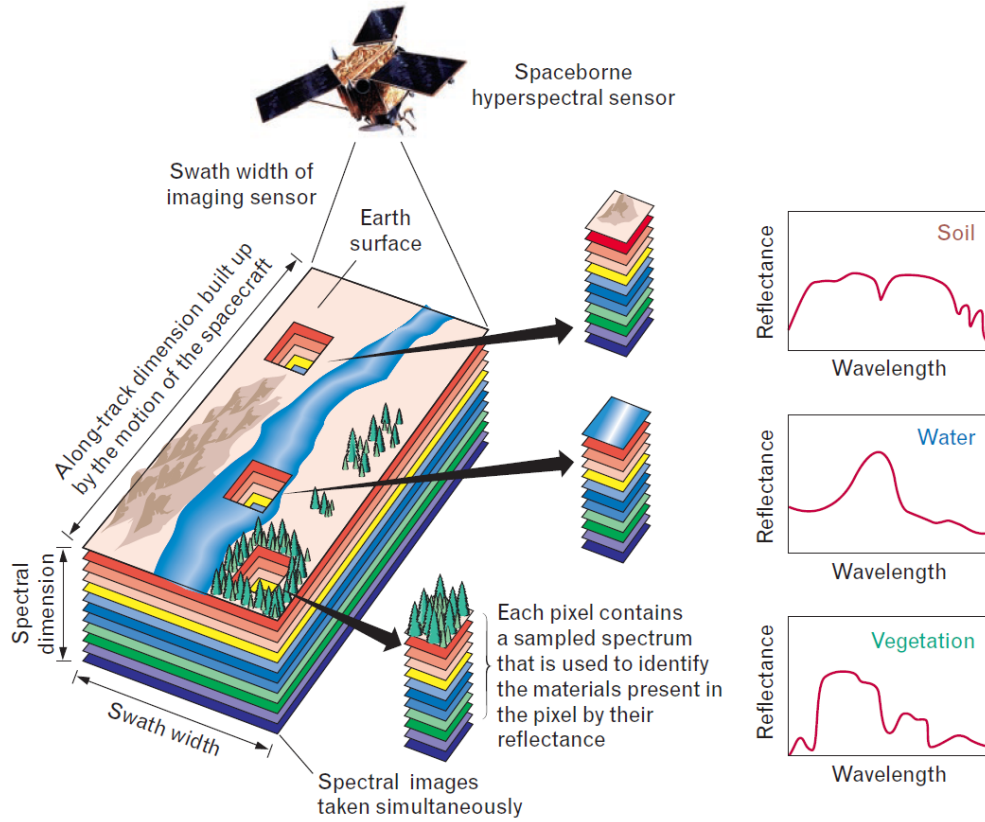


Fig. 2.2: Composition of a hyperspectral image [11]

Due to the altitude operated at for most aerial hyperspectral applications the spatial resolution on the ground can be anything from a few square centimetres to a few square meters, therefore there is usually more than one type of material in the pixel. This causes the produced spectra to be a linear combination of the fractional abundance of the related materials, these are called "mixed pixels" and can be modelled using either a linear or non-linear mixing model.

2.2.1 Types of Hyperspectral Imaging

There are various different types of HSI currently available, these include ultraviolet (UV), visible (VIS), near-infrared (NIR), short wave infrared (SWIR), mid wave infrared

(MWIR) and long wave infrared (LWIR) regions of the electromagnetic spectrum, see Fig. 2.3.

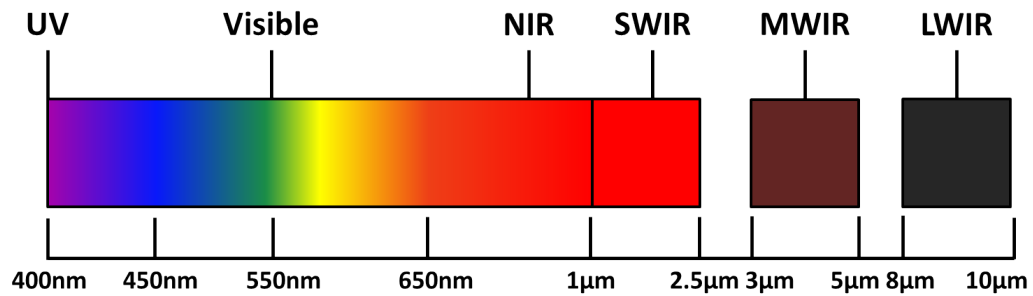


Fig. 2.3: Electromagnetic spectrum

Ultraviolet

The ultraviolet (UV) region of the spectrum occurs directly before (shorter wavelength than) the visible range that we see with our eyes. This region is usually divided into three regions UVA (315 to 400nm), UVB (280 to 315 nm), and UVC (100 to 280 nm) [15]. Some examples of the ultraviolet systems currently on the market are shown in Table 2.1.

Visible

The visible (VIS) region of the spectrum is the region of the electromagnetic spectrum that is 'visible' to the human eye (around 390 nm to 700 nm [16]). However, the shortest wavelength that most VIS systems currently available measure is only 400nm, see Table 2.1.

Near Infrared

The near infra-red (NIR) region of the spectrum begins as soon as the light is longer than detectable by the human eye, this is usually quoted as between 750 and 1400nm (see Table 2.1) and includes the main vegetation features so is a very important region for precision agriculture.

Short Wave Infrared

The short wave infrared (SWIR) region is directly after the NIR region and encompasses the wavelengths of about 1400 to 2500nm, see Table 2.1.

Mid Wave and Long Infrared

There are very few current systems readily available for measuring in both the mid wave infrared (MWIR) and long wave infrared (LWIR), the few systems that are available operate in the 3 to 5 micron and 8 to 12 micron ranges respectively. This is mainly due to the cost and complexity of these systems, a few examples systems are shown in Table 2.1.

Name	Wavelength Range	Manufacturer
Blue Eye	190-380nm	Innospec [17]
Hyperspec UV-VIS	250-500nm	Headwall [18]
Hyperspec VNIR	380-1000nm	Headwall [19]
HySpex VNIR-1800	400-1000nm	HySpex [20]
Green Eye	400-1000nm	Innospec [21]
Hyperspec NIR	900-1700nm	Headwall [22]
Red Eye 1.7	950-1700nm	Innospec [23]
Hyperspec SWIR	900-2500nm	Headwall [24]
HySpex SWIR-384	1000-2500nm	HySpex [25]
Red Eye 2.2	1200-2200nm	Innospec [26]
Black Eye	2900-4200nm	Innospec [27]
Hyperspec MWIR	3000-5000nm	Headwall [28]
Hyper-Cam MW	3000-5000nm	Telops [29]
Hyperspec LWIR	8000-12000nm	Headwall [28]
Hyper-Cam LW	7700-11800nm	Telops [29]

Table 2.1: Examples hyperspectral cameras

2.2.2 Multispectral vs Hyperspectral vs Ultraspectral

There are three types of spectral images; multispectral, hyperspectral and ultraspectral. A multispectral image will usually consist of no more than 20 bands, a hyperspectral image will consist of anything more than this to a few hundred and an ultraspectral image will consist of a few thousand, see Fig. 2.4. These labels are often irrelevant as they often have no relation to the image quality or spectral resolution. A hyperspectral

camera with 100 bands between the region of 400 and 1000nm could have a spectral resolution of 6nm, where a multispectral camera with 20 bands between the region of 950 and 1050nm could have a resolution of 5nm

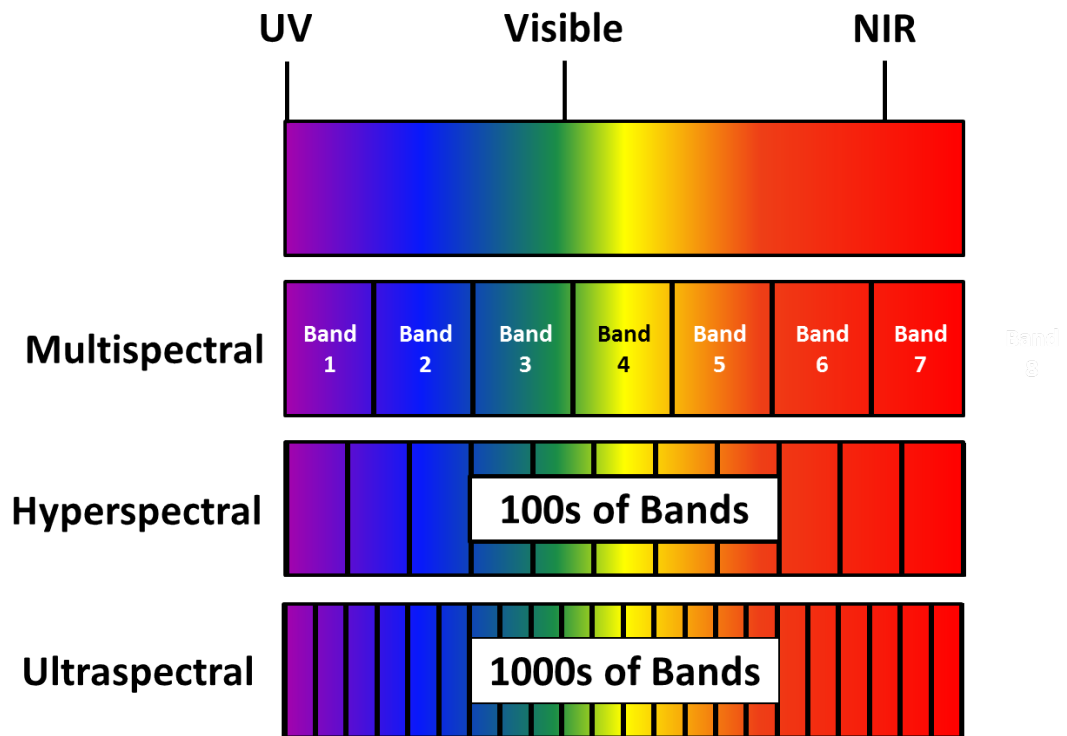


Fig. 2.4: Number of bands for multispectral, hyperspectral, and ultraspectral

2.3 Mixing Model

Spectral mixing occurs in most remotely sensed hyperspectral images, this is due to the finite spatial resolution of the sensors. In most aerial hyperspectral applications the sensor is at least several kilometres above the target area, thus producing a pixel size on the ground of several square metres. This will mean that there will be several distinct materials covered by each pixel, therefore all pixels in the image will have a varying amount of spectral mixing. Currently, the main mixing model that tries to explain this behaviour is the linear mixing model (LMM) [30].

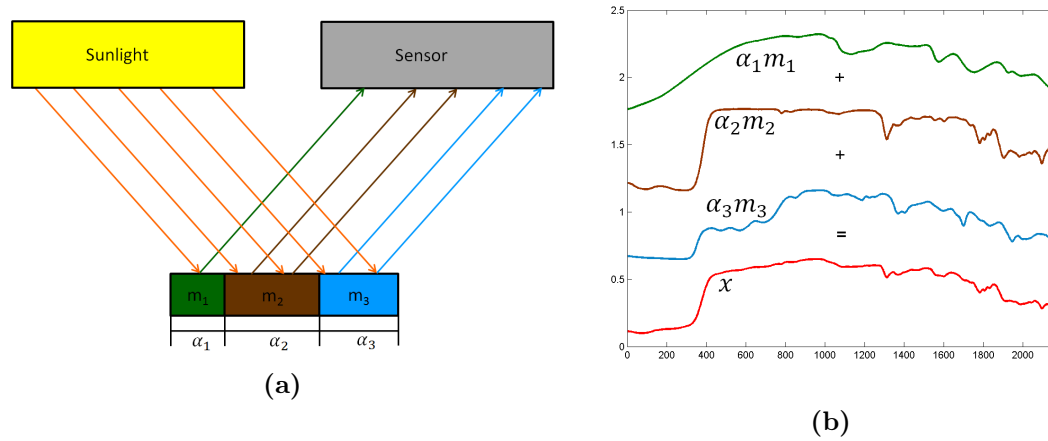


Fig. 2.5: The measured radiance of a mixed pixel is the weighted sum of fractional abundances of each material present in the pixel in the LMM

2.3.1 Linear Mixing Model

The LMM assumes that in any given scene there is a small number of spectrally distinct materials present. These materials, e.g. grass, mud and water, are called endmembers and the amount they appear in each mixed pixel are called their fractional abundances. Fig. 2.5a shows a pixel in a scene that is depicted in the resulting image by a mixture of three distinct materials, it is also assumed that each incident light photon only comes in contact with one material, i.e. there is no multiple scattering between materials. If the surface area proportion of each material is known, then it can be assumed that the resulting spectra will be a linear combination of the fractional abundances of each endmember, see Fig. 2.5b. Using this model each pixel in the scene can be expressed by $x = \sum_{i=1}^k \alpha_i E_i + w$ where k is the number of endmembers in the scene, E_1, \dots, E_k are the endmembers, $\alpha_1, \dots, \alpha_k$ are the fractional abundance of each corresponding endmember, and w is a vector representing the system noise.

2.3.2 Non-Linear Mixing Model

One of the assumptions made in the LMM [30] is that each incident light photon only comes in contact with one material, and therefore there is no multiple scattering between materials. Several studies have shown [31–37] that this is not always the case, when

the pixel surface area contains intimate mixtures (i.e. when the materials are mixed on a spatial scale smaller than the path length of a photon [38]), see Fig. 2.6. This occurs for materials like sand and gravel, the light will generally interact with multiple particles before returning to the sensor as multiple scatter light. This causes the received radiance by the sensor to not be a linear combination of the fractional abundances of each material but a non-linear mixture based on the multiple scattering effects of each material. Despite the advantages of using a non-linear mixing model for intimate mixtures, for the purpose of remote sensing this model has not been widely applied. Because of this for the rest of this thesis we will only be considering the LMM for analysis as this is the most widely used approach.

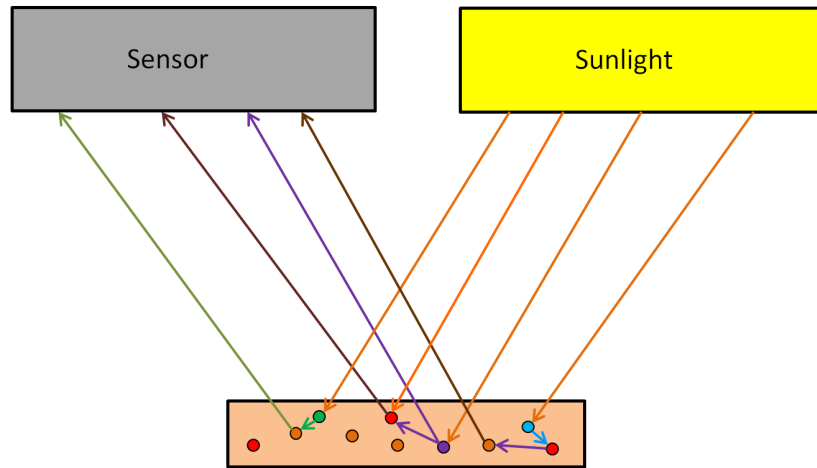


Fig. 2.6: The measured radiance of a mixed pixel is a non-linear mixture based in the multiple scattering effects of each material present in the pixel

2.4 Data Reduction

One of the preprocessing steps in the anomaly detection process is the data reduction stage (see Fig. 2.1). Due to the volume of hyperspectral data for many practical applications in anomaly detection, it is not always possible to store or transmit the full data cube. Therefore various techniques have been developed to reduce the size of the data, some of these techniques involve lossless compression [39–41], these techniques allow for a compression of about 3:1. However for some applications, this is still insuffi-

cient, other techniques look at performing onboard preprocessing of the data before it is stored or transmitted. These algorithms generally quickly determine pixels or bands in the data that contain no useful information, the criteria for determining the usefulness of data obviously change dependant on application. The rest of this section describes data reduction techniques for defence applications where the aim is to locate man-made anomalies.

2.4.1 Band Reduction

Mutual Information (MI) is a measure of the independence between two random variables and was first used by Pluim [42] on greyscale medical images, this was later adapted to work for hyperspectral images [43,44]. The mutual information between two greyscale images is defined by Equation 2.1. This technique works by calculating the Shannon Entropy [45] between bands, to determine a relevant subset of bands for the application.

$$MI(X, Y) = H(X) + H(Y) - H(X, Y) \quad (2.1)$$

where,

X and Y are the two images

$H(X)$ is the Shannon Entropy of the image X

$H(Y)$ is the Shannon Entropy of the image Y

$H(X, Y)$ is the Joint Entropy between images X and Y

Other techniques for band reduction focus on finding bad bands in the data, these bad bands can occur in a number of ways, due to atmospheric attenuation in the atmosphere or simply the spectral sensitivity of the hyperspectral sensor. The bands with a low signal to noise ratio (SNR) result in poor performance from unmixing algorithms due to the inconsistent data. A method developed by Keshava [46] focuses on maximising the angle between spectra of distinct objects by using the spectral angle mapper (SAM) [47]. Another technique by Vibhute [48] looks at analysing bands with low signal-to-noise ratio (SNR) and a canonical based analysis by Tu [49].

2.4.2 Pixel Reduction

Similar to the band reduction techniques, the aim of pixel reduction techniques is to classify pixels in the image that relate to the background, e.g. trees, grass and water.

The most widely used index for remote sensing proposed by Rouse [50] is the normalized difference vegetation index (NDVI) [51–53], this can differentiate between vegetative and non-vegetative surfaces and is particularly useful when looking for man-made anomalies. The NDVI value for each pixel can be calculated by using Equation 2.2.

$$\text{NDVI} = \frac{\text{NIR} - \text{VIS}}{\text{NIR} + \text{VIS}} \quad (2.2)$$

where,

NIR is the value of the spectrum at the near infrared wavelength (approx. 750nm)

VIS is the value of the spectrum at the green visible wavelength (approx. 650nm)

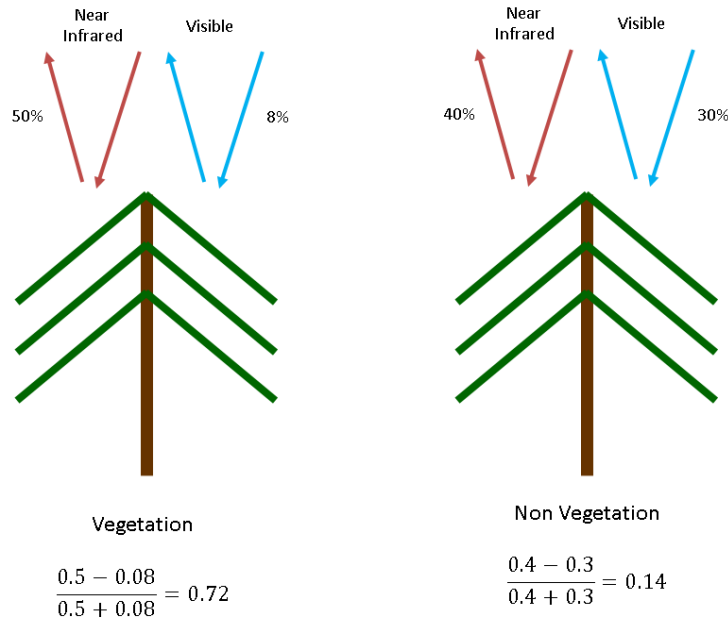


Fig. 2.7: Vegetation vs. non-vegetation NDVI

Because a large portion of visible light incident on vegetation, e.g. plants, trees, etc., is absorbed and a large amount of the near-infrared (NIR) light is reflected; this gives

a large NDVI value. In the case of non-vegetative materials less NIR light is reflected and more visible light is reflected than for vegetation, this, therefore, yields a small NDVI value, see Fig. 2.7. Because of this all pixels with an NDVI value above a certain threshold can be classified as vegetative and removed from the stored or transmitted data.

Another technique for the removal of background pixels was proposed by Bochow [54] that looks at low albedo classification. This works by creating an image containing only the near-infrared (NIR) bands, all bands within the range of 860nm - 900nm, a mean spectral image can be taken of all these bands to produce a greyscale map of the image. Due to the absorption of NIR light by water, low radiance values measured in these bands relate to water pixels.

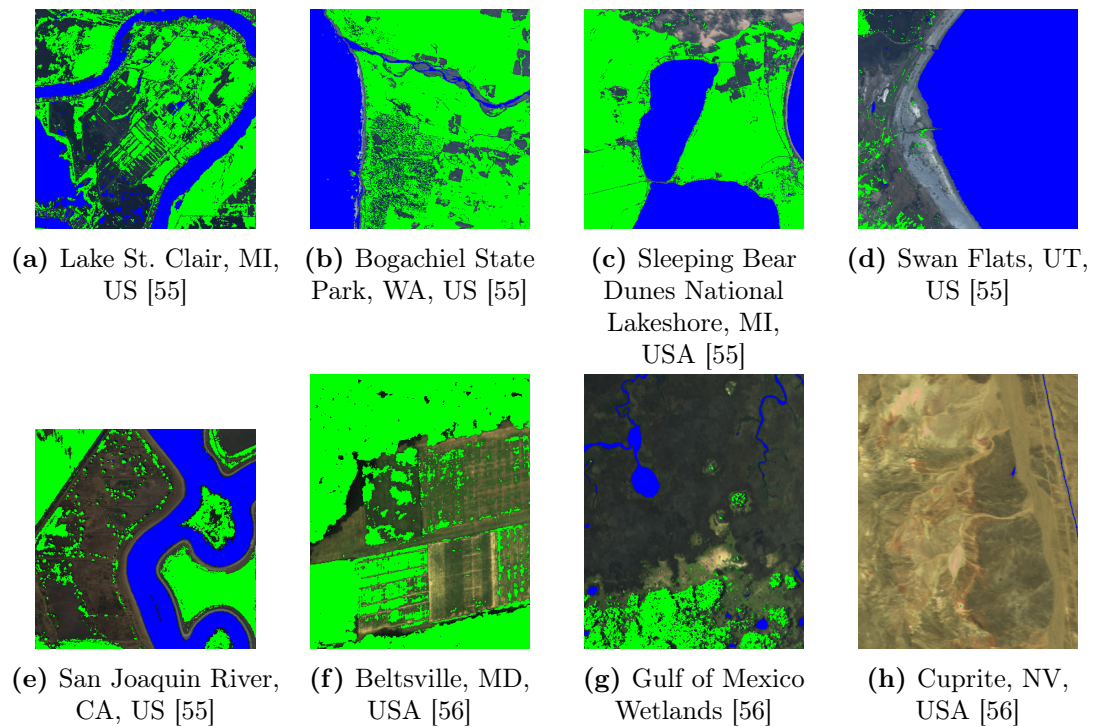


Fig. 2.8: NDVI (green) and low albedo (blue) classification for various hyperspectral images

In a few of these example images shown in in Fig. 2.8:, there is no water present, using the low albedo classification algorithm [54], the shadowed vegetation in these images is classified as water because they share many of the same spectral features. To be able

to differentiate between shadowed vegetation and standing water. This technique was further developed by Bochow [57] to detect the location of shadows in the image and then remove any of these pixels that had been previously been classified as water. The shadows in the image are detected by looking at the NIR region of the image, due to water absorption in this region the radiance from water bodies decreases between 710nm - 740nm and 815nm - 880nm. Therefore once the original mask is created classifying all the pixels with a low albedo, any pixels that match the following two checks are eliminated, see Equations 2.3 and 2.4.

$$L_{obs}(740) - L_{obs}(710) \geq 0 \quad (2.3)$$

$$L_{obs}(880) - L_{obs}(815) \geq 0 \quad (2.4)$$

where $L_{obs}(\lambda)$ is the radiance measured by the camera at the selected wavelength (λ)

2.5 Atmospheric Correction

The second preprocessing step in the anomaly detection process is atmospheric correction (see Fig. 2.1). This can be an important step in aerial hyperspectral imagery because as well as the reflectance characteristics of the ground, the radiance received by an aerial hyperspectral camera depends on the atmospheric attenuation at the specific location, the time of the measurement and the spectrum of the incident solar illumination. Atmospheric attenuation of the received light is caused due to the many aerosols in the atmosphere. Out of the thirty gases that make up the atmosphere, only eight of these have visible absorption effects on hyperspectral images in the Visible to SWIR region of the spectrum. These eight gases are water vapour (H_2O), carbon dioxide (CO_2), ozone (O_3), nitrous oxide (N_2O), carbon monoxide (CO), methane (CH_4), oxygen (O_2) and nitrogen dioxide (NO_2) [58]. Another reason for atmospheric attenuation is caused by aerosols in the atmosphere. These can be naturally occurring aerosols, e.g. dust and volcanic aerosols, as these are made up of several types minerals each of which can either scatter or absorb light [59, 60]. Other aerosols include man-made aerosols, e.g.

smoke and sulphate aerosols which also have effects on the measured radiance by a hyperspectral camera [61, 62].

Because of all these effects, since the mid-1980s, there have been various techniques developed to attempt to remove or reduce the atmospheric effects in hyperspectral data. These techniques can be split up into two main categories, the scene based and radiative transfer based approaches, see Table 2.2.

2.5.1 Scene Based Approaches

The Internal Average Relative Reflectance (IARR) technique developed by Kruse [64] in 1988 was one of the first scene-based approaches developed. This works by calculating an average pixel spectrum for the image, then any pixel vector can be divided by this average pixel spectrum to produce a relative reflectance for that pixel. Another technique developed around this time by Chavez [63] was the Dark Object Subtraction (DOS) method. This method works by assuming the light measured by the sensor to be a combination of the directly reflected and scattered light. As materials like water and tarmac reflect no or very little light, the light measured by the sensor for these pixels must be from the scattered light contribution. By subtracting the spectrum for these pixels from the rest of the image, the atmospheric scattering can be removed. Flat field

Description	Type	Author
Dark Object Subtraction	Scene-Based	Chavez [63]
Internal Average Relative Reflectance	Scene-Based	Kruse [64]
Flat Field	Scene-Based	Roberts [65]
Empirical Line	Scene-Based	Conel [66]
Log Residuals	Scene-Based	Green [67]
Atmospheric Removal Algorithm	Radiative Transfer	Gao [68]
High-accuracy ATmospheric Correction for Hyperspectral Data	Radiative Transfer	Qu [69]
Atmosphere CORrection Now	Radiative Transfer	Kruse [70]
Atmospheric and Topographic Correction	Radiative Transfer	Richter [71]
Fast Line-of-sight Atmospheric Analysis of Spectral Hypercubes	Radiative Transfer	Adler [72]
Moderate Resolution Atmospheric Transmission	Radiative Transfer	Berk [73]

Table 2.2: Types of atmospheric correction algorithms

calibration developed by Roberts [65] works on the assumption that there is at least one spectrally neutral area in the image, i.e. there is very little variation in this area for each band. The mean of this area can then be used to calculate a relative reflectance spectrum for each pixel. The empirical line calibration approach proposed by Conel [66] uses measurements of both a light and dark pixel to derive a gain and offset curve that can be used to estimate the surface reflectance.

2.5.2 Radiative Transfer Approaches

Due to the limitations of the scene-based approaches, towards the end of the 1980s Alexander Goetz first proposed the need for a radiative transfer based approach to atmospheric correction [74]. Soon after this, the first radiative transfer based approach was developed by Gao [68] called the Atmospheric Removal Algorithm (ATREM) by assuming the horizontal surfaces in the image are Lambertian this technique retrieves scaled surface reflectance by simulating the atmospheric transmittance and aerosol scattering of the atmosphere at the time of measurement. Since then there have been a number of radiative transfer based approaches developed, e.g. High-accuracy ATmospheric Correction for Hyperspectral Data (HATCH) [69], Atmosphere CORrection Now (ACORN) [70], Atmospheric and Topographic Correction (ATCOR) [71] and the Fast Line-of-sight Atmospheric Analysis of Spectral Hypercubes (FLAASH) [72].

2.5.3 Statistical Scene Based Approach

In 2004 Bernstein proposed the Quick Atmospheric Correction (QUAC) algorithm due to the fact that none of the methods described above provide high accuracy, are quick to run and do not require any prior known information. This technique requires only an approximate identification of sensor band locations and radiometric calibration. It works by producing a spectral library for both natural and man-made materials in the scene using SMACC, then uses these to calculate a gain and offset based on a reference library to atmospherically correct the data. Because of this QUAC is significantly faster and simpler than radiative transfer based approaches and provides better results than most scene based techniques. However, it has been shown by Smith [75] that radiative

transfer methods still produce far better results.

This research has shown that there is indeed a need for a fast and easy to use radiative transfer based atmospheric correction technique, that still produces comparable results with the current radiative transfer approaches.

2.6 Spectral Unmixing Algorithms

An example of a complete spectral unmixing technique is the Sequential Maximum Angle Convex Cone (SMACC) technique which was first proposed by Gruninger [76]. This method works by first calculating the length of each pixel vector in the image. The pixel vector with the maximum length is then selected as the first endmember and using this a subspace projection is performed with every other pixel vector in the image, to calculate the abundance of that endmember (pixel vector) in each pixel. The final step is to remove all reference to this endmember from the image; this is done by using Equation 2.5. Using this equation a new updated image is produced which the process can be repeated on until all endmembers are found, another method that uses the same approach is Convex Cone Analysis (CCA) [77].

$$h_1 = h_0 - EM \times AI \quad (2.5)$$

where,

h_0 is the original data set

h_1 is the updated data set

EM is the spectral profile of the endmember selected, and

AI is the corresponding abundance image for the endmember selected

Other types of unmixing algorithms use a projection onto a simplex rather than onto a convex cone, the Pixel Purity Index (PPI) proposed by Chaudry [78] uses this approach. This works by first assuming that there is one pure pixel in the dataset for every material, an assumption that for all datasets may not hold true. A large set of random unit vectors "skewers" are generated and every pixel is then projected onto the

first skewer and the two pixels that relate to the maximal and minimal projection are both given one point. This process then continues for the remaining skewers and at the end the pixels with the highest number of points are considered to be the endmembers. An example of how this process works for a 2D dataset with three skewers is shown below, see Fig. 2.9. Other examples of these types of algorithms include Vertex Component Analysis (VCA) [79], Simplex Growing Algorithm (SGA) [80] and Maximum Distance (MaxD) [81].

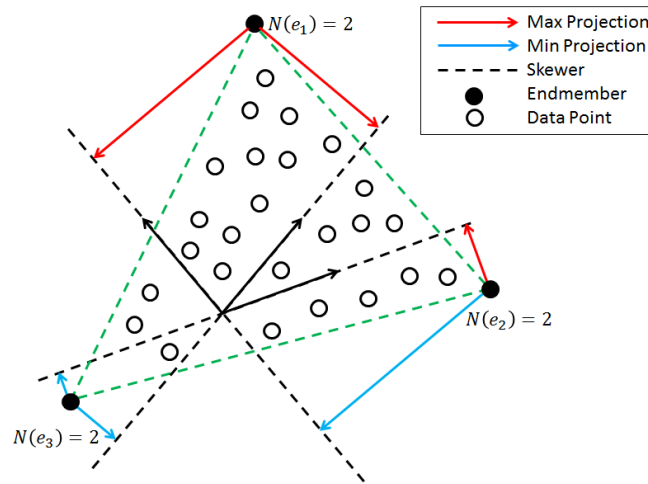


Fig. 2.9: Pixel purity index 2D example

2.7 Anomaly Detection Algorithms

Unlike complete spectral unmixing techniques described previously, anomaly detection techniques work by determining pixels in the image that are spectrally distinct from the background pixels. In these types of algorithms, there is no estimation of the underlying materials making up the mixed pixel. This approach works under the assumption that all targets are at least one pixel in size and therefore have at least one pure pixel in the image, this constraint can be slightly relaxed to encompass targets that will be the dominant material in a mixed pixel.

The Reed-Xiaoli Detector (RXD) detector proposed by Reed and Yu [82] is an example of one of these techniques, this can be considered as the inverse operation to

Principal Component Analysis (PCA) [83]. PCA works by compressing all the major features of an image into a few principal components based on the eigenvectors of the covariance matrix, therefore targets that are small will have low eigenvectors and will be ignored by PCA. The RX detector works by producing high values for spectra with low eigenvectors, the main problem with this technique is attempting to separate small eigenvalues from noise in the data. Another example of these type of techniques involves using the Mahalanobis Distance (MD) first proposed by Mahalanobis [84] and adapted for HSI by Mark [85]. To calculate the MD Distance ($MD(x)$) in Equation 2.6 for an image pixel vector (x) requires first calculating a mean pixel vector (μ) containing the average pixel intensity values for all the bands or wavelengths measured, as well as an estimated covariance matrix (Γ) of the pixel intensity values for all of the bands using Equation 2.7, where N is the number of pixels in the image. Doing this for every pixel vector gives a measure of how far that pixel is from the mean pixel vector where larger values indicate more extreme pixel vectors.

$$MD(x, \mu) = \sqrt{(x - \hat{\mu})^T \Gamma^{-1} (x - \hat{\mu})} \quad (2.6)$$

$$\hat{\Gamma} = \frac{1}{N} \sum_{n=1}^N (x - \hat{\mu})(x - \hat{\mu})^T \quad (2.7)$$

where

x is the pixel vector,

μ is the mean pixel vector,

$(\cdot)^T$ is the vector transposed,

Γ is the estimated covariance of the background, and

N is the number of pixels

2.8 Practical Applications

2.8.1 Defence and Security

Many of the initial advances in hyperspectral imaging are owed to military funded projects, because of this there are a number of practical applications for the technology.

In [86] a target detection method is proposed for the classification of airborne aircraft using AVIRIS hyperspectral data in the wavelength range of 370 to 2510nm. A band reduction method is first applied to remove water absorption, low SNR and bad bands, then a low-rank regularised least squares method based matched subspace detector is proposed. The results are shown in Fig. 2.10.

The automatic target detection and classification algorithm and desired target detection and classification algorithm proposed in [87] and later improved for real-time processing in [88]. For the case of known targets, an orthogonal subspace projection is performed to find any reference to the target spectrum in the pixels of the image. For the case of unknown targets it attempts to solve the linear mixing model to produce pure pixels then uses the undesired target signature annihilator developed in [89] to remove any undesired pixel signatures.

In [90] a technique to remove the effects of the atmosphere, ground conditions, scene temperature, solar illumination, and sensor response on hyperspectral imaging for military vehicle detection is proposed. This works by trying to transform the distorted spectra based on the covariance of neighbouring pixels.

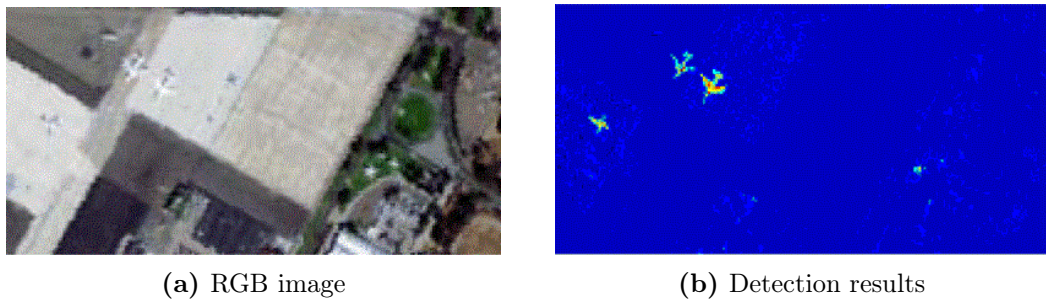


Fig. 2.10: Detection of aircraft over San Diego, CA, USA [86]

2.8.2 Precision Agriculture

The technique proposed in [91] looks at improving the existing crop classification techniques. Using hyperspectral images in the range of 400 to 850nm various types of rice in the Jintan farm, China are classified using a combination of PCA and vegetation feature band sets, the results from this are shown in Fig. 2.11.

In [92] the monitoring of leaf nitrogen concentrations in rice from a UAV-based hyperspectral camera is proposed. Using various vegetation indices it is shown that by using a hyperspectral camera it is possible to derive leaf nitrogen concentration.

Neural networks, support vector machine and k-nearest neighbour (kNN) classifiers were compared in [93] for hyperspectral imagery captured from a UAV. It was found that using a kNN it is possible to classify red clover ground coverage with an accuracy of 91%.

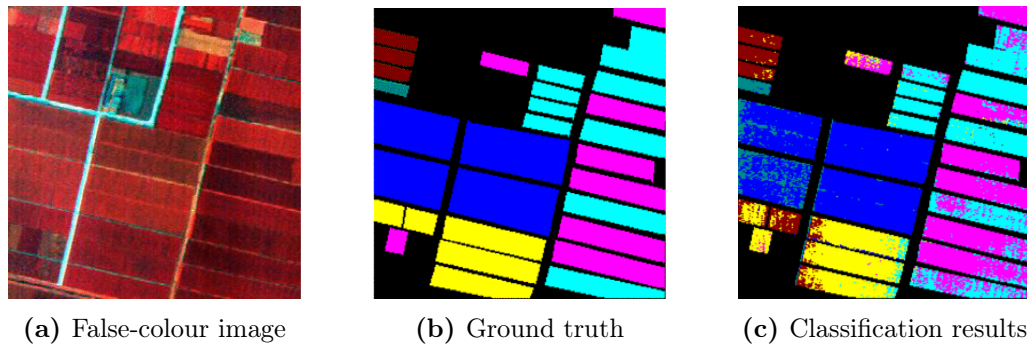


Fig. 2.11: Classification of various Japanese crop types [91]

2.8.3 Environmental Monitoring

In [6] using the fusion of NDVI and the Haralick texture features [94] obtained from the greyscale image for NIR aerial hyperspectral imagery it is possible to accurately classify the vegetation on the ground, the results from this are shown in Fig. 2.12.

Boreal and Arctic region lakes have a very high concentration of coloured dissolved organic matter (CDOM) making signals received from hyperspectral sensors over these lakes almost negligible in the VIS region of the spectrum (400 - 700 nm) due to the absorption by CDOM, except for peaks at 710 and 810nm. In [95] it is shown there is a good correlation between the 810nm peak and phytoplankton organic biomass but no

correlation between it and phytoplankton inorganic biomass.

In [96] the spectra of zoisite, mica, quartz, alunite, cummingtonite, sodalite, dolomite, anorthite, and actinolite are extracted from VNIR hyperspectral data using the PPI [78] and the abundance of each is then found in the image pixels using a combination of Binary Encoding [97], the Spectral Angle Mapper [98], and Spectral Feature Fitting [99–102]

2.9 Summary

In the first section of this chapter, it was explained what hyperspectral imaging was, how it compared to conventional imaging systems, the various regions of the EM spectrum that current systems can also measure and the subtleties between multispectral, hyperspectral and ultraspectral were also all explained. The next section explained the issues caused by insufficient spatial resolution on the ground causing 'mixed pixels' and how this effect can be modelled. Following, the next three sections each covered an individual stage in the target detection process these were Data Reduction, Atmospheric Correction and Target/Anomaly Detection. A summary of some practical applications for aerial hyperspectral imagery was presented in the final section.

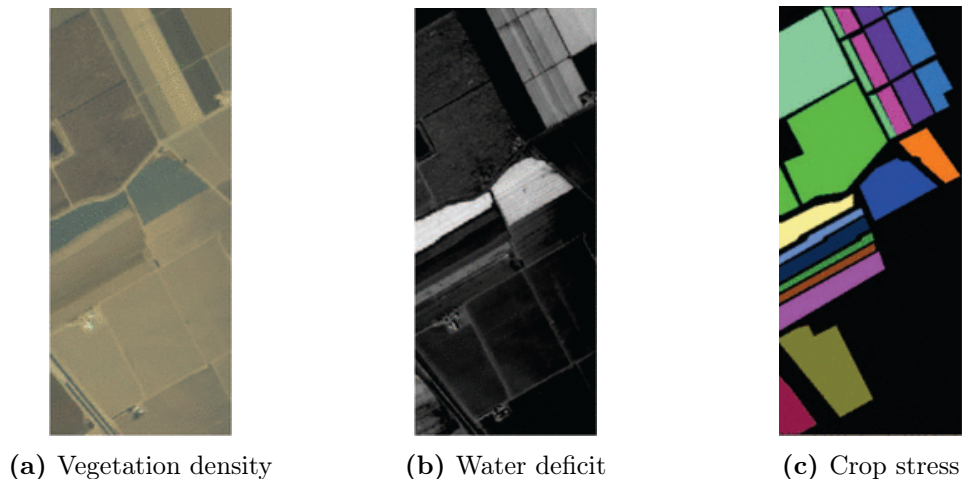


Fig. 2.12: Vegetation classification using NDVI [6]

Chapter 3

Anomaly Detection in Aerial Hyperspectral Imagery

3.1 Introduction

Anomaly detection for the purpose of hyperspectral imaging is the process of applying a binary classification to the image, pixels are either labelled as an anomaly (a pixel that is spectrally different from the background) or as a background pixel. Unlike traditional target detection algorithms, e.g. Spectral Angle Mapper (SAM) [47], Modified Spectral Angle Mapper (MSAM) [103], Spectral Correlation Angle (SCA) [104], Spectral Gradient Angle (SGA) [104], Extended Spectral Angle Mapper (ESAM) [105] and Adaptive Coherence Estimator (ACE) [106], where prior knowledge of the target spectral characteristics is known. Anomaly detection is performed by determining the pixels in the image that are spectrally different from the surrounding background pixels [107, 108]. Each of these anomalies can then be defined into separate classes for further study.

In the linear mixing model [30, 89] the radiance measured at each pixel is assumed to be the linear combination of the radiance of each material present in the pixel. The non-linear mixing model [109–111] the radiance measured is assumed to be a weighted sum of the radiance of each material within the pixel plus the contribution due to scattering from both the sun to ground path and ground to sensor path. Pure pixels are considered to be pixels where there is only one material present in the entire pixel.

In this chapter, the aim is to develop a new anomaly detection technique that is an improvement over what is currently available. Several anomaly detection techniques are investigated and used on the raw image data to find anomalous pixels in the image. The results are then compared with a validated technique for endmember selection, Sequential Maximum Angle Convex Cone (SMACC) [76].

The rest of this chapter is structured as follows. Section 3.2 introduces the hyperspectral data used in this chapter, how, when and where it was taken and how it was supplied. An introduction to Anomaly Detection is given in Section 3.3. Section 3.4 discusses the uses of the Mahalanobis Distance for the purpose of anomaly detection in hyperspectral imagery and attempts to improve upon the initial results by altering some of the key parameters. The Hough Transform is discussed in Section 3.5 and how by drawing a straight line between the origin and every point in a scatter plot the outliers might be found. Section 3.6 proposes using the Hit or Miss transform to detect the outliers in the scatter plot, while Section 3.7 develops on the results from the previous section to produce a viable technique. Section 3.8 is devoted to the implementation of all the proposed techniques and the results produced for each image. A comparison is also made between an established technique in terms of accuracy and processing speed. A summary of the contributions is then presented in Section 3.9.

3.2 Image Acquisition

All the data used in this chapter was supplied by BAE Systems Military Air & Information, the Defence Science and Technology Laboratory (DSTL) and the University Defence Research Collaboration (UDRC). The imagery is all remotely sensed aerial hyperspectral images acquired from an aeroplane flying at an altitude of approximately 1 km with a mounted hyperspectral sensor. Both images were captured in Salisbury, Wiltshire in 2014 and are taken in the visible to near-infrared (400 nm to 1000 nm) region of the spectrum. Fig. 3.1 shows two false colour representations of these images, also shown in Table 3.1 are additional details about each image.



(a) Moll Harris, Salisbury, Wiltshire



(b) Porton Down, Salisbury, Wiltshire

Fig. 3.1: False-colour representations of hyperspectral images (636nm, 555nm 460nm respectively for RGB)

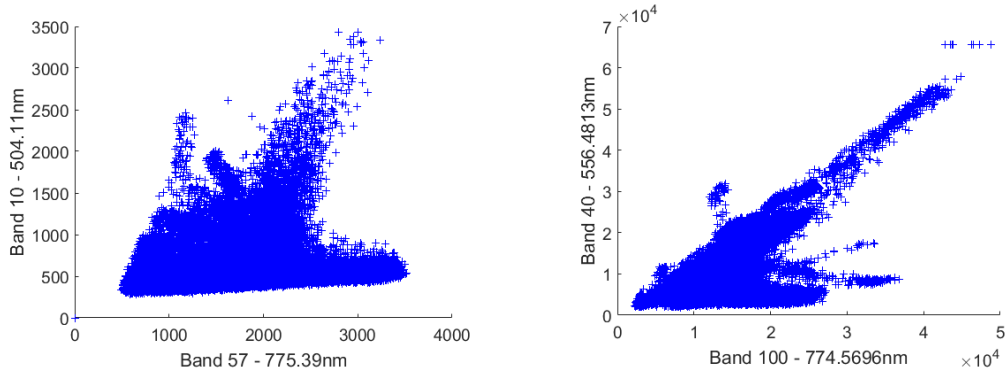
Parameter	Moll Harris	Porton Down
Date	18th May 2014	15th August 2014
Time	11:29	15:17
Latitude	51 °N	51 °N
Longitude	−2 °E	−2 °E
Altitude	0.781 km	1.197 km

Table 3.1: Dataset details

3.3 Anomaly Detection

Anomaly detection is used in hyperspectral imagery to find materials and objects that are spectrally different from the image background; Fig. 3.2 shows two example scatter plots created using two bands of each of the datasets. As both images are mostly trees and grass which are spectrally similar, there is a large cluster of pixels with similar radiance values in each band. The targets or anomalies in the scene are the pixels furthest from these clusters. Pixels closer to the centre are more mixed and ones further away are purer. By determining these outliers it is possible to detect anomalous objects or materials in the image that require further attention. Numerous techniques are used to find anomalies in hyperspectral data [2, 112–114].

Here we try to improve upon the standard MD technique [84] for hyperspectral data and propose a new method based on the Percentage Occupancy Hit or Miss Transform (POHMT) [14], to try to improve upon a selected method i.e SMACC.



(a) Moll Harris data using bands 10 and 57 (b) Porton Down data using bands 40 and 100

Fig. 3.2: Scatter plot for both hyperspectral images

3.4 Mahalanobis Distance

To calculate the MD Distance ($MD(x, \mu)$) in Equation 3.1 for an image pixel vector (x) requires first calculating a mean pixel vector (μ) containing the average pixel intensity values for all the bands or wavelengths measured, as well as an estimated covariance matrix (Γ) of the pixel intensity values for all of the bands using Equation 3.2, where N is the number of pixels in the image. Doing this for every pixel vector gives a measure of how far that pixel is from the mean pixel vector where larger values indicate more extreme pixel vectors.

$$MD(x, \mu) = \sqrt{(x - \hat{\mu})^T \Gamma^{-1} (x - \hat{\mu})} \quad (3.1)$$

$$\hat{\Gamma} = \frac{1}{N} \sum_{n=1}^N (x - \hat{\mu})(x - \hat{\mu})^T \quad (3.2)$$

where,

x is the pixel vector

μ is the mean pixel vector

$(\cdot)^T$ is the vector transposed

Γ is the estimated covariance of the background

N is the number of pixels

3.4.1 2D vs 3D vs nD Cases

2D Dataset

Before trying to detect outliers in multivariate data it was first necessary to detect them in two dimensional data. Using Equation 3.1 the squared MD was calculated for each point for two randomly selected bands. To determine the outliers the standard deviation (SD) of the MD was calculated. All points that had a MD larger than one SD were considered to be anomalies. The results for six random scatter plots are shown below, see Fig. 3.3.

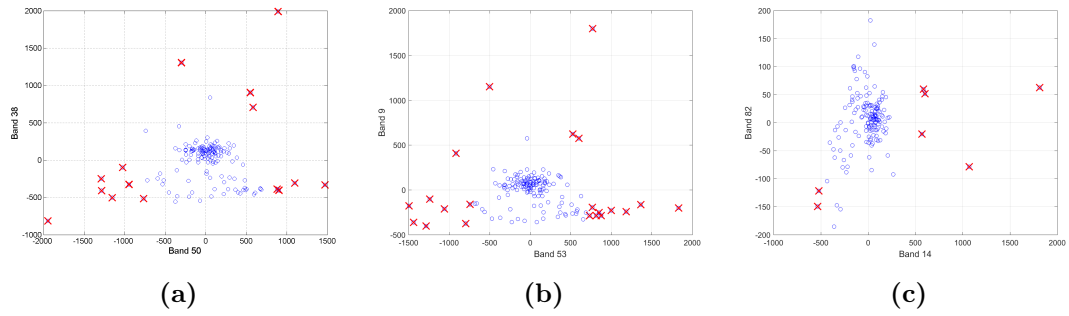


Fig. 3.3: Random scatter plots with outliers highlighted

3D Dataset

The selected data was then expanded to include a third randomly selected band, to simulate a 3D dataset. Using the same method as used in the previous section, the outliers were found and the results are shown below, see Fig. 3.4.

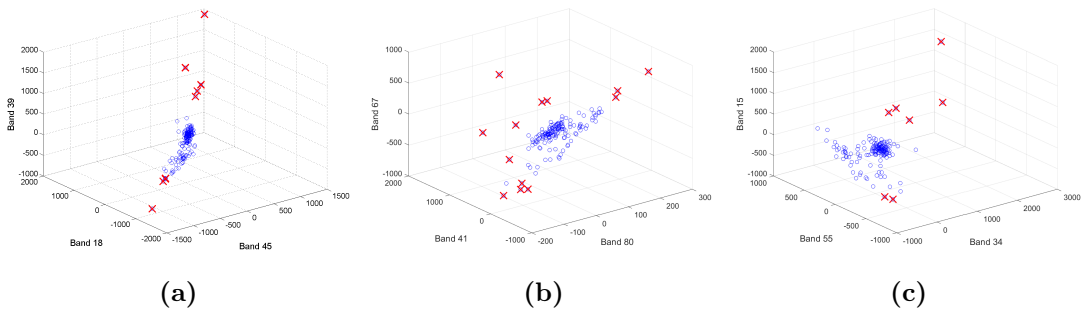


Fig. 3.4: Random 3D scatter plots with outliers highlighted

nD Dataset

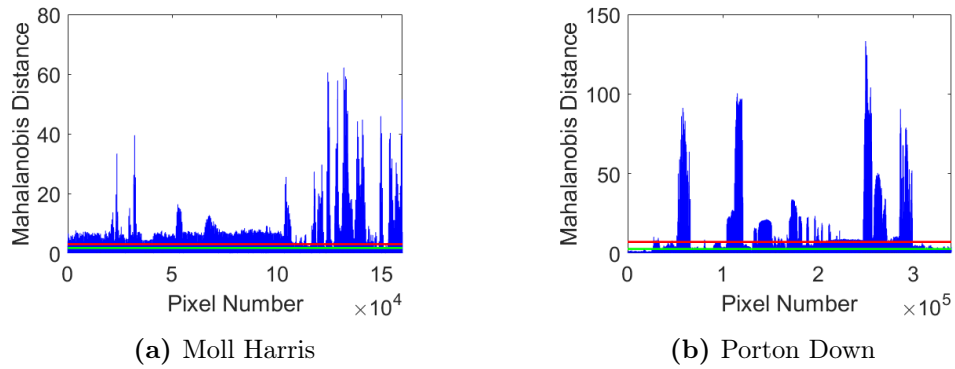


Fig. 3.5: Plot of Mahalanobis distance for all pixels, includes line indicating standard deviation (Green) and variance (Red)

Finally, the selected data was expanded to include all bands between 450nm and 940nm, as before the same method was used to calculate the squared Mahalanobis distance for each pixel. The results are shown in Fig. 3.5, also shown are two lines that indicated the standard deviation and variance of the distances. Unlike before as these lines fall within the mean of the data, it is not possible to use these as a threshold to find the outliers of the data. Using the built-in MATLAB function, `findpeaks` [115], that finds local maxima of the data by finding points that are larger than both of its neighbours, the peaks of the distances were found. The amount of peaks was then reduced by finding the points that were spatially far apart, a limit was set to a minimum distance between

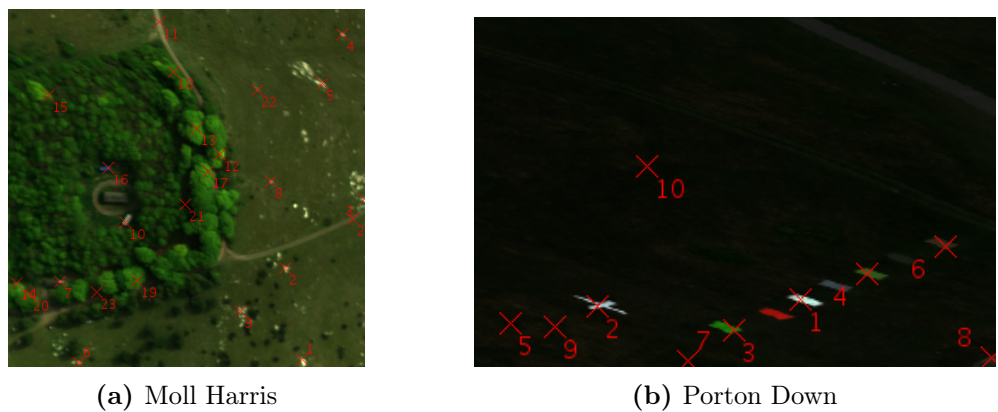


Fig. 3.6: Outliers produced using peaks from Mahalanobis distance plot

peaks of 5000 pixels. This produced twenty-four peaks for the Moll Harris image and ten for the Porton Down image, the locations of all of these points were plotted and are shown below, see Fig. 3.6.

3.4.2 Find Centre of Data

A main component of the MD technique requires the calculation of a covariance matrix [116], a key factor in this calculation is the estimation of the centre of the data, or the mean pixel spectrum (μ). Below the standard method along with three new methods for calculating the centre of the data are described.

Mean Pixel Spectrum

The standard technique is to simply calculate the mean value in each column (band) this produces a mean pixel spectrum.

Median Pixel Spectrum

Similar to the standard method the median approach calculates the median of each column (band) to produce a median pixel spectrum.

Interquartile Mean Pixel Spectrum

Using Equation 3.3, the interquartile mean of each column (band) is calculated to produced an interquartile mean pixel spectrum.

$$x_{IQM} = \frac{2}{N} \sum_{i=\frac{N}{4}+1}^{\frac{3N}{4}} x_i \quad (3.3)$$

One-Step Tukey's Biweight

Another method proposed was to calculate a value for Tukey's biweight [117] for each band in the hyperspectral image. This produced a spectral profile that should represent the centre of each band.

1. $m = \text{median}(x)$
2. $s = \text{median}(\text{abs}(x(i) - m))$
3. $u = \frac{x-m}{(c \times s)}$
4. $i = \text{abs}(u) \leq 1$
5. $w = ((1 - u^2)^2) \times i$
6. $\text{Tukey's Biweight} = \frac{\sum(w \times x)}{\sum w}$

where,

x is the pixel vector

c is the tuning constant (typically $c = 5$)

3.4.3 Centre of Data Results

Having previously used the mean pixel spectrum to calculate the MD for each pixel in both hyperspectral images (see Equation 3.1). By using the median, interquartile mean and one-step Tukey's biweight approaches, the MD was again calculated for both images (see Fig. 3.7).

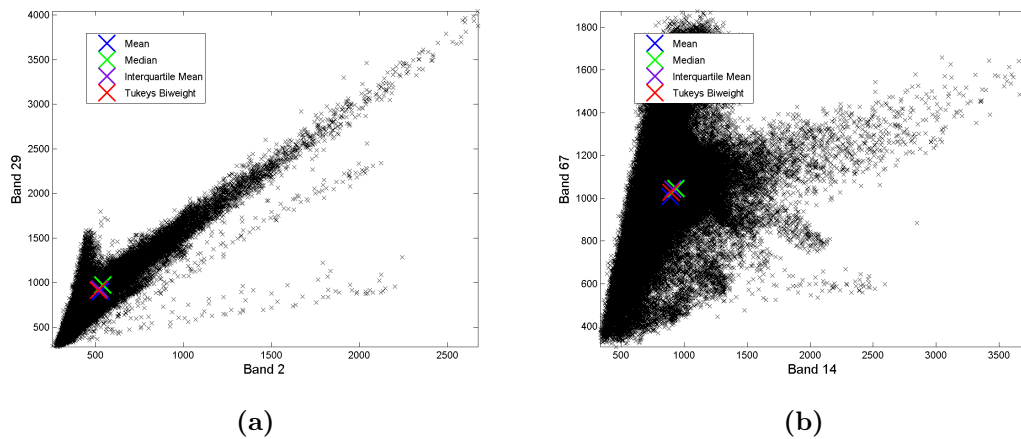


Fig. 3.7: Random scatter plots with mean (blue), median (green), interquartile mean (purple) and one-step Tukey's bi-weight (red)

The results show that there is very little difference between the four methods, therefore it can be concluded that due to the massive number of points in the data, regardless of the method for calculating the centre, all methods proposed produce approximately the same point.

3.5 Modified Hough Transform Approach

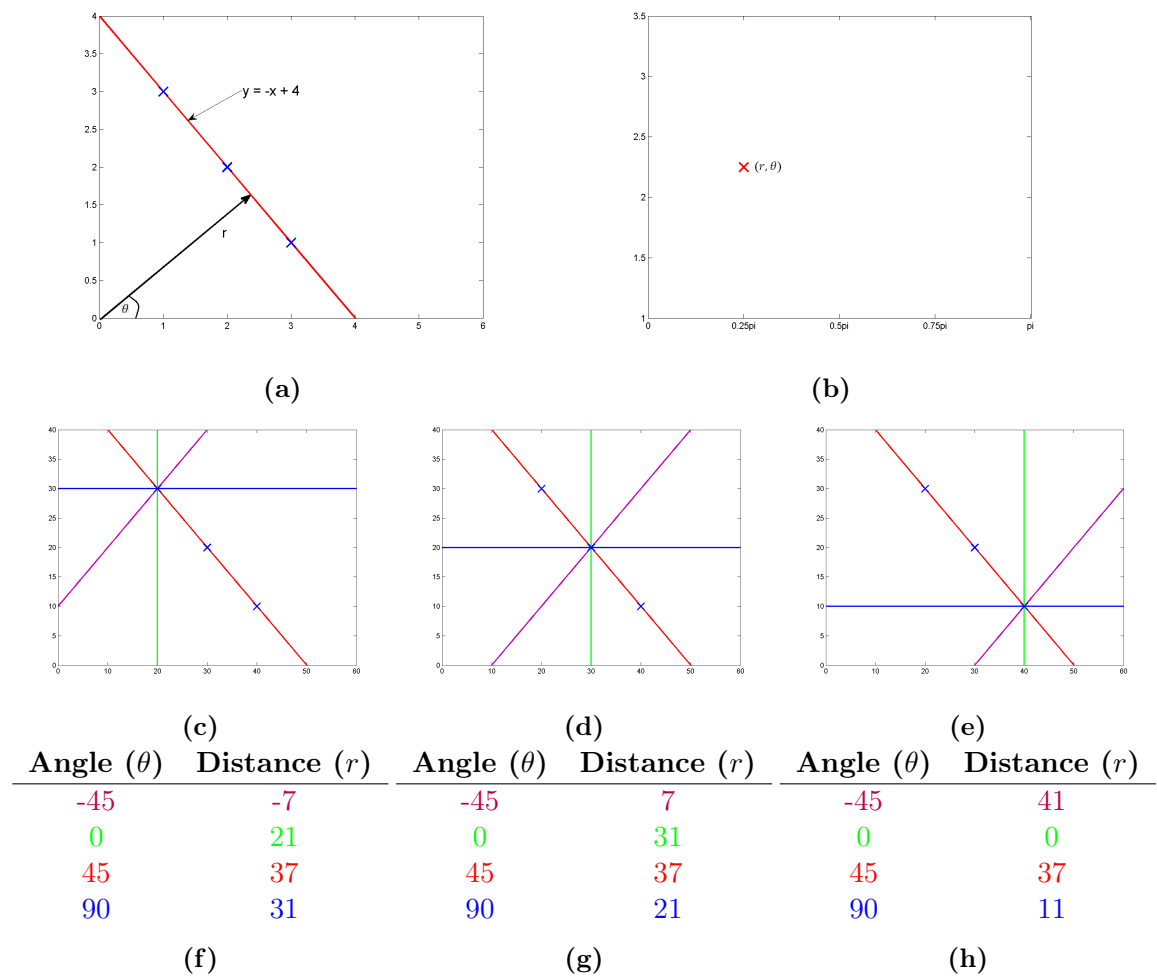


Fig. 3.8: Cartesian to polar coordinates

The points (20,30), (30,20) and (40,10) can all be connected by the line $y = -x + 4$, see Fig. 3.8. To find this line using the Hough Transform [118] every point needs to be converted into the polar coordinate system. This is done by first defining a finite set

of angles between $-\frac{1}{2}\pi$ and $\frac{1}{2}\pi$, an accumulator matrix is created, where each column corresponds to the angles of θ and the rows correspond to the distances calculated for r . For every point in the image, each distance for r for every angle θ is calculated, see Equation 3.4, as each distance is produced the corresponding element in the blank matrix is increased by one. Once this is complete the highest values in the accumulator matrix correspond to the lines in the image. In the example, see Fig. 3.8, all three points lie on the same line therefore when converted into the polar domain these three points have the same distance from the origin (r) when θ is equal to 45° .

$$r = x \cos(\theta) + y \sin(\theta) \quad (3.4)$$

where,

x and y are the Cartesian coordinates of the point

θ is a set of angles

Based on this idea, another method that was looked at for detecting outliers in a scatter plot was to draw a line between each point and the origin of the plot, then measure the angle between this line and the x-axis, see Fig. 3.9. This would create a set of angles for every point in the image, a histogram could then be created for all these points. Again using the built-in MATLAB function, `findpeaks` [115] the local maxima for this histogram can be found. Any peaks that have zero or a very small number of angles neighbouring can be considered to be anomalies.

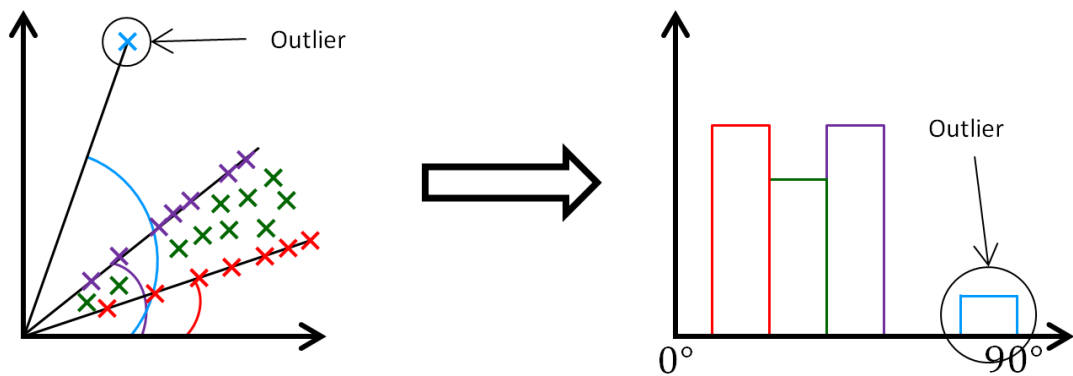


Fig. 3.9: Scatter plot angles diagram

The process detailed above was then performed on three random sets of bands from the Moll Harris image. The resulting span of angles produced is shown below, see Fig. 3.10.

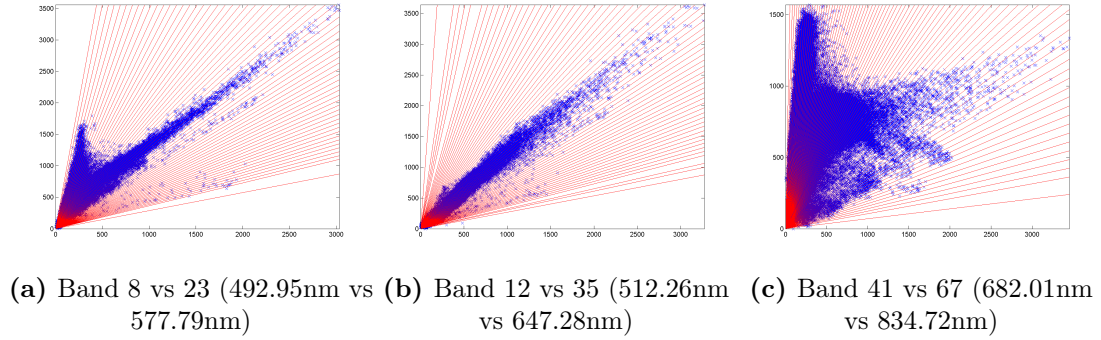


Fig. 3.10: Span of angles for Moll Harris 18-5 1129 dataset

Having produced angles for every point in the scatter plot, a histogram was created to combine all these angles, so that any outliers could be found using the method discussed previously, see Fig. 3.11.

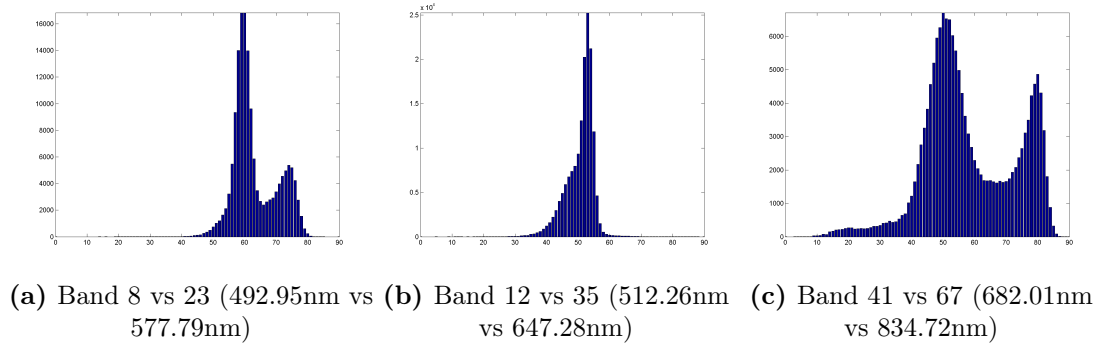


Fig. 3.11: Histogram for each set of angles

Because there is such a large number of points at the centre of the scatter plot any small peak relating to outliers cannot be seen due to the size of these central angles. To make these outliers more prominent a log scale was used for the number of points relating to each angle, see Fig. 3.12. From this, it is now possible to see several peaks that were not visible before.

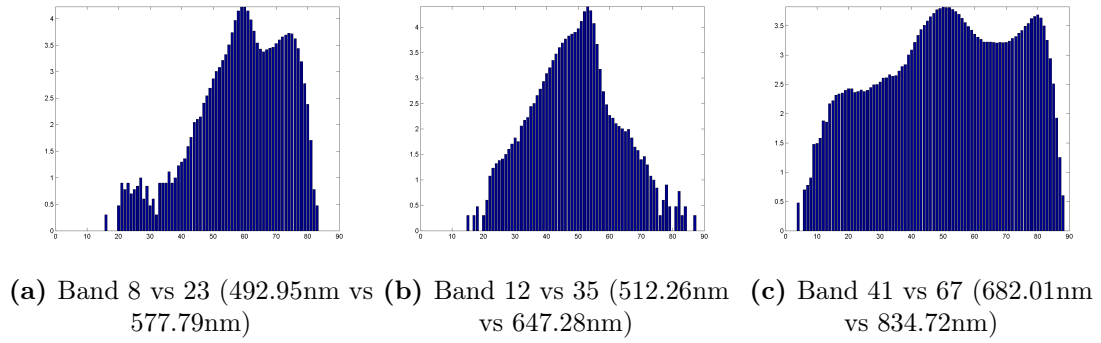


Fig. 3.12: Histogram for each set of angles using log scale

Using the built-in MATLAB function, `findpeaks` [115] it was attempted to extract these local maxima from the histogram. These local maxima were then related to the angles that produced them and then plotted on the original scatter plot, see Fig. 3.13. The resulting plots show that the majority of peaks found in the histogram did not relate to outliers, therefore another approach would be necessary to find the outliers in these scatter plots.

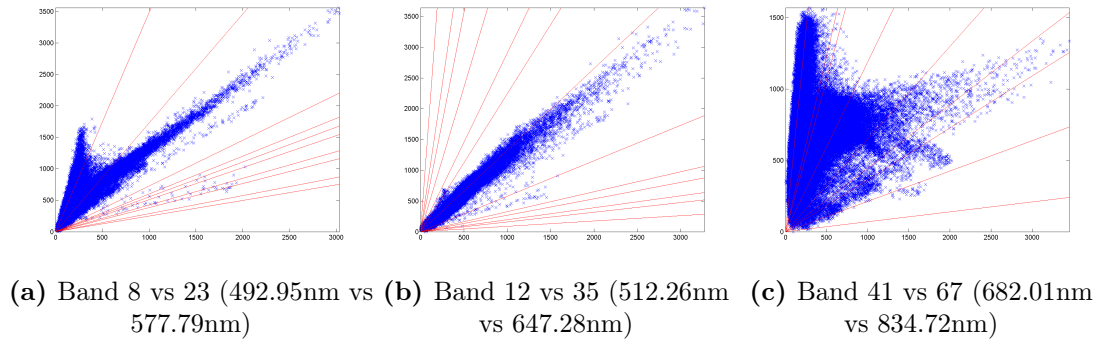


Fig. 3.13: Local maxima for each histogram

3.6 Hit or Miss Transform Approach

Mathematical morphology was first introduced by Matheron in 1974 [119] and Serra in 1982 [120], this was later expanded by Heijmans in 1994 [121]. One of these image processing tools is the hit-or-miss transform [119,120]. The Hit or Miss transform works by taking an erosion and dilation of an image by two structuring elements (SEs). A SE

in mathematical morphology is defined as a subset of the image that is used to analyse the image under study. These SEs can have any shape and size, but are usually defined by the features in the image that are meant to be highlighted or removed [122]. An erosion works by taking the minimum of all the values in the SE and a dilation works by taking the maximum, see Fig. 3.14a. This means that points, where the dilation is less than the erosion, will only occur when the background SE fits around the object and the foreground SE fits inside it, see Fig. 3.14b.

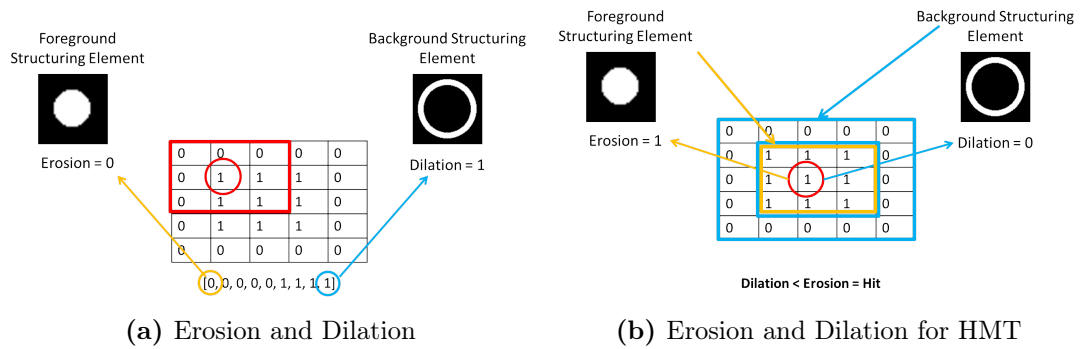


Fig. 3.14: Erosion, Dilation and HMT Diagrams

Based on this, the next method looked at for the process of anomaly detection, was to use the hit or miss transform (HMT) to detect any point in the scatter plot that was a target or anomaly. This was done by first dividing the scatter plot into several small cells, the cell size could vary between 5x5 to 150x150 pixels wide. Fig. 3.15 shows an example for both images.

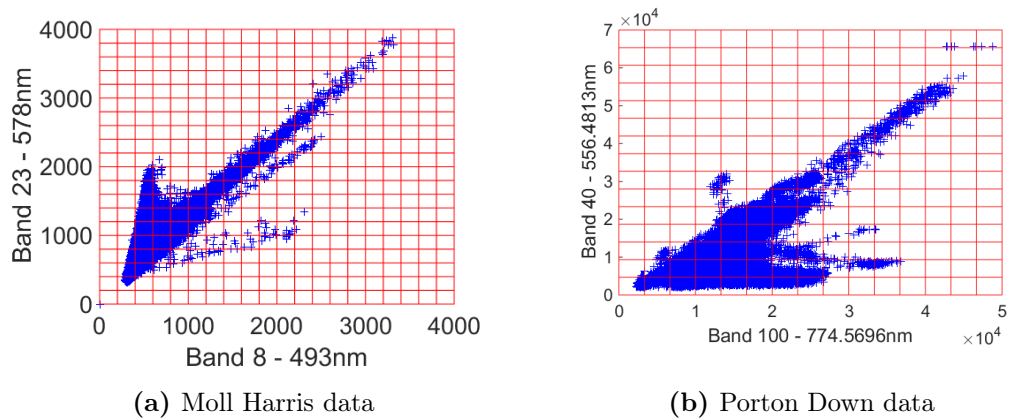


Fig. 3.15: Scatter plot divided into cells

This process was performed on the Moll Harris scatter plot created by using bands 8 vs 23 (493nm vs 578nm), a cell size of 50x50 was used for this initial test. The new image that was created was therefore 1/50 th of the size of the original scatter plot, the values for this were found by looking at the individual cells in the scatter plot. Every cell that had at least one white pixel contained within it, the corresponding pixel in the new image was set to a value of 1, see Fig. 3.16.

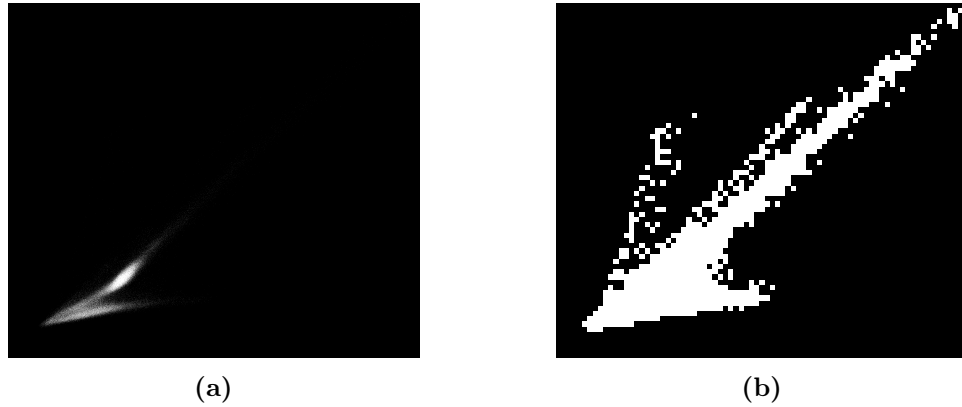


Fig. 3.16: Moll Harris scatter plot for bands 8 vs 23 (493nm vs 578nm)

As the outliers in each scatter plot are not all found in the same direction, four sets of structuring elements were required to detect each different case, see Fig. 3.17.

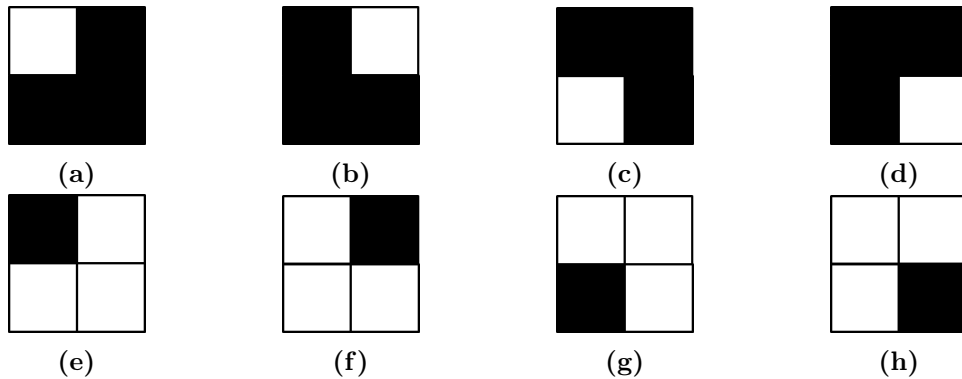


Fig. 3.17: (a-d) Foreground, (e-h) Background structuring elements

Using each of these sets of structuring elements the hit or miss transform was run on the newly created images. This produced four output images, see Fig. 3.18, each of these images shows points in the scatter plot that have been defined as outliers using each

of these SEs. The four images were then combined into one image so that all outliers could be shown in one image, see Fig. 3.19.

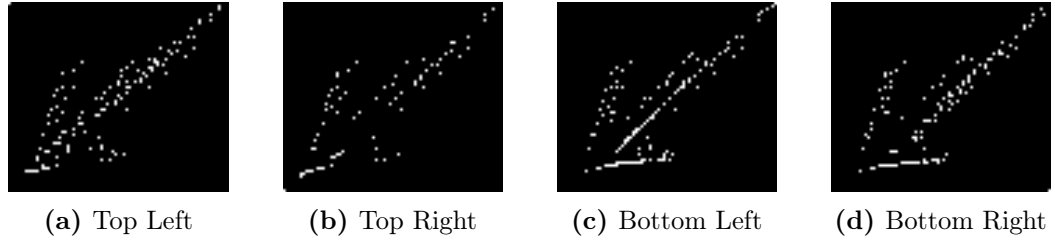


Fig. 3.18: Hit or Miss Transform output for each set of structuring elements

The results from this show that while the HMT is an improvement over the modified Hough transform technique, however there are still too many false positives found in the centre of the data. This is thought to be due to the binary nature of the HMT, therefore a more robust method of finding these outliers is necessary.

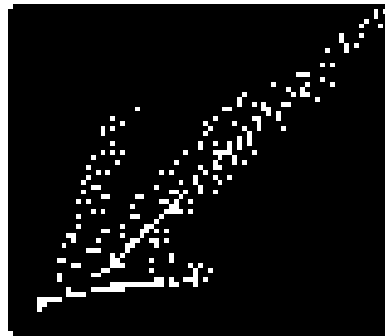


Fig. 3.19: Combined Hit or Miss Transform output

3.7 Percentage Occupancy Hit or Miss Transform

There have been many implementations of the standard Hit or Miss Transform [123–131], however very few work in the presence of noise. One implementation that does work in the presence of noise first introduced by Murray in 2009 [14, 132] is the Percentage Occupancy Hit or Miss Transform (POHMT). Therefore this is a more robust implementation of the HMT. Using the process outlined previously for the standard HMT, this can be easily modified to take advantage of the POHMT robustness to noise to detect outliers in the scatter plot.

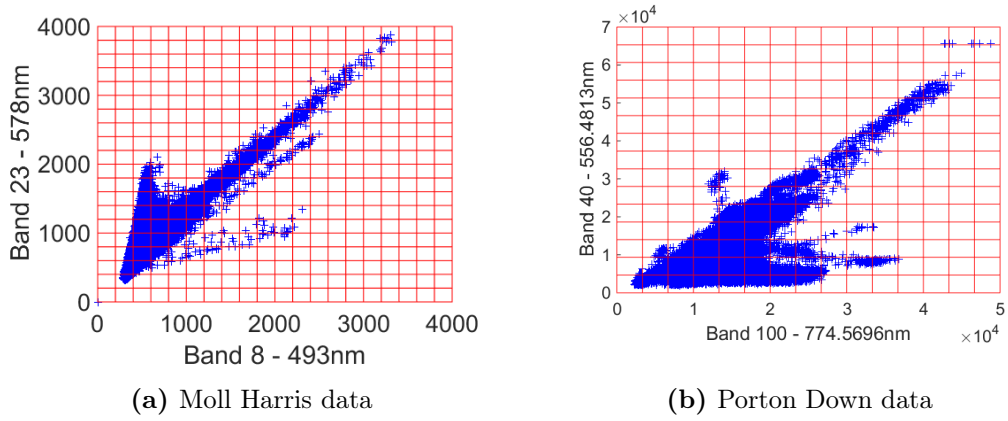


Fig. 3.20: Scatter plot divided into cells

The first step was to divide the scatter plot into several small cells, depending on the data and scatter plot chosen (Fig. 3.20). As before the cell sizes vary from 5x5 to 150x150 pixels wide, also as before the scatter plot is split into cells however unlike for the previous method the number of pixels in each cell was counted and this value was taken as the corresponding pixel value in the new image.

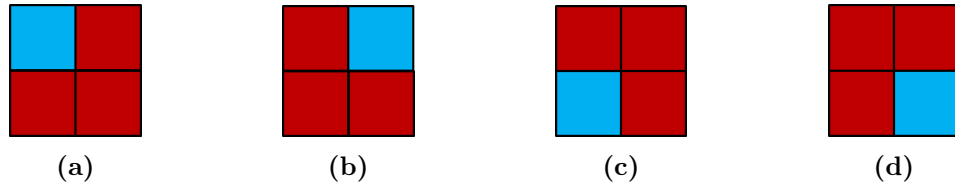


Fig. 3.21: Four states used to detect outliers; blue is foreground, red is background

Using the same four SEs as selected before (see Fig. 3.21). To find the outliers in the scatter plot, the ratio between the number of pixels in the foreground (C_{FG} ; blue zone) and background (C_{BG} ; red zone) was examined (Fig. 3.21). This ratio is calculated for every pixel using Equation 3.5, where n is now the number of pixels in the cell.

$$R = \frac{\sum C_{FG}}{\sum C_{BG}} \quad (3.5)$$

Using the Moll Harris scatter plot for Bands 8 vs 23 (493nm vs 578nm), the ratio for every point in the scatter plot was calculated for each of the four states and is plotted below (see Fig. 3.22). Using the built-in MATLAB function, `findpeaks` [115],

the local maxima for each of these plots were found and those peaks were considered to be outliers (see Fig. 3.23).

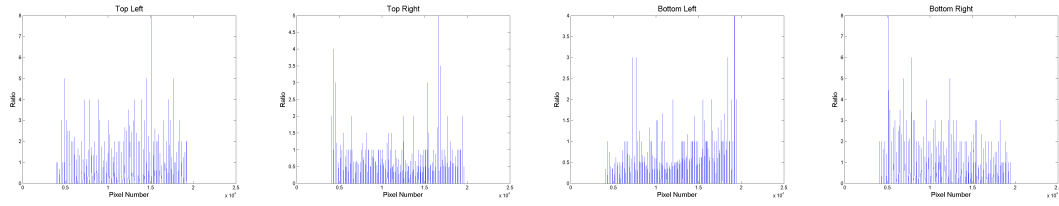


Fig. 3.22: Ratios for each condition

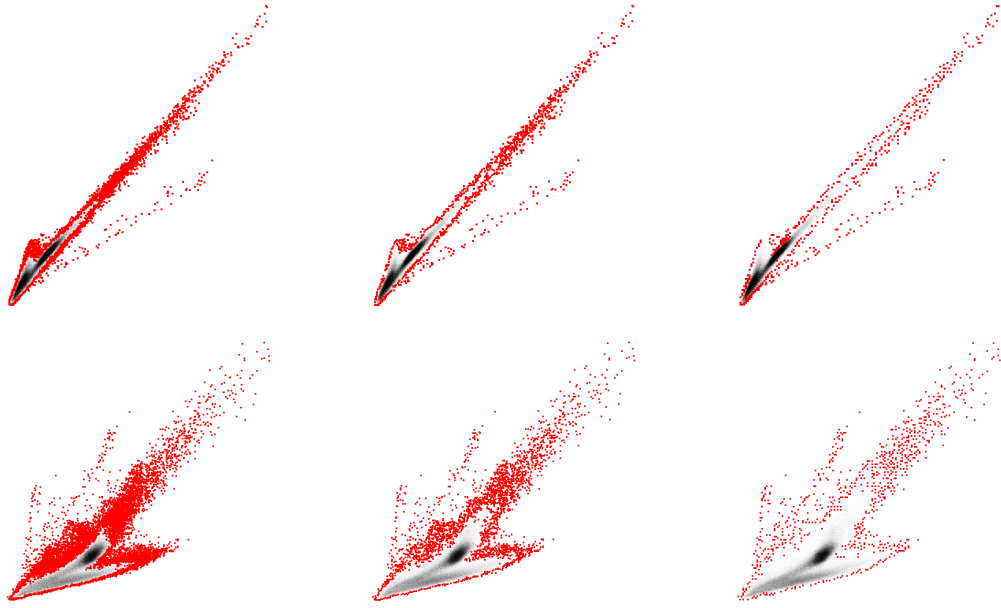


Fig. 3.23: Left 5x5 blocks, Middle 10x10 blocks, Right 20x20 blocks

The results show that using the POHMT we are able to detect all outlying pixels in the scatter plot. By increasing the cell size we are also able to define how many points are detected, with a smaller cell size producing more points and a larger cells size producing fewer.

3.7.1 Band Selection

As there are over 100 bands in each of the images various methods were explored to create the scatter plot that the POHMT technique can be used on. The four best that

have been explored so far are detailed below.

Random

The first and most basic technique was to use two randomly selected bands to create the scatter plot. These bands must be kept at least 20% of the total number of bands apart. The idea behind this is that it gives a very quick way of detecting outliers in the dataset.

5 Random

A variation of the random technique is to do the random process 5 times. Cumulate the outliers found for all five, then discard all the outliers that only appear once. This method is slightly slower than the first but allows for more accurate outlier detection as more of the data is included in the calculations.

Principal Component Analysis (PCA)

The next alternative approach for band selection that was looked at was based on principal component analysis first proposed by Pearson in 1901 [133]. The idea was to use the first two principal components to create the scatter plot. The POHMT technique is then performed on this scatter plot and the outliers are produced as normal. The idea behind this is that in the original method two randomly selected bands are used, therefore the majority of the data is discarded. By using the first two principal components, this scatter plot more accurately represents the entire dataset [134].

Normalized Difference Vegetation Index (NDVI)

The final method that was explored was to use a variation of the NDVI [50, 135]. This is normally used to classify pixels that are vegetative and non-vegetative. This is done by looking at a band in the near-infrared (NIR) region, approximately 750nm, and a band in the visible region, approximately 650nm. These points in the spectrum are either side of the chlorophyll edge, or red edge, where there is a rapid change in reflectance of vegetation. Equation 3.6 is used to calculate the difference between the bands. An

example of how this can be applied is shown in Fig. 3.24. The idea behind this is it would allow us to select bands that would highlight man-made targets.

$$NDVI = \frac{NIR - VIS}{NIR + VIS} \quad (3.6)$$

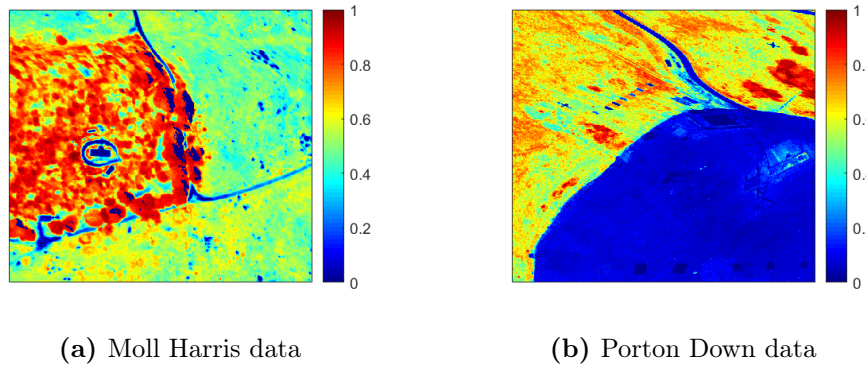


Fig. 3.24: NDVI plot. Low value indicates non-vegetative pixel, high value indicates vegetative surface

3.8 Results and Analysis

Before it would be possible to compare any produced techniques against any existing techniques it was necessary to locate every target in both images, see Fig. 3.25.

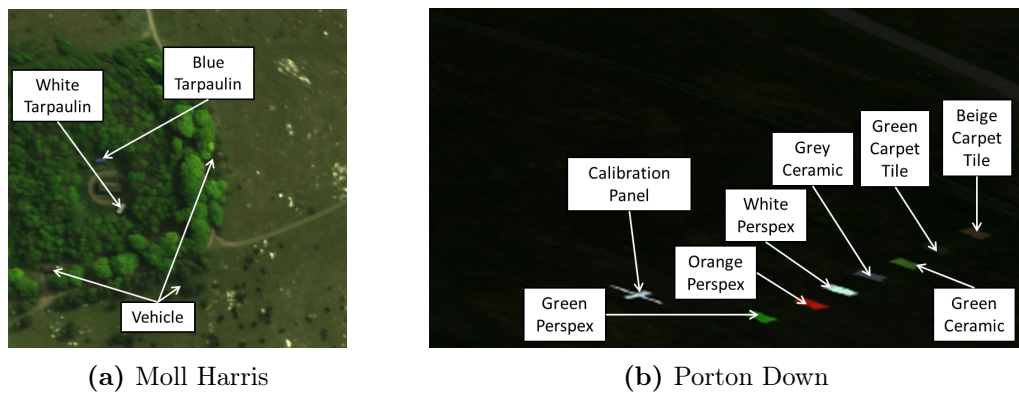


Fig. 3.25: Target locations

Both the SMACC and MD algorithms were then run on both dataset the results are shown in Figs 3.26 and 3.27. Also shown below are statistics for targets detected, targets

missed and false positives for each algorithm, see Table 3.2. The results show that for the Moll Harris dataset the SMACC algorithm produces the best results as it detects 4 out of the 5 targets. For the Porton Down dataset the results are fairly similar and the SMACC algorithm is only slightly better as it produces fewer false positives. From this it is confirmed that there is indeed room for improvement on both these algorithms.

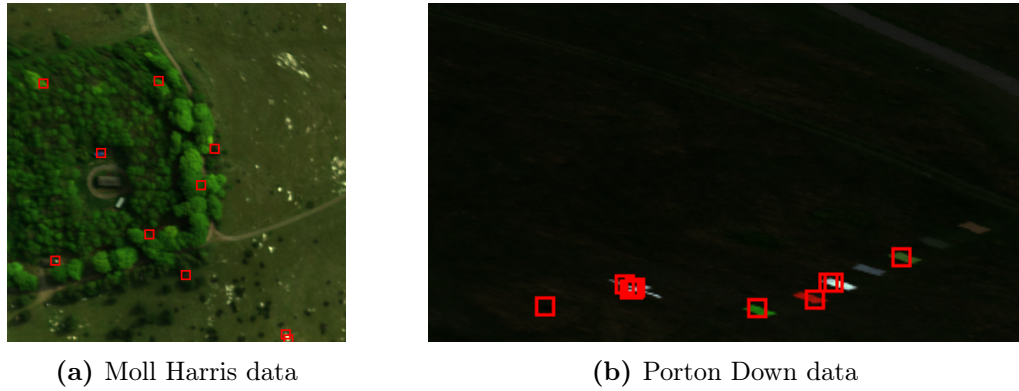


Fig. 3.26: SMACC results

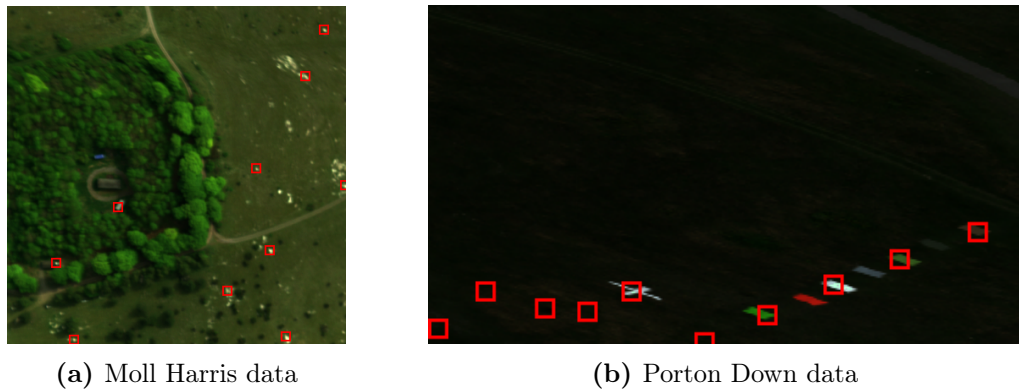


Fig. 3.27: Mahalanobis distance results

Dataset	Algorithm	Targets Detected	Targets Missed
Moll Harris	SMACC	4	1
Moll Harris	MD	2	3
Porton Down	SMACC	5	3
Porton Down	MD	5	3

Table 3.2: SMACC and MD results for both datasets

As the proposed algorithm has several variables that will each produce different

results, in order to compare the methods, a ground truth for the image was created; see Fig. 3.28. Each separate algorithm was then run on the image to produce a set of targets. This was then compared with the ground truth and the score was calculated by using Equation 3.7.

$$Score = \frac{Targets\ Found}{No.\ of\ Targets} \quad (3.7)$$

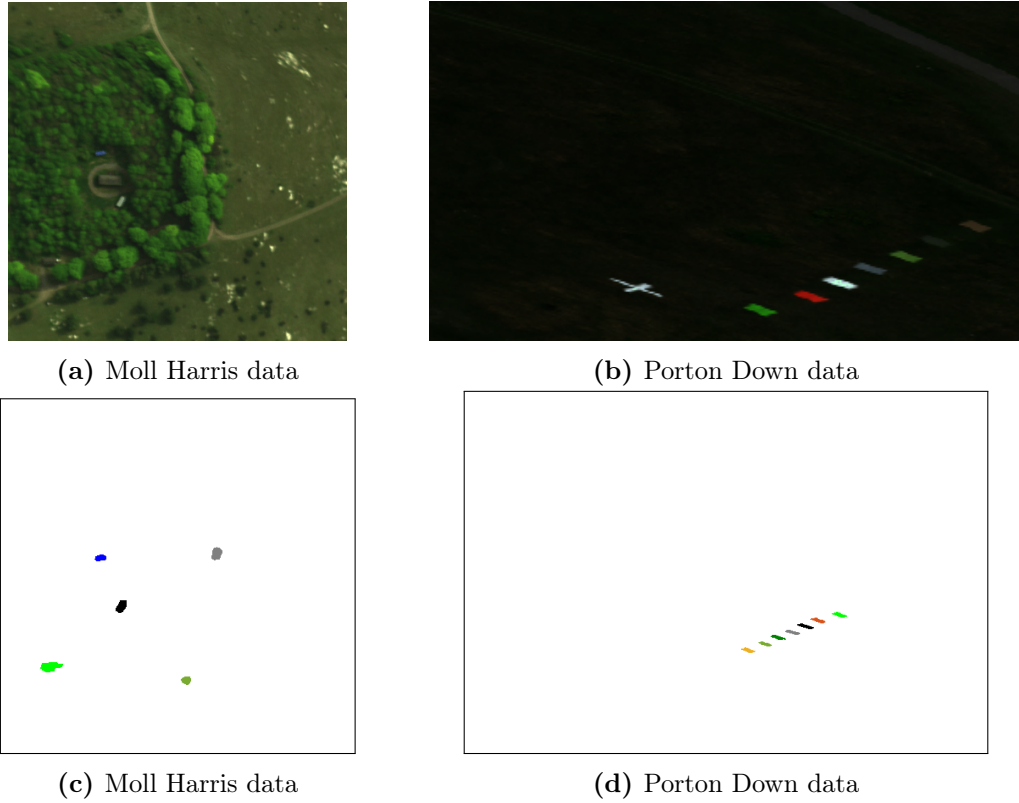


Fig. 3.28: Ground truth

For every method a cell size from 5x5 to 150x150 was used going up in increments of 5. As both the PCA and NDVI produce the same results every time, these were both run once and the results recorded. Due to the random nature of the "Random" and "5 Random" techniques, these were both run 1000 times and the average score for each cell size was recorded. The results are shown in Fig. 3.29. Also shown are the two standard anomaly detection approaches discussed earlier, the Sequential Maximum Angle Convex Cone [76] and Mahalanobis Distance [84].

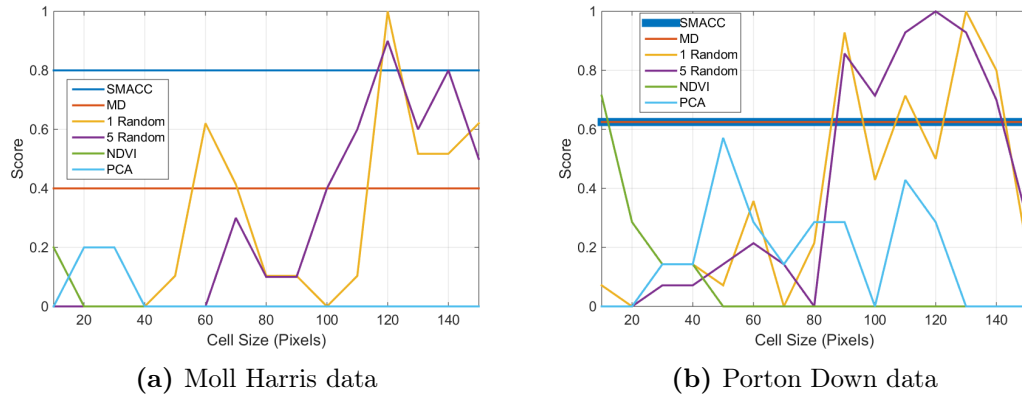


Fig. 3.29: Results for each algorithm for every cell size

3.8.1 Speed Test

Although in some cases the POHMT will calculate the anomalies faster than both the other techniques, due to the random nature of the method the time is not consistent compared with the other two, see Fig. 3.30. For cell sizes between 20 and 50, the time for all techniques except the POHMT using the PCA bands is similar. When the cell size is increased beyond 70 pixels the time taken also increases quite drastically.

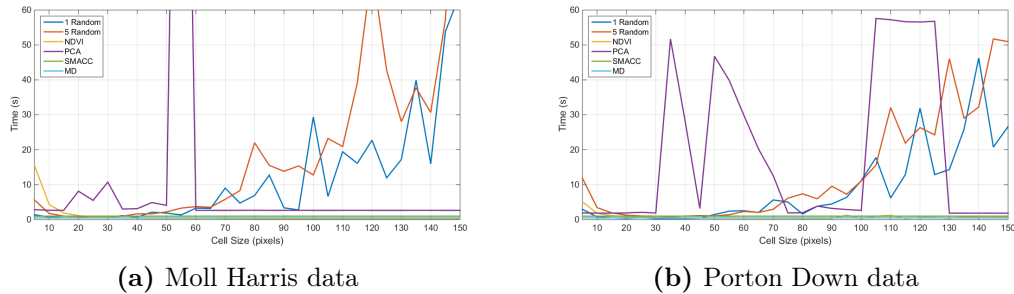


Fig. 3.30: Speed test for each algorithm on both datasets

3.9 Summary

For the purpose of aerial hyperspectral imagery anomaly detection is a very important area. In this chapter, we proposed a new type of anomaly detection technique based on the Percentage Occupancy Hit or Miss Transform [14], as well as trying to adapt and improve upon the Mahalanobis Distance [84] for several remotely sensed hyperspectral

images. Both techniques were then used on two hyperspectral images, each with various different targets present, the results were then compared with an established technique for endmember extraction the Sequential Maximum Angle Convex Cone [76].

One of the key parameters for this technique is the initially selected bands to create the scatter plot. It was discussed in detail the various methods that could be used to select these bands, however without any prior knowledge of the target, which for the majority of practical applications is unlikely, it proved very difficult to select any optimal bands that would display all targets. The best that can be achieved by using this technique is to display potential target locations and for applications where there is a limit on data transmission or storage, this could be used to send or store only the points of interest where the rest of the data could be discarded. Another key parameter is the cell size, even from the limited data available by exhaustively trying every cell size between 5x5 and 150x150 it is clear that there is no one cell size that will suit every dataset or even one cell size to suite each image. Due to the nature of how the scatter plot is created there are numerous factors that can change the cell size required, e.g. bit depth of the data, spectral reflectivity and atmospheric absorption in bands selected.

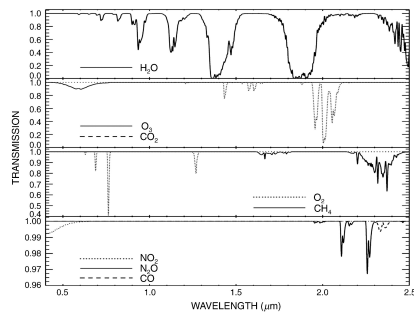
For the majority of cell sizes, this method does not yield a better result. However, the main object of this chapter was to produce an improved anomaly detection method. Under certain circumstances, this has been achieved using the POHMT with a pair of random bands selected, see Fig. 3.29, as it outperforms all the current anomaly detection techniques tested.

Chapter 4

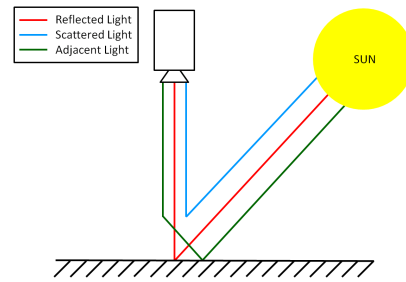
Atmospheric Modelling Using MODTRAN for VNIR Imagery

4.1 Introduction

For aerial hyperspectral imagery, target detection is a very important research area, one of the key components of this target detection process is the atmospheric correction stage. When a hyperspectral camera takes an image of a scene, the radiance received by the camera can be expressed as a combination of the ground reflectance characteristics, the atmospheric attenuation (see Fig. 4.1a), the atmospheric scattering and the solar illumination spectrum, excluding the reflectance characteristics of the ground the rest



(a) Simulated transmittance spectra of atmospheric aerosols using Atmospheric Removal Program (ATREM) [136]



(b) Atmospheric scattering

Fig. 4.1: Atmospheric effects on aerial hyperspectral imagery

are all dependent on the location and time of the measurement. Shown in Fig. 4.1 are the three paths that the light measured by the observer will take. The main path (red) is the light that is directly reflected off the object, the scattered light (blue) is the light that is reflected off some aerosol in the atmosphere and the adjacent light (green) is the light that is reflected off a material near to the object being measured and this light is diverted back to the observer by aerosols in the atmosphere [137].

The atmosphere is made up of approximately thirty gasses, some of these such as water vapour (H_2O), carbon dioxide (CO_2), ozone (O_3), nitrous oxide (N_2O), carbon monoxide (CO), methane (CH_4), oxygen (O_2) and nitrogen dioxide (NO_2) cause light on both the sun to ground and ground to observer path to be absorbed [58]. Fig. 4.1a shows the affects these eight gases have in the 400 nm to 2500 nm region of the spectrum [136]. Another aerosol that significantly affects the measurement of hyperspectral data is dust, as it is composed of several different types of minerals, the particles that make up the dust both absorb and scatter light. The third and final type of aerosols are those that are man-made, these include both smoke from burning trees and sulphate aerosols from burning either coal or oil.

The next issue that needs to be considered is the viewing zenith angle (see Fig. 4.2), for this application, it can be assumed that this is 180 degrees, directly above and looking down on the object. The solar zenith angle, however, is dependent on the time, day and location which will be different for each data cube.

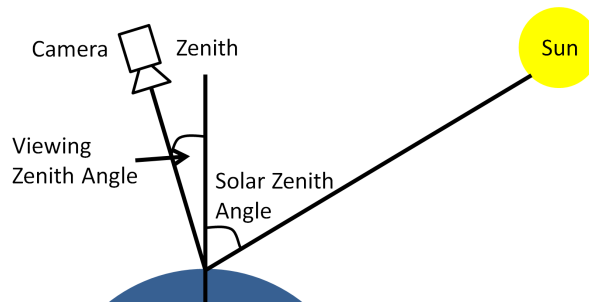


Fig. 4.2: Viewing and solar zenith angle

Because of all these effects since the mid-1980s, there have been various techniques developed to attempt to remove or reduce the atmospheric effects in hyperspectral

data. [63, 64, 66–70, 72].

MODTRAN may be used to estimate certain key absorption and scattering parameters pertaining at the time of measurement, e.g. atmospheric profile, water vapour content, carbon dioxide mixing ratio, observer altitude, viewing zenith angle, date, time and location. In this chapter, the aim is by using MODTRAN, create a look-up table in which these parameters change, this could then be used to convert recorded radiance domain hyperspectral images into reflectance domain images that can be used for improved spectral matching purposes.

The rest of this chapter is structured as follows. Section 4.2 introduces the hyperspectral data and the key parameters about each image necessary to complete a MODTRAN simulation. An initial simulation is then performed using the PcModWin 5 [138] software. The process of creating an interface in MATLAB [139] to read and write MODTRAN LTN files is detailed in Section 4.3. Following this, the process of creating a batch simulation that can be used to create a MODTRAN look-up table is detailed in Section 4.4. Section 4.5 discusses the various user input methods, and how they can be used to set-up a batch simulation. Using this developed interface, in Section 4.7 detailed analysis is performed on various parameters of a MODTRAN simulation to determine which of these parameters are important and which are not. A summary of the contributions is then presented in Section 4.9.

4.2 Hyperspectral Data

4.2.1 Details

The only two data cubes initially available where all the necessary details that have to be known to complete a MODTRAN simulation were the Moll Harris and Operation 7 data cubes (see Fig. 4.3). These were both supplied by BAE Systems Military Air & Information and are remotely sensed visible to near-infrared hyperspectral (VNIR) imagery. Shown below in Table 4.1 are the required parameters for each data cube to complete a MODTRAN simulation.



(a) Moll Harris



(b) Operation 7

Fig. 4.3: False-colour representations of hyperspectral images (636nm, 555nm 460nm respectively for RGB)

	Moll Harris 18-5 1129	OP7 18-5 1119
Height of Sensor	0.781km	0.781km
Time	11:29am	11:19am
Day	18th May	18th May
Latitude	51°N	51°N
Longitude	2°W	2°W
Atmospheric Profile ¹	Sub-Arctic Summer	Sub-Arctic Summer

¹ Atmospheric profile selected by using Table 4.2

Table 4.1: Dataset details

Latitude (°N)	Jan	March	May	July	Sept	Nov
80 (Greenland)	SAW	SAW	SAW	MLW	MLW	SAW
70 (Alaska)	SAW	SAW	MLW	MLW	MLW	SAW
60 (Oslo)	MLW	MLW	MLW	SAS	SAS	MLW
50 (Prague)	MLW	MLW	SAS	SAS	SAS	SAS
40 (Madrid)	SAS	SAS	SAS	MLS	MLS	SAS
30 (Cairo)	MLS	MLS	MLS	T	T	MLS
20 (Mexico City)	T	T	T	T	T	T
10 (Ho Chi Minh City)	T	T	T	T	T	T
0 (Singapore)	T	T	T	T	T	T
-10 (Lima)	T	T	T	T	T	T
-20 (São Paulo)	T	T	T	MLS	MLS	T
-30 (Bloemfontein)	MLS	MLS	MLS	MLS	MLS	MLS
-40 (Wellington)	SAS	SAS	SAS	SAS	SAS	SAS
-50 (Falkland Islands)	SAS	SAS	SAS	MLW	MLW	SAS
-60 (Antarctica)	MLW	MLW	MLW	MLW	MLW	MLW
-70 (Antarctica)	MLW	MLW	MLW	MLW	MLW	MLW
-80 (Antarctica)	MLW	MLW	MLW	MLW	MLW	MLW

Table 4.2: Seasonal latitude surface temperature profile [140]

4.2.2 MODTRAN Atmospheric Profile Description

Having used Table 4.2 to select the appropriate built-in atmospheric profile for each data cube, further research had to be performed to determine if the values in each of these models were appropriate for the current images. Table 4.3 shows how the model dependent variables change with atmospheric profile, also shown below in Table 4.4 are the model independent gases that do not change with the atmospheric profile.

Profile Dependent Variables

	T	MLS	MLW	SAS	SAW	US	Units
Pressure	904	902	897	896	888	899	mbar
Temperature	294	290	269	282	259	282	K
Water Vapour (H₂O)	19 500	13 800	3450	8700	1620	6070	ppmv
Ozone (O₃)	0.0315	0.0334	0.0280	0.0294	0.0201	0.0296	ppmv
Nitrous Oxide (N₂O)	0.320	0.320	0.320	0.310	0.320	0.320	ppmv
Carbon Monoxide (CO)	0.145	0.145	0.150	0.145	0.145	0.145	ppmv
Methane (CH₄)	1.70	1.70	1.70	1.70	1.70	1.70	ppmv

Table 4.3: Atmospheric profile description at 1 km [141]

Profile Independent Variables

Aerosol	0 km	1 km	Units
Carbon Dioxide (CO₂)	330	330	ppmv
Oxygen (O₂)	209×10^3	209×10^3	ppmv
Nitric Oxide (NO)	300×10^{-6}	300×10^{-6}	ppmv
Sulfur Dioxide (SO₂)	300×10^{-6}	274×10^{-6}	ppmv
Nitrogen Dioxide (NO₂)	23×10^{-6}	23×10^{-6}	ppmv
Ammonia (NH₃)	500×10^{-6}	500×10^{-6}	ppmv
Nitric Acid (HNO₃)	50×10^{-6}	59.6×10^{-6}	ppmv

Table 4.4: Profile independent variables [141]

Having found the values for the atmospheric profile built into MODTRAN, it was now possible to compare these values with real world values at the time the datasets were taken. Several of these parameters, i.e. atmospheric pressure, water vapour, carbon diox-

ide, ozone, nitrous oxide, carbon monoxide and methane, historical values are readily available and could be checked against the MODTRAN profiles. However for the remaining parameters to find an accurate historical value for them is next to impossible, e.g. nitric oxide, sulphur dioxide and ammonia, because of this to define a specific model for each dataset will be impossible. Therefore it was decided based on the fact that all the available known parameters were more or less correct, the built in models would be sufficient.

4.2.3 MODTRAN Setup

From the required parameters shown in Table 4.1 it was possible to complete a MODTRAN simulation using the PcModWin 5 [138] software. This software acts as a graphical user interface (GUI) for the main MODTRAN atmospheric modelling code. Using this GUI a simulation was set-up for the Moll Harris data cube, see Fig. 4.4. The file created by this software is called an LTN file which contains all the necessary details to complete the simulation in MODTRAN (see Fig. 4.5). As there is no functionality built into the PcModWin 5 [138] software to easily create a batch of simulations, the process of manually reading and writing these LTN files in MATLAB [139] is detailed in the next section.

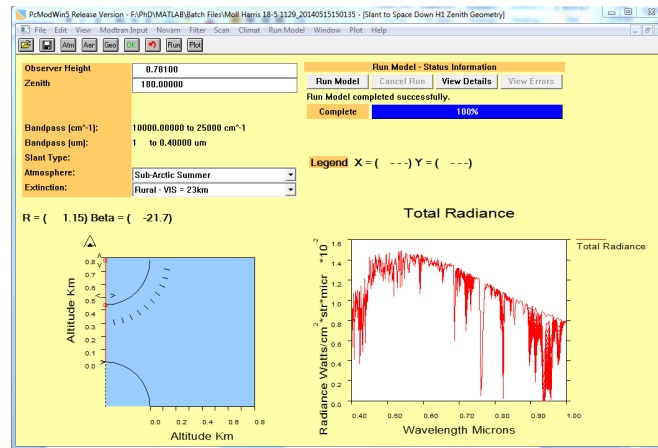


Fig. 4.4: Initial set-up for Moll Harris data cube

```

T F 4 3 2 0 0 0 0 0 0 0 0 0 0.0000.00000 0.000 0.000 0
fff 2 0 360.000 0.000000 0.0000000 F F F F 0.000 0.000 0.000 0
1 0 1 0 0 0 0 0.000 0.000 0.000 0.000 0.000 0.000
0.781000 0.000000180.000000 0.000000 0.000000 0.000000 0 0.000000 0.000000
1 0 138 0 0 0 0.000 0.000 0.000 0.000 0.000 0.000
51.000 1.000 0.000 0.000 11.483 0.000 0.000 0.000
10000.000025000.00001.0000E+06 1.00000 W 0 0.000 0.000
0

```

Fig. 4.5: LTN file for initial set-up

4.3 MODTRAN 5.2.1 MATLAB User Interface

4.3.1 LTN File Explained

To produce a database using the MODTRAN atmospheric modelling code, it would be necessary to develop an interface between MODTRAN and MATLAB [139]. Before this task was performed, research was done into what code may already be available to do this. Currently there is a "*Matlab Class Wrapper for MODTRAN 4*" written by Griffith [142]. However, due to the significant changes between MODTRAN 4 and MODTRAN 5, this code has been rendered obsolete. Also written by Griffith is a MODTRAN 5.0 implementation [143], however when attempting to run simulations set up

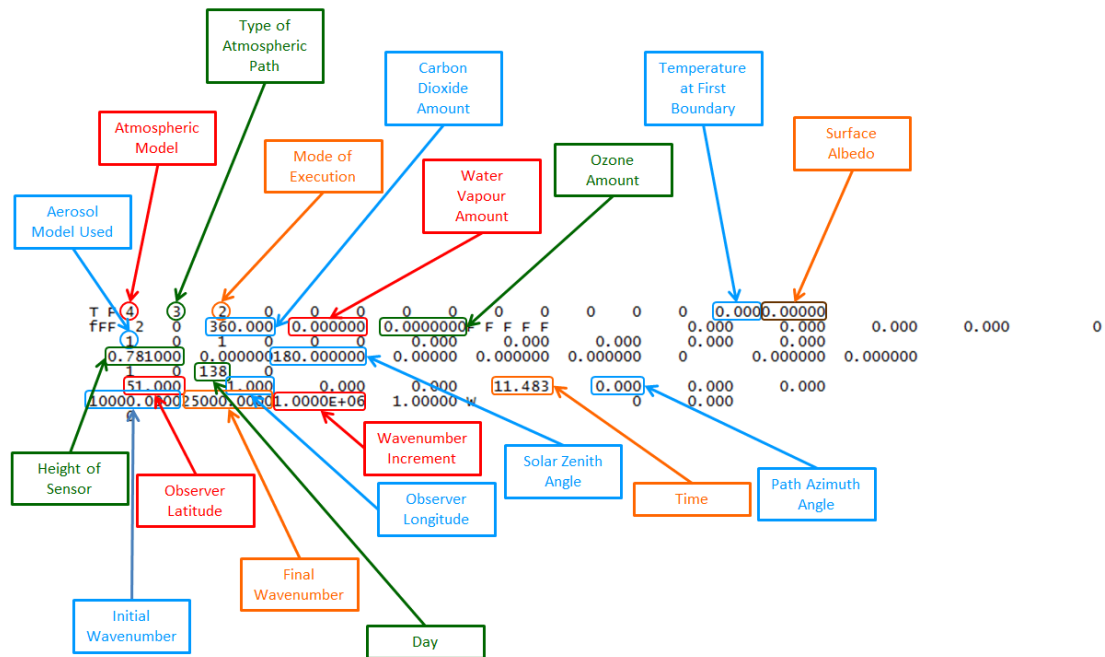


Fig. 4.6: LTN file with important variables highlighted

for MODTRAN 5.2.1 numerous errors were encountered. From reading the reviews of the MATLAB [139] toolbox it was clear that these errors had also been encountered by other people attempting to use the software. This is believed to be due to the subtle differences in several Cards between MODTRAN 5.0 and 5.2.1. Even if these issues could all be resolved, and it is unclear if they could, it was decided to create a bespoke interface that would be designed specifically to create a look-up table for MODTRAN.

Having produced an initial LTN file in the previous section using the PcModWin 5 [138] software, before it would be possible to generate multiple LTN files using MATLAB [139]. Each of the important variables had to be found in the LTN file, this was done by using the MODTRAN documentation [73] to determine where each of the important variables were stored. Fig. 4.6 highlights where each of the important variables are defined in the LTN file. Also shown in Appendix A is a detailed explanation of each of the important MODTRAN input parameters we are looking to vary.

4.3.2 Format

Once the LTN file had been deciphered, the next stage was to look at the format of each card (each line in the LTN file). Again by looking at the MODTRAN documentation [73] it was found that the format for each card is defined by a series of three character variables, the format for Card 1 is shown in Fig. 4.7. The first of these three character variables indicates the number of parameters, the second the type of parameter and the third the number of characters in the string. For example, 4A1 indicates four parameters each defined by a single character string, 11I5 would be eleven parameters each defined by a five character length integer (see Table 4.5). Shown in Appendix B is the format for all five cards and each of their sub-cards in the MODTRAN software.

**CARD 1: MODTRN, SPEED, BINARY, LYMOLC, MODEL, ITYPE, IEMSCT, IMULT,
M1, M2, M3, M4, M5, M6, MDEF, I_RD2C, NOPRNT, TPTEMP, SURREF
FORMAT (4A1, I1, 11I5, 1X, I4, F8.0, A7)**

Fig. 4.7: Card 1 format from MODTRAN documentation [73]

Variables	Format
MODTRAN	A1
SPEED	A1
BINARY	A1
LYMOLC	A1
MODEL	I1
ITYPE	I5
IEMSCT	I5
IMULT	I5
M1	I5
M2	I5
M3	I5
M4	I5
M5	I5
M6	I5
MDEF	I5
I_RD2C	I5
NOPRNT	I4
TPTEMP	F8.0
SURREF	A7

¹ A is a Character

² I is an Integer

³ F is a Float

Table 4.5: Card 1 variables and format

4.3.3 Reading a Card

As the structure of the LTN files and the format of each card are now known. Shown in Fig. 4.8 is an example of what the string for Card 1 can look like, using MATLAB R2015b [139] the data from this string can be extracted. Fig. 4.9 shows pseudocode for how this was done, Appendix C.1 and C.2 also show the MATLAB [139] code used for Card 1. Having extracted the data from the first card the same process was then repeated for each of the remaining cards.

T F 6 2 0 0 0 0 0 0 0 0 0 0 0.000 0.00
--

Fig. 4.8: Example of Card 1 string

- Input:** String representing every parameter in card
Output: Individually stored parameters for card in structured array
- 1 FieldWidth = [1 1 1 1 1 5 5 5 5 5 5 5 5 5 1 4 8 7];
 - 2 FieldFormat = 'c', 'c', 'c', 'c', 'd', 'd', 'd','d','d','d','d','d','d','d','d','d','*', 'd', 'f', '7c';
 - 3 Read input string, based on FieldWidth and FieldFormat;
 - 4 Store all output parameters in structured array;

Fig. 4.9: Pseudocode for reading Card 1

4.3.4 Reading an LTN File

Once the individual functions were set-up to read each card, it was then necessary to create a new function that would analyse the LTN file, and based on the inputs decide which cards to read for each line. Shown in Fig. 4.10 is pseudocode for that function. As before Appendix C.3 has the MATLAB [139] code used to read the complete LTN file.

- Input:** LTN filename and path
Output: Individually stored parameters for complete LTN file in structured array
- 1 Open LTN file, and store all lines as strings;
 - 2 Read all parameters from Card 1;
 - 3 Read all parameters from Card 1A;
 - 4 Based on already imported values determine which remaining Card 1 inputs are necessary (1A1 to 1A7, 1B and 1B Alt);
 - 5 Read all parameters from Card 2;
 - 6 Based on already imported values determine which remaining Card 2 inputs are necessary (2A, 2A Alt, 2B to 2E);
 - 7 Read all parameters from Card 3, 4 and 5;
 - 8 Store all imported parameters in structured array;

Fig. 4.10: Function for reading complete LTN file

4.3.5 Writing an LTN File

Having created all the functions for reading cards and LTN files, it was then possible to reverse this process so that all the information extracted from the files could be written to another LTN file. This is necessary as it will allow us to vary key parameters and then generate a batch of LTN files that can be run in MODTRAN. As before the pseudocode

for writing an LTN file is shown below (see Fig. 4.11), also shown in Appendix C.4 and C.5 is the MATLAB [139] code used to write Card 1 and an LTN file respectively.

Input: LTN filename and storage path

Output: LTN file containing all necessary card details

- 1 Open blank LTN file, for storing all data;
- 2 Write all parameters for Card 1;
- 3 Write all parameters for Card 1A;
- 4 Based on stored values determine which remaining Card 1 parameters are necessary (1A1 to 1A7, 1B and 1BAlt);
- 5 Write all parameters for Card 2;
- 6 Based on stored values determine which remaining Card 2 parameters are necessary (2A, 2A Alt, 2B to 2E);
- 7 Write all parameters for Card 3, 4 and 5;

Fig. 4.11: Function for writing LTN file

4.4 Batch Mode

4.4.1 Generate Batch of LTNs

Now that the basic functions were in place for both reading and writing LTN files, the next step was to look into how to run a number of these in a batch mode, rather than having to load them each individually into PcModWin 5 [138]. Built into the PcModWin 5 [138] software is a function that allows for the creation of multiple runs, using this feature it is possible to create any number of runs, however, they each need to be set-up individually and there is no option to vary one parameter across multiple runs. To better understand how this capability works an example simulation with multiple runs was set-up. The resultant LTN file is shown below, see Fig. 4.12, from this it was very easy to determine how the multiple run function worked. Each run was stored in a single LTN file with a "1" at the end indicating there was another run or "0" to indicate it was the last run. Therefore rather than creating a number of LTN files, it would be possible to create a single LTN file that would contain all the information for each run of the database. This could easily be implemented using the current code by placing the write function in a series of "for" loops, the MATLAB [139] code to do this is shown in Appendix C.6.

```

TF4 3 2 0 0 0 0 0 0 0 0 0 0 0.0000.00000
fff 2 0 360.000 0.000000 0.0000000F FFFF 0.000 0.000 0.000 0.000 0
1 0 1 0 0 0 0.000 0.000 0.000 0.000 0.000
0.781000 0.000000180.000000 0.00000 0.000000 0.000000 0 0.000000 0.000000
1 0 138 0
51.000 1.000 0.000 0.000 11.483 0.000 0.000 0.000
10000.000025000.00001.0000E+06 1.00000 W 0 0.000
1
TF4 3 2 0 0 0 0 0 0 0 0 0 0 0.0000.00000
fff 2 0 360.000 0.000000 0.0000000F FFFF 0.000 0.000 0.000 0.000 0
1 0 1 0 0 0 0.000 0.000 0.000 0.000 0.000
0.781000 0.000000180.000000 0.00000 0.000000 0.000000 0 0.000000 0.000000
1 0 138 0
51.000 1.000 0.000 0.000 11.483 0.000 0.000 0.000
10000.000025000.00001.0000E+06 1.00000 W 0 0.000
1
TF4 3 2 0 0 0 0 0 0 0 0 0 0 0.0000.00000
fff 2 0 360.000 0.000000 0.0000000F FFFF 0.000 0.000 0.000 0.000 0
1 0 1 0 0 0 0.000 0.000 0.000 0.000 0.000
0.781000 0.000000180.000000 0.00000 0.000000 0.000000 0 0.000000 0.000000
1 0 138 0
51.000 1.000 0.000 0.000 11.483 0.000 0.000 0.000
10000.000025000.00001.0000E+06 1.00000 W 0 0.000
0

```

0 – If last run
1 – If another run

Fig. 4.12: LTN file containing multiple runs

Once the batch LTN file was created using this method, the simulation was then performed using MODTRAN. Having done this it was noted that all the results were stored in one output file, therefore it would be extremely difficult to determine which order MODTRAN had done the calculations. To try and improve this a shell program was created that would run in a folder and in sequence run a number of LTN files in MODTRAN, pseudocode for this is shown in Fig. 4.13, and the MATLAB [139] code to do this is shown in Appendix C.7. Using this method a separate output file will be created for each run, so it will be not only easier to read the results it will also be

Input: Multiple LTN files
Output: Multiple FL7 files, each for separate LTN file

- 1 Retrieve all filenames in root directory;
- 2 **for** *Every LTN file in directory* **do**
- 3 Copy file to MODTRAN directory;
- 4 Run MODTRAN from command line;
- 5 Copy output file (FL7) to results directory;
- 6 **end**

Fig. 4.13: Function for writing multiple LTN files

possible to process the results before all the simulations have been complete.

4.5 User Input

As it was now possible to create a batch of LTN files and then run these using MODTRAN, the next step was to develop a method that the user could input the parameters for the simulation and decided on which of these to vary. Fig. 4.14 shows the concept for three types of user input required and the various steps that would be required for each to integrate with the current code.

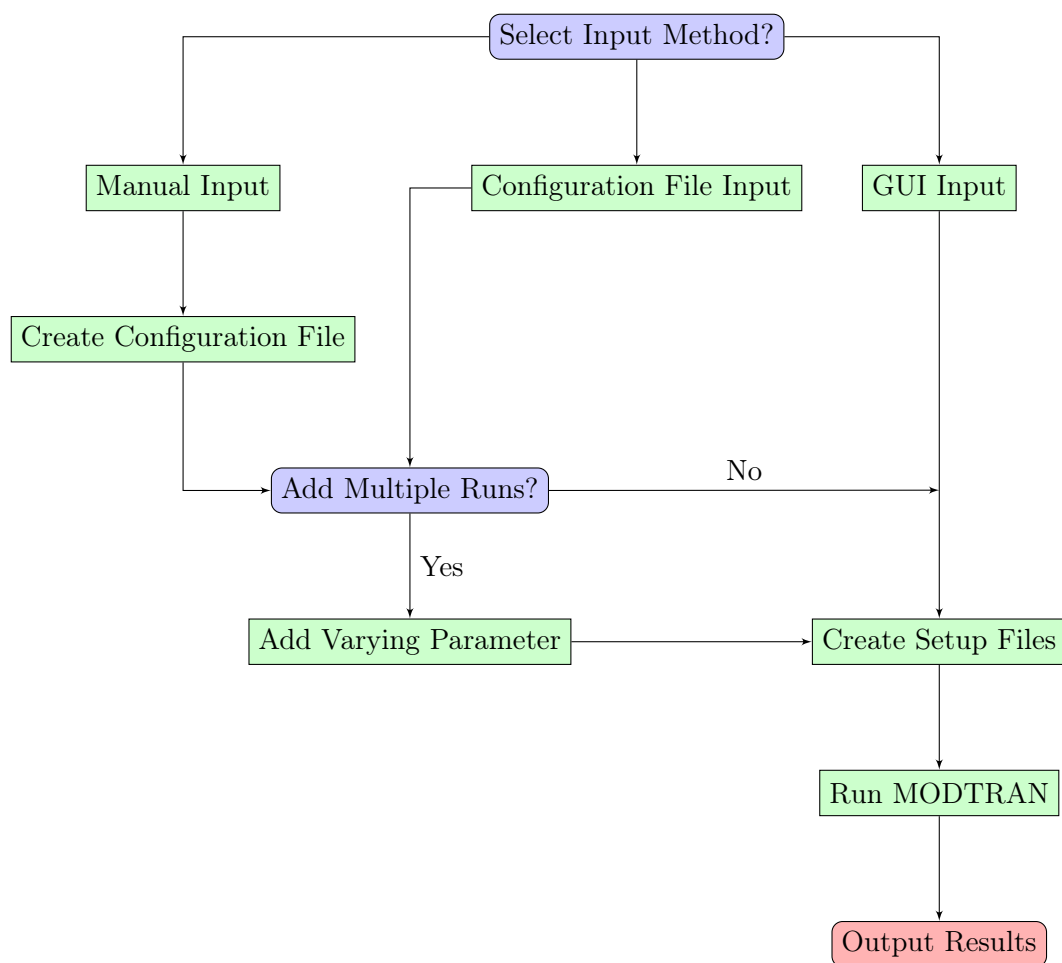


Fig. 4.14: Complete process for each type of user input

4.5.1 Manual Input

The first and most basic input method is the manual input method. If the manual input method is selected a prompt for each parameter appears in the MATLAB [139] command window, see Fig. 4.15. After the user has entered the value for each parameter, the user is again prompted with an option to vary any of the variables into an array of variables, which will each be run separately. Once all this has been input the function then generates a configuration file like the one shown in Fig. 4.16. The MATLAB [139] code used to do this is shown in Appendix C.8.

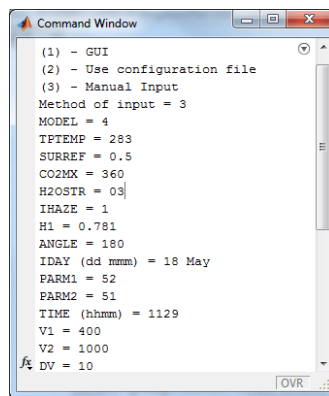


Fig. 4.15: Manual input function running in MATLAB command window

4.5.2 Configuration File Input

If the configuration file input method is selected the user is asked for a path of a configuration file, the structure of these files is shown in Fig. 4.16, this can also be

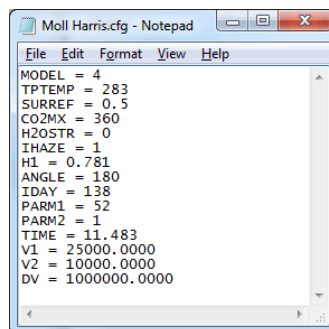


Fig. 4.16: Configuration file for Moll Harris dataset

generated using the manual input method. This is all done by using the code shown in Appendix C.9.

4.5.3 GUI Input

The final input method is to use a graphical user interface (GUI). The MATLAB App Building toolbox [144] provides an extensive set of tools for creating GUIs, one of these the graphical user interface development environment (GUIDE) was used to create a GUI for this application, see Fig. 4.17. This allows for the input of every parameter and also adds the ability to vary each parameter individually.

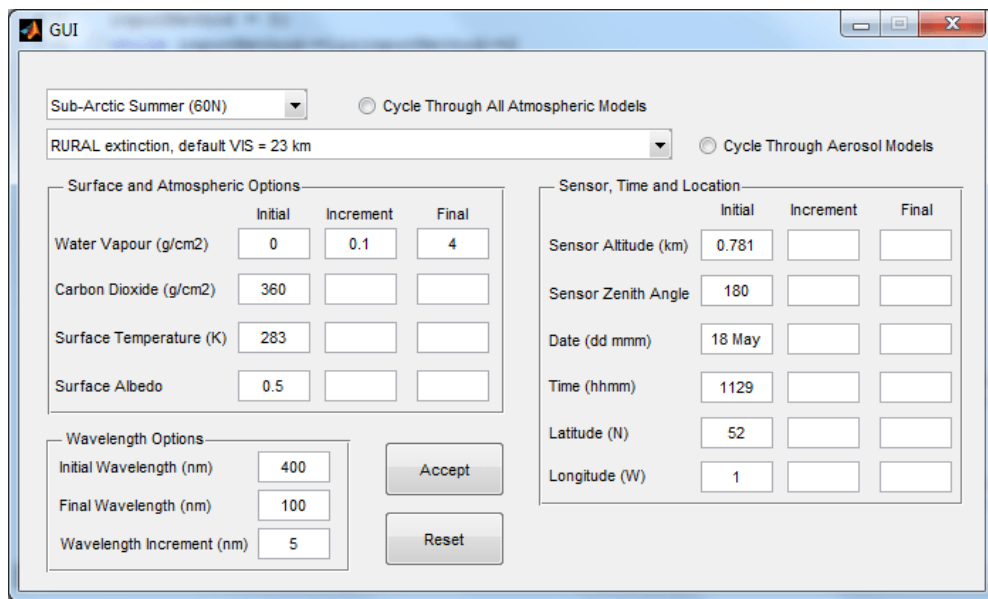


Fig. 4.17: MATLAB GUI

4.6 Saving the Output Data

4.6.1 Read MODTRAN Output Files

After any simulation is run, MODTRAN outputs all the data into a single MODOUT2 or FL7 file, the file structure of these files is shown below (see Fig. 4.18). The initial few lines relate to the LTN file used to generate the results, and the rest relates to the output radiance profiles. The data is output in a number of columns each for a

specific property, the three columns that are of interest are the *FREQ*, *SOL_SCAT*, and *TOTAL_RAD* columns, these relate to the frequency or wavenumber, the solar scattering, and the total radiance respectively.

```

T F 6 3 2 0 6 6 6 6 6 6 1 0 0 288.150 LAMBER
1 1 1 0 0 0 30.00000 0.00000 0.00000 0.00000 0.00000
-99.000 -99.000 -99.000
-99.00000 -99.00000 -99.00000 1.416445 0.000736 ! H2O & O3 COLUMNS [GM/CM2]
361976 U S STANDARD
2.00000 0.00000 180.00000 2.00000 0.00000 0.00000 0
0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
10000.0 22222.0 1.0 1.0NW
1 0 0.000 0 0.000 0 1.000
FREQ TOT_TRANS PTH_THRML THRML_SCT SURF_EMIS SOL_SCAT SING_SCAT GRND_RFLT DRCT_RFLT TOTAL_RAD
10000.00 0.92999816 7.0981E-24 0.0000E+00 3.8878E-23 3.4431E-08 3.4431E-08 1.6544E-06 1.6544E-06 1.6888E-06
10001.00 0.93020999 7.0429E-24 0.0000E+00 3.8705E-23 3.4401E-08 3.4401E-08 1.6530E-06 1.6530E-06 1.6874E-06
10002.00 0.92858231 7.1586E-24 0.0000E+00 3.8456E-23 3.4089E-08 3.4089E-08 1.6348E-06 1.6348E-06 1.6689E-06
10003.00 0.92908448 7.0866E-24 0.0000E+00 3.8297E-23 3.4248E-08 3.4248E-08 1.6433E-06 1.6433E-06 1.6776E-06
10004.00 0.91426247 8.4490E-24 0.0000E+00 3.7509E-23 3.3359E-08 3.3359E-08 1.5758E-06 1.5758E-06 1.6091E-06
10005.00 0.92810678 7.1140E-24 0.0000E+00 3.7899E-23 3.4262E-08 3.4262E-08 1.6415E-06 1.6415E-06 1.6757E-06
10006.00 0.93004870 6.8927E-24 0.0000E+00 3.7800E-23 3.4410E-08 3.4410E-08 1.6518E-06 1.6518E-06 1.6862E-06
10007.00 0.92935377 6.9284E-24 0.0000E+00 3.7595E-23 3.4621E-08 3.4621E-08 1.6606E-06 1.6606E-06 1.6952E-06
10008.00 0.92629635 7.1850E-24 0.0000E+00 3.7296E-23 3.4440E-08 3.4440E-08 1.6459E-06 1.6459E-06 1.6804E-06
10009.00 0.92806935 6.9862E-24 0.0000E+00 3.7193E-23 3.4380E-08 3.4380E-08 1.6459E-06 1.6459E-06 1.6803E-06
10010.00 0.93008095 6.7596E-24 0.0000E+00 3.7099E-23 3.4472E-08 3.4472E-08 1.6537E-06 1.6537E-06 1.6882E-06
10011.00 0.92756623 6.9586E-24 0.0000E+00 3.6825E-23 3.4349E-08 3.4349E-08 1.6430E-06 1.6430E-06 1.6774E-06
10012.00 0.92998111 6.7046E-24 0.0000E+00 3.6748E-23 3.4411E-08 3.4411E-08 1.6501E-06 1.6501E-06 1.6845E-06

```

Fig. 4.18: MODOUT2 or FL7 file structure

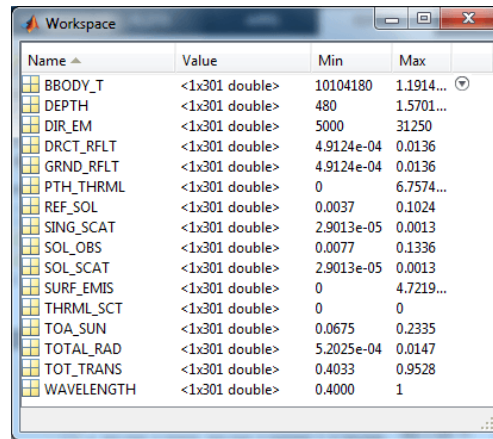
Using the FL7 file shown in Fig. 4.18 a function was created in MATLAB [139] that would extract all the data for each column in the output file and save it into a MATLAB [139] MAT file, see Fig. 4.20. Pseudocode for how this was performed is shown in Fig. 4.19 and the MATLAB [139] code used to do this is shown in Appendix C.10. Also shown in Appendix D is a table with a description of all these output variables including their units.

Input: User input via command window

Output: Structured array of all variables required to create LTN file

- 1 Open LTN file, and store all lines as strings;
- 2 Find first line relating to output variables;
- 3 **for** *Remaining lines in LTN file* **do**
- 4 Split line into 16 floats;
- 5 Store each value in separate vector;
- 6 **end**
- 7 Convert data from wavenumber to wavelength;
- 8 Save all data to MATLAB MAT file for data processing

Fig. 4.19: Function for writing LTN file



Name	Value	Min	Max
BBDY_T	<1x301 double>	10104180	1.1914...
DEPTH	<1x301 double>	480	1.5701...
DIR_EM	<1x301 double>	5000	31250
DRCT_RFLT	<1x301 double>	4.9124e-04	0.0136
GRND_RFLT	<1x301 double>	4.9124e-04	0.0136
PTH_THRML	<1x301 double>	0	6.7574...
REF_SOL	<1x301 double>	0.0037	0.1024
SING_SCAT	<1x301 double>	2.9013e-05	0.0013
SOL_OBS	<1x301 double>	0.0077	0.1336
SOL_SCAT	<1x301 double>	2.9013e-05	0.0013
SURF_EMIS	<1x301 double>	0	4.7219...
THRML_SCT	<1x301 double>	0	0
TOA_SUN	<1x301 double>	0.0675	0.2335
TOTAL_RAD	<1x301 double>	5.2025e-04	0.0147
TOT_TRANS	<1x301 double>	0.4033	0.9528
WAVELENGTH	<1x301 double>	0.4000	1

Fig. 4.20: MAT file contents

4.7 Analysis

4.7.1 Initial Tests

An initial simulation was set up using the newly created interface, to test that the interface was working correctly. The atmospheric model was set to Sub-Arctic Summer, using Table 4.2, the latitude and longitude were set to match Wiltshire and the visibility set to Rural. The other variables that were set were the column water vapour which was set to 0 g/cm^2 , the temperature to 15°C and altitude of the observer to 1km. Also, the carbon dioxide (CO_2) mixing ratio was set to 365 ppmv, which was the amount found in the atmosphere in 1999 [145]. The parameter that was varied was the surface reflectance which was set to match Forest, Desert, Ocean, Sea Ice and Granite. The complete details of all five runs are shown in Table 4.6 and the results in Fig. 4.21.

From the results, it can be noticed that the atmospheric models are producing distinct spectra, with significant atmospheric effects present for the visible and near-infrared (VNIR) region of the spectrum. The general shape of all these spectra is correct when matched with the USGS Spectral Database [146], therefore it can be assumed that the software has been implemented correctly and the results produced are correct.

Parameter	Run 1	Run 2	Run 3	Run 4	Run 5
MODEL	SAS	SAS	SAS	SAS	SAS
TPTEMP	288K	288K	288K	288K	288K
SURREF	Forest	Desert	Ocean	Sea Ice	Granite
CO2MX	365ppmv	365ppmv	365ppmv	365ppmv	365ppmv
H2OSTR	0 g/cm ²	0 g/cm ²	0 g/cm ²	0 g/cm ²	0 g/cm ²
IHAZE	Rural	Rural	Rural	Rural	Rural
H1	1km	1km	1km	1km	1km
ANGLE	180°	180°	180°	180°	180°
IDAY	30th June	30th June	30th June	30th June	30th June
PARM1	51°N	51°N	51°N	51°N	51°N
PARM2	2°W	2°W	2°W	2°W	2°W
TIME	12pm	12pm	12pm	12pm	12pm

Table 4.6: MODTRAN settings

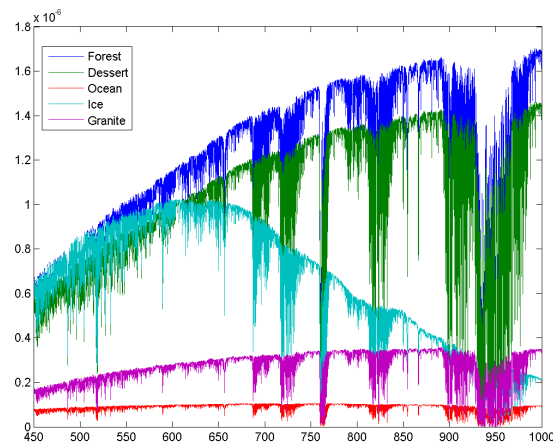


Fig. 4.21: Wavelength vs total radiance for various materials

4.7.2 MODTRAN Key Parameters

Having produced an interface for MODTRAN in MATLAB [139] it was then possible to generate a batch of atmospheric models. These models can then be used to look at which out of the ten parameters highlighted earlier produced the biggest effects on the atmospheric model. This was done by setting up a test simulation (see Table 4.7) and in turn varying each of the ten parameters while keeping the others constant. The results of the initial setup are shown in Fig. 4.22, this gives us a basis to compare any of the following results.

Parameter	Initial Setup
MODEL	SAS
TPTEMP	288K
SURREF	1.0
CO2MX	400ppmv
H2OSTR	0 g/cm ²
IHAZE	Rural - 23km
H1	1km
ANGLE	180°
IDAY	30th June
PARM1	51°N
PARM2	2°W
TIME	12pm

Table 4.7: MODTRAN initial simulation settings

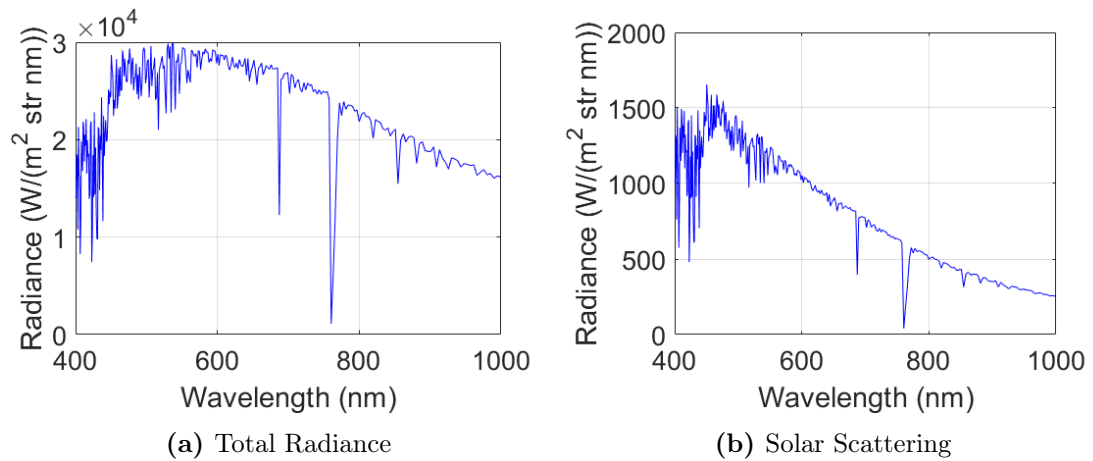


Fig. 4.22: MODTRAN initial output with no variation

Day of the Year (IDAY)

The first parameter that was varied was the day of the year (IDAY), as would be expected the summer months, April to July, have the highest received total radiance, while the winter months, November to February, have the smallest received total radiance (see Fig. 4.23). This is due to the tilt of the Earth's axis, in the summer the light from the sun hits the Earth at a steep angle, therefore the light does not spread out as much, so the radiance measured at any given spot is higher and the opposite in the winter when then light hits the Earth at a shallower angle. This same effect can also be observed

in the solar scattering contribution, the steeper the angle and the more atmosphere the light has to pass through the higher the solar scattering contribution. From these results, it is clear that to create an accurate atmospheric model in MODTRAN the day of the year must be known.

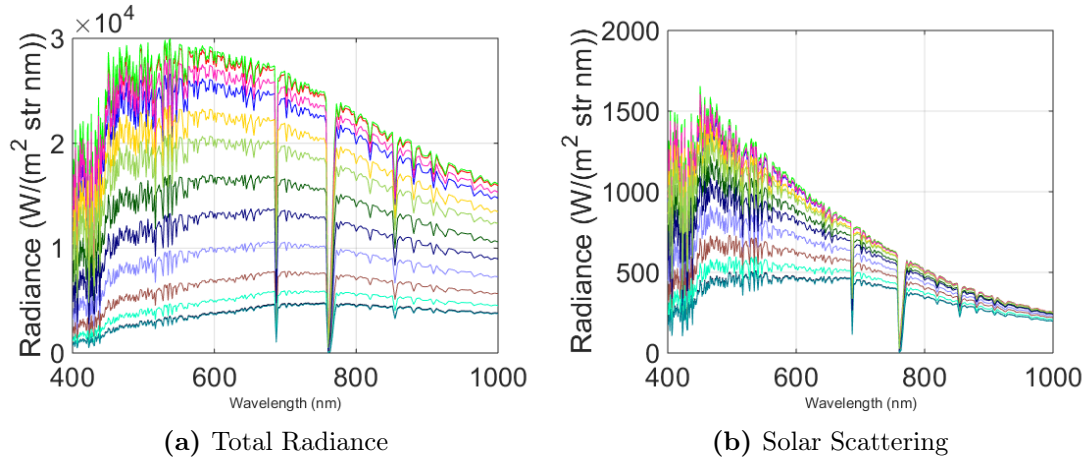


Fig. 4.23: MODTRAN output by varying Day of Year. 1st Jan (Black), 1st Mar (Navy), 31st Mar (Light Green), 30th Apr (Blue), 30th May (Red), 29th Jun (Green), 29th Jul (Pink), 28th Aug (Yellow), 27th Sep (Dark Green), 27th Oct (Lilac), 26th Nov (Cyan), 26th Dec (Dark Cyan)

Time of Day (TIME)

The next parameter that was varied was the time of day. Unsurprisingly at midday (12pm) the total radiance received is at its highest and at midnight (12am) the radiance received is zero (see Fig. 4.24). This relates back to the day of the year, the steeper the angle of incident light, when the sun is above the object, the higher the total radiance and solar scattering contribution. The shallower the angle, when the sun is rising or setting, the lowest measured radiance and solar scattering contribution, anything beyond these two extremes is zero for both radiance profiles. As before with the day of the year, it is clear from the results that the time of day is critical to creating an accurate atmospheric model.

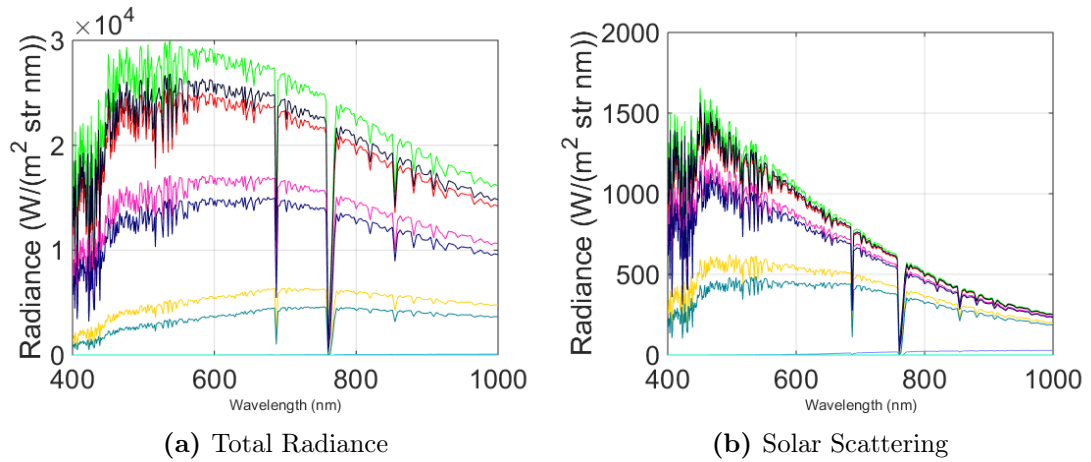


Fig. 4.24: MODTRAN output by varying Time. 12am (Blue), 2am (Dark Green), 4am (Cyan), 6am (Dark Cyan), 8am (Navy), 10am (Red), 12pm (Green), 2pm (Black), 4pm (Pink), 6pm (Yellow), 8pm (Lilac), 10pm (Brown)

Latitude of Observer (PARM1)

The observer latitude is the next variable to be considered, as the simulation was set for the 30th June, which is very near the summer solstice [147] this is when the Earth's rotational axis is most inclined towards the Sun (23.5°). This causes objects at the latitude of 23.5° to have the greatest total radiance received, and as the object moves

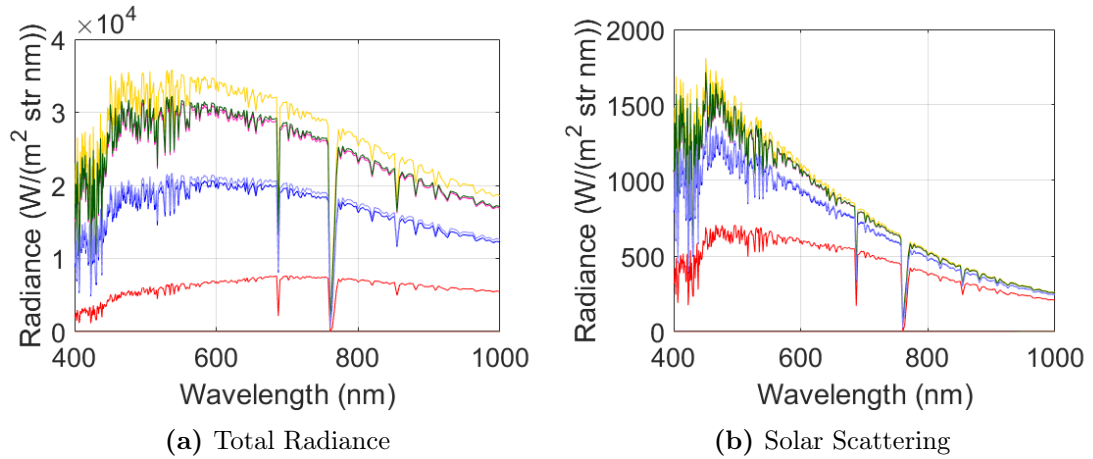


Fig. 4.25: MODTRAN output by varying Latitude. -90° (Black), -67.5° (Green), -45° (Red), -22.5° (Blue), 0° (Pink), 22.5° (Yellow), 45° (Dark Green), 67.5° (Lilac), and 90° (Brown)

away from this latitude it can be observed that the total radiance and solar scattering contribution decreases (see Fig. 4.25). Anything below -62.5° latitude is in darkness due to the Earth's axis tilt therefore the received radiance is zero.

Longitude of Observer (PARM2)

As MODTRAN works on Greenwich Mean Time (GMT), when varying the longitude the time must also be varied accordingly. For every degree moved West of the Greenwich Meridian 4 minutes ($\frac{1440}{360}$) is added to the local time (12pm) to calculate the exact GMT equivalent, using the location the time was also calculated based on time zone (see Table 4.8). The results show that there is absolutely no difference varying the longitude if the local time is always kept at 12pm (see Fig. 4.26). When using the time zone equivalent rather than the exact time, there is a minute difference in the results but nowhere near large enough to be considered noteworthy. Therefore as long as the GMT time is known the longitude has no effect on the accuracy of the atmospheric model.

Longitude	Location	Exact Time	Time Zone
0°W	United Kingdom	12:00	12:00
40°W	Greenland	09:20	09:00
80°W	Canada (East)	06:40	07:00
120°W	Canada (West)	04:00	04:00
160°W	US (West)	01:20	01:00
200°W	Japan	22:40	23:00
240°W	Russia (East)	20:00	20:00
280°W	Kazakhstan	17:20	17:00
320°W	Ukraine	14:40	15:00

Table 4.8: MODTRAN simulation settings

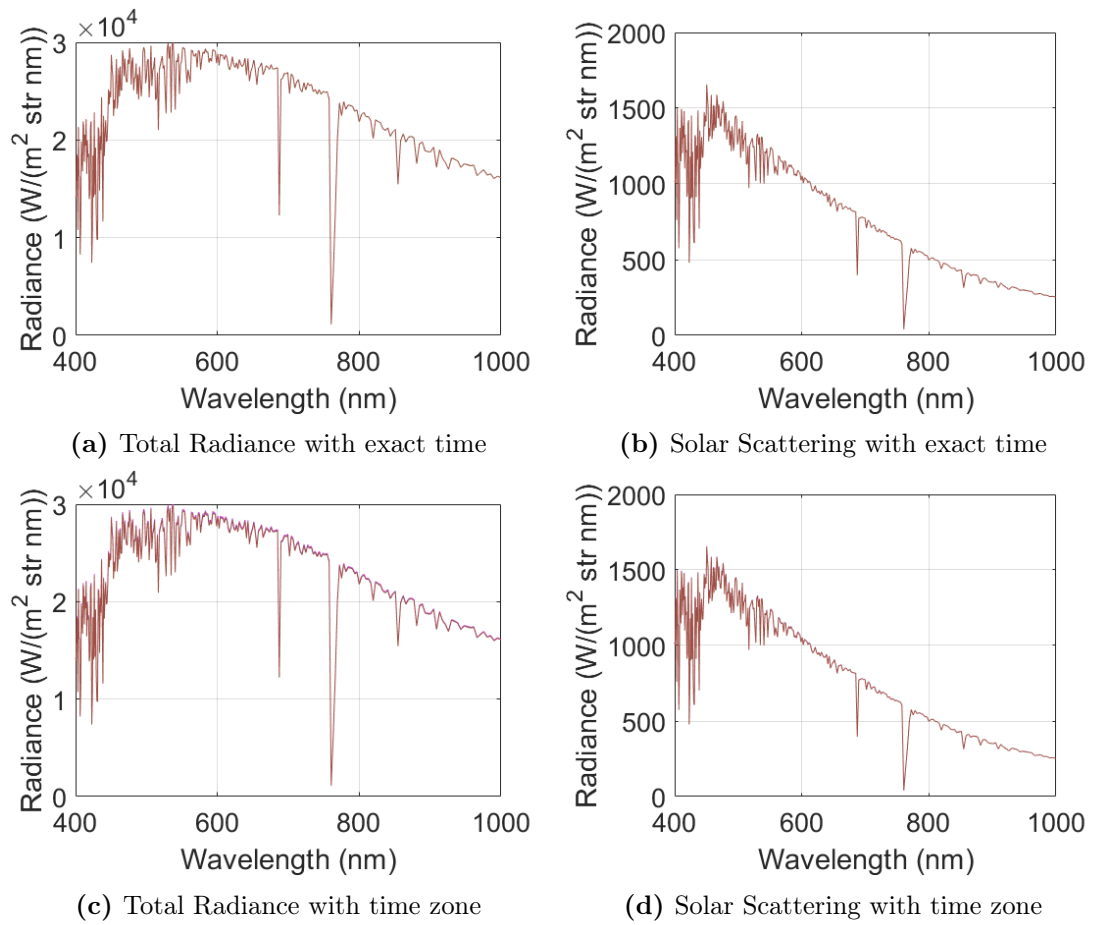


Fig. 4.26: MODTRAN output by varying Longitude. 0°W (Blue), 40°W (Lilac), 80°W (Brown), 120°W (Red), 160°W (Green), 200°W (Black), 240°W (Pink), 280°W (Yellow), 320°W (Dark Green)

Height of Observer (H1)

Another parameter that was varied was the height of the observer, this will obviously have a greater impact depending on the aerosol model, therefore, two simulations were performed. The first used the *Rural - 23km* aerosol model and the second the *Rural - 5km* model. From the results, it is clear that as the visibility decreases, the total radiance received by the observer also decreases (see Fig. 4.27). It can also be observed that due to the increase in aerosols the solar scattering contribution increases as more light is reflected off these aerosols and back to the observer. Based on these results like the day of the year, time and latitude the height can also be considered a crucial variable in

producing an accurate atmospheric model, especially depending on the aerosol model used.

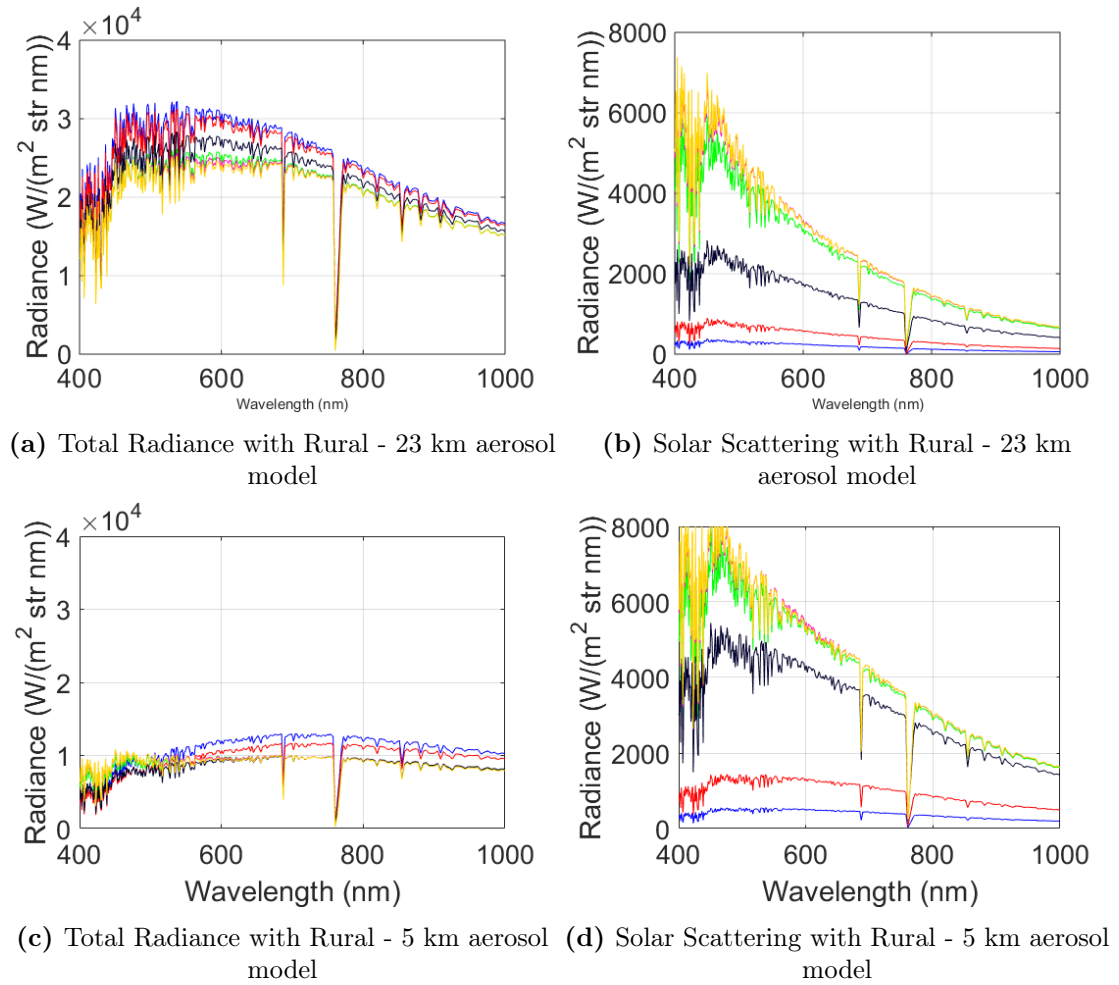


Fig. 4.27: MODTRAN output by varying altitude of observer. 0.2km (Blue), 0.5km (Red), 2km (Black), 10km (Green), 23km (Pink), 50km (Yellow)

Aerosol Model (IHAZE)

To confirm the previous statements about the aerosol model and height, the next parameter that was varied was the aerosol model while keeping the observer at 1km altitude. The results confirm what was stated previously, as the aerosol model visibility increases the total radiance received and the solar scattering contribution decreases and the opposite occurs as the aerosol model visibility decreases (see Fig. 4.28). One of the main technical limitations of VNIR imagery is its ability to penetrate fog is

very limited [148,149], unlike laser and thermal imaging. Therefore for both *Fog1* and *Fog2* atmospheric profiles it is unreasonable to assume any useful results from a VNIR hyperspectral camera at these visibilities, however, this parameter will definitely be considered as an important factor in atmospheric modelling.

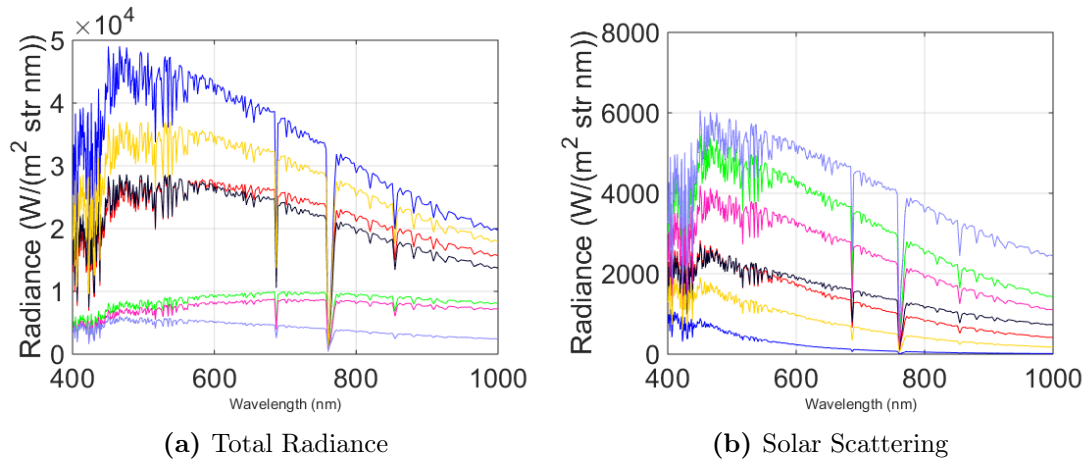


Fig. 4.28: MODTRAN output by varying IHAZE. No Aerosol (Blue), Rural - 23km (Red), Rural - 5km (Green), Maritime - 23km (Black), Urban - 5km (Pink), Tropospheric - 50km (Yellow), Fog1 - 0.2km (Dark Green), Fog2 - 0.5km (Lilac)

Viewing Zenith Angle (ANGLE)

The next parameter that was looked into was the viewing zenith angle, this is the angle the observer makes with a vertical line coming from the object (see Fig. 4.2). Varying this from 135° to 225° has very little effect on the total radiance received, this is mainly due to the difference in magnitude of the sun to ground path compared with the ground to observer path (see Fig. 4.29). Even at both extremes, the overall path length is still approximately the same. For the solar scattering contribution, however, the effects are more noticeable at both extremes the measured radiance is higher, this is simply due to the increase in path length on the ground to observer path. Even though there are more effects in the solar scattering at the extreme angles, the viewing angle has a very small effect overall on the atmospheric model, assuming this is kept at a reasonable angle.

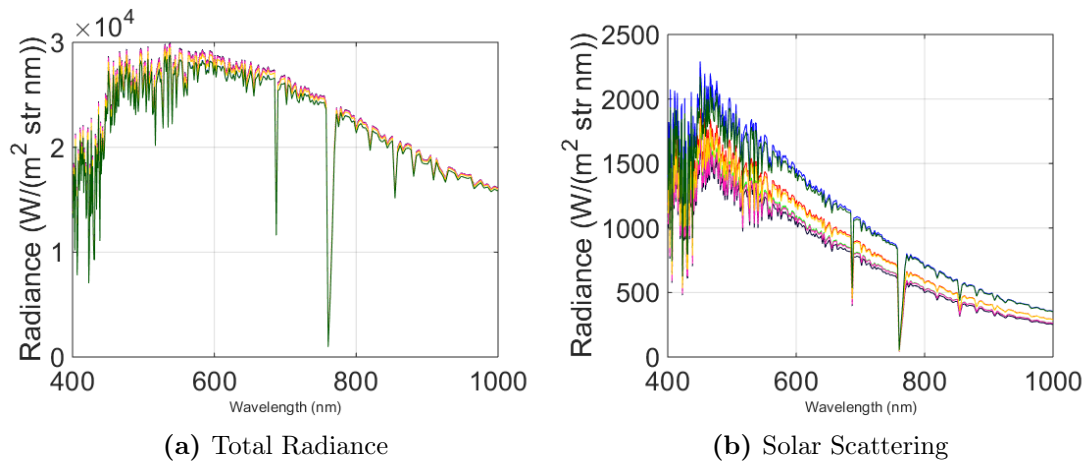


Fig. 4.29: MODTRAN output by varying zenith angle. 135° (Blue), 150° (Red), 165° (Green), 180° (Black), 195° (Pink), 210° (Yellow), 225° (Dark Green)

Atmospheric Profile (MODEL)

The six built-in atmospheric profiles were the next parameter to be varied these were Tropical, Mid-Latitude Summer, Mid-Latitude Winter, Sub-Arctic Summer, Sub-Arctic Winter and 1976 US Standard. The results show that from an altitude of 1km there is very little difference in all six atmospheric profiles (see Fig. 4.30a and 4.30b), when this altitude was increased to 23km the signs of the differences begin to show especially

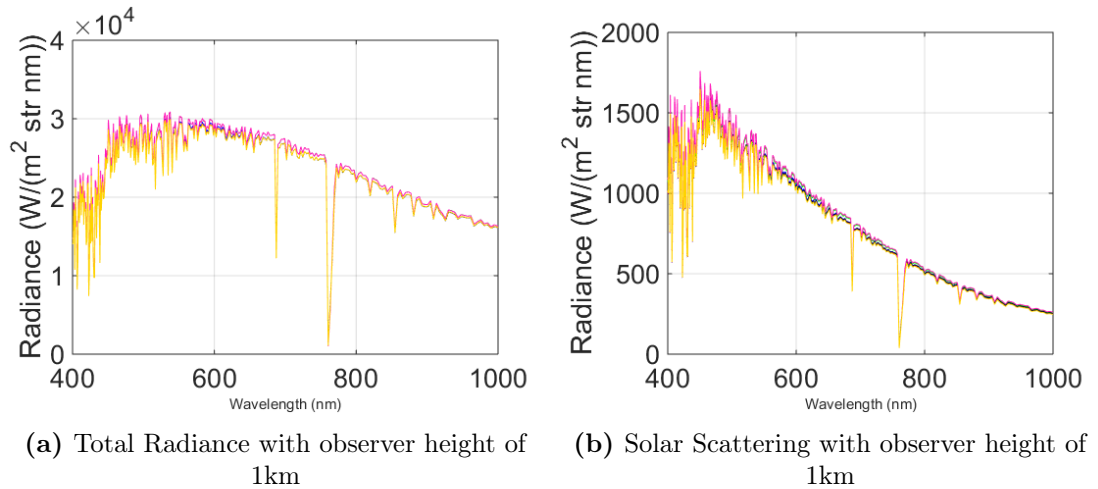


Fig. 4.30: MODTRAN output by varying MODEL. Tropical (Blue), MLS (Red), MLW (Green), SAS (Black), SAW (Pink), 1976 US (Yellow)

for the Sub-Arctic Winter profile (see Fig. 4.31a and 4.31b). However, these effects are small compared with other parameters, so as before this is another parameter that can be excluded from the list of key parameters for atmospheric modelling, unless used for an application where the altitude is far higher than the simulated altitudes for example for NASA's Earth Observation One which operates at an altitude of 705km [150].

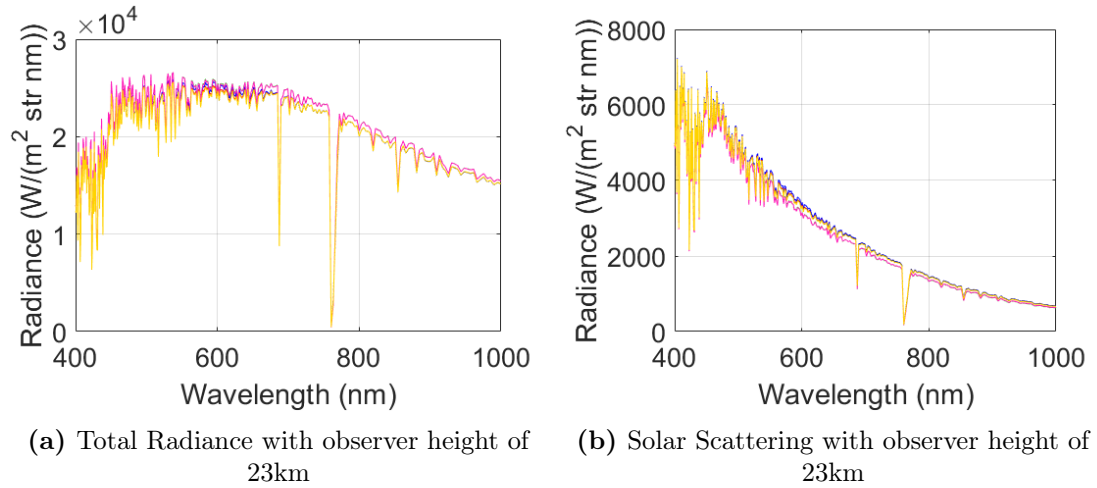


Fig. 4.31: MODTRAN output by varying MODEL. Tropical (Blue), MLS (Red), MLW (Green), SAS (Black), SAW (Pink), 1976 US (Yellow)

Column Water Vapour (H2OSTR)

The penultimate parameter to be varied was the column water vapour, this was varied from 0 g/cm² to 5 g/cm². Due to the numerous water absorption bands in the VNIR region of the spectrum (see Fig. 4.1a) the effects of this parameter are noticeable only in certain regions of the spectrum. A water vapour amount of 0 g/cm² represents the same as the initial results (see Fig. 4.22), however by altering this to 1 g/cm² the effects on the region between 900nm and 1000nm are already significant by the time the final value of 5 g/cm² is reached there is virtually no received spectral radiance. That being said other than the water absorption band at 760nm [151] there is very little additional effects on the radiance profile produced. For many applications in VNIR hyperspectral images, this parameter could be ignored, unless precise atmospheric correction was required for the above-mentioned water absorption bands.

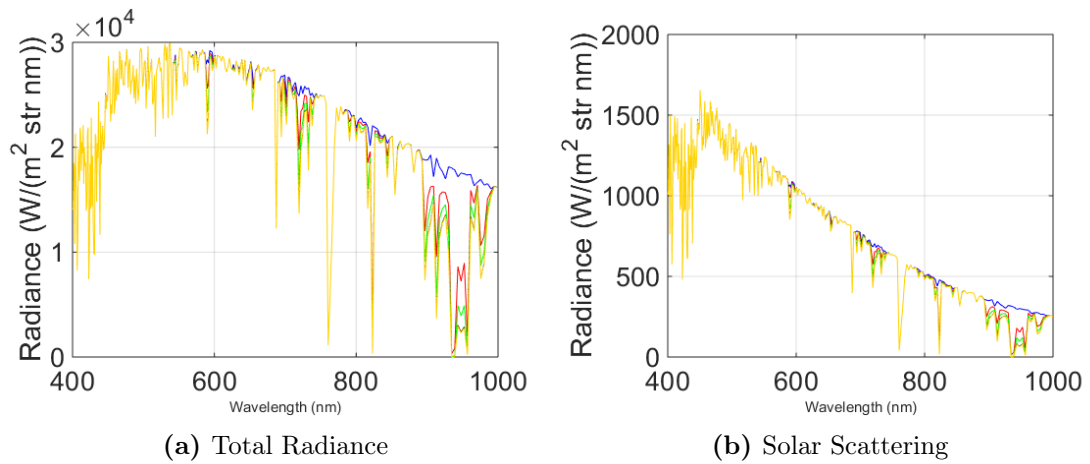


Fig. 4.32: MODTRAN output by varying H₂OSTR. 0 g/cm² (Blue), 1 g/cm² (Red), 2 g/cm² (Green), 3 g/cm² (Black), 4 g/cm² (Pink), 5 g/cm² Yellow

CO₂ Mixing Ratio (CO₂MX)

The final parameter that was varied was the carbon dioxide (CO₂) mixing ratio, this was varied from values pre-dating 1999 of 300ppmv [145] to values not expected for the next 40-60 years [152,153] of 500ppmv. Straight away it is clear that even high levels of CO₂ have no effect on the produced radiance profiles from MODTRAN in the VNIR region of the spectrum (see Fig. 4.33). Again this is another parameter that can be excluded from any future atmospheric modelling for this application.

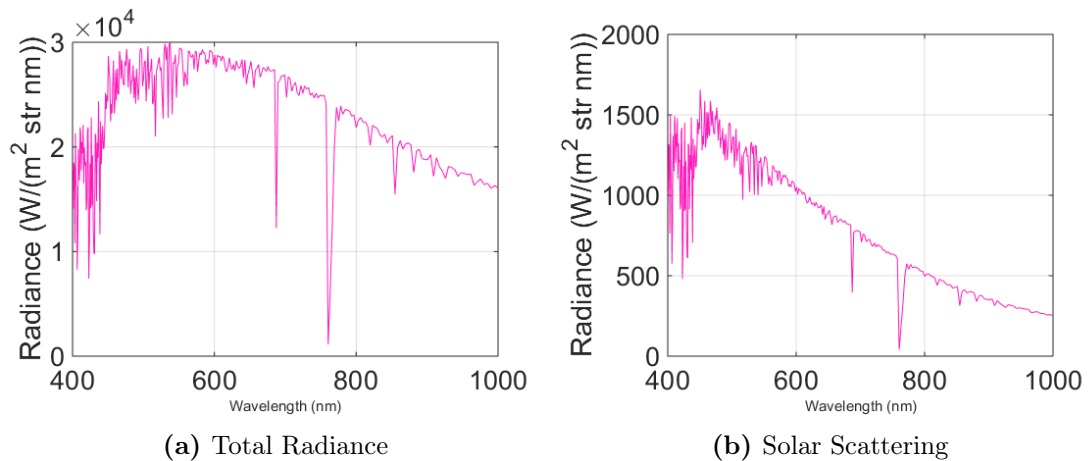


Fig. 4.33: MODTRAN output by varying CO₂MX. 300ppmv (Blue), 350ppmv (Red), 400ppmv (Green), 450ppmv (Black), 500ppmv (Pink)

4.8 Results

For the case of an aerial hyperspectral camera at an altitude of approximately 1 - 2km. From the list of ten parameters studied it was determined that six of these are essential for accurate atmospheric modelling (see Table 4.9). These parameters all have a significant effect on the MODTRAN produced radiance profiles for total radiance and solar scattering contribution. The four non-essential parameters found each has varying effects on the produced results the carbon dioxide mixing ratio and longitude both have zero effect while the atmospheric profile and viewing zenith angle have negligible effects on the results for this scenario.

Essential Parameter	Non-essential Parameter
Time of day	Atmospheric profile
Day of the year	Carbon dioxide mixing ratio
Latitude	Longitude
Height of the observer	Viewing zenith angle
Aerosol model	
Column water vapour	

Table 4.9: List of MODTRAN essential and non-essential parameters for low altitude VNIR imagery

4.9 Summary

For precise target detection in hyperspectral images, atmospheric correction is a necessary step. One way of performing this correction is by modelling the effects of the atmosphere in an atmospheric modelling software, e.g. MODTRAN, and then removing them from the data using various atmospheric correction techniques [63, 64, 66–70, 72]. In order to do this, however, an accurate model of the effects must be produced.

In this chapter a user interface was designed for MATLAB [139] to create various atmospheric models using MODTRAN 5.2.1, this would then produce radiance profiles for key information, i.e. total radiance received by the observer and solar scattering contribution. Using this created software various simulations were run where key parameters relating to the atmosphere and scene were varied.

The results of each of these simulations were then compared and it was found that parameters such as the time of day, the day of the year, latitude, the height of the observer and aerosol model all have significant effects on the produced radiance profiles for the VNIR region of the spectrum. Where other parameters such as the atmospheric profile, carbon dioxide (CO₂) mixing ratio, longitude and viewing zenith angle have small effects but are much less severe and can essentially be ignored in the calculations for this application. The one parameter that has severe effects in certain regions of the visible near-infrared spectrum is the column water vapour, due to water absorption bands at approximately 760 nm and from about 900 nm to 1000 nm there are major effects in these regions, however, the rest of the data is unaffected.

Chapter 5

Atmospheric Correction Using MODTRAN Radiative Transfer Models

5.1 Introduction

Atmospheric correction is a necessary step in the target detection process for all aerial hyperspectral imagery. As well as reflectance characteristics of the ground, the radiance received by an aerial imager depends on the spectrum of the incident solar illumination and wavelength dependence of the atmospheric attenuation at the specific location and time of the measurement. Due to various aerosols in the atmosphere, some of the light emitted from the ground is absorbed, see Fig. 5.1. There are approximately thirty gases that make up the atmosphere. However, only eight of these have visible absorption effects on hyperspectral images between the range of 0.45 and 2.50 μm). These gases are water vapour (H_2O), carbon dioxide (CO_2), ozone (O_3), nitrous oxide (N_2O), carbon monoxide (CO), methane (CH_4), oxygen (O_2) and nitrogen dioxide (NO_2) [58]. Another aerosol that significantly affects the measurement of hyperspectral data is dust. As it is composed of several different types of minerals, the particles that make up the dust both absorb and scatter sunlight [59,60]. The third and final type of aerosols are those that are man-made. These include both smoke from burning trees and sulphate aerosols

from burning either coal or oil [61,62].

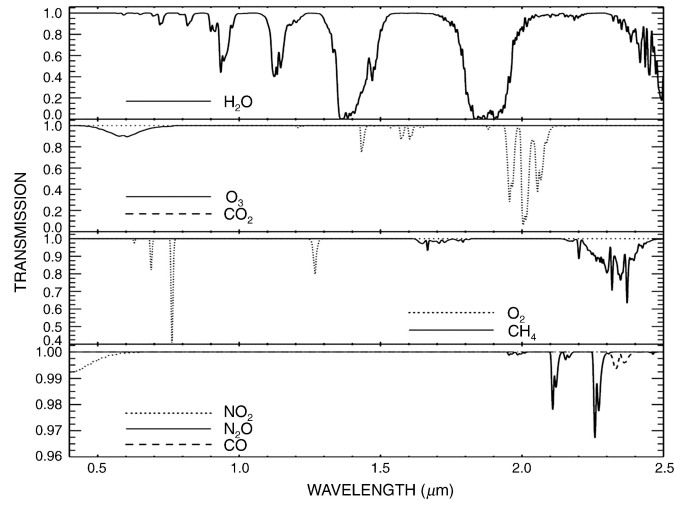


Fig. 5.1: Simulated transmittance spectra of atmospheric aerosols using Atmospheric Removal Program (ATREM) [136]

Also, a significant amount of scattering occurs where the radiance from nearby objects is also measured by the sensor this is called adjacent light, see Fig. 5.2. Because of this since the mid-1980s there have been various methods developed to attempt to remove or reduce these effects.

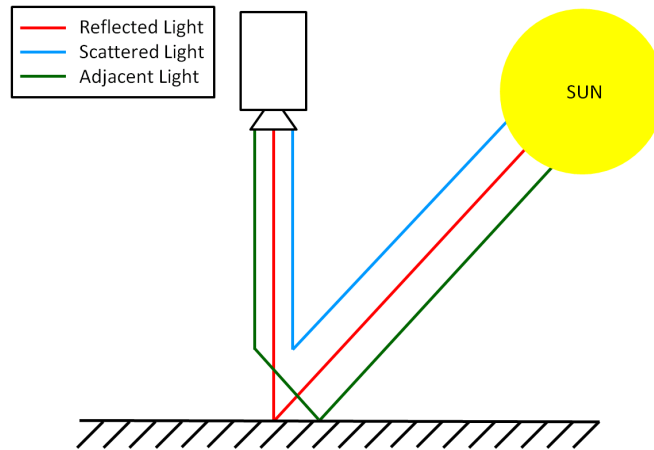


Fig. 5.2: Atmospheric scattering

In the previous chapter a system was proposed for atmospheric modelling based around the MODTRAN atmospheric modelling software using MATLAB. From this

its was found what parameters must be known to produce an accurate atmospheric model. In this chapter, we propose two new techniques, one based on the atmospheric modelling software MODTRAN and the other using prior known spectra to estimate which atmospheric effects are present in the hyperspectral image. These two techniques are then compared with two currently used scene-based approaches, namely Internal Average Relative Reflectance [64] and Dark Object Subtraction [63].

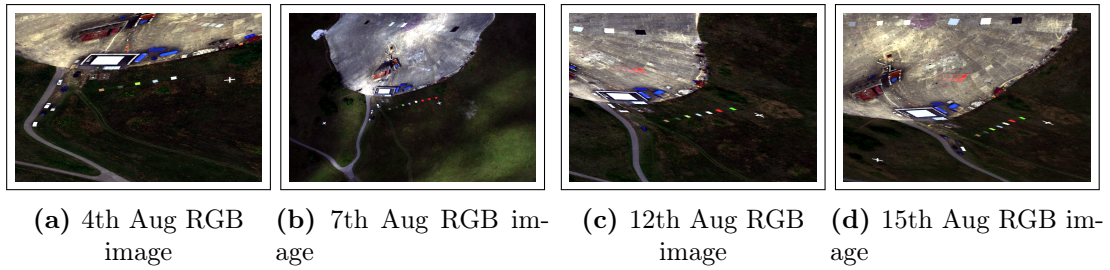
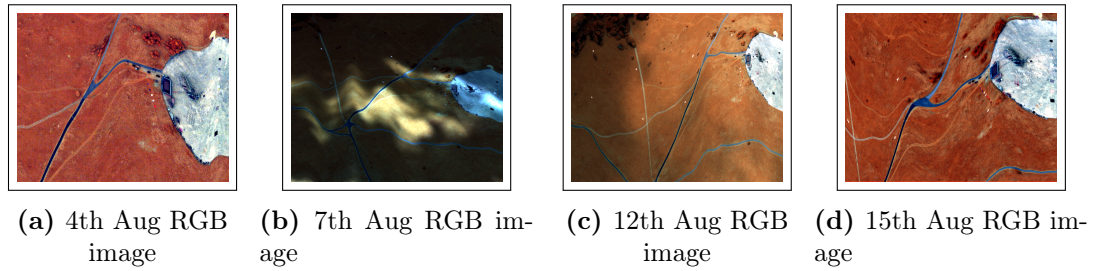
The rest of this chapter is structured as follows. Section 5.2 introduces the hyperspectral data, the targets contained within them and the process for producing the ground truth. The various atmospheric correction techniques both current and proposed methods are discussed in Section 5.3. Section 5.4 shows the research into the identification of the atmospheric effects present in the data, along with comparisons between the original spectra. Section 5.5 is devoted to the implementation of all the proposed techniques. Results and a summary of the contributions are then presented in Sections 5.6 and 5.7 respectively.

5.2 Hyperspectral Images

The dataset supplied contained hundreds of images so it was necessary to select a few of these that had similar contents so that an accurate comparison could be made between images. As we were looking at both the visible and near-infrared (VNIR) spectrum ($450 - 1.00 \times 10^3$ nm) and short wave infrared (SWIR) spectrum ($0.95 - 2.50$ μ m), four dates were selected (4th, 7th, 12th and 15th August) as they had images for both wavelength regions at similar locations. A single region of interest (ROI) was then selected, that was present in all eight of these images. This same region in each image was cropped, so that a quick classification could be done and so that all the images had fairly similar objects contained within them. All the cropped images are shown in Fig. 5.3 and 5.4.

Name of object	4th Aug	7th Aug	12th Aug	15th Aug
Green Perspex	No	No	Yes	Yes
Orange Perspex	No	Yes	Yes	Yes
White Perspex	Yes	Yes	Yes	Yes
Grey Ceramic	Yes	Yes	Yes	Yes
Green Ceramic	Yes	Yes	Yes	Yes
Beige Carpet Tile	Yes	Yes	Yes	Yes
Green Carpet Tile	Yes	Yes	Yes	Yes

Table 5.1: Presence of objects in images at the 4 dates

Fig. 5.3: RGB image and ground truth for each selected ROI for VNIR imagery (450 - 1.00×10^3 nm)Fig. 5.4: False colour image and ground truth for each selected ROI for SWIR imagery (0.95 - $2.50 \mu\text{m}$)

Having completed this, the next stage was to select several objects that were common to most of the images. Table 5.1 shows the 7 objects selected and whether each was located in the images for all of the 4 dates selected previously. Also provided with the data was a database of lab-measured spectra for all of these objects, so a comparison could be made before and after any atmospheric effects. The GPS coordinates for all 7 objects on each day was also provided, so from this by manually selecting each target an accurate ground truth could be made for each of the images.

5.3 Atmospheric Correction

In this section two existing scene-based approaches to atmospheric correction are discussed, and a radiative transfer method using the atmospheric modelling software MODTRAN and a method proposed by extracting known spectra from the scene to atmospherically correct the images are also discussed.

5.3.1 Scene-Based Approach

Internal Average Relative Reflectance

Unlike several other approaches, such as Flat Field and Empirical Line correction, the Internal Average Relative Reflectance (IARR) approach [154] allows for the atmospheric calibration of radiance data into reflectance data. This method is useful for converting radiance hyperspectral data into relative reflectance when little is known about the scene or no ground measurements exist. This is done by using the average pixel spectrum (APS) of the hyperspectral data cube. The spectrum for each pixel is then divided by this APS to produce a relative reflectance spectrum.

Dark Object Subtraction

Assuming that there is at least one dark object in the scene, such as water or tarmac, it can be assumed that no light would be reflected from them. Any light measured by the sensor must be caused by atmospheric scattering. In this case, it would then be possible to remove these values from the overall image to produce an atmospherically corrected image [63]. This was done by first calculating the length of each pixel vector (see Equation 5.1):

$$pixel\ vector\ length = \sqrt{\vec{x}^T \times \vec{x}} \quad (5.1)$$

where x is the pixel vector

The vector with the minimum length was then selected and subtracted from all the pixel vectors in the original image, and any negative values were set to zero.

Known Spectra Grass and Tarmac (KS Grass and KS Tarmac)

Rather than trying to predict the atmospheric transform from a radiative transfer based approach, we use the fact that both grass and tarmac are present in all the images used for this work. This method was to use known spectra for both grass and tarmac to calculate the atmospheric transform. The pixels containing either of these materials were then classified, and an average pixel spectrum was taken. Fig. 5.5 shows the standard atmospheric correction process: if we know both the raw and lab spectra, then the atmospheric transform can be calculated. This atmospheric transform can then be used to atmospherically correct the hyperspectral data.

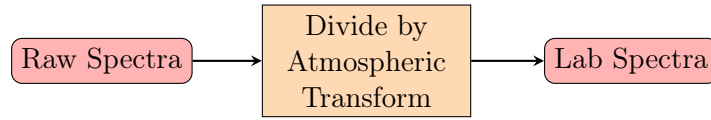


Fig. 5.5: Atmospheric correction process

5.3.2 Radiative Transfer Approach

The radiances measured by a hyperspectral camera are a combination of the atmospheric path and the surface-reflected solar radiances (see Fig. 5.2). These radiance images can be converted to apparent reflectance using Equation 5.2 [136]:

$$\rho_{obs}(\lambda, \theta, \theta_0, \phi_0) = \frac{\pi L_{obs}(\lambda, \theta, \theta_0, \phi_0)}{\mu_0 F_0(\lambda)} \quad (5.2)$$

where,

θ_0 is the solar zenith angle,

ϕ_0 is the solar azimuth angle,

θ is the sensor zenith angle,

ρ is the sensor azimuth angle,

λ is the wavelength,

L_{obs} is the radiance measured at the satellite,

F_0 is the solar flux at the top of the atmosphere and,

μ_0 is the cosine of solar zenith angle.

According to Tanre et al. [155], when the surface is assumed to be Lambertian and the adjacency effect is neglected, $\rho_{obs}(\lambda, \theta, \theta_0, \phi_0)$ can be expressed approximately as:

$$\rho_{obs}(\lambda, \theta, \theta_0, \phi_0) = [\rho_{atm}(\lambda, \theta, \theta_0, \phi_0) + \frac{t_d(\lambda, \theta_0)t_u(\lambda, \theta)\rho(\lambda)}{(1 - s(\lambda)\rho(\lambda))}]T_g(\lambda, \theta, \theta_0) \quad (5.3)$$

where,

ρ_{atm} is the path reflectance,

t_d is the downward scattering transmittance,

t_u is the upward scattering transmittance,

s is the spherical albedo of the atmosphere,

ρ is the surface reflectance and,

T_g is the total gaseous transmittance in the Sun-surface-sensor path.

Using the atmospheric modelling software MODTRAN, estimations can be made for all 15 outputs shown in Table 5.2. By using these MODTRAN outputs Equation 5.2 can then be altered to be expressed entirely using these outputs (see Equation 5.4). The complete derivation of this equation is shown in Appendix F.

$$\rho = \frac{L_{obs} - SOL_SCAT}{TOTAL_RAD} \quad (5.4)$$

where

ρ is the surface reflectance,

L_{obs} is the radiance measured at the camera,

SOL_SCAT is the solar multiple-scattered radiance term of the path radiance, and

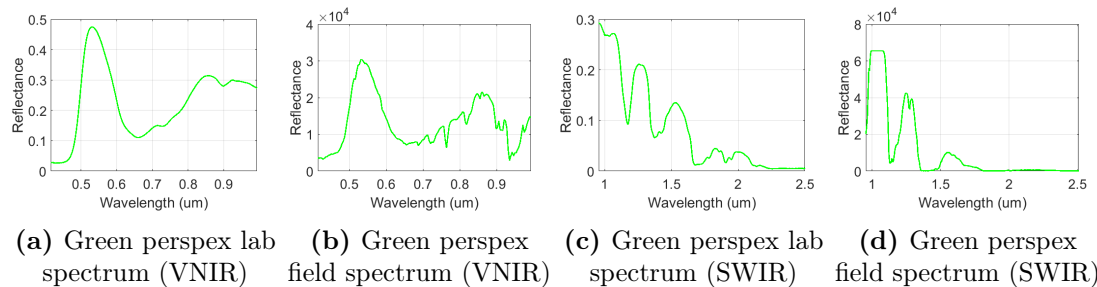
$TOTAL_RAD$ is the total radiance reaching the sensor.

.fl7 Name	Name	Units
FREQ	Wavenumber	1/cm
TOT_TRANS	Transmission Total	N/A
PTH_THRML	Path Thermal Radiance	W/(m ² str nm)
THRML_SCT	Thermal Scatter	W/(m ² str nm)
SURF_EMIS	Surface Emission Radiance	W/(m ² str nm)
SOL_SCAT	Solar Scatter Radiance	W/(m ² str nm)
SING_SCAT	Single Scatter Radiance	W/(m ² str nm)
GRND_RFLT	Total Ground Reflected Radiance	W/(m ² str nm)
DRCT_RFLT	Direct Ground Reflected Radiance	W/(m ² str nm)
TOTAL_RAD	Total Radiance	W/(m ² str nm)
REF_SOL	Reflected Solar	W/(m ² str nm)
SOL@OBS	Solar at Observer	W/(m ² str nm)
DEPTH	Optical Depth	W/(m ² str nm)
DIR_EM	Emissivity	W/(m ² str nm)
TOA_SUN	Top of Atmosphere (TOA) Sun	W/(m ² str nm)
BBODY_T[K]	Blackbody Temperature	K

Table 5.2: Description of output file values

5.4 Atmospheric Effects

To better understand the effects of the atmosphere on hyperspectral images, the spectra for each of the objects discussed earlier was collated and plotted against the original lab-measured spectra (see Fig. 5.6). It can be seen that the general shape and majority of the spectral features in the VNIR data remain when compared with the lab spectra, except for the absorption bands at 760, 820 and 930 nm, and all the objects appear still to be spectrally different from each other. However, for the SWIR data, while the initial lab spectra are all spectrally different there are far more significant atmospheric effects in the SWIR data. This means that the resulting field spectra have very little of the distinct spectral features remaining and all appear to be very similar.



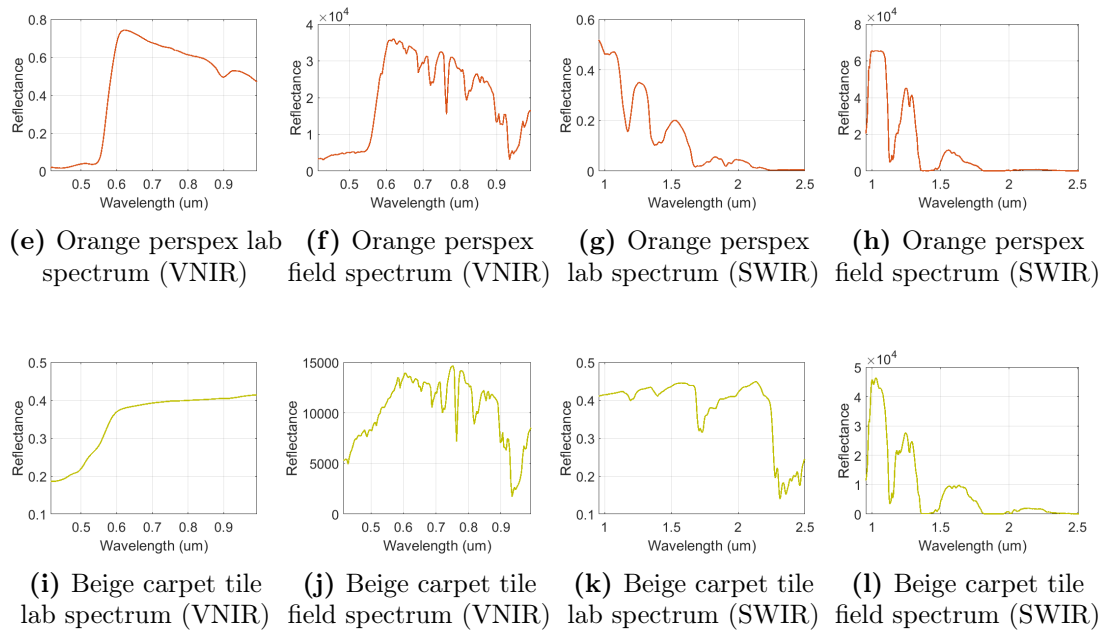


Fig. 5.6: Comparison between lab-based spectra and spectra measured in field for both VNIR and SWIR hyperspectral imaging

The next stage was to examine if our initial observations were correct concerning how spectrally different each object was in the VNIR and SWIR datasets. This was done by using the ground truth created previously to collect all the spectra available for each object in each of the images. The average spectrum was then taken for each object, and all of these spectra were plotted on the same graph. See Fig. 5.7 and 5.8.

5.4.1 Visible and Near-infrared (VNIR)

For all four VNIR datasets acquired there are some atmospheric effects present. The majority of these effects are due to the water absorptions bands (see Fig. 5.7). However, from looking at the individual spectra, each object is sufficiently different that it can reasonably be assumed that the original spectra can be recovered, e.g. the green ceramic and perspex having a peak 530 nm, the white perspex having the highest radiance and the orange perspex having a peak 650 nm. The only slight matter for concern is in the 7th August dataset, as the image was captured from an altitude of 2 km, and was also taken on a cloudy day, whereas the images on the other 3 dates were taken from 1 km. Therefore the maximum radiance measured by the camera on the 7th August is much

lower, meaning that the spectra are not as spectrally different, so this will provide a much harder atmospheric correction problem.

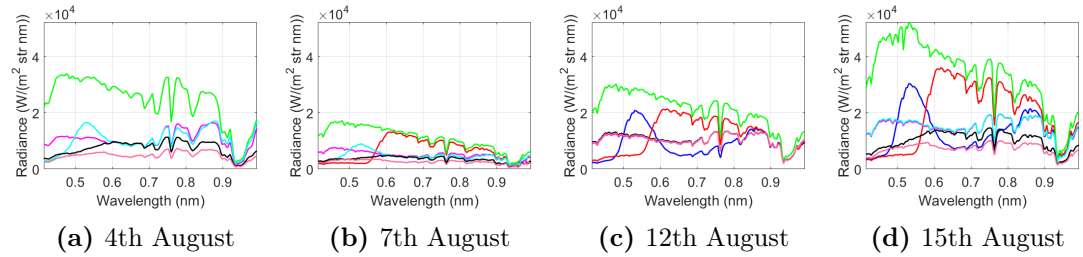


Fig. 5.7: Spectral profile of each material on the same days for VNIR images. Green Perspex (Blue), Orange Perspex (Red), White Perspex (Green), Grey Ceramic (Purple), Green Ceramic (Cyan), Beige Carpet Tile (Black), Green Carpet Tile (Pink)

5.4.2 Short Wave Infrared (SWIR)

Due to the significant amount and width of the water absorption bands in the SWIR region of the electromagnetic spectrum for all of the SWIR data acquired, there is a considerable amount of atmospheric attenuation. This means that there are very little distinct features in any of the spectra for the objects under observation (see Fig. 5.8). Due to this, it will make it almost impossible to recover the original spectra in the SWIR imagery.

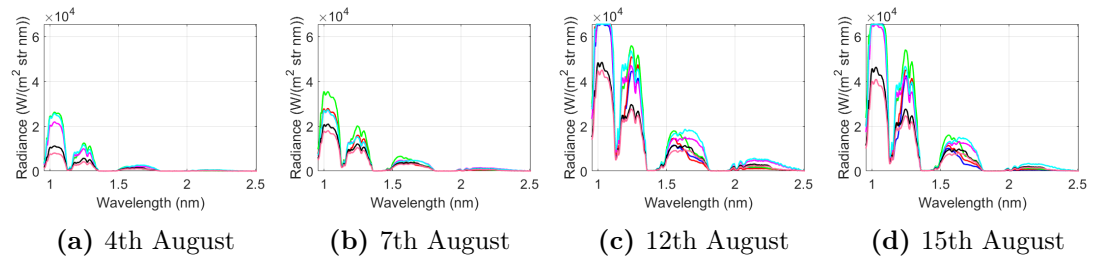


Fig. 5.8: Spectral profile of each material on the same days for SWIR images. Green Perspex (Blue), Orange Perspex (Red), White Perspex (Green), Grey Ceramic (Purple), Green Ceramic (Cyan), Beige Carpet Tile (Black), Green Carpet Tile (Pink)

Date	Time	Lat.	Long.	Altitude	Temp.
4th August	13:23	51N	-1.6E	988-1005m	17.67
7th August	12:29	51N	-1.6E	2116-2118m	19.57
12th August	10:55	51N	-1.6E	1203-1215m	16.14
15th August	13:17	51N	-1.6E	1191-1197m	15.27

Table 5.3: Important parameters for each VNIR image

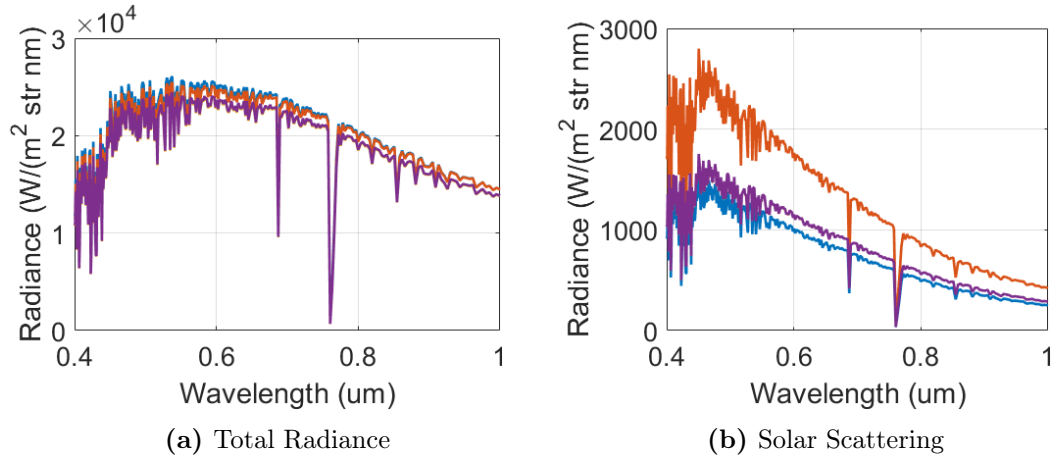


Fig. 5.9: MODTRAN predictions

5.5 Analysis

5.5.1 MODTRAN Atmospheric Modelling

Having analysed the atmospheric effects in the available data, it was decided that the main focus should be on the VNIR imagery, as that is the dataset which is more realistic to atmospherically correct due to the remaining spectrally distinct features. To do this, several key parameters pertaining to each image were acquired (Table 5.3). Then, by using the atmospheric modelling software MODTRAN [73], the total radiance and solar scattering contribution can be modelled for each image (Fig. 5.9). From these two plots, it can be seen that as all the dates are relatively close to each other, and are taken from similar heights (1-2km) the total radiance is almost identical for all the images. However, even an increase of 1km altitude for the 7th August dataset has a significant effect on the solar scattering present in the image. Having produced these two curves using the process detailed earlier, the data can be atmospherically corrected.

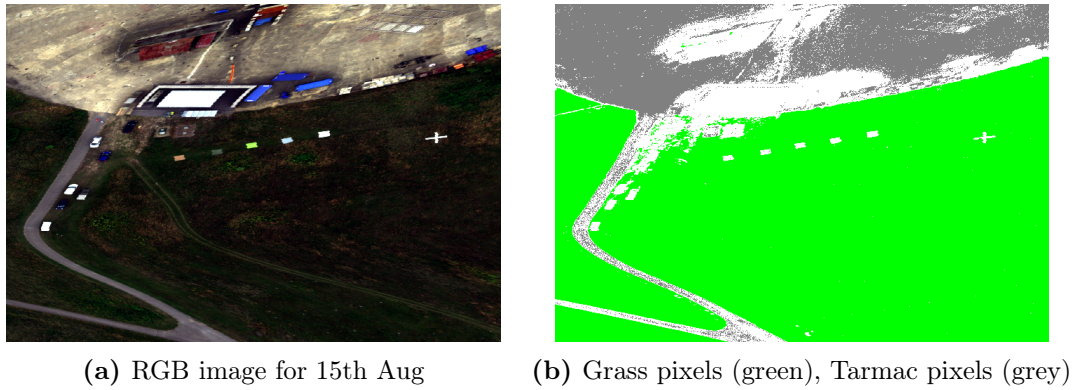
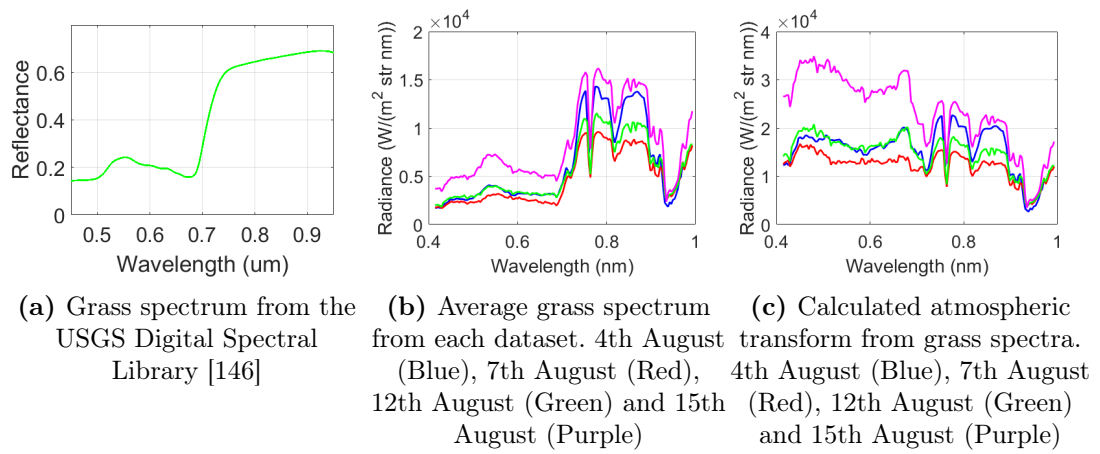


Fig. 5.10: Material classification for 15th August VNIR image



5.5.2 Predict Atmospheric Transform from Known Spectra

The final technique proposed is a very quick method for atmospheric correction. As discussed earlier the initial step was to classify the majority of either the tarmac or grass pixels in the image. This was done by first manually selecting a few reference pixels in all of the VNIR images for both materials. By using the Spectral Angle Mapper (SAM) [47] it was then possible to classify all the spectrally similar pixels for each image (see Fig. 5.10).

From this, an average spectrum for each of these materials could be produced for all 4 of the images (see Fig. 5.11b and 5.11e). Then by retrieving a lab-measured spectrum from the USGS Spectral Database [146] (see Fig. 5.11a and 5.11d) the atmospheric transform can be calculated, as we have both the raw and lab spectra for both materials (see Fig. 5.11c and 5.11f).

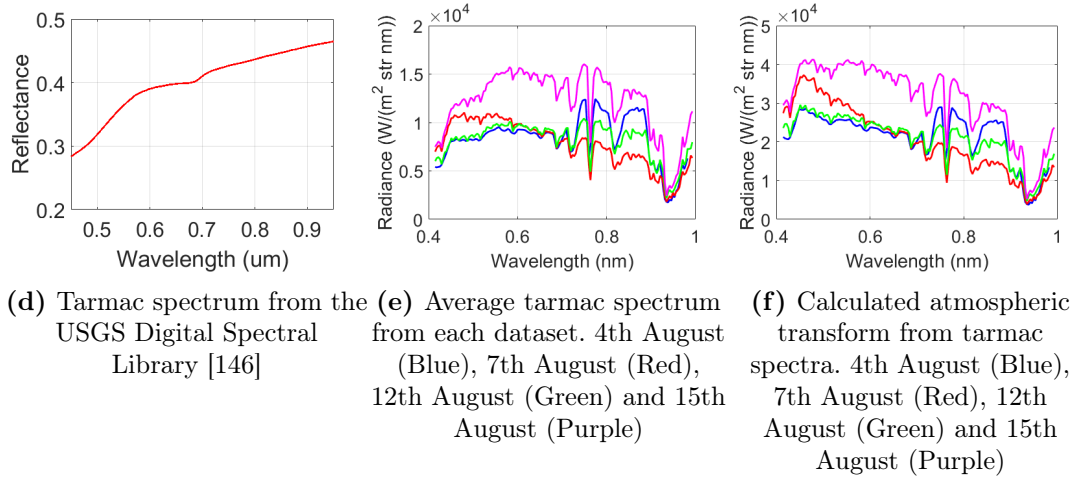


Fig. 5.11: Atmospheric transform calculation steps for VNIR imagery

5.6 Results

In order to determine the effectiveness of the various atmospheric correction techniques described previously, a well established spectral matching algorithm was selected to try to classify the atmospherically corrected image using the lab-measured spectra. The more accurate the atmospheric correction, the better the spectral matching and classification of the image are expected to be.

5.6.1 Spectral Angle Mapper (SAM)

The Spectral Angle Mapper (SAM) [98] is a very commonly used analysis tool for spectral matching in remote sensing hyperspectral data. It works by measuring the spectral similarity between a reference and a target spectrum. This similarity or angle is calculated in Equation 5.5:

$$SAM = \cos^{-1}\left(\frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \cdot \|\vec{b}\|}\right) \quad (5.5)$$

where

\vec{a} is the reference spectrum,

\vec{b} is the target spectrum, and

$\|\cdot\|$ is the Euclidean norm.

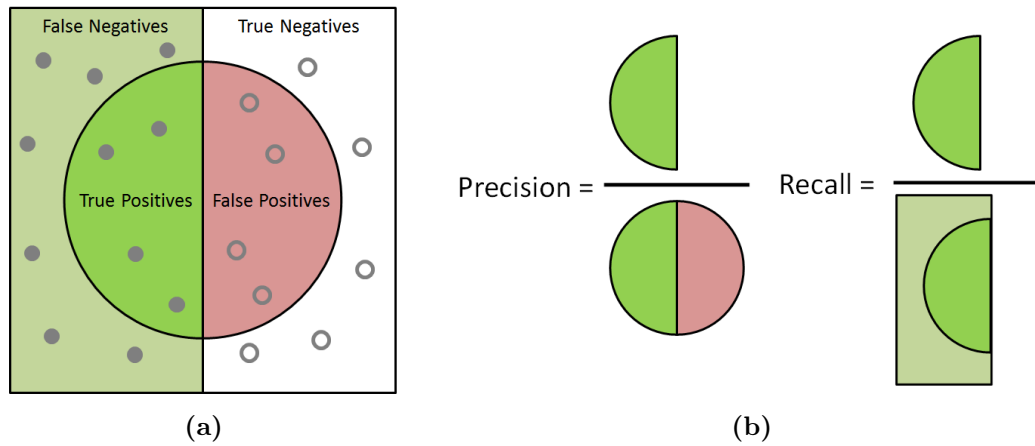


Fig. 5.12: Diagram for precision and recall for true and false negatives and positives

Small angles produced by the SAM represent more similar spectra, whereas higher angles represent lower similarity. The main benefit of using this algorithm is that it is insensitive to changes in illumination, due to it calculating similarity based on vector direction rather than vector length.

5.6.2 Comparison of Techniques

As several algorithms have been created it was necessary to develop a way in which to score the performance of each of these techniques compared to the ground truth. This was done by considering various metrics for evaluating classification scores [156]. Two metrics were chosen, the first being the recall (see Fig. 5.12 and Equation 5.6), and the second the percentage of targets found:

$$Recall = \frac{TP}{TP + FN} \quad (5.6)$$

where,

TP is the number of true positives found, and

FN is the number of false negatives.

Having atmospherically corrected all of the images using the techniques described, we then attempted to classify each image using the SAM. The metrics described above

to score each of the classifications were used and the results are shown in Tables 5.4 to 5.7. In all the tables the technique that produces the best result for each metric on each image is highlighted in green, as well as the overall best technique for that metric. Conversely, the technique with the worst results for each metric is highlighted in red. The results show that overall for both the Recall and Percentage of Targets Found for the VNIR imagery the MODTRAN technique produces the best results. Overall KS Tarmac produces the worst results for both the Recall and the Percentage of Targets Found, this is thought to be due to relatively low spectral reflectance from the tarmac in the VNIR region of the spectrum that is distorting the results. For the SWIR imagery, it is clear that both the KS Tarmac and KS Grass techniques produce the best results for both Recall and Percentage of Targets Found, with the KS Tarmac only producing slightly better results than the KS Grass technique.

Technique	Recall				
	4th	7th	12th	15th	Mean
Lab Spectra	0.124	0.196	0.245	0.272	0.209
IARR	0.069	0.180	0.181	0.222	0.163
DOS	0.271	0.211	0.259	0.262	0.251
MODTRAN	0.382	0.391	0.357	0.527	0.414
KS Grass	0.131	0.192	0.333	0.430	0.272
KS Tarmac	0.067	0.178	0.192	0.117	0.139

Table 5.4: Recall results for VNIR imagery

Technique	Percentage of Targets Found				
	4th	7th	12th	15th	Mean
Lab Spectra	40%	33%	57%	71%	50%
IARR	20%	33%	43%	43%	35%
DOS	60%	50%	57%	71%	60%
MODTRAN	60%	67%	57%	71%	64%
KS Grass	40%	33%	71%	86%	58%
KS Tarmac	20%	33%	43%	43%	35%

Table 5.5: Percentage of targets found for VNIR imagery

Technique	Recall				
	4th	7th	12th	15th	Mean
Lab Spectra	0.000	0.123	0.086	0.078	0.072
IARR	0.199	0.083	0.069	0.128	0.120
DOS	0.000	0.114	0.057	0.078	0.062
MODTRAN	0.000	0.021	0.086	0.091	0.050
KS Grass	0.321	0.132	0.143	0.195	0.198
KS Tarmac	0.324	0.222	0.237	0.156	0.235

Table 5.6: Recall results for SWIR imagery

Technique	Percentage of Targets Found				
	4th	7th	12th	15th	Mean
Lab Spectra	0%	17%	14%	14%	11%
IARR	60%	17%	29%	29%	33%
DOS	0%	17%	14%	14%	11%
MODTRAN	0%	17%	14%	14%	11%
KS Grass	60%	33%	29%	43%	41%
KS Tarmac	80%	33%	29%	29%	43%

Table 5.7: Percentage of targets found for SWIR imagery

5.7 Summary

This chapter has introduced three new techniques for atmospheric correction of aerial hyperspectral imagery. Two scene-based approaches using prior known spectra, KS Grass and KS Tarmac, and one radiative transfer based approach using MODTRAN radiative transfer models were compared with two well established scene-based approaches (Internal Average Relative Reflectance and Darks Object Subtraction). All three proposed techniques showed very good performance. Having compared the results of these techniques with the currently used scene-based approaches, it is clear that for the VNIR imagery the proposed radiative transfer technique has higher recall than all the other techniques for each of the four VNIR images, and in terms of percentage of targets found it performs as well as or better than the current techniques, however the KS Grass technique does provide better results for the 12th and 15th August images. This is believed to be due to the variation in both of these images caused by the cloud coverage that cannot be accurately modelled in MODTRAN.

However, as predicted, the SWIR imagery did provide a much greater challenge than the VNIR imagery, due to the much greater atmospheric effects present in the SWIR images. For both techniques, the MODTRAN approach provides no improvement in terms of recall and percentage of targets found, and in fact, produces worse results for the recall for certain images. Nonetheless, using both the KS Tarmac and KS Grass techniques the results for both performance metrics for all four SWIR images are an improvement over all the current techniques tested.

Chapter 6

Erosion Detection in Wind Turbines Using Hyperspectral Imaging

6.1 Introduction

Inspection of wind turbine blades is required to identify any defects or failures and decide on any remedial actions e.g. blade repair or replacement. Traditionally, inspections have been performed by rope-access technicians who visually inspect the blades and record damage using standard photographic equipment [157] and [158]. Recent developments have seen an increase in popularity in the use of remotely based inspection techniques using ground-mounted cameras [159] and cameras installed on Remotely Operated Aerial Vehicles (ROAVs), more commonly referred to as drones [160].

Whilst these techniques remove the need for human access to the blades, imaging is performed remotely and does not always provide adequate image quality using standard high definition cameras. As a result, there is a growing interest in imaging techniques based on other regions of the electromagnetic spectrum [161–166]. Laboratory and field-based trials are required to properly examine this potential and understand which frequencies can be applied to imaging blades.

This chapter demonstrates a hyperspectral imaging technique in its application to imaging surface defects on a section of wind turbine blade at varying light levels in a laboratory. Processing of the data is then performed to determine the optimal bands

required to reveal each type of erosion. By using a polynomial surface subtraction, the results are improved upon and then compared with conventional images taken with a high-definition RGB camera.

The rest of this chapter is structured as follows. Section 6.2 introduces the wind turbine blades, their composition and the types of damages that can occur during operation. Section 6.3 presents the methodology for capturing the initial images, adding realistic defects to the blade, further imaging, and then finally the alteration of illumination levels. Results and a summary of the contributions are then presented in Sections 6.4 and 6.5 respectively.

6.2 Wind Turbine Blades

This section describes various possible types of damage to offshore wind turbine blades. Fig. 6.1 illustrates the main elements of a typical wind turbine blade design. The structural design is optimised by adopting a shell structure with a long central main spar. The spar caps provide stiffness and strength in bending and extension, while the spar webs provide shear stiffness. The upwind side and downwind side of the blade are manufactured, then joined together at both the leading edge and the trailing edge using adhesive glue.

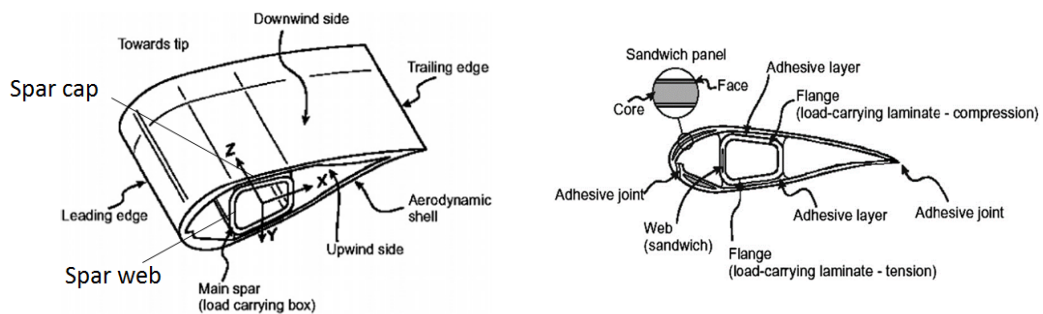


Fig. 6.1: Main elements of a wind turbine blade [167]

The materials of blades must be strong and stiff, yet as light as possible to satisfy the blade design criteria and to minimise both the weight-induced fatigue loads and the loads on the tower and foundations [168]. The materials of contemporary blades

are usually fibre-reinforced composites which provide low weight, high strength and stiffness and optimal performance in fatigue. The majority of blades are made of glass fibre/epoxy, glass fibre/polyester or carbon fibre/epoxy composites [169].

Blades are generally designed to last for a minimum of 20 years, during which time they will be subjected to varying weather patterns and wind loads that become more extreme in the hostile marine environment. Thus, they should be designed to withstand different types of damage, e.g. fatigue damage, erosion and damage due to extreme conditions [170].

Damage can occur in a number of ways. It is important to properly identify different damage types so that appropriate remedial actions can be performed. Fig. 6.2 displays sketches and images of the typical damage types found on the removed sections of the blade. Table 6.1 provides descriptions of these damages types.

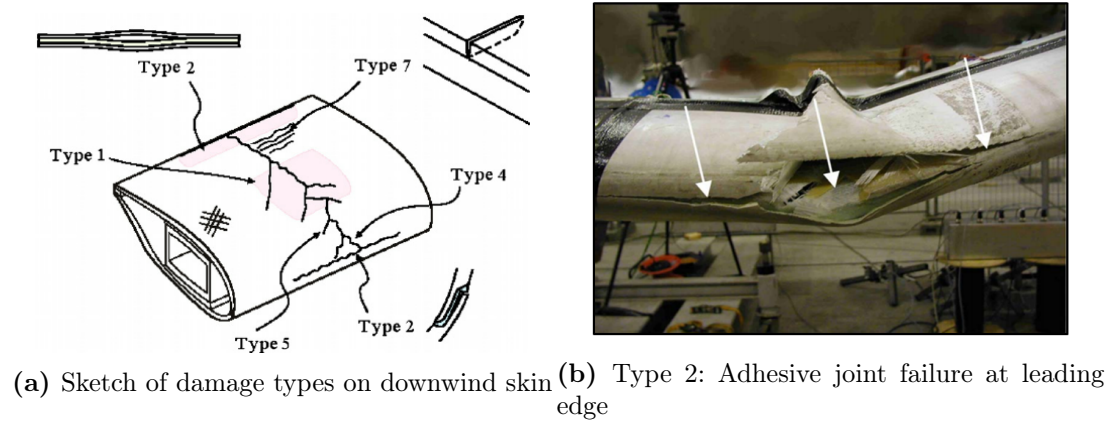


Fig. 6.2: Examples of damage types identified on a blade structure tested to failure by a large compressive load [171]

Leading edge erosion is one of the biggest issues during the operations and maintenance phase of wind farms, particularly offshore. It occurs as a result of different interactions between the blades rotating at high speeds and the environment, e.g. icing, strong winds and impact with rain, hailstones, salt spray, dust and other atmospheric contaminants [172]. It consists of peeling and cracking of the final blade surface along its leading edge, and can lead to pitting, gouging and de-lamination. Damage to the leading edge during manufacturing, transportation and handling can accelerate the de-

Type	Description
Type 1	Damage formation and growth in the adhesive layer joining skin and main spar flanges
Type 2	Damage formation and growth in the adhesive layer joining the up and downward skins along leading and/ or trailing edges (adhesive joint failure between skins)
Type 3	Damage formation and growth at the interface between face and core in sandwich panels in skins and main spar web (sandwich panel face/core debonding)
Type 4	Internal damage formation and growth in laminates in skin and/or main spar flanges, under a tensile or compression load (delamination driven by a tensional or buckling load)
Type 5	Splitting and fracture of separate fibres in laminates of the skin and main spar (fibre failure in tension; laminate failure in compression).
Type 6	Buckling of the skin due to damage formation and growth in the bond between skin and main spar under compressive load (a specific case of type 1)
Type 7	Formation and growth of cracks in the gel-coat; debonding of gel coat from the skin
Type 8	Blade material surface degradation in the first laminate layer of the leading edge (driven by erosion)

Table 6.1: Typical damage to load-carrying structural members of a wind turbine blade [171]

velopment of erosion during operation. Fig. 6.3 shows three blade inspection images of leading edge erosion on an offshore wind turbine blade.

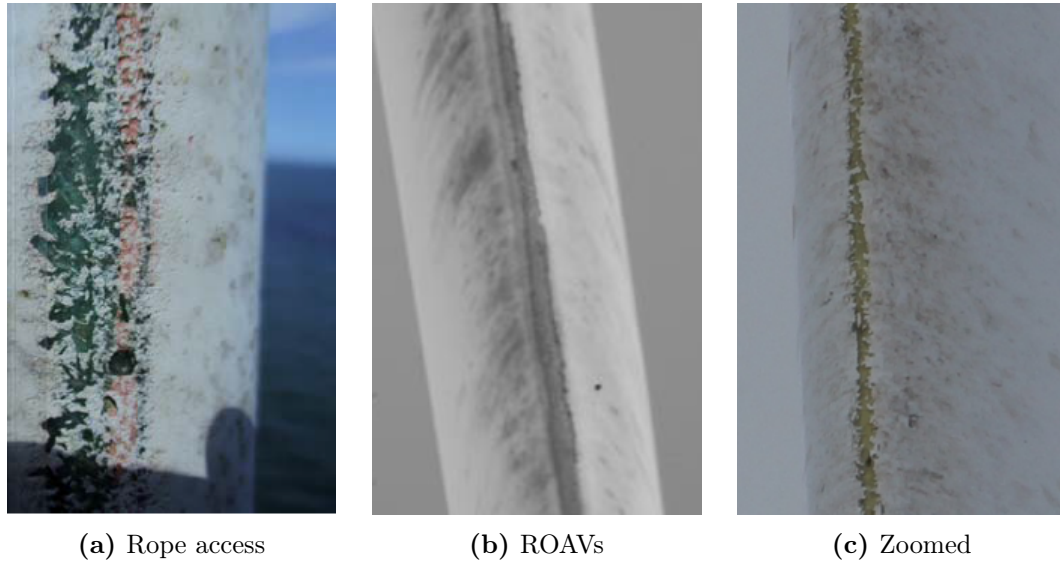


Fig. 6.3: Images taken with HD digital camera using different methods (Source: Offshore Renewable Energy Catapult, 2015)

6.3 Methodology

6.3.1 Image Acquisition

The hyperspectral camera selected to be used for this project was the Innospec Red Eye 1.7 (see Fig. 6.4a), this is a near-infrared system operating in the spectral range of 950 - 1.70×10^3 nm and outputs 256 bands over this range. This was chosen over the VIS system as it provided the best initial results.

Having selected the camera the next stage of the process was to set up the hyperspectral camera as follows; see Fig. 6.4. In the lab, it is possible to image the blade at a distance that would not be safe to operate at in practice without risking damage to the system or turbine blade. Therefore a distance of 1.5 metres was selected both to provide adequate resolution of the blade from the camera but also as a feasible distance at which to operate at in the field.

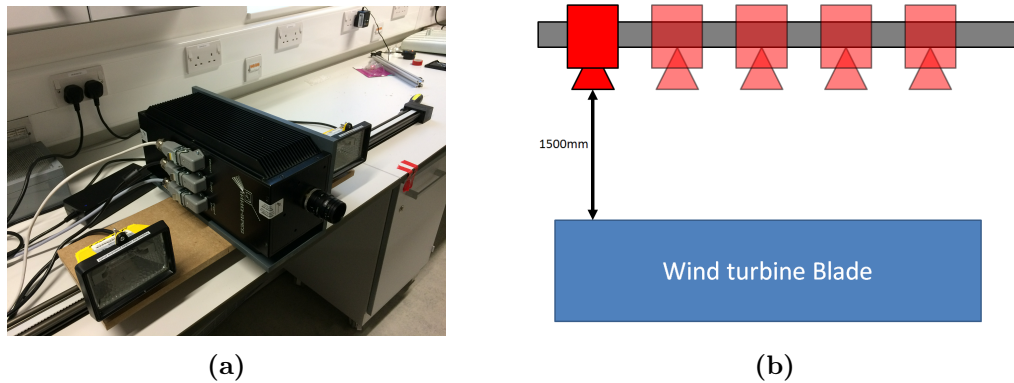


Fig. 6.4: Hyperspectral camera set-up

6.3.2 Initial Imaging

The blade and camera were positioned and the speed was determined by using a calibration checker-board. As the camera is a line scan device, it is necessary to move the camera at a constant speed over the full length of the blade to provide square pixels in the image. Prior to hyperspectral imaging, some images were taken with a standard high-definition RGB camera to provide details of any defects already present within the blade; see Fig. 6.5.

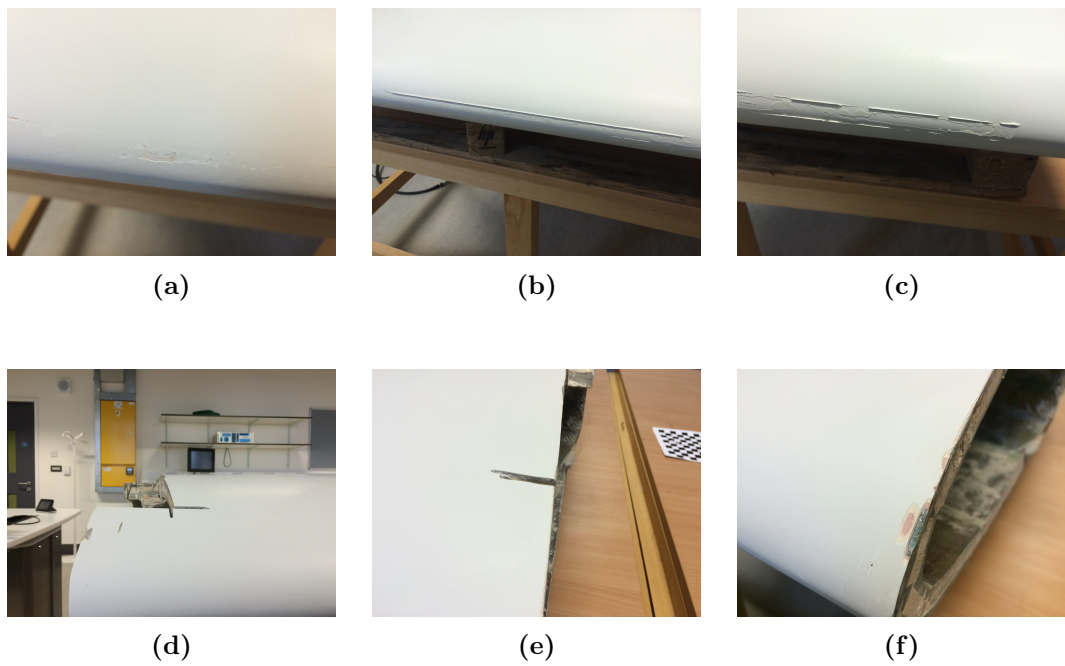


Fig. 6.5: RGB images of initial defects

Following this, the first image was taken using the hyperspectral camera; see Fig. 6.6. For each of the defects already present and highlighted in Fig. 6.5, the location in this new hyperspectral image has been highlighted. For this example, an average of all the bands in the hyperspectral image was used to display this greyscale image.

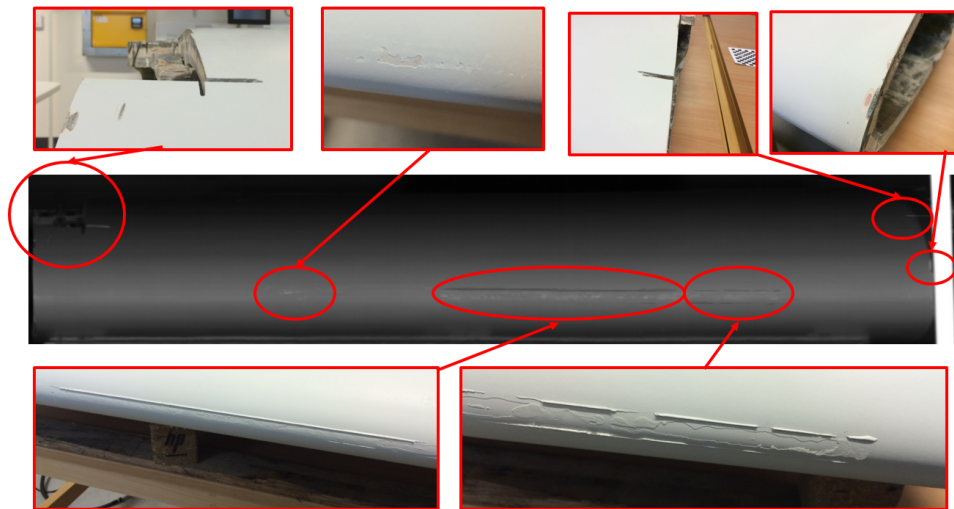


Fig. 6.6: Initial hyperspectral image with existing defects highlighted

6.3.3 Introduction of Damage Types

Having done the above analysis of the existing defects present on the blade, some realistic defects were added to the blade. Those selected were of the type that would be present in a wind turbine operating in the real world over several years [172]; see Fig. 6.7. Each of these damage types provide a different challenge for detection. Fig. 6.7a shows very light erosion of the outer laminate of the blade. Fig. 6.7b and 6.7c show erosion down to the second and third layer of laminate, the pink and grey laminate respectively. Finally, Fig. 6.7d shows a realistic type of damage where the turbine has been hit by an object and the damage has penetrated down through all layers of laminate to the fibreglass.



(a) Light Erosion



(b) Erosion to pink laminate



(c) Erosion to grey laminate



(d) Realistic damage type

Fig. 6.7: Damage types

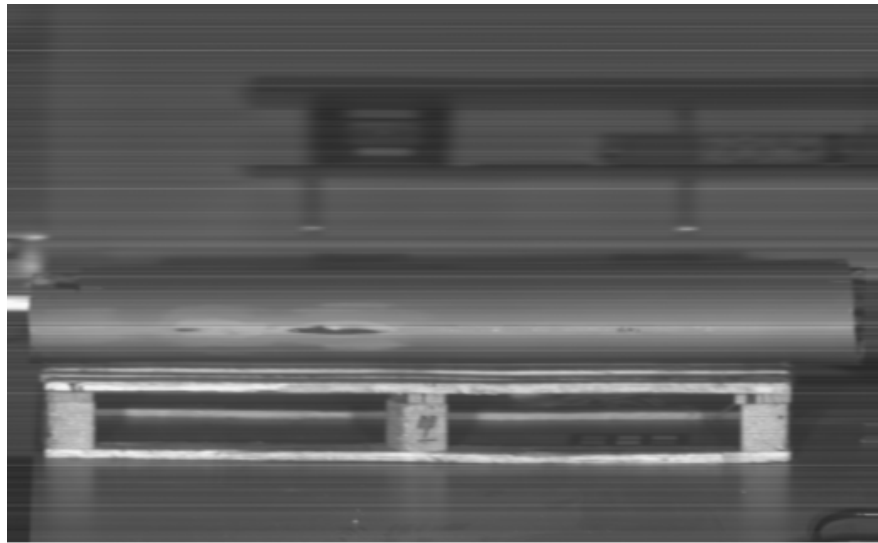


Fig. 6.8: Hyperspectral image of turbine blade after defects have been added

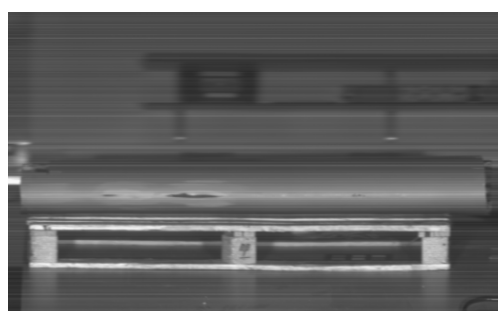
6.3.4 Further Imaging

Having introduced these new damage types to the blade, it was then necessary to image the blade again; see Fig. 6.8. While the majority of the new damage is visible to the naked eye in this image, it was necessary to see at what point the damage was no longer visible.

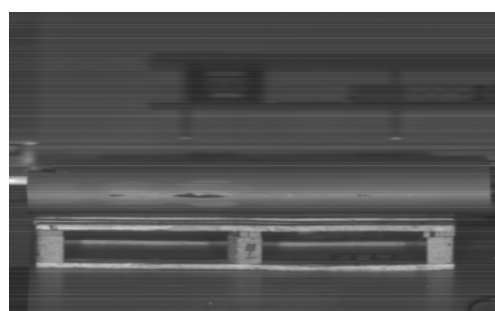
As the images were all taken at a fixed distance on a stationary blade, this provided the ideal conditions for defect detection. To ensure that this technique could be used in all conditions, it was then explored how to create more realistic images. As the distance was fixed, the only other variable was the illumination. As we were using artificial light, via two halogen lamps, the illumination was reduced by dimming each of the lamps; see Fig. 6.9. Then, by using a Lux meter, the illumination was measured at each step and a new hyperspectral image was taken at each level of illumination; see Fig. 6.10.



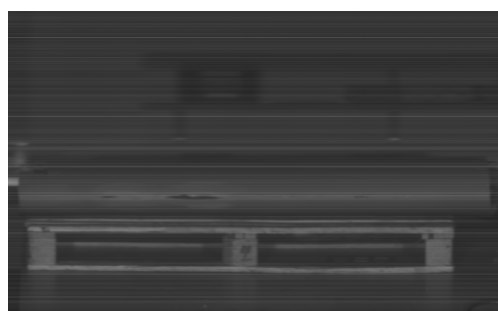
Fig. 6.9: Hyperspectral images at different illumination levels



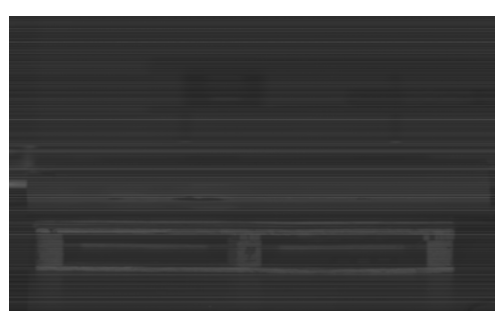
(a) 1000 Lux



(b) 500 Lux



(c) 100 Lux



(d) 25 Lux

Fig. 6.10: Hyperspectral images at different illumination levels

6.4 Results and Analysis

6.4.1 Band Selection

As the hyperspectral camera outputs 256 bands between 950 nm and 1.70×10^3 nm before any further processing is performed, it was necessary to determine which were the optimal bands to use for the analysis. This was done by manually selecting each region of eroded surface along, with a non-eroded surface. The difference between each eroded and non-eroded spectrum was calculated, an average was taken and the results are shown in Fig. 6.11. It was found that the bands between 1.15×10^3 nm and 1.23×10^3 nm were the best bands to use for the analysis in terms of maximising contrast. Having defined a range of optimal bands, for the rest of the analysis it was decided that the 1.20×10^3 nm band would be used.

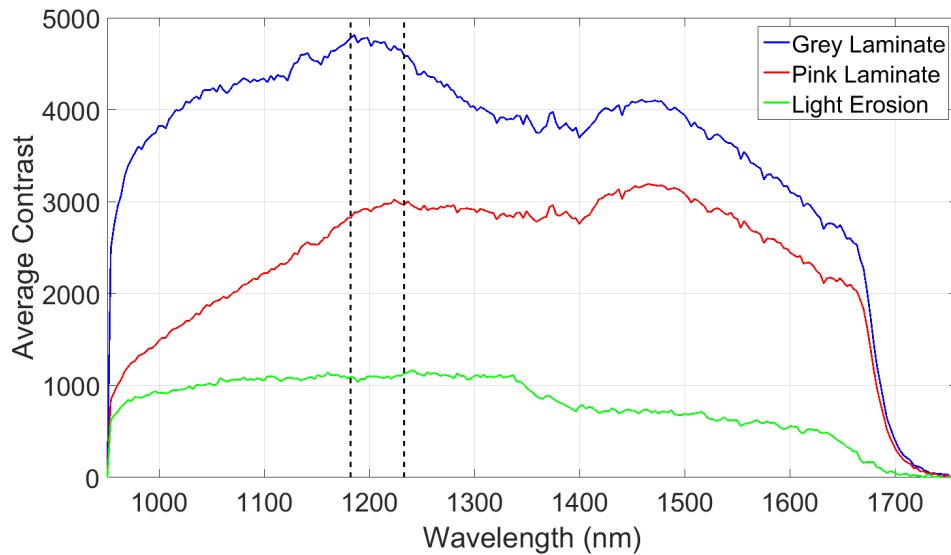
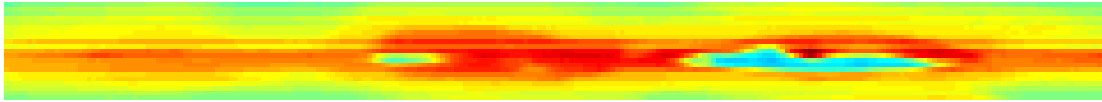


Fig. 6.11: Average contrast between eroded and not eroded surface

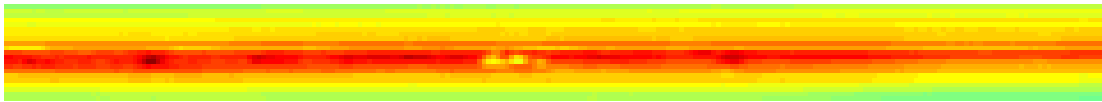
6.4.2 Classification

Having determined an appropriate band it was then possible to create a single image from each hyperspectral image; see Fig. 6.12 to 6.15. This was created by first selecting the leading edge of the blade and then 10 pixels to either side of this edge were selected, and the rest of the data discarded. As the turbine blade is too large to show in one

image, it has been split up into two sections. Section A has damage types simulating erosion down to three different layers of laminate (Type 8), see Table 6.1. Section B has one damage type simulating an object striking the blade and penetrating down to the fibreglass.

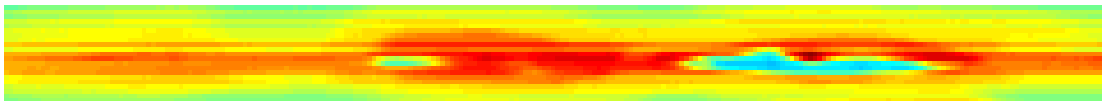


(a) Section A

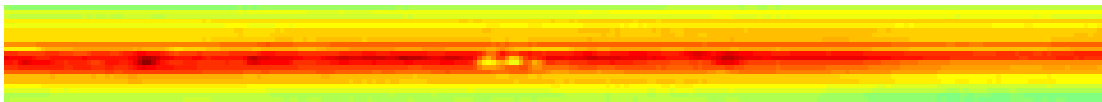


(b) Section B

Fig. 6.12: 1000 Lux

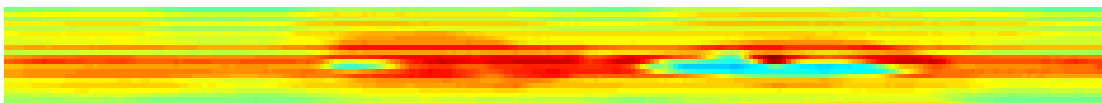


(a) Section A

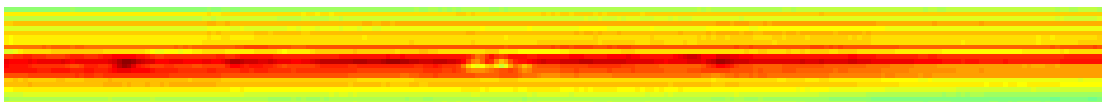


(b) Section B

Fig. 6.13: 500 Lux

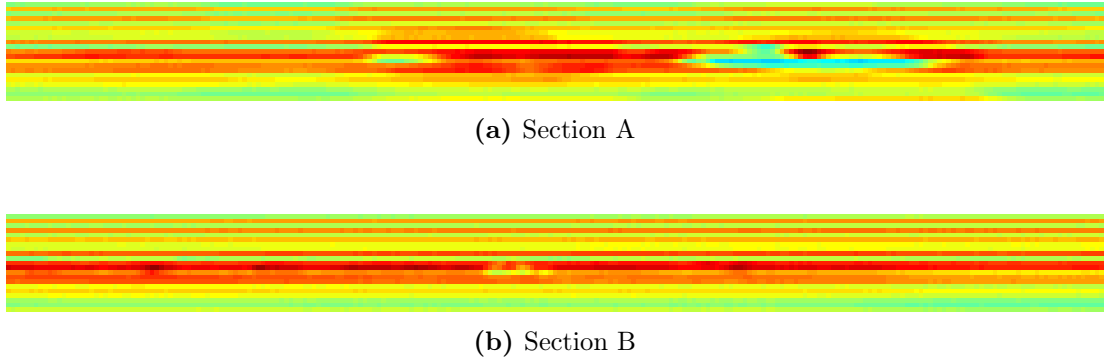


(a) Section A



(b) Section B

Fig. 6.14: 100 Lux

**Fig. 6.15:** 25 Lux

From the 1000 and 500 Lux images (Fig. 6.12 and 6.13) all types of erosion are clearly visible, as the various depths of erosion can be seen from the intensity of the images. However, for the 100 and 25 Lux images (Fig. 6.14 and 6.15) in section A only the erosion down to the pink and grey layer of the laminate are visible and the depth can still be retrieved for these. In section B the damage is almost visible but no other depth information can be extracted.

6.4.3 Image Flattening

To try to improve the contrast of the images, the surface of the wind turbine blade was assumed to have an elliptical geometry. Therefore by using the MATLAB Curve Fitting Toolbox [173] it could be estimated by using a polynomial surface fitting which is defined in Equation 6.1 [174]; see Fig. 6.16. This profile was then subtracted from each hyperspectral image to flatten the image. This increased the contrast between the eroded and non-eroded surface, and the results are shown in Fig. 6.17 to 6.20.

$$\begin{aligned}
 poly55(x, y) = & p_{00} + p_{10}x + p_{01}y + p_{20}x^2 + p_{11}xy \\
 & + p_{02}y^2 + p_{30}x^3 + p_{21}x^2y + p_{12}xy^2 \\
 & + p_{03}y^3 + p_{40}x^4 + p_{31}x^3y + p_{22}x^2y^2 \\
 & + p_{13}xy^3 + p_{04}y^4 + p_{50}x^5 + p_{41}x^4y \\
 & + p_{32}x^3y^2 + p_{23}x^2y^3 + p_{14}xy^4 + p_{05}y^5
 \end{aligned} \tag{6.1}$$

where

x is x-axis image data

y is the y-axis image data and,

p_{ij} are the coefficients of the basis functions for the i th and j th term

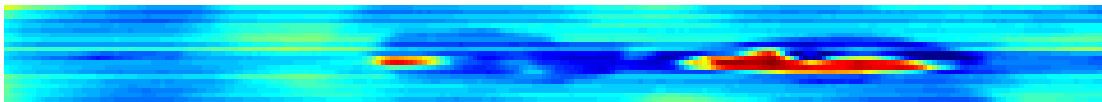


(a) Hyperspectral image of leading edge (1200nm)

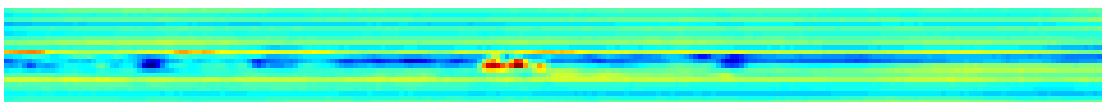


(b) Polynomial surface estimation

Fig. 6.16: Leading edge surface estimation

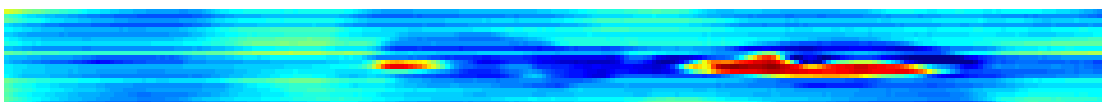


(a) Section A

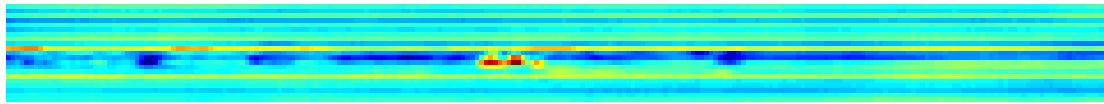


(b) Section B

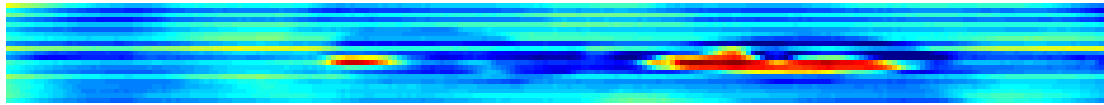
Fig. 6.17: 1000 Lux after surface subtraction



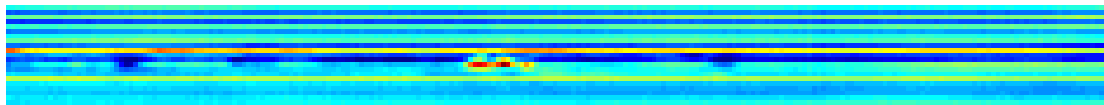
(a) Section A



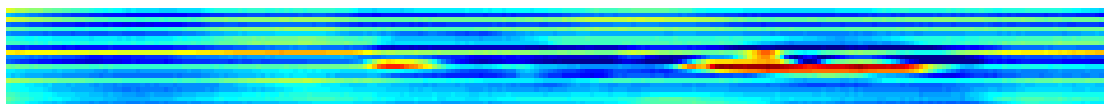
(b) Section B

Fig. 6.18: 500 Lux after surface subtraction

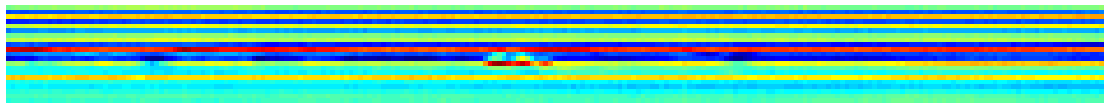
(a) Section A



(b) Section B

Fig. 6.19: 100 Lux after surface subtraction

(a) Section A



(b) Section B

Fig. 6.20: 25 Lux after surface subtraction

Having performed the surface subtraction it is now possible to see all damage types on the blade, and depth information can be retrieved for all except the light erosion in section A.

6.5 Summary

Having compared the results from the hyperspectral camera (see Fig 6.17 to 6.20) to those from the high resolution camera (see Fig. 6.21), it is clear that there is more detail

shown in the hyperspectral images, specifically, the depth of the erosion indicated by the intensity of the image. Inspections of surface damage to the blades of operational wind turbines are mainly focussed on capturing evidence of damage that may require maintenance. This is usually when a defect, such as leading edge erosion, has progressed to a stage where it is clearly visible. Currently, the damage is classified into four main categories (see Table 6.2). Using the standard imaging techniques it is possible to classify erosion into the four categories, therefore for this application hyperspectral imagery may not be necessary. However, there are many other applications for the technology in this field. It should also be noted that the effects of cloud coverage on the NIR illumination could not be simulated in the lab and would need to be a consideration for any field based trials.



(a) 18mm Zoom from 1.5m



(b) 24mm Zoom from 1.5m



(c) 55mm Zoom from 1.5m (Section A)



(d) 55mm Zoom from 1.5m (Section B)

Fig. 6.21: High-resolution images taken with Nikon D5500 camera with various zoom levels

Hyperspectral imaging may have a place in the testing of blades, for example where

it is more imperative to capture early signs of damage to components. This is most applicable to blade erosion testing, which involves rotating a small sample of a blade (around 300mm in length) through a simulated rain field at very high speeds. The test procedure is repeatedly stopped, and the sample is removed and weighed to assess if there has been any material loss and hence erosion. This technique is not effective since the test has to be stopped for long periods of time and the method of assessing the onset of erosion through weight loss is not accurate. It would be useful to have a method for accurately imaging the sample during testing, without having to remove it from the test chamber, to inspect it for early signs of erosion. These early signs may not be clearly visible, however, they are important to capture as they may indicate the end of useful life for a protective material or coating. The approach described provides a means to do this testing.

Class of Erosion	Description
Class 0	No or very little signs of erosion
Class 1	First signs of erosion but no abrasion, top coat and gel coat damages
Class 2	Advanced erosion but no abrasion, cavities, gel-coat damages exposing laminate
Class 3	Severe erosion and gel coat damages exposing laminate and laminate damages

Table 6.2: Erosion classification scheme (Source: Offshore Renewable Energy Catapult, 2015)

Chapter 7

Conclusion and Further Work

7.1 Conclusion

The general objective of this thesis was to improve the existing target detection process. The three steps of this process are data reduction, atmospheric correction and target or anomaly detection. Each of these steps were explored in a different chapter of the thesis, in Chapter 3 the target or anomaly detection stage was looked at, Chapter 4 and 5 looked into the atmospheric correction stage and finally Chapter 6 explored data reduction.

1. In Chapter 3, a new type of anomaly detection technique based on the Percentage Occupancy Hit or Miss Transform [14] was proposed, ways were also explored to improve upon the Mahalanobis Distance [84] for several aerial hyperspectral images. Both of these techniques were then used on two VNIR hyperspectral images, each with different targets present in the images. The results of both of these algorithms were then compared with an established technique for target detection the Sequential Maximum Angle Convex Cone [76].

The POHMT based technique has two key parameters these are the initially selected bands to create the scatter plot and the cell size used to segment the scatter plot. For the first of these key parameters, the band selection, it was discussed in detail various methods that could be used to select these bands. However, for most practical applications where information of the target is unknown, it proved very

difficult to determine a set off optimal bands that would work for every target. What can be achieved with the proposed technique is for applications where there is a limit on data transmission or storage, this could be used to determine a list of potential target locations and discard any information relating to the background and thereby reduce the size of the dataset. The other key parameter mentioned was the cell size, an exhaustive approach was taken for this parameter, the cell size was varied from 5x5 to 150x150 in steps of 5 pixels, and it was found that even for the limited data available, there was no clear cell size that worked for each image, or even worked for each target. This was believed to be due to the nature of how the scatter plot was created and the numerous factors that cause the shape and sparsity of the scatter plot to change e.g. bit depth of the data, spectral reflectivity and atmospheric absorption in bands selected.

Despite these limitations, the main object of Chapter 3, was to produce an improved anomaly detection method. This was achieved for both hyperspectral images tested using certain bands and cell sizes when compared with the established endmember selection algorithm SMACC.

2. Chapter 4 developed a MATLAB [139] based user interface for the atmospheric modelling software MODTRAN, this is important because to achieve accurate target detection in hyperspectral images, atmospheric correction is a necessary step. Using this interface various atmospheric models were created using MODTRAN 5.2.1, these each produced radiance profiles for key information, i.e. total radiance received by the observer and solar scattering contribution. Each of these models varied a key parameter in the atmospheric model, the results of all models were then compared and it was found that parameters such as the time of day, the day of the year, latitude, the height of the observer and aerosol model all had significant effects on the produced radiance profiles for the VNIR region of the spectrum. Where other parameters such as the atmospheric profile, carbon dioxide (CO₂) mixing ratio, longitude and viewing zenith angle had small effects but were much less severe and could essentially be ignored in any future calculations for this application. It was also found that the one parameter that had severe

effects on certain regions of the VNIR imagery was the column water vapour, this was due to the water absorption bands at approximately 760 nm and from about 900 nm to 1000 nm causing major absorption effects to occur in these regions, however, the rest of the data was unaffected.

3. The process for using MODTRAN atmospheric models to atmospherically correct hyperspectral data was proposed in Chapter 5, also proposed were two scene-based approaches that used prior known spectra, KS Grass and KS Tarmac, all three of these techniques were then compared with two well established scene-based approaches (Internal Average Relative Reflectance and Darks Object Subtraction).

The results from this comparison showed that all three proposed techniques have very good performance, and are an improvement over the currently used scene-based approaches for the purpose of spectral matching using the SAM. It was also clear that for the VNIR imagery, the proposed radiative transfer technique had higher recall than all the other techniques tested for each of the four VNIR images, and in terms of percentage of targets found it performed as well as or better than the current techniques.

However, as predicted, the SWIR imagery did provide a much greater challenge than the VNIR imagery, due to the much greater atmospheric effects present in the SWIR images. For both techniques, the MODTRAN approach provided no improvement in terms of recall and percentage of targets found, and in fact, produces worse results for the recall in certain images. Nonetheless, using both the KS Tarmac and KS Grass techniques the results for both performance metrics for all four SWIR images are an improvement over all the current techniques tested.

4. In Chapter 6 a practical application for HSI was proposed for the purpose of erosion detection in wind turbine blades. Having selected various key types of erosion that can occur in wind turbine blades operating in the field, these damage types were simulated on a section of wind turbine blade in the lab. Using a NIR hyperspectral camera images were taken off the blade and the results were compared with images taken from a high-definition RGB camera. The results showed that

there was far more detail present in the hyperspectral images, mainly key information about the depth of erosion could be extracted. Further studies were then performed on finding the limits that this technique could still accurately detect the erosion by lowering the level of illumination. It was found that by reducing the light level it was no longer possible to detect any signs of erosion from the hyperspectral images. However by modelling the surface of the blade with a polynomial surface estimation and subtracting this from the image. For the new images produced it was again possible to detect the erosion and retrieve detail about the depth of the erosion present on the wind turbine blade.

7.2 Future Work

From the research and results presented in this thesis, there are a number of areas that could be researched further in any future work. These are summarised below:

1. Based on the results from the proposed POHMT technique for anomaly detection in hyperspectral images, it is clear that this technique shows initial progress. Despite the good results, future work could be carried out to create an unsupervised method by automatically determining the optimal bands for creating the scatter plot, also a supervised technique could also be developed where input information of the target could be used to direct the selection of bands. Another part that could be developed further is the cell size used, by classifying the shape or sparsity of the scatter plot it could be determined based on the scatter plot what a necessary cell size could be.
2. Building on the interface developed in Chapter 4, it would be possible to perform a more extensive examination of the additional parameters used when creating a MODTRAN model. Although extensive research has already been performed on most of these other parameters. It was found through the literature that they are unlikely to have any major effects, it could still be useful to have a complete study of every parameter built into the models.
3. In Chapter 5 the process of atmospheric correction of hyperspectral data using

the models generated in Chapter 4 was proposed, however for this chapter the adjacency effect was ignored, further research could be performed that would take this effect into consideration. Also, a software such as HydroLight [175] could also be explored for the correction of underwater targets.

4. Chapter 6 demonstrated a practical application of hyperspectral imaging for the purpose of erosion detection in wind turbine blades. As all the testing was performed on a section of blade in the lab, and all the erosion present on the blade was simulated erosion. Future work could be undertaken to test the application of the techniques presented in this work on wind turbines operating in the field with genuine erosion present on the blade. These results could then be compared with the traditional rope-access and RGB methods discussed in the chapter. Another area for further research of these techniques would be for the testing of wind turbine blades, due to the additional information extracted from the hyperspectral images, mainly the depth of erosion, this data could be used more extensively in the manufacturing and testing of wind turbine blades.

Appendix A

MODTRAN Variables

MODEL selects one of six geographical-seasonal model atmospheres or specifies that user-defined meteorological or radiosonde data are to be used.

MODEL	=	1	Tropical Atmosphere (15° North Latitude)
		2	Mid-Latitude Summer (45° North Latitude)
		3	Mid-Latitude Winter (45° North Latitude)
		4	Sub-Arctic Summer (60° North Latitude)
		5	Sub-Arctic Winter (60° North Latitude)
		6	1976 US Standard Atmosphere

TPTEMP	>	0	Boundary temperature [K] of 'image pixel' (i.e., at H2), used in the radiation mode (if IEMSC = 1, 2 or 4) for slant paths that intersect the earth OR terminate at a gray boundary (for example, cloud, target). If the 'area-average' temperature (AATEMP, CARD 4A) is not entered and the line-of-sight intersects the earth, TPTEMP is also used as the lower boundary temperature in the multiple scattering models.
	≤	0	No surface emission if H2 is above ground. If the path intersects the Earth and TPTEMP is either not positive or left blank, MODTRAN uses the temperature of the first atmospheric level as the boundary temperature. If the 'area-average' temperature (AATEMP, CARD 4A) is not entered and the line-of-sight intersects the earth, the temperature of the first atmospheric level is also used as the lower boundary temperature in the multiple scattering models.

SURREF	=	BRDF	Surface spectral BRDFs (Bidirectional Reflectance Distribution Functions) are specified by CARD 4A, 4B1, 4B2 and 4B3 inputs
	=	LAMBER	Spectral Lambertian surface(s) is (are) specified by CARD 4A, 4L1 and 4L2 inputs
	≥	0	Albedo of the earth, equal to one minus the surface emissivity and spectrally independent (constant).
	<	0	Negative integer values allow the user to access pre-stored spectrally variable surface albedos from the 'DATA/spec_alb.dat' file.

CO2MX		CO2 mixing ratio in ppmv. The default value (used when CO2MX = blank or 0.) is 330 ppmv; the current (1999) recommended value is closer to 365 ppmv (Dutton, 1999).
--------------	--	---

H2OSTR		Vertical water vapor column character string. If blank or 0., the default water vapor column is used. If the first non-blank character is 'g', the water vapor column in g / cm2 follows 'g' (e.g., g 2.0). If the first non-blank character is an 'a', the water column in ATM-cm follows 'a' (e.g., a 3000.). Otherwise, a positive value is interpreted as a scaling factor for the water column (e.g., 2.0 doubles the default water vapor column). If H2OSTR is used with a constant pressure path, i.e., MODEL = 0 (CARD 1), the scaling will be applied to the water density; if an absolute water column amount is input, then the scaling factor is defined relative to the model atmosphere specified by M2 (assuming it is not 0). The water density within water clouds (ICLD = 1-10) is not scaled. Also, the water number density at each profile altitude will not be increased above 100% RH (relative humidity) or by more than 5 times the original value. When the 100% RH limit is reached, the water is distributed to other levels to the extent possible to achieve the input water column. There is a new H2OSTR option to ignore the 100% relative humidity limit. This option is invoked by setting the first non-blank character in H2OSTR to "+", a plus sign. Thus, if one wants to set the water column to 3.0 g / cm2 without the 100% RH limit, set H2OSTR to '+g3.0'.
---------------	--	--

IHAZE selects the type of extinction and a default meteorological range for the boundary-layer aerosol models only. If VIS is also specified, it will override the default IHAZE value. Interpolation of the extinction coefficients based on relative humidity is performed only for the RURAL, MARITIME, URBAN, and TROPOSPHERIC coefficients used in the boundary layer (0 to 2 km altitude).

IHAZE	=	-1	No aerosol attenuation, but the model clouds may be included (i.e., ICLD = 1, 2... 10, 18 and 19).
	=	0	No aerosol or cloud attenuation included in the calculation.
	=	1	RURAL extinction, default VIS = 23 km.
	=	2	RURAL extinction, default VIS = 5 km.
	=	3	NAVY MARITIME extinction. Sets VIS based on wind speed and relative humidity.
	=	4	MARITIME extinction, default VIS = 23 km (LOW-TRAN model).
	=	5	URBAN extinction, default VIS = 5 km.
	=	6	TROPOSPHERIC extinction, default VIS = 50 km.
	=	7	User-defined aerosol extinction coefficients. Triggers reading CARDS 2D, 2D1 and 2D2 for up to 4 altitude regions of user-defined extinction, absorption and asymmetry parameters. (This option is kept for backward compatibility; the ARUSS = 'USS' option affords greater flexibility in specifying user-defined aerosols).
	=	8	FOG1 (Advection Fog) extinction, 0.2 km VIS.
	=	9	FOG2 (Radiative Fog) extinction, 0.5 km VIS.
	=	10	DESERT extinction, sets visibility from wind speed (WSS).

H1	=	Initial altitude (km). H1, the initial altitude, always defines the position of the observer (or sensor).
-----------	---	---

ANGLE	=	Initial zenith angle (degrees) as measured from H1.
--------------	---	---

IDAY	=	Day of the year, used to correct for variation in the earth-to-sun distance.
-------------	---	--

PARM1	=	Observer latitude (-90° to $+90^\circ$)
PARM2	=	Observer longitude (0° to 360° west of Greenwich)
TIME	=	Greenwich mean time in decimal hours, that is, 8:45 a.m. is 8.75, 5:20 p.m. is 17.33 etc.
V1	=	Initial frequency in wavenumber [cm ⁻¹]
V2	=	Final frequency in wavenumber [cm ⁻¹]
DV	=	Frequency increment used for spectral outputs.

Table A.1: MODTRAN variable descriptions and options [176]

Appendix B

MODTRAN Card Formats

In the following, optional cards are indented. The mandatory input CARDS are **CARD 1**, **CARD 1A**, **CARD 2**, **CARD 3**, **CARD 4** and **CARD 5**. Newer inputs are in *Italics*. Note that all floating point inputs are entered using a 'Fn.0' format; this format will properly read any floating point entry, e.g. '1.233', **AND** will also properly read integers as floating point real variables, (either 1234. or 1234 with no decimal).

CARD 1: MODTRN, SPEED, BINARY, LYMOLC, MODEL, T_BEST, ITYPE, IEMSCT, IMULT, M1, M2, M3, M4, M5, M6, MDEF, I_RD2C, NOPRNT, TPTEMP, SURREF)
FORMAT (4A1, I1, A1, I4, 10I5, 1X, I4, F8.0, A7)

CARD 1A: DIS, DISAZM, DISALB, NSTR, SFWHM, CO2MX, H2OSTR, O3STR, C_PROF, LSUNFL, LBMNAM, LFLTNM, H2OAER, CDTDIR, SOLCON, CDASTM, ASTMC, ASTMX, ASTMO, AERRH, NSSALB
FORMAT(3A1, I3, F4.0, F10.0, 2A10, 2A1, 4(1X, A1), F10.0, A1, F9.0, 3F10.0, I10)

CARD 1A1: USRSUN
FORMAT (A256) (If LSUNFL = 'T')

CARD 1A2:	BMNAME FORMAT (A256)	(If LBMNAM = 'T','t','4'or'2')
CARD 1A3:	FILTNM FORMAT (A256)	(If LFLTNM = ' T ')
CARD 1A4:	DATDIR FORMAT (A256)	(If CDDIR = ' T ')
CARD 1A5:	(S_UMIX(IMOL), IMOL = 4, 12) FORMAT (9F5.0)	(If C_PROF = 1, 3, 5 or 7)
CARD 1A6:	(S_XSEC(IMOL), IMOL = 1, 13) FORMAT (13F5.0)	(If C_PROF = 2, 3, 6 or 7)
CARD 1A7:	(S_TRAC(IMOL), IMOL = 1, 16) FORMAT (16F5.0)	(If C_PROF = 4, 5, 6 or 7)
CARD 1B:	(AWAVLN(ISSALB), AS- SALB(ISSALB), ISSALB=1, NSSALB) FORMAT ((8F10.0))	(If NSSALB >0)

Alternative CARD 1B:	ACOALB, RHASYM FORMAT(2F10.0)	(If NSSALB <0 and ASTMX >0.)
--------------------------------	--------------------------------------	---------------------------------

Table B.1: Format for MODTRAN Card 1 [176]

CARD 2:	APLUS, IHAZE, CNOVAM, ISEASN, ARUSS, IVULCN, ICSTL, ICLD, IVSA, VIS, WSS, WHH, RAINRT, GNDALT FORMAT (A2, I3, A1, I4, A3, I2, 3I5, 5F10.0)	
----------------	--	--

CARD 2A+:	ZAER11, ZAER12, SCALE1, ZAER21, ZAER22, SCALE2, ZAER31, ZAER32, SCALE3, ZAER41, ZAER42, SCALE4 FORMAT ((3(1X, F9.0), 20X, (If APLUS = 'A+') 3(1X, F9.0)))	
------------------	--	--

CARD 2A:	CTHIK, CALT, CEXT FORMAT (3F8.0)	(If ICLD = 18 or 19)
-----------------	-------------------------------------	----------------------

Alternate CARD 2A:	CTHIK, CALT, CEXT, NCRALT, NCRSPC, CWAVLN, CCOLWD, CCOLIP, CHUMID, ASYMWD, ASYMIP FORMAT (3F8.0, 2I4, 6F8.0)	(If 0 <ICLD ≤ 10)
------------------------------	--	-------------------

CARD 2B:	ZCVSA, ZTVSA, ZINVSA FORMAT (3F10.0)	(If IVSA = 1)
-----------------	---	---------------

CARD 2C:	ML, IRD1, IRD2, HMODEL, REE, NMOLYC, E_MASS, AIRMWT FORMAT(3I5, A20, F10.0, I5, 2F10.0)	(If MODEL = 0, 7 or 8; & I_RD2C = 1)
-----------------	--	---

CARD 2CY: (YNAME(I), I=1, NMOLYC)
 FORMAT ((8A10)) (If NMOLYC >0)

CARDS 2C1, 2C2, 2C2X, 2C2Y and 2C3 (as required) are each repeated ML times.

CARD 2C1: ZM, P, T, WMOL(1), WMOL(2),
 WMOL(3), (JCHAR(J), J = 1,
 14), JCHARX, JCHARY
 FORMAT 6F10.0, 14A1, 1X, 2A1)

CARD 2C2: (WMOL(J), J = 4, 12)
 FORMAT (8F10.0, /F10.0) (If IRD1 = 1)

CARD 2C2X: (WMOLX(J), J = 1, 13)
 FORMAT (8F10.0, /5F10.0) (If MDEF = 2 &
 IRD1 = 1)

CARD 2C2Y: (WMOLY(J), J = 1, NMOLYC)
 FORMAT ((8F10.0)) (If NMOLYC >0 &
 IRD1 = 1)

CARD 2C3: AHAZE, EQLWCZ, RRATZ,
 IHA1, ICLD1, IVUL1, ISEA1,
 ICHR
 FORMAT (10X, 3F10.0, 5I5) (If IRD2 = 1)

CARD 2C3: AHAZE(1), RRATZ, AHAZE(2),
 AHAZE(3), AHAZE(4)
 FORMAT(10X, F10.0, 10X, (If IRD2 = 2)
 4F10.0)

CARD 2D: (IREG(N), N = 1, 2, 3, 4)
 FORMAT (4I5) (If IHAZE = 7,
 ICLD = 11 or
 ARUSS='USS')

CARDs 2D1 and 2D2 pairs are repeated for each N (1 to 4) for which

IREG(N) > 0 (ARUSS='USS') or
IREG(N) \neq 0 (IHAZE=7 or ICLD=11)

CARD 2D1: AWCCON, AERNAM
 FORMAT (F10.0, A70)

CARD 2D2: (VARSPC(N, I), EXTC(N, I),
 ABSC(N, I), ASYM(N, I), I = 1,
 2, ..., I_{max})
 FORMAT ((3(F6.2, 2F7.5, F6.4))) If ARUSS = 'USS'
 & IREG(N) > 1, then
 I_{max} = IREG(N);
 Else I_{max} = 47

CARD 2E1: (ZCLD(I, 0), CLD(I, 0),
 CLDICE(I, 0), RR(I, 0), I =
 1, NCRAIT)
 FORMAT ((4F10.5)) (If ICLD = 1 -
 10, NCRAIT \geq 2,
 MODEL < 8)

Alternate
CARD 2E1: (PCLD(I, 0), CLD(I, 0),
 CLDICE(I, 0), RR(I, 0), I =
 1, NCRAIT)
 FORMAT ((4F10.5)) (If ICLD = 1 -
 10, NCRAIT \geq 2,
 MODEL = 8)

Alternate	CFILE, CLDTYP, CIRTYP
CARD 2E2:	
	FORMAT ((A256)) (If ICLD = 1 - 10, NCRSPC = 1)

Table B.2: Format for MODTRAN Card 2 [176]

Alternate	H1, H2, ANGLE, IDAY, RO, ISOURC, AN-
CARD 3:	GLEM
	FORMAT (3F10.0, I5, 5X, F10.0, I5, F10.0) (If IEMSCT = 3)

CARD 3A2: PARM1, PARM2, PARM3,
 PARM4, TIME, PSIPO, AN-
 GLEM, G
 FORMAT (8F10.0) (If IEM SCT = 2 or 4)

CARD 3B1: NANGLS, NWLF
 FORMAT (2I5) (If IEMSCT = 2
 or 4; IPH = 1)

CARD 3B2:	(ANGF(I), F(1, I, 1), F(2, I, 1), F(3, I, 1), F(4, I, 1), I = 1, NAN- GLS) FORMAT (5F10.0)	(If IEMSCCT = 2 or 4; IPH = 1; NWLF = 0)
------------------	---	---

CARD 3C1:	(ANGF(I), I = 1, NANGLS) FORMAT (8F10.0)	(If IEMSCCT = 2 or 4; IPH = 1; NWLF >0)
------------------	---	--

CARD 3C2:	(WLF(J), J = 1, NWLF) FORMAT (8F10.0)	(If IEMSCCT = 2 or 4; IPH = 1; NWLF >0)
------------------	--	--

In **CARDs 3C3-3C6**, 'IANG' is angle index as in **CARD3C1** and 'JWAV' is the wavelength index as in **CARD3C2**.

CARD 3C3:	(F(1, IANG, JWAV), JWAV = 1, NWLF) FORMAT (8F10.0)	(If IEMSCCT = 2 or 4; IPH = 1; NWLF >0)
------------------	--	--

CARD 3C4:	(F(2, IANG, JWAV), JWAV = 1, NWLF) FORMAT (8F10.0)	(If IEMSCCT = 2 or 4; IPH = 1; NWLF >0)
------------------	--	--

CARD 3C5:	(F(3, IANG, JWAV), JWAV = 1, NWLF) FORMAT (8F10.0)	(If IEMSCCT = 2 or 4; IPH = 1; NWLF >0)
------------------	--	--

CARD 3C6:	(F(4, IANG, JWAV), JWAV = 1, NWLF) FORMAT (8F10.0)	(If IEMSCCT = 2 or 4; IPH = 1; NWLF >0)
------------------	--	--

Table B.3: Format for MODTRAN Card 3 [176]

CARD 4:	V1, V2, DV, FWHM, YFLAG, XFLAG, DLIMIT, FLAGS, MLFLX, VRFRAC FORMAT (4F10.0, 2A1, A8, A7, I3,F10.0)
----------------	---

CARD 4A:	NSURF, AATEMP, DH2O, ML- TRFL FORMAT (I1, 2F9.0, A1)	(If SURREF = 'BRDF' or 'LAM- BER')
-----------------	--	--

The set of **CARD4B1**, **4B2**, and **4B3** inputs is repeated NSURF times.

CARD 4B1:	CBRDF FORMAT (A80)	(If SURREF = 'BRDF')
------------------	-----------------------	-------------------------

CARD 4B2:	NWVSRF, SURFZN, SURFAZ FORMAT (*)	(If SURREF = 'BRDF')
------------------	--------------------------------------	-------------------------

CARD4B3 is repeated NWVSRF times.

CARD 4B3:	WVSURF, (PARAMS(I), I = 1, NPARAM) FORMAT (*)	(If SURREF = 'BRDF')
------------------	---	-------------------------

CARD 4L1:	SALBFL FORMAT (A256)	(If SURREF = 'LAMBER')
------------------	-------------------------	---------------------------

CARD4L2 is repeated NSURF times.

CARD 4L2:	CSALB FORMAT (A80)	(If SURREF = 'LAMBER')
------------------	-----------------------	---------------------------

Table B.4: Format for MODTRAN Card 4 [176]

CARD 5:	IRPT FORMAT (I5)
----------------	---------------------

Table B.5: Format for MODTRAN Card 5 [176]

Appendix C

MODTRAN GUI

C.1 Read Card 1

```
1      function [line_count,MOD] = readCard1(lines,line_count,MOD)
2          line = lines{line_count};
3          FieldWidth = [1 1 1 1 1 5 5 5 5 5 5 5 5 5 5 5 1 4 8 7];
4          FieldFormat = {'c', 'c', 'c', 'c', 'd', 'd', ...
                          'd', 'd', 'd', 'd', 'd', 'd', 'd', 'd', 'd', 'd', '*', 'd', ...
                          'f', '7c'};
5          [data] = readCard(line,FieldWidth,FieldFormat);
6          [MOD.MODTRN, MOD.SPEED, MOD.BINARY, MOD.LYMOLC, ...
            MOD.MODEL, MOD.ITYPE, MOD.IEMSCT, MOD.IMULT, MOD.M1, ...
            MOD.M2, MOD.M3, MOD.M4, MOD.M5, MOD.M6, MOD.MDEF, ...
            MOD.I_RD2C, MOD.NOPRNT, MOD.TPTEMP, MOD.SURREF] = ...
            data{:};
7          line_count = line_count + 1;
8      end
```

Fig. C.1: Function for reading card 1

C.2 Read Card General

```
1      function [data] = readCard(line,FieldWidth,FieldFormat)
2          Stop = cumsum(FieldWidth);
3          Start = Stop - FieldWidth + 1;
4          if length(line) < max(Stop)
5              line = [line blanks(max(Stop) - length(line))];
6          elseif length(line) > max(Stop)
7              line = line(1:max(Stop));
8          end
9          data = cell(1,length(FieldWidth));
10         for i = 1:length(FieldWidth)
11             if FieldFormat{i} ≠ '*'
12                 data{i} = sscanf(line(Start(i):Stop(i)), ['%' ...
13                     FieldFormat{i}]);
14                 if isempty(data{i})
15                     data{i} = 0;
16                 end
17             end
18         end
19         emptyCells = cellfun(@isempty,data);
20         data(emptyCells) = [];
```

Fig. C.2: Function for reading general card

C.3 Read LTN File

```

1      function MOD = readLTN(filename)
2          MOD = MODTRAN;
3          splitStr = regexp(filename, '\\', 'split');
4          splitStr = regexp(splitStr{length(splitStr)}, '/', 'split');
5          MOD.caseName = splitStr{length(splitStr)};
6          MOD.caseIndex = 1;
7          fileID = fopen(filename);
8          i=0;
9          while ~feof(fileID)
10             fgetl(fileID);
11             i = i+1;
12         end
13         frewind(fileID)
14         lines = cell(i,1);
15         for i=1:i
16             lines{i} = fgetl(fileID);
17         end
18         fclose(fileID);
19         [line_count,MOD] = readSingleRun(lines,1,MOD);
20         fileFinished = (isempty(MOD.IRPT) || MOD.IRPT == 0 || ...
21             MOD.IRPT == 2);
22     end

```

Fig. C.3: Function for reading complete LTN file

C.4 Write Card 1

```

1      function writeCard1(fileID,MOD)
2          stringFormat = ...
3              '%c%c%c%c%1d%5d%5d%5d%5d%5d%5d%5d%5d%5d ...
4              %4d%8.3f%7s\n';
5          fprintf(fileID,stringFormat,MOD.MODTRN, MOD.SPEED, ...
6              MOD.BINARY, MOD.LYMOLC, MOD.MODEL, MOD.ITYPE, ...
7              MOD.IEMSCT, MOD.IMULT, MOD.M1, MOD.M2, MOD.M3, ...
8              MOD.M4, MOD.M5, MOD.M6, MOD.MDEF, MOD.I_RD2C, ...
9              MOD.NOPRNT, MOD.TPTEMP, MOD.SURREF);
10     end

```

Fig. C.4: Function for writing card 1

C.5 Write LTN File

```

1      function writeLTN(MOD)
2          save_path = 'Output Data\LTN Files\';
3          if ~exist(save_path, 'dir'),
4              mkdir(save_path)
5          end
6          fileID = fopen([save_path MOD(1).caseName], 'wt');
7          %% Card 1
8          writeCard1(fileID, MOD);
9          %% Card 1A
10         writeCard1A(fileID, MOD);
11         %% Card 1A1
12         if strcmpi(MOD.LSUNFL, 'T')
13             writeCard1A1(fileID, MOD);
14         end
15         %% Card 1A2
16         if any(MOD.LBMNAM == 'Tt42')
17             writeCard1A2(fileID, MOD);
18         end
19         %% Card 1A3
20         if strcmpi(MOD.LFLTNM, 'T')
21             writeCard1A3(fileID, MOD);
22         end
23         %% Card 1A4
24         if strcmpi(MOD.CDTHDR, 'T')
25             writeCard1A4(fileID, MOD);
26         end
27         %% Card 1A5
28         if any(MOD.C_PROF == '1357')
29             writeCard1A5(fileID, MOD);
30         end
31         %% Card 1A6
32         if any(MOD.C_PROF == '2367')
33             writeCard1A6(fileID, MOD);
34         end
35         %% Card 1A7
36         if any(MOD.C_PROF == '4567')
37             writeCard1A7(fileID, MOD);
38         end
39         %% Card 1B and Alt 1B
40         if ~isempty(MOD.NSSALB) && MOD.NSSALB > 0
41             writeCard1B(fileID, MOD);
42         elseif ~isempty(MOD.NSSALB) && MOD.NSSALB < 0 && ...
43             MOD.ASTMX > 0
44             writeCard1BAlt(fileID, MOD);
45         end
46         %% Card 2
47         writeCard2(fileID, MOD);

```

```

47     %% Card 2A+
48     if strcmp(MOD.APLUS, 'A+')
49         writeCard2APlus(fileID,MOD);
50     end
51     %% Card 2A
52     if MOD.ICLD == 18 || MOD.ICLD == 19
53         writeCard2A(fileID,MOD);
54     end
55     %% Card 2A Alt
56     if any(MOD.ICLD == 1:10)
57         writeCard2AAlt(fileID,MOD);
58     end
59     %% Card 2B
60     if MOD.IVSA == 1
61         writeCard2B(fileID,MOD);
62     end
63     %% Card 2C
64     if any(MOD.MODEL == [0 7 8]) && MOD.I_RD2C == 1
65         writeCard2C(fileID,MOD);
66     %% Card 2CY
67     if any(MOD.NMOLYC > 0) && MOD.MDEF == 2
68         writeCard2CY(fileID,MOD);
69     end
70     %% Card 2C1, 2C2 & 2C3
71     for i = 1:MOD.ML
72         writeCard2C1(fileID,MOD,i);
73         if MOD.IRD1 == 1
74             writeCard2C2(fileID,MOD,i);
75         end
76         if MOD.MDEF == 2 && MOD.IRD1 == 1
77             writeCard2C2X(fileID,MOD,i);
78         end
79         if MOD.NMOLYC > 0 && MOD.IRD1 == 1
80             writeCard2C2Y(fileID,MOD,i);
81         end
82         if MOD.IRD2 == 1
83             writeCard2C3(fileID,MOD,i);
84         elseif MOD.IRD2 == 2
85             writeCard2C3Alt(fileID,MOD,i);
86         end
87     end
88     %% Card 2D
89     if MOD.IHAZE == 7 || MOD.ICLD == 11 || strncmpi(MOD.ARUSS, 'USS', 3)
90         writeCard2D(fileID,MOD);
91         for iNREG = 1:4
92             if MOD.IREG(iNREG) > 0
93                 writeCard2D1(fileID,MOD,iNREG);
94                 if strncmpi(MOD.ARUSS, 'USS', 3) && (MOD.IREG(iNREG) > 1)
95                     writeCard2D2(fileID,MOD,iNREG,MOD.IREG(iNREG));

```

```
97     else
98     writeCard2D2(fileID,MOD,iNREG,47);
99     end
100    end
101    end
102    end
103    %% Card 2E1
104    if any(MOD.ICLD == 1:10) && MOD.NCRALT ≥ 3
105    if MOD.MODEL < 8
106    for iNCRALT = 1:MOD.NCRALT
107    writeCard2E1(fileID,MOD,iNCRALT);
108    end
109    elseif MOD.MODEL == 8
110    for iNCRALT = 1:MOD.NCRALT
111    writeCard2E1Alt(fileID,MOD,iNCRALT);
112    end
113    end
114    end
115    %% Card 2E2
116    if any(MOD.ICLD == 1:10)
117    if MOD.NCRSPC ≥ 2
118    for iNCRSPC = MOD.NCRSPC
119    writeCard2E2(fileID,MOD,iNCRSPC);
120    end
121    elseif MOD.NCRSPC == 1
122    writeCard2E2Alt(fileID,MOD);
123    end
124    end
125    %% Card 3
126    writeCard3All(fileID,MOD);
127    %% Card 4
128    writeCard4(fileID,MOD);
129    %% Card 4A, 4B1-2 & 4L1-2
130    writeCard4Ato4L2(fileID,MOD);
131    %% Card 5
132    writeCard5(fileID,MOD);
133    fclose(fileID);
134    end
```

Fig. C.5: Function for writing LTN file

C.6 Generate Multiple LTN Files

```

1      for a = 1:length(model)
2      MODEL = num2str(model(a));
3      for b = 1:length(tptemp)
4      TPTEMP = num2str(tptemp(b), '%.3f');
5      for c = 1:length(surref)
6      SURREF = num2str(surref(c), '%.5f');
7      for d = 1:length(co2mx)
8      CO2MX = num2str(co2mx(d), '%.3f');
9      for e = 1:length(h2ostr)
10     H2OSTR = num2str(h2ostr(e), '%.6f');
11     for f = 1:length(ihaze)
12     IHAZE = num2str(ihaze(f));
13     for g = 1:length(h1)
14     H1 = num2str(h1(g), '%.6f');
15     for h = 1:length(angle)
16     ANGLE = num2str(angle(h), '%.6f');
17     for i = 1:length(iday)
18     IDAY = num2str(iday(i));
19     for j = 1:length(parm1)
20     PARM1 = num2str(parm1(j), '%.3f');
21     for k = 1:length(parm2)
22     PARM2 = num2str(parm2(k), '%.3f');
23     for l = 1:length(time)
24     TIME = num2str(time(l), '%.3f');
25     writeLTN(MODEL, TPTEMP, SURREF, CO2MX, H2OSTR, IHAZE, H1, ...
              ANGLE, IDAY, PARM1, PARM2, TIME, V1, V2, DV, save_path);
26     end      end      end      end      end      end      end      end ...
              end      end      end      end

```

Fig. C.6: Code used to create multiple LTN files

C.7 Running MODTRAN

```
1 function runMODTRAN(filename,plot_flag)
2     if nargin == 1
3         plot_flag = 0;
4     end
5     if exist('C:\PcModWin5\bin', 'dir')
6         pcmowin_dir = 'C:\PcModWin5\bin';
7     else
8         pcmowin_dir = uigetdir;
9     end
10    tmp = pwd;
11    name = regexp(filename, '\\', 'split');
12    name = regexp(name{length(name)}, '/', 'split');
13    name = name{length(name)};
14    if any(name==43)
15        name = regexpprep(name, '\\+', 'Plus');
16        filename = regexpprep(filename, '\\+', 'Plus');
17    end
18    path = regexp(filename, name, 'split');
19    path = path{1};
20    name = regexp(name, '.ltn', 'split');
21    name = name{1};
22    save_path_MODTRAN = 'Output Data\MODTRAN Output\';
23    if ~exist(save_path_MODTRAN, 'dir'),
24        mkdir(save_path_MODTRAN)
25    end
26    save_path_MATLAB = 'Output Data\MATLAB Output\';
27    if ~exist(save_path_MATLAB, 'dir'),
28        mkdir(save_path_MATLAB)
29    end
30    cd(pcmowin_dir)
```

```
31     %% Run MODTRAN
32     count = 0;
33     while 1
34         copyfile(filename, 'modin')
35         copyfile('modin', 'tape5')
36         [~, ~] = system('Mod5_2_2_0.exe');
37         copyfile('tape7', [tmp '\ ' save_path_MODTRAN name '.fl7'])
38         d = dir([tmp '\ ' save_path_MODTRAN name '.fl7']);
39         f_size = [d(:).bytes];
40         if f_size~=0 %#ok<BDSCI>
41             cd(tmp)
42             read_fl7(name,tmp,save_path_MODTRAN,save_path_MATLAB);
43             break
44         end
45         count = count + 1;
46         if count == 5
47             errordlg({'Error in MODTRAN Code', 'Check .ltn Files'});
48             cd(tmp)
49             return;
50         end
51     end
52 end
```

Fig. C.7: Code used to run multiple LTN files

C.8 Manual Input

```

1  function manualInput(save_path)
2      MODEL = input('MODEL = ');
3      TPTEMP = input('TPTEMP = ');
4      SURREF = input('SURREF = ');
5      CO2MX = input('CO2MX = ');
6      H2OSTR = input('H2OSTR = ');
7      IHAZE = input('IHAZE = ');
8      H1 = input('H1 = ');
9      ANGLE = input('ANGLE = ');
10     IDAY = datenum([input('IDAY (dd mmm) = ','s') ' 0001'])-366;
11     PARM1 = input('PARM1 = ');
12     PARM2 = input('PARM2 = ');
13     TIME = time2dec(input('TIME (hhmm) = '));
14     V1 = num2str(10^7/input('V1 = ','%.4f'));
15     V2 = num2str(10^7/input('V2 = ','%.4f'));
16     DV = num2str(10^7/input('DV = ','%.4f'));
17     %% Create Settings File
18     fid = fopen([save_path 'settings.cfg'],'wt');
19     fprintf(fid,['MODEL = ' num2str(MODEL) '\n']);
20     fprintf(fid,['TPTEMP = ' num2str(TPTEMP) '\n']);
21     fprintf(fid,['SURREF = ' num2str(SURREF) '\n']);
22     fprintf(fid,['CO2MX = ' num2str(CO2MX) '\n']);
23     fprintf(fid,['H2OSTR = ' num2str(H2OSTR) '\n']);
24     fprintf(fid,['IHAZE = ' num2str(IHAZE) '\n']);
25     fprintf(fid,['H1 = ' num2str(H1) '\n']);
26     fprintf(fid,['ANGLE = ' num2str(ANGLE) '\n']);
27     fprintf(fid,['IDAY = ' num2str(IDAY) '\n']);
28     fprintf(fid,['PARM1 = ' num2str(PARM1) '\n']);
29     fprintf(fid,['PARM2 = ' num2str(PARM2) '\n']);
30     fprintf(fid,['TIME = ' num2str(TIME) '\n']);
31     fprintf(fid,['V1 = ' V1 '\n']);
32     fprintf(fid,['V2 = ' V2 '\n']);
33     fprintf(fid,['DV = ' DV '\n']);
34     fclose(fid);
35 end

```

Fig. C.8: Manual input function

C.9 Configuration File Input

```

1      function fileInput(save_path)
2          [FileName,PathName,-] = uigetfile('*.cfg','Select ...
           Configuration File'); % Prompt user for config file
3          [MODEL, TPTEMP, SURREF, CO2MX, H2OSTR, IHAZE, H1, ANGLE, ...
           IDAY, PARM1, PARM2, TIME, V1, V2, DV] = ...
           importSettings(PathName, FileName);
4          %% Create Settings File
5          fid = fopen([save_path 'settings.cfg'],'wt');
6          fprintf(fid,['MODEL = ' num2str(MODEL) '\n']);
7          fprintf(fid,['TPTEMP = ' num2str(TPTEMP) '\n']);
8          fprintf(fid,['SURREF = ' num2str(SURREF) '\n']);
9          fprintf(fid,['CO2MX = ' num2str(CO2MX) '\n']);
10         fprintf(fid,['H2OSTR = ' num2str(H2OSTR) '\n']);
11         fprintf(fid,['IHAZE = ' num2str(IHAZE) '\n']);
12         fprintf(fid,['H1 = ' num2str(H1) '\n']);
13         fprintf(fid,['ANGLE = ' num2str(ANGLE) '\n']);
14         fprintf(fid,['IDAY = ' num2str(IDAY) '\n']);
15         fprintf(fid,['PARM1 = ' num2str(PARM1) '\n']);
16         fprintf(fid,['PARM2 = ' num2str(PARM2) '\n']);
17         fprintf(fid,['TIME = ' num2str(TIME) '\n']);
18         fprintf(fid,['V1 = ' V1 '\n']);
19         fprintf(fid,['V2 = ' V2 '\n']);
20         fprintf(fid,['DV = ' DV '\n']);
21         fclose(fid);
22     end

```

Fig. C.9: Configuration file input function

```

1 search_str = '      FREQ  TOT_TRANS  PTH_THRML  THRML_SCT  ...
      SURF_EMIS  SOL_SCAT  SING_SCAT  GRND_RFLT  DRCT_RFLT  ...
      TOTAL_RAD  REF_SOL  SOL@OBS  DEPTH DIR_EM  TOA_SUN  ...
      BBODY_T[K]';
2 for k = 1:length(nameFiles)
3     fid = fopen([save_path 'MODTRAN Output\' nameFiles{k} ...
        '.fl7'],'r');
4     count = 0;
5     while ~feof(fid)
6         while ~feof(fid)
7             line = fgetl(fid);
8             if strncmp(line, search_str, length(search_str)) == 1
9                 break
10            end
11        end
12        while ~feof(fid)
13            line = fgetl(fid);
14            count = count + 1;
15            if strncmp(line, ' -9999.',7) == 1
16                break
17            end
18        end
19    end
20    nwaves = count - 1;
21    frewind(fid)
22    %% Extract Details
23    while ~feof(fid)
24        while ~feof(fid)
25            line = fgetl(fid);
26            if strncmp(line, search_str, length(search_str)) == 1
27                break
28            end
29        end
30        count = 1;
31        line = fgetl(fid);
32        while strncmp(line, ' -9999.', 7) == 0
33            z = textscan(line, '%f %f %f %f %f %f %f %f %f %f ...
                %f %f %f %f');
34            factor = z{1}.^2./10^4;
35            WAVELENGTH(1,count) = z{1}./factor;
36            TOT_TRANS(1,count) = z{2};
37            PTH_THRML(1,count) = factor*z{3};
38            THRML_SCT(1,count) = factor*z{4};
39            SURF_EMIS(1,count) = factor*z{5};
40            SOL_SCAT(1,count) = factor*z{6};
41            SING_SCAT(1,count) = factor*z{7};
42            GRND_RFLT(1,count) = factor*z{8};
43            DRCT_RFLT(1,count) = factor*z{9};
44            TOTAL_RAD(1,count) = factor*z{10};

```

```
45     REF_SOL(1,count) = factor*z{11};
46     SOL_OBS(1,count) = factor*z{12};
47     DEPTH(1,count) = factor*z{13};
48     DIR_EM(1,count) = factor*z{14};
49     TOA_SUN(1,count) = factor*z{15};
50     BBODY_T(1,count) = factor*z{16};
51     count = count + 1;
52     line = fgetl(fid);
53     end
54     save([save_path 'MATLAB Output\' nameFiles{k} '.mat'], ...
          'WAVELENGTH', 'TOT_TRANS', 'PTH_THRML', 'THRML_SCT', ...
          'SURF_EMIS', 'SOL_SCAT', 'SING_SCAT', 'GRND_RFLT', ...
          'DRCT_RFLT', 'TOTAL_RAD', 'REF_SOL', 'SOL_OBS', 'DEPTH', ...
          'DIR_EM', 'TOA_SUN', 'BBODY_T');
55     end
56     fclose(fid);
```

Fig. C.10: Function to run LTN files in MODRTAN

Appendix D

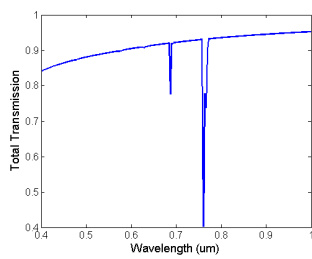
MODTRAN Output Variables

FL7 Name	Name	Units
FREQ	Wavenumber	1/cm
TOT_TRANS	Transmission Total	N/A
PTH_THRML	Path Thermal Radiance	W/cm ² /str/cm ⁻¹
THRML_SCT	Thermal Scatter	W/cm ² /str/cm ⁻¹
SURF_EMIS	Surface Emission Radiance	W/cm ² /str/cm ⁻¹
SOL_SCAT	Solar Scatter Radiance	W/cm ² /str/cm ⁻¹
SING_SCAT	Single Scatter Radiance	W/cm ² /str/cm ⁻¹
GRND_RFLT	Total Ground Reflected Radiance	W/cm ² /str/cm ⁻¹
DRCT_RFLT	Direct Ground Reflected Radiance	W/cm ² /str/cm ⁻¹
TOTAL_RAD	Total Radiance	W/cm ² /str/cm ⁻¹
REF_SOL	Reflected Solar	W/cm ² /str/cm ⁻¹
SOL@OBS	Solar at Observer	W/cm ² /str/cm ⁻¹
DEPTH	Optical Depth	W/cm ² /str/cm ⁻¹
DIR_EM	Emissivity	W/cm ² /str/cm ⁻¹
TOA_SUN	TOA Sun	W/cm ² /str/cm ⁻¹
BBODY_T[K]	Blackbody Temperature	K

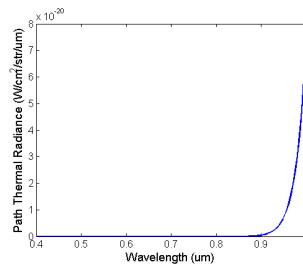
Table D.1: Description of output file values

Appendix E

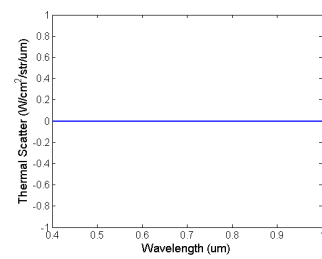
MODTRAN Output Results



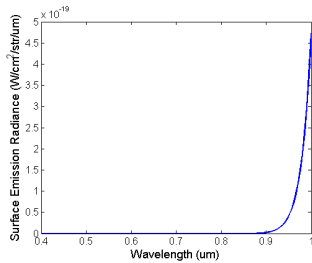
(a) Transmission total



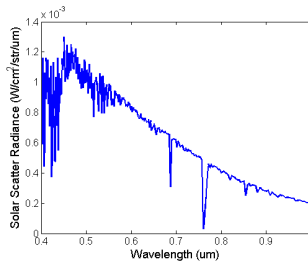
(b) Path Thermal Radiance



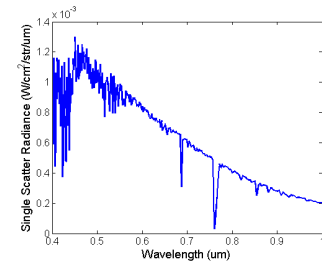
(c) Thermal Scatter



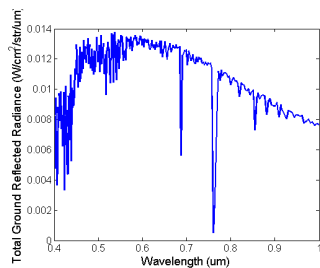
(d) Surface Emission Radiance



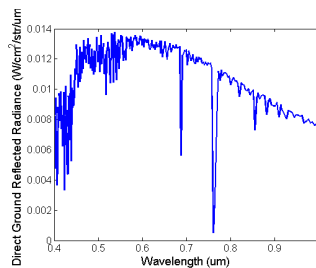
(e) Solar Scatter Radiance



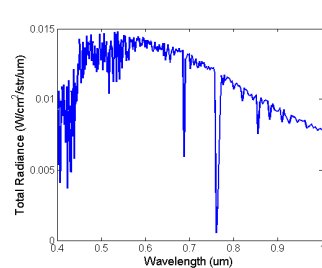
(f) Single Scatter Radiance



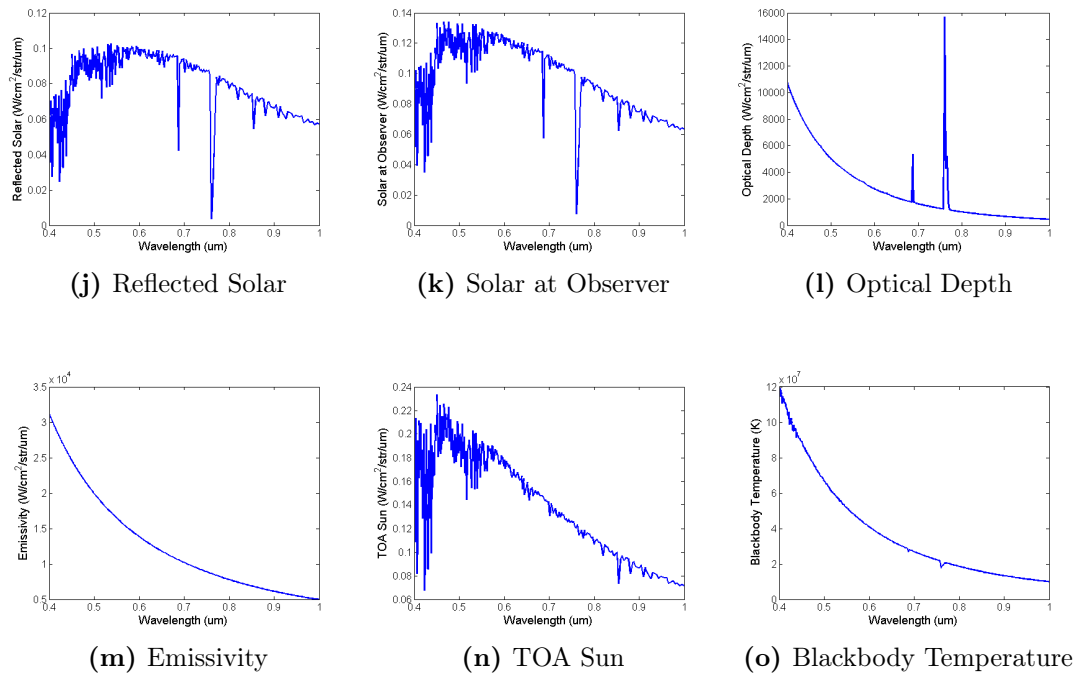
(g) Total Ground Reflected Radiance



(h) Direct Ground Reflected Radiance



(i) Total Radiance

**Fig. E.1:** Output results for initial set-up

Appendix F

Atmospheric Correction Equation Derivation

$$\rho_{obs}(\lambda, \theta, \theta_0, \phi_0) = \frac{\pi L_{obs}(\lambda, \theta, \theta_0, \phi_0)}{\mu_0 F_0(\lambda)} \quad (F.1)$$

where,

θ_0 is the solar zenith angle,

ϕ_0 is the solar azimuth angle,

θ is the sensor zenith angle,

ρ is the sensor azimuth angle,

λ is the wavelength,

L_{obs} is the radiance measured at the satellite,

F_0 is the solar flux at the top of the atmosphere when the solar zenith angle is equal to zero and,

μ_0 is the cosine of solar zenith angle.

According to Tanre et al. [155], when the surface is assumed to be Lambertian and the adjacency effect is neglected, $\rho_{obs}(\lambda, \theta, \theta_0, \phi_0)$ can be expressed approximately as:

$$\rho_{obs}(\lambda, \theta, \theta_0, \phi_0) = [\rho_{atm}(\lambda, \theta, \theta_0, \phi_0) + \frac{t_d(\lambda\theta_0)t_u(\lambda, \theta)\rho(\lambda)}{(1 - s(\lambda)\rho(\lambda))}]T_g(\lambda, \theta, \theta_0) \quad (F.2)$$

where,

ρ_{atm} is the path reflectance,

t_d is the downward scattering transmittance,

t_u is the upward scattering transmittance,

s is the spherical albedo of the atmosphere,

ρ is the surface reflectance and,

T_g is the total gaseous transmittance in the Sun-surface-sensor path.

$$\rho_{obs} = T_g(\rho_{atm} + \frac{t_d t_u \rho}{(1 - s\rho)}) \quad (F.3)$$

$$\rho_{obs} = T_g \rho_{atm} + \frac{T_g t_d t_u \rho}{(1 - s\rho)} \quad (F.4)$$

$$\rho_{obs} = SOL_SCAT + \frac{T_g t_d t_u \rho}{(1 - s\rho)} \quad (F.5)$$

$$\rho_{obs} - SOL_SCAT = \frac{T_g t_d t_u \rho}{(1 - s\rho)} \quad (F.6)$$

$$\rho_{obs} - SOL_SCAT = TOTAL_RAD \times \rho \quad (F.7)$$

$$\rho = \frac{L_{obs} - SOL_SCAT}{TOTAL_RAD} \quad (F.8)$$

References

- [1] D. Hooker, “Lena (Söderberg), a standard test image originally cropped from the November 1972 issue of Playboy magazine,” 1972. [Online]. Available: <http://sipi.usc.edu/database/database.php?volume=misc&image=12>
- [2] D. Manolakis, D. Marden, and G. A. Shaw, “Hyperspectral image processing for automatic target detection applications,” *Lincoln Laboratory Journal*, vol. 14, no. 1, pp. 79–116, 2003.
- [3] Y. Zhang, B. Du, and L. Zhang, “A sparse representation-based binary hypothesis model for target detection in hyperspectral images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 3, pp. 1346–1354, Mar 2015. [Online]. Available: <http://dx.doi.org/10.1109/tgrs.2014.2337883>
- [4] C. Zhang and J. M. Kovacs, “The application of small unmanned aerial systems for precision agriculture: a review,” *Precision Agriculture*, vol. 13, no. 6, pp. 693–712, Jul 2012. [Online]. Available: <http://dx.doi.org/10.1007/s11119-012-9274-5>
- [5] K. Hirose, M. Osaki, T. Takeda, O. Kashimura, T. Ohki, H. Segah, Y. Gao, and M. Evri, “Contribution of hyperspectral applications to tropical peatland ecosystem monitoring,” in *Tropical Peatland Ecosystems*. Springer Nature, 2016, pp. 421–431. [Online]. Available: http://dx.doi.org/10.1007/978-4-431-55681-7_28
- [6] B. Garcia-Salgado and V. Ponomaryov, “Feature extraction scheme for a textural hyperspectral image classification using gray-scaled HSV and NDVI image features vectors fusion,” in *2016 IEEE International Conference on Electronics,*

- Communications and Computers*. Cholula Puebla, Mexico: IEEE, Feb 2016, pp. 186–191. [Online]. Available: <http://dx.doi.org/10.1109/conielecomp.2016.7438573>
- [7] R. Pu, “Detecting and mapping invasive plant species by using hyperspectral data,” *Hyperspectral Remote Sensing of Vegetation*, pp. 447–466, Oct 2016. [Online]. Available: <http://dx.doi.org/10.1201/b11222-27>
- [8] D. Wu and D.-W. Sun, “Advanced applications of hyperspectral imaging technology for food quality and safety analysis and assessment: A review: Part II: Applications,” *Innovative Food Science & Emerging Technologies*, vol. 19, pp. 15–28, Jul 2013. [Online]. Available: <http://dx.doi.org/10.1016/j.ifset.2013.04.016>
- [9] S. Marshall, T. Kelman, T. Qiao, P. Murray, and J. Zabalza, “Hyperspectral imaging for food applications,” in *23rd European Signal Processing Conference*. Nice, Côte d’Azur, France: IEEE, Aug 2015, pp. 2854–2858. [Online]. Available: <http://dx.doi.org/10.1109/eusipco.2015.7362906>
- [10] T. Qiao, J. Ren, J. Zabalza, and S. Marshall, “Prediction of lamb eating quality using hyperspectral imaging,” in *OCM 2015 - Optical Characterization of Materials Conference Proceedings*. Karlsruhe, Germany: KIT Scientific Publishing, 2015, p. 15.
- [11] G. A. Shaw and H. K. Burke, “Spectral imaging for remote sensing,” *Lincoln Laboratory Journal*, vol. 14, no. 1, pp. 3–28, 2003.
- [12] M. Govender, K. Chetty, and H. Bulcock, “A review of hyperspectral remote sensing and its application in vegetation and water resource studies,” *Water SA*, vol. 33, no. 2, Dec 2009. [Online]. Available: <http://dx.doi.org/10.4314/wsa.v33i2.49049>
- [13] N. H. Broge and E. Leblanc, “Comparing prediction power and stability of broadband and hyperspectral vegetation indices for estimation of green leaf area index and canopy chlorophyll density,” *Remote Sensing of Environment*, vol. 76,

- no. 2, pp. 156–172, May 2001. [Online]. Available: [http://dx.doi.org/10.1016/S0034-4257\(00\)00197-8](http://dx.doi.org/10.1016/S0034-4257(00)00197-8)
- [14] P. Murray and S. Marshall, “A new design tool for feature extraction in noisy images based on grayscale hit-or-miss transforms,” *IEEE Transactions on Image Processing*, vol. 20, no. 7, pp. 1938–1948, Jul 2011. [Online]. Available: <http://dx.doi.org/10.1109/tip.2010.2103952>
- [15] International Agency for Research on Cancer *et al.*, *IARC Monographs on the Evaluation of Carcinogenic Risks to Humans. V. 55: Solar and Ultraviolet Radiation*. Lyon, France: IARC, 1992, vol. 55.
- [16] C. Starr, C. Evers, and L. Starr, *Biology: Concepts and applications without physiology*. Belmont, CA: Wadsworth, 2010.
- [17] Innospec, *Blue Eye Datasheet*, 2016. [Online]. Available: http://52ebad10ee97eea25d5e-d7d40819259e7d3022d9ad53e3694148.r84.cf3.rackcdn.com/UK_INNO-SPEC_BlueEye-UV_DS_CAT-15.pdf
- [18] Headwall, *Hyperspec UV imaging sensor for the 250-500nm spectral range*, 2015. [Online]. Available: <http://cdn2.hubspot.net/hubfs/145999/docs/UV-VIS.pdf?t=1480688133439>
- [19] Headwall, *Hyperspec VNIR imaging sensors for the 380-1000nm spectral range*, 2016. [Online]. Available: <http://cdn2.hubspot.net/hubfs/145999/docs/VNIR.pdf?t=1480688133439>
- [20] HySpex, *HySpex VNIR Cameras Datasheets*, 2016. [Online]. Available: http://www.hyspex.no/products/vnir_1800.php
- [21] Innospec, *Green Eye Datasheet*, 2016. [Online]. Available: http://52ebad10ee97eea25d5e-d7d40819259e7d3022d9ad53e3694148.r84.cf3.rackcdn.com/UK_INNO-SPEC_GreenEye-VIS%20_DS_CAT-15.pdf

- [22] Headwall, *Hyperspec NIR imaging sensor for the 900nm to 1700 nm spectral range*, 2016. [Online]. Available: <http://cdn2.hubspot.net/hubfs/145999/docs/NIR.pdf?t=1480688133439>
- [23] Innospec, *Red Eye 1.7 Datasheet*, 2016. [Online]. Available: http://www.inno-spec.de/fileadmin/user_upload/documents/Spectral_Imaging_System-NIR-1-7_01.pdf
- [24] Headwall, *Hyperspec SWIR imaging sensor for the 900nm to 2500nm spectral range*, 2015. [Online]. Available: <http://cdn2.hubspot.net/hubfs/145999/docs/SWIR.pdf?t=1480688133439>
- [25] HySpex, *HySpex SWIR Cameras Datasheets*, 2016. [Online]. Available: http://www.hyspex.no/products/swir_384.php
- [26] Innospec, *Red Eye 2.2 Datasheet*, 2016. [Online]. Available: http://www.inno-spec.de/fileadmin/user_upload/documents/Spectral_Imaging_System-NIR-2-2.pdf
- [27] Innospec, *Black Eye Datasheet*, 2016. [Online]. Available: http://52ebad10ee97eea25d5e-d7d40819259e7d3022d9ad53e3694148.r84.cf3.rackcdn.com/UK_INNO-SPEC_BlackEye-MIR_DS_CAT-15.pdf
- [28] Headwall, *Imaging Solutions*, 2016. [Online]. Available: <http://www.headwallphotonics.com/spectral-imaging>
- [29] Telops, *Hyper-Cam Datasheets*, 2016. [Online]. Available: http://telops.com/products/hyperspectral-cameras/item/download/59_05764d2882c26a890ef2574fb2e8bf03
- [30] D. Manolakis, C. Siracusa, and G. Shaw, "Hyperspectral subpixel target detection using the linear mixing model," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 7, pp. 1392–1409, Jul 2001. [Online]. Available: <http://dx.doi.org/10.1109/36.934072>

- [31] B. Hapke, "Bidirectional reflectance spectroscopy: 1. theory," *Journal of Geophysical Research: Solid Earth*, vol. 86, no. B4, pp. 3039–3054, Apr 1981. [Online]. Available: <http://dx.doi.org/10.1029/JB086iB04p03039>
- [32] D. Roberts, J. Adams, and M. Smith, "Transmission and scattering of light by leaves: Effects on spectral mixtures," in *10th Annual International Symposium on Geoscience and Remote Sensing*. IEEE, 1990, pp. 1381–1384. [Online]. Available: <http://dx.doi.org/10.1109/igarss.1990.688757>
- [33] M. M. Verstraete, B. Pinty, and R. E. Dickinson, "A physical model of the bidirectional reflectance of vegetation canopies: 1. theory," *Journal of Geophysical Research*, vol. 95, no. D8, pp. 11 755–11 765, 1990. [Online]. Available: <http://dx.doi.org/10.1029/jd095id08p11755>
- [34] J. Iaquinta, B. Pinty, and J. L. Privette, "Inversion of a physically based bidirectional reflectance model of vegetation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 35, no. 3, pp. 687–698, May 1997. [Online]. Available: <http://dx.doi.org/10.1109/36.581988>
- [35] G. P. Asner, A. R. Townsend, and B. H. Braswell, "Satellite observation of el niño effects on amazon forest phenology and productivity," *Geophysical Research Letters*, vol. 27, no. 7, pp. 981–984, Apr 2000. [Online]. Available: <http://dx.doi.org/10.1029/1999gl011113>
- [36] B. Hapke, *Theory of Reflectance and Emittance Spectroscopy*. Cambridge University Press (CUP), 2012. [Online]. Available: <http://dx.doi.org/10.1017/cbo9781139025683>
- [37] R. Heylen and P. Scheunders, "A multilinear mixing model for nonlinear spectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 1, pp. 240–251, Jan 2016. [Online]. Available: <http://dx.doi.org/10.1109/tgrs.2015.2453915>

- [38] N. Keshava and J. F. Mustard, "Spectral unmixing," *IEEE Signal Processing Magazine*, vol. 19, no. 1, pp. 44–57, Jan 2002. [Online]. Available: <http://dx.doi.org/10.1109/79.974727>
- [39] Y. Liang, J. Li, and K. Guo, "Lossless compression of hyperspectral images using hybrid context prediction," *Optics Express*, vol. 20, no. 7, pp. 8199–8206, Mar 2012. [Online]. Available: <http://dx.doi.org/10.1364/oe.20.008199>
- [40] J. Zhang, H. Li, and C. W. Chen, "Distributed lossless coding techniques for hyperspectral images," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 6, pp. 977–989, Sep 2015. [Online]. Available: <http://dx.doi.org/10.1109/JSTSP.2015.2402118>
- [41] J. Wu, W. Kong, J. Mielikainen, and B. Huang, "Lossless compression of hyperspectral imagery via clustered differential pulse code modulation with removal of local spectral outliers," *IEEE Signal Processing Letters*, vol. 22, no. 12, pp. 2194–2198, Dec 2015. [Online]. Available: <http://dx.doi.org/10.1109/lsp.2015.2443913>
- [42] J. P. Pluim, J. A. Maintz, and M. A. Viergever, "Mutual-information-based registration of medical images: a survey," *IEEE Transactions on Medical Imaging*, vol. 22, no. 8, pp. 986–1004, Aug 2003. [Online]. Available: <http://dx.doi.org/10.1109/TMI.2003.815867>
- [43] A. Martínez-Usó, F. Pla, J. M. Sotoca, and P. García-Sevilla, "Clustering-based hyperspectral band selection using information measures," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 12, pp. 4158–4171, Dec 2007. [Online]. Available: <http://dx.doi.org/10.1109/tgrs.2007.904951>
- [44] J. Feng, L. Jiao, X. Zhang, and T. Sun, "Hyperspectral band selection based on trivariate mutual information and clonal selection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 7, pp. 4092–4105, Jul 2014. [Online]. Available: <http://dx.doi.org/10.1109/tgrs.2013.2279591>

- [45] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, Jul 1948. [Online]. Available: <http://dx.doi.org/10.1002/j.1538-7305.1948.tb01338.x>
- [46] N. Keshava, "Best bands selection for detection in hyperspectral processing," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings*, vol. 5. Salt Lake City, UT, US: IEEE, 2001, pp. 3149–3152. [Online]. Available: <http://dx.doi.org/10.1109/icassp.2001.940326>
- [47] J. Boardman, "Spectral angle mapping: a rapid measure of spectral similarity," *AVIRIS*, 1993.
- [48] A. D. Vibhute, K. Kale, R. K. Dhumal, and S. Mehrotra, "Hyperspectral imaging data atmospheric correction challenges and solutions using QUAC and FLAASH algorithms," in *2015 IEEE International Conference on Man and Machine Interfacing*. Odisha, India: IEEE, Dec 2015, pp. 1–6. [Online]. Available: <http://dx.doi.org/10.1109/mami.2015.7456604>
- [49] T.-M. Tu, C.-H. Chen, J.-L. Wu, and C.-I. Chang, "A fast two-stage classification method for high-dimensional remote sensing data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 36, no. 1, pp. 182–191, 1998. [Online]. Available: <http://dx.doi.org/10.1109/36.655328>
- [50] J. Rouse Jr, R. Haas, J. Schell, and D. Deering, "Monitoring vegetation systems in the great plains with ERTS," *Third Earth Resources Technology Satellite-1 Symposium, Washington, DC, US*, vol. 351, p. 309, 1974.
- [51] S. P. Boyte, B. K. Wylie, D. J. Major, and J. F. Brown, "The integration of geophysical and enhanced moderate resolution imaging spectroradiometer normalized difference vegetation index data into a rule-based, piecewise regression-tree model to estimate cheatgrass beginning of spring growth," *International Journal of Digital Earth*, vol. 8, no. 2, pp. 118–132, Nov 2013. [Online]. Available: <http://dx.doi.org/10.1080/17538947.2013.860196>

- [52] D. Roy, V. Kovalsky, H. Zhang, E. Vermote, L. Yan, S. Kumar, and A. Egorov, "Characterization of Landsat-7 to Landsat-8 reflective wavelength and normalized difference vegetation index continuity," *Remote Sensing of Environment*, vol. 185, pp. 57–70, Nov 2016. [Online]. Available: <http://dx.doi.org/10.1016/j.rse.2015.12.024>
- [53] Z. Gong, K. Kawamura, N. Ishikawa, M. Goto, T. Wulan, D. Alateng, T. Yin, and Y. Ito, "MODIS normalized difference vegetation index (NDVI) and vegetation phenology dynamics in the inner Mongolia grassland," *Solid Earth*, vol. 6, no. 4, pp. 1185–1194, Nov 2015. [Online]. Available: <http://dx.doi.org/10.5194/se-6-1185-2015>
- [54] M. Bochow, B. Heim, T. Küster, C. Rogaß, I. Bartsch, K. Segl, and H. Kaufmann, "Automatic detection and delineation of surface water bodies in airborne hyperspectral data," in *2012 IEEE International Geoscience and Remote Sensing Symposium*. Munich, Germany: IEEE, Jul 2012, pp. 5226–5229. [Online]. Available: <http://dx.doi.org/10.1109/igarss.2012.6352431>
- [55] Airborne Visible / Infrared Imaging Spectrometer (AVIRIS), *AVIRIS Data Portal 2006-Present*, 2016. [Online]. Available: http://aviris.jpl.nasa.gov/alt_locator/
- [56] SpecTIR, *Free Data Samples*, 2014. [Online]. Available: <http://www.spectir.com/free-data-samples/>
- [57] M. Bochow, C. Rogaß, K. Segl, and H. Kaufmann, "Automatic shadow detection in hyperspectral VIS-NIR images," in *Proceedings of the 7th EARSeL Workshop of the Special Interest Group in Imaging Spectroscopy*, Edinburgh, UK, Apr 2011.
- [58] M. K. Griffin and H. K. Burke, "Compensation of hyperspectral data for atmospheric effects," *Lincoln Laboratory Journal*, vol. 14, no. 1, pp. 29–54, 2003.
- [59] R. Miller and I. Tegen, "Climate response to soil dust aerosols," *Journal of Climate*, vol. 11, no. 12, pp. 3247–3267, Dec 1998. [Online]. Available: [http://dx.doi.org/10.1175/1520-0442\(1998\)011<3247:crtsta>2.0.co;2](http://dx.doi.org/10.1175/1520-0442(1998)011<3247:crtsta>2.0.co;2)

- [60] I. Sokolik, D. Winker, G. Bergametti, D. Gillette, G. Carmichael, Y. Kaufman, L. Gomes, L. Schuetz, and J. Penner, "Introduction to special section: Outstanding problems in quantifying the radiative impacts of mineral dust," *Journal of Geophysical Research: Atmospheres*, vol. 106, no. D16, pp. 18 015–18 027, Aug 2001. [Online]. Available: <http://dx.doi.org/10.1029/2000jd900498>
- [61] J. Hansen, M. Sato, and R. Ruedy, "Radiative forcing and climate response," *Journal of Geophysical Research: Atmospheres*, vol. 102, no. D6, pp. 6831–6864, Mar 1997. [Online]. Available: <http://dx.doi.org/10.1029/96jd03436>
- [62] J. F. B. Mitchell, T. C. Johns, J. M. Gregory, and S. F. B. Tett, "Climate response to increasing levels of greenhouse gases and sulphate aerosols," *Nature*, vol. 376, no. 6540, pp. 501–504, aug 1995. [Online]. Available: <http://dx.doi.org/10.1038/376501a0>
- [63] P. S. Chavez, "An improved dark-object subtraction technique for atmospheric scattering correction of multispectral data," *Remote Sensing of Environment*, vol. 24, no. 3, pp. 459–479, Apr 1988. [Online]. Available: [http://dx.doi.org/10.1016/0034-4257\(88\)90019-3](http://dx.doi.org/10.1016/0034-4257(88)90019-3)
- [64] F. A. Kruse, "Use of airborne imaging spectrometer data to map minerals associated with hydrothermally altered rocks in the northern grapevine mountains, Nevada, and California," *Remote Sensing of Environment*, vol. 24, no. 1, pp. 31–51, Feb 1988. [Online]. Available: [http://dx.doi.org/10.1016/0034-4257\(88\)90004-1](http://dx.doi.org/10.1016/0034-4257(88)90004-1)
- [65] D. Roberts, Y. Yamaguchi, and R. Lyon, "Comparison of various techniques for calibration of AIS data," *Proceedings 2nd AIS Workshop JPL Publication 86-35 Jet Propulsion Laboratory, Pasadena, CA*, vol. 87, 1986.
- [66] J. E. Conel, R. O. Green, G. Vane, C. J. Bruegge, R. E. Alley, and B. J. Curtiss, "Airborne imaging spectrometer-2: Radiometric spectral characteristics and comparison of ways to compensate for the atmosphere," in *Imaging Spectroscopy*

- II. San Diego, CA, US: International Society for Optics and Photonics, Jan 1987, pp. 140–157. [Online]. Available: <http://dx.doi.org/10.1117/12.942294>
- [67] A. Green and M. Craig, “Analysis of aircraft spectrometer data with logarithmic residuals,” in *Proceedings, AIS workshop, JPL Publication 85-41*, Pasadena, CA, US, 1985, pp. 111–119.
- [68] B.-C. Gao, K. B. Heidebrecht, and A. F. Goetz, “Derivation of scaled surface reflectances from AVIRIS data,” *Remote Sensing of Environment*, vol. 44, no. 2-3, pp. 165–178, May 1993. [Online]. Available: [http://dx.doi.org/10.1016/0034-4257\(93\)90014-o](http://dx.doi.org/10.1016/0034-4257(93)90014-o)
- [69] Z. Qu, B. C. Kindel, and A. F. Goetz, “The high accuracy atmospheric correction for hyperspectral data (HATCH) model,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 6, pp. 1223–1231, Jun 2003. [Online]. Available: <http://dx.doi.org/10.1109/tgrs.2003.813125>
- [70] F. Kruse, “Comparison of ATREM, ACORN, and FLAASH atmospheric corrections using low-altitude AVIRIS data of Boulder, CO,” in *Summaries of 13th JPL Airborne Geoscience Workshop*, Pasadena, CA, US, 2004.
- [71] R. Richter, “Atmospheric correction of satellite data with haze removal including a haze/clear transition region,” *Computers & Geosciences*, vol. 22, no. 6, pp. 675–681, Jul 1996. [Online]. Available: [http://dx.doi.org/10.1016/0098-3004\(96\)00010-6](http://dx.doi.org/10.1016/0098-3004(96)00010-6)
- [72] S. M. Adler-Golden, M. W. Matthew, L. S. Bernstein, R. Y. Levine, A. Berk, S. C. Richtsmeier, P. K. Acharya, G. P. Anderson, J. W. Felde, J. Gardner *et al.*, “Atmospheric correction for shortwave spectral imagery based on MODTRAN4,” in *Imaging Spectrometry V*. Denver, CO, US: International Society for Optics and Photonics, Oct 1999, pp. 61–69. [Online]. Available: <http://dx.doi.org/10.1117/12.366315>
- [73] A. Berk, G. P. Anderson, P. K. Acharya, L. S. Bernstein, L. Muratov, J. Lee, M. Fox, S. M. Adler-Golden, J. H. Chetwynd, M. L. Hoke *et al.*, “MODTRAN

- 5: A reformulated atmospheric band model with auxiliary species and practical multiple scattering options,” in *Multispectral and Hyperspectral Remote Sensing Instruments and Applications II*. Honolulu, HI, US: International Society for Optics and Photonics, Jan 2005, pp. 662–667. [Online]. Available: <http://dx.doi.org/10.1117/12.578758>
- [74] G. Vane and A. F. Goetz, “Terrestrial imaging spectroscopy,” *Remote Sensing of Environment*, vol. 24, no. 1, pp. 1–29, Feb 1988. [Online]. Available: [http://dx.doi.org/10.1016/0034-4257\(88\)90003-x](http://dx.doi.org/10.1016/0034-4257(88)90003-x)
- [75] M. J. Smith, “A comparison of dg acomp, flaash and quac atmospheric compensation algorithms using worldview-2 imagery,” Master’s thesis, University of Colorado, Boulder, Colorado, 2015.
- [76] J. H. Gruninger, A. J. Ratkowski, and M. L. Hoke, “The sequential maximum angle convex cone (SMACC) endmember model,” in *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery X*. Anaheim, CA, US: International Society for Optics and Photonics, aug 2004, pp. 1–14. [Online]. Available: <http://dx.doi.org/10.1117/12.543794>
- [77] A. Ifarraguerri and C.-I. Chang, “Multispectral and hyperspectral image analysis with convex cones,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 2, pp. 756–770, Mar 1999. [Online]. Available: <http://dx.doi.org/10.1109/36.752192>
- [78] F. Chaudhry, C.-C. Wu, W. Liu, C.-I. Chang, and A. Plaza, “Pixel purity index-based algorithms for endmember extraction from hyperspectral imagery,” *Recent Advances in Hyperspectral Signal and Image Processing*, vol. 37, no. 2, pp. 29–62, 2006.
- [79] J. M. Nascimento and J. M. Dias, “Vertex component analysis: A fast algorithm to unmix hyperspectral data,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 4, pp. 898–910, Apr 2005. [Online]. Available: <http://dx.doi.org/10.1109/tgrs.2005.844293>

- [80] C.-I. Chang, C.-C. Wu, W. Liu, and Y.-C. Ouyang, "A new growing method for simplex-based endmember extraction algorithm," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 10, pp. 2804–2819, Oct 2006. [Online]. Available: <http://dx.doi.org/10.1109/tgrs.2006.881803>
- [81] K. Lee, "A subpixel scale target detection algorithm for hyperspectral imagery," Ph.D. dissertation, Rochester Institute of Technology, 2003.
- [82] I. S. Reed and X. Yu, "Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 10, pp. 1760–1770, 1990. [Online]. Available: <http://dx.doi.org/10.1109/29.60107>
- [83] J. A. Richards, *Remote Sensing Digital Image Analysis*. Springer-Verlag GmbH, 2012, vol. 3. [Online]. Available: http://www.ebook.de/de/product/18922470/john_a_richards_remote_sensing_digital_image_analysis.html
- [84] P. C. Mahalanobis, "On the generalized distance in statistics," *Proceedings of the National Institute of Sciences of India*, vol. 2, pp. 49–55, 1936.
- [85] H. Mark, "Chemometrics in near-infrared spectroscopy," *Analytica Chimica Acta*, vol. 223, pp. 75–93, 1989. [Online]. Available: [http://dx.doi.org/10.1016/s0003-2670\(00\)84075-1](http://dx.doi.org/10.1016/s0003-2670(00)84075-1)
- [86] Y. Xu, Z. Wu, F. Xiao, T. Zhan, and Z. Wei, "A target detection method based on low-rank regularized least squares model for hyperspectral images," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 8, pp. 1129–1133, Aug 2016. [Online]. Available: <http://dx.doi.org/10.1109/lgrs.2016.2572090>
- [87] H. Ren and C.-I. Chang, "Automatic spectral target recognition in hyperspectral imagery," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1232–1249, Oct 2003. [Online]. Available: <http://dx.doi.org/10.1109/taes.2003.1261124>

- [88] S. Bernabe, S. López, A. Plaza, and R. Sarmiento, "GPU implementation of an automatic target detection and classification algorithm for hyperspectral image analysis," *IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 2, pp. 221–225, Mar 2013. [Online]. Available: <http://dx.doi.org/10.1109/lgrs.2012.2198790>
- [89] J. C. Harsanyi and C.-I. Chang, "Hyperspectral image classification and dimensionality reduction: an orthogonal subspace projection approach," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 32, no. 4, pp. 779–785, Jul 1994. [Online]. Available: <http://dx.doi.org/10.1109/36.298007>
- [90] R. Mayer, F. Bucholtz, and D. Scribner, "Object detection by using "whitening/dewhitening" to transform target signatures in multitemporal hyperspectral and multispectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 5, pp. 1136–1142, May 2003. [Online]. Available: <http://dx.doi.org/10.1109/tgrs.2003.813553>
- [91] X. Zhang, Y. Sun, K. Shang, L. Zhang, and S. Wang, "Crop classification based on feature band set construction and object-oriented approach using hyperspectral images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 9, pp. 4117–4128, Sep 2016. [Online]. Available: <http://dx.doi.org/10.1109/jstars.2016.2577339>
- [92] H. Zheng, X. Zhou, T. Cheng, X. Yao, Y. Tian, W. Cao, and Y. Zhu, "Evaluation of a UAV-based hyperspectral frame camera for monitoring the leaf nitrogen concentration in rice," in *2016 IEEE International Geoscience and Remote Sensing Symposium*. Beijing, China: IEEE, jul 2016, pp. 7350–7353. [Online]. Available: <http://dx.doi.org/10.1109/igarss.2016.7730917>
- [93] A. M. Abuleil, G. W. Taylor, and M. Moussa, "An integrated system for mapping red clover ground cover using unmanned aerial vehicles: A case study in precision agriculture," in *2015 IEEE Conference on Computer and Robot Vision*. Halifax, Nova Scotia, Canada: IEEE, jun 2015, pp. 277–284. [Online]. Available: <http://dx.doi.org/10.1109/crv.2015.43>

- [94] R. M. Haralick, K. Shanmugam *et al.*, “Textural features for image classification,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 6, pp. 610–621, Nov 1973. [Online]. Available: <http://dx.doi.org/10.1109/tsmc.1973.4309314>
- [95] T. Kutser, B. Paavel, C. Verpoorter, M. Ligi, T. Soomets, K. Toming, and G. Casal, “Remote sensing of black lakes and using 810 nm reflectance peak for retrieving water quality parameters of optically complex waters,” *Remote Sensing*, vol. 8, no. 6, p. 497, Jun 2016. [Online]. Available: <http://dx.doi.org/10.3390/rs8060497>
- [96] W. Ge, Q. Cheng, L. Jing, Y. Chen, X. Guo, H. Ding, and Q. Liu, “Mineral mapping in the western Kunlun mountains using Tiangong-1 hyperspectral imagery,” in *IOP Conference Series: Earth and Environmental Science*, vol. 34, no. 1. Halifax, Canada: IOP Publishing, Apr 2016, p. 012011. [Online]. Available: <http://dx.doi.org/10.1088/1755-1315/34/1/012011>
- [97] A. S. Mazer, M. Martin, M. Lee, and J. E. Solomon, “Image processing software for imaging spectrometry data analysis,” *Remote Sensing of Environment*, vol. 24, no. 1, pp. 201–210, feb 1988. [Online]. Available: [http://dx.doi.org/10.1016/0034-4257\(88\)90012-0](http://dx.doi.org/10.1016/0034-4257(88)90012-0)
- [98] F. Kruse, A. Lefkoff, J. Boardman, K. Heidebrecht, A. Shapiro, P. Barloon, and A. Goetz, “The spectral image processing system – interactive visualization and analysis of imaging spectrometer data,” *Remote Sensing of Environment*, vol. 44, no. 2, pp. 145–163, 1993. [Online]. Available: <http://dx.doi.org/10.1063/1.44433>
- [99] R. N. Clark, A. J. Gallagher, and G. A. Swayze, “Material absorption band depth mapping of imaging spectrometer data using a complete band shape least-squares fit with library reference spectra,” in *Proceedings of the Second Airborne Visible/Infrared Imaging Spectrometer Workshop*. Pasadena, CA, US: JPL Publication, 1990, pp. 176–186.
- [100] R. Clark, G. Swayze, A. Gallagher, N. Gorelick, and F. Kruse, “Mapping with imaging spectrometer data using the complete band shape least-squares algorithm

- simultaneously fit to multiple spectral features from multiple materials,” in *Proceedings of the Third Airborne Visible/Infrared Imaging Spectrometer Workshop*, vol. 42, Pasadena, CA, US, 1991, pp. 2–3.
- [101] J. Crowley and R. Clark, “AVIRIS study of death valley evaporite deposits using least-squares band-fitting methods,” in *Summaries of the Third Annual JPL Airborne Geoscience Workshop*. JPL Publication, 1992, pp. 29–31.
- [102] G. A. Swayze and R. N. Clark, “Spectral identification of minerals using imaging spectrometry data: evaluating the effects of signal to noise and spectral resolution using the tricorder algorithm,” in *Summaries of the 5th Annual JPL Airborne Earth Science Workshop*, Pasadena, CA, US, 1995, pp. 95–1.
- [103] K. Staenz, J. Schwarz, L. Vernaccini, F. Vachon, and C. Nadeau, “Classification of hyperspectral agricultural data with spectral matching techniques,” in *Proceedings of the International Symposium on Spectral Sensing Research*, Las Vegas, NV, US, 1999, p. 12.
- [104] S. A. Robila and A. Gershman, “Spectral matching accuracy in processing hyperspectral data,” in *International Symposium on Signals, Circuits and Systems, 2005*, vol. 1. Iasi, Romania: IEEE, 2005, pp. 163–166. [Online]. Available: <http://dx.doi.org/10.1109/isscs.2005.1509878>
- [105] H. Li, W. S. Lee, K. Wang, R. Ehsani, and C. Yang, “Extended spectral angle mapping for citrus greening disease detection using airborne hyperspectral imaging,” *Precision Agriculture*, vol. 15, no. 2, pp. 162–183, Jul 2013. [Online]. Available: <http://dx.doi.org/10.1007/s11119-013-9325-6>
- [106] L. L. Scharf and L. T. McWhorter, “Adaptive matched subspace detectors and adaptive coherence estimators,” in *Conference Record of The Thirtieth Asilomar Conference on Signals, Systems and Computers*. Pacific Grove, CA, US: IEEE, 1996, pp. 1114–1117. [Online]. Available: <http://dx.doi.org/10.1109/acssc.1996.599116>

- [107] D. W. Stein, S. G. Beaven, L. E. Hoff, E. M. Winter, A. P. Schaum, and A. D. Stocker, "Anomaly detection from hyperspectral imagery," *IEEE Signal Processing Magazine*, vol. 19, no. 1, pp. 58–69, 2002. [Online]. Available: <http://dx.doi.org/10.1109/79.974730>
- [108] S. Matteoli, M. Diani, and G. Corsini, "A tutorial overview of anomaly detection in hyperspectral images," *IEEE Aerospace and Electronic Systems Magazine*, vol. 25, no. 7, pp. 5–28, Jul 2010. [Online]. Available: <http://dx.doi.org/10.1109/maes.2010.5546306>
- [109] J. M. Nascimento and J. M. Bioucas-Dias, "Nonlinear mixture model for hyperspectral unmixing," in *Image and Signal Processing for Remote Sensing XV*. International Society for Optics and Photonics, sep 2009, pp. 74 770I–74 770I. [Online]. Available: <http://dx.doi.org/10.1117/12.830492>
- [110] A. Halimi, Y. Altmann, N. Dobigeon, and J.-Y. Tournet, "Nonlinear unmixing of hyperspectral images using a generalized bilinear model," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 11, pp. 4153–4162, Nov 2011. [Online]. Available: <http://dx.doi.org/10.1109/TGRS.2010.2098414>
- [111] Y. Altmann, A. Halimi, N. Dobigeon, and J.-Y. Tournet, "Supervised nonlinear spectral unmixing using a postnonlinear mixing model for hyperspectral imagery," *IEEE Transactions on Image Processing*, vol. 21, no. 6, pp. 3017–3025, Jun 2012. [Online]. Available: <http://dx.doi.org/10.1109/tip.2012.2187668>
- [112] D. Manolakis, E. Truslow, M. Pieper, T. Cooley, and M. Brueggeman, "Detection algorithms in hyperspectral imaging systems: An overview of practical algorithms," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 24–33, Jan 2014. [Online]. Available: <http://dx.doi.org/10.1109/msp.2013.2278915>
- [113] N. M. Nasrabadi, "Hyperspectral target detection : An overview of current and future challenges," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 34–44, Jan 2014. [Online]. Available: <http://dx.doi.org/10.1109/msp.2013.2278992>

- [114] Y. Xu, Z. Wu, J. Li, A. Plaza, and Z. Wei, "Anomaly detection in hyperspectral images based on low-rank and sparse representation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 4, pp. 1990–2000, 2016.
- [115] The MathWorks Inc, *Signal Processing Toolbox, R2016b*, 2016. [Online]. Available: https://uk.mathworks.com/help/pdf_doc/signal/signal_tb.pdf
- [116] S. Matteoli, M. Diani, and G. Corsini, "Improved estimation of local background covariance matrix for anomaly detection in hyperspectral images," *Optical Engineering*, vol. 49, no. 4, p. 046201, Apr 2010. [Online]. Available: <http://dx.doi.org/10.1117/1.3386069>
- [117] D. C. Hoaglin, F. Mosteller, and J. W. Tukey, *Understanding Robust and Exploratory Data Analysis*. John Wiley & Sons, 1983, vol. 3. [Online]. Available: http://www.ebook.de/de/product/3602313/understanding_robust_and_exploratory_data_analysis.html
- [118] D. H. Ballard, "Generalizing the hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111–122, Jan 1981. [Online]. Available: [http://dx.doi.org/10.1016/0031-3203\(81\)90009-1](http://dx.doi.org/10.1016/0031-3203(81)90009-1)
- [119] G. Matheron, *Random Sets and Integral Geometry*. John Wiley & Sons, 1975.
- [120] J. Serra, "Introduction to mathematical morphology," *Computer Vision, Graphics, and Image Processing*, vol. 35, no. 3, pp. 283–305, Sep 1986. [Online]. Available: [http://dx.doi.org/10.1016/0734-189x\(86\)90002-2](http://dx.doi.org/10.1016/0734-189x(86)90002-2)
- [121] H. J. Heijmans, "Morphological image operators," *Advances in Electronics and Electron Physics Suppl., Boston: Academic Press, c1994*, vol. 1, 1994.
- [122] P. Murray and S. Marshall, "A review of recent advances in the hit-or-miss transform," *Advances in Imaging and Electron Physics*, vol. 175, pp. 221–282, 2013. [Online]. Available: <http://dx.doi.org/10.1016/b978-0-12-407670-9.00005-6>

- [123] M. Khosravi and R. W. Schafer, "Template matching based on a grayscale hit-or-miss transform," *IEEE Transactions on Image Processing*, vol. 5, no. 6, pp. 1060–1066, Jun 1996. [Online]. Available: <http://dx.doi.org/10.1109/83.503921>
- [124] C. Ronse, "A lattice-theoretical morphological view on template extraction in images," *Journal of Visual Communication and Image Representation*, vol. 7, no. 3, pp. 273–295, Sep 1996. [Online]. Available: <http://dx.doi.org/10.1006/jvci.1996.0024>
- [125] B. Raducanu and M. Grana, "A grayscale hit-or-miss transform based on level sets," in *Proceedings 2000 International Conference on Image Processing*, vol. 2. Vancouver, British Columbia, Canada: IEEE, 2000, pp. 931–933. [Online]. Available: <http://dx.doi.org/10.1109/icip.2000.899869>
- [126] D. S. Bloomberg and L. Vincent, "Pattern matching using the blur hit-miss transform," *Journal of Electronic Imaging*, vol. 9, no. 2, pp. 140–150, Apr 2000. [Online]. Available: <http://dx.doi.org/10.1117/1.482734>
- [127] P. Soille, "Advances in the analysis of topographic features on discrete images," in *Discrete Geometry for Computer Imagery*, Springer. Bordeaux, France: Springer Nature, 2002, pp. 175–186. [Online]. Available: http://dx.doi.org/10.1007/3-540-45986-3_16
- [128] C. Barat, C. Ducottet *et al.*, "Pattern matching using morphological probing," in *Proceedings 2003 International Conference on Image Processing*, vol. 1. Barcelona, Spain: IEEE, 2003, pp. I–369. [Online]. Available: <http://dx.doi.org/10.1109/icip.2003.1246975>
- [129] B. Naegel, N. Passat, and C. Ronse, "Grey-level hit-or-miss transformsâ€”part i: unified theory," *Pattern Recognition*, vol. 40, no. 2, pp. 635–647, Feb 2007. [Online]. Available: <http://dx.doi.org/10.1016/j.patcog.2006.06.004>
- [130] S. Lefèvre, J. Weber, and D. Sheeren, "Automatic building extraction in VHR images using advanced morphological operators," in *2007 IEEE Urban Remote*

- Sensing Joint Event*. Paris, France: IEEE, apr 2007, pp. 1–5. [Online]. Available: <http://dx.doi.org/10.1109/urs.2007.371825>
- [131] B. Perret, S. Lefevre, and C. Collet, “A robust hit-or-miss transform for template matching applied to very noisy astronomical images,” *Pattern Recognition*, vol. 42, no. 11, pp. 2470–2480, Nov 2009. [Online]. Available: <http://dx.doi.org/10.1016/j.patcog.2009.02.013>
- [132] P. Murray, S. Marshall, and E. Bullinger, “The percentage occupancy hit or miss transform,” in *Signal Processing Conference, 2009 17th European*. Glasgow, UK: IEEE, 2009, pp. 253–257.
- [133] K. Pearson, “On lines and planes of closest fit to systems of point in space,” *Philosophical Magazine*, vol. 2, pp. 559–572, 1901.
- [134] I. Jolliffe, *Principal Component Analysis*. Wiley Online Library, 2002.
- [135] F. Kriegler, W. Malila, R. Nalepka, and W. Richardson, “Preprocessing transformations and their effects on multispectral recognition,” in *Proceedings of the Sixth International Symposium on Remote Sensing of the Environment*, vol. 1, Ann Arbor, MI, US, 1969, p. 97.
- [136] B.-C. Gao, M. J. Montes, C. O. Davis, and A. F. Goetz, “Atmospheric correction algorithms for hyperspectral remote sensing data of land and ocean,” *Remote Sensing of Environment*, vol. 113, pp. S17–S24, Sep 2009. [Online]. Available: <http://dx.doi.org/10.1016/j.rse.2007.12.015>
- [137] J. P. Kerekes and J. E. Baum, “Hyperspectral imaging system modeling,” *Lincoln Laboratory Journal*, vol. 14, no. 1, pp. 117–130, 2003.
- [138] Ontar Corporation, *PcModWin 5.3 Overview*, 2015. [Online]. Available: <http://www.ontar.com/Software/ProductDetails.aspx?item=PcModWin>
- [139] The MathWorks Inc, *MATLAB, version 8.6 (R2015b)*, 2015. [Online]. Available: <https://uk.mathworks.com/help/releases/R2015b/index.html>

- [140] Harris Geospatial Solutions, *Fast Line-of-sight Atmospheric Analysis of Hypercubes (FLAASH)*, 2016. [Online]. Available: <http://www.harrisgeospatial.com/docs/FLAASH.html>
- [141] L. Abreu and G. Anderson, “The MODTRAN 2/3 report and LOWTRAN 7 model,” *Contract*, vol. 19628, no. 91-C, p. 0132, 1996.
- [142] D. Griffith, *Matlab Class Wrapper for MODTRAN 4*, 2010. [Online]. Available: <https://uk.mathworks.com/matlabcentral/fileexchange/26927-matlab-class-wrapper-for-modtran-4>
- [143] D. Griffith, *Matlab Class Wrapper for MODTRAN 5*, 2011. [Online]. Available: <https://uk.mathworks.com/matlabcentral/fileexchange/31961-derekjgriffith-matlab-modtran-5>
- [144] The MathWorks Inc, *App Building, R2016b*, 2016. [Online]. Available: https://uk.mathworks.com/help/pdf_doc/matlab/buildgui.pdf
- [145] T. Machida, K. Kita, Y. Kondo, D. Blake, S. Kawakami, G. Inoue, and T. Ogawa, “Vertical and meridional distributions of the atmospheric co2 mixing ratio between northern midlatitudes and southern subtropics,” *Journal of Geophysical Research*, vol. 108, no. D3, Nov 2002. [Online]. Available: <http://dx.doi.org/10.1029/2001jd000910>
- [146] R. N. Clark, G. A. Swayze, R. Wise, K. E. Livo, T. Hoefen, R. F. Kokaly, and S. J. Sutley, “USGS digital spectral library splib06a,” *US Geological Survey, Digital Data Series*, vol. 231, 2007.
- [147] United States Naval Observatory, *Earth’s Seasons: Equinoxes, Solstices, Perihelion, and Aphelion*, 2015. [Online]. Available: <http://aa.usno.navy.mil/data/docs/EarthSeasons.php>
- [148] C. A. MacCarley, B. M. Hemme, and L. Klein, “Evaluation of infrared and millimeter-wave imaging technologies applied to traffic management,” in *SAE*

- Technical Paper Series*. SAE International, Mar 2000, p. 1. [Online]. Available: <http://dx.doi.org/10.4271/2000-01-1303>
- [149] L. J. Denes, “High performance cameras for hyperspectral and polarimetric imaging research: to support advanced dod imaging technology initiatives,” *Carnegie Mellon University, Research Showcase, Pittsburgh, PA, US*, 2003.
- [150] S. G. Ungar, J. S. Pearlman, J. A. Mendenhall, and D. Reuter, “Overview of the earth observing one (EO-1) mission,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 6, pp. 1149–1159, Jun 2003. [Online]. Available: <http://dx.doi.org/10.1109/tgrs.2003.815999>
- [151] J. A. Curcio and C. C. Petty, “The near infrared absorption spectrum of liquid water,” *Journal of the Optical Society of America*, vol. 41, no. 5, pp. 302–304, May 1951. [Online]. Available: <http://dx.doi.org/10.1364/josa.41.000302>
- [152] S. S. Myers, A. Zanobetti, I. Kloog, P. Huybers, A. D. Leakey, A. Bloom, E. Carlisle, L. H. Dietterich, G. Fitzgerald, T. Hasegawa *et al.*, “Rising CO2 threatens human nutrition,” *Nature*, vol. 510, no. 7503, p. 139, 2014.
- [153] M. Kainuma, E. La Rovere, A. Matyssek, A. Rana, K. Riahi, R. Richels, S. Rose, D. Van Vuuren, R. Warren, P. Ambrosi *et al.*, “Issues related to mitigation in the long-term context,” *Climate Change*, p. 169250, 2007.
- [154] F. D. Van der Meer and S. M. De Jong, *Imaging Spectrometry: Basic Principles and Prospective Applications*. Springer Science & Business Media, 2011, vol. 4. [Online]. Available: http://www.ebook.de/de/product/3810304/imaging_spectrometry.html
- [155] D. Tanre, C. Deroo, P. Duhaut *et al.*, “The second simulation of the satellite signal in the solar spectrum (6s),” *User Guide. Laboratoire dOptique Atmospherique*, 1986.
- [156] J. Wang, *Data Warehousing and Mining: Concepts, Methodologies, Tools, and Applications*. Information Science Reference, 2008, vol. 3.

- [157] G. Marsh, "Meeting the challenge of wind turbine blade repair," *Reinforced Plastics*, vol. 55, no. 4, pp. 32–36, Jul 2011. [Online]. Available: [http://dx.doi.org/10.1016/s0034-3617\(11\)70112-6](http://dx.doi.org/10.1016/s0034-3617(11)70112-6)
- [158] D. Cripps, "The future of blade repair," *Reinforced Plastics*, vol. 55, no. 1, pp. 28–32, Jan 2011. [Online]. Available: [http://dx.doi.org/10.1016/s0034-3617\(11\)70034-0](http://dx.doi.org/10.1016/s0034-3617(11)70034-0)
- [159] J. Wallace and M. Dawson, "O&M strategies: Wind turbine blades," *Renewable Energy Focus*, vol. 10, no. 3, pp. 36–41, May 2009. [Online]. Available: [http://dx.doi.org/10.1016/s1755-0084\(09\)70119-x](http://dx.doi.org/10.1016/s1755-0084(09)70119-x)
- [160] G. Kocer, M. Mansour, N. Chokani, R. Abhari, and M. Müller, "Full-scale wind turbine near-wake measurements using an instrumented uninhabited aerial vehicle," *Journal of Solar Energy Engineering*, vol. 133, no. 4, p. 041011, 2011. [Online]. Available: <http://dx.doi.org/10.1115/1.4004707>
- [161] S. Nakauchi, K. Nishino, and T. Yamashita, "Selection of optimal combinations of band-pass filters for ice detection by hyperspectral imaging," *Optics Express*, vol. 20, no. 2, pp. 986–1000, Jan 2012. [Online]. Available: <http://dx.doi.org/10.1364/oe.20.000986>
- [162] H. Zhang and J. Jackman, "Feasibility of automatic detection of surface cracks in wind turbine blades," *Wind Engineering*, vol. 38, no. 6, pp. 575–586, Dec 2014. [Online]. Available: <http://dx.doi.org/10.1260/0309-524x.38.6.575>
- [163] H. Zhang and J. Jackman, "A feasibility study of wind turbine blade surface crack detection using an optical inspection method," in *2013 IEEE International Conference on Renewable Energy Research and Applications*. Madrid, Spain: IEEE, oct 2013, pp. 847–852. [Online]. Available: <http://dx.doi.org/10.1109/icrera.2013.6749869>
- [164] V. Nagarajan, J. S. Rajadurai *et al.*, "A digital image analysis to evaluate delamination factor for wind turbine composite laminate blade," *Composites*

- Part B: Engineering*, vol. 43, no. 8, pp. 3153–3159, Dec 2012. [Online]. Available: <http://dx.doi.org/10.1016/j.compositesb.2012.04.013>
- [165] M. Pieraccini, F. Parrini, M. Fratini, C. Atzeni, and P. Spinelli, “In-service testing of wind turbine towers using a microwave sensor,” *Renewable Energy*, vol. 33, no. 1, pp. 13–21, Jan 2008. [Online]. Available: <http://dx.doi.org/10.1016/j.renene.2007.02.001>
- [166] S. Melnyk, I. Tuluzov, and A. Melnyk, “Method of remote dynamic thermographic testing of wind turbine blades,” in *Proceedings of the 2014 International Conference on Quantitative InfraRed Thermography*, Bordeaux, France, 2014. [Online]. Available: <http://dx.doi.org/10.21611/qirt.2014.037>
- [167] C. C. Ciang, J.-R. Lee, and H.-J. Bang, “Structural health monitoring for a wind turbine system: a review of damage detection methods,” *Measurement Science and Technology*, vol. 19, no. 12, p. 122001, Oct 2008. [Online]. Available: <http://dx.doi.org/10.1088/0957-0233/19/12/122001>
- [168] DNV GL, *DNVGL-ST-0376 Rotor blades for wind turbines - Rules and standards*, 2015. [Online]. Available: <https://rules.dnvgl.com/docs/pdf/DNVGL/ST/2015-12/DNVGL-ST-0376.pdf>
- [169] M. Jureczko, M. Pawlak, and A. Mezyk, “Optimisation of wind turbine blades,” *Journal of Materials Processing Technology*, vol. 167, no. 2-3, pp. 463–471, Aug 2005. [Online]. Available: <http://dx.doi.org/10.1016/j.jmatprotec.2005.06.055>
- [170] P. Brøndsted, H. Lilholt, and A. Lystrup, “Composite materials for wind power turbine blades,” *Annual Review of Materials Research*, vol. 35, no. 1, pp. 505–538, Aug 2005. [Online]. Available: <http://dx.doi.org/10.1146/annurev.matsci.35.100303.110641>
- [171] B. F. Sørensen, E. Jørgensen, C. P. Debel, F. M. Jensen, H. M. Jensen, T. K. Jacobsen, and K. Halling, “Improved design of large wind turbine blade of fibre composites based on studies of scale effects (phase 1). summary report,” Risø National Laboratory, Denmark, Tech. Rep., 2004.

-
- [172] D. Rivkin and L. Silk, *Wind Turbine Operations, Maintenance, Diagnosis, and Repair*. Burlington, MA: Jones & Bartlett Publishers, 2012.
- [173] The MathWorks Inc, *Curve Fitting Toolbox, R2016b*, 2016. [Online]. Available: https://uk.mathworks.com/help/pdf_doc/curvefit/curvefit.pdf
- [174] E. H. Georgoulis, A. Iske, and J. Levesley, *Approximation Algorithms for Complex Systems*. Springer Berlin Heidelberg, 2009. [Online]. Available: <http://dx.doi.org/10.1007/978-3-642-16876-5>
- [175] Sequoiasci, *HydroLight*, 2016. [Online]. Available: <http://www.sequoiasci.com/product/hydrolight/>
- [176] A. Berk, G. Anderson, P. Acharya, and E. Shettle, “MODTRAN 5.2.1 user’s manual,” Spectral Sciences, Inc., Tech. Rep., 2011.

Publications by Author

- [1] A. Young, A. Killey, S. Marshall, and A. Gray, “Outlier and target detection in aerial hyperspectral imagery,” *HSI 2014 Hyperspectral Imaging and Applications, Coventry, UK*, 2014.
- [2] A. Young, S. Marshall, and A. Gray, “Outlier and target detection in aerial hyperspectral imagery: a comparison of traditional and percentage occupancy hit or miss transform techniques,” in *Automatic Target Recognition XXVI*, International Society for Optics and Photonics. SPIE-Intl Soc Optical Eng, May 2016, pp. 98 440S–98 440S. [Online]. Available: <http://dx.doi.org/10.1117/12.2213530>
- [3] A. Young, A. Kay, S. Marshall, R. Torr, and A. Gray, “Hyperspectral imaging for erosion detection in wind turbine blades,” *HSI 2016 Hyperspectral Imaging and Applications, Coventry, UK*, 2016.
- [4] A. McCluskey, J. Sindt, A. Young, N. Sommerdijk, P. Murray, P. Camp, and F. Nudelman, “Self-assembly of collagen molecules into fibrils in solution,” *Gordon Conference on Biomineralisation, Barcelona, Spain*, 2016.
- [5] A. Young, S. Marshall, A. Gray, and H. White, “Atmospheric correction and target detection in aerial hyperspectral imagery,” *Marine Enterprise Innovation Scotland Conference 2016, Glasgow, UK*, 2016.

Publications Under Consideration

- [1] A. Young, S. Marshall, A. Gray, and H. White, “An improved method of atmospheric correction for aerial hyperspectral imagery using modtran radiative transfer models,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. ##, no. #, pp. #####–#####, 2016.
- [2] A. Young, A. Kay, S. Marshall, R. Torr, and A. Gray, “Erosion detection in wind turbine blades using hyperspectral imagery,” *IEEE Transactions on Industrial Electronics*, vol. ##, no. #, pp. #####–#####, 2016.
- [3] A. Young, S. Marshall, A. Gray, and H. White, “Data reduction for hyperspectral imaging applications,” *25th European Signal Processing Conference, Kos, Greece*, 2017.