Doctorado en Tecnologías de Telecomunicación e Ingeniería Computacional

PhD Thesis

Contributions to hyperspectral acquisition and processing for UAV based systems

Pablo Horstrand Andaluz

Las Palmas de G.C., December 2019

UNIVERSIDAD DE LAS PALMAS
DE GRAN CANARIA

UNIVERSIDAD DE LAS PALMAS
DE GRAN CANARIA

D. GUSTAVO MARRERO CALLICÓ COORDINADOR DEL PROGRAMA DE DOCTORADO EN TECNOLOGÍAS DE TELECOMUNICACIÓN E INGENIERÍA COMPUTACIONAL DE LA UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA, INFORMA,

Que la Comisión Académica del Programa de Doctorado, en su sesión de fecha **cuatro de octubre de 2019**. tomó el acuerdo de dar el consentimiento para su tramitación, a la tesis doctoral titulada ***"Contributions to hyperspectral acquisition and processing for UAV based systems"***, presentada por el doctorando D. Pablo Horstrand Andaluz y dirigida por los Doctores D. José Francisco López Feliciano y D. Sebastián López Suárez. Además, se ha acordado que la tesis **reúne todos los requisitos para optar a la mención de DOCTORADO INTERNACIONAL**.

Y para que así conste, y a efectos de lo previsto en el Artº 11 del Reglamento de Estudios de Doctorado (BOULPGC 7/10/2016) de la Universidad de Las Palmas de Gran Canaria, firmo la presente en Las Palmas de Gran Canaria, a **cuatro de octubre de 2019**.

MARRERO CALLICO
GUSTAVO -

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Instituto Universitario de Microelectrónica Aplicada

Instituto: INSTITUTO UNIVERSITARIO DE MICROELECTRÓNICA APLICADA

Programa de doctorado: DOCTORADO EN LAS TECNOLOGÍAS DE TELECOMUNICACIÓN E INGENIERÍA COMPUTACIONAL

Título de la Tesis

CONTRIBUTIONS TO HYPERSPECTRAL ACQUISITION AND PROCESSING FOR UAV BASED SYSTEMS

Tesis Doctoral presentada por D. PABLO HORSTRAND ANDALUZ

Dirigida por el Dr. D. JOSÉ FCO. LÓPEZ FELICIANO

Codirigida por el Dr. D. SEBASTIÁN LÓPEZ SUÁREZ

El Director/a,	El Codirector/a,	El Doctorando,
(firma)	(firma)	(firma)

Las Palmas de Gran Canaria, a 10 de octubre de 2019

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Instituto Universitario de Microelectrónica Aplicada

DIVISIÓN DE DISEÑO DE SISTEMAS INTEGRADOS

TESIS DOCTORAL

Contributions to hyperspectral acquisition and processing for UAV based systems

Pablo Horstrand Andaluz

Abstract

It results evident that Unmanned Aerial Vehicles (UAV), also known as drones, are playing an important role in today's consumers market, as they are presently affordable for anyone who wants to enjoy them as a gaming device. However, the real added value that widens the spectrum of applications is based on its payload, normally sensors and other type of electronic and/or mechanical systems, permitting their use as a very valuable tool in different domains such as agriculture, environmental monitoring, security, construction or mineralogy, just to name some. However, it happens that rarely a standalone technology is sufficient to solve everyday problems, and the combination of some of them enriches these solutions making them smarter and productive. This is the case of hyperspectral technology, traditionally mounted on-board satellites and airplanes, acquiring Earth's surface information and providing for each single pixel a huge amount of data related to the inner physical and chemical composition of matter. The miniaturization of this type of sensors has facilitated its integration in drones at a much lower cost, resulting in a better spatial and temporal resolution.

Nonetheless, in order to exploit all the advantages that such hyperspectral flying platform could bring into a specific application, several steps need to be accomplished. In this sense, the main goal of this PhD thesis is the development of a methodology that fills the gap between hyperspectral technology and drones, facilitating the combination of both technologies and applying the resulted solution to the particular case of precision agriculture.

This document conducts the reader to the process of setting up a hyperspectral flying platform, and details how the encountered difficulties have been solved. First, a UAV hyperspectral imaging system has been developed based on a VNIR pushbroom sensor for acquiring images over vegetation fields. Then, the raw captured bitstreams by the hyperspectral camera have been pre-processed in order to prepare the data sets for the ulterior image processing algorithms. Although there are many algorithm candidates to be included in the validation process, a new one named *Line-by-Line Anomaly Detection* (LbL-AD) has been developed since its inception to uncover anomalies as soon as they are captured by the hyperspectral pushbroom sensor. Testing such algorithms needs of a simulation environment able to generate virtual scenarios that are a faithful representation of reality. This created environment accelerates the testing process and permits the reproducibility of similar conditions whenever needed. This is especially interesting in

the field of agriculture, where seasonal crops reduce enormously the number of tests to be performed within a year. Finally, the collected information throughout the performed flight missions is processed with state-of-the-art algorithms to obtain vegetation indices, classification and abundance maps that provide information related to the health status of the cropping areas, the differences among elements and possible different vegetation variants contained in a scene.

The accomplished experiments prove that remote sensing could be more than just earth monitoring of large surfaces from time to time. The scenario has changed considerably with the appearance of UAVs and the possibility of customizing the systems, which enables even the utilization of such technology in applications that demand real time executions for a rapid response.

Resumen

Resulta evidente que los vehículos aéreos no tripulados, también conocidos como drones, actualmente tienen un papel importante en el mercado, ya que constituyen soluciones atractivas y asequibles para los usuarios que quieran hacer uso de los mismos como herramientas de ocio. No obstante, el valor añadido real que amplía el espectro de aplicaciones, viene determinado por la carga útil que pueden transportar, normalmente sensores y sistemas electrónicos o mecánicos. De esta forma se permite el uso de estos vehículos en diferentes áreas como agricultura, medioambiente, seguridad o mineralogía, por citar algunos ejemplos. Pocas veces una única tecnología por sí sola es suficiente para resolver los problemas, y se requiere de una combinación de varias de ellas para alcanzar una solución más inteligente y productiva. Este es el caso de la tecnología hiperespectral, tradicionalmente embarcada a bordo de satélites y aviones para adquirir información de la superficie terrestre, obteniendo para cada píxel de la imagen una gran cantidad de información relacionada con las propiedades fisicoquímicas de la materia. La miniaturización de este tipo de sensores ha posibilitado su integración en drones a un coste reducido, dando como resultado una mejor resolución espacial y temporal.

Sin embargo, para conseguir aprovechar todas las ventajas que estos sistemas pueden aportar a la aplicación en concreto, se tienen que llevar a cabo varios pasos. En esta Tesis doctoral, se propone una nueva metodología para usar tecnología hiperespectral en aplicaciones reales basadas en el uso de drones, tratando de reducir el espacio existente entre adquisición y procesamiento de imágenes hiperespectrales, y aplicando los resultados a un caso particular: la agricultura de precisión.

Este documento presenta el proceso de construcción de una plataforma hiperespectral, detallando las dificultades encontradas por el camino y cómo han sido resueltas. En primer lugar, se ha desarrollado una plataforma de vuelo basada en un vehículo aéreo no tripulado y una cámara hiperespectral pushbroom VNIR, con el fin de adquirir imágenes sobre diferentes campos de cultivo. Asimismo, las imágenes capturadas son tratadas en una cadena de preprocesamiento para prepararlas para un posterior procesamiento encaminado a la extracción de información para la aplicación. Un algoritmo, denominado *Line-by-Line Anomaly Detection* (LbL-AD) ha sido desarrollado, con el fin de detectar anomalías en tiempo real en imágenes capturadas por sensores de tipo pushbroom. A su vez, un entorno de simulación hiperespectral ha sido creado para la generación de escenarios virtuales que son una fiel representación de la realidad y que tienen como objetivo, permitir la correcta

validación de los algoritmos hiperespectrales. Esto es de especial relevancia en un ámbito como la agricultura, donde la disponibilidad para realizar pruebas es reducida debido al carácter estacional de las cosechas. Finalmente, la información recolectada durante todas las misiones de vuelo realizadas, ha sido procesada con algoritmos del estado del arte para obtener mapas de índices de vegetación, clasificación y de abundancia, que proporcionan información relacionada con el estado de salud de la cosecha, las posibles diferencias entre elementos presentes y posibles variantes de vegetación en un mismo terreno.

Los experimentos realizados demuestran que la teledetección es mucho más que simplemente la observación terrestre de grandes extensiones de terreno cada cierto tiempo. El tablero ha cambiado considerablemente con la aparición de los vehículos aéreos no tripulados, y la posibilidad de customizar los sistemas, lo que ha permitido la utilización de esta tecnología en aplicaciones en tiempo real que requieren de respuestas inmediatas, complementando los métodos tradicionales de monitorización.

Contents

Resumen **iii**

List of Figures **ix**

List of Tables **xiii**

Abbreviations **xv**

1 Introduction **1**
1.1 Outline 2
1.2 Preliminary concepts 5
1.2.1 Definition and characterization of spectral images 6
1.2.2 Unmanned aerial platforms 7
1.2.3 Smart Farming 9
1.2.4 Targeted domains and motivations of this Thesis 10
1.2.4.1 International and national project contributions 12
1.2.4.2 Hyperspectral image acquisition on-board a UAV 14
1.2.4.3 Anomaly Detection Algorithms 15
1.2.4.4 Virtual simulation environments 17
1.2.4.5 Captured hyperspectral data processing 17
1.3 Research goals and contributions 21
1.4 Organization of this document 25
1.4.1 Chapter 2: A UAV platform based on a hyperspectral pushbroom sensor 25
1.4.2 Chapter 3: Hyperspectral image acquisition 25
1.4.3 Chapter 4: Hyperspectral anomaly detection algorithms 25
1.4.4 Chapter 5: Hyperspectral simulation environment 26
1.4.5 Chapter 6: Farming Use Case 26
1.4.6 Chapter 7: Conclusions 26

2 A UAV Platform based on a Hyperspectral Pushbroom Sensor **27**
2.1 Outline 28
2.2 Spectral sensors 29

2.2.1 Whiskbroom imagers . . . 31
2.2.2 Pushbroom imagers . . . 31
2.2.3 Multi-Camera 2D Imagers . . . 33
2.2.4 Sequential 2D Imagers . . . 33
2.2.5 Multi-point spectrometer . . . 34
2.2.6 Mosaic filter-on-chip cameras . . . 34
2.2.7 Spatiospectral filter cameras . . . 35
2.3 State-of-the-art in UAV Platforms based on multi and hyperspectral sensors 36
2.4 Platform devices selection and description . . . 38
2.4.1 Hyperspectral camera . . . 38
2.4.2 Flying Platform. Drone. . . . 42
2.4.2.1 Precision GPS System: Dji RTK . . . 44
2.4.2.2 Camera Stabilization System: Dji Ronin-Mx . . . 45
2.4.3 Industrial Frame Camera . . . 47
2.4.4 On-board embedded platform . . . 49
2.5 Platform devices interconnection. Hardware integration. . . . 52
2.6 Application for controlling the system. Software integration. . . . 56
2.6.1 End-user application . . . 57
2.6.1.1 User inputs . . . 58
2.6.1.2 Waypoint layout . . . 59
2.6.2 Flight control application . . . 61
2.6.3 Camera control application . . . 64
2.6.4 Communication protocol . . . 65
2.7 Flight Performance . . . 69
2.7.1 Assessment Metrics . . . 69
2.7.2 Geographic units conversion . . . 71
2.7.3 Flight mission results . . . 71
2.8 Conclusions . . . 78

3 Hyperspectral image acquisition 81
3.1 Outline . . . 82
3.2 Flight missions: captured data . . . 83
3.2.1 Campus Tafira: Football field . . . 85
3.2.2 Veneguera: Mango, banana and avocado fields . . . 86
3.2.3 Tejeda: Bentayga Vinyard . . . 88
3.3 Image calibration . . . 90
3.4 Radiometric calibration . . . 92
3.5 Band Signal to Noise Ratio . . . 97
3.6 Image Georectification . . . 98
3.6.1 Direct Georreferencing . . . 99
3.6.2 Geometric correction . . . 102
3.7 Conclusions . . . 107

4 Hyperspectral anomaly detection algorithms 117
4.1 Outline . . . 118
4.2 State-of-the-art in hyperspectral anomaly detection algorithms . . . 119
4.3 LbL-AD algorithm description . . . 121
4.3.1 Computation of first n lines . . . 125
4.3.2 Subspace calculation. Power Iteration method and deflation. . . . 125
4.3.3 Image projection and Mahalanobis distance calculation . . . 127
4.4 Data and evaluation metrics . . . 128
4.4.1 Reference Hyperspectral Data . . . 129
4.4.2 Reference Algorithms . . . 130
4.4.3 Assessment Metrics . . . 131
4.5 Experimental Results . . . 133
4.5.1 Analysis of input parameters . . . 133
4.5.2 Detection Performance . . . 135
4.5.2.1 Validation against images from the state-of-the-art . . . 135
4.5.3 Execution Performance . . . 139
4.6 Conclusions . . . 141

5 Hyperspectral simulation environment 143
5.1 Outline . . . 144
5.2 State-of-the-art in simulation environments . . . 146
5.3 Agriculture Graphical Simulation Environment: AgSim . . . 147
5.4 Proposed Simulation Environment . . . 150
5.5 Creation of Hyperspectral Scenarios . . . 154
5.5.1 Target detection scenarios . . . 154
5.5.2 Hyperspectral map scenarios based on real bitmaps . . . 158
5.6 Experimental Results . . . 162
5.6.1 Anomaly Detection Algorithms . . . 162
5.6.2 Flight mission results . . . 165
5.7 Conclusions . . . 172

6 Application Use Case: Precision Farming 175
6.1 Outline . . . 176
6.2 State-of-the-art in precision farming with unmanned aerial vehicles . . . 177
6.3 Anomaly Detection . . . 179
6.4 Vegetation Index Maps . . . 183
6.4.1 Fusarium infection of banana trees . . . 188
6.5 Hyperspectral Image Classification: Labelled maps . . . 193
6.6 Hyperspectral linear unmixing and abundance maps . . . 203
6.7 Conclusions . . . 209

7 Conclusions and further research **211**
7.1 Conclusions 212
7.2 Further research work 217

A Sinopsis en español **221**
A.1 Introducción 222
A.2 Objetivos y metodología de trabajo 224
A.3 Desarrollo de una plataforma de vuelo no tripulada basada en un sensor hiperespectral de barrido lineal 225
A.4 Adquisición de imágenes hiperespectrales 226
A.5 Detección de anomalías 227
A.6 Entorno de simulación de algoritmos hiperespectrales 228
A.7 Caso de uso: Agricultura de precisión 228
A.8 Conclusiones 229

B Publications **231**
B.1 Journals 232
B.2 International Conferences 233

References **235**

List of Figures

1.1 Spectral imaging concept. 7
1.2 Electromagnetic spectrum. 8
1.3 ENABLE S-3 farming use case scenario representation. 13
1.4 Main goals and contributions 24

2.1 Hyperspectral pushbroom optical chain. 32
2.2 Ronin Mx configuration software. 46
2.3 Filter installation in the IDS camera. 48
2.4 Hardware devices to provide two additional USB 3.0 ports to the Jetson TK1. 51
2.5 General overview of the UAV platform components. 53
2.6 Gimbal construction elements breakout. 54
2.7 Gimbal assembly. 56
2.8 UAV flying platform. 56
2.9 iOS application screenshots. 58
2.10 UAV flight schematics. 60
2.11 Camera control code flow diagram. 64
2.12 Communication protocol message structure. 67
2.13 Communication flow diagram. 68
2.14 Flight trajectory of two different missions performed in two different regions of Gran Canaria. 73
2.15 Platform height above ground of mission 1 performed over the University Campus football field. 74
2.16 Platform heading of mission 2 performed over the Veneguera vegetation fields. 75
2.17 Gimbal orientation angles of mission 2 performed over Veneguera vegetation fields. 75
2.18 Platform speed of mission 1 performed over Veneguera vegetation fields. . 76

3.1 Proposed pre-processing chain. 83
3.2 Flight Mission locations. (a) Overview of Gran Canaria. (b) Football field in Campus Tafira. (c) Veneguera. (d) Tejeda vineyard location 1. (e) Tejeda vineyard location 2. 84
3.3 Football field mission outlook. 86
3.4 Veneguera mission outlook. 87

3.5 Tejeda mission outlook. 89
3.6 Spectral response of the white reference panel measured. 91
3.7 Pixel calibration for the real acquired hyperspectral data. 93
3.8 Measured spectral lamps at the center column. Calibration peaks are marked with red circles. 94
3.9 Zenith Polymer material acquisition using the Specim FX10. 95
3.10 Zenyth Polymer spectral response captured with the Specim fx10. 96
3.11 Zenyth Polymer spectral curves, measured and certified. 96
3.12 Specim fx10 SNR estimation. 98
3.13 Geometry of the acquisition at 50 m of altitude. 100
3.14 Georeferencing of swath 1 displayed in Figure 3.19. 102
3.15 Mission 1 over the football field, sector 1. Platform real trajectory, ideal trajectory and trajectory compensation. 103
3.16 Georectification of swath 1 displayed in Figure 3.19. 105
3.17 Image overlay on top of Google Earth map. 106
3.18 IDS frame camera captured images. 107
3.19 Captured swathes of football field mission 1. 109
3.20 Captured swathes of football field mission 2. 110
3.21 Veneguera, mission 1. Captured swathes. 111
3.22 Veneguera, mission 2. Captured swathes. 112
3.23 Veneguera, mission 3. Captured swathes. 113
3.24 Tejeda, mission 1. Captured swathes. 114
3.25 Tejeda, mission 2. Captured swathes. 115

4.1 General overview of the LbL-AD algorithm. 124
4.2 Test datasets. (a) Synthetic image. (b) WASP RIT scene. (c) AVIRIS WTC scene. (d) HYDICE WDC scene. 131
4.3 Image average pixel. Analysis based on the used lines in the buffer 134
4.4 Two-dimensional plots of the detection results obtained by the AD algorithms considered in this work (rows) for all datasets (columns). To make the results easier to understand, the ground-truths of the anomalous maps are shown in the last row. 137
4.5 The three-dimensional (3D) plots of the detection results obtained by the LbL-FAD and their corresponding binary maps thresholding by τ for the five datasets: (a) Synthetic image, (b) WASP RIT scene, (c) AVIRIS WTC scene, (d) WDC scene. 138
4.6 Execution time comparison in terms of availability of the result. 140

5.1 Proposed validation and verification strategy. 145
5.2 Farming simulation environment and machinery representation. 149
5.3 Hyperspectral simulation environment block diagram. 150
5.4 Hyperspectral imaging tool windows. 151
5.5 View from the drone perspective. 153
5.6 Background and anomaly spectra. 156

5.7 Synthetic image generation based on field ground-truth. On the left hand side captions of the simulator environment view from the drone with some different objects are represented. On the right hand side, the RGB representation of the synthetic image produced from that view are shown. . 157
5.8 Mission control application window built over the simulation environment map layout. 159
5.9 Image of the football field located at the University of Las Palmas de Gran Canaria campus. Image taken with a Phantom 4 drone from Dji at a height of 80 meters. 161
5.10 Three captions of the simulation run with results shown in table 5.2, with parameter d set to 5. 165
5.11 Mission flight drone orientation angles. 167
5.12 Mission flight drone altitude above ground and trajectory. 168
5.13 Mission flight speed. 168
5.14 Captured swathes. 171

6.1 LbL-AD validation against UAV hyperspectral images captured over Bentayga vineyards, mission 2. (a) Image 1, cropped from swath 4. (b) Image 2, cropped from swath 5. (c) LbL-AD result image 1. (d) LbL-AD result image 2. (e) Result displayed in (c) masking out the background. (f) Result displayed in (d) masking out the background. 181
6.2 LbL-AD validation against UAV hyperspectral images captured over Veneguera fields, mission 2. (a) Image 1, cropped from swath 3. (b) Image 2, cropped from swath 1. (c) LbL-AD result image 1. (d) LbL-AD result image 2. (e) Result displayed in (c) masking out the background. (f) Result displayed in (d) masking out the background. 182
6.3 Vegetation indices applied to selected region of image swath 2, mission 1 over Bentayga Vineyard, displayed in Figure 3.24. 186
6.4 Soil and vegetation separation using the OSAVI index. 188
6.5 NDVI analysis of two banana tree leaves, one healthy and one affected by the fungus. 190
6.6 Spectral reflectance of the dots indicated in Figures 6.5a and 6.5c. 191
6.7 Banana plantation UAV image analysis. 192
6.8 Linear SVM classification result, over image subset of mission 1 performed over Bentayga vineyards. 196
6.9 Sample spectra from the classification map displayed in Figure 6.8. 197
6.10 Linear SVM classification result, over image subset of mission 2 performed over Bentayga vineyards. 198
6.11 Sample spectra from the classification map displayed in Figure 6.10. 200
6.12 Grape sort classification. RGB representations of the selected regions containing Tintilla grape sort are displayed in a) and b). Classifications maps are presented in c) and d). 201
6.13 Grape sort classification. RGB representations of the selected regions containing Vijariego grape sort are displayed in a) and b). Classifications maps are presented in c) and d). 202

6.14 Selected image portion from flight mission 1 performed over Veneguera to apply linear unmixing. 205
6.15 Representation of the image information representation based on the endmembers determined by the FUN. 206
6.16 Abundance maps of the calculated endmembers. a) Endmember 1. b) Endmember 2. c) Endmember 3. d) Endmember 4. e) Endmember 5. . . . 207
6.17 Spectral signatures of the calculated endmembers. 209

A.1 Plataforma de aérea no tripulada con cámara hiperespectral de barrido lineal. 225
A.2 Representación en color de las capturas hiperespectrales realizadas en una de las misiones con la plataforma de vuelo. 226
A.3 Aplicación de índices de vegetación a una imagen hiperespectral. a) Representación en color de la imagen. b) Resultado NDVI. 229

List of Tables

2.1 Spectral sensor types and characteristics. 30
2.2 Hyperspectral pushbroom devices fulfilling the required characteristics. . . 41
2.3 Drone main characteristics. 44
2.4 Industrial frame sensor characteristics. 47
2.5 Embedded boards main characteristics. 50
2.6 Flight mission data captured over the football field in the University Campus. 77
2.7 Table 2.6 continuation. 77
2.8 Flight mission data captured over a Vegetation filed located in the west side of the island of Gran Canaria. 77
2.9 Table 2.8 continuation. 78

3.1 University Campus football field missions setup. 85
3.2 University Campus football field missions setup. 88
3.3 Tejeda, Bentayga Vinyard missions setup. 90

4.1 PLP-KRXD parameter table. 135
4.2 Implementation results against the datasets described in Section 4.4.1 and using the assessment metrics proposed in Section 4.4.3. 136

5.1 AgSim API drone functions. 153
5.2 Simulation run of 51.5 seconds, 1030 frames captured and d set to 5. 163
5.3 Simulation run of 44.85 seconds, 897 frames captured and d set to 4. 164
5.4 Waypoint mission setup. 166
5.5 Flight mission data metrics. 167
5.6 Table 5.5 continuation. 169

6.1 Vegetation indices used in this manuscript. 185
6.2 Vegetation indices obtained values. 187
6.3 Classification results pixel coverage. 196
6.4 Classification results, pixel coverage. 197
6.5 Classification results pixel coverage. 199
6.6 Classification results pixel coverage. 199
6.7 Total summation of pixels abundances per endmember and the equivalent percentage. 208

Abbreviations

2D	Two-Dimensional
3D	Three-Dimensional
AD	Anomaly Detection
AgSim	Agriculture Graphical Simulation Environment
AIS	Airborne Imaging Spectrometer
API	Application Programming Interface
AUC	Area Under Curve
ARVI	Atmospherically Resistant Vegetation Index
AVIRIS	Airbone Visible/Infrared Imaging Spectrometer
CAN	Controller Area Network
CARI	Chlorophyll Absorption Ratio Index
CMOS	Complementary Metal–Oxide–Semiconductor
CSP	Creanex Simulation Platform
BS	Brier Score
CPU	Central Processing Unit
CRD	Collaborative Representation Detector

CSCAD Combined Similarity Criterion Anomaly Detector

CSNMF Constrained Sparse Nonnegative Matrix Factorization

CUDA Computer Unified Device Architecture

DL Deep Learning

ECUs Electronic Control Units

ENABLE-S3 European Initiative to Enable Validation for Highly Automated Safe and Secure Systems

EnMAP Environmental Monitoring and Analysis Program

ER-2 Earth Resources Aircraft 2

EO-1 Earth Observing Mission 1

FAO Food and Agriculture Organization of the United Nations

FCLSU Fully Constrained Least Squares linear spectral Unmixing

FIFO First-In-First-Out

FN False Negatives

FOV Field of View

FP False Positives

FPR False Positive Rates

FPGA Field-Programmable Gate Array

FPI Fabry–Pérot Interferometer

FPS Frames per Second

FUN Fast algorithm for linearly UNmixing hyperspectral images

FWHM Full Width at Half Maximum

GigE	Gigabit Ethernet
GPIOs	General Purpose Input Output connectors
GPS	Global Positioning System
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HiL	Hardware in the Loop
HSI	Hyperspectral Imaging
HYDICE	Hyperspectral Digital Imagery Collection Experiment
HyperLCA	Lossy Compression Algorithm for Hyperspectral image systems
HySIME	Hyperspectral Subspace Identification by Minimum Error
IASI	Infrared Atmospheric Sounding Interferometer
ICA	Independent Component Analysis
IMU	Inertial Measuring Units
IoT	Internet of Things
JPL	Jet Propulsion Laboratory
LbL-AD	Line by Line Anomaly Detection
LbL-FAD	Line by Line Fast Anomaly Detection
LAI	Leaf Area Index
Lidar	Light Detection and Ranging
LiPo	Lithium Polymer
LSU	Least Squares linear spectral Unmixing
LWIR	Long Wavelength Infrared

MCARI	Modified Chlorophyll Absorption Ratio Index
MSAVI	Modified Soil-Adjusted Vegetation Index
MSI-A	Multispectral Instrument Sensor A
MSI-B	Multispectral Instrument Sensor B
MVCA	Modified Vertex Component Analysis
MWIR	Medium Wavelength InfraRed
NASA	National Aeronautics and Space Administration
NDA	Non-Disclosure Agreement
NDVI	Normalize Difference Vegetation Index
NED	North-East-Down
NIR	Near InfraRed
OSAVI	Optimized Soil-Adjusted Vegetation Index
OSPRX	Reed-Xiaoli after Orthogonal Subspace Projection
PCA	Principal Component Analysis
PID	Proportional Integral Derivative
PLP-KRXD	Progressive Line Processing of Kernel Reed-Xiaoli Anomaly Detection
PRISMA	Prototype Research Instruments and Space Mission technology Advancement
PPI	Pixel Purity Index
Radar	Radio Detection and Ranging
RGB	Red, Green, Blue
RIT	Rochester Institute of Technology

RMSE	Root Mean Square Error
ROC	Receiver Operating Characteristics
RX	Reed-Xiaoli
SAM	Spectral Angle Mapper
SATA	Serial Advanced Technology Attachment
SDKs	Software Development Kits
SER	Squared Error Ratio
SfM	Structure from Motion
SNR	Signal-to-Noise Ratio
SiL	Software in the loop
SSFC	Spectral-Spatial Feature-based Classification
SSRX	Subspace Reed-Xiaoli
SVD	Singular Value Decomposition
SVM	Support Vector Machines
SAVI	Soil-Adjusted Vegetation Index
SWIR	Short Wavelength Infrared
TN	True Negatives
TP	True Positives
TPR	True Positive Rates
UAVs	Unmanned Aerial Vehicles
USGS	United States Geological Survey
USMS	Simulated MultiSpectral

VCA Vertex Component Analysis

VIs Vegetation Indices

VNIR Visible and Near InfraRed

VSWIR Visible Shortwave InfraRed

WASP Wildfire Airborne Sensor Program

WGS World Geodetic System

WTC World Trade Centre

Chapter 1

Introduction

The main objective of this Chapter is to present the motivation for this Thesis, whose major contribution is a methodology for using on-board hyperspectral technology in a practical application, from the conception of an unmanned aerial platform based image acquisition system, to the development and application of processing algorithms that serve to extract useful information for the targeted domain. First, the hyperspectral imaging basics are briefly outlined alongside with the capturing process. Then, different aerial platforms are discussed, highlighting the advantages that a copter flying platform bring compared to the others. Finally, the research goals and contributions of this Thesis as well as the organization of the document are presented.

1.1 Outline

Since its early development, in the 1970's, imaging spectrometry, or hyperspectral imaging as it is called today, has experienced an enormous progress due to the improvements seen in electronics, computing and software throughout these years, becoming one of the most powerful tools for acquiring information in several different fields [1–4]. Whereas conventional Red, Green, Blue (RGB) cameras are only able to capture light in the visible range, providing information exclusively for three spectral bands, hyperspectral sensors are able to divide the spectrum into many more contiguous bands, generating an image of the scene under observation with information across a much wider range of the electromagnetic spectrum for each pixel.

The main advantage of this technology is that certain materials, when exposed to light, reflect a unique spectral signature in the electromagnetic spectrum, making it easier to identify the elements that compose the scene under analysis when the right techniques are applied.

The first sensors collecting hyperspectral data were used in the field of Remote Sensing, installed on-board satellites or airplanes, gathering spectral information of the earth surface. More specifically, it was the Jet Propulsion Laboratory (JPL) under funding from the National Aeronautics and Space Administration (NASA) that developed in 1983 the Airborne Imaging Spectrometer (AIS) [5], the first of a new class of remote sensing instruments. The Airbone Visible/Infrared Imaging Spectrometer (AVIRIS) [6], developed by the same laboratory, followed a few years later, setting a milestone in the technology. Based on a whiskbroom imager that acquires data in 224 narrow, contiguous spectral bands, and flown aboard the NASA's Earth Resources Aircraft 2 (ER-2) at high altitudes (20 km above sea level approximately), it has proven to be one of the prime providers of high quality hyperspectral datasets for the scientific community throughout the last decades.

On-board satellites, the hyperspectral instrument Hyperion [7], mounted on the spacecraft NASA's Earth Observing Mission 1 (EO-1), meant a breakthrough in the technology using lightweight materials that enabled the utilization of such sensors in these aerial platforms.

It recorded data with 30 meters ground sample distance and 400–2500 nm as the spectral range, spanned in 200 bands. For that reason Hyperion has had a long lifetime and has not been decommissioned until 2017. Several years after the launch of the EO-1, the Chinese Academy of Science released the Shenzou-8 platform which had the TianGong-1 instrument [8] also covering the same range as the Hyperion with a reduced number of bands, 128. In recent years, many are the countries that have launched new space missions or are going to in the close future, incorporating a multispectral or a hyperspectral sensor aboard to perform earth inspection tasks. Some of the most relevant platforms and their sensors are, Sentinel-2 with the multiespectral sensors Multispectral Instrument Sensor A (MSI-A) and Multispectral Instrument Sensor B (MSI-B) [9], Prototype Research Instruments and Space Mission technology Advancement (PRISMA), with the hyperspectral sensor PRISMA [10] and Environmental Monitoring and Analysis Program (EnMAP) platform with the hyperspectral sensor EnMAP Hyperspectral Imaging (HSI) [11].

After the mentioned improvements in the technology, having uncovered to the scientific community its main advantages, hyperspectral imaging systems have experienced an exponential growth in the last few years, with new players coming into the scene, such as camera manufacturers, software and system developers. This has permitted the massive reduction of costs allowing the expansion of the technology into several other fields. For this reason, laboratory and industrial applications in the areas of food processing, health care, mineralogy and surveillance, among others have been gaining importance [12–14]. For instance, in the health care domain, special focus has been set to the tumour tissue identification in different parts of the human body, since the hyperspectral images have demonstrated to provide information which allows to distinguish between the tumour and healthy tissue without the necessity of using invasive techniques [15].

The major technological advancements has meant a great improvement in terms of camera weight reduction and miniaturization, what is enabling the installation of such devices into platforms that have payload limitations. That is the case of Unmanned Aerial Vehicles (UAVs), which present some very attractive advantages for the users with respect to airplanes and satellites, in the sense that it is a lower cost approach with a more flexible

revisit time and a better spatial resolution, permitting a deeper and more accurate data analysis [16]. Moreover, missions can be customized which allows the selection of the exact area to be scanned, the flight speed and altitude among other parameters, making this type of solution a very appealing one for certain applications. Some other interesting available solutions of hyperspectral cameras on-board UAVs can be found in the scientific literature [17–20].

The use of UAV with sensors aboard for surveillance and inspection opens a plethora of applications. Just to name some, in [21] UAVs are used to detect power lines to achieve automatic power line inspection. [22] presents several missions in different safety, security and rescue field tests using UAVs and [23] illustrates the advantages offered by the introduction of small-scale UAVs in the near future for civilian applications, concretely, in police departments, fire brigades and other homeland security organizations. However, it is in the agriculture field where remote sensing relying on UAVs has been strongly positioned as an emerging field of application [24].

It is important to highlight that although the advantages are immense, the integration of systems based on hyperspectral sensors poses some major challenges, specially when equipped with hyperspectral pushbroom sensors that are currently very often being used due to the good trade-off they present between spectral and spatial resolution compared to other available hyperspectral sensor technology [25, 26]. These technological challenges have translated into the development of general purpose hyperspectral platforms for capturing images and later analysing them, but there is still a huge gap between the acquisition and the specific application, which leaves unexploited several compelling advantages that could be offered by the technology. For instance, real time applications are seldom considered in the hyperspectral remote sensing domain, for a few relevant reasons: the lack of suited platforms equipped with the necessary hardware and software to achieve real time, algorithms developed to process complete hypercubes and not adapted to the sensor acquisition and the difficulty of carrying out the necessary tests for system verification and validation.

For all the mentioned reasons, in this Thesis a set of contributions for bringing closer a hyperspectral remote sensing application with UAV to the targeted domain is

presented. The standard approach so far has been, on the one hand, to put together the flying platform with the hyperspectral sensor and the necessary devices to perform the acquisition, and on the other hand, completely decoupled from this first task, to start designing and implementing offline the processing algorithms that are going to be embedded in the platform. In this way when everything is ready and the first tests are to be carried out on a real environment it is noticed that several things do not fit and have to be reworked. This gap left between offline implementations and real applications is meant to be reduced in this work through a virtual environment that enables the validation and verification of processing algorithms in combination with flight control. The platform simulates the machinery and the sensors aboard. This is specially appealing when it comes to applications that have real-time processing requirements, when the system has to immediately respond to a stimulus.

In particular, this Thesis contributes to all the steps involved in a real application. First, a UAV flying platform based on a pushbroom hyperspectral sensor has been setup, obtaining outstanding results in terms of data collection, both images and telemetry data, thanks to the efforts put in the mission preparation, flight control and task synchronization. Secondly, implementations of algorithms targeting different scenarios, such as anomaly detection and geometric correction, have been developed and the application of existing state-of-the-art algorithms such as vegetation index maps generation, classification and linear unmixing have been tested. Finally, a hyperspectral simulation environment has been created to enable the testing of the hyperspectral algorithms before going into the field and reduce the gap between offline and real implementations. Furthermore, the whole work has been performed with a specific use case in sight: Precision Farming.

1.2 Preliminary concepts

In the following sections the reader is guided to the main concepts involved in the manuscript that will recursively show up, starting with the definition and characterization of multispectral and hyperspectral images. Then, some general aspects of unmanned aerial platforms are presented. An introduction to precision farming will follow, highlighting the

contributions that multi and hyperspectral technology on-board aerial platforms can bring to the application. Later, the targeted domains of this Thesis are discussed, emphasizing different processing methods that can help to extract useful information for the application and the use of virtual environments in testing and validating the system, as it has been exploited in this Thesis.

1.2.1 Definition and characterization of spectral images

Spectral imaging is the sensing technique that measures the reflectance of areas on the surface of the scene under observation. Reflectance of a material is the fraction of incident electromagnetic energy that is reflected at the surface of that material, providing a unique curve or spectrum across a specific range of wavelength that enables the identification of that material. Spectral images are made of several pixels, each of them comprised of a vector, which is the spectral pattern just defined, for the specific area covered by that individual pixel. This results in three-dimensional images, with the spatial information represented in the X-Y plane and the spectral information along the Z axis. Figure 1.1 presents an example of a hyperspectral image where the described axis are indistinctively shown and two individual pixels representing the vegetation and soil have been selected with their spectral patterns displayed on the right hand side. Hyperspectral images are also called datacubes with N_r, N_c and N_b being the edges of the cube, corresponding to the number of rows, columns and bands of the image, respectively.

Spectral images are characterized by the size of the datacube and the spatial and spectral resolutions. The spatial resolution refers to the size in meters of each pixel in the X-Y plane, while the spectral resolution refers to the wavelength separation of two contiguous spectral bands. A higher spatial and spectral resolution indicates a smaller size of the pixel in the X-Y plane and a smaller wavelength width for each spectral band, respectively.

Since all this information is collected by a digital sensor, one extra resolution measurement is needed, namely the radiometric resolution. The radiometric resolution of the sensor corresponds to its sensitivity, i.e. its ability to discriminate very slight differences in electromagnetic energy. It determines the number of bits required for storing the energy

FIGURE 1.1: Spectral imaging concept.

sensed in each spectral band of each individual pixel. The higher radiometric resolution of the sensor, the more sensitive it is detecting small differences in the reflected energy and the more bits are required.

Spectral imaging is inherently linked to the electromagnetic spectrum, represented in Figure 1.2, where wavelength ranges of different radiation types are displayed. Most spectral imaging sensor devices work in a wavelength range from the visible to the infrared, the latter commonly divided in three different sectors, Near InfraRed (NIR), λ: 0.7 - 1.1 μm; Short Wavelength Infrared (SWIR), λ: 1.1 - 3.0 μm; Medium Wavelength InfraRed (MWIR), λ: 3.0 - 5.5 μm; and Long Wavelength Infrared (LWIR), λ: 7.7 - 14 μm.

1.2.2 Unmanned aerial platforms

In recent years, unmanned aerial vehicles (UAVs) have emerged as an alternative platform to satellites and airplanes in remote sensing operations, to collect data for inspection, surveillance and monitoring in the areas of defence, security, environmental protection and civil domains, among others. The discussed reduction in size and weight of hyperspectral sensors make UAV an interesting candidate when it comes to perform different flight campaigns.

FIGURE 1.2: Electromagnetic spectrum.

The advantages offered by these platforms are numerous. For instance, missions can be created ad-hoc and repeated, with the exact same flight parameters, as many times as desired, as opposed to the inflexible revisit times airplanes or satellites have. Moreover, flying at lower altitudes, captured data have a better spatial resolution, which permits a deeper and more accurate data analysis. Additionally, they represent an open easily customizable solution at a much lower-cost.

As an emerging technology, drones are continuously evolving. There are nonetheless two main drone models based on their construction philosophy, that are most commonly utilized: fixed wing and multi-rotor drones, which will be further evaluated in the next lines.

Fixed-wing drones have a single long wing on either side of their body and need either a catapult or a runway to lift it off the ground. They are more commonly flown for the purpose of long-distance tasks due to their ability to glide throughout some parts of the flight and therefore have a lower consumption. This approach offers a flight autonomy of about 16 hours. Their inability to hover makes the landing manoeuvre challenging as well, and for that particular reason also not very convenient to accommodate costly equipment.

Multi-Rotor Drones are based on multiple rotors and propellers used for the displacement alongside the three spatial axis, and its rotation about the vertical axis, called heading.

Propellers are basically the small wings or blades fixed on top of the rotors, that are just the motors that spin the propellers. They usually have four, six or eight rotors, in which case they receive the names quadcopter, hexacopter and octocopter, respectively.

Their ability to autonomously take-off, rotate, move in all directions and land make them the preferable option for most developers. Commercial multi-rotor drones are already able to carry up to 10 kg of payload, which can certainly fit the requirements of many applications. However, their main limitation is their autonomy, which is in the order of tens of minutes when powered by batteries, making the whole operation too discontinuous and sometimes unattractive for potential users. There are nonetheless works in the scientific literature proposing different solutions for such problem. On one hand, a wireless power charge approach in order to reduce the stopping time [27]. On the other hand the use of alternative propulsion systems (or hybrid combinations of some of them), which alleviate the restrictions imposed by electric batteries in terms of autonomy, are attracting great attention from the scientific community [28–30]

1.2.3 Smart Farming

Recent studies on world population growth paint a rather worrying future: according to the Food and Agriculture Organization of the United Nations (FAO), by 2050 the number of people around the planet should reach the unseen record of nearly 10 billion [31]. This forecast, combined with elements such as the declining rural labour supply (mostly due to the increase in urbanization rates), limited availability of arable land and growing scarcity of water and energy sources, represent an enormous challenge for agriculture. As a matter of fact, current agricultural practices will not cope with such demands with a sustainable production thus undermining the chances of granting food to sufficiently provide the society.

Therefore, technology shall play a key role in what have been called 'precision agriculture' or 'smart farming', defined as a modern farming management concept using digital techniques to monitor and optimise agricultural production processes.

So far, smart farming has been supported by three main technologies: Internet of Things (IoT) sensors, Global Positioning System (GPS) enhanced vehicles and UAVs equipped with imaging devices. However further contributions are expected for the near future in the form of process automation, using autonomous driving systems.

Sensors placed in the fields and connected through the Internet are able to record variables such as acidity, soil temperature and humidity, wind speed and direction, among others, providing a valuable information not only to farmers, but also to data analysers that can use such information to predict future events using artificial intelligence techniques [32], optimizing intervention and resource use.

GPS-enhanced tractors receive the maps from the field sensors and the drones, supporting their drivers taking decisions in every step, from field preparation to harvesting, which techniques to apply for the crop at hand, where to seed in order to avoid wasteful overlapping or planting through waterways, or where to focus the application of crop care products.

UAVs equipped with vision sensors are employed for periodically supervising and monitoring the fields status and get information about the crops's health, analysing the plants during their growth until the harvest, and controlling all external conditions that may affect their strength. They allow collecting periodical information of the crops using different kinds of imaging devices. In this scenario, several multispectral sensors are being widely used on-board UAVs for collecting spectral information that allows the generation of maps for indicating the aspects of the plant state [33]. One of the most typical examples is the Normalize Difference Vegetation Index (NDVI), which indicates the vigorousity of the plants using the information of two different spectral channels placed in the red and near infrared parts of the electromagnetic spectrum [34].

1.2.4 Targeted domains and motivations of this Thesis

Remote sensing with hyperspectral imaging have proven to be a very powerful tool for supervision and monitoring tasks. With the emergence of UAVs into the scene, targeted

domains have broaden including, for the first time, applications that can be executed in real time. For instance, fire detection, rescue operations, supervision of autonomous vehicles manoeuvres, among several others, are certainly applications that demand a prompt reply since the reaction time is very limited. However, the process to arrive at the point where the system is set, fully tested and running in continuous operation mode, might take several years and lead to extensive costs. Furthermore, the whole process might incur in waste of manpower resources if the system development is done sequentially. For example, algorithms might only be tested in a realistic scenario once the whole platform has been completely developed and is already flying on the field.

In this Thesis four main domains have been targeted. First, hyperspectral image acquisition through the development of a fully automated UAV platform based on a hyperspectral sensor. Second, the proposal of novel implementations adapted to the acquisition technology, enabling real time execution as images are being sensed, to obtain results on the fly. Third, the processing of the captured images using well-known algorithms such as Vegetation Indices (VIs), classification and linear unmixing, to extract valuable information for a specific domain, precision farming. Fourth, the development of a virtual environment that allows the testing of some of the mentioned algorithms in a faithful representation of the real scenario, before even the actual platform has been fully developed.

Furthermore, the work performed in this Thesis is framed in one of the use cases of a European project called European Initiative to Enable Validation for Highly Automated Safe and Secure Systems (ENABLE-S3) [35], funded by the Ecsel program during the period 2016-2019. The farming use case has the goal of implementing autonomous processes into the agriculture industry to guarantee a sustainable production increase that would meet the demands for the next decades. In this way, the contributions of this Thesis by the acquisition and processing of the hyperspectral images would support the farmers taking decisions, minimize fatal accidents and help in the early detection of plant unhealthy state.

1.2.4.1 International and national project contributions

ENABLE-S3 is a European H2020 ECSEL project whose main goal is to contribute to the accomplishment of highly automated and autonomous systems in six different domains: automotive, aerospace, rail, maritime, health care and farming. The objective is to develop verification and validation processes of real systems with reasonable efforts by using virtual testing tools.

The research presented in this Thesis has been demonstrated within the scenario called "Farming Use Case" [36], which consists in the automated driving of a harvester in an agriculture field performing the cropping, watering or pesticide spraying tasks. Figure 1.3 depicts the goal of the use case using four schemas. Initially, a drone equipped with a hyperspectral camera inspects the terrain in order to generate a set of maps with different VIs which extracts information related to the status of the crop, as depicted in Figure 1.3a. This information, together with the specific coordinates, is then sent to the harvester, which aided by its global planning system defines the trajectory, initiates its autonomous driving to the region of interest and starts its labour. Additionally, in order to minimize the risk of fatal accidents due to agricultural mowing operations and/or collisions with big obstacles (animals or rocks), the scheme shown in Figure 1.3b has been considered, in which the drone flies at a certain height above ground, and some meters ahead of the harvester (security distance) scanning the field for the possible appearance of static or dynamic objects.

During the labour of the harvester if an obstacle is detected by the drone, as in Figure 1.3c, this shall signal the harvester to alter its route and follow a different trajectory based on the information given by the drone, as shown in Figure 1.3d. The harvester is equipped with other sensors such as a Radio Detection and Ranging (Radar) [37] or a Light Detection and Ranging (Lidar), in order to have redundant information being lives and costly equipment at stake.

Following the main objective of the ENABLE-S3 project, one of the main targets of the farming use case has been the development of the necessary virtual tools to enable the validation and verification of the whole system. Therefore, the Agriculture Graphical

a) Drone scans the field.

b) Harvesters starts cropping tasks.

c) Obstacle detected by the drone.

d) Obstacle avoidance manoeuvre.

FIGURE 1.3: ENABLE S-3 farming use case scenario representation.

Simulation Environment (AgSim), has been used for the purpose of testing the control software of the machinery, i.e. harvester, tractor and drone, and the processing algorithms of the sensors on-board these machinery.

Additional to the research support this Thesis has provided to the ENABLE-S3 project, it has contributed to a national project named Distributed HW/SW Platform for Intelligent Processing of Heterogeneous Sensor Data in Large Open Areas Surveillance Applications (PLATINO) as well. The PLATINO project aims at providing a set of different solutions that enable further developments in the design of distributed hardware/software platforms able to smartly process heterogeneous sensor information. The feasibility of these solutions is assessed in a Smart Farming scenario, where the use of multisensory networks is a major breakthrough in terms of economy, but also in terms of food quality and safety.

In order to tackle this challenge, the PLATINO project focuses on five key factors that have been identified as essential for the use and management of the sensor networks:

- Automatic data acquisition using cutting-edge hyperspectral and photonic sensors.

- Development of new algorithms to efficiently extract relevant information from captured sensor data.
- Development of a secure network infrastructure to support safe/secure data transfers, and distributed data processing.
- Implementation of the processing algorithms and data fusion in specifically-tailored computing platforms, suited for different power consumption scenarios.
- Design of expert systems and mixed-reality user interfaces to enable smart decision-making processes, based on heterogeneous sensor data fusion.

1.2.4.2 Hyperspectral image acquisition on-board a UAV

Application-oriented solutions based on the combination of different unmanned aerial vehicles (UAVs), spectral sensors, precise positioning and orientation devices and embedded boards is exponentially attracting the attention of the scientific community. However, it is important to highlight that the whole system development entails some efforts that shall not be overlooked from its inception until the first valid data is collected.

Right at the beginning, before starting to even select the elements that comprise the whole system, developers should clearly identify the purpose of the system in order to achieve a proper selection. For instance, should the system perform some processing on-board, which will impact on the selection of the embedded system to be mounted. The type of drone to be selected, which shall mainly depend on the payload to be mounted and the missions to be performed. Finally, the type of spectral sensors that shall be installed, which may depend on the processing that is to be performed on the data.

As it has been already discussed, the main difference between multispectral and hyperspectral cameras resides on the number of bands captured by the sensors. While the multispectral sensors typically carried by UAVs collect just some spectral bands which shall serve for the purpose of calculating a reduced set of VIs, the hyperspectral sensors are able to sense hundreds of very narrow spectral channels, providing information that may be extremely useful not only for smart farming applications, but other remote sensing

applications as well. Acquiring hundreds of bands per pixel exponentially expands the use of the data compared to the few bands that are provided by multispectral sensors. In hyperspectral imaging, each pixel is well characterized by a unique signature, which not only enables the calculation of many more VIs than with multispectral technology, but permits the application of other techniques ranging from classification and segmentation to linear unmixing, supported by sophisticated methods such as machine learning and deep learning among others.

Although there are several references of different hyperspectral cameras installed on UAVs, the preferable option so far has been pushbroom imagers type [25, 26, 38]. These cameras acquire the hyperspectral cubes in a line by line fashion, which means that each Two-Dimensional (2D) captured frame contains in one direction the spatial information of a line of pixels, and in the other direction, the wavelength information which corresponds to those pixels. In order to generate a complete hypercube, the missing spatial dimension of the cube is captured along with the flying vehicle motion. Pushbroom cameras currently present a better trade off between spatial and spectral resolution, with a better range of prices available in the market. Nonetheless, their integration into a UAV is complex, since the frames captured share no overlapping, therefore much more care and attention has to be put into the acquisition and data pre-processing. In [39], a very precise positioning system was selected to acquire the camera orientation and position for every captured frame and in [40], a panchromatic snapshot sensor was incorporated so that through image processing a Three-Dimensional (3D) reconstruction can be performed and obtain the same information.

1.2.4.3 Anomaly Detection Algorithms

In recent years, the use of hyperspectral imagery to perform target detection and recognition has been widely investigated and has proven valuable in several domains including search-and-rescue operations, border surveillance, and mine detection. The goal of this technique is to search the image for the presence of a specific object/material. From

a theoretical point of view, target detection can be considered as a binary classification problem: the aim is to classify the image into the target class and the background class.

Anomaly Detection (AD) can be viewed as a particular case of target detection in which no a-priori information about the spectrum of the targets of interest is provided. The goal of anomaly detection is to find objects in the image that are anomalous with respect to the background, and represent a much lower portion of the image than the latter. Anomalies are generally identified as an observation that deviates in some way from the background clutter. Taking into account this concept, most of the AD algorithms principle is based on modelling the background with probabilistic distributions.

The Reed-Xiaoli (RX) algorithm [41] is considered to be the benchmark AD for multi/hyperspectral data and one of the first developments in this field. The RX assumes that the background follows a single Gaussian normal distribution and the probability density function is used to classify pixels as part of the background class. The Mahalanobis distance between the pixel under test and the background class is used to determine whether the pixel belongs to the background or is indeed an anomaly. The whole process relies on a well estimation of the background mean and the inverse covariance matrix; otherwise, they could be contaminated by anomalies causing a subsequent misclassification. Several variations of the RX detection technique have been proposed in the literature in order to improve its performance. Subspace Reed-Xiaoli (SSRX) [42] and Reed-Xiaoli after Orthogonal Subspace Projection (OSPRX) [43] are global anomaly detectors that apply Principal Component Analysis (PCA) or Singular Value Decomposition (SVD) to the datacube. The goal is to reduce the data volume to a smaller subspace where the first PCA/SVD bands are supposed to represent the background class. SSRX discards these bands, and then, RX is applied to the remaining subspace. On the contrary, OSPRX projects the data onto the orthogonal subspace before applying RX.

Due to the mentioned relevance of the subject, several authors have invested their time in compiling the most important algorithms in the field. A general overview of the current available methods is reported in [44]. A comparison of six relevant methods is performed in [45], using two well known datasets from the state-of-the-art. [46] groups into four

categories the AD algorithms, based on the underlying techniques used to achieve the detection.

1.2.4.4 Virtual simulation environments

Nowadays UAVs are widely utilized in many contexts for several applications, due to the operation flexibility proposed by such platforms and the fact that missions might be planned in a fully autonomous manner. It has to be considered that before the platform can be deployed for the actual mission, it is crucial to have the system fully tested against several possible scenarios to avoid having any system malfunctioning or even accidents.

Nevertheless, real experiments using UAVs are costly for many reasons. A proper flying location must be found and sometimes booked in advance, the heavy equipment have to be transported to the location along with the development team for the tests, at least a licensed pilot has to be present in order to be in charge of the operation, among others.

That is why the use of virtual simulation environments have become so critical, testing the performances of UAV systems before their deployments [47, 48]. There are two ways of system verification Software in the loop (SiL) and Hardware in the Loop (HiL) [49], depending whether just the software is being tested or some piece of hardware is connected to the simulation environment, respectively.

In general, simulation environments allow the repetitive testing of a system that becomes essential when dealing with autonomous systems. The amount of time and expenses that are saved thanks to such platforms is enormous.

1.2.4.5 Captured hyperspectral data processing

Hyperspectral imaging is a very powerful tool for many different applications due to the exhaustive and useful information it provides. In this work, the focus will be set into one particular use case though, precision agriculture. In this scenario, results provided by three different type of data processing are of special interest: VIs, classification and linear unmixing.

Vegetation indices

The generation of a set of VIs is one of the direct results that can be obtained from hyperspectral remote sensing images to provide information about the status of the crop [50]. This is performed by combining or transforming two or more spectral bands designed to enhance the contribution of vegetation properties, allowing reliable spatial and temporal inter-comparisons of terrestrial photosynthetic activity and canopy structural variations. One of the most typical examples of VI is the Normalized Difference Vegetation Index (NDVI), which indicates the vigorousness of the plants using the information of two different spectral channels placed in the Red and Near Infrared (NIR) parts of the electromagnetic spectrum [34].

Nevertheless, there are many more indices, in addition to the NDVI index, that provide useful information for smart farming applications [51]. For instance, given the limitations of NDVI under atmospheric effects, Kaufman and Tanré [52] proposed the Atmospherically Resistant Vegetation Index (ARVI), based on the knowledge that the atmosphere affects significantly the red part of the electromagnetic spectrum compared to the NIR. The Soil-Adjusted Vegetation Index (SAVI), originally proposed by Richardson and Wiegand [53], tries to enhance the distinction of vegetation from the soil background. There are even indices able to estimate the content of a particular substance in the plants, as it is the case of chlorophyll, estimated by the Chlorophyll Absorption Ratio Index (CARI) [54].

Hyperspectral image classification

HSI classification is defined as the assignment of each pixel in the image to a certain class based on its spectral characteristics. According to [55], classification constitutes nowadays one of the most vibrant field of research in the hyperspectral community and has drawn especial attention in the remote sensing field. In HSI classification tasks, there exist two main challenges: 1) the large spatial variability of spectral signatures and 2) the limited available training samples versus the high dimensionality of hyperspectral data. The first challenge is often brought by many factors such as changes in illumination,

environmental, atmospheric, and temporal conditions. The second challenge will result in ill-posed problems for some methods and reduces the generalization ability of classifiers.

In the early stage of HSI, classification methods have mainly focused on exploring the role of the pixel spectral signatures. Thus, numerous pixelwise classification methods such as neural networks [56], Support Vector Machines (SVM) [57], multinomial logistic regression [58, 59], and random subspace [60], have been proposed to classify HSIs. In addition, some other classification approaches have suggested a pre-processing of the data, designing an effective feature extraction or dimension-reduction, such as PCA [61, 62] or Independent Component Analysis (ICA) [63].

Recently, spatial features have been reported to be very useful in improving the representation of hyperspectral data and increasing the classification accuracies [64, 65]. More and more Spectral-Spatial Feature-based Classification (SSFC) frameworks have been developed, which incorporate the spatial contextual information into pixelwise classifier. This discovery, in combination with the growing trend of Deep Learning (DL) techniques, has enabled the appearance of new classification algorithms based on these methods, as evaluated in [66]. However, it has to be said that they rely on having enormous amount of data for its training which is unfortunately still not the case for some HSI applications.

Hyperspectral linear unmixing

Linear unmixing has become one of the most popular techniques to determine the content of a remotely sensed hyperspectral images [67]. It is based on the idea that each captured pixel in the image can be represented as a linear combination of a set of spectrally pure constituent spectra or endmembers, weighted by an abundance factor, which establishes the proportion of each endmember in the pixel under inspection. This linear mixture model assumes that the measured spectra can be expressed as a linear combination of the spectral signatures of materials present in the pixel, neglecting the secondary reflections and scattering effects coming from the data collection procedure. In those cases in which the impact of the latter is relevant enough, it is necessary to resort to complex nonlinear

models [68], which results in a further increase of the computational complexity of the unmixing process.

The linear unmixing process is usually divided in three different steps: estimation of the number of endmembers, endmember extraction, and abundance calculation. Additionally, there are other operations that can be performed and might provide enhanced unmixing results, such as noise filtering or dimensional reduction of the dataset. Many different algorithms have been proposed to solve the different steps of the unmixing process. For example, the virtual dimensionality algorithm [69] and the Hyperspectral Subspace Identification by Minimum Error (HySIME) algorithm [70] are popularly employed to estimate the number of endmembers. These algorithms aim at finding the intrinsic dimensionality of the image dataset, attending to different criteria such as the correlation between the bands or between the eigenvalues extracted from the dataset.

For the endmember extraction step, the Vertex Component Analysis (VCA) algorithm [71] and the N-finder algorithm [72] are frequently used. The N-finder algorithm selects the endmembers as the pixels of the image that form the simplex with the largest volume. On the other hand, the VCA algorithm extracts the endmembers as the pixels with the largest orthogonal projection, starting with a plane spanned by a number of random vectors that are sequentially replaced by the extracted endmembers. This algorithm iterates the number of endmembers to be extracted, computing for each iteration the orthogonal projection of each pixel of the hyperspectral image to the selected plane. Finally, the Fully Constrained Least Squares linear spectral Unmixing (FCLSU) algorithm [73] is popular for calculating the abundances of each endmember, minimizing the square error in the linear approximation of the hyperspectral image, imposing the nonnegative constraint and the sum-to-one constraint for the abundances.

Recently, new approaches have tried to contain all three defined steps in one implementation. For instance, the Fast algorithm for linearly UNmixing hyperspectral images (FUN) [74] does so, based on the concept of orthogonal projections. The FUN allows performing the estimation of the number of endmembers and their extraction simultaneously, using simple operations which can be also easily parallelized . Moreover, it is able to calculate the abundances based on the same principle, orthogonal projections.

The operations performed by the algorithm in all the steps are very similar, which makes it easier to achieve a hardware implementation of the entire unmixing process [75].

1.3 Research goals and contributions

The main objective of this Thesis is, through the processing of hyperspectral images captured on-board a UAV, to contribute the precision farming domain achieve a higher degree of automation in its processes, with the aim of increasing production in a sustainable way, help farmers take decisions with the additional information provided and avoid fatal accidents. Moreover, a new methodology for testing the hyperspectral real time algorithm implementations extensible to other application domains is proposed. The defined objective is divided in five main goals which are detailed next.

- Hyperspectral image acquisition on-board a UAV platform over agriculture fields. Missions shall be performed autonomously with none or very little interaction from the user during their execution. The platform flight control shall favour a proper capturing of the frames, trying to minimize the data pre-processing efforts.
- Preparing the acquired data for the future information extraction algorithms, by properly calibrating the sensed bitstreams, reducing noise in the captured signal and correcting possible misalignments.
- Developing a new implementation of an anomaly detection algorithm for its real time execution, adapting it to the constraints imposed by the camera capturing fashion. The algorithm implementation shall be able to process the data as soon as they are being captured.
- Developing a virtual environment for the verification of the algorithms for real time applications in a realistic scenario that virtually represents the developed hardware setup, based on a UAV with a hyperspectral sensor.
- Selecting and applying a proper set of algorithms to the acquired images in order to obtain results that support the defined objective. Colour maps based on vegetation

status, classification maps of the different type of crops, or the presence of anomalies in the field, among others, are the information demanded by the application.

The main contributions of this Thesis for accomplishing the defined goals are detailed below.

- A UAV platform based on a hyperspectral pushbroom camera has been developed. A commercial drone, the Matrice 600 from Dji, has been selected as a flying platform, to carry the Specim FX10 hyperspectral pushbroom camera, mounted on the Ronin Mx stabilizer, also from Dji, to soften vibrations and keep the camera continuously facing downwards. Flight missions are planned on the ground with a mobile device by a newly developed user friendly interface and then its execution is performed on-board with a control system that focuses on obtaining high quality images. Flight missions have been performed over vineyards, banana, avocado and mango fields, among others.

- A pre-processing algorithm chain has been proposed for preparing the data for the subsequent processes. Image calibration is accomplished using white and dark reference samples captured prior the flight. Camera radiometric calibration performed by the manufacturer is double-checked in the laboratory. Signal-to-noise ratio is obtained for all the captured spectral bands. Georectification is applied to the images in two steps. First each pixel of the image is geopositioned using the telemetry data captured with the positioning sensors available in the drone. Second, using the same location coordinates, image is geometrically corrected and laid on top of an available map of the inspected area.

- The *Line by Line Anomaly Detection* (LbL-AD) algorithm has been developed. A novel hyperspectral anomaly detection algorithm, specially conceived for being able to process the lines of pixels captured by a pushbroom scanner as soon as they are sensed. More concretely, the proposed algorithm is based, similar to the OSPRX algorithm, on the concept of orthogonal subspace projections but employing a low computational processing chain with hardware friendly operations, that guarantees a precise detection of the anomalies present in a hyperspectral scene.

- A hyperspectral simulation environment has been created. Built on top of the pre-existing AgSim, able to model and simulate machinery and its sensors, the proposed environment virtually creates the scenarios that permit the realistic testing of hyperspectral processing algorithms. Two main hyperspectral scenarios have been developed aided by the generation of synthetic images in two different fashions. On one hand, the scenarios created to test hyperspectral anomaly detection algorithms. On the other hand, the generation of hyperspectral maps based on a real grayscale bitmap overlaid on top of the virtual scenario, captured during simulated flight missions.

- A set of VIs have been applied to the captured images over vegetation fields to generate colour maps indicating plant stress levels, with the application of the NDVI, the chlorophyll content, with the application of the Modified Chlorophyll Absorption Ratio Index (MCARI), the distinction of vegetation from soil background, with the Modified Soil-Adjusted Vegetation Index (MSAVI) and Optimized Soil-Adjusted Vegetation Index (OSAVI), and the estimation of plant water stress with the Red-edge ratio. Classification maps to distinguish the crop and other elements in the scene have been obtained using a linear SVM classifier. Linear unmixing is applied to a portion of the image of which very little is known in advance, in order to figure out its composition. Finally, the image anomalies have been successfully detected with the developed LbL-AD algorithm.

Figure 1.4 graphically summarizes the main goals and contributions of this Thesis work.

Further contributions accomplished throughout the progress of the performed work are the following.

- The UAV platform based on the Specim FX10 camera has been developed in a product oriented way, following a modularity principle that permits the incorporation of other vision sensors in a very easy way. As a matter of fact, it has already attracted the attention of the industry for its possible future exploitation. For instance, a Non-Disclosure Agreement (NDA) has been signed with the camera manufacturer, Specim.

FIGURE 1.4: Main goals and contributions

- An application to be run on a mobile device, customized for the parametrization of the mission from the ground, has been programmed with a user-friendly interface that permits an easy interaction. Additionally, missions can be stored in memory for its repetition as many times as desired, in case the same terrain wants to be inspected several times under the same conditions.

- The developed anomaly detection algorithm have been exhaustively evaluated and compared against other state-of-the-art algorithms, with the purpose of both proving the accuracy of the results as well as the comparison in computational execution time. These comparisons result in a collection of different evaluation metrics.

1.4 Organization of this document

The present document is structured in seven chapters, including this introductory one, which is dedicated to present the main motivations and research goals of the Thesis work. The remaining six chapters are briefly introduced next.

1.4.1 Chapter 2: A UAV platform based on a hyperspectral pushbroom sensor

This chapter details the setup of a hyperspectral camera system into a UAV platform. First the selection of all major elements is detailed, then an overview of all involved components presented with the physical connections involved. Finally the software developments, supported by the Application Programming Interface (API) of the different platform component products, are presented and the intercommunications explained.

1.4.2 Chapter 3: Hyperspectral image acquisition

This chapter presents the data acquisition throughout the flight campaigns performed with the previously introduced platform. Hyperspectral images are presented along with additional data captured before and during the flight. Image pre-processing is performed in order to prepare the data for the subsequent processes.

1.4.3 Chapter 4: Hyperspectral anomaly detection algorithms

This chapter details the algorithm developed in this Thesis for finding anomalies in hyperspectral images. This algorithm, named *Line-by-Line Anomaly Detection* (LbL-AD) algorithm, has been conceived to uncover the anomalies in a line by line fashion, as soon as they have been sensed from a pushbroom sensor.

1.4.4 Chapter 5: Hyperspectral simulation environment

This chapter presents the hyperspectral simulation environment with the purpose of testing some of the algorithms prior to their execution in the real platform. Two main scenarios are created, one for the testing of anomaly detection algorithms and the other for testing the platform flight missions.

1.4.5 Chapter 6: Farming Use Case

This chapter presents the processing results on the captured hyperspectral images to provide useful information to a farming use case. First, the LbL-AD algorithm proposed in this Thesis is tested against the images captured by the hyperspectral platform, with real anomalies present in the field. Then, a set of VIs are applied to the images to obtain color maps indicating plant status. Finally, classification and linear unmixing algorithms are applied to distinguish between different elements in the scene and identify different vegetation types.

1.4.6 Chapter 7: Conclusions

Finally, the collection of contributions and achieved goals of this Thesis are summarized and further research work is proposed.

Chapter 2

A UAV Platform based on a Hyperspectral Pushbroom Sensor

This chapter covers in detail the development of a hyperspectral remote sensing acquisition system based on a UAV and a pushbroom sensor. First, the main elements that make up the whole system are introduced and the reasons for choosing each specific product properly argued. Then an overview of the whole system is presented, detailing the physical connections involved. Later, the software developments, supported by the main platform component products APIs, are presented and the intercommunication explained, paying special attention to the communication between ground and the flying platform. Finally, some flight mission results are reported.

2.1 Outline

Hyperspectral sensors on-board unmanned aerial vehicles (UAVs) are a very powerful tool in inspection, supervision and monitoring tasks due to their capability of, in a completely autonomous manner, covering large area surfaces and provide valuable information of the field that supports any decision or action to be taken.

However, in order to achieve the setup of a platform performing flight missions autonomously and acquiring high quality data, several considerations have to be taken into account beforehand. For instance, the selection of main elements involved, their physical interconnection (hardware integration) and their processes synchronization (software integration) have to be thoroughly thought and designed in advanced before jumping into the purchase of all components.

The flying platform definitely plays a key role in the system. It shall comply with the whole system weight requirement, it has to have sufficient autonomy to perform the missions and it has to provide an API to be able to control the platform from an external device. The hyperspectral sensor shall be selected carefully as well. Its operating spectral range, number of bands, spectral sampling and Field of View (FOV), among others, have an impact on the data that is going to be acquired and therefore have to be considered. The final main element is the embedded board to be installed in the platform. For this type of sensors to work, they have to be connected to a PC that performs the acquisition. It does also control the flight of the platform through the available API. Hence, hardware interfaces shall be checked, both between the board and the camera, and between the board and the drone. Another important aspect is the processing capability of the device, in case on-board processing shall be performed.

In this Thesis, a UAV platform based on a hyperspectral pushbroom camera has been developed. This chapter covers in detail the whole development process. First, the different spectral sensor technologies along with their capturing elements, suited to be mounted in UAV platforms, are discussed in Section 2.2. Next, a review of the state-of-the-art in multi and hyperspectral UAV platforms is presented in 2.3. The main system elements are introduced in Section 2.4, discussing the reasons why specific products have

been selected over other commercial available options. Later, hardware integration, both mechanical and electrical, is presented in Section 2.5. Then, the software integration is covered in Section 2.6. Platform flight telemetry data captured while preforming specific missions are evaluated and presented in order to assess the platform flight accuracy in Section 2.7. Finally, some conclusions are drawn in Section 2.8

2.2 Spectral sensors

The acquisition of the 3D image certainly poses a challenge for the engineers, since standard image sensors are all two dimensional. Thanks to the mentioned evolution in software and electronics, several different new options have appear throughout the years, opening nowadays a wide range of choices when it comes to select the device to utilize. The end application does of course play a very important role, since the mentioned available options have different characteristics that can be more suitable for a specific task.

In general terms, spectral sensors capture information in spectrally and radiometrically characterized bands. Commonly, they are distinguished by the wavelength arrangement and/or the number of bands [76, 77]. Furthermore, sensors can be classified on the basis of the method by which they scan the object spatially, and the method by which they obtain spectral information [78]. In [79] an overview of the state-of-the-art in hyperspectral image sensing is presented. Table 2.1 displays a summary of that state-of-the-art, some examples available in the industry of cameras following each technology principle and their main characteristics. In the following subsections further details of the technologies are given.

In Table 2.1, both spatial and spectral resolutions examples of the different type of sensors are given, with a qualification depending whether it is considered a good or poor resolution. Concretely, plus signs are used, 1 plus sign meaning a poor resolution, 2 plus signs meaning a fair resolution, 3 plus signs meaning a good resolution and 4 plus signs meaning an excellent resolution.

	Data Cube Slice	Scanning Dimension	Spatial Resolution	Spectral Bands	Example Sensors
Whiskbroom		spatial $[X, Y]$	none	++++ (224)	AVIRIS, Hyperion
Pushbroom		spatial $[Y]$	+ (1024)	++++ (224)	Specim FX10, Headwall Nano, Resonon Pika L, Specim FX17
Multi-camera		spatial $[Y]$	+++ (1280×960)	+ (5)	Micasense RedEdge-m, Parrot Sequoia, Tetracam Mini MCA
Sequential		spectral	+++ (1088×2040)	++ (12)	Rikola FPI VNIR, SpectroCam
Multi-point		none	+ (50×50)	+++ (125)	Cubert FireFly
Filter-on-chip		none	++ (512×272)	++ (25)	Imec SNm4x4, imec SNm5x5
Spatiospectral		spatial $[Y]$	++++ (2000)	+++ (160)	COSI cam Cubert ButterflEYE LS

TABLE 2.1: Spectral sensor types and characteristics.

2.2.1 Whiskbroom imagers

Whiskbroom imagers were the first hyperspectral device type utilized in remote sensing missions. It mainly relies on a complex optic system that uses a mirror to reflect light onto a single detector, which means that one single pixel, with its corresponding spectral bands, is captured at a time. In order to obtain the whole hypercube, motion is required in both spatial dimensions. For instance, the mentioned mirror rotates back and forth, to collect measurements in the X direction and for the displacement in the Y direction it is counted on the carrying aerial platform. For these reasons, this type of sensor is limited to be mounted on-board airplanes or satellites, which usually performed a very stable flight. As a matter of fact, whiskbroom imagers were the first devices capturing images on-board these aerial platforms with a very high spectral resolution. The already mentioned AVIRIS [6] or Hyperion [7] are of this type.

However, one of their biggest disadvantage is that the rotating parts make this type of sensor quite expensive and are more likely to wear out. For this reason, they are being replaced by newer approaches that do not require as much displacement to capture an image.

2.2.2 Pushbroom imagers

Pushbroom imagers record a line of spatial pixels, with its spectral information, at each exposure. Hence, with this type of devices, as opposed to whiskbroom cameras, motion is just required in one direction, in order to capture the complete datacube. The pushbroom design has been the standard design for large airborne imaging spectrometers for many years. However, recent miniaturization of the devices has enabled its mount into platforms such as UAVs, just able to lift lighter payloads of about 6 kg. Current state-of-the-art pushbroom cameras weight between 0.5–4 kg, including the lens, fulfilling the aforementioned requirement.

The benefits of pushbroom devices include a high trade-off between spatial and spectral resolution. Similar to whiskbroom cameras, when mounted on aerial platforms,

pushbroom systems need additional equipment on-board to enable accurate georeferencing of the acquired data.

They also rely on complex optics. Figure 2.1 shows the main elements of a pushbroom optical chain, that share a lot of commonalities with the whiskbroom. First, the light passes through the objective optics, an element that is also present in standard cameras. Then, the entrance slit just lets the electromagnetic energy through corresponding to the sensed line. In case of whiskbroom optics, instead of a rectangular slit there is just a point. Next, the beams out of the slit are aligned thanks to the collimator. This is followed by the key step, that consists in dispersing the incoming light into the different wavelengths, done by the dispersing grid. The rays out of the grid, are conveyed into a focusing lens, usually position in a certain angle with regards to the dispersing grid. Finally, the light is collected by the sensor, usually a standard 2D Complementary Metal–Oxide–Semiconductor (CMOS) panchromatic with high frame rate capabilities.

FIGURE 2.1: Hyperspectral pushbroom optical chain.

Pushbroom cameras are widely used in several applications of different domains, from remote sensing mounted on-board satellites, for instance in the Hyperspectral Digital Imagery Collection Experiment (HYDICE) [80], or drones [26]; to laboratory installation [81]. They present a good trade off between spatial and spectral resolution, with a good radiometric performance that results in datacubes spectrally well characterized. Additionally, they offer a competitive range of prices.

2.2.3 Multi-Camera 2D Imagers

A multi-camera 2D imager uses several integrated sensors to record a multispectral or hyperspectral image. Often, this is done by placing band pass filters with a specific wavelength configuration in front of the detector. In recent years, manufacturers have released different products following this principle. Among them, the most popular ones are the Parrot Sequoia+ [82] with four spectral bands, the MicaSense RedEdge-Mx [83] with five spectral bands and the MAIA camera [84] with nine spectral bands captured by separate imaging sensors operating simultaneously.

2.2.4 Sequential 2D Imagers

Sequential band systems record bands or sets of bands sequentially in time, with a time lag between two consecutive spectral bands. These systems have often been called image-frame sensors [85]. An example of such a system is the Rikola hyperspectral imager by Senop Oy [86] that is based on the tunable Fabry–Pérot Interferometer (FPI). The camera weights 720 g. The desired spectral bands are obtained by scanning the spectral range with different air gap values within the FPI [87]. The current commercial camera has approximately (1010×1010) pixels and 380 spectral bands in the range of approximately 500–900 nm.

Another type of sequential 2D imagers is a multispectral camera consisting in a continuously rotating wheel of filters with one single detector. Currently in the market the model SpectromCam [88] from Pixelteq, offers such capabilities, with a wheel containing 6-8 interchangeable optical filters to be mounted in five different cameras, with specific image sensors, cover the spectral range from 200 nm to 1700 nm.

Benefits of the sequential 2D imagers are the comparably high spatial resolution and the flexibility to choose spectral bands. At the same time, the more bands it has, the longer it takes to record all of them. In mobile applications, the bands in individual cubes have spatial offsets that need to be corrected in post-processing.

2.2.5 Multi-point spectrometer

Multi-point spectrometers use a beam splitter to divide the spatial image into sections, of which the signal is spread in the spectral domain [89]. An example is the Cubert Firefleye [90], that weights about 1 kg and records an image cube with a spatial resolution of 50×50 pixels with spectral information from 450–900 nm. Simultaneously, a one megapixel panchromatic image covering the same extent is acquired and can be used to fuse the images into a full scene [91].

The advantage of multi-point 2D imagers is that they record all the spectral information for each point in the image at the same time, and typical integration times are very short due to the high light throughput. However, the disadvantage is the low spatial resolution of the spectral information.

2.2.6 Mosaic filter-on-chip cameras

In the mosaic filter-on-chip technology, each sensor pixel carries a spectral filter built on chip, right on top of the detector, such as the principle of a Bayer pattern in an RGB sensor. The combined information of the pixels within a mosaic or tile, represents the spectral information of the area seen by the tile. The technology is based on a thin wafer on top of a monochromatic CMOS sensor in which the wafer contains band pass filters that isolate spectral wavelengths according to the FPI principle [92]. The wafers are produced in different spatial configurations, including linear, mosaic, and tile-based filters. This technique was developed by the Belgian research and development institution Imec [93], using the FPI filters to provide different spectral bands. Currently, the chip is available for the range of 470–630 nm in 16 (4×4 pattern) bands with a spatial resolution of 512×256 pixels, and in the range of 600–1000 nm in 25 bands (5×5 pattern) 409×216 pixels.

This type of sensing technology offers the obvious advantage of being able to record all bands at the same time. Additionally, they are very light, including casing and optics, their weight is below 400 gr., which makes them suitable for being carried by small UAVs. The

disadvantage is that each band is just measured once within each tile and thus, accurate spectral information for one band is only available once every few pixels. Currently, this is tackled by slightly defocusing the camera and interpolation techniques. They have a higher spatial resolution than multipoint spectrometers, but the radiometric performance of the filter-on-chip technology has not yet reached the quality of established sensing principles that is used in point and line scanning devices. This mainly results from the technical challenges during the manufacturing process (i.e., strong variation of the thickness of the filter elements between adjacent pixels) and the novelty of the technique.

2.2.7 Spatiospectral filter cameras

The spatiospectral cameras consist of a continuous filter in front of a 2D panchromatic sensor. The filter can either be built on-chip, the same way it was indicated in the previous subsection, or in a separate piece of glass located a few millimetres in front of the detector. The camera captures a full 2D image, where each vertical line of pixels correspond to a different wavelength. Hence, the spectral range is defined by the filter and the spectral resolution by the length of the sensor. In principle, on every exposure, the camera would sense the diagonal of an hypothetical hyperspectral datacube. In order the capture a whole hypercube, they require scanning over an area, similar to the operation of a pushbroom camera. This is why these sensors are also referred to as spatiospectral scanners.

The main advantage of this type of device compared to pushbroom imagers is that in the spectral dimension (corresponding to the length of the sensor), there is also spatial information available what enables the use of mosaicing techniques to compose the whole hypercube. On the other hand, similar to mosaic filters on-chip, the spectral characterization of spatiospectral filters built on chip has not yet reach high quality standards due to a challenging manufacturing process.

2.3 State-of-the-art in UAV Platforms based on multi and hyperspectral sensors

Traditionally, remote sensing tasks were carried out by satellites and manned aircraft platforms, with the benefits of supervising large extension of surfaces from a top view with highly advanced sensing technology. Nonetheless, they present three major constrains related to technological, operational and economic factors that limits their use for certain applications.

The emergence of Unmanned Aerial Vehicles (UAVs) has managed to fill the gap left by the mentioned traditional platforms, as highlighted in [94, 95], bringing in high spatial and temporal image resolutions, flexible sensor integration and much lower operational costs, broadening the application range within different areas such as defence, security, environmental protection, civil domains and agriculture, thanks to their highly adaptability to ad-hoc systems. In [96], a technical comparison between multi-rotor UAVs, manned aircrafts and satellites is presented, evaluating their cost-effectiveness trade off within the precision agricultural scope. UAVs were classified with the best flexibility, processing tasks, resolution and precision. However, there are some areas where they perform poorly compared to the other two, namely, the coverage range, flight endurance, mosaicking, and geocoding efforts.

The continuous and rapid development of such aerial vehicles combined with the exponential growth of sensors and instruments to be mounted aboard have led to the expansion of these systems, attracting special attention from the scientific community that is increasingly dedicating efforts to this field. For instance, in [97] a general overview of the UAV platforms and sensor technology is presented. [98] focuses on new aspects provided by the technology, such as ultra-high resolution, availability of coherent geometric and spectral data and multi-sensor data for fusion. [99, 100] review the broad range of practical applications and processing techniques.

As mentioned, UAV sensors are continuously experiencing technological improvements. Very often UAV imaging systems are understood just as drones with RGB cameras.

However, it is worth noting, that there exists a wide range of sensor options when considering professional applications. Many existing remote sensing instruments for aerial and satellite platforms have started to embrace their miniature and low-cost versions for UAV platforms, with significantly less payload capacity. Those sensor technologies are multispectral, hyperspectral, short/mid-wave range cameras (e.g., thermal) and light-weight Lidar. In the literature, several publications can be found which integrate those sensors into UAV platforms. Just to name some, [33, 101] present solutions of multispectral sensors dedicated to precision farming thorough the calculation of different proposed vegetation indices. In [102] a combination of a multispectral camera and a thermal sensor is being used in disease detection of opium vegetation. [103] presents an evaluation of Lidar sensors suitable for UAVs and the wide range of applications that appear thanks to the incorporation of these sensors into aerial platforms. In [104, 105], a UAV platform based on a miniaturized Lidar sensor was developed for high scale mapping.

Special attention should be paid to UAV platforms based on hyperspectral sensors, which offer a vast amount of information, indeed necessary for systems targeting different domains and applications within just one platform. The emergence of different hyperspectral sensor technologies has lead to the appearance of multiple systems in the literature incorporating such sensors. For instance, [17] includes a multipoint spectrometer camera, [106] a liquid tunable filter camera and [107] a spectrometer device. Nonetheless, it is worth mentioning that pushbroom acquisition still remains as the most used hyperspectral device technology with a better trade off between spatial and spectral resolution and better range of prices available in the market. Some examples of those sensors installed in UAVs are described in [25, 26, 38].

Making an exhaustive analysis in the literature of the available UAV platforms based on pushbroom hyperspectral cameras, it is noted that some of the major advantages offered by those systems are not fully exploited. For instance, most of the systems focus on the acquisition and do not incorporate an embedded board with processing capabilities to be able to provide real time results. In [108], this issue is addressed, presenting a solution including a data processing unit board aside from the hyperspectral acquisition hardware comprised of a frame grabber, a control board and a Central Processing Unit (CPU)

board. Real time results of a Spectral Angle Mapper (SAM) are unfortunately obtained at the cost of a high complexity system with a heavy payload.

Although based on commercial drones that incorporate several Software Development Kits (SDKs) for the vehicle programming, another missing aspect in the available systems is the fully integration of all the devices into a unique user-friendly easy-to-handle application, used to setup the missions and to parametrize the flight and elements of the platform, and afterwards start and run the mission in an autonomous way. Most of the implementations are based on two decoupled codes, one for the camera and one for the drone, in which their only way of interaction is through the flight coordinates. Missions have to be prepared in advance, in both codes. An example of such implementation can be found in [109].

Finally, nowadays these vehicles incorporate several sensors such as Global Positioning System (GPS) and Inertial Measuring Units (IMU), used for the flight control and post-processing of the data. However, in the literature, the flight performance based on the measurements of those devices is seldom covered in detail, although its analysis is in fact critical for the assessment of the acquired image quality.

2.4 Platform devices selection and description

This section thoroughly covers the hardware elements involved in the platform and their main characteristics. The hyperspectral camera, the drone and all its accessories, the embedded board, the mechanical stabilizer and the panchromatic frame camera are discussed. Decision criteria, the main advantages they offer and how they fit into the whole system are the main topics approached in the discussion.

2.4.1 Hyperspectral camera

Once the decision of using a pushbroom hyperspectral sensor over other hyperspectral sensor options has been justified in Section 2.2, the next point is to select the proper technology among the available pushbroom cameras in the market.

In order to narrow the search, the following requirement list has been created after analysing the targeted farming application.

- Spectral range. The scientific literature has dedicated some efforts in determining the optimal spectral range for agriculture applications [20, 24, 110]. All of them come to the conclusion that most of the biochemical and biophysical attributes of the crops are obtained in the visible and near infrared range. Proposed Vegetation Indices (VIs) found in the literature utilize bands that are comprised within that range as well. For that reason the selected sensor must be able to acquire information in the visible and near infrared, λ=400-1000 nm.

- Spectral bands and resolution. It is desired to have a considerable number of bands separated by a short spectral distance not only to be able to calculate as many VIs as possible, but to be able to obtain characterized spectral signatures per pixel as well. The scientific community has not yet agreed upon a fixed number of spectral bands that clearly separates multispectral from hyperspectral technology, although it is commonly said that hyperspectral sensors usually acquire hundreds of bands while multispectral only a dozen at most. In this work, a minimum of 100 bands have been selected to be sufficient to achieve the mentioned goals. This gives a minimum spectral resolution of 6 nm, to cover the whole aforementioned range.

- Spatial resolution. The acquired image spatial resolution depends upon a few factors, such as the flight height, the camera FOV and the spatial sampling (number of pixels per captured line). Considering that a reasonable flight height should be between 20 and 80 metres, the hyperspectral camera shall be able to provide spatial resolutions of a few centimetres per pixel, in order to be able to distinguish vegetation leaves, a few centimetres wide and long, in the acquired images.

- Hardware interface. As it has been already mentioned, the camera shall be connected to an embedded device that is in charge of its control. For that reason, it is desired a straight connection between the elements avoiding, for instance, frame grabbers that complicate the installation and adds up extra weight to the platform. Additionally, frame rates of up to 200 Frames per Second (FPS) are desirable, enabling the

possibility of flying at speeds higher than 2 m/s, which generally confers more stability to the platform. In this sense, there are two interfaces that fulfil the aforementioned requirements, USB3 and Gigabit Ethernet (GigE).

- Weight and dimensions. The camera shall be installed in a stabilizer that helps reducing vibrations and makes sure it is constantly facing downwards. In order to fit in any of the commercial available stabilizer, the camera should be a parallelepiped of maximum dimensions 100 mm width, 100 mm height and 180 mm length. Regarding the weight, it is desired that the whole flying platform payload comprised by the camera, stabilizer, embedded board and hardware interconnections, shall not be heavier than 5 kg, in order to fit well in commercial drone solutions available in the market. Therefore the camera shall weight less than 2 kg.

- API. The camera manufacturers shall provide as well an API compatible with the embedded board operating systems which are most commonly Linux based.

Evaluating the products the market has to offer, just a few of them fulfil the outlined requirements. Table 2.2 reveals the characteristics of the three different products that have been considered, the Specim FX10, the Headwall Nano and the Resonon Pika L. The term Full Width at Half Maximum (FWHM) represents the spread of the band around the wavelength center point.

The FX series hyperspectral pushbroom cameras from Specim have been specifically designed for industrial applications to enhance quality control processes. In particular, the FX10 whose characteristics are shown in Table 2.2, covers the VNIR range, from 400 nm to 1000 nm. With a weight lower than 2 kg, this product is also a feasible candidate to be installed on a UAV. It comes with a third party API, eBUS SDK, that can be installed in both Windows-based or Linux-based machines. Finally, it does also provide a package for hardfloat architectures that has been tested for a set of commercial embedded boards.

Headwall's Nano-Hyperspec is a completely integrated hyperspectral sensor designed for the VNIR (400-1000nm) spectral range as well. It includes an integrated data acquisition and storage subsystem of 480 GBytes and direct-attach capabilities for optional

Model	Specim FX10	Headwall Nano	Resonon Pika L
Spectral Range	400 - 1000 nm	400 - 1000 nm	400 - 1000 nm
Spectral Bands	224	270	281
Spatial Sampling	1024 px	640 px	900 px
Spectral Resolution	2.7 nm	2.2 nm	2.1 nm
Spectral FWHM	5.5 nm	6 nm	n.a.
Max Frame Rate	300 FPS	300 FPS	249 FPS
Bit Depth	12	12	12
FOV (α)	38°	5.8 - 26.3°	4.3 - 47.5°
Min Spatial Resolution (at 20 m height)	1.3 cm	0.3 cm	0.2 cm
Camera SNR (Peak)	600:1	n.a.	326:1
Camera Interface	GigE	GigE	USB 3.0
Dimensions	$150 \times 85 \times 71mm$	$76.2 \times 76.2 \times 120mm$	$100 \times 125 \times 53mm$
Weight	1.26 kg	0.52 kg	0.6 kg
Price	∼13000€	>50000€	>50000€
API	eBUS SDK	Hyperspec SDK	Resonon API

*n.a. = not available

TABLE 2.2: Hyperspectral pushbroom devices fulfilling the required characteristics.

GPS/IMU. Weight and size are minimized, making Nano-Hyperspec a good candidate for airborne sensor solutions. The manufacturer offers the possibility of choosing between a set of lenses with different focal distances from 8 mm to 35 mm, depending on the spatial resolution that wants to be achieved. It provides an acquisition software and an API, Hyperspec SDK, that is unfortunately only supported for Windows operating system machines.

The Pika L hyperspectral camera is a lightweight, small VNIR imager covering the 400 - 1000 nm spectral range as the previous two devices. It does only weight 0.6 kg, making

it ideal for drone-based remote sensing applications. As it was the case for the previous device, the manufacturer also offers for this product the possibility of selecting a lense with a FOV that ranges between 4.3 and 47.5 degrees. However, it includes an API, Resonon API, that is only supported for Windows operating system machines.

After having deeply analysed these three options, the sensor FX10 from Specim has been the chosen one. Considering that the three devices fulfil the set of requirements, the reasons for selecting the FX10 are: on the one hand, it is a much lower cost-approach compared to the others, and on the other hand it offers the required flexibility to be used in an open system under development.

However, it has to be taken into account that this is the first time that an industrial hyperspectral camera like this is installed in a UAV, and therefore, challenges are numerous, in terms of mechanical installation, communications, synchronization and control. Under well controlled conditions, such as in a laboratory, the integration is straightforward, but that is definitely not the case on-board an aerial platform.

2.4.2 Flying Platform. Drone.

As it has been previously discussed, there are two main type of drones that are most commonly utilized: fixed wing and multi-rotor drones. Due to the flexibility the second option offers, being able to complete a mission in a fully autonomous way (fixed wing solution is assisted in the take-off manoeuvre) and capable of hovering in a steady position for some time, multi-rotor drones has been the selected option.

Among multi-rotors there are several options in the market as well, and therefore, a requirement list has been produced in order to facilitate that selection.

- Maximum payload. In order for the system to acquire high quality hyperspectral images, on top of the hyperspectral sensor there are many other devices to be installed and hence, have to be lifted as well by the drone. As previously defined, altogether they shall make up to approximately 5 kg of weight, so in order to have some margin, the platform should have at least a 6 kg payload lifting capability.

- Autonomy. Flight missions of approximately 15 minutes of time are the target, as the fields that are going to be inspected have a maximum area of 15 hectares, which can be covered in that amount of time. Thus a set of batteries shall endure at least 20 minutes of the platform flying time with the maximum take-off weight.

- Controllability. The platform shall be externally controllable by an embedded platform, that through a serial connection communicates to the drone flight controller and issues predefined commands to perform a mission.

An approach that would definitely fulfil the aforementioned requirements is based on individually purchasing all the parts and putting them together. The exponential growth of these copters has led to the appearance of suppliers of the parts involved, for instance, a flight controller [111], motors and its drivers, batteries and a frame. [112] can be used as a guideline for product selection.

However, this option has been discarded due to the extra work that it would involve as well as a longer development time cycle that would not fulfil the project schedule. Therefore a commercial solution has been chosen.

Table 2.3 shows the characteristics of two products from the market that satisfy the defined needs: the Dji Matrice 600 and the FreeFly ALTA 6. The Dji Matrice 600 is an hexacopter with a payload lifting capacity of approximately 6 kg. Additionally, the manufacturer provides two SDKs to program missions either on a mobile device located on the ground and connected to the remote controller, or on a embedded board connected hard-wired to the drone controller, A3.

The Freefly ALTA 6 is an hexacopter as well with a payload lifting capacity of approximately 6.5 kg. The Pro version of the platform offers the possibility of installing the flight controller PX4 that can be programmed for instance using the DroneCode SDK running the code on a mobile device or on an on-board PC serially connected to the controller.

Among these two options, the Dji Matrice 600 drone has been selected for three main reasons. First, Dji has become the primary drone supplier in the market and therefore,

		Dji Matrice 600	FreeFly ALTA 6
General	Max take off weight	15.1 kg	13.6 kg
	Max speed (without wind)	18 m/s	15 m/s
	Number of Batteries	6	2
	Weight (with batteries)	9.1 kg	7.2 kg
Battery	Model	TB47S	ALTA Flight Pack
	Type	LiPo 6S	LiPo 6S
	Capacity	4500 mAh	10000 mAh
	Weight	595 g	1360 g
Motor	Model	Dji 6010	Freefly F45
	Kv	130 rpm/V	384 rpm/V
Propeller	Model	Dji 2170	Freefly ALTA Propeller
	Diameter	533 mm	457 mm
	Material	Carbon fiber reinforced	Carbon fiber with balsa core

Table 2.3: Drone main characteristics.

there is a huge community and support to all their products. Secondly, it has a considerably lower price including extras, 10000€, compared to the FreeFLY platform, whose price goes up to 15000€. Thirdly, it includes a wide range of products to enhance the capabilities of the platform. Some of them have been acquired and will be discussed next.

2.4.2.1 Precision GPS System: Dji RTK

Flying platform positioning and heading are very important for an accurate flight control, and useful as well for post-processing the captured data. Due to this reason, captured telemetry data are not only used during the flight for control but also stored into a file. In

this work, a precise device such as the Dji RTK [113] has been acquired in order to have an accurate positioning of the system. The device can deliver 1 centimeter of horizontal accuracy and 2 centimeters of vertical accuracy.

The presented device has two main drawbacks. On the one hand, its sampling frequency can not be set higher than 50 Hz. This means that for some flights there will be frames not positioned and hence, an ulterior interpolation is required. This limitation comes from the serial communication between the flight controller A3 and the on-board computer. On the other hand, its connectivity is very limited, which means that in the tested environments, most of the times, the device was not able to connect to enough satellites to achieve a proper position triangulation. Therefore, most of the flights were performed with the regular GPS from the Matrice 600 platform.

2.4.2.2 Camera Stabilization System: Dji Ronin-Mx

During the flight, the camera has to be always looking downwards aligned to the perpendicular direction of the captured surface plane. Ideally, any disturbance produced by flight vibrations shall be damped and not reach the camera in order to obtain high quality images. For that purpose, Dji has a product that fits straight into the Matrice 600 drone, easing the integration, and that is mainly intended for filming cameras. It is the gimbal called Ronin Mx [114]. In order to adapt it to hold the Specim FX10 camera a great deal of effort has been spent to design the mechanical elements, as it is explained in Section 2.5.

Dji provides a setup software to fine tune the Ronin Mx motor control in order to achieve the previously defined goal. Figure 2.2 displays the software that enables the setup of the tuning parameters. It has some basic functions that perform a standard calibration and it also grants access to the controller parameters. For instance, Figure 2.2a shows the standard parameters and how they have been setup and Figure 2.2b shows the advanced parameters and their setup. Special attention is to be paid to the tilt axis, where the camera has a relatively high inertial moment, which has to be compensated by the control.

a) Standard parameters.

b) Advanced parameters.

FIGURE 2.2: Ronin Mx configuration software.

The device incorporates a gyroscope which in this case has also been used to get the camera orientation, roll, pitch and yaw values. However, again the drawback is the maximum sampling frequency of 50Hz, when reading the values through the onboard SDK, as it is the case, and an accuracy of 0.1 degrees in all directions. Since having an accurate orientation of every captured hyperspectral frame is quite critical for geometric correction,

a second system based on an industrial RGB camera has been installed and it is explained hereafter.

2.4.3 Industrial Frame Camera

An industrial frame camera with a very high frame rate has also been incorporated into the system to assist hyperspectral frame registration and later correction. As a matter of fact three sensors have been alternatively used in the performed flying missions. The sensors come with a USB 3.0 interface and an SDK that allows its integration in real-time system platforms, being able to do parameter setup and image capture autonomously. These characteristics make the selected sensors optimal for this application.

Table 2.4 shows the main characteristics of the sensors. As it can be seen, the maximum resolution of all of them is over 1 mega-pixel, but since we are using the images for registration and the exact same area is going to be captured several times, the user is given the possibility to define an image spatial binning in both the horizontal and vertical directions, separately, thus potentially reducing the amount of information that the system has to handle during the flight mission.

Sensor Model	UI-3140CP	UI-3240LE	UI-3360CP
Sensor Type	CMOS Color	CMOS Mono	CMOS Mono
Resolution	1280 × 1024 Pixel	1280 × 1024 Pixel	2048 × 1088 Pixel
Optical Sensor Class	1/2”	1/1.8”	2/3”
Max. Frame Rate	224	60	152
Exposure time (min - max)	0.035ms - 434ms	0.009 ms - 2000 ms	0.025 ms - 500 ms
Sensor Interface	USB 3.0	USB 3.0	USB 3.0

Table 2.4: Industrial frame sensor characteristics.

Regarding the optics, as we are dealing with small sensors and it is desired to have a large depth of field and focus the camera to infinity, M12 optic mounts have been used, with focal lengths of 4.5, 6 and 7.5 mm alternatively.

The first tests performed with the sensors proved their high sensitivity, which means that they saturate very easily even when setting the exposure time to the minimum value and having an infrared filter mounted behind the optics. This poses a problem, since the images are to be taken at open air in very luminous environments.

For that reason, a filter that reduces the amount of radiance reaching the sensor has been mounted in front of the lens as represented in Figure 2.3. In Figure 2.3a, a picture of how the filter has been installed in front of the camera is shown. Two 3D printed pieces have been designed and made with a 3D Ultimaker 3 Extended printer, to hold the filter tight and fix the lens at a defined position when this has been focused. Figure 2.3b shows a wide variety of lenses with different sizes and spectral response curves, with a specific width and center position. In this case a filter of diameter size 25 mm has been used and transmittance curve centered at 550 nm and 20nm width.

a) Filter mount in front of the lens. b) Filter models.

FIGURE 2.3: Filter installation in the IDS camera.

2.4.4 On-board embedded platform

The market offers nowadays several options in terms of embedded boards to be mounted on such a flying platforms. However, not all of them fulfil the requirements the system has.

- Peripherals. The board shall have at least 1 GigE port to connect the Specim FX10 camera, 1 USB 3.0 port to connect the IDS camera, 1 USB 2.0 port to serially connect to the drone flight controller and 1 Serial Advanced Technology Attachment (SATA) port to plug in a SSD storage device.
- RAM. Hyperspectral images take up a huge amount of memory space. During the capturing process a certain amount of memory buffers are defined in order to avoid stalling. For that reason a minimum of 1 GByte RAM memory is required.
- CPU Cores. Since several processes are going to be running in parallel with a considerable computational burden, it is recommendable to have a CPU with at least 4 cores.

Three solutions have been acquired by the laboratory and thoroughly evaluated for their use in the flying platform. Table 2.5 displays their main characteristics. Two NVIDIA Jetson development kits, the TK1 and the TX2, super computers with a Graphics Processing Unit (GPU) based on the Kepler and the Pascal architecture, respectively. They both include all the basic functions to develop embedded applications and provide a NVIDIA Computer Unified Device Architecture (CUDA) platform with all the necessary tools to accelerate developments of high computational loads. The third option is the Odroid X4U, a new generation of computing device with energy-efficient hardware and a smaller form factor. It is based on the Samsung Exynos 5422 chip which includes a Mali-T628 MP6 GPU supporting OpenGL ES 3.0, OpenCL 1.1 as well as DirectX 11.

Both NVIDIA devices offer a more powerful GPU compared to the Odroid. The Jetson TX2 development kit is the largest of the three and the heaviest, making its integration in the flying platform more complicated.

		Jetson TK1	Odroid XU4	Jetson TX2
Chip	Model	Tegra K1 SoC	Samsung Exynos 5422	TX2 SoM
	ISA	ARMv7	ARMv7	ARMv8
CPU	Processor	ARM Cortex-A15	ARM Cortex-A15 + ARM Cortex-A7	ARM Cortex-A57 + NVIDIA Denver2
	N cores	4	4 + 4	4 + 2
	Frequency	2.32GHz	2.1GHz + 1.4GHz	2GHz
GPU	Family	NVIDIA Kepler	Mali	NVIDIA Pascal
	Model	GK20a	T628 MP6	—
	Cores	192 SM3.2 CUDA cores	6 clusters	256-core Pascal
	GFLOPS	326	122	1500
Mem	RAM	2GBytes 64-bit DDR3L	2GBytes 64-bit LPDDR3	8GBytes 128-bit LPDDR4
	Storage	16GBytes eMMC 4.51	32GBytes eMMC 5.0	32GBytes eMMC 5.1
Peripherals	GigE	1 RJ45	1 RJ45	1 RJ45
	USB 3.0	1 Type A	2 Type A	1 Type A
	USB 2.0	1 Micro USB	1 Type A	1 Micro USB
	SATA	1 full size	—	1 full size

Table 2.5: Embedded boards main characteristics.

After having tested all the boards, it was finally decided to use the Jetson TK1. First, it presents a good trade-off between size and weight, and the number of available computational resources. Secondly, it proves to integrate easier the hyperspectral camera driver, eBUS SDK version 4.9, than the other two boards.

As represented in Table 2.5, the Jetson TK1 has the required interfaces to build up this platform. It comes with GigE port RJ45, a USB 3.0 port, a USB 2.0 port to make the serial connection with the A3 controller through a USB-TTL device and finally, a SATA interface that allows the integration of an external SSD for a very fast data transfer to

memory, which is a key point in this type of applications handling an enormous amount of information.

Although these interfaces shall suffice the system, it was realised that an extra USB would be needed when for instance, an external USB modem has to be connected for debugging purposes. Figure 2.4 shows the additional hardware that was purchased to expand the USB 3.0 ports. Making use of the fact that the Jetson TK1 comes with a Mini PCIe port, a Syba 19-Pin USB 3.0 Header Mini PCIe card, Figure 2.4a, was acquired together with a female USB 3.0 cable SD-MPE2014, Figure 2.4b, that provides two additional USB 3.0 ports.

a) Syba 19-Pin USB 3.0 Header Mini PCIe card. b) female USB 3.0 cable SD-MPE2014.

FIGURE 2.4: Hardware devices to provide two additional USB 3.0 ports to the Jetson TK1.

At this point, it is worth saying that there are other alternative boards in the market that would also meet completely or partially the defined set of requirements and provide some extra advantages. As a matter of fact, during the development of this research, the Jetson Nano was released, a compact solution with similar processing resources as the Jetson TX2. The board was indeed acquired by our group and preliminary tests to be performed in the laboratory have been already defined.

Another interesting approach is to use a Field-Programmable Gate Array (FPGA) based board, whose power consumption is much lower than the GPU based ones. Recently, new devices based on the Zynq 7010 or Zynq 7020 chips have been released, which are as well compact solutions that would definitely fit on the drone. Just to name some examples, the PYNQ board in its Z1 and Z2 versions, the Zynq Berry or even a more sophisticated

one such as the Zedboard. Our group has recently acquired units of all the mentioned boards and now it will start evaluating its use in the drone for specific projects and tasks.

2.5 Platform devices interconnection. Hardware integration.

To the author's knowledge, this is the first work carried out to integrate a Specim FX10 camera, mainly intended for laboratory used, into a Matrice 600, typically used and adapted for filming cameras. For this reason, quite a lot of efforts had to be taken in terms of the design of mechanical pieces and electrical connections, in order to create a proper functioning system.

Figure 2.5 shows a general overview of the platform elements and how they have been connected.

The first step is to accomplish a proper mechanical integration of the camera into the gimbal. As indicated, the Ronin-Mx can easily accommodate any filming device such as the Canon 5D MK IV, Panasonic GH3 and GH4, Nikon D800, among others, but the integration of the Specim FX10 is not as straight forward. Some of the structural elements of the gimbal have to be replaced for new ones in order to properly hold the camera tight. The following design considerations have been taken into account to replace these components:

- Camera degrees of freedom for position adjustment. It is critical that the system is properly balanced in off mode, so when the gimbal is turned on, the motors are not continuously correcting the deviations and hence, consuming extra energy, plus the risk of having vibrations. The designed components allow therefore to adjust the camera position in the x direction (flying direction), y direction (pointing right of the flight direction), and z direction (vertical axis).

FIGURE 2.5: General overview of the UAV platform components.

- The camera is by default pointing downwards. Since the system is designed to scan fields beneath the drone, by default it makes sense to have the sensor already targeting the scanning surface.

- Weight optimization. It is important to keep the system as light as possible, so the components design provides a good trade off between weight and resistance to hold everything tight.

Figure 2.6 shows a 3D layout of the mechanical elements designed and created with the 3D printer, already introduced in Section 2.4.3, to fit the gimbal and replace some of the existing structural components.

As it is seen in Figure 2.6, mainly the bottom and top original plates of the Ronin Mx have been replaced by new custom ones, keeping the original vertical rods that close up the whole structure. Since the original bottom plate contains the electrical ports and

(a) Front view. (b) Rear view.

FIGURE 2.6: Gimbal construction elements breakout.

the gyroscope, the new custom made one has been developed to be able to contain these elements as well, which are critical for the functioning of the system. The custom designed plate to hold the camera grabbing it from the top also has the holes in place to fit the on-board system, i.e, the Jetson TK1 board. On top of this board another element has been designed and fixed into the same holes as the Jetson TK1, to bear the SSD device where images are stored after being captured.

The second aspect of the design integration are the electrical connections and communication links. The FX10 camera has to be supplied with an input voltage within the range $12V \pm 1V$, and the Jetson TK1 board spans the limits into a slightly broader range $12V \pm 15\%$. The Ronin Mx provides two output connections out of the 4S Lithium Polymer (LiPo) 1580 mAh battery mounted in it and a voltage regulator that supplies at $13V$. This value has been measured for reassurance throughout the battery discharge cycle and it is kept constant thanks to an internal regulator.

Initially, both the camera and the board were supplied from the Ronin-Mx at $13V$. The Jetson TK1 has experienced no issue at all, since this voltage is well within the specification limits. However, in the case of the camera, this voltage is indeed in the limits. Therefore, the decision of supplying the camera from an external LiPo battery

was taken. Since there is no commercial battery that supplies at exactly $12V$, a 4S Lipo battery was purchased, with a nominal voltage of $14.8V$, and a voltage regulator as the one indicated in [115], to bring the voltage down to the specified $12V$.

After having defined the devices supply, it is just a matter of making the physical connection. The output port in the Ronin Mx is Dtap female and the input port in the Jetson TK1 is a standard 2.1mm DC barrel plug. The external LiPo battery has on the other hand, a XT60 female output connector, that is plugged into a XT60 male cable, fixed on the other side to the voltage regulator input. From the voltage regulator output a cable comes out with a Fisher-type S1031-Z012-130+ connector, to fit to the input of the Specim FX10.

The communication cable between the FX10 and the Jetson TK1 is a standard Cat 6a ethernet cable with a male RJ45 connector on the board end and a male M12 connector on the camera end. Finally, the serial connection between the Jetson TK1 and the A3 controller from the Matrice 600 has been implemented with a USB-TTL converter connected to the board and the jumpers connected to the A3. The reason why a USB-TTL solution was used instead of the serial port available in the General Purpose Input Output connectors (GPIOs) of the Jetson TK1 has been mainly the voltage level difference from both. The Jetson TK1 serial output voltage level on the GPIOs is $1.8V$ and the one on the A3 works at a standard value of $3.3V$.

The whole assembly of the gimbal with the discussed elements is presented in Figure 2.7. The front view is displayed on Figure 2.7a, and the top view on Figure 2.7b.

Figure 2.8 shows the system totally assembled and flying during one of the flight campaigns performed the 24th of July 2018 in Tejeda, Gran Canaria, Spain.

(a) Front view. (b) Top view.

FIGURE 2.7: Gimbal assembly.

FIGURE 2.8: UAV flying platform.

2.6 Application for controlling the system. Software integration.

In this section a detailed description of the developed application for controlling the system is presented. The implemented software makes use of different SDKs for controlling the individual components. The main applications involved in the whole process are enumerated below:

- End-user application running on an iOS device, programmed in Objective C and based on the Dji Mobile SDK [116].
- Flight control application running on the Jetson TK1, programmed in C++ and based on the Dji Onboard SDK [117].
- Camera control application. All camera units connected to the board shall share the same interface and functions structure, so their integration is straight forward. In this case, two camera devices have been connected:
  - FX10 camera control application running on the Jetson TK1, programmed in C++ and based on the eBUS SDK from Pleora [118].
  - IDS camera control application running on the Jetson TK1, programmed in C++ and based on the IDS uEye SDK [119].
- Communication protocol. A communication protocol layer on top of Dji Lightbridge communication link has been created, to guarantee a proper wireless data transmission between the end-user application running on the mobile device located on the ground and the on-board device.

2.6.1 End-user application

The end-user application purpose is triple. On one side, it receives the basic inputs from the user, thanks to a graphical interface, to perform the first flight mission calculation. After these calculations are completed, the mission waypoints are presented on the map, defining the flight swathes. Finally, during the flight, the end-user application communicates with the on-board application to provide valuable information to the user regarding its progress. Figure 2.9 shows a few snapshots taken from the end-user application. In Figure 2.9a, the user is defining the scouting area by selecting two corners in the diagonal of the rectangle, considered as region of interest (ROI). Figure 2.9b lays out the mission waypoints after having applied all user defined parameters. Figure 2.9c shows the user mission parameters window and Figure 2.9d indicates the status of some of the system variables, in this particular case, the charging state of the batteries.

(a) Input initial coordinates.

(b) Waypoint layout.

(c) Mission parameter setup.

(d) System parameter overview.

FIGURE 2.9: iOS application screenshots.

2.6.1.1 User inputs

The first step is to define the area to be inspected by the drone. The application allows the user to provide this area in two different ways: by selecting the two corners in the diagonal of the rectangle (see Figure 2.9a); or by directly selecting all the corners of an irregular area, in which case the implemented code will calculate the rectangle with minimum size including that area.

Next, the user defines some required information to be able to perform the calculations and start the mission. For instance, the sensors to be enabled and capturing during the mission, the camera sensor binnings and two out of the three main mission parameters:

- Relative height, h, from the ground
- Speed
- Frames per second, FPS

These parameters are linearly dependent, so that the third one not defined by the user is obtained based on the following relationships:

$$\text{resolution} = \frac{spatial_sampling}{2 \cdot h \cdot \tan \frac{\alpha}{2}} \tag{2.1}$$

$$\text{speed} = \text{resolution} \cdot FPS \tag{2.2}$$

Equation 2.1 defines the resolution based on the camera and flight parameters, such as the camera *spatial_sampling* and field of view α, and the flight height h. The *spatial_sampling* depends on the hyperspectral camera spatial binning. The default value for the binning is 1, which means that 1024 pixels will be captured per frame.

Figure 2.10 shows a schematic representation of the flight parameters previously introduced.

2.6.1.2 Waypoint layout

After the user has defined all inputs and the remaining parameters have been obtained by the application, it is time to lay down the mission waypoints (Figure 2.9b), for which again the resolution obtained in Equation (2.1) is essential together with the overlap between different swathes. The calculations performed by the application are explained in the following four steps:

FIGURE 2.10: UAV flight schematics.

1. First, the rectangle sides are obtained in meters based on the GPS coordinates of the corners, and the shortest distance is selected to lay down the waypoints in between those corners. In this way, each individual flight swath has a bigger length and therefore, the flying time is reduced.

2. Second, the image width in meters is obtained by multiplying the resolution by the spatial sampling of the camera.

3. Third, the image width is corrected by the user input overlap, following Equation 2.3

4. Finally, the number of waypoints to be added in between the corner waypoints is calculated.

$$width' = width \cdot \left(1 - \frac{overlap}{100}\right) \tag{2.3}$$

$$N_{\text{waypoints}} = \left(\frac{side_distance}{width'} - 1\right) \tag{2.4}$$

where *side_distance* is the distance in meters of the shortest rectangle side. It is obvious that this calculation not always provides an integer number, so it is rounded up to the next integer value, giving as a result a larger overlap between images than what was initially defined.

Based on all the mentioned inputs, the application defines a rectangle and the intermediate waypoints to cover the area fulfilling the user requirements. It does also take into account the remaining battery percentage and in case the calculated remaining flight time is less than the estimated time to complete the mission, it warns the user and suggests to increase the altitude or the speed.

Once the calculation is consolidated, it is sent together with the defined user input parameters to the application running on the Jetson TK1, making use of the built-in functions in both the Mobile SDK (MSDK) and Onboard SDK (OSDK) from Dji and the already mentioned protocol running on top of those SDKs which will be presented afterwards in the section.

2.6.2 Flight control application

As it has already been mentioned, the application running on the Jetson TK1 uses the Onboard SDK from Dji for the purpose of implementing the drone flight control and the communication with the device on the ground. The application follows the steps described next.

1. The first step of the on-board application, after the board has successfully initialized, is to start the listening task implemented with a callback function that is constantly listening messages from the mobile application and implements the mentioned protocol in order to receive the mission setup parameters.

2. Once all parameters and required information have been successfully transferred and after the user has hit the *start* button, the listening task is closed and the main task together with the camera capture tasks are started and run in parallel. The main

task, responsible for the drone flight control, is also in charge of collecting telemetry data. The flight mission starts, following the steps that are described next.

(a) The take-off command is issued, followed by the drone elevating itself up to 2 meters above the ground, and hovering in that position.

(b) Then the function Z-movement is called with the relative height above the ground given by the user as an input.

(c) At this point, when the UAV has reached the user defined height, the RGB sensor is calibrated in order to obtain good quality images. For this purpose, images are being automatically captured while adjusting the sensitivity and exposure time until a defined pixel average value range has been reached.

(d) Afterwards the program enters into the waypoint loop, consisting in first orientating towards the next waypoint, calling the yaw control function.

(e) Once the drone is oriented, it starts moving towards the next waypoint. It must be pointed out that a relative coordinate system is being used, with distances defined in meters and with the system reference fixed at the drone starting position. Since the waypoint coordinates are defined in World Geodetic System (WGS), they have to be converted to the new system. Another point to be taken into account is the fact that for the hyperspectral capturing process it is crutial that images are captured at a constant speed.

(f) The waypoint application is running inside a control loop revisited every 5 ms that is constantly checking whether the next waypoint coordinates have already been reached. Some measures against overshoot have also been implemented in order to make the drone stop as closest to the waypoint as possible.

(g) If it is the last point, the drone moves back to its initial position. Otherwise steps d to f must be repeated until the last waypoint is reached.

(h) Once the drone is located back in its initial coordinates (where the system reference axes were located), Z-movement function is called again to descend and finally the drone lands, completing the whole mission in a fully automatic fashion.

3. After the flight is over, the board gathers all data and copies them into a newly created folder in the SSD disk, closes up the flight control application and restarts again the listening function, so the user can perform another mission or simply shut down the system.

It is important to highlight that after initialization, each task in the board runs in parallel in a different thread and all running threads are executed in the four cores available at the board CPU in an optimal manner to allow a smooth system overall functioning. The drone flight control is in charge of implementing the synchronization between itself and the camera capture tasks. After the drone has reached the first waypoint of an even swath the main task will trigger the capturing start of all camera sensors which internally are in charge of performing the capture and saving it in memory. When the drone reaches the next waypoint (end of the swath) the main task issues a stop command to the camera tasks. Additionally, the main task will store telemetry data in memory during those swathes, synchronized with the camera tasks. In this way only in the desired sectors data is being captured and the whole system performance and memory space is optimized. Finally once the drone has reached the end of the mission the main task will issue the command to stop the capturing process and close the camera tasks.

The entire application has been implemented in a modular way in order to allow future inclusion of additional sensors. This means that the camera tasks have been developed sharing a common interface, so the main task interaction with any camera already mounted and/or to be mounted on the drone is exactly the same. This interface includes parameters such as frame rate, exposure time, horizontal/spatial and vertical/spectral binning.

For inter-task communication and synchronization purposes, a shared memory has been created allowing the exchange of parameters between the processes being executed.

2.6.3 Camera control application

In order for the system to function in a modular way, all camera modules that are connected to the board should have the same interface so their integration is completed following a few steps and with very little effort. In this particular case, two camera units have been integrated to the system, the Specim FX10 and the IDS frame camera. Figure 2.11 shows a simplified version of the camera control code flow diagram.

FIGURE 2.11: Camera control code flow diagram.

The program is launched after the callback, described in Section 2.6.2, point 2, has finished its execution and the mission is ready to start. This means that the user has defined all required parameters for the proper parametrization of the camera.

The first step of the camera code is to look for a connected device. If for any reason, no device is connected to the application (the cable was left unplugged, the power supply is faulty, etc.), an initialization error is produced and the program exits. If on the other hand, the initialization succeeds, the next phase starts, which is setting the user defined parameters. These are, among others, the frame rate, the exposure time, and the region of interest or binning factors.

Later, the acquisition process starts. This depends on the signals from the flight control application described in Section 2.6.2, that will issue start, stop and finish commands using the share memory registers created for that purpose. The camera application code simply acquires a new frame if the flag is set to start acquisition, and save it in memory. It keeps doing so, so long the flag remains in acquisition mode, whenever the flag changes to stop acquisition, the code will stall, for another acquisition start to be issued or for a finish program command, which will be sent when no more images are wished to be captured.

An important remark in the described process is the use of a different thread for the image save task, just to avoid that once the next capture cycle has arrived and the process being executed in the previous thread has not yet finished, there is no system delay and images are properly stored in memory. A memory buffer with a capacity of up to 32 frames has been defined so that each thread can handle a different captured frame. The images are all saved with an index that identifies when it was captured so there is no confusion when a future reconstruction process is to be accomplished.

2.6.4 Communication protocol

One of the main advantages of using Dji products is that it provides a wireless communication link, called Dji Lightbridge [120], which allows high definition video transmission and supports remote controls using the 2.4GHz frequency band.

Both the MSDK and OSDK incorporate functions to transmit and receive messages from an on-board and a mobile device, respectively. Those functions have been used in this

development to exchange mission data information and debugging parameters, as it was explained in the previous section. In the early stages of the development, while performing some flight tests with the platform, it was realized that some packages were missed or sent/received incomplete, which was causing system malfunctioning.

In order to reassure the proper transmission of information a communication protocol has been developed on top of the aforementioned SDK functions. Figure 2.12 displays the message structure, which has been divided in three blocks. The first block, which takes up just a single byte of space, is a code providing information of the subsequent block of data. The values that can be assigned to that code parameter are described at the bottom of Figure 2.12. For instance, when equal to 1 ($CODE_CONTROL$) a system action will proceed, i.e. board shut down, restart, reload default configuration or initiate communication handshake. From $11 - 16$, mission configuration parameters are being transmitted, flight altitude, frame rate, speed, different thresholds and IDS and Specim FX10 camera configuration parameters, among others. Finally, when the code is equal to 255, all mission waypoints are transmitted and after that the board finishes the callback and starts the mission. As it can be observed, there are unused numbers to be assign to the code, in case the communication wants to be extended to new sensors to be mounted on the platform, or extra parameters.

The second block of data corresponds to the information bytes that want to be sent to the on-board device. Parameters are sent in float representation, meaning that each parameter takes up four bytes of memory, therefore the number of bytes in the second block must be a multiple of four. The last block of data is a so called checksum value, which represents the sum of all bytes in the second block of data, taking as well four bytes of memory space.

The whole communication process works in the following way. In the mobile application, a First-In-First-Out (FIFO) buffer is created that stores the messages generated from the user mission configuration. For instance, when the user sets mission parameters, debugging parameters, wants to issue a communication handshake or shut down the board, among others, a new message with the structure presented in Figure 2.12 is created and buffered into the FIFO.

FIGURE 2.12: Communication protocol message structure.

As long as there are messages in the FIFO, an internal function controlled by a timer makes sure these are sent to the on-board device. Whenever a new message is transmitted, the system waits 10 seconds for an acknowledgement from the on-board device, in order to delete the message from the buffer and go on with the next one. If after the defined time there is no confirmation received, or instead an error message from the on-board device is issued, the message is kept in the buffer, so it will be sent again in the next iteration.

From the on-board device perspective, the communication process is represented in the flow diagram displayed in Figure 2.13.

The OSDK callback function is initialised and the system remains waiting for incoming messages. Once a new message arrives, the first thing the program does is dividing it in the three blocks shown in Figure 2.12. The last four bytes contain, as mentioned, the checksum value, calculated on the mobile device. So the first step is to check whether in the on-board device that sum gives the same value in order to proceed decoding the message, or not, what would bring the program to the initial state and send an error message to the mobile device.

In the decoding phase, the program checks whether the code parameter is a valid one in which case it proceeds sorting the information. If that code parameter does not exist in

FIGURE 2.13: Communication flow diagram.

a predefined list, an error is sent back to the mobile device and again the program goes to the initial position.

Finally, once the message information has been sorted the system goes back to listening new incoming messages, unless the previous one was a shut down, restart or mission waypoints command, in which case the callback is finished and the next action in executed.

2.7 Flight Performance

One of the main goals of the proposed platform is to fine tune the flight control software in order to capture high quality hyperspectral images and reduce the georectification efforts in the image correction phase, which will be discussed in detail in Chapter 3.

The designed software controls position and speed on every loop that runs every 5 ms to obtain a stable flight at an almost constant speed, which is very important for the captured samples. The trajectory is constantly adjusted as well, keeping the drone on track throughout the whole mission. The system minimizes overshot in position when reaching a waypoint, and in angle when rotating. This has been achieved by defining extra control parameters that have been carefully selected through several tests, both in the provided Dji simulation environment and in real flights. These parameters can be also adjusted by the user, depending on the weather conditions, and send them from the mobile to the on-board device.

This section presents the flight mission performance results. In order to evaluate such performance, first, some metrics are introduced. Then, coordinate conversion formulas are presented and finally, the results are displayed both graphically and numerically.

2.7.1 Assessment Metrics

Some metrics are proposed here to asses the quality of the flight control, based on position and orientation data acquired during the mission.

First the real drone trajectory is compared with the ideal trajectory defined by the imaginary straight line that connects the initial and final waypoint of each sector. The metric Root Mean Square Error (RMSE) has been selected for this assessment.

$$\text{RMSE} = \frac{1}{N} \cdot \sqrt{\sum_{i=1}^{N} (d_i \perp)^2} \tag{2.5}$$

In Equation 2.5, N represents the number of data samples available during the trajectory and $d\perp$, the perpendicular distance from the real trajectory curve to the ideal trajectory line, in meters.

Another important flight variable that is monitored throughout the whole sector is the flight height above ground. This parameter is directly related to the resolution of every pixel. Ideally, it should remain constant at the value defined by the user. Since this is not the case in reality, the deviations are evaluated using the mean, μ, and the standard deviation, σ, metrics, shown in Equations 2.6 and 2.7.

$$\mu = \frac{1}{N} \cdot \sum_{i=1}^{N} (h_i) \tag{2.6}$$

$$\sigma = \sqrt{\frac{1}{N} \cdot \sum_{i=1}^{N} (h_i - \mu)^2} \tag{2.7}$$

In Equations 2.6 and 2.7, the variable h_i corresponds to the height above ground measured by the platform on each iteration. Orientation angles of the camera have a huge impact in the quality of the images. On the one hand, the gimbal angles, sensed by the gyroscope located in the lower plate of the Ronin-Mx, are analysed. On the other hand, the heading of the vehicle, which should remain stable, aligned to the trajectory defined by the initial and final waypoints in each sector. The drone heading is measured using the geographic coordinates, with the North as reference. The heading evaluation is done using the mean and standard deviation values, as represented in Equations 2.6 and 2.7, substituting in this case h_i for the corresponding angles in each iteration. The gimbal angle assessment metrics will just be the mean value since the variation is rather low.

Finally, the speed should as well remain at the set value, with an acceleration phase at the beginning of each sector and a deceleration phase at the end. The speed analysis is once again accomplished calculating the mean and standard deviation values over each sector.

2.7.2 Geographic units conversion

The drone navigation system acquires the positioning of the platform following the standard WGS in its latest version of 1984, WGS84 [121]. For each point a latitude and longitude is recorded.

In this work, the data captured from the GPS device is converted to a geometric coordinates system using the equirectangular approximation, based on the Pythagoras' theorem, and the assumption of a spherical earth, which is accurate enough for most purposes.

$$\Delta x = R(Latitude_{target} - Latitude_{current}) \tag{2.8}$$

$$\Delta y = R(Longitude_{target} - Longitude_{current})cos(Latitude_{target}) \tag{2.9}$$

Equations 2.8 and 2.9 express the relative distances in the two directions, x and y, corresponding to north and east, respectively. R is the earth radius approximated to the value 6378137 meters.

2.7.3 Flight mission results

Several flight campaings have been performed with the platform over different regions in the island of Gran Canaria. Details about these flight campaigns will be provided in Chapter 3. Here, an exhaustive analysis of the flight data is presented though.

It is important to highlight that for the data analysis a software package in Python language has been developed, supported by the Matplotlib library [122] for the plotting functions. The main purpose of the created code is to read all mission generated logging files, evaluate the numeric position and orientation data read during the flight from the corresponding sensors together with the time stamp at which every capture takes place and generate a report that first displays the mission parameters, then plots different charts

of the variables under analysis and finally evaluates the previously introduced assessment metric over each individual sector of the mission.

Next, telemetry data captured over two different regions in the island of Gran Canaria are being compared. The first region, a football field within the premises of the University of Las Palmas de Gran Canaria campus, is located in the North-East of the island, where the climate conditions are relatively tough for a flying platform, with strong gusts of wind and illumination changes due to partial clouds. The second region, a vegetation field in a place called Veneguera, located in the South-West of the island, where the climate conditions are very stable with low wind speed and clear sky.

Figure 2.14 shows telemetry data captured for the two different missions performed in the regions just described. Coordinates have been converted to a geometric system using Equations 2.8 and 2.9. The ideal trajectory is shown in red, and it is defined by the iOS application based on the area selected by the user. In blue, the real trajectory profile is represented, where at each individual data position a perpendicular line to the ideal trajectory is drawn. GPS longitude and latitude coordinate values have been converted to meters to create the charts. In the representation, the initial point of the first sector has been set as the origin and the X axis corresponds to the North in the real world.

In Figure 2.14a, the trajectory plot of the mission performed over the football field is shown and in Figure 2.14b, the trajectory plot of the mission performed over the Veneguera fields is displayed. As it can be appreciated from a first visual inspection, the deviations from the ideal trajectory are more pronounced in the flight over the field with tougher climate conditions. However, the real trajectory generally behaves quite well.

The platform height above ground should also be taken into analysis as it might have an impact in the image scale and add distortions. Figure 2.15 shows the height above ground measured over the sectors of the flight mission 2 performed over the football field within the University Campus. The sensor measuring the altitude in the drone provides the value above sea level, which means that the initial altitude number has to be stored throughout the whole mission, to be able to plot the values as represented in Figure 2.15. As it can be appreciated, the height stays stable at almost 72 meters for the assessed mission.

a) University Campus football field mission 1

b) Veneguera mission 3

FIGURE 2.14: Flight trajectory of two different missions performed in two different regions of Gran Canaria.

FIGURE 2.15: Platform height above ground of mission 1 performed over the University Campus football field.

Changes in the vehicle heading affect the quality of images as well, as it adds undesired rotations and therefore distortions. Figure 2.16 displays a visual representation of the platform heading angle relative to the North, over the three sectors that comprise mission 2 performed in Veneguera. As expected, two sectors are heading into the same direction, and the sector in the middle heads into the opposite direction, 180 degrees shifted from the other two.

Figure 2.17 shows the gimbal roll, pitch and yaw angles, over all three sectors of mission 2 performed in Veneguera. The yaw angle is measured from gimbal front axis to the drone flying direction axis. Ideally it should be as close to zero as possible. As it is seen, the yaw angle starts with a considerable deviation due to the abrupt rotation the drone does when it reaches the waypoint and orientates itself to the next waypoint. This deviation is slowly corrected in the drone by the software until it reaches an offset of about 5 degrees.

The absolute speed of the drone over the five sectors of the flight mission 1 performed over the Veneguera vegetation fields is plotted in Figure 2.18. As it is appreciated, the vehicle takes a couple of seconds to achieve the user set speed of 6 m/s, in the acceleration phase

FIGURE 2.16: Platform heading of mission 2 performed over the Veneguera vegetation fields.

FIGURE 2.17: Gimbal orientation angles of mission 2 performed over Veneguera vegetation fields.

that takes places in the first couple of seconds and then the speed is kept rather constant until the deceleration phase starts, right before the next waypoint is reached.

FIGURE 2.18: Platform speed of mission 1 performed over Veneguera vegetation fields.

The results presented on Figures 2.17 and 2.18 suggest that the frames captured in the proximities of the waypoints should be cropped out since they are going to be more distorted than desired. It is indeed recommendable to set the waypoints out of the limits of the region of interest to be monitored. The rest of the flight looks very stable and therefore high quality images are expected.

It is important to mention that the maximum sampling frequency telemetry data can be obtained is 50 Hz, while the cameras are able to reach frame rates above 200 FPS. That means there is not telemetry data available for each individual frame and interpolation has to be performed.

Numeric results of the two missions performed in the football field at the University campus are displayed in Table 2.6 and Table 2.7, and the results corresponding to the three missions performed over Veneguera vegetation fields are displayed in 2.8 and 2.9.

The corresponding mission is indicated in the left hand side of the table with the letter "M" followed by the mission number.

		Trajectory $[m]$		Altitude $[m]$			Speed $[m/s]$		
		RMSE	max	target	μ	σ	target	μ	σ
M1	Sector 1	1.90	4.97	60	59.38	0.41	3.03	3.03	0.24
	Sector 2	1.30	2.87		60.41	0.36		3.03	0.22
	Sector 3	1.71	3.22		59.54	0.45		3.03	0.22
	Sector 4	1.20	2.57		60.29	0.43		3.06	0.23
M2	Sector 1	0.78	1.52	72	72.28	0.45	3.63	3.68	0.48
	Sector 2	0.37	0.77		72.22	0.42		3.58	0.42

TABLE 2.6: Flight mission data captured over the football field in the University Campus.

		Roll $[deg]$	Pitch $[deg]$	Yaw $[deg]$	Heading $[deg]$		
		μ	μ	μ	target	μ	σ
M1	Sector 1	0.0	-0.11	-5.36	46.36	45.75	7.95
	Sector 2	0.0	-0.1	5.63	-133.64	-132.06	6.41
	Sector 3	0.0	-0.07	-5.07	46.32	45.52	6.53
	Sector 4	0.0	-0.1	5.75	-133.67	-132.30	6.57
M2	Sector 1	0.0	-0.11	6.15	46.27	46.88	4.67
	Sector 2	0.0	-0.28	6.12	-133.74	-133.77	4.69

TABLE 2.7: Table 2.6 continuation.

		Trajectory $[m]$		Altitude $[m]$			Speed $[m/s]$		
		RMSE	max	target	μ	σ	target	μ	σ
M1	Sector 1	0.79	2.33	60	60.03	0.36	6.05	5.98	0.58
	Sector 2	1.83	4.49		59.97	0.25		6.02	0.46
	Sector 3	1.04	1.73		60.01	0.37		6.02	0.52
	Sector 4	1.74	2.69		59.92	0.27		6.03	0.47
	Sector 5	0.66	1.48		59.98	0.38		6.01	0.53
M2	Sector 1	2.19	3.81	72	71.92	0.26	6.05	6.04	0.41
	Sector 2	2.15	3.38		71.90	0.41		6.03	0.46
	Sector 3	2.38	4.29		71.96	0.25		6.03	0.42
M3	Sector 1	1.56	2.59	45	45.06	0.26	4.54	4.54	0.43
	Sector 2	0.52	1.16		44.89	0.36		4.53	0.51
	Sector 3	0.99	2.62		44.92	0.25		4.53	0.41
	Sector 4	0.88	2.05		44.80	0.37		4.58	0.5

TABLE 2.8: Flight mission data captured over a Vegetation filed located in the west side of the island of Gran Canaria.

A few considerations about the presented values. The calculation of the mean and standard deviation values of the speed have been performed leaving out the initial and

final phases, just keeping the stable period. The heading calculation is done using the flying coordinates and compared to the reference value obtained from the mission waypoint coordinates. Since the coordinate resolution is low a smoothing pattern had to be apply to the calculation by just selecting one coordinate out of ten, otherwise too many peaks were affecting the final obtained value.

		Roll [deg]	Pitch [deg]	Yaw [deg]	Heading [deg]		
		μ	μ	μ	target	μ	σ
M1	Sector 1	0.02	0.2	5.21	-164.01	-162.89	19.0
	Sector 2	0.0	-0.14	-5.08	15.99	15.89	2.87
	Sector 3	0.06	0.06	5.22	-164.01	-163.91	2.75
	Sector 4	0.0	-0.15	-5.07	15.99	15.92	2.29
	Sector 5	0.05	0.11	5.32	-164.01	-162.82	19.10
M2	Sector 1	0.0	-0.16	-4.87	36.66	36.64	2.31
	Sector 2	0.03	0.14	5.29	-143.35	-143.32	2.49
	Sector 3	0.0	-0.08	5.15	36.64	36.72	2.76
M3	Sector 1	0.0	-0.25	-5.45	36.65	36.64	3.08
	Sector 2	0.05	-0.09	5.52	-143.35	-143.34	2.92
	Sector 3	0.0	-0.16	5.61	36.65	36.86	4.41
	Sector 4	0.08	-0.13	5.55	-143.38	-143.44	3.48

Table 2.9: Table 2.8 continuation.

In general, as already perceived in the displayed charts, the flight accuracy achieved by the platform is high, with low deviations in all analysed parameters. The focus in the future work has to be set to reducing the gimbal yaw offset and reducing trajectory deviations in adverse climate conditions.

2.8 Conclusions

The development of a hyperspectral UAV platform for image acquisition has been fully described. The platform is made up of a pushbroom camera, a drone, an embedded board and a full frame camera, among others. All of them have been carefully selected following a set of predefined requirements based on the target application.

The components have been integrated to form a functioning platform. First, mechanical adaptation of the stabilizer plates have to be carried out in order to fit the hyperspectral

camera and other elements into the system. Then, electrical connections were engineered from the batteries to the components to achieve a stable supply. Communication cables were also laid down to enable data exchange and synchronization.

In order to complete the integration, different software applications were programmed. An end-user application running on the mobile device located on the ground to configure the mission, a flight control application to execute and control the flight mission and camera applications to perform the image acquisition have been developed. All flight mission configuration parameters and device parameters are to be set in the developed user friendly application running on the mobile device. Mission is started and supervised from that terminal as well. This is a considerable improvement to the solutions reviewed from the literature where devices and platform are decoupled and have to be configured beforehand.

All on-board tasks, flight control, frame acquisition and data storage are carried out in one embedded device which makes the system quite compact and means yet another novelty compared to the state-of-the art solutions. On top of that, the board has sufficient processing capabilities to perform on-board real time algorithms in parallel with the aforementioned tasks.

A novel software package has been implemented to evaluate the flight telemetry data captured by the positioning and orientation sensors aboard, display the necessary diagrams to asses the accuracy of the performed missions and calculate the assessment metrics over all mission sectors. The presented results demonstrate that the platform flight accuracy is considerably high, what definitely translates into high quality image acquisition, reducing the efforts in the post-processing phase. Moreover, additional data is acquired, such as a full frame image, that combined with the platform position and orientation would potentially support a future correction phase.

Hence, it is concluded that the developed platform represents an advanced solution compared to the available ones in the state-of-the-art, for the developed user friendly application, its flight accuracy, what turns into high quality image acquisition process,

for its capacity of performing on-board processing and for the developed software package that generates mission reports analysing the flight accuracy.

Finally, it is worth mentioning that minimal changes in the platform presented in this Chapter could lead to other interesting applications related to environmental studies or mineralogy, just to name some, which gives an added value to this prototype in terms of flexibility, modularity and robustness.

Chapter 3

Hyperspectral image acquisition

This chapter presents the hyperspectral images acquired with the UAV flying platform introduced in Chapter 2, during the performed flight campaigns. A pre-processing algorithm chain is proposed to be applied to the acquired data in order to prepare them for the subsequent processing algorithms. In this sense, image calibration, camera radiometric calibration, Signal-to-Noise Ratio (SNR) estimation per spectral band and georectification are presented.

3.1 Outline

Between the captured raw data by the flying platform and the utilization of the images by algorithms such as anomaly detection, vegetation indices, classification maps and so on, there is a set of steps that need to be carried out in order to prepare the data. Figure 3.1 shows an overview of the proposed chain.

The aim of remote sensing devices is to collect the energy reflected from the Earth's surface. However, the sensor just collects digital signals in a range defined by the pixel data resolution, i.e. number of bits per pixel. Hence, in order to obtain the desired reflectance, an image calibration process has to be accomplished first. This essentially consists in transforming the measured radiance to reflectance values using a white reference measurement, which corresponds to the maximum energy a material reflects, and a dark reference measurement, which provides the offset values some pixel may have. These references are captured at the same conditions than the camera performs the acquisition during the flight mission.

Another important pre-processing that comes in a further step is the radiometric calibration of the sensor which basically consists in determining the exact wavelength value that corresponds to each value in the Z dimension (spectral dimension) of the image. This permits the inter-comparison with other images and spectra from different types of sensors.

The next step is to perform a signal to noise ratio estimation for all captured bands, to check whether the sides of the sensor experience a decay in that ratio, which could eventually suggest the elimination of such bands from the images.

Finally, a georectification process is performed using the data captured by the additional sensors in the flying platform. The intention is to determine the geographic coordinates of each individual pixel in the image, in a so-called georeferencing phase, and later a correction of the distortions in the captured swathes produced by the disturbances on the platform during the flight. This last phase receives the name of geometric correction.

The rest of the Chapter is organized as follows. First, the performed missions are presented in 3.2, in which the three main locations where flight campaigns have taken place are introduced, then flight missions are outlined and finally the captured images are displayed. The processing steps are described, starting by the image calibration process in 3.3, continuing with the radiometric calibration of the sensor in 3.4, followed by the SNR estimation in 3.5 and finishing with the georectification phase in 3.6, split in two parts, the georeferencing, explained in 3.6.1 and geometric correction, detailed in 3.6.2. Finally, conclusions are outlined in 3.7.

FIGURE 3.1: Proposed pre-processing chain.

3.2 Flight missions: captured data

One of the main objectives of the platform development, covered in detail in Chapter 2, was the creation of a user friendly system that allows the performance of several flight missions in different locations and even the possibility of repeating them in the exact same conditions if desired.

This has been achieved and hence, has given the possibility of flying in different locations performing numerous flight campaigns. In this section, the performed flight missions are introduced with details such as, location, number of waypoints and mission parameters, among others.

Figure 3.2 presents an overview of the island of Gran Canaria, highlighting the spots where flight campaigns have been carried out. As it can be seen, the platform has been flown over three different regions: (1) The football field next to the place where we have our offices, at the University Campus, (2) a region called Veneguera, located in the South-West of the island, where different sorts of vegetation coexist, and finally, (3) a vineyard located in the center of the island, in a region called Tejeda, where two different terrains have been analysed.

FIGURE 3.2: Flight Mission locations. (a) Overview of Gran Canaria. (b) Football field in Campus Tafira. (c) Veneguera. (d) Tejeda vineyard location 1. (e) Tejeda vineyard location 2.

Before starting the presentation of the acquired data in the different performed missions, it is worth clarifying that for a better document arrangement, the RGB representation of the captured images, that take up a considerable amount of space in the document, have been included at the end of the Chapter. This encompasses from Figure 3.19 to Figure 3.25.

3.2.1 Campus Tafira: Football field

Being the closest open area available next to our research labs, the football field has been used as a testing area, when new implementations wanted to be validated on the real world. Its exact location lies on coordinates 28°4'15.51"N 15°27'26.56"W. Out of all the performed missions over the football field, two have been selected to be shown here, carried out the 26th of May 2019.

Besides the fact of being a close location, other important advantages of the University Campus is its weather, much different than the climate conditions available at the center or west part of the island, where the other flight campaigns have been carried out. The weather at the football field is characterized by relatively strong gusts of wind and low illumination due to clouds. This poses a challenge since on the one hand, it is desired to fly at high speeds that confer more stability to the platform which is advantageous, given the wind disturbances, but on the other hand, the low luminosity limits the camera frame rate and hence, the mentioned platform mission speed.

Therefore a compromise has been targeted, as displayed in Table 3.1, where the performed mission parameters are shown. Frame rate has been set to 75 FPS in both missions, achieving flying speeds higher than 3 m/s. The covered area is very reduced, since the objective of these missions is not to cover a large area but rather to test the platform behavior in the mentioned severe conditions.

	Height	Frame rate	Speed	Resolution	Overlap	Area	Waypoints
Mission 1	60 m	75 FPS	3 m/s	0.04 m	25 %	0.6 ha	8
Mission 2	72 m	75 FPS	3.6 m/s	0.048 m	50 %	0.5 ha	4

Table 3.1: University Campus football field missions setup.

Figure 3.3 displays on top of a Google Earth map, the mission waypoints for mission 1 and 2, represented in 3.3a and 3.3b, respectively.

In Figure 3.19, a false RGB representation extracted from the hyperspectral data acquired during mission 1 is displayed. The mission consisted of 8 waypoints hence, 4 swathes have been captured, with approximately 3200 frames per swath. This results in approximately

a) Mission 1. b) Mission 2.

FIGURE 3.3: Football field mission outlook.

1.4 GBytes and a total of 128 m length and 41 m width coverage per swath. The mission took approximately 7 minutes to complete from take-off until landing.

In Figure 3.20, a false RGB representation of the 2 hyperspectral swathes acquired during mission 2 is displayed. Each swath is made up of approximately 2000 frames, resulting in approximately 875 MBytes and a total of 96 m length and 50 m width coverage per swath. The mission took approximately 4 minutes to complete.

3.2.2 Veneguera: Mango, banana and avocado fields

As part of the Farming Use Case of the ENABLE-S3 project, the aim of the construction of the proposed platform is to use it in the precision agriculture domain. That is the reason why flight campaigns have been carried out over Veneguera, a region in the island of Gran Canaria located in the South-West at coordinates 28°4'15.51"N 15°27'26.56"W. The fields belong to a local company, with whom the University has signed a collaboration agreement, and is used to harvest the fruits and vegetables, that are later supplied to hotel resorts located on the South of the island. Three different fields have been inspected in that region, as displayed in Figure 3.4, where mission waypoints have been laid on top. In the field presented in Figure 3.4a mango and lemon trees can be found, whereas in

the field shown in Figure 3.4b banana trees are the type available. Figure 3.4c shows a clearing, that has been recently planted with avocado trees. Missions were carried out the 25th of July 2019.

a) Mission 1.

b) Mission 2.

c) Mission 3.

FIGURE 3.4: Veneguera mission outlook.

Acquired images during the first mission performed over Terrain 1 are shown in Figure 3.21. This figure displays a false RGB representation extracted from the hyperspectral

data aquired for each swath. Main flight parameters are presented in Table 3.2, where it is shown that the flight mission consisted of 10 waypoints which provides 5 swathes as it is represented on Figure 3.21. The number of frames per swath is between 10300 and 10400, what results in approximately 4.5 GBytes per swath. Considering the ground sampling distance of 4 cm, this gives a total of 415 m length and 41 m width coverage per swath. The mission took approximately 15 minutes to complete from take-off until landing.

Images from the second mission are shown in Figure 3.22. Main flight parameters are presented in the second row of Table 3.2, where it is shown that the flight mission consisted of 6 waypoints which provides 3 swathes. The number of frames per swath is between 10200 and 10400, what results in approximately 4.5 GBytes per swath. Considering the ground sampling distance of 4.8 cm, this gives a total of 500 m length and 50 m width coverage per swath. The mission took approximately 12 minutes to complete from take-off until landing.

Finally, images captured in the third mission are shown in Figure 3.23. Main flight parameters are presented in the last row of Table 3.2, where it is shown that the flight mission consisted of 8 waypoints which provides 4 swathes. The number of frames per swath is between 5650 and 5750, what results in approximately 2.4 GBytes per swath. Considering the ground sampling distance of 3 cm, this gives a total of 168 m length and 31 m width coverage per swath. The mission took approximately 9 minutes to complete from take-off until landing.

	Height	Frame rate	Speed	Resolution	Overlap	Area	Waypoints
Mission 1	60 m	150 FPS	6 m/s	0.04 m	10 %	10 ha	10
Mission 2	72 m	125 FPS	6 m/s	0.048 m	10 %	14.5 ha	6
Mission 3	45 m	150 FPS	4.5 m/s	0.03 m	10 %	1.8 ha	8

Table 3.2: University Campus football field missions setup.

3.2.3 Tejeda: Bentayga Vinyard

Bentayga vineyard is located in Tejeda, in the center of the island of Gran Canaria, at an altitude of approximately 1300 meters above the sea level. The vines planted on the

fields belong to a local company, with whom the University has signed a collaboration agreement, and is used to grow the grapes, that are later used to produce the Agala wine. Figure 3.5 shows the two Google Earth pictures of the terrains under analysis with the waypoints overlaid on top. Figure 3.5a shows terrain 1, whose exact coordinates are 27°59'35.6"N 15°36'25.6"W and Figure 3.5b shows terrain 2 whose coordinates are 27°59'15.2"N 15°35'51.9"W.

a) Mission 1. b) Mission 2.

FIGURE 3.5: Tejeda mission outlook.

Acquired images during the first mission performed over Terrain 1 are shown in Figure 3.24. This figure displays a false RGB representation extracted from the hyperspectral data aquired for each swath. Main flight parameters are presented in Table 3.3, where it is shown that the flight mission consisted of 12 waypoints which provides 6 swathes as it is represented on Figure 3.24. The number of frames per swath is between 4100 and 4200, what results in approximately 1.9 GBytes per swath. Considering the ground sampling distance of 3 cm, this gives a total of 125 m length and 31 m width coverage per swath. The mission took approximately 12 minutes to complete from take-off until landing.

Acquired images during the second mission performed over Terrain 2 are shown in Figure 3.25. This figure displays a false RGB representation extracted from the hyperspectral data aquired for each swath. Again, main flight parameters are presented in Table 3.3, where it is shown that the flight mission consisted of 10 waypoints which provides 5

swathes as it is represented on Figure 3.25. The number of frames per swath is between 4900 and 5000, resulting in approximately 2.3 GBytes per swath. With a ground sampling distance of 4 cm, the total coverage per swath is approximately 200 m length and 41 m width. The mission took approximately 9 minutes to complete from take-off until landing.

	Height	Frame rate	Speed	Resolution	Overlap	Area	Waypoints
Mission 1	45 m	150 FPS	4.5 m/s	0.03 m	40 %	1 ha	12
Mission 2	60 m	200 FPS	6 m/s	0.04 m	40 %	1 ha	10

Table 3.3: Tejeda, Bentayga Vinyard missions setup.

3.3 Image calibration

The raw captured frames by the sensor are a measurement of the sensed light per sensor pixel, each pixel value expanding from 0 to 4096 considering the camera pixel depth of 12 bits. However, sensor response is not uniform across the covered spectral range. The consequence is that, even if the same amount of radiance is hitting all the pixels of the sensor, the digital value measured in each pixel will be different, especially for different wavelengths. Additionally, the illumination conditions may not be uniform across the covered spectral range. These facts make not possible to directly use the raw images, which are affected by the sensor response, for the subsequent hyperspectral imaging applications. For instance, Figure 3.6 displays the spectral response of a certified Zenith Polymer white panel that reflects 99% of the incident radiation in the VNIR range [123]. In Figure 3.6a a captured hyperspectral frame taken with the Specim FX10 over the whole spectra is displayed. As it can be observed in this image, the sensor varies with the spectral wavelength. Additionally, the response of different pixels that measure the same spectral wavelength also varies, what typically results in stripping noise [124].

The material manufacturer, Sphere Optics, provides the reflection curve, displayed in Figure 3.6b. Although very close to 100%, the curve shall be used to correct the white reference measurement.

a) Image taken with the Specim FX10.

b) Material reflection characteristic.

FIGURE 3.6: Spectral response of the white reference panel measured.

In order to solve these issues, the captured images are converted to reflectance values, in such a way that each image value is scaled between 0 and 1, representing the percentage of incident radiation that the scanned object reflects at each specific wavelength. The procedure for doing so is explained next. Prior to the mission flight, an image of the Zenith Polymer white calibration panel is acquired from the ground, as the one shown in Figure 3.6, just using the gimbal with the camera pointing downwards at a distance of around 1 meter. It is important to make sure that the camera line of pixels is entirely

sensing the white panel and that no pixel is left out. Up to 50 samples at the exact same frame rate and exposure time, as the one used during the flight, are taken and then averaged to have a final reference that is used for calibration. A dark sample is also acquired. This dark reference collects the minimum values that the sensor measures when no radiance is hitting it. In order to obtain the dark reference, the camera lens is completely closed. Again, 50 samples are taken and then averaged.

$$\text{reflectance} = \frac{\text{sensed_bitarray} - \text{dark_refence}}{\text{white_reference} - \text{dark_reference}} \tag{3.1}$$

Equation 3.1 shows how the raw data is calibrated for obtaining its corresponding reflectance values. Figure 3.7 shows an example of how a set of real hyperspectral signatures collected by the described acquisition system are calibrated using the described procedure. In particular, Figure 3.7a shows the average value of the white and dark references across all the sensor pixels. Figure 3.7b shows a portion of the first image swath displayed in Figure 3.24, corresponding to mission 1 performed over the vinyard in Tejeda, from which some pixels have been selected, corresponding to vegetation, soil and shadows, marked in the image in green, blue and red colours, respectively. The raw and calibrated spectral signatures of these pixels are displayed in Figures 3.7c and 3.7d. As it can be observed in these graphs, the raw spectral signatures are strongly affected by the sensor spectral response.

Images displayed in Figure 3.19 to Figure 3.25 have all been calibrated with the corresponding white and dark references.

3.4 Radiometric calibration

Radiometric calibration of the hyperspectral camera assigns to each captured band, a specific wavelength. This in turn allows the use of images from different platforms as long as the acquisition system is properly calibrated, being in that case the data extrapolation straight forward.

a) White and dark signatures

b) Selected pixels

c) Raw signatures

d) Calibrated signatures

FIGURE 3.7: Pixel calibration for the real acquired hyperspectral data.

Several factors influence the radiometric calibration, from the lens to the sensor. That is why camera manufacturers must provide a specific calibration file per device. Any slight adjustment, in the optics for instance, requires the repetition of the calibration process.

In this case, the camera manufacturer, Specim, has performed the radiometric calibration using a Oriel Instrument's spectral calibration lamp [125], which contains a small amount of mercury that dominates the output spectrum, and argon as starter gas. The process is based on the so-called irradiance technique in which the camera is irradiated directly by the mentioned lamps, whose spectral characteristic is known in advance.

The setup was done using an integrating sphere, with the lamps fitted inside and the spectrograph positioned in such a way that the exit aperture of the integrating sphere fills

its Field of View (FOV). A total of 20 frames were collected, averaged and dark reference was subtracted. From the collected data, 12 spectral peaks were located, as displayed in Figure 3.8, and a Gaussian function was used to fit to each peak. Center position and FWHM of the spectral peaks were obtained from the fitted data. Wavelength calibration was calculated using a 2nd degree polynomial fit to the spectral peak positions.

FIGURE 3.8: Measured spectral lamps at the center column. Calibration peaks are marked with red circles.

At our labs, we do not have at our disposal irradiance lamps to perform the presented procedure and double check the wavelength values provided by the manufacturer. However, what we do have, is a certified material that presents a characteristic spectral response and a setup with a linear drive and a lamp with a homogeneous emission throughout the whole spectra. Figure 3.9 displays the material acquisition by the Specim FX10. In Figure 3.9a, it is shown how the camera has been mounted in the lab for the acquisition and in Figure 3.9b, the Zenyth Polymer Material from Sphere Optics is displayed. As it can be appreciated the material has a circular shape with a diameter of 7 cm. The material is positioned on top of the chessboard pattern in Figure 3.9 with the camera at a height of 1.6 meters approximately. At this height the camera resolution is 1.1 mm, therefore the material will take up between 70 and 80 pixels of the image.

a) Specim FX10 mounted on the linear drive at the IUMA Lab.

b) Sphere Optics Zenyth Polymer.

FIGURE 3.9: Zenith Polymer material acquisition using the Specim FX10.

The polymer manufacturer provides its certified spectral reflectance in the range 250-2450 nm, with a spectral resolution of 0.1 nm, providing 22001 values. The procedure to compare the given certified spectra with the spectra obtained by the Specim FX10 camera is described next.

First, the material is sensed by the camera. A total of 100 frames have been acquired, of the material sample, the white reference sample and the dark reference sample, and then averaged. Afterwards the calibration is performed using the expression in Equation 3.1. The resulting image is displayed in Figure 3.10.

Second, the reflectance values of the certified spectra have to be calculated for the camera wavelength provided by the irradiance test. For that purpose, linear interpolation is performed.

Finally, both curves are laid on top of each other as represented in Figure 3.11 and a metric is calculated. In this case, the RMSE value has been chosen, providing a result of

FIGURE 3.10: Zenyth Polymer spectral response captured with the Specim fx10.

0.0054, indicating that both curves are very similar, specially in the range of 450 to 900 nm.

The procedure was also performed with the material placed on the side of the camera FOV, producing a similar result than the one shown, with the material placed in the center.

FIGURE 3.11: Zenyth Polymer spectral curves, measured and certified.

3.5 Band Signal to Noise Ratio

In the scientific literature there are two main procedures for calculating the device SNR. A first approach consists in calculating the signal radiation that passes through the camera entrance slit and hits the area sensor, and divide it by the noise signal, which is a combination of sensor, focal and dark current noise [126]. The issue of this approach is that several device parameters need to be known in advance to put them into the equation. Another disadvantage is that it just provides one value and as it has been seen in the previous sections, spectral response varies from the center of the sensor line to the far ends.

A second approach, based on the data captured by the sensor has been discussed in [127]. It consists in calculating the ratio between average band value and band standard deviation, providing a SNR value for each band.

In this Section an analysis of the SNR across the whole camera spectral range is made, using captured data signals and following the mentioned second approach. The intention is to retain just the bands that mainly contain information, i.e. a high SNR, trying to avoid a possible mislead in the application of algorithms in future image processing phases.

The formula to calculate the SNR is presented in Equation 3.2, where P_S represents the power of the signal, calculated as the average value, and P_N represents the power of the noise.

$$SNR = 10 \log \frac{P_S}{P_N} \tag{3.2}$$

The power of the signal noise is calculated taken into account two factors. The standard deviation of the signal itself, σ_S, which is proportional to the noise of the signal, and the mean of the signal offset, μ_{dark}, which is calculated from the dark reference sample, as represented in Equation 3.3.

$$P_N = \sqrt{\sigma_S^2 + \mu_{dark}^2} \tag{3.3}$$

For the calculation of the SNR, the captured data of the white reference panel has been used, since this offers the highest contrast to the dark reference value, as it reflects 99% of the incident light. Figure 3.12, shows the plot of the SNR calculation using a white reference taken outdoors (natural light) with 150 FPS and 4 ms exposure time.

FIGURE 3.12: Specim fx10 SNR estimation.

As it can be seen, both ends of the spectral range present a rather low SNR compared to the bands in the middle. If for instance a threshold of 10 dB is selected, which is approximately half the maximum SNR value, 10 bands from the lower end of the spectral range and 33 bands from the higher end must be removed, retaining the information contained within the range 421 - 911 (nm).

3.6 Image Georectification

Unlike 2D images, each line of the airborne pushbroom frame is collected at different time instants, which means that perspective geometry varies with every pushbroom line.

Hence, each line has six degrees of freedom from the exterior orientation parameters, leading to different line displacement, and thus turning the rectification process into a much more complicated task.

In this section the georectification process is presented, split in two main parts. First, the georreference of the hyperspectral data, which is the projection of the point from the image reference frame to the ground. Second, the geometric correction process of the image, that seeks to rectify the distortions produced in the image swath due to the fact that, as already mentioned, lines are acquired at different instants and the drone normally deviates from the ideal path and is subjected to changes in its orientation as well.

In order to better understand what is coming next, an overview of the acquisition geometry is provided in Figure 3.13. The specified values are based on the Specim FX10 camera parameters, and assuming a flight height of 50 m. The ground sampling distance, in this particular case, equal to 3.4 cm, is the pixel resolution on the ground. The same resolution is applied to both x and y directions. The focal lenght, f, is the distance between the front part of the lens and the sensor array, and it can be measured in mm or in pixels. The sensor pixel size is 16 um.

3.6.1 Direct Georreferencing

In order to fulfil the requirements set for the ENABLE-S3 project, hyperspectral acquisition, processing, detection and georeferencing the targets location, have to be accomplished. Accuracy for this specific case is not that critical and errors in the order of couple of meters are tolerated.

There are different approaches that can be used to perform georreferencing, as explained in [108]. In this work, a direct georreferencing approach is applied for the fact that the vehicle is equipped with navigation sensors, as already explained, that provide positioning and orientation of the flying platform. This method does not require ground control points, which are difficult and costly to obtain, but it is more susceptible to positioning errors between the camera and the navigation sensor.

FIGURE 3.13: Geometry of the acquisition at 50 m of altitude.

The determination of the geoposition of every single pixel in the image is accomplished by converting the image points from the image frame to the world reference frame. The world coordinates are represented in geographic coordinates, while for the IMU, located underneath the camera, the reference frame considered for the roll, pitch and yaw angles is the North-East-Down (NED) reference frame. We will also have the camera reference frame, thus we are able to obtain the relationship between the object coordinates measured in the image and the real world coordinates, as indicated in the following Equation.

$$\begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{sensor} + s\, R_{IMU} \begin{pmatrix} 0 \\ y_i \\ -f \end{pmatrix} \tag{3.4}$$

where:

- $\begin{pmatrix} 0 \\ y_i \\ -f \end{pmatrix}$ corresponds to the target pixel in the image. x is 0 due to the model of the pushbroom camera, y_i is the pixel location in the captured line and it changes following Equation 3.5.

$$y_i = -\frac{N-1}{2} + i, \qquad with\ i\ =\ 0,\ ...,\ N-1 \tag{3.5}$$

  where N is the number of pixels in each acquired line and i is the pixel under analysis.

- R_{IMU} is the Rotation matrix, measured by the IMU, in world coordinates. This matrix can be represented in terms of Euler angles, $roll(\psi)$, $pitch(\theta)$, and $yaw(\phi)$, as defined in Equation 3.6

$$R_{IMU}\ =\ R_x(\psi)\ R_y(\theta)\ R_z(\phi) \tag{3.6}$$

  As mentioned, the angles have to be given relative to the world coordinates, which means that to the IMU yaw angle, the drone heading has to be added, in order to have that angle relative to the absolute North.

- s is a scale factor that can be approximated to the ground sampling distance.

- f is the camera focal length in pixels. For the Specim FX10 camera, f equals 1487 pixels ($\frac{23.8mm}{16um}$).

As it can be seen, the method is computationally simple but very useful for certain applications. For instance, a system detecting anomalies in real time through hyperspectral image processing on-board a UAV, if an anomaly is present in the scene, with this method, its geographic coordinates can be immediately calculated and sent to, for example, an autonomous vehicle on the field.

The method has been applied to the first image swath presented in Figure 3.19 acquired in the first mission performed over the football field. Figure 3.14 shows the results,

represented as a grid of points. Each individual point corresponds to one pixel in the image, having a coordinate in the x direction (aligned to the North) and in the y direction (aligned to the East). Two individual points have been selected as an example, and their coordinates in meters are displayed. Distances in meters are calculated relative to the starting waypoint.

FIGURE 3.14: Georeferencing of swath 1 displayed in Figure 3.19.

3.6.2 Geometric correction

Some of the captured images in the performed missions present obvious distortions, especially the ones captured in the football field of the University campus, since as mentioned, wind conditions are heavier at that location.

The method presented in Section 3.6.1 might be sufficient for some of the applications where just knowing the coordinates of some selected pixels is desired. Some other applications though, require a more precise result and the overlay of the hyperspectral captured image onto a map.

For that purpose, the image is corrected based on the additional information captured by the other sensors mounted on the platform. In the next lines, the proposed geometric correction procedure is described by exclusively using the orientation and positioning data captured by GPS and IMU sensors. The same image swath used in the previous Section, the first swath from Figure 3.19, is again utilized here.

First, the drone trajectory is carefully analysed. Figure 3.15 displays, in blue, the real trajectory followed by the drone, in red, the ideal trajectory, and in green, how the real trajectory should be compensated to obtain a straight line. This last trajectory compensation line is obtained by calculating the perpendicular distance from the real to the ideal trajectory, doubling it and adding or subtracting it to the real trajectory, depending whether the latter is below or above the ideal line, respectively.

FIGURE 3.15: Mission 1 over the football field, sector 1. Platform real trajectory, ideal trajectory and trajectory compensation.

Once the compensated trajectory has been obtained the procedure is done following the next steps:

- First, Equation 3.4 is applied, using the compensated trajectory instead of the real one, different as how it was done in Section 3.6.1. The result is something similar

to what it was presented in Figure 3.14, a grid of points, that provide the x and y coordinate of each individual point.

- Second, the point x and y coordinates are converted back to image index values by dividing them by the factor s, equivalent to the ground sampling resolution. This process gives the index in both, column and row directions, for each individual pixel in the image.
- A matrix that will contain the final corrected image is created with the dimensions defined by the corners of the pixel indexes calculated in the previous step.
- Finally, interpolation is performed to obtain all pixel values to be placed into the created matrix.

Figure 3.16 presents the resulting image after having applied the aforementioned steps into the first swath captured during mission 1 over the football field. In this particular case, nearest neighbor interpolation technique has been applied as it is the simplest method and does not modify the original captured values. As it is appreciated in the presented result, the lines are straighten compared to the original image (see for comparison the image displayed in 3.19) giving a general better overlook.

Nonetheless, there are still minor misalignment, specially in the lower part, that can be better appreciated when the image is overlaid on top of a existing map of that specific area, using the Google Earth tool, as displayed in Figure 3.17. In order to generate this result, a geotiff image has been created, defining the corners in geographic coordinates and some other extra parameters such as starting position, background color, among others.

There are different options to improve the obtained result. For instance, installing a more accurate IMU and GPS sensors with higher sampling frequency, to be able to acquire data with at least the same frame rate as the hyperspectral camera. Another option is installing a frame camera, that captures a full frame image per hyperspectral line. Then through image processing techniques, images are positioned and oriented. In the literature, there are a few publications found following that procedure [40, 128].

FIGURE 3.16: Georectification of swath 1 displayed in Figure 3.19.

In this work, the latter has been the selected procedure to be further explored. As already introduced in Section 2.4.3, a full frame camera has been installed in the flying platform. Figure 3.18 presents two of the thousands of images captured during the first sector of flight mission 1 over the football field.

In order to create a reconstruction, commercial and open source software packages have been used. The commercial tool chosen was Photoscan from Agisoft [129] and the open source tool was OpenSfM [130]. Both tools implement the so-called Structure from Motion (SfM) which seeks to estimate the camera position and orientation of each captured image, together with the spatial position of some key 3D points in the scene and camera intrinsic parameters, such as focal length and distortion parameters. The estimation is supported by the stereo vision concept, based on the principle of detecting the same point or object,

FIGURE 3.17: Image overlay on top of Google Earth map.

using feature extraction techniques, in the scene from different views (camera shots). Then, using the information provided by the point descriptor, points are matched with each other in the contiguous images. Finally, applying triangulation the sought parameters can be estimated. As usually, this type of problem deals with several images, thus, in order to achieve a good estimation result that minimizes the errors, a so-called bundle adjustment technique is applied, using an equation solver.

Unfortunately, as described, the reconstruction process is complex (it involves several stages), and in this particular case two added difficulties have been observed.

- The amount of images and redundant information makes the process very slow and almost impossible to solve. Executions of the software packages with all images from a single sector have ended up with no valid solution.
- Quality of the images is very critical. For instance from a deeper analysis of the images presented in Figure 3.18, it is seen that the image in Figure 3.18a presents

much sharper features than the image shown in Figure 3.18b. This can affect the accuracy of the reconstruction as well.

a) Sharp image b) Slightly blurred image

FIGURE 3.18: IDS frame camera captured images.

Based on the aforementioned complexities new strategies have been defined for future research. First, a main reconstruction is executed taking an image every 30 or 40 frames from the hyperspectral camera. Once this is done, partial executions for the intermediate intervals are run to obtain camera position and orientation for all individual frames. This requires the modification of the software package and hence, the commercial software is automatically discarded.

Image blurring is caused by the fact that the exposure time set for the acquisition is too high. This is in its turn a consequence of the filter place in front of the camera, that reduces excessively the light hitting the sensor. Other filters with less attenuation factor shall be applied.

3.7 Conclusions

In this Chapter a set of hyperspectral images captured by the platform introduced in Chapter 2 have been presented, followed by an image pre-processing chain, detailing all the steps involved in the process and completed with results obtained in the validation process. A proper treatment of the data is crucial in its preparation for further processing phases, in order to achieve reliable results.

Sensor response is not uniform across the covered spectral range and illumination varies from one day to the other, so in order to be able to compare captured images from the same acquisition platform and across different platforms, raw captured values have to be converted to reflectance values using a white and dark reference. There is a substantial difference between raw and calibrated spectra, the second one having a physical meaning, i.e. in which range a certain material reflects the light and in which range it absorbs the light.

Radiometric calibration plays a significant role when using the acquired data across different platforms. The scientific community has invested a significant amount of efforts in generating a huge database of spectrally characterized material, as well as in sharing tagged hyperspectral datacubes. All these data can be compared with each other only if the exact wavelengths of the pixel bands are known. In this work, the provided wavelength values from the camera manufacturer have been verified with laboratory tests performed onto available calibrated material, confirming the accuracy of the wavelength values provided in the first place by the manufacturer.

It has been proven as well that not all image bands present the same SNR value. Specially the sensor extreme pixels have a much lower ratio and for some specific algorithms it might be worth detecting and eliminating those bands before applying it to the image.

Finally the proposed georectification process, split in two subprocesses, have shown a good trade off between quality of the results and algorithm complexity. Some future work have been laid out as well to further improved the obtained result using the full frame camera already available in the flying platform.

FIGURE 3.19: Captured swathes of football field mission 1.

FIGURE 3.20: Captured swathes of football field mission 2.

FIGURE 3.21: Veneguera, mission 1. Captured swathes.

FIGURE 3.22: Veneguera, mission 2. Captured swathes.

FIGURE 3.23: Veneguera, mission 3. Captured swathes.

FIGURE 3.24: Tejeda, mission 1. Captured swathes.

FIGURE 3.25: Tejeda, mission 2. Captured swathes.

Chapter 4

Hyperspectral anomaly detection algorithms

This chapter presents the algorithm developed in this Thesis for finding anomalies in hyperspectral images. This algorithm, named *Line-by-Line Anomaly Detection* (LbL-AD) algorithm, is fully described in this Chapter. The LbL-AD algorithm is specifically designed for uncovering anomalies in real time applications when images are being acquired with pushbroom sensors, that captures a spatial line of pixels at a time. Additionally, the algorithm has been designed with simple, hardware friendly operations that shall enable its implementation on-board an embedded platform that could eventually be placed on the flying platform to generate an immediate output.

4.1 Outline

Several practical hyperspectral anomaly detection (AD) applications require real-time processing for detecting targets from their background. This is especially critical in defence and surveillance domains, but also in many other scenarios in which a rapid response is mandatory to save human lives. Dealing with such a high dimensionality of data requires the conception of new algorithms to ease the demanding computing performance.

Moreover, pushbroom scanning represents the mainstream in hyperspectral imaging, introducing added complexity to the equation as there is no information of future pixels. If anomalies are wished to be uncovered in real time with such sensors, new methods have to be developed, as the new ones require the sensing of the whole image before starting the anomaly detection processing.

In this Thesis a novel technique named Line-by-line Anomaly Detection (LbL-AD) algorithm, is presented as a way of performing real-time processing with pushbroom sensors. First, the algorithm is described in detail, explaining how the anomaly information is extracted from individual lines of pixels without reprocessing the already processed image frames. Then, the parameters involved are thoroughly discussed, proposing robust values that shall perform for most of the image datasets.

Finally, the proposed algorithm is validated in terms of accuracy, based on different metrics, and time availability of the result, against captured images with the developed aerial platform, together with others from the scientific literature. Comparisons with state-of-the-art algorithms are carried out as well in order to evaluate the goodness of the LbL-AD.

The rest of the Chapter is organized as follows. A state-of-the-art review in hyperspectral anomaly detection algorithms is discussed in Section 4.2. Section 4.3 describes, step by step, the proposed LbL-AD algorithm in three subsections. First, the processing of the first n lines of the image is introduced in Section 4.3.1, followed by the calculation of the new subspace, described in Section 4.3.2, and finishing with the image projection and distance calculation in Section 4.3.3. Section 4.4 presents the methods for evaluating the

performance of the proposed algorithm, divided in three subsections. The introduction of the hyperspectral datasets to test the algorithms, in Section 4.4.1, the presentation of the state-of-the-art algorithms to be used as a comparison base, in Section 4.4.2, and the proposal of assesment metrics, described in Section 4.4.3. Section 4.5, first discusses the algorithm main parameters, in Section 4.5.1, and then outlines the main results obtained in terms of detection performance, Section 4.5.2, and execution performance, Section 4.5.3. Finally, Section 4.6 draws the most representative conclusions achieved in this work.

4.2 State-of-the-art in hyperspectral anomaly detection algorithms

Anomaly detection (AD) applied to hyperspectral data allows distinguishing rare objects with unknown spectral signatures that are not particularly abundant in a scene. This possibility of distinguishing a scarce group of pixels whose spectral signature significantly differs from their surroundings represents a crucial feature and has inspired the appearance of a huge amount of hyperspectral AD algorithms in the recent scientific literature.

The Reed-Xiaoli (RX) algorithm [41] is one of the first developments in this field, being viewed as a benchmark to which other methods are compared. The RX anomaly detector is based on the Mahalanobis distance between the pixel under test and the background class. It assumes that the background follows a single Gaussian normal distribution and the probability density function is used to classify pixels as part of the background class. Thus, the background mean and the inverse covariance matrix must be well estimated; otherwise, they could be contaminated by anomalies causing a subsequent misclassification.

Several variations of the RX detection technique have been proposed in the literature in order to improve its performance. SSRX [42] and OSPRX [43] are global anomaly detectors that apply PCA or SVD to the datacube. The goal is to reduce the data volume to a smaller subspace where the first PCA/SVD bands are supposed to represent the background class. SSRX discards these bands, and then, RX is applied to the remaining subspace. On the contrary, OSPRX projects the data onto the orthogonal subspace before

applying RX. There are on the other hand, local RX versions, that use a sliding window, to try to benefit not only from the spectral information but from the spatial information as well to improve detection performance [131]. This approach is computationally heavy though, therefore, new implementations have been developed to decrease the burden of the algorithm [132].

The aforementioned techniques assume a linear relationship between the background and the anomalous pixels. When that is not the case, Kernel RX versions [133] have appeared, to exploit the non-linearity and improve the detection results. New versions of the Kernel RX algorithm seek to simplify the operations involved for a hardware implementation and try to incorporate sliding windows to again profit from the spatial information [134].

In the recent years, new approaches to the problem have emerged, in order to cope, for instance, with inadequate Gaussian-distributed representations for nonhomogeneous backgrounds [135], the presence of noise in the images, by using a Combined Similarity Criterion Anomaly Detector (CSCAD) method [136], as well as the removal of outliers by using a Collaborative Representation Detector (CRD) [137][138]. Other authors have explored techniques extensively used in other machine vision domains as it is the use of dictionaries, to separate the anomalies from the background [139], and even combined the anomaly detection process with another very common processing stage in HSI such as compression [140].

Unfortunately, all these algorithms require the sensing of the whole hyperspectral image before starting with the process of finding the anomalies in the captured scene. However, the most widely used sensors in nowadays remote sensing applications are based on push-broom hyperspectral scanners, as it is the case of this Thesis, in which the image is captured in a line-by-line fashion, since they provide an outstanding spectral resolution and take advantage of the movement of the UAV, aircraft or satellite that carries them for capturing the whole hypercube. Hence, for applications under strict real-time constraints in which the captured images must be processed in a short period of time, it is more efficient if the anomalies are uncovered as soon as the hyperspectral data are sensed. Moreover, this kind of on-the-fly anomaly detection drastically reduces the vast amount

of memory that is required on-board the sensing platform in order to store the entire hypercubes.

In this scenario, a hyperspectral anomaly detection algorithm specially conceived for being able to process the lines of pixels captured by a pushbroom scanner as soon as they are sensed is proposed in this Thesis. More concretely, the proposed Line-by-Line Anomaly Detection (LbL-AD) algorithm is based, similar to the OSPRX algorithm, on the concept of orthogonal subspace projections but employing a low computational processing chain that guarantees a precise detection of the anomalies present in a hyperspectral scene. Recently, some new approaches have emerged in the literature addressing this issue. For instance, the Progressive Line Processing of Kernel Reed-Xiaoli Anomaly Detection (PLP-KRXD) algorithm [141], follows as well a line-by-line processing fashion. However, the complexity of the operations involved makes it unsuitable for a hardware implementation.

4.3 LbL-AD algorithm description

As it has been already mentioned, the LbL-AD algorithm is an anomaly detection algorithm suitable for real-time applications using pushbroom hyperspectral sensors whose main advantage is its fast computation combined with a good detection performance, being able to process lines of pixels as soon as they are sensed. This implies a reduction in the required memory resources since it is not necessary to store the full hypercube prior to its processing.

In order to keep a low computational complexity of the anomaly detection process, the proposed LbL-AD algorithm follows a twofold strategy: it starts processing as a whole bunch the first n lines captured and then it follows a progressive line-by-line processing for the rest of the lines of the image in which only a reduced amount of operations are performed in order to update the results for the new acquired line of pixels.

In particular, the following procedure is carried out for the first n lines captured by the sensor:

1. A reference average pixel is computed considering all the pixels contained in these lines.

2. The previously obtained average value is subtracted to each sensed pixel.

3. The covariance matrix corresponding to these pixels is calculated. At this point, it is worth to highlight that this covariance matrix is not divided by the total number of pixels, since in this way this matrix can be reutilized in the next iterations (next lines of pixels), adding to it the new covariance matrix of the acquired line. For this to work, the number of pixels so far processed needs to be accounted as well.

4. Principal component analysis is performed then onto the covariance matrix to find the d highest eigenvalues and their associated eigenvectors. Here, in order to keep the LbL-AD algorithm within a low computational burden, the following computing strategy is applied:

   a) First, the highest eigenvalue and its associated eigenvector are obtained by means of the power iteration method [142].

   b) Afterwards, deflation [143] is performed onto the covariance matrix to calculate the next $d-1$ eigenvalues and eigenvectors, by means of successively applying the power iteration method. This process is repeated until the d desired number of components is obtained.

   The reason for having selected this combination of methods is dual: on one hand, it is a very fast method for just obtaining a few principal components, and on the other hand, if the algorithm is wisely initialized, the computation time can be significantly reduced. This last characteristic is exploited by the proposed LbL-AD algorithm, as for each iteration (each new line of pixels that is acquired after the first n lines) the subspace calculated from the previously sensed pixels is used for initializing the algorithm, which brings a significant speedup factor to the process.

5. The pixels are projected onto the subspace spanned by the d eigenvectors obtained in the previous step.

6. The Mahalanobis distance is calculated for each pixel, as it is done in the original RX algorithm. This step involves the calculation of the pseudoinverse of the covariance matrix. However, as far as the covariance matrix in the new projected subspace is a diagonal matrix, its inverse is obtained by just inverting its single elements individually, which is again a huge save in terms of computing time.

7. Based on the calculated distance result, it is decided whether each pixel is an anomaly or not. Thanks to the good separability between background and anomalies of the result provided by the algorithm, this task is carried out applying a simple technique of outlier detection.

8. The background pixels Mahalanobis distance mean value and standard deviation are computed for later use in the process.

Once these n lines of pixels have been processed the following line-by-line procedure is carried out for the rest of the pixels in the hyperspectral scene under analysis:

A. The average pixel obtained previously in step 1 is subtracted to each pixel of the line under processing.

B. The covariance matrix of the line under processing is calculated and added to the existing covariance matrix calculated previously in step 3. As with step A, this is performed under the assumption that the average pixel obtained in step 1 remains approximately constant, which allows to skip the full computation of a new covariance matrix for each new line of pixels. Moreover, this method allows us to keep in memory just the previous covariance matrix, but not the entire amount of pixels processed, which means a huge save in memory space and number of memory transactions.

C. Steps 4 to 7 are applied to the line of pixels under processing.

D. If in step 7, anomalies were detected, then the recently calculated covariance matrix is discarded, and the previous covariance matrix available recovered for processing the next line. The reason for doing this is to keep as much as possible the covariance

matrix modeling only the background, so whenever an anomaly is coming into the scene it will be something new. This confides the result an homogeneity that otherwise it would not have.

E. Finally, previous step 8 is applied, exclusively to the background pixels.

Figure 4.1 shows a general vision of the stages involved in the described algorithm. The following subsections will detail the different parts of the proposed algorithm.

FIGURE 4.1: General overview of the LbL-AD algorithm.

4.3.1 Computation of first n lines

The first n lines are buffered into a matrix, M, which afterwards is used to calculate the average pixel, μ, to be substracted to every processed image line. The covariance matrix, Σ, results from the obtained normalized matrix M', for which the eigenvectors, $\vec{u_1}, ..., \vec{u_d}$, and the eigenvalues, $s_1, ..., s_d$, are calculated. After that, the new subspace is calculated as explained in the next subsection. Algorithm 4.1 shows the pseudocode of the algorithm initialization, where r stands for each captured line of n_c pixels.

Algorithm 4.1: Computation of first n lines

Inputs: $\mathbf{M} = [\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_n]$ where $\mathbf{r}_i = [\mathbf{s}_{i1}, \mathbf{s}_{i2}, ..., \mathbf{s}_{inc}]$; {Initial portion of the HSI}
$\mathbf{d}$, {Number of principal components to be selected}
$\mathbf{N} = \mathbf{n} \times \mathbf{n}_c$, {Number pixels contained in the first n lines}

```
P.1 μ = mean(M); {Average pixel}
P.2 for i = 1 to n do
P.3     for j = 1 to nc do
P.4         s_ij = s_ij − μ;
P.5     end
P.6 end
P.7 Σ = M^T M;
P.8 M' = Σ/N;
P.9 [Psi, S] = PCA(M', d);
```

Outputs:
$\mathbf{\Sigma}$, {Covariance matrix}
$\mathbf{S}$, {Eigenvalues}
$\mathbf{PSI}$, {Eigenvectors}
$\mathbf{n}$, {Number of processed pixels}
μ, {Average pixel}

4.3.2 Subspace calculation. Power Iteration method and deflation.

On the first run the subspace is calculated for the computed covariance matrix, Σ, from the buffered n lines and then it is updated on each iteration after the covariance matrix has been updated. Since in the first iteration there are no previous eigenvectors for feeding the algorithm, the input eigenvectors, PSI_{in}, are initialized with random values. However,

on the subsequent iterations the set of eigenvectors defining the previous subspace are introduced as inputs of the algorithm and used to initialize the calculation, considering that there shall not be a big difference between the set of eigenvectors computed from two consecutive iterations. The other two inputs to the algorithm are the number of eigenvalues and eigenvectors to be calculated, d, and a stop condition for the convergence of the power iteration method, tol. An auxiliary matrix, B, is used to deflate the original covariance matrix and keep calculating subsequent eigenvalues and eigenvectors. The deflation process mainly consists in obtaining a new matrix that provides the same result than the original one in the orthogonal direction to it and it is zero otherwise. A detailed description of the subspace calculation process is presented in Algorithm 4.2.

Algorithm 4.2: Subspace calculation

Inputs: $\mathbf{\Sigma}$, {Covariance matrix}
$\mathbf{PSI}_{in}$, {Input Eigenvectors }
d, {number of eigenvalues to be calculated}
tol, {stop condition for the power iteration method}

P.1 $[PSI_1, S_1] = power\ method(\Sigma, tol, PSI_in_1)$;
P.2 $\mathbf{i = 2}$;
P.3 $\mathbf{B} = \mathbf{\Sigma}$; {Support matrix to compute deflation}
P.4 **while** $i <= d$ **do**
P.5 $\quad \mathbf{B} = \mathbf{B} - \mathbf{PSI}_{i-1}^T \mathbf{PSI}_{i-1}$;
P.6 $\quad [PSI_i, S_i] = powermethod(\Sigma, tol, PSI_in_i)$;
P.7 $\quad \mathbf{i = i + 1}$;
P.8 **end**

Outputs:
S, {Eigenvalues}
PSI, {Eigenvectors }

The subspace algorithm makes a recurrent call to the *Power Method* function, whose behavior is described in Algorithm 4.3, until all d eigenvalues and eigenvectors have been obtained. As explained earlier, the initialization with the previous obtained eigenvector is used to speed up convergence, which combined with the use of the Rayleigh coefficient, $rayl$, has proven to be very efficient [144]. This coefficient is calculated using the eigenvector approximation and represents in its turn a very good approximation of the eigenvalue. The method consists in iteratively calculating the new coefficient, $rayl$, with the recently obtained vector and comparing it with the value calculated in the previous iteration, $rayl_0$, until the difference between them is lower than the input tolerance, tol.

Algorithm 4.3: Power Method

Inputs: $\mathbf{\Sigma}$, {Covariance matrix}
$\mathbf{b}_k$, {Input Eigenvector}
tol, {stop condition}

P.1 $\mathbf{rayl}_0 = \mathbf{0}$;
P.2 $\mathbf{rayl} = \frac{\mathbf{b}_k^T \mathbf{\Sigma} \mathbf{b}_k}{\mathbf{b}_k^T \mathbf{b}_k}$;
P.3 **while** $abs(rayl_0 - rayl) > tol$ **do**
P.4 $\quad \mathbf{rayl}_0 = \mathbf{rayl}$;
P.5 $\quad \mathbf{b}_{k+1} = \frac{\mathbf{\Sigma} \mathbf{b}_k}{\|\mathbf{\Sigma} \mathbf{b}_k\|}$;
P.6 $\quad \mathbf{rayl} = \frac{\mathbf{b}_k^T \mathbf{\Sigma} \mathbf{b}_k}{\mathbf{b}_k^T \mathbf{b}_k}$;
P.7 **end**

Outputs:
rayl, {Calculated Eigenvalue}
$\mathbf{b}_k$, {Calculated Eigenvector}

4.3.3 Image projection and Mahalanobis distance calculation

After the first n lines have been processed, the proposed algorithm processes the sensed data line by line, as it is described in Algorithm 4.4. As it has been already mentioned, the average pixel, μ, obtained in the first stage of the algorithm, is substracted from each pixel, and then, the covariance matrix is updated to obtain Σ_{new} and the new set of d eigenvalues and eigenvectors from the updated Σ_{new}. The process then continues projecting the image onto the new subspace and obtaining the Mahalanobis distance for each pixel. Finally, the mean value, μ_{dist}, and the standard deviation, σ_{dist}, of the Mahalanobis distance of just the background pixels are calculated and updated on every iteration, in order to account for possible outliers, and thus, anomalies. In case anomalies are found, the covariance matrix of the previous iteration is used, instead of the updated one as long as it is not the first iteration. As it can be seen in the pseudocode, a very large value, for instance 15 times the standard deviation is used to be very conservative and avoid covariance update in case anomalies are found. The reason for updating the covariance matrix selectively is to obtain a more homogeneous result in terms of the anomalies.

Algorithm 4.4: Image Projection and RX

Inputs: $\mathbf{r}_k = [\mathbf{s}_{k1}, \mathbf{s}_{k2}, ..., \mathbf{s}_{knc}]$ where $k > n$; {captured frame of nc pixels}
$\mathbf{\Sigma}_in$, {Covariance matrix of the already processed image}
μ, {Average pixel}
$\mu_{dist_in}, \sigma_{dist_in}$, {Background distance mean and standard deviation}
$\mathbf{B}_{mask_in} = [\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_{k-1}]$, {Binary result of processed pixels so far}

```
P.1  L = k × n_c, {Number of pixels processed so far, line k included }
P.2  for j = 1 to nc do
P.3      s_kj = s_kj − μ;
P.4  end
P.5  Σ = M^T M + Σ_in;
P.6  [PSI, S] = Algorithm_2(Σ/L, PSI, d, tol);
P.7  Y = PSI^T × M; {Projected Image to the new calculated subspace}
P.8  P = RX(s) = s^T S^{-1} s
P.9  update = 1
P.10 for j = 1 to nc do
P.11     if p_j > μ_dist_in + σ_dist_in * 15 then
P.12         b_{n+j} = 1;
P.13         update = 0;
P.14     else
P.15         b_{n+j} = 0;
P.16     end
P.17 end
P.18 μ_dist = (Σ_{i=1}^{nc} p_i * b_i) / L
P.19 σ_dist = sqrt(Σ_{i=1}^{nc} (p_i − μ_dist)^2 * b_i) / (L−1);
```

Outputs:
$\mathbf{P}$ {Vector of Mahalanobis distances}
$\mathbf{B}_{mask}$ {Binary result of processed pixels}
PSI, {Eigenvectors}
$\mu_{dist}, \sigma_{dist}$, {Bck distance mean and std deviation}

```
if update then
    Σ, {Covariance matrix};
else
    Σ_in, {Input covariance matrix};
end
```

4.4 Data and evaluation metrics

The proposed algorithm shall be validated against hyperspectral datasets, using a set of assessment metrics and comparing the result with other proposals from the scientific literature. In this section the aforementioned topics are covered in detail.

Regarding the hyperspectral images to test the algorithms against, synthetic and real images from the state-of-the-art are used, both sharing a key characteristic, the ground-truth are given, which enables the performance of a deeper numerical evaluation of the implementations and a better comparison with the selected algorithms from the literature.

4.4.1 Reference Hyperspectral Data

For the evaluation of the performance and effectiveness of the proposed method, both simulated and real HSIs have been used. The simulated data have a size of 150x150 pixels and 429 spectral bands. It was generated using a spectral library collected from the United States Geological Survey (USGS) [145]. Background was simulated using four different spectral signatures whose abundances were generated using a Gaussian spherical distribution [146]. Twenty panels of different sizes arranged in a 5x4 matrix were introduced as anomalies. There are five 4x4 pure-pixel panels lined up in five rows in the first column, five 2x2 mixed-pixel panels in the second column, five subpixel panels combined with the background in a proportion of 50% in the third column and five subpixel panels blended with the background at 75%. Therefore, the simulated image has 110 anomaly pixels, a 0.49% of the image, as it is illustrated in Figure 4.2a. This dataset is very challenging for AD because of the high spectral similarities between some anomalous and background signatures.

In order to test the proposed algorithm in a more realistic scenario, three real hyperspectral datasets have been also used, three well known datasets available in the literature. The first real dataset was taken over the Rochester Institute of Technology (RIT) by the Wildfire Airborne Sensor Program (WASP) Imaging System [147]. The system covers the visible, short, mid and long-wave infrared regions of the spectrum. The sensor was comprised by a high-resolution colour camera covering the visible spectrum, a short infrared imager covering 0.9 - 1.5 μm, mid infrared imager covering 3.0 - 5.0 μm and a long infrared imager covering 8.0 - 9.2 μm. A portion of the overall image taken over a parking lot with a size of 180 x 180 pixels and 120 bands has been used in this study, as can be seen in Figure 4.2b where anomalies are fabric targets which consist of 72 pixels and account

for 0.22% of the image. The second real dataset was collected by the NASA Jet Propulsion Laboratory's Airborne Visible Infra-Red Imaging Spectrometer (AVIRIS) over the World Trade Centre (WTC)) area in New York City on September 16, 2001 [148]. The original dataset has a size of 614 x 512 pixels and 224 spectral bands from 0.4 to 2.5 μm although a smaller region with size of 200 x 200 pixels was selected as dataset. Anomalies are thermal hot spots which consist of 83 pixels and account for 0.21% of the image scene. Figure 4.2c shows a representation of this image. The third real dataset is a portion of 67×80 pixels cropped from the well−known Washington DC Mall hyperspectral dataset, obtained through Purdue University's MultiSpec freeware project. This dataset was acquired by the airborne mounted Hyperspectral Digital Imagery Collection Experiment (HYDICE) which measures pixel response in the 0.4 to 2.4 μm region of the visible and infrared spectrum. Bands in the 0.9 and 1.4 μm region where the atmosphere is opaque have been omitted from the dataset, leaving 191 bands. Figure 4.2d shows a RGB representation of the area chosen for this analysis, which mainly consists of trees and a river and where the anomalous pixels belong to a building placed among the trees. The number of anomalous pixels is 15, 0.28 % of the image scene.

4.4.2 Reference Algorithms

The LbL-AD algorithm has been compared against the most relevant algorithms of the state-of-the-art in the field of hyper-spectral AD, both computing the image as a whole and in a line-by-line fashion as the LbL-AD algorithm does. The intention is to test the quality of the detection results and the separability between anomalies and background, as well as the complexity and required computational effort. The selected algorithms are the OSPRX, which computes the anomalies once the entire image is captured, and the PLP-KRXD algorithm [141], which follows a line-by-line fashion. The PLP-KRXD is an adaption for real time processing purposes of the Kernel-RX detector, using parallel causal sliding windows.

FIGURE 4.2: Test datasets. (a) Synthetic image. (b) WASP RIT scene. (c) AVIRIS WTC scene. (d) HYDICE WDC scene.

4.4.3 Assessment Metrics

Receiver Operating Characteristics (ROC) curves and the Area Under Curve (AUC) have been widely used in the literature to evaluate different anomaly detection algorithms. ROC curves are two dimensional graphical plots which illustrate the relationship between True Positive Rates (TPR) and False Positive Rates (FPR) obtained for various threshold settings. To compare the performance of several AD algorithms, AUC is used as a scalar measure, so that, a representation with the biggest AUC outperforms the others. However, this metric does not always reflect how well the algorithm separates the anomalies from the background. For this reason, two extra quality metrics will be employed in this work: Brier Score (BS) and Squared Error Ratio (SER)[149].

BS measures the accuracy of probability predictions in terms of marking anomalies and background pixels with the highest and the lowest scores, respectively. If anomaly pixels are represented as *ones* and background pixels as *zeros* in the ground-truth, then, the BS for each type of pixel is calculated as:

$$\mathrm{BS}_{\mathrm{anomaly}} = (\mathbf{p}_i - 1)^2; \tag{4.1}$$

$$\mathrm{BS}_{\mathrm{background}} = (\mathbf{p}_i - 0)^2; \tag{4.2}$$

being $\mathbf{p}_i$, the output of the algorithm, spanned in the range $[0 - 1]$.

Adding the $\mathrm{BS}_{\mathrm{anomaly}}$ of all individual pixels in the image provides an insight into the error made scoring anomalous pixels, the same way that adding $\mathrm{BS}_{\mathrm{background}}$ of all individual pixels in the image gives an insight into the background misclassification error. These error scores are represented by the Anomaly_Error and Bck_Error metrics defined below.

$$\mathrm{Anomaly_Error} = \sum_{i=1}^{n_{\mathrm{anomaly(GT)}}} (\mathbf{p}_i - 1)^2 \tag{4.3}$$

$$\mathrm{Bck_Error} = \sum_{j=1}^{n_{\mathrm{bck(GT)}}} (\mathbf{p}_j - 0)^2 \tag{4.4}$$

Finally, a global scalar metric combining the Anomaly_Error and the Bck_Error, named SER, is obtained dividing the sum of all squared differences by the total number of pixels in the image.

$$\mathrm{SER} = \frac{\sum_{i=1}^{n_{\mathrm{anomaly(GT)}}} (\mathbf{p}_i - 1)^2 + \sum_{j=1}^{n_{\mathrm{bck(GT)}}} (\mathbf{p}_j - 0)^2}{N_p} \cdot 100 \tag{4.5}$$

with the first addend in the summation being the Anomaly_Error and the second addend the Bck_Error.

4.5 Experimental Results

In this Section, experimental results are reported. First the algorithm input parameters are discussed, arguing the selection of the chosen values as being the most robust ones for every dataset. Then the detection performance results are displayed using the selected datasets from the state-of-the-art.

Finally, the execution performance is evaluated, comparing the time availability of the result for an algorithm that performs the processing to the entire image, and the proposed LbL-AD that does the computation line-by-line.

4.5.1 Analysis of input parameters

There are a few input parameters to the algorithm such as n, d and tol, which were selected based on the experience acquired during the development of the proposed algorithm. These parameters have been adjusted after a deep analysis of the algorithm behaviour in order to ensure a good performance in all analysed scenarios. To verify that they are adequate settings, experiments have been made changing the values of each parameter in a certain range while keeping the others constant in order to evaluate the performance of the algorithm.

As it was previously described, n is the number of captured lines that have to be accumulated in memory to calculate its average value and use it as a reference throughout the whole process. These lines are then, after captured, processed as a whole bunch and the result is provided to the user. From then on, lines are processed as they are being captured. Therefore, this parameter has been set trying to achieve a good trade-off between having a representative average value, but also small enough so the user does not have to wait a significant amount of time in order to get the detection results.

For the purpose of the analysis of this parameter, the real HSI datasets presented in Section 4.4.1 have been evaluated. The test consists in progressively increasing the parameter from 1 line up to 50 lines, and calculating the average pixel vector. In Figure 4.3, the analysis

is shown for the AVIRIS image captured over the WTC. From this figure, it is noticed that the average shows a very regular pattern, which starts stabilizing after buffering 20 image lines, and from then on it just slightly smooths some of the peaks. Proceeding in the same way with the other three datasets, it gives that for all of them, the selected value of 20 lines is a good trade-off between enough representativeness in the curve together with a fast response to the user.

FIGURE 4.3: Image average pixel. Analysis based on the used lines in the buffer

The number of representative principal components d is a rather critical parameter that has to be carefully selected. In this case simulations show an absolute maximum value of 5 as a good choice in terms of information being retained and computation time. In addition, it has been imposed that the decay between two consecutive eigenvalues is not higher than 10^{-6}, otherwise the number of elements until that decay are taken. The main reason for such restriction is that if components with a very low value are taken the algorithm is not robust enough.

Finally for the stop condition in Algorithm 4.3, again a trade-off between computation time and goodness of the results has been taken into consideration. Making the tolerance for the power iteration algorithm too small has the disadvantage of having to call the function several extra times to refine the result, which does not produce an improvement in terms of the algorithm result. For this reason, a value of 10^{-8} has been selected.

4.5.2 Detection Performance

4.5.2.1 Validation against images from the state-of-the-art

In order to test the performance of the proposed algorithm, its detection efficiency has been compared with the selected anomaly detection algorithms from the scientific literature, presented in Section 4.4.2 and against the state-of-the-art images whose ground truth are available.

Due to the fact that the performance of the PLP-KRXD algorithm depends on a few input factors such as window length l, window width w and kernel order d, the values providing the best results for each image scene has been found out and detailed in Table 4.1.

	l	w	d
Synthetic Image	15	5	1
WASP RIT	15	5	5
AVIRIS WTC	15	7	1
HYDICE WDC	15	7	2

Table 4.1: PLP-KRXD parameter table.

The results obtained in terms of the metrics described in Section 4.4.3 are summarized in Table 4.4.3 for each of the already mentioned images. From these results, it is concluded that the proposed LbL-AD algorithm is able to provide very similar results with respect to the OSPRX algorithm and even outperforms it in some cases, as it is in the case of the HYDICE WDC image, with the major advantage of being able to compute the anomalies present in the image as soon as the line of pixels is sensed. In addition, it is also concluded from Table 4.2 that if we focus the attention only on the solution that allows to process

the hyperspectral images in a line-by-line fashion, the proposed LbL-AD delivers a much better detection performance than the one provided by the PLP-KRXD.

	AUC	Anomaly_Error	Bck_Error	SER
	Synthetic Image			
OSPRX	1.00	48.70	0.15	0.21
LbL-AD	1.00	77.84	0.01	0.34
PLP-KRXD	0.95	92.80	0.63	0.41
	WASP RIT			
OSPRX	0.99	40.93	1.15	0.12
LbL-AD	0.99	43.98	2.05	0.14
PLP-KRXD	0.77	71.80	10.32	0.25
	AVIRIS WTC			
OSPRX	0.97	72.93	0.76	0.18
LbL-AD	0.96	74.63	1.32	0.19
PLP-KRXD	0.51	82.97	5.87	0.22
	HYDICE WDC			
OSPRX	0.99	7.42	0.27	0.14
LbL-AD	0.99	5.69	0.48	0.11
PLP-KRXD	0.97	8.91	0.21	0.17

Table 4.2: Implementation results against the datasets described in Section 4.4.1 and using the assessment metrics proposed in Section 4.4.3.

Finally, Figure 4.4 illustrates the two-dimensional plots of the detection results given by the proposed LbL-AD algorithm and the OSPRX algorithm for all the considered datasets together with the ground truth. It is visually demonstrated that our proposal is able to capture in a very precise way all the anomalies present in a given hyperspectral image independently of the characteristics of the background and of the anomalies present in the scene under analysis, and also showing that the visual result provided by the LbL-AD algorithm is very similar to the one provided by the OSPRX as it was already demonstrated with the results displayed in Table 4.2.

FIGURE 4.4: Two-dimensional plots of the detection results obtained by the AD algorithms considered in this work (rows) for all datasets (columns). To make the results easier to understand, the ground-truths of the anomalous maps are shown in the last row.

Our group has further pursued this line of research, developing new hardware friendly implementations of anomaly detection algorithms to process the acquired lines as they are being sensed from pushbroom cameras. Recently, another algorithm has been released, named *Line-by-Line Fast Anomaly Detector for Hyperspectral Imagery* (LbL-FAD) [150], working as well under the assumption that anomalous pixels cannot be

well represented by the background distribution, but proposing an alternative method to compute the orthogonal subspace spanned by the selected background samples, making the anomalous pixels better detectable. Additionally, the LbL-FAD incorporates an automatic thresholding method which provides line-by-line and real-time binary maps where anomalous targets are segmented from the background. For that reason a numeric comparison with the LbL-AD can not be achieved, as this implementation does not threshold the outcome. However, the same images from the state-of-the-art have been used to test the LbL-FAD and the output results are presented in Figure 4.5.

FIGURE 4.5: The three-dimensional (3D) plots of the detection results obtained by the LbL-FAD and their corresponding binary maps thresholding by τ for the five datasets: (a) Synthetic image, (b) WASP RIT scene, (c) AVIRIS WTC scene, (d) WDC scene.

In Figure 4.5, 3D plots and thresholding results of the anomaly detection algorithm outputs are displayed. The selected threshold is applied based on a parameter τ that is calculated within the algorithm and specific for each dataset. As it can be seen, the LbL-FAD performs equally good as the LbL-AD against the selected set of images from the state-of-the-art.

4.5.3 Execution Performance

In Section 4.5.2, the detection performance comparison of the LbL-AD algorithm versus the state-of-the-art has revealed that the results obtained by LbL-AD are much better than the ones provided by the line-by-line PLP-KRXD algorithm and equivalent to those provided by the OSPRX algorithm. The main advantage of the proposed LbL-AD algorithm with respect to the OSPRX algorithm is its capability of processing the lines of pixels as they are being sensed, while the OSPRX needs to wait until the complete image has been captured to produce a result. In order to emphasize this aspect, a comparison between the time required for processing an image captured by the developed UAV platform, and have the result ready for both algorithms is shown in Figure 4.6. The selected image is a portion of the image swath 3 captured during mission 2 performed over Veneguera fields, displayed in Figure 3.22. The selected portion has a size of 825×1024 pixels.

First of all, the acquisition has to be taken into account and has been displayed in Figure 4.6 in green color. The acquisition frame rate for that mission was set to 125 FPS, which gives a frame every 8 ms. Then, after the first n lines have been buffered, the LbL-AD implementation starts its processing. As argued in Section 4.5.1, the parameter n has been set to 20, which corresponds approximately to a 2.5% of the total image number of lines, 825.

As it can be appreciated, the LbL-AD software implementation takes more than the acquisition time to process each frame, and that is why the LbL-AD curve, represented as a dot line in blue color, deviates from the acquisition line. In this case, the frame processing time has been estimated to 17 ms. It must be highlighted, that this is a

FIGURE 4.6: Execution time comparison in terms of availability of the result.

software implementation where parallelization is not at all exploited, matrix products are well optimized, hence, executions involving the whole image are implemented much faster than in a line-by-line fashion. This shall not be the case in a hardware implementation, where much better time performances are expected.

Finally, after the acquisition is finished, the OSPRX execution starts, taking a total of approximately 6.1 seconds, added to the 6.6 seconds of total acquisition time, gives a result after 12.7 seconds. No matter where the anomaly is located, information will be available after that amount of time. If the image is processed with the proposed LbL-AD algorithm, depending on where the anomaly is present at the image, a result will be available right after its detection. For instance if the anomaly is located at a 40% of the image, that is in this case at line 330, the result is available after approximately 5 seconds, which is more than 7 seconds earlier than with the other method.

The algorithms were implemented in Matlab 2019 and executed on a PC with an Intel Core i7-8550U CPU at 1.8 GHz and 12 GBytes of RAM memory running a Windows 10 operating system.

4.6 Conclusions

In this Chapter a novel algorithm, named Line by line Anomaly Detection (LbL-AD) algorithm, has been described. Its conception has been intended for hyperspectral anomaly detection applications to be performed in real time capturing images with pushbroom hyperspectral sensors, which are extensively utilized nowadays.

The algorithm is premised on the concept of orthogonal subspace projection, similar to other ones encountered in the scientific literature, but as opposed to these other algorithms, the LbL-AD, after an initial stage involving the first n scanned lines, it executes the processing chain in a line-by-line fashion.

A detailed behavior of the algorithm has been described, with all steps involved, starting from the buffering and processing of the first n lines, continuing with the new subspace calculation, which constitutes another novelty of the presented work, as singular value decomposition is only executed for the first few components which contains most of the information, making the implementation much simpler involving hardware friendly operations. The process finishes with the projection of the image onto the new subspace and the calculation of the Mahalanobis distance.

Another advantage of the proposed method is the fact that the already processed lines do not need to be stored in memory, meaning a huge save in terms of disk space resources as well as an optimization in the processing time. Memory accesses are one of the major bottle-necks in computation, specially in cases like this, dealing with considerable amount of data.

All the mentioned algorithm features make it a very good candidate to be implemented in the on-board embedded platform available in the drone, the Jetson TK1. The implementation process shall be similar to the one defined for the HyperLCA compression algorithm in [151], adapted for the images captured on-board the flying platform and to be executed in parallel.

Experimental results have been obtained using quality metrics for different synthetic and real images both from the state-of-the-art as well as the ones captured with the developed

hyperspectral flying platform, showing an excellent performance compared to relevant proposals such as OSPRX and PLP-KRXD. From these results it is concluded that LbL-AD is an attractive candidate for hyperspectral imaginary, opening a new frontier in myriad of applications in which real-time and low computational cost is mandatory.

Chapter 5

Hyperspectral simulation environment

Testing and validation of the algorithms and methodologies presented so far is a very important step in the entire development process. Individual components have been successfully validated, but in order to test the system as a whole, with all integrated components, a step further has to be taken. Traditionally, this would be performed right in the field with the inconveniences and extra costs that this approach would incur in.

In this Chapter a novel approach is presented, through the development of a hyperspectral simulation environment in which the main elements have been virtually modelled and can be simulated together, permitting the execution of a set of tests, that enables researchers and developers fine tune the implemented software, prior to upload it to the real platform for its testing on the field.

5.1 Outline

Chapter 2 has described the construction of a UAV platform based on a hyperspectral sensor. As it has been mentioned, they offer very attractive characteristics compared to conventional remote sensing systems, such as a flexible revisit time, enhanced spatial resolution and their capability of being controlled autonomously. The combination of these characteristics enables the possibility of performing missions that eventually provide results in real time, what makes them suitable for specific monitoring and surveillance applications that demand a prompt reaction to an stimulus.

However, the use of such autonomous systems equipped with hyperspectral sensors poses a new challenge for researchers and developers, not only adapting the algorithm implementations to be able to work in real-time as images are being captured, but also making the appropriate modifications to let it coexist with the autonomous system control. Presently, the algorithm development process starts with a design phase consisting in analysing the target application needs and defining a few possible solutions, followed by the implementation phase where the solution is coded into a high level language to rapidly have results available and start looking for the most suitable solution. Usually this latter phase is carried out offline, with synthetic or real hyperspectral images downloaded from public repositories.

The aforementioned approach, followed in Chapter 4, provided offline results that are valid to, for instance, assess the goodness of the detection performance, compared to other state-of-the-art proposals. The main problem of this approach though, comes when the image processing algorithms are transferred to the target device and the real system is brought to the field to run test campaigns. Then, it is usually realized that several aspects were overlooked both in the design and implementation phase carried out offline, and hence, the system does not perform as expected. It is then when a back and forth process starts in order to try to accomplish the desired results with the algorithm, incurring in delays and extra costs that were not initially foreseen.

The contribution of the work presented in this Chapter, consists in a novel intermediate phase for testing and validating the behaviour of hyperspectral image processing

algorithms for real-time systems, based on a simulation environment that provides a faithful virtual representation of the real world. Figure 5.1 shows all the involved stages, including the new one proposed in this work, highlighted in red, which significantly shortens the gap between the offline image processing algorithm implementation and its use in a real scenario.

FIGURE 5.1: Proposed validation and verification strategy.

The virtual environment developed in this Thesis offers the possibility of simulating and stimulating the sensor, virtually creating hyperspectral images as they are captured by the targeted hyperspectral sensor and applying the processing algorithms to them in order to obtain the results straight away. In this way, before performing a real testing with the whole system, simulations can be run several times making modifications in the code and/or in the input parameters. This shortens the testing phase and reduces the required overall manpower required, what in the end translates into considerable cost savings in the whole system implementation.

The created environment is embedded in the preexisting Agriculture Graphical Simulation Environment (AgSim) which is able to virtually recreate the different scenarios considered within the European H2020 project, ENABLE-S3 farming use case, introducing different machinery such as a harvester, a tractor, and a drone onto a farming field. The one

that focuses on the drone and its sensor (the on-board hyperspectral camera), is being considered here as the device under test.

The rest of the Chapter is organized as follows. First, a state-of-the-art review in simulation environments is discussed in Section 5.2. Section 5.3 introduces the AgSim environment as an actor that allows to virtually recreate a realistic farming scenario in which we can test the benefits of the proposed simulation environment for the validation and verification of real-time hyperspectral image processing algorithms on-board a UAV. Section 5.4 presents the proposed hyperspectral simulation environment and its main elements. Section 5.5 introduces the creation of two types of hyperspectral scenarios: target detection scenarios, presented in Section 5.5.1, and map scenarios based on real bitmaps, presented in Section 5.5.2. Section 5.6 displays the results using the scenarios presented earlier. In Section 5.6.1, target detection scenarios are used to test an anomaly detection algorithm, while in Section 5.6.2, a virtual mission flight results are laid out. Finally, Section 5.7 discloses the obtained conclusions and outlines further research lines.

5.2 State-of-the-art in simulation environments

The role of simulation tools and environments has gained momentum in the last few decades, as system complexity is increasing, involving the integration of a great number of individual elements. These virtual environments let engineers perform a battery of tests, that can be either manually or automatically executed, to on the one hand fine tune the software that is going to run on the components and on the other hand validate the whole system performance. Additionally, those environments can be used to train qualified workers manipulate sophisticated equipment, such as heavy machinery, or learn to perform critical tasks, such as a surgical operations.

Although usually linked to the industry, for product development purposes, the use of simulation environments has started to attract the attention of the scientific community as well. For instance, in [152], a traction control system is tested through the modelling and simulation of a railway vehicle. [153] presents how simulators and simulation environments

are a game changer in surgical education, as learners have the ability to improve their skills without endangering any life. In the farming domain, with the appearance of autonomous vehicles, the use of virtual models and tools have become essential for its development [154].

Involving multi-copter aerial vehicles, several are the developments achieved. A network simulation environment to identify performance and security issues is introduced in [48]. How the operational gaps are covered with the modelling and simulation of aerial vehicles is detailed in [47]. A guidance in generating the environment requirements is proposed in [155]. The differences of system verification Software in the loop (SiL) and Hardware in the Loop (HiL) are clearly stated in [49].

The aforementioned publications set their focus on the vehicle and its control, but do not take into account the sensors that can be embarked on such aerial platforms. In this sense, [156] presents a UAV gas sensor system simulation and analysis for target detection and positioning. In [157], a Hummingbird multirotor with a Microsoft Kinetic RGBD camera has been simulated using the commercial platform Gazebo Software, for 3D modelling and navigation purposes.

In this Thesis, a hyperspectral camera on-board a UAV has been incorporated to a simulation environment. This is the first time, to the author's knowledge, that such a sensor is simulated in a virtual tool. The development process follows in the next sections.

5.3 Agriculture Graphical Simulation Environment: AgSim

The AgSim has been utilized in this work in order to recreate a virtual farming domain in which the proposed simulation environment is validated. The simulated scenario consists of a farming field together with a harvester, a tractor with trailer, and the already mentioned drone. AgSim is comprised of two main parts: the simulation engine and the 3D visual system. Machine models, i.e. harvester, tractor and drone models, are

built on top of the simulation engine which interfaces to external control systems, either simulated, in case the testing setup is a SiL, or a real controller, in case the testing setup is a HiL. A script based test command interface is used to load test cases and to set proper initial state for the machine simulation variables. Based on the control system behaviour and test script control of the real time simulation, results are produced for the test case evaluation.

Figure 5.2 shows the main equipment simulated in the AgSim environment. The drone with the hyperspectral camera, its axis and the vision scope can be seen in Figure 5.2a. In Figure 5.2b, the combine-harvester is presented. Figure 5.2c displays a parallel driving representation involving all the autonomous vehicles available in the AgSim environment, namely, the combine-harvester, the tractor and the drone.

AgSim has been developed by the company Creanex [158], a partner in the Farming Use Case of the ENABLE-S3 Project. It is based on the Creanex Simulation Platform (CSP), which is a product to create custom simulation solutions to support development and testing of embedded distributed control systems.

The 3D visual system helps the final user to see the simulated events, allowing to check that the simulation runs as expected and fulfils the machine operation requirements under test. The initial state of a testing scenario may contain a description of different situations and events. These are, for instance, stationary or moving objects, obstacles not known by the control systems, or animals running on the field, just to name some.

The simulated models of the machines have interfaces for external control algorithms running in Python. The simulation environment supports the hardware in the loop testing setup through I/O boards or adapters for serial communication such as Controller Area Network (CAN) or Ethernet connected to real embedded control modules. As an example, also within the project ENABLE-S3, the partner TTControl has integrated one of their Electronic Control Units (ECUs), the model HY-TTC 580, into the AgSim using CAN signals.

a) Drone representation

b) Combine-harvester representation.

c) Drone, combine-harvester and tractor representation, together with a simulated farming environment.

FIGURE 5.2: Farming simulation environment and machinery representation.

5.4 Proposed Simulation Environment

The main purpose of creating the proposed hyperspectral simulation environment is to be able to validate and verify real-time image processing algorithms in combination with flight control functions within realistic scenarios. Figure 5.3 displays the diagram of the implemented system in terms of functional modules. As it is seen, it has been designed and programmed in a modular way, easing the replacement of modules if another hyperspectral sensor and/or image processing algorithm is to be tested. Another important aspect that supports the mentioned modularity is the parameterization of the whole environment, making use of configuration files that are loaded when the system is started.

FIGURE 5.3: Hyperspectral simulation environment block diagram.

The whole system is comprised of four main modules: the User Interface module, the Hyperspectral Camera module, the Drone Control module, and finally, the AgSim environment already introduced. It is important to highlight that these four modules run in separate threads completely decoupled one from another.

The User Interface in Figure 5.3 is made of a visualizer where images are displayed, the mission is selected and its parameters are setup by the user. Figure 5.4 displays the layout of the main user interface window, where its different sections are shown.

FIGURE 5.4: Hyperspectral imaging tool windows.

The first section, highlighted in blue, is meant for the simulation environment's main functions, such as establishing a connection at a given IP address and port number, starting a specific training load from memory and opening the flight mission window to setup a custom mission, as it will be later detailed. Highlighted in green is the section related to the UAV platform, where it can be issued a takeoff command at a predefined height above ground, land the drone, position it in front of the harvester at a given distance and start the *follow function* routine (where the drone stays ahead of the harvester a selected distance), and inspecting the field. At the bottom of this section, a drop-down menu is available where the targeted hyperspectral sensor is selected. The section on the right, in red, is the graphics window where synthetic hyperspectral images are displayed in real time, as they are being generated while a mission is running. If a target detection mission is being executed, the window is split into two, in order to show the false RGB composition of the synthetic image on one display, and on the other one, the real time processing result obtained with the applied anomaly detection algorithm. Finally, the bottom panel, highlighted in orange, displays relevant information for the user in a logging window.

The Drone Control module in Figure 5.3 creates an instance of the AgSim drone and implements functions such as take-off and land, return home, perform a flight mission

according to the defined waypoints, and a Proportional Integral Derivative (PID) control that allows the drone to follow another vehicle, i.e, the combine-harvester. These implemented functions are based on Application Programming Interface (API) drone functions that will be discussed later in the section.

As it was mentioned before, the Hyperspectral Camera module runs in a different process than the other main three modules, hence a data sharing policy has to be implemented for the other processes to access the captured images, and a synchronization mechanism has to be set between the capturing process and the flight in order to capture images at the desired sectors. The block that facilitates such implementation is the *memmap*, also displayed in Figure 5.3. It mainly consists of a memory block that buffers a predefined number of captured frames into the system RAM memory as well as other signals, such as the camera trigger and the image data type, among others. The memory mapped by the *memmap* block is accessible by all other processes running in parallel within the application software, permitting, for instance, that the image visualization, processing or storing in memory of data take place simultaneously. This block is implemented using the homonymous library from Python which essentially works as a shared memory that is initialized with a unique id and the required memory size.

Regarding AgSim, as it was previously introduced, it includes an API that can be used for externally controlling the machinery through available commands and read out sensor signals. This API connects to the environment software via a TCP-IP tunnel and it is programmed using Python language. It is mainly used to support external models and external sensor simulations, enabling the possibility of defining and running test cases which interact with the environment. Table 5.1 presents the main API functions to control the drone. First, a drone instance has to be created by setting a proper connection, using the right IP address and port, and afterwards, the control commands and signals are ready to be called.

As mentioned before, AgSim offers the capability of reading out pixel label information that can be transformed into sensor data. The information provided per line of pixels by the tool depends on how many materials from a predefined list of materials are present in the line, which materials are they, and in which proportions are they mixed in each pixel

Function	Description
addRotation	Set rotation angle to turn the drone on Z-axis.
getPixelData	Get hyperspectral pushbroom camera pixel data.
getPose	Get position and orientation of a drone.
isLandingGearDown	Check if the landing gear is down.
setConnection	Set Creanex simulator remote services instance for the module.
setDroneEngineRunningState	Set drone engine to on or off.
setLandingGearState	Set drone landing gear state.
setVelocity	Set velocity vector for the drone.

TABLE 5.1: AgSim API drone functions.

of the sensed line. With this information and the stored hyperspectral signature of each material, the tool is able to generate the synthetic hyperspectral images that emulate the targeted scenario. Figure 5.5 shows how the material label assignment works. In Figure 5.5a, the scenario from the drone perspective is shown, where it can be seen that some objects have been placed in the environment, such as a car, some rocks and two mooses located in the center of the image. In Figure 5.5b just the material label image is shown, highlighting in different colors each material group. Last but not least, it is worth to mention that the same procedure has been applied to other sensors mounted on the harvester, such as radar sensors [37].

a) Simulation environment scenario b) Material label

FIGURE 5.5: View from the drone perspective.

5.5 Creation of Hyperspectral Scenarios

The proposed simulation environment works with two hyperspectral scenarios aided by the generation of synthetic images in two different fashions. On one hand, the scenarios created to test hyperspectral anomaly detection algorithms. On the other hand, the generation of hyperspectral maps based on a real grayscale bitmaps overlayed on top of the virtual scenario, that allow testing among others, geometric correction algorithms.

5.5.1 Target detection scenarios

The creation of synthetic target detection scenarios is based on the process described in [146], with a set of adjustments that will be discussed throughout this section, due to the fact the process was not conceived for real time applications. This synthetic image generation process can be split into two parts: the background generation and the insertion of anomalies.

The background generation follows the linear unmixing principle [159]. This is based on the idea that each captured pixel in a hyperspectral image, $M = [r_1, r_2, ..., r_{N_p}]$, which is composed by N_p pixels with N_b spectral bands each, can be represented as a linear combination of a set of p spectrally pure constituent spectra or endmembers, e, weighted by an abundance factor, a, that establishes the proportion of each endmember in the pixel under inspection. This is shown in Equation (5.1), where e_j represents the j-th endmember signature, p is the number of endmembers in the image and $a_{i,j}$ is the abundance of endmember e_j in the pixel r_i. The noise present in the pixel r_i is contained in the vector n_i.

$$r_i = \sum_{j=1}^{p} e_j \cdot a_{i,j} + n_i \tag{5.1}$$

For the synthetic image generation, the endmembers are the background signatures stored in memory which are combined based on the generated abundances. There are two main methods to obtain the abundances as described in [146], Legendre and Gaussian

fields, and within the latter, four extra options appear, depending on the selected type: Spheric, Exponential, Rational and Mattern. In this work, Spheric Gaussian fields have been extensively used due to the pattern resemblance to a cropping field and its higher adaptability to a real time calculation thanks to the slightly simpler operations involved. The steps followed are described below:

a) The number of endmembers used to create the background is set to 4. Endmembers are selected from real images taken over a vineyard and from a database, as it will be described later .

b) The simulated abundance image for each endmember is computed as a Gaussian random field following the procedure proposed in [160]. Since this process is computationally intensive, it has been split into the calculation of a matrix called *lambda*, performed offline using different *theta* values which define the size of the gaussian spheres. The rest of the process, which contains the random part, is executed in real time. The *lambda* matrices are stored in memory and the one selected by the user loaded during run-time.

c) The abundances of each pixel are normalized. The greatest abundance coefficient value is preserved, normalizing the remaining coefficients to ensure that the sum of the abundance coefficients gives one as a result.

Generating the abundances for the background of an image of size 1024×100 pixels takes about two seconds, clearly beyond the simulator minimum step time of 20 ms, which has been established after running tests on machines with different characteristics with the goal of ensuring that no frame is dropped. For this reason, the background generation is performed in a different block that is launched in a separate thread that runs in parallel. Every time it is executed, a buffer of b lines is generated, so when the main process needs a new line to create a new hyperspectral synthetic frame, a line from the buffer is withdrawn. When the number of lines in the buffer goes below a predefined threshold, the background generation function is invoked again to fill it up.

The insertion of anomalies is the last step in the synthetic generation process. Spectral signatures of the labels available in AgSim are stored in memory and every time a specific label is present in the field, its spectra is inserted in the image at the specific pixel location. In order to make it more realistic, the final pixel array is a combination of the background and the anomaly spectra equally proportioned.

The spectra to be used in the synthetic generation of hyperspectral images must be carefully selected. In this particular case, both real spectra captured with the developed platform presented in Chapter 2, based on the Specim FX10 camera, and from a spectral library collected from the United States Geological Survey (USGS) [145] have been used. Since a specific hyperspectral sensor is being tested, signatures not captured with this camera shall be interpolated to its wavelengths. The FX10 has 224 bands spanned in the range of 400 nm to 1000 nm. However, due to the low SNR of the first and last spectral bands, bands 1-11 and 191-224 have been removed, so that only 180 bands are retained. Figure 5.6 displays the spectra used to obtain the background, represented on chart 5.6a, and some of the spectra selected for the anomalies, represented on chart 5.6b. For instance, the spectra named "car", "person" and "vinyard_vegetation" were obtained on a real flight at 45 meters of height above the ground, during the flight campaigns performed over the Bentayga vineyards.

a) Background Spectra b) Anomaly Spectra

FIGURE 5.6: Background and anomaly spectra.

Figure 5.7 shows examples of the synthetic image generation process in which the virtual tool inserts different objects. For the representation of the synthetic images, a false RGB composition has been created. In Figure 5.7a, the drone view of the field is represented with a tractor incorporated to the scene. The synthetic image of that view from the drone is represented in Figure 5.7b. Figure 5.7c shows two rocks located on the virtual scene and the synthetic image of that view is represented on Figure 5.7d. Finally, in Figure 5.7e, a human being has been placed on the scene and its synthetic image is shown in Figure 5.7f. For clarity, the anomalies placed on the synthetic image has been circled in red and pointed with an arrow.

a) Tractor representation

b) Synthetic image with the tractor

c) Rocks representation

d) Synthetic image with rocks

e) Person representation

f) Synthetic image with a person

FIGURE 5.7: Synthetic image generation based on field ground-truth. On the left hand side captions of the simulator environment view from the drone with some different objects are represented. On the right hand side, the RGB representation of the synthetic image produced from that view are shown.

5.5.2 Hyperspectral map scenarios based on real bitmaps

The creation of hyperspectral maps from a set of image swathes taken over a field faces several challenges in the real world, in terms of variations in illumination and disturbances suffered by the platform during the flight. For that reason, in Chapter 3 a whole pre-processing chain was proposed, in order to prepare the images for future information extraction processes.

Among all the steps described in Chapter 3, the most critical one, due to the complexity that it involves, is the georectification process, described in Section 3.6. In this sense, there are two considerations that have to be taken into account. On the one hand, it is paramount to have a steady drone flight to achieve a proper image capturing. On the other hand, together with acquisition of the image it is essential to capture position and orientation information with enough accuracy, to be later used in the georectification process.

In this Section, creation of flight mission scenarios within the proposed virtual environment is covered in detail, with the purpose of acquiring image swathes based on real bitmap images inserted onto the tool. Additional position and orientation data are captured together with the image, the same way it is done with a real UAV platform, in order to gather enough information for a pre-processing phase. As it was previously mentioned, the image capturing process and the drone flight control are completely decoupled from each other, therefore, both the frame timestamp and the position and orientation timestamp are stored separately in different files to later perform an interpolation of the position and orientation values.

First, the flight mission preparation is described. It must be highlighted that the exact same procedure detailed in Section 2.6.1 has been followed and adapted to Python language and the AgSim API, presented earlier in Section 5.3. Figure 5.8 shows the user interface created for the purpose of selecting a region of interest to be scouted and laying out the waypoints.

The steps followed for the waypoints layout are defined below:

a) Selection of corner waypoints b) Waypoints layout

FIGURE 5.8: Mission control application window built over the simulation environment map layout.

1. First, the user selects the diagonally opposed rectangle corners providing the result shown in Figure 5.8a. After that, the other two corners are internally calculated and added to the interface.

2. Second, the shortest distance of the created rectangle is selected to lay down additional waypoints in between those corners. In this way, each individual flight swath has a bigger length and therefore, the flying time is reduced. It has to be mentioned that distances in AgSim are defined in meters, being the reference coordinates located at the center position of the virtual environment.

3. Third, when the user clicks the setup button, a new window appears requesting the following parameters:

   - Relative height, h, from the ground, in meters
   - Frames per second, in FPS
   - Overlapping of captured swathes, in %, in order to facilitate a mosaicking process

4. Fourth, based on the inputs provided by the user in step 3, the image width in meters is obtained by multiplying the resolution by the spatial sampling of the camera and corrected with the input overlap.

5. Finally, the number of waypoints to be added in between the corner waypoints is calculated and added to the user interface, as indicated in Figure 5.8b.

The calculation of the number of waypoints is performed using the formulas defined in Equations 2.3 and 2.4.

Based on the given inputs defined before, the drone mission flight speed is obtained in order to capture the whole scene, using the expression defined in Equation 2.2.

As previously indicated, the tool lets the user define a bitmap pattern in order to simulate a real scenario. In this particular case, an image taken from the football field located at the University of Las Palmas de Gran Canaria campus has been used, due to the fact that it presents straight lines which are an excellent visual inspection tool to check distortions in the captured swathes. Figure 5.9 shows the pattern image on the left, taken with a Phantom 4 UAV at a height of 80 meters and on the right hand side, the image has been placed in the upper right corner, on top of the cropping field. The location coordinates of the football field center point are 28°04'17.9"N 15°27'25.9"W, obtained with the GPS sensor on-board the Phantom 4.

The procedure to import the image to AgSim is done manually, before the software is even started, consisting in two main steps. First, the image has to be converted to grayscale values. Secondly, in the AgSim configuration files, the simulation scenario corner positions where the pattern is to be placed are defined in meters. For this purpose, it is important to know the width and height of the imported image in order to avoid having any distortions in the final result.

Obtaining the width and height of the image in meters requires knowing the FOV and the image resolution. In this case, the Phantom 4 has a camera FOV with 94° in diagonal, and a resolution of 3000 × 4000 pixels. Based on these values, the image taken at 80 meters of height has a width of 103 meters and a length of 137 meters.

FIGURE 5.9: Image of the football field located at the University of Las Palmas de Gran Canaria campus. Image taken with a Phantom 4 drone from Dji at a height of 80 meters.

Now, if the bitmap layer is enabled in the simulation environment, when the drone flies on top of the area where the bitmap has been placed, an extra layer of information will be given to the acquisition camera, containing the array of values corresponding to the real image.

As already outlined, the utilization of real images permits the creation of more realistic scenarios with multiple purposes. On the one hand, the utilization of bitmaps, ranged from 0 to 1, instead of other type of images, allows a direct creation of hyperspectral datasets based on vegetation index patterns. If for instance, the pattern is an already obtained result of a Normalize Difference Vegetation Index (NDVI) [161], ranged between 0 and 1 as well, pixels of the pattern can be replaced by spectra from a database that has the same NDVI value. This would enable the simulation of vegetation fields with areas presenting different levels of vigorousness.

On the other hand, as specified, the georectification process is within the pre-processing phase the one that entails a higher complexity. Therefore, the simulation environment can be used to train the algorithms to improve the quality of the results. Additionally, a real time implementation of those algorithms might be desired if the system has to deliver immediately the position of a specific sector of the image after its processing.

5.6 Experimental Results

In Section 5.5, the possible scenarios to be created with the proposed hyperspectral simulation environment were outlined. In this Section, experimental results of both scenarios are reported.

First, the anomaly detection algorithm LbL-AD, presented in Chapter 4 is tested using the scenarios introduced in Section 5.5.1, with the purpose of proving that the algorithm performs satisfactorily in a faithful representation of the real target scenario.

Secondly, a flight mission is performed in the scenario presented in Section 5.5.2, obtaining a set of image swathes and position and orientation data that is displayed in data charts for an overall evaluation of the flight steadiness and stability.

5.6.1 Anomaly Detection Algorithms

Anomaly Detection (AD) algorithms are certainly the best choice for detecting distinct spectral signatures, significantly deviating from their neighborhoods, as already discussed in Chapter 4. As AD algorithms do not require any prior feature or information of the interested targets and do not have any spectral matching limitations, they constitute a more practical solution to find the targets in actual applications.

These features have been of great interest in the HSI processing domain, what has turned into the emergence of several AD implementations. Most of these implementations are not suitable for applications that demand a real-time response though, since they require

the sensing of the whole hyperspectral image before starting with the process of finding the anomalies in the captured scene.

For applications under strict real-time constraints, in which the captured images must be processed in a short period of time, it is more efficient if the anomalies are uncovered as soon as the hyperspectral data are sensed. For that purpose, in Chapter 4 a novel algorithm named Line-by-line Anomaly Detection (LbL-AD) was introduced to cope with this situation. There are other implementations in the literature as well, aiming to solve that problem [141, 150].

In the next lines, the results of the LbL-AD in the scenarios introduced in Section 5.5.1 are reported. The LbL-AD implementation has a set of fixed parameters, as discussed in Section 4.5.1, that have to be set before its execution. Next, the results of the algorithm varying the parameter d, which is the number of eigenvalues and eigenvectors to be calculated when applying the transformation to the new subspace, are displayed. Table 5.2 shows the results of a simulation run spanned in 51.5 seconds, in which 1030 frames were captured and the mentioned d parameter set to 5. Table 5.3 in its turn, shows the results of a simulation run spanned in 44.85 seconds, in which 897 frames were captured and the mentioned d parameter set to 4.

		Ground truth		**Positive rate**
		✚	▬	
Results	✚	171442	56	0.9967
	▬	1017	1036503	0.999

Table 5.2: Simulation run of 51.5 seconds, 1030 frames captured and d set to 5.

The displayed values on the tables are the True Positives (TP) on the top left hand side and the False Negatives (FN) on the bottom left hand side. On the right hand side, from top to bottom, False Positives (FP) and True Negatives (TN) are respectively displayed.

		Ground truth		Positive rate
		✚	▬	
Results	✚	1587	15	0.9906
	▬	12321	904605	0.9866

TABLE 5.3: Simulation run of 44.85 seconds, 897 frames captured and d set to 4.

In order to obtain normalized metrics that let us compare the results between different simulation runs, a True Positive Rates (TPR) and a False Positive Rates (FPR) have been defined as:

$$TPR = \frac{TP}{TP + FP} \tag{5.2}$$

$$FPR = \frac{FP}{TP + FP} \tag{5.3}$$

The TPR is the probability that the anomaly is present when the test is positive and the FPR is the probability that the anomaly is not present when the test is negative. As it is seen in the result displayed in Table 5.2 and Table 5.3, when the number of eigenvalues is decreased the number of false negatives substantially increases, mainly because the new calculated subspace does not contain enough information to detect all the anomaly signatures. This translates into a lower FPR, which for the specific application of the ENABLE-S3 project is quite critical since one of the goals is the avoidance of fatal accidents. Therefore, for such application, it is desirable to achieve a FPR as close to 1 as possible.

Figure 5.10 visually presents the results previously shown in Table 5.2, with parameter d set to 5. It has to be highlighted that in order to produce these results and the ones shown in Table 5.2, the output of the algorithm has been binarized using a threshold. In Figure 5.10a the detection of a tractor is depicted, showing some false negatives, probably due to a non-optimal threshold set. Figure 5.10b and Figure 5.10c show the results of a

rock detection and person detection, respectively. In both cases, just a few false positives are present in the result.

a) Tractor detection results

b) Rock detection results

c) Person detection results

FIGURE 5.10: Three captions of the simulation run with results shown in table 5.2, with parameter d set to 5.

5.6.2 Flight mission results

UAV platforms equipped with vision sensors are normally used to scan large fields that require the platform to split the area in several sectors which are assembled together in an ulterior processing stage. In case of dealing with pushbroom sensors the situation becomes even more complex since the captured frames/lines within each sector share no overlapping so they are subjected to miss-alignments as the flight of the platform is unfortunately not entirely steady due to internal unbalanced forces that appear in the system and adverse weather conditions, such as wind, among others.

In this Section the flight results of a mission performed in the scenario presented in Section 5.5.2 are presented. Table 5.4 depicts the mission parameters, with the inputs defined by the user on the left hand side, and on the right hand side the values calculated using equations (2.1)-(2.4). After having calculated all the required parameters, the application proceeds to layout the mission waypoints, which matches the setup already presented in Figure 5.8b.

Inputs	Height [m]	50	**Outputs**	Speed [m/s]	0.84
	Frame rate [FPS]	25		Resolution [m]	0.0336
	Overlap [%]	40		Scanned Area [m^2]	31519.88
	Area [m^2]	14773.47		Number Waypoints [-]	10

Table 5.4: Waypoint mission setup.

As mentioned before, during the mission, the system records important data that serves both to analyse the flight stability and accuracy, as well as input to a possible geometric correction phase. The flight performance is assessed in the same way it was done for the real UAV platform, in Section 2.7, using the same metrics and variables with one exception. In the simulation environment, the camera stabilizer or gimbal has not been modelled, hence, it is assumed that the camera is fixed to the bottom of the chassis of the drone. This implies that instead of gimbal angles, drone angles have been logged during the flight.

In Figure 5.11, the drone orientation angle values recorded during the flight are shown. Roll and pitch angles are depicted in Figure 5.11a. Discontinuities in the chart come from the fact that data are just recorded in the long sectors when images are also being captured, therefore, only five different sectors are displayed in the Figure 5.11. Drone heading angle values are plotted in a separate graph (Figure 5.11b) due to the fact that it changes from 0° to 180°, when it travels in reverse direction, producing a much noticeable scale difference in the Y-axis compared to the other two orientation signals. Roll and pitch angles displayed in Figure 5.11a show a slight peak at the beginning of the sector but then remain constant throughout each sector.

a) Drone pitch and roll angles

b) Drone heading angle

FIGURE 5.11: Mission flight drone orientation angles.

In Table 5.5, the metrics discussed in Section 2.7.1 have been calculated for the drone angles displayed in Figure 5.11. The mean and standard deviation values obtained for the drone heading have been calculated removing the first few data points in each sector to ignore the angle changeover that is appreciated in Figure 5.11b, sectors 2 and 4. In general, the angles obtained are another confirmation of a very stable flight.

	Roll $[deg]$	Pitch $[deg]$	Heading $[deg]$		
	μ	μ	target	μ	σ
Sector 1	0.0	0.01	0.00	-0.33	0.12
Sector 2	0.0	0.02	180.00	178.47	0.13
Sector 3	0.0	0.02	0.00	1.57	0.13
Sector 4	0.0	0.01	180.00	178.86	0.10
Sector 5	0.0	0.01	0.00	1.18	0.10

TABLE 5.5: Flight mission data metrics.

In Figure 5.12 data plots of altitude and trajectory recorded values are displayed. In Figure 5.12a, drone altitude data charts, of the same five sectors, are shown. As it is appreciated, altitude does not vary at all throughout the entire flight mission for the fact that in the environment, so far no disturbances have been modelled. The drone real trajectory versus the ideal one is presented in Figure 5.12b, in blue and red, respectively. The real trajectory does deviate from the ideal trajectory, as this has more to do with the drone flight control than the environment conditions.

a) Drone altitude above ground b) Drone trajectory

FIGURE 5.12: Mission flight drone altitude above ground and trajectory.

In Figure 5.13, the flight speed is displayed over each individual sector. As it is seen, speed values present a noticeable oscillations in all sectors, which is caused by the method utilized to estimate the speed value. This consists in simply dividing the travelled distance by the time elapsed. Unfortunately, time elapsed is calculated as the difference between the time stamps of two consecutive samples, and this time stamp, collected from the computer where the simulation is running, does not have a good accuracy.

FIGURE 5.13: Mission flight speed.

Table 5.6 summarizes the results displayed in Figures 5.12 and 5.13, using the metrics discussed in 2.7.1. The deviations in the real trajectory are confirmed by looking at both the Root Mean Square Error (RMSE) and max value per sector. Nonetheless, it is still a minor deviation which shall not affect the quality of the result, just a slight rotation between image swathes is to be expected.

The speed calculated mean value per sector does also deviate significantly from the target value, but this is, as explained before, caused by the calculation method.

	Trajectory $[m]$		Altitude $[m]$			Speed $[m/s]$		
	RMSE	max	target	μ	σ	target	μ	σ
Sector 1	3.01	3.45	50	50.09	0.0	0.84	0.97	0.06
Sector 2	1.70	3.87		50.09	0.0		0.88	0.08
Sector 3	1.77	3.98		50.09	0.0		0.87	0.06
Sector 4	1.27	2.97		50.09	0.0		0.87	0.07
Sector 5	1.31	3.03		50.09	0.0		0.94	0.05

Table 5.6: Table 5.5 continuation.

Evaluating the entire flight mission data, it is confirmed that this is indeed a very stable and steady flight. Comparing it with the flight mission results obtained by the developed real flying platform based on a UAV with a hyperspectral camera as payload, reported in Section 2.7.3, variations in altitude above ground and angles are less significant or even non existent in the simulated environment.

The trajectory does look more alike a real situation deviating from the ideal path. However, it is appreciated that the drone still follows a straight line, which means that it is just an offset and/or an angle difference that remains constant throughout the entire sector.

Therefore, from these charts and metrics, it can be inferred that the intra-sector miss-alignments are going to be negligible and the captured lines shall fit well one after the other without applying a geometric correction algorithm, which is unfortunately not the usual case in a real scenario.

In Figure 5.14, the captured swathes per each sector are displayed. There were 10 waypoints layout in the map, thus, 5 swathes were captured. The images shown prove that

the assumption made before about the lack of intra-sector distortions, was correct. There is as well a slight rotation distortion between sectors that was foreseen by looking at the chart displayed in Figure 5.12b. Anyhow, this effect is not corrected in the flight control code, since it is preferable to have a straight line rather than rotations and changes of trajectory in the middle of the sector in order to facilitate the post-processing algorithm's task.

As it was introduced in Section 5.5.2, with the acquired image data, different type of algorithms can be tested. On the one hand, if spectral signatures from a library are used to replace the captured values based on predefined levels, synthetic hyperspectral datasets can be generated to test, for instance, vegetation index expressions, classification algorithms, segmentation algorithms and any other that could eventually profit from such synthetic datasets.

On the other hand, another interesting use for such acquired data, is the test of the georeferencing and geometric correction algorithms. These processes require not only the image bitarray information but also the position and orientation of the platform at the time every capture takes place. Using these data, the image is projected onto a plane on the ground, performing the necessary interpolations to fill in the existing gaps.

The information related to the position and orientation of the platform can be easily obtained in the simulation environment, as it has been already discussed, with a high degree of accuracy. Nonetheless, there is a minor problem in the captured position data, it is in meters relative to the center of the AgSim environment map. In order to use the data for the georreferencing of the image in the real world, latitude and longitude coordinates should be given by the environment.

In order to test the geometric correction algorithms, the position and orientation data shall be altered, for instance, by adding some distortions or by obtaining samples in a lower rate than the image capturing frame rate, to test the robustness of such algorithms. Otherwise, in the current situation, their application makes no sense as there are no apparent distortions in the captured images.

FIGURE 5.14: Captured swathes.

The aim of the developed work has been to present a methodology for testing algorithms using virtual environments. We certainly acknowledge that the proposed environment should be further improved, trying to bring it closer to a real situation by, for instance, developing a more sophisticated physical model of the flying platform that behaves more

realistic, incorporating weather phenomena, such as wind and dust, and providing position coordinates that are easily extrapolated to the real world, among others.

5.7 Conclusions

Validation and verification (V&V) processes during the development cycle of any commercial product represent a high percentage of its success as they define the product functionality and its reliability. Autonomous systems require a tough battery of testing in different scenarios in order to ensure low risk affecting humans and/or costly equipment. However, there is a balance between financial aspects and time to market when conducting these tests in the real world, and repeating exactly the same testing conditions, time after time, is not an easy task and in many times impossible to reproduce. This is especially true for the particular case of UAVs, in which climate (mainly wind and rain) is unpredictable and affect the performance of these flying platforms. A new level of difficulty is introduced depending on the target application. For example, using UAVs and spectral sensors in precision farming means dealing in some cases with temporary croplands which in the worst case feature a 1-year growing cycle.

In order to deal with all of these variables, in this Chapter a hyperspectral simulation environment has been presented to analyse the virtual behaviour of a UAV equipped with a pushbroom hyperspectral camera used for harvesting applications. As result, this tool alleviates the process of testing different phases involved in the automation of future farming such as image generation, health status inspection and target detection.

The target detection scenario has been tested with the LbL-AD algorithm presented in Chapter 4, reaffirming the overall good detection performance, and what is more important, the suitability of the implementation for real-time anomaly detection, as the simulation environment has permitted the testing of the algorithm under the same constraints it would face in reality.

As a future line, the perturbations from wind and other climate phenomena should be brought into the physical model of the AgSim environment to be able to have an even

closer representation of real scenarios with additional distortions, testing the algorithms in tougher conditions.

Chapter 6

Application Use Case: Precision Farming

This chapter presents the results of applying the developed anomaly detection implementation together with other state-of-the-art algorithms to the captured images for a specific domain: precision farming. More precisely, vegetation index, classification and abundance maps are generated from the images captured by the flying platform, contrasting the results with gathered knowledge from the ground either provided by experts or by captured samples on site. Input images are the ones acquired in the Bentayga vineyards and the vegetation fields in Veneguera, presented in Chapter 3.

6.1 Outline

As defined in the ENABLE-S3 project, the task of developing the platform presented in Chapter 2 has been carried out with the goal of supporting the farming industry improve its processes. By first acquiring hyperspectral datasets of the scanned fields and then processing them, the duty of the system is to provide valuable inputs to the farmers, to help them managing the crop in an effective and efficient way.

This Chapter has therefore the aim of proposing methodologies to extract relevant information from the acquired datasets, using state-of-the-art algorithms in order to exploit the high dimensionality of hyperspectral images as opposed to conventional RGB or multispectral cameras.

First, the Line-by-Line Anomaly Detection (LbL-AD) algorithm, covered in detail in Chapter 4, is applied to the hyperspectral images acquired with the developed UAV based system.

Later, a set of Vegetation Indices (VIs) are calculated. Health status of the crop, soil and vegetation separation or chlorophyll abundance estimation among others, are the information that can be extracted from such indices. In this work, just a selection of the most important ones defined in the literature are applied. Nonetheless, with the available hyperspectral sensor, many more could be obtained.

Then, a set of classification maps are obtained using a well-known linear Support Vector Machines (SVM) classifier, trained with samples defined by the user. The purpose of such maps is to separate the pixels of the image into different classes, obtaining vegetation, soil, and shadow pixels among others.

Finally, several vegetation fields are planted with more than just one vegetation type, and it is certainly valuable to be able to identify each of them, in case they have to be managed differently from each other. This process entails an algorithm chain comprised of several steps as it will be presented.

The rest of the Chapter is organized as follows. First, a state-of-the-art review in precision farming is discussed in Section 6.2. Next, anomaly detection using the algorithm LbL-AD is applied in Section 6.3. The application of VIs to a selected image dataset is presented in Section 6.4, followed by a specific disease analysis of a region with the so-called NDVI. Later, classification maps using linear SVM are obtained in Section 6.5. Then, linear unmixing is performed and abundance maps are obtained, with the aim of distinguishing between different vegetation types, in Section 6.6. Finally, some conclusions are drawn in Section 6.7.

6.2 State-of-the-art in precision farming with unmanned aerial vehicles

Precision farming is defined as a farming management concept based on observing, measuring and responding to inter and intra-field variability in crops. The goal of precision agriculture research is to define a decision support system for a whole crop management with the goal of optimizing returns on inputs while preserving resources.

The aforementioned concept is supported by the information and communication technologies (ICTs) in three different phases: data acquisition, data processing and decision taking. Data acquisition is performed in the field with fixed sensors on the ground supported by the technological expansion of the Internet of Things (IoT), or by aerial platforms capturing information from the air. Data processing consists in transforming the acquired data into a meaningful outcome. Decision taking, in its turn, gathers all processed information from the previous phase and with the support of experts and artificial intelligence techniques, elaborates an action plan.

The UAV platform developed in this work mainly contributes to the first and second phases. Hence, this state-of-the-art analysis will focus on UAV data acquisition and processing to support the precision farming domain.

There are several publications in the scientific literature that provide a review of platforms, sensors and applications. In [162], remote sensing with a UAV is presented as a game changer, opening a wide variety of applications within precision agriculture such as drought stress, nutrient status, growth vigor assessment and yield prediction among others. [163] focuses mainly on the aerial platform technologies and on-board hardware controllers and sensors. In [164], a general overview is provided, covering from the different aerial platform and on-board camera technologies, to some image processing techniques, with special focus on VIs.

Vegetation index maps are indeed one of the direct results that can be obtained with the captured images on-board UAVs, providing valuable easy to interpret information that can be used for multiple purposes. Just to name some, climate trends examination [165], water content of soils remote estimation [166], drought monitoring [167, 168], schedule crop irrigation [169], and crop management [170, 171], among others. Additionally, any other kind of index [172], not necessarily oriented to smart farming applications, can also be calculated, thus increasing the overall applicability of airborne acquisition systems.

Some researchers have taken upon the task of developing a UAV platform based on spectral sensors, for supporting the smart farming domain with the aerial acquisition. However, most of the times these sensors are simply multispectral, which permits the calculation of a reduced number of VIs, as it is the case of [33, 173]. Other works indeed consider a hyperspectral sensor, but again the application is limited exclusively to the calculation of VIs [17].

A few publications can be found that tries to target a different processing method. For instance, in [106] a machine learning approach using an SVM model is applied to the captured data to obtain a high-resolution classification map. Nonetheless, in the document it is claimed that results show large areas of misclassified pixels.

The work presented in the next sections will emphasize different ways of exploiting hyperspectral datasets captured on-board a UAV platform.

6.3 Anomaly Detection

The Line-by-Line Anomaly Detection (LbL-AD) algorithm developed in this Thesis was presented in Chapter 4, where it was tested against images from the state-of-the-art whose ground truth was known in advance and hence, different metrics were obtained to compare the performance of the algorithm with other implementations from the literature. Nonetheless, the main purpose of the LbL-AD algorithm is its use with images captured with the hyperspectral pushbroom camera on-board the UAV platform. For this reason, the performance of the algorithm using images captured with the UAV platform is validated in this Section.

During the flight missions performed over the Veneguera fields and over the Bentayga vineyards, individual members of our team intentionally moved to the captured scene to appear as possible anomalies for a future analysis. Therefore, some of the images presented in Chapter 3 contain anomalies which will be used in this Section.

Concretely, image datasets were selected from mission 2 performed over the Veneguera fields and displayed in Figure 3.22 and from mission 2 performed over the Bentayga vineyards and displayed in Figure 3.25. All selected image subsets have the same spatial size of 825×1024 pixels.

Anomaly detection analysis is very sensitive to noise, therefore those bands presenting a low Signal-to-Noise Ratio (SNR) have been removed. As indicated in the study introduced in Section 3.5, the bands 1-10 located at the lower end, and 191-224 located at the upper end, have these undesired characteristics. They have been removed, so that, out of the total 224 captured bands, 180 have been retained. This procedure is applied to all images used in this Chapter from here on, as noise can potentially introduce errors in the results.

The exact location of the anomalies was not acquired during the flight campaign, which means the position of the anomalies is not known in advance. Therefore, accurate ground truth maps for the captured datasets are not available. Since anomalous pixels cover a very small percentage of the total image, creating the ground truth maps manually would potentially introduce errors in the whole process. For that reason, in this Section

the metrics proposed in Section 4.4.3 are not obtained and the evaluation is performed visually. In the selected datasets, the region where the anomaly is located is circled with a considerable margin, so the test consists in checking whether the anomalous pixels are detected and properly placed within that circle.

Figure 6.1 presents the results of the LbL-AD algorithm tested against two image subsets selected from mission 2 performed over Bentayga vineyards. Image 1, displayed in Figure 6.1a, has been selected from swath 4, of the images shown in Figure 3.25. The anomaly is located in the center slightly shifted to the left. Image 2, displayed in Figure 6.1b, has been selected from swath 5. The image contains two anomalies, a panel made of a metal material located in the bottom center of the image and a person located in the right corner. In both images, anomalies have been marked with blue circles to show them clearly.

Figures 6.1c and 6.1d, show the results of applying the LbL-AD algorithm onto the images 1 and 2, respectively. Results span from 0 to 1, where values closer to 0 are classified as background and values closer to 1 as anomaly. In order to improve the readability of the results performing a better visual separation between the background and the anomalies, Figures 6.1e and 6.1f show the same results displayed in Figures 6.1c and 6.1d, masking out all pixels below 0.2. The pixels above that threshold are represented following the same colormap as before. It is assumed that all background pixels are concentrated between 0 and 0.2, which is certainly true for algorithms that generate results with a good separability between background and anomalous pixels, as it is the case of the LbL-AD.

Based on the images displayed in Figure 6.1, the obtained results are satisfactory, as anomalous pixels are detected and properly placed within the areas marked by the circles. A calculation of the number of pixels above the selected masking threshold in image results presented in Figures 6.1e and 6.1f, reports a total of 89 and 901, respectively.

Figure 6.2 presents the results of the LbL-AD algorithm tested against two image subsets selected from mission 2 performed over Veneguera fields. Image 1, displayed in Figure 6.2a, has been selected from swath 3 of the images shown in Figure 3.22. The anomaly is located in the upper right part of the image. Image 2, displayed in Figure 6.2b, has been

FIGURE 6.1: LbL-AD validation against UAV hyperspectral images captured over Bentayga vineyards, mission 2. (a) Image 1, cropped from swath 4. (b) Image 2, cropped from swath 5. (c) LbL-AD result image 1. (d) LbL-AD result image 2. (e) Result displayed in (c) masking out the background. (f) Result displayed in (d) masking out the background.

selected from swath 1. The anomaly is located in the center left part of the image. In both subsets, the anomalies, marked with blue circles for a better readability, consist of a person in a banana trees plantation.

FIGURE 6.2: LbL-AD validation against UAV hyperspectral images captured over Veneguera fields, mission 2. (a) Image 1, cropped from swath 3. (b) Image 2, cropped from swath 1. (c) LbL-AD result image 1. (d) LbL-AD result image 2. (e) Result displayed in (c) masking out the background. (f) Result displayed in (d) masking out the background.

Figures 6.2c and 6.2d, show the results of applying the LbL-AD algorithm onto the images 1 and 2, respectively. Results span from 0 to 1, values closer to 0 are classified as background and values closer to 1 as anomaly. In order to improve the readability of the results performing a better visual separation between the background and the anomalies, Figures 6.2e and 6.2f show the same results displayed in Figures 6.2c and 6.2d, masking out all pixels below 0.2, which means all background pixels. The pixels above that threshold are represented following the same colormap as before.

As it was shown in the previous case, the results presented in Figure 6.2 display a proper detection of the anomalous pixels. Nonetheless, some extra pixels have been marked as well as possible anomalies, out of the indicated blue circles. These pixels have the characteristic of a high brightness and therefore reflectance value, which bring them to the radar of the AD algorithms. In any case, the obtained result for those pixels is much lower than the marked anomalies, which reach values close to 1.

A calculation of the number of pixels above the selected masking threshold in Figure 6.2e reports a total of 85 pixels, and in Figure 6.2f a total of 100 pixels, after applying the threshold.

In general, it can be said that the LbL-AD implementation performs well in detecting anomalous pixels in the UAV acquired images, which makes the algorithm a good candidate to be programmed in the on-board device, the Jetson TK1, for an application running on real time.

6.4 Vegetation Index Maps

The calculation of VIs is simply performed by combining two or more spectral bands designed to enhance the contribution of vegetation properties, permitting an spatial and temporal assessment of terrestrial photosynthetic activity and canopy structural variations. As already discussed, the fact of carrying a hyperspectral scanner on-board the UAV, gives the possibility of making a more exhaustive evaluation, since it provides

the advantage of calculating any VI whose involved bands are within the VNIR spectral range.

In this work a selection of some of the most important VIs from the state-of-the-art are applied. Table 6.1 displays a summary of five of these VIs with their respective formulas. The well-known NDVI [174], quantifies vegetation by measuring the difference between near-infrared (which vegetation strongly reflects) and red light (which vegetation absorbs). This index is often used in research related to regional and global vegetation assessments, giving an orientation of the health condition of the vegetation under evaluation. It has shown to be related not only to canopy structure and Leaf Area Index (LAI) but also to canopy photosynthesis.

The Modified Soil-Adjusted Vegetation Index (MSAVI)[175] is an evolution of the original SAVI [53], whose main contribution is to emphasize the distinction between soil and vegetation. Since NDVI has some deficiencies in describing the spectral behaviour of vegetation and soil background, the SAVI proposed relationship seeks to improve the sensitivity of the latter. The modified version utilized in this work simply replaces a factor from the original SAVI, with a variable function dependent on the wavelength bands. This improvement reduces the impact of bare soil on the original expression.

Going one step further, in [176] a new modification of the original SAVI was proposed, defining the Optimized Soil-Adjusted Vegetation Index (OSAVI), whose main advantage is not depending on the soil line and being able to eliminate the influence of the soil background effectively.

An indication of the abundance of chlorophyll in the plant is also very valuable for certain assessments. According to the spectral reflectance of plant leaves, between 550 nm and 700 nm, it can be considered constant even if the chlorophyll content of leaves is variable. Based on this relationship, in [54] an index called Chlorophyll Absorption Ratio Index (CARI) was proposed, to estimate the plant chlorophyll content by measuring the level of absorption at 670 nm and linked the reflection peak at 700 nm and 550 nm. Later, a modified version of the CARI (MCARI), was developed in [177], trying to increase sensitivity to leaf chlorophyll concentrations.

Water stress is another important phenomena that has to be spotted in its early stage to avoid plant damage. Some authors have proposed the use of the infrared thermal radiation for its detection, using for instance a combination of thermal sensors with visible imagery, as detailed in [178]. However, when just a VNIR sensor is available, other band combinations have to be applied to detect that phenomena. In [179], the Red-edge ratio index is used to asses water stress in cotton along with other useful crop characteristics.

Vegetation Index	Equation
NDVI	$(R_{800} - R_{670})/(R_{800} + R_{670})$
MSAVI	$\frac{1}{2}\left[2 * R_{800} + 1 - \sqrt{(2 * R_{800} + 1)^2 - 8 * (R_{800} - R_{670})}\right]$
OSAVI	$\left[(1 + 0.16) * (R_{800} - R_{670})\right]/(R_{800} + R_{670} + 0.61)$
MCARI	$\left[(R_{700} - R_{670}) - 0.2 * (R_{700} - R_{550})\right] * (R_{700}/R_{670})$
Red-edge ratio	(R_{720}/R_{670})

Table 6.1: Vegetation indices used in this manuscript.

The indices shown in Table 6.1 have been applied to a selected region of image swath 2, captured during mission 1 performed over the Bentayga vineyard, and displayed in Figure 3.24. Results are displayed in Figure 6.3. The RGB representation of the selected region under analysis is shown in Figure 6.3a, where as it is seen, grape vines are present together with common bushes located on the upper part and right hand side of the image. An NDVI map is displayed in Figure 6.3b, highlighting vegetation from soil and indicating a healthy status of the vines in comparison with the tree located on the upper right corner. In Figure 6.3c, an MSAVI map is displayed, where a distinction between soil and background is made but with some errors, specially on the upper right hand side of the image. The result obtained applying the OSAVI index, shown in Figure 6.3d, is able to correct the mentioned errors, giving a much better outcome, with vegetation having a value close to 1.

In Figure 6.3e, the MCARI index is applied, which as explained, gives an orientation of the chlorophyll abundance in the plants. As it is appreciated, the vines clearly have higher values than the rest of the vegetation of the image, which is yet another indication of the health status of the crop. Finally, in Figure 6.3f, a red-edge ratio index map is

a) RGB Representation.

b) NDVI.

c) MSAVI.

d) OSAVI.

e) MCARI.

f) Red-edge ratio.

FIGURE 6.3: Vegetation indices applied to selected region of image swath 2, mission 1 over Bentayga Vineyard, displayed in Figure 3.24.

shown, providing lower values for the vines located in the middle and bottom left side of the image, which could eventually mean a higher water stress for that area.

A software package tool developed in Matlab has been created in this Thesis to read the raw images, perform the calibration, display the images in a user interface application and obtain any of the displayed results shown in Figure 6.3, for a selected region of the image, or the complete one. The generated vegetation index maps were, as a matter of fact, discussed with the experts on the field, working in the Bentayga vineyards, who validated them and indeed, found them useful as an orientation tool to know the status of the crop by just performing one automatic flight that takes up a few minutes.

A numerical analysis of the vegetation index results is provided in Table 6.2, where for each image the minimum, maximum and mean values are provided. The NDVI formula theoretically spans the result in the range $\{-1, 1\}$. However, in most elements the NIR reflection is higher than the red and therefore, the practical results tend to be positive, as indicated in Table 6.2, with the minimum value being very close to zero. The OSAVI practical values are also contained within the range $\{0, 1\}$, with a very few values going below and above that range, as indicated by the maximum and minimum values obtained and displayed in Table 6.2. The other three indices, MCARI, MSAVI and Red-edge ratio, do have limits beyond the mentioned range, so in order to properly displayed the data for a visual analysis, the values had to be normalized to produce the images shown in Figure 6.3.

Vegetation Index	Min	Max	Mean
NDVI	-0.058	0.947	0.442
MSAVI	1.027	2.125	1.431
OSAVI	-0.0673	1.098	0.513
MCARI	1.206×10^3	1.081×10^5	3.292×10^4
Red-edge ratio	0.892	13.238	2.714

Table 6.2: Vegetation indices obtained values.

After visually analysing the OSAVI index result, a deeper numerical study is performed in order to check whether vegetation and soil can be separated from each other, producing

a mask that could be applied to obtain further processing results. Figure 6.4 presents the results that have been used for that study. On the one hand, the OSAVI result histogram is shown in Figure 6.4a, where it is seen that soil pixels are clearly grouped at the lower end of the chart, between 0 and 0.4. The threshold where vegetation starts is not clearly defined and in this case it has been set to 0.6, to produce the mask displayed in Figure 6.4b.

a) Histogram b) Obtained mask

FIGURE 6.4: Soil and vegetation separation using the OSAVI index.

Depending whether it is desired to keep all vegetation pixels within the mask, at the cost of including some possible background pixels as well, or if it is desired to be certain to leave out all soil pixels, the threshold can be shifted left or right, respectively, producing a slightly different result than the one shown in Figure 6.4b.

6.4.1 Fusarium infection of banana trees

Fusarium wilt of banana, popularly known as *Panama disease*, is a lethal fungal disease caused by the soil-borne fungus called, *Fusarium oxysporum f. sp. cubense*. It is the first disease of bananas to have spread globally in the first half of the 20th century. The epidemic started in Central America, but it has spread to other regions in Asia and Africa during the last years.

Part of the field inspected by the flying platform, located in Veneguera, in the South-West of Gran Canaria, is unfortunately affected by this disease as it was informed by the experts on the field. More precisely, the flight mission 2, displayed in Figure 3.22, was performed over a field in which the dominant vegetation sort is banana, concentrated in the upper part of the captured images.

This fungus does not only affect bananas, but other crops as well, such as wheat, rye, oat and barley, among others. In the literature there are several works that have analysed this disease with the use of hyperspectral imaging, using either samples collected in the field and examining them in the laboratory or with airborne sensors. For instance, in [180] a hyperspectral sensor is used in the laboratory to detect *fusarium*, in its early stage on wheat. In [181], a combination of hyperspectral and thermal imaging is used for the detection of *fusarium*, again on wheat. But the use of spectral images to detect the *fusarium* does not end with the laboratory analyses, as some researchers have exploited the possibility of using airborne sensors for the detection of the disease, using, for instance, VIs [182].

One of the main issues of this fungus is that it spreads very quickly, so an early detection is of essence in order to take upon the necessary measurements. When dealing with large extension of fields, an aerial platform, as the one developed in this work, can be very useful performing several flights that allow the inspection of the field in almost an instant.

In this case, NDVI has been selected to make the analysis. First, samples on the field have been collected to perform a deeper evaluation in the laboratory with a setup based on a Headwall VNIR E-Series, a pushbroom camera with 1600 spatial pixels, covering the spectral range 400nm - 1000nm spread in 974 bands, with a spectral resolution of 3nm. Results of the laboratory analysis are displayed in Figure 6.5. Two leaf samples have been obtained from the field, one healthy leaf, shown in Figure 6.5a and one clearly affected by the fungus, shown in Figure 6.5c.

NDVI results of the healthy and infected leaves are displayed in Figures 6.5b and 6.5d, respectively. Index values for the areas located in the center of the ill leaf are marginally lower than the ones obtained for the healthy leaf, which is indicated in Figure 6.5d by a

yellowish tendency of the red part. The main differences are on the edges of the ill leaf, that clearly show a very low NDVI value.

a) Healthy banana tree leaf.

b) Healthy banana tree NDVI result.

c) Banana tree leaf affected by the Panama disease.

d) NDVI result of a banana tree leaf affected by the fungus.

FIGURE 6.5: NDVI analysis of two banana tree leaves, one healthy and one affected by the fungus.

A further inspection of the spectra in the different zones of both leaves is displayed in Figure 6.6. The spectra of six points per zone have been averaged and shown in the chart. For the healthy leaf just 6 points have been collected, in the locations indicated in Figure 6.5a, in blue. For the ill leaf, in its turn, a total of 12 points have been selected, 6 points that are in the center zones of the leaf apparently less affected by the fungus, and 6 points from the edges of the leaf, very affected by the fungus. The exact locations of the selected image pixels is displayed in Figure 6.5c, in green and red colors.

The spectra plotted confirms the conclusion drawn before by looking at the NDVI result. The areas of the leaf less affected by the fungus have a very similar spectra compared to the healthy one. In fact, there is just a slight offset deviation between the two of them. The spectra in the edges of the infected leaf clearly shows a different pattern than the other two.

Finally, the NDVI analysis of the aerial images captured by the UAV platform are presented in Figure 6.7, where original images displayed in Figure 3.22 have been cropped to just show the banana plantations. It must be highlighted that one of the difficulties

FIGURE 6.6: Spectral reflectance of the dots indicated in Figures 6.5a and 6.5c.

that show up when analysing aerial images is the fact that the image pixel resolution is a few centimetres, which means that a pixel can contain more than one element.

Nonetheless, the results displayed in Figure 6.7 show an evident pattern, which is that on the right hand side of image swath 2 and right bottom side of image 3, the NDVI values of the vegetation plants decrease considerably. This pattern has been confirmed by the experts, which mentioned that the part located closer to the south, that is the one previously indicated, reveal clearer symptoms of the Fusarium disease. As a matter of fact, the ill leaf was collected from that area.

In general terms, VI results can not be taken as an absolute truth, but rather as useful information to focus on a specific area. One of the best approaches when dealing with such maps is to generate them every couple of weeks at least, to be able to evaluate the evolution. In this case, just a single flight mission has been performed over that field, but the intention is to keep monitoring it with subsequent flights that have been scheduled for the coming months.

FIGURE 6.7: Banana plantation UAV image analysis.

6.5 Hyperspectral Image Classification: Labelled maps

The advantage of having a hyperspectral sensor over a multispectral one, is that in addition to the calculation of different vegetation indices, as shown in the Section 6.4, the hyperspectral data collected by the system developed in this work is potentially useful for many other applications that may benefit from the captured data on-board a UAV.

In this section, hyperspectral image classification is applied onto the captured data to obtain labelled maps. Classification algorithms work in the following way: given a set of observations with known class labels, the goal is to assign a class label to each pixel. Hence, it is a supervised method, in which the algorithm is fed with information known in advance.

In this particular case, neither individual pixels nor complete region ground truth are available. For that reason, the labelling is performed visually on the hyperspectral image RGB representation. This indeed has the consequence of producing a less accurate result. Nonetheless, for the purpose of showing the processing capabilities of the acquired images, the procedure that is explained in the next lines shall suffice.

There are several classification methods in the state-of-the-art, being the more traditional ones focused on exploring the pixel spectral signature. Some new ones that have appeared in recent years are exploiting the spatial features as well. These new ones are more sophisticated, entail more complexity, and require a deep analysis of the problem at hand for their application. For all these reasons, in this work it has been opted for the first approach.

Among the pixelwise classification methods, the focus has been set to the Support Vector Machine (SVM) classifier, due to the potentialities it offers in hyperdimensional feature spaces, as described in [57], where a comparison with other methods has been performed, highlighting the effectiveness of SVM classifiers. In addition, they offer a good trade-off between simplicity of the algorithm and quality of the achieved results. A proof of the

aforementioned simple operations involved in the algorithm is the existence of numerous hardware implementations of the classifier found in the literature [183–185].

Considering that it is a supervised binary classification problem, the mathematical formulation of the linear SVM can be expressed as follows. Assuming a training set of N vectors from the d-dimensional feature space $x_i \in \mathbb{R}^d (i = 1, 2, ..., N)$, a target $y_i \in \{-1, 1\}$ is associated to each vector x_i. If the two classes are linearly separable, this means that it is possible to find at least one hyperplane (linear surface) defined by a vector $w \in \mathbb{R}^d$, normal to the hyperplane, and a bias $b \in \mathbb{R}$, that can perfectly split the two classes.

$$f(x) = w \cdot x + b \tag{6.1}$$

The membership decision rule can be formulated using the hyperplane associated formula, as expressed in 6.1. The SVM approach consists in finding the optimal hyperplane, estimating w and b, that maximizes the distance between the closest training sample and the separating hyperplane, which turns out to be determined as the solution of the convex quadratic programming problem.

$$\begin{cases} minimize : \dfrac{1}{2} \|w\|^2 \\ subject\ to : y_i(w \cdot x + b) \geq 1, \qquad i = 1, 2, ..., N \end{cases} \tag{6.2}$$

The classical linearly constrained optimization problem displayed in Equation 6.2 can be translated into a dual problem, using a Lagrangian formulation.

$$\begin{cases} maximize : \sum_{i=1}^{N} \alpha_i - \dfrac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \\ subject\ to : \sum_{i=1}^{N} \alpha_i y_i = 0\, and\, \alpha_i \geq 0, \qquad i = 1, 2, ..., N \end{cases} \tag{6.3}$$

The α terms in Equation 6.3 are the so-called Lagrange multipliers, and the discriminant function associated to the optimal hyperplane can be defined as a function of them as follows:

$$f(x) = \sum_{i \in S} \alpha_i y_i (x_i \cdot x) + b \tag{6.4}$$

where S in Equation 6.4 is the subset of training samples corresponding to the nonzero Lagrange multipliers. It is worth mentioning that the Lagrange multipliers effectively weight each training sample according to its importance in determining the discriminant function. The training samples associated to nonzero weights are called support vectors and lie at a distance exactly equal to $1/\|w\|$ from the optimal separating hyperplane.

There are other variants of the SVM method, such as the non-linear kernel version or linear non-separable case. However, these methods were not explored, as the method described before produced good enough results in the tested scenarios. In order to obtain such results, an existing library implementing SVM [186] has been utilized within the Matlab environment.

Two data subsets have been extracted from images captured over the Bentayga vineyards. First image subset and classification result are presented in Figure 6.8. In Figure 6.8a the selected image region is presented. This area has been selected from the first swath of images displayed in Figure 3.24, obtaining a dataset of 1024×1024 pixels. The classification map is displayed in Figure 6.8b, where red color has been used to represent the pixels classified as soil, green for the vines, in this case of Tintilla grape sort, blue for the shadows and black for the wall, present at the bottom of the scene.

As mentioned earlier, the pixels are manually selected with the aid of a Graphical User Interface (GUI), and are overlaid on top of the RGB image represented in Figure 6.8a. In this case, a total of 49 pixel samples are selected of which 33 have been randomly selected for training and the rest 16 for testing, an approximate 65%-35% split ratio. Table 6.3 displays the total number of pixels per class and the accuracy classifying the test pixels. As it can be appreciated at the bottom of the table, the overall accuracy for this particular case is 100%.

The aforementioned numeric result is supported by a visual inspection, which certifies that the classes have been appropriately classified. There are a few soil and wall pixels

a) RGB representation b) Classification map

FIGURE 6.8: Linear SVM classification result, over image subset of mission 1 performed over Bentayga vineyards.

class	total samples	training samples	test samples	correctly classified	wrongly classified	accuracy
shadow	12	8	4	4	0	100%
vines	12	8	4	4	0	100%
soil	15	10	5	5	0	100%
wall	10	7	3	3	0	100%
					Overall Accuracy	**100%**

TABLE 6.3: Classification results pixel coverage.

that might have been mixed up, specially at the lower right end of the image. A potential reason is that these two elements have similar spectral signatures.

A summary of the complete image classification results is displayed in Table 6.4, where the absolute amount of pixels per class and their percentage coverage over the whole image area have been calculated. As it was expected, the greatest classes are the ones containing vine and soil pixels.

The training samples per class have been averaged and the outcome signature is represented in Figure 6.9. The same colors from the classification map have been used again to represent each element. Indeed, it can be appreciated that, as previously

class	number of pixels	percentage
shadow	112499	10.73
vines	349280	33.31
soil	574477	54.79
wall	12320	1.17

TABLE 6.4: Classification results, pixel coverage.

mentioned, soil and wall signatures are similar, a factor that can lead to potential missclassifications.

FIGURE 6.9: Sample spectra from the classification map displayed in Figure 6.8.

A second image subset that presents more details and elements than the previous image has been selected to perform another classification test. The image RGB representation and the classification results are displayed in Figure 6.10. In Figure 6.10a the selected image region is shown. This area has been selected from the last swath of images displayed in Figure 3.25, obtaining a dataset of 1024×1024 pixels. The classification map is displayed in Figure 6.10b, where blue color has been used to represent the pixels classified as soil mixed with weed present in the ground, green for the vines, in this case Vijariego grape

sort, red for the concrete, dark for the metal (from the roof of the wine cellar) and light green for the shadows.

a) RGB representation b) Classification map

FIGURE 6.10: Linear SVM classification result, over image subset of mission 2 performed over Bentayga vineyards.

The manually selected pixels have been overlaid on top of the RGB image represented in Figure 6.10a. In this case, a total of 71 pixel samples are selected of which 47 have been randomly selected for training and the rest 24 for testing, once again targeting an approximate 65%-35% split ratio. Table 6.5 displays the total number of pixels per class and the accuracy classifying the test pixels.

In this case, one pixel of the vines class have been missclassified, probably confussed by the algorithm with a soil and weed pixel. The presence of weed vegetation in the soil spectra makes the distinction from the vine pixel slightly fuzzier. The overall accuracy for this particular case, displayed at the bottom of the table, is 95.8%.

A visual check of the obtained classification map confirms as well the quality of the result. As mentioned earlier there are some soil and vine pixels that might have been mixed up, and some pixels on the ground have been wrongly marked as metal, probably for their brightness, similar to the pixels on the roof.

class	total samples	training samples	test samples	correctly classified	wrongly classified	accuracy
soil and weeds	20	13	7	7	0	100%
vines	15	10	5	4	1	80%
concrete	12	8	4	4	0	100%
metal	12	8	4	4	0	100%
shadow	12	8	4	4	0	100%
					Overall Accuracy	**95.8%**

Table 6.5: Classification results pixel coverage.

A summary of the complete image classification results is displayed in Table 6.6, where mainly the absolute amount of pixels per class and their percentage coverage over the whole area have been calculated. In this case, soil pixels cover by far the largest area of the scene, compared to the other four elements.

class	number of pixels	percentage
soil and weeds	681233	64.97
vines	91010	8.68
concrete	148837	14.19
metal	55103	5.26
shadow	72393	6.90

Table 6.6: Classification results pixel coverage.

The training samples per class have been averaged and the outcome signatures are represented in Figure 6.11. The same colors from the classification map have been used to represent each element.

The classification results in the two proposed tested scenarios look very accurate. Nonetheless, as already indicated, these have been obtained in the best possible conditions for the SVM classifier, using training samples collected from the same image that is going to be classified. This indeed constitutes the best possible scenario for classification algorithms.

FIGURE 6.11: Sample spectra from the classification map displayed in Figure 6.10.

In order to test the classifier in tougher conditions, the following test is proposed. The test consists in classifying grape leaf sorts using as training samples the ones already used in the previous tests, which have been collected from sets of images different than the ones that are going to be classified. Flight mission 1 performed in the Bentayga vineyards and displayed in Figure 3.24, was made over a field with a plantation of grape sort named Tintilla. Flight mission 2 performed in the same region and displayed in Figure 3.25, was made over a field with a plantation of grape sort named Vijariego.

A total of four different datasets of size 400×400 pixels containing specially vines of the different sorts, have been selected from the indicated images. These datasets are going to be classified with the pixels samples already collected from the previous datasets namely, the pixels corresponding to Tintilla vegetation sort, selected from Figure 6.8a, Vijariego vegetation sort, selected from Figure 6.10a and the pixels corresponding to soil and shadow selected from Figure 6.10a as well.

In Figure 6.12, classification results of images with Tintilla grapes are displayed. Figures 6.12a and 6.12b show the RGB representation of the images, that have been extracted

from swathes 3 and 4, respectively, from the images displayed in Figure 3.24. Figures 6.12c and 6.12d show the classification maps obtained.

FIGURE 6.12: Grape sort classification. RGB representations of the selected regions containing Tintilla grape sort are displayed in a) and b). Classifications maps are presented in c) and d).

As it can be seen in the results displayed in Figure 6.12, soil, vegetation and shadow are correctly distinguished. However, when identifying the vegetation type, namely Vijariego or Tintilla, specially in the borders of the vine rows, the algorithm makes some errors. This error probably comes from a brightness difference issue. Going back to the spectra plotted in Figures 6.9 and 6.11, looking carefully at the green line, representing the vegetation of the respective grape sort, the curve has a similar profile, but a considerable scale difference. The Tintilla spectra goes up to almost 1 of reflectance value, while Vijariego

spectra reaches just 0.5 of reflectance value. This fact might make the algorithm trip, specially in the borders closer to the shadow, where the brightness of the pixels decreases.

In Figure 6.13, classification results of images with Vijariego grapes are displayed. Figures 6.13a and 6.13b show the RGB representation of the images, that have been extracted from swathes 2 and 3, respectively, from the images displayed in Figure 3.25. Figures 6.13c and 6.13d show the classification maps obtained.

FIGURE 6.13: Grape sort classification. RGB representations of the selected regions containing Vijariego grape sort are displayed in a) and b). Classifications maps are presented in c) and d).

In the results displayed in Figure 6.13, the grape sort has been correctly identified in all vegetation pixels. However, there are some vegetation pixels that have been marked as soil. This is once again due to a brightness issue and the fact that the soil samples contained some pixels with weed that could have misled the algorithm.

The brightness effect, that has misled the classifier in the last results displayed, comes from the fact that the white reference used to calibrate the image was captured with illumination conditions slightly different than the ones available when the flight was performed. In order to correct that phenomena, in the future white references shall be acquired together with an illumination measurement, measured by a photo-resistor sensor. Another photo-resistor shall be installed as well in the drone in order to measure the light conditions at the time images are being captured, and then apply the white reference that matches that same light conditions. This is a task that takes several months of work to complete, as samples have to be collected at various light conditions and different frame rates and exposure times.

6.6 Hyperspectral linear unmixing and abundance maps

In Section 6.5, the last test scenario consisted in identifying two types of vegetation available in different image datasets. In this Section, the intention is to identify different types of vegetation within a single image. Certain vegetation fields contain more than one type of plants, specially in those cases where the field does not have a regular planning from the beginning and plantations were added as time went by. A valuable information for the farmer and the application is to obtain abundance maps where the different types of vegetation might be present, in order to, for instance, apply a specific treatment to each type or do the cropping on a specific area that is populated by that type.

The scientific community has put a vast amount of efforts in trying to achieve the distinction of different elements in a scene, as it constitutes a great application for remote sensing. In [187], the authors emphasize in the difficulty of that task supported by two main reasons, the presence of mixed pixels in the data or the limited information available a priori. In the specific task of differentiating among vegetation types in a scene, in [188] a study of the Alkali landscapes, one of the most extended semi-natural open landscapes of the European Union, was conducted comparing various methods from the state-of-the-art

to separate the various vegetation classes. In this study, it is concluded that, although the achieved results are satisfactory, the task at hand is very complex due to the fact that sometimes spectral noise is higher than the species dissimilarities.

Whenever very limited information is available beforehand, linear unmixing arises as a powerful technique to obtain maps indicating the different elements present in the selected area. This technique was initially developed for figuring out the composition of pixels in hyperspectral datasets acquired from airplanes or satellites, that were usually made up of several pure constituents or endmembers due to the poor spatial resolution. Recently, it has been investigated the use of such approach in complete scenes made up of various elements with subtle differences between each other [189, 190], as it is the case at hand. The goal is to obtain an initial set of classes that can be later used in a second clustering or classification process for a further refinement.

Essentially, in this case it is given from the experts in the field that in the inspected area coexist different types of mango plantations (keitt and osteen varieties) as well as lemon trees. Based on that, the approach is to obtain abundance maps that indicate the presence or abundance of each element in the scene. The whole process is accomplished in a few steps described below:

- In order to simplify the unmixing algorithm and let it focus on just the vegetation pixels, a mask is applied to the image in order to extract vegetation from the rest. For that purpose, the process described at the end of Section 6.4 is applied using the OSAVI index and a threshold. By applying this mask, the next stage of the process will be just fed with vegetation pixels.

- Using an unmixing algorithm that follows the principle described in Section 5.5.1, the endmembers of the image are obtained. In the state-of-the-art literature there are several options available to accomplish that task. For instance, the Fast algorithm for linearly UNmixing hyperspectral images (FUN) [74], with low computational effort requirements, is a good candidate due to the simplicity of the operations involved. Another alternative, the N-finder algorithm [72], consists in performing thousands of projections of the simplex points into random vectors to detect the

vertices of that simplex, which are the endmembers. One of the most well-known algorithms, the Vertex Component Analysis (VCA) [71], assumes the data form a simplex and by performing affine transformations onto them the vertices are figured out. Hardware friendly versions of the VCA can be found at [191, 192], which mainly try to implement the space transformation by applying Gram-Schmidt.

- Finally the abundances for each of the calculated endmembers are obtained, representing the presence of each endmember in every pixel, as described in Section 5.5.1.

The data subset is selected from flight mission 1 performed over Veneguera and displayed in Figure 6.14, a field that contains different types of vegetation, as confirmed by the experts. In Figure 6.14a the dataset of 1024×917 pixels is shown, taken from swath 4 of images displayed in Figure 3.21. After applying the OSAVI vegetation index and a threshold to extract the vegetation pixels, the resulting image is shown in Figure 6.14b.

a) RGB representation b) Masked RGB representation

FIGURE 6.14: Selected image portion from flight mission 1 performed over Veneguera to apply linear unmixing.

The masked image containing just the vegetation pixels is passed to the unmixing algorithm, in this case the FUN algorithm. The reasons for using this algorithm over the other proposed methods in the literature are multiple. Apart from its simplicity already

discussed, the FUN has the characteristic of self determining the number of endmembers of the scene by just defining a stop factor which constitutes the amount of information in the image that can not be represented by the set of endmembers already determined.

Another great advantage of the FUN is the fact that it provides exactly the same endmembers every time it is executed. The other proposals from the literature rely on random factors that make the executions vary from each other.

Figure 6.15 shows how the unrepresented information of the image decreases as the number of extracted endmemembers increases. Concretely, in this case a stop factor of 5% has been selected, as the target is to try to focus on the pixels representing the majority of the image information. Since the image is not completely free of noise, a further reduction of this threshold would imply obtaining additional endmembers that might just represent this undesired noise. As it is seen in the plotted chart, a number of 5 endmembers satisfy the predefined condition of 5% stop factor.

FIGURE 6.15: Representation of the image information representation based on the endmembers determined by the FUN.

Once the endmembers have been obtained, the abundance map for each of them is calculated, in this case using the fully constraint method [73], which implies that the sum of all abundance per pixel shall be 1, and the abundance values can not have negative values. The obtained abundances are displayed in Figure 6.16 for each endmember.

FIGURE 6.16: Abundance maps of the calculated endmembers. a) Endmember 1. b) Endmember 2. c) Endmember 3. d) Endmember 4. e) Endmember 5.

Table 6.7 presents the results of adding all abundance pixels per endmember and the percentage representation of that value. This is calculated in order to get an orientation of the presence of each endmember in the image composition and hence, which endmembers might constitute a different vegetation type and which ones might just be an anomaly or any other effect.

Based on the results displayed in Table 6.7, it can be asserted that out of the five pure constituents of the scene, three of them have a considerably higher presence than the other two, corresponding to endmember 2, endmember 3 and endmember 5. The graphical results shown in Figure 6.16 confirm that assertion. In the abundance map of the first endmember displayed in Figure 6.16a, the borders of the trees are specially highlighted,

Abundance	sum	percentage
1	22709.9	6.3
2	127081.6	35.2
3	96538.6	26.7
4	18625.1	5.2
5	95950.7	26.6

Table 6.7: Total summation of pixels abundances per endmember and the equivalent percentage.

which indicates that the first endmember was obtained because of the brightness of a particular pixel, but that does not mean that it is a representative constituent of the scene.

The abundance map obtained for endmember 4, shown in Figure 6.16d, is somehow harder to interpret. It could represent a group with just one tree, as highlighted in the scene. However, according to the inputs received from the experts, it is very unusual to have a vegetation type in a field with that reduced presence. Normally the vegetation types are arranged in bigger clusters.

From here on, the way to proceed further should be to show the maps to the experts and after inspecting the field, confirm the presence of the three possible different types of vegetation. Then a more sophisticated chain of processing could be defined, with the support of ground truth information and using clustering or classification algorithms, in order to improve the results and provide with a higher degree of certainty the specific location of each individual type.

In Figure 6.17, the calculated endmember signatures are displayed, with the purpose of showing the difficulty that involves the separation of vegetation classes, based on the fact that spectral curves are very much alike. There is mainly a scale difference that sometimes could be caused by brightness pixel variations as it has been discussed in previous sections.

FIGURE 6.17: Spectral signatures of the calculated endmembers.

6.7 Conclusions

In this Chapter different methods have been proposed to extract valuable information from the acquired hyperspectral data with the aerial platform. These methods shall serve as a proof of concept for a deeper analysis, when flight campaigns are performed every couple of weeks, having much more data available, that can be analyzed to find trends. For instance, vegetation indices shall be used with that purpose, as a simple VI snapshot of the area does not really reflect the evolution of the crop, but having a series of VI results over a period of some months, specially before the cropping season, can definitely help its management.

In the literature there are numerous correlation analysis between plant and soil properties and vegetation indices. This is performed by doing several flight campaigns contrasted with the chemical analysis of ground samples collected in the same field. In this process, having a hyperspectral sensor on-board a UAV constitutes yet another advantage as correlations can be tested against many more indices, made up of different bands of the spectra covering from the 400 nm to 1000 nm.

The classification and unmixing processes have shown the potential of hyperspectral data, which shall serve to other purposes than just the calculation of vegetation indices. The methods presented could be eventually complemented with laboratory analysis of leaves with hyperspectral imaging equipment as well, to try to find correlations using state-of-the-art artificial intelligence techniques that for instance could help in the early detection of plagues or diseases.

These are only some examples of the algorithms that can be applied to the acquired hyperspectral images, showing potential applications. These results show promising benefits to the farming community, facilitating automatic processes to be included in future hardware platforms both, on-board and on the ground.

Chapter 7

Conclusions and further research

This Chapter summarizes the main contributions of this Thesis and proposes further research topics, which are expected to complement and enhance the future development of this work.

7.1 Conclusions

Hyperspectral imaging systems are considered nowadays one of the most powerful tools for acquiring surface information in many different fields. These systems are able to provide images where single pixels contain information from across the electromagnetic spectrum of the scene under observation. Whereas the human eye sees just in the visible range of the spectrum, hyperspectral imaging is able to capture information in a wider range, dividing the spectrum into several contiguous bands. The main advantage of this technique is that certain objects leave unique 'fingerprints' in the electromagnetic spectrum, known as spectral signatures, which enable the identification of the materials that make up a scanned object, providing a much better understanding of the scene under analysis than conventional imaging sensors.

Remote sensing is the field in which hyperspectral imaging techniques have traditionally acquired more relevance. In this field, hyperspectral sensors have been typically launched on-board satellites or airplanes for collecting spectral information of the Earth's surface or the different layers of the atmosphere. These data can be used for many different applications in the fields of oceanography, enviromental science, geology, volcanology, soil and land management, atmospheric and aerosol studies, and agriculture among others.

The emergence of unmanned aerial vehicles has meant a breakthrough in remote sensing. Compared to traditional platforms, UAVs present some very attractive advantages for the users, it is a lower cost solution with a more flexible revisit time and a better spatial and spectral resolution, which permits a deeper and more accurate data analysis. Additionally, missions can be customized, which permits the selection of the exact area to be scanned, the flight speed, altitude and orientation among other parameters, making this type of solution a very appealing one, opening a plethora of new applications. Nonetheless, it is worth mentioning that the installation of hyperspectral sensors on-board UAVs would not have been possible without the technological improvements in terms of camera weight reduction and miniaturization.

The use of hyperspectral sensors on-board UAV has been deeply exploited by different research groups, developing platforms that have served for multiple applications in safety,

security and rescue field, civilian applications, such as police departments, fire brigades and other homeland security organizations. It is in the agriculture field though, where remote sensing relying on UAVs has been strongly positioned as an emerging field of application. Nevertheless, the trend has been to build up general purpose hyperspectral platforms for capturing images and later processing them offline, with an important gap between the acquisition and the specific application, missing out several potential advantages that could be offered by the technology. For instance, real time applications are seldom considered in hyperspectral remote sensing, for the lack of some important elements: first, there are no suited platforms equipped with the necessary devices and software to achieve real time, second, algorithms are mainly intended to process complete hypercubes and not adapted to the sensor acquisition, and last, the testing scenarios to carry out system verification and validation are costly and hard to achieve.

This is why a methodology for using hyperspectral technology in a practical application, facilitating the expansion of remote sensing activities, is an important and exciting challenge. In this Thesis a new approach for hyperspectral imaging acquisition, processing and testing has been developed. This includes the development of an acquisition system for its application in a real scenario, new algorithm implementations adapted to the need of the application and virtual scenarios for validating both, flight control code and algorithms execution.

In particular, this Thesis has achieved the contributions detailed below:

- A hyperspectral image acquisition system on-board a UAV has been developed. The commercial drone, Matrice 600 from Dji, has been selected as a flying platform to carry a VNIR sensor, the Specim FX10 hyperspectral pushbroom camera, acquiring information in the 400-1000 nm range. The camera is mounted on the Ronin Mx stabilizer, also from Dji, to soften vibrations and keep it continuously facing downwards. Mechanical adaptations were carried out for a proper fit into the Ronin Mx, as these gimbals are conceived for industrial filming cameras. Flight missions are planned on the ground within an ad-hoc user friendly mobile device application, developed as well within this Thesis, by just clicking a few buttons and then its

execution is performed on-board with a control system that focuses on achieving a steady and stable flight.

In order to verify the behaviour of the developed platform, a novel software package has been implemented to create a mission report that evaluates the flight telemetry data captured by the positioning and orientation sensors mounted on-board. These mission reports display the necessary diagrams to asses the accuracy of the performed mission as well as the calculated assessment metrics over all sectors. An overall analysis of all presented mission flight results indicate that the platform flight accuracy is considerably high, what definitely translates into high quality image acquisition, reducing the efforts in the post-processing phase. Moreover, additional data is acquired, such as a full frame image that can potentially enhance the quality of the measured position and orientation values by applying imaging stereo vision algorithms.

- A pre-processing algorithm chain has been proposed for preparing the raw captured bitstreams from the hyperspectral sensor to calibrate, correct and geoposition hypercubes. Image calibration is achieved using white and dark reference samples captured prior the flight. The radiometric calibration performed by the camera manufacturer has been double-checked in the laboratory, where the Specim FX10 has been mounted to acquire images from a spectrally well characterized polymer. The given spectral curve from that polymer has been compared to the one obtained by the FX10, proving that the peaks are located at the same wavelength values. Signal-to-noise ratio analysis of all the captured spectral bands has been successfully performed and the removal of a few spectral bands corresponding to the edges of the sensor device has been recommended, in order to reduce the noise contamination and achieve better results. Image georectification has been applied in two steps. First, the geoposition of every pixel of the image is calculated using the telemetry data captured with the positioning sensors available in the drone. Second, using the same position coordinates, the image has been geometrically corrected.

The evaluation of the georectification method has been performed laying the obtained corrected image on top of a Google Earth map of the inspected area. The achieved results are very promising, even under adverse weather conditions.

- Two versions of the proposed LbL-AD algorithm have been designed and implemented, one using Matlab environment and the other in Python language, in order to be able to run the algorithm in the simulation environment. Due to the fact that the algorithm is based on the concept of orthogonal subspace projections, employing a low computational processing chain and simplified operations, the software implementations are able to process acquired frames in real time, as it was proven in the implemented version for the simulation environment, where the acquisition frame rate is limited to 50 FPS. Accuracy of the results is not compromised by the mentioned operations simplification. On the contrary, as it has been demonstrated, the proposed implementation obtains outstanding results.

  Detection and execution performances of the algorithm have been extensively compared with state-of-the-art proposals, not only with those processing the whole image but also the ones doing it in a line-by-line fashion. Synthetic and real HSI images from the literature were used in the numerous experiments that were performed, producing visual results that span from 0 to 1, which corresponds to background and anomalous pixels, respectively. Numerical results were also generated using different proposed assessment metrics. Overall, the proposed LbL-AD obtains excellent results with the advantage of processing the lines of pixels as they are captured and using relatively simple operations.

- A hyperspectral simulation environment has been created, built on top of the pre-existing agriculture graphical simulation environment, AgSim, able to model and simulate machinery and its sensors. The proposed environment virtually creates scenarios that are a faithful representation of the reality and permit testing of hyperspectral processing algorithms. Two main hyperspectral scenarios have been developed aided by the generation of synthetic images in two different fashions: target detection scenarios created to test hyperspectral anomaly detection algorithms and simulated flight mission scenarios, where the generation

of hyperspectral maps is based on a real grayscale bitmap overlaid on top of the virtual environment.

The target detection scenarios have been tested with the proposed LbL-AD algorithm, reaffirming the overall good detection performance, and what is more important, the suitability of the implementation for real-time anomaly detection, as the simulation environment has permitted the testing of the algorithm under the same constraints it would face in reality.

- As part of the farming use case of the European ENABLE-S3 project, where these contributions have been applied, flight missions have been mainly performed over vegetation fields and acquired images have been processed to obtain valuable information about those fields. Image anomalies have been successfully detected with the application of the LbL-AD algorithm. A set of VIs have been applied to obtain colour maps indicating plant stress levels, with the application of the NDVI, chlorophyll content, with the application of the MCARI, distinction of vegetation from soil background, with the MSAVI and OSAVI, and estimation of plant water stress with the Red-edge ratio. A particular analysis of a region was done with the application of the NDVI, where it was known in advance that it was affected by a fungus. Some specific areas of the entire analysed region did show some hints of the disease, fact that was confirmed by the experts on the field. Classification maps have been obtained using a linear SVM classifier in order to distinguish vegetation and other elements in the scene. Linear unmixing has been applied to a portion of the image of which was just known in advance that it contained more than one vegetation type, with the objective of figuring out its composition.

The presented research work achieves the main objectives set for this Thesis. Hyperspectral image acquisition on-board a UAV platform has been successfully attained, performing several flight campaigns over agricultural fields. The acquired data have been analysed and pre-processed for its preparation for the ulterior information extraction algorithms. A set of algorithms have been applied to the images to detect anomalies, obtain crop status maps, indicating health and vigorousness, classification maps of the

different elements conforming the scene, and abundance maps indicating the composition of the scene when very little information is known in advance. A new implementation of an anomaly detection algorithm for its real time execution has been developed. A virtual environment for the verification of the algorithm implementations in a realistic scenario has been created. The accomplished experiments prove that remote sensing could be more than just earth monitoring of large surfaces every once in a while. The scenario has changed considerably with the appearance of unmanned aerial vehicles and the possibility of customizing the systems, which enables even the utilization of such technology in applications that demand real time executions for a rapid response.

7.2 Further research work

Many of the proposed topics in this manuscript can be further enhanced in order to improve the achieved results and manageability of the developed system. This will be outlined in this Section as part of future research lines of work.

Regarding the acquisition of hyperspectral images, flight campaigns shall be performed every few weeks over the same terrains, in order to capture enough data to evaluate trends and evolutions of the crop. The acquired data should be complemented with several white references taken at different light conditions and camera parameters, such as frame rate and exposure time, in order to achieve a better calibration much more homogeneous throughout the whole captured hypercubes. Additionally, the frame camera installed in the platform to improve the accuracy of the hyperspectral geometric correction algorithms by obtaining for every captured frame its position and orientation, should be further adjusted and tested. A proper filter shall be mounted in front of the lens that allows improving the quality of the acquired images, reducing the blurring effect.

Improvements in the geometric correction algorithms shall come by applying the structure from motion (SfM) process to the captured frame images and correctly assign the obtained position and orientation to the corresponding hyperspectral line. For the application of SfM, as it was already mentioned, commercial tools could not suffice, as the whole

processing chain might require some adaptations to fit the amount of images fed to the process. This implies a deeper understanding of the whole algorithm sequence for its proper modification.

Regarding the algorithmic development, specially the proposed LbL-AD, hardware implementations into GPU based boards and/or FPGA based boards should be considered in order to further exploit the parallelisms offered by the algorithm as well as the hardware friendly operations involved. A hardware implementation of the algorithm would definitely speed up its execution what would let it meet real time executions with very high acquisition frame rates.

Furthermore, it should be explored the possibility of updating the flying platform on-board PC. There are very interesting embedded boards, both GPU and FPGA based, recently released, such as the Jetson Nano, the Pynq or the Zynqberry, which offer a more advanced system on chip solution, with higher processing capabilities, lower consumption, and a hardware structure that allows a deeper exploitation of parallel task execution. These are all very appealing characteristics, that would definitely bring the platform one step further.

The work presented in this Thesis has focused on providing results to a particular use case, precision farming. However, there are other very interesting applications that should be explored, as they could deeply benefit from the contributions provided by aerial monitoring and supervision tasks. As our group is particularly concerned about climate change and the impact it is having in the world, environmental related applications are currently being evaluated to be part of the group research activities portfolio.

It is worth mentioning that minimal changes in the presented platform could lead to other interesting applications, providing an added value to this prototype in terms of flexibility, modularity and robustness. Indeed, in the same FX camera series from Specim, there is a SWIR unit, the Specim FX17, that acquires information in the 900-1700 nm range, and its installation in the platform would be straight forward, expanding even more the potential target domains.

The simulation environment might be improved for a better system validation and verification. A more sophisticated physical model of the drone with the inclusion of weather phenomena into the environment, more realistic scenarios closer to real ones with the addition of using GPS real position coordinates in the control system and the inclusion of hardware elements into the system under test to have hardware in the loop simulations, are among others the areas to be further improved.

Further research work is already being carried out related to the development of our own multi and hyperspectral camera prototypes, which will bring more flexibility to our implementations in the future. Two multispectral cameras are currently under development. The first one consists in a wheel of filters that rotates during the acquisition to capture a frame at every selected wavelength. The other device has as many sensors as filters, and captures the multispectral image at one shot. The first version is arranged with four sensors and the same amount of filters. The positive aspect of the selected design is the interchangeability of filters which allows targeting different applications. Additionally, a hyperspectral camera prototype is under development with special focus on the optics. The model is much lighter and cheaper than commercial available solutions and hence, better suited for airborne mount. On top of that, it provides a software API valid for any Linux based system.

Moreover, flying platforms based on the integration of individual components supported by open source software packages are as well being explored, in our seek of further increasing overall system flexibility. Currently Ardupilot developer site [112] offers many interesting options to create complete aerial vehicles, being a much lower cost approach than the currently utilized Matrice 600. This approach provides more possibilities in terms of the flight controller to integrate in the platform, its programming API, the sensors to be used for positioning and orientation, and the overall electrical design that depends on the model and number of motors, propellers and batteries to be used. This will have an impact in the payload capacity of the platform which in its turn will allow the installation of one or many imaging sensors on-board.

The development of the aforementioned open-source and low cost cameras and drones is part of one of the research interests of our group as it is the development of a swarm of

drones, which together with artificial intelligence algorithms will enrich the portfolio of applications to be offered to the society.

Appendix A

Sinopsis en español

En este Capítulo se presenta una visión general del trabajo de investigación realizado en esta Tesis Doctoral, poniendo especial interés en las contribuciones en el campo de la adquisición, procesamiento y validación de imágenes hiperespectrales para sistemas basados en plataformas aéreas no tripuladas.

A.1 Introducción

Las imágenes multiespectrales e hiperespectrales se forman captando la energía reflejada o emitida por objetos terrestres en un gran número de longitudes de onda, dando como resultado un cubo de datos que contiene desde unas pocas hasta cientos de bandas. Tradicionalmente estas imágenes han sido adquiridas a bordo de un satélite o avión, que eran las únicas plataformas capaces de transportar cargas útiles pesadas, y transmitirlas posteriormente a una estación terrestre para ser procesadas.

En la actualidad, el continuo desarrollo de los sensores de visión, la óptica y la electrónica que soporta la captura de estas cámaras, ha permitido una disminución tanto en tamaño como en peso de los dispositivos, haciéndolos mucho más portables e incluso competitivos en precio. Este hecho, junto con la aparición de aeronaves no tripuladas, ha permitido la inclusión de cámaras multi- e hiperespectrales a bordo de dichas aeronaves, lo que supone una revolución en el área de la teledetección.

El binomio formado por plataformas aéreas no tripuladas y cámaras multi- e hiperespectrales puede ser empleado en multitud de aplicaciones de supervisión y monitorización. Algunas de las aplicaciones más destacadas en las que ya se está empleando este binomio con éxito es en la inspección de torres y líneas de alta tensión para la prevención de fallos en el sistema, ejecución de diferentes misiones en los ámbitos de seguridad y salvamento, o en tareas de detección de incendios operando como soporte de los equipos en tierra.

Sin embargo, es importante destacar, que la inclusión de este tipo de cámaras en plataformas aéreas no tripuladas no está exenta de retos tecnológicos, especialmente cuando las cámaras son hiperespectrales de barrido lineal (*pushbroom*), que actualmente dominan el mercado por las buenas resoluciones espectrales que ofrecen. Al realizar la caputura línea a línea, el vuelo tiene que ir perfectamente acompasado con la adquisición y se debe reducir al máximo posibles vibraciones a las que pueda estar sometida la cámara, para evitar distorsiones en la imagen capturada.

En los últimos años han aparecido plataformas de este tipo, basadas en sensores hiperespectrales, que se han caracterizado por dar servicios de manera genérica a los distintos dominios de aplicación, acquiriendo imágenes por un lado y procesándolas a posteriori una vez ha concluido el vuelo. No obstante, de esta manera no se aprovechan todas las posibilidades que podrían ofrecer tales soluciones tecnológicas, como por ejemplo, dando servicio a determinadas aplicaciones que requieren de una respuesta inmediata.

Por estos motivos expuestos, en esta Tesis doctoral, se realizan una serie de aportaciones que permiten acercar y adecuar la plataforma de vuelo a la aplicación específica. Hasta ahora el procedimiento que se ha seguido por parte de la comunidad científica, ha sido por un lado, llevar a cabo el desarrollo de la plataforma con los sensores y dispositivos electrónicos necesarios, y por otro, el diseño e implementación de algoritmos de procesamiento, de manera completamente desacoplada, una tarea de la otra. Lo que se pretende en esta Tesis es proporcionar una metodología que posibilite eliminar ese vacío que queda entre la adquisición de las imágenes y el procesamiento de las mismas para la extracción de información relevante para una aplicación concreta.

En concreto, esta Tesis contribuye a todas las fases implicadas en una aplicación real. En primer lugar, se construye una plataforma aérea no tripulada que incorpora un sensor de barrido lineal, con la cual se obtienen excelentes resultados de captura de datos, tanto imágenes como de telemetría, gracias a los esfuerzos realizados en la preparación de misiones de manera automatizada y el control de vuelo. En segundo lugar, algoritmos específicos que se adaptan a la forma de captura de la cámara instalada han sido desarrollados para la detección de anomalías y correcciones geométricas de la imagen. A su vez, algoritmos del estado del arte han sido seleccionados para extraer resultados de las imágenes capturadas proporcionando información útil a la aplicación, agricultura de precisión. Finalmente, se ha desarrollado un entorno de simulación hiperespectral para validar y verificar los algoritmos que se han creado conjuntamente con el vuelo del dron, en escenarios que resultan fieles representaciones de la realidad.

A.2 Objetivos y metodología de trabajo

La revolución tecnológica es una realidad que evoluciona de manera exponencial y que cada vez más está presente en más campos de aplicación. La meta principal de este trabajo de Tesis es aportar nuevas soluciones para aumentar el grado de autonomía en los procesos que se llevan a cabo en la industria agrícola, con el propósito de ayudar a aumentar la producción de una manera sostenible, ayudar a los expertos de campo a la toma de decisiones aportando información relevante y evitar accidentes fatales. A continuación se enumeran los objetivos de esta Tesis:

- Construir una plataforma de vuelo basada en un dron y una cámara hiperespectral para realizar campañas de vuelo sobre terrenos agrícolas. Las misiones deben ser ejecutadas de manera completamente autónoma y con mínima o nula intervención humana. El control y la estabilidad de vuelo deben propiciar la acquisición de imágenes de alta calidad.
- Proponer e implementar una cadena de preprocesado que prepare las imágenes adquiridas para la aplicación posterior de algoritmos específicos de extracción de información.
- Desarrollar de un algoritmo de detección de anomalías capaz de procesar la imagen a medida que va siendo capturada por el sensor.
- Desarrollar un entorno virtual de verificación con escenarios realistas para el test de los algoritmos suceptibles de ser implementados a bordo del dron.
- Seleccionar y aplicar los algoritmos adecuados para la obtención de mapas de colores que proporcionen información útil sobre el estado de la cosecha, como por ejemplo, el estado de salud, el nivel de estrés de las plantas, etc. A su vez, generar mapas de clasificación que distingan claramente entre vegetación y los otros elementos de la escena.

La consecución de cada uno de los objetivos planteados se detalla en las secciones que se presentan a continuación.

A.3 Desarrollo de una plataforma de vuelo no tripulada basada en un sensor hiperespectral de barrido lineal

Con el objetivo de adquirir imágenes hiperespectrales de terrenos agrícolas, en esta Tesis, se ha construído una plataforma de vuelo para la adquisición de imágenes hiperespectrales usando vehículos aéreos no tripulados (drones). En concreto, se ha adquirido la solución comercial Matrice Dji 600, para portar la cámara Specim FX10 que adquiere imágenes hiperespectrales en el rango visible e infrarojo cercano (VNIR) λ :400-1000 nm. La cámara se monta en un estabilizador Ronin Mx, también de la empresa Dji, cuyo objetivo es absorber las vibraciones externas que se producen durante giros, cambios bruscos de trayectoria del dron y golpes de viento y para a su vez mantener la cámara apuntando siempre hacia abajo. Las misiones de vuelo se planifican desde el terreno, a través de una aplicación particularizada que ha sido también parte del trabajo de esta Tesis, y se ejecutan en un PC a bordo del dron. El sistema de control de vuelo implementado en el PC de abordo tiene como principal objetivo la consecución de un vuelo estable que permita adquirir imágenes de alta calidad.

En la Figura A.1, se observa la plataforma aérea desarrollada en una de las misiones desarrolladas sobre unos viñedos en la región de Tejeda, Gran Canaria.

Figura A.1: Plataforma de aérea no tripulada con cámara hiperespectral de barrido lineal.

A.4 Adquisición de imágenes hiperespectrales

Con la plataforma de vuelo desarrollada e introducida en la Sección A.3, se han ejecutado una serie de misiones, principalmente sobre terrenos agrícolas. En concreto se ha volado en tres localizaciones distintas:

- Campo de fútbol localizado en el Campus Universitario de Tafira, en Las Palmas de Gran Canaria.
- Campos de cultivo, principalmente árboles de mango, plataneras y aguacateros, localizados en Veneguera, municipio de Mogán, Gran Canaria.
- Bodegas Bentayga, localizadas en el municipio de Tejeda, Gran Canaria.

La Figura A.2 muestra una representación a color de una de las capturas hiperespectrales realizadas durante uno de los vuelos sobre las bodegas Bentayga.

Figura A.2: Representación en color de las capturas hiperespectrales realizadas en una de las misiones con la plataforma de vuelo.

Los datos "en crudo" que se han obtenido de la cámara, deben ser correctamente tratados para después poderles aplicar distintos algoritmos de procesamiento. Para ello, en esta Tesis, se propone una cadena de preprocesamiento que dispone de varios pasos:

- Calibración de la imagen usando referencias de blanco y negro capturadas a priori.
- Calibración radiométrica del sensor usando un material del que se conoce perfectamente su respuesta espectral.
- Estimación de la relación señal a ruido de las bandas del sensor.
- Georectificación de la imagen usando la posición y orientación de la plataforma de vuelo, adquiridos con los sensores adicionales de que dispone la misma como son el GPS y el giróscopo.

A.5 Detección de anomalías

El algoritmo *Line by Line Anomaly Detection* (LbL-AD) ha sido diseñado e implementado para el descubrimiento de píxeles anómalos en la escena a medida que se van adquiriendo las líneas hiperespectrales con la cámara de barrido lineal. Al igual que el algoritmo OSPRX (*"Orthogonal Subspace Projections Reed-Xiaoli"*), el LbL-AD está basado en el concepto de proyecciones ortogonales pero usando una cadena de procesamiento de menor coste computacional, que no obstante garantiza la precisa detección de anomalías presentes en el terreno.

Para la evaluación de la bondad del algoritmo se han realizado comparativas con otros algoritmos del estado del arte, implementaciones que requieren de la captura de toda la imagen para compenzar el procesamiento e implementaciones que al igual que el LbL-AD, realizan el procesamiento línea a línea. Se han empleado como muestras para realizar los tests, imágenes de la literatura científica, de las que se conoce previamente el mapa de verdad.

A.6 Entorno de simulación de algoritmos hiperespectrales

Un nuevo entorno de simulación para la validación de algoritmos que se ejecutan a bordo de la plataforma de vuelo hiperespectral ha sido creado en esta Tesis. El entorno desarrollado es soportado por un dominio denominado *entorno de simulación gráfico para agricultura* (AgSim por sus siglas en inglés), capaz de modelar distintos tipos de maquinaria presentes en la agricultura y sus sensores.

Dentro del entorno de simulación hiperespectral dos tipos de escenarios virtuales han sido creados. Un primer escenario que sirve para la validación de algoritmos de detección de anomalías en tiempo real. Un segundo escenario, que permite verificar la captura de imágenes hiperespectrales en misiones de vuelo automáticas.

A.7 Caso de uso: Agricultura de precisión

Uno de los objetivos de la Tesis, es procesar las imágenes hiperespectrales adquiridas con la plataforma de vuelo con el fin de generar distintos mapas de color que indiquen el estado de la cosecha que se está analizando. Para ello se emplean los conocidos índices de vegetación. En la literatura existen cientos de propuestas de distintos índices que han sido analizados en base a correlaciones con muestras tomadas en el terreno. En el trabajo realizado, se ha seleccionado una muestra de los más empleados. En concreto, el más conocido y utilizado es el *índice de vegetación de diferencia normalizada* (NDVI por sus siglas en inglés). En la Figura A.3, se muestra un ejemplo de la aplicación del NDVI a una región particular de una de las imágenes adquiridas por la plataforma hiperespectral de vuelo.

Además de índices de vegetación, también se han obtenido mapas de clasificación, con el objetivo de diferenciar diferentes elementos de la imagen, entre ellos la vegetación, el terreno, sombras, etc. Asimismo, en una región específica de la que se sabía que coexisten diferentes tipos de vegetación, limoneros y árboles de mango, se realizó un análisis por

FIGURA A.3: Aplicación de índices de vegetación a una imagen hiperespectral. a) Representación en color de la imagen. b) Resultado NDVI.

descomposición lineal de la imagen y se calcularon las abundacias correspondientes a cada píxel principal. El resultado indica la existencia de tres posibles tipos de vegetación en la región analizada.

Finalmente se aplicó el LbL-AD definido anteriormente, para la detección de anomalías en algunas de las imágenes, con un resultado satisfactorio.

A.8 Conclusiones

Las sistemas de adquisición de imágenes hiperespectrales son considerados hoy en día una herramienta muy potente para la captura de información de la superficie del terreno. Estos sistemas son capaces de obtener imágenes en las que cada píxel individual contiene información en un amplio rango del espectro electromagnético de los objetos observados. La principal ventaja de este tipo de técnicas es que se puede considerar que cada objeto deja una 'huella dactilar' en el espectro, conocida como firma espectral, que permite la identificación de los materiales que componen el objeto escaneado, proporcionando una información mucho más completa de la escena a analizar, que los sensores convencionales.

Las imágenes hiperespectrales han tenido una mayor relevancia, sobre todo en el ámbito de la teledetección. En este campo, los sensores se embarcan a bordo de satélites o aviones para recopilar información de la superficie terrestre. La aparición de los vehículos aéreos no tripulados ha significado un gran avance en este campo, ya que estos aportan una serie de ventajas hasta ahora inalcanzables, como la flexibilidad para realizar misiones de vuelo sobre un terreno, un coste mucho mas asequible, una mejor resolución espacial, etc.

En esta Tesis se ha tratado de exprimir todas esta ventajas, e ir un paso más allá en la utilización de estas plataformas en aplicaciones reales. Se ha propuesto una metodología para la adecuada acquisición, procesamiento y test de los diferentes elementos que compone un sistema de esta complejidad. En concreto se ha construido una plataforma de vuelo no tripulada basada en una cámara hiperespectral de barrido lineal, con la que se han realizado multitud de campañas de vuelo sobre diferentes campos agrícolas. Se ha propuesto una cadena de preprocesado para preparar los datos adquiridos para un posterior procesamiento. Se ha desarrollado un nuevo algoritmo para detectar anomalías en tiempo real, procesando las imágenes según son adquiridas. Un entorno de simulación creado también en esta Tesis, ha permitido la validación de algunos de estos algoritmos en escenarios que representan fielmente la realidad. Las imágenes adquiridas han sido procesadas con algoritmos del estado del arte para obtener distintos mapas de información de utilidad para la aplicación de agricultura de precisión.

En definitiva, se ha demostrado con los experimentos llevados a cabo, que la teledetección puede ser mucho más que simplemente la monitorización de largas extensiones de terreno una vez cada cierto tiempo. Con la aparición de las plataformas aéreas no tripuladas, el tablero ha cambiado completamente dando la posibilidad de customizar las plataformas para su utilización en aplicaciones específicas.

Appendix B

Publications

B.1 Journals

[1] P. Horstrand, R. Guerra, A. Rodríguez, M. Díaz, S. López and J. Fco. López. A UAV Platform Based on a Hyperspectral Sensor for Image Capturing and On-Board Processing. In *IEEE Access*, vol. 7, pp. 66919-66938, 2019.

[2] P. Horstrand, M. Díaz, R. Guerra, S. López and J. Fco. López. A Novel Hyperspectral Anomaly Detection Algorithm for Real-Time Applications With Push-Broom Sensors. In *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing.*

[3] P. Horstrand, J. Fco. López, S. López, T. Leppälampi, M. Pusenius and M. Rooker. A Simulation Environment for Validation and Verification of Real Time Hyperspectral Processing Algorithms on-Board a UAV. In *Remote Sens.* 2019, *11*, 1852.

[4] M. Díaz, R. Guerra, P. Horstrand, E. Martel, S. López, J. Fco. López and R. Sarmiento. Real-Time Hyperspectral Image Compression Onto Embedded GPUs. In *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 8, pp. 2792-2809, Aug. 2019.

[5] M. Díaz, R. Guerra, P. Horstrand, S. López and R. Sarmiento. A Line-by-Line Fast Anomaly Detector for Hyperspectral Imagery. In *IEEE Transactions on Geoscience and Remote Sensing.*

[6] S. López, P. Horstrand, G. M. Callicó, J. Fco. López and R. Sarmiento. A Low-Computational-Complexity Algorithm for Hyperspectral Endmember Extraction: Modified Vertex Component Analysis. In *IEEE Geoscience and Remote Sensing Letters*, vol. 9, no. 3, pp. 502-506, May 2012.

[7] S. López, P. Horstrand, G. M. Callicó, J. Fco. López and R. Sarmiento. A Novel Architecture for Hyperspectral Endmember Extraction by Means of the Modified Vertex Component Analysis (MVCA) Algorithm. In *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 6, pp. 1837-1848, Dec. 2012.

B.2 International Conferences

[1] P. Horstrand, S. López and J. Fco. López. (2018). A Novel Implementation of a Hyperspectral Anomaly Detection Algorithm For Real Time Applications With Pushbroom Sensors. In *Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS 2018).*

[2] P. Horstrand, R. Guerra, M. Díaz, A. Morales, A. Jiménez, S. López and J. Fco. López (2019). A spectral imaging system for precision agriculture: from its inception till a pre-commercial prototype. In *Conference on Design of Circuits and Integrated Systems (DCIS 2019).*

[3] R. Guerra, P. Horstrand, A. Rodríguez, M. Díaz, A. Morales, A. Jiménez, S. López and J. Fco. López (2019). Optimal UAV movement control for farming area scanning using hyperspectral pushbroom sensors. In *Conference on Design of Circuits and Integrated Systems (DCIS 2019).*

[4] A. Rodríguez, P. Horstrand, J. Fco. López and S. López. (2018). Setting up an autonomous hyperspectral flying platform for precision agriculture. In *SPIE Sensing Technology + Applications (SPIE 2018).*

[5] M. Rooker, P. Horstrand, A. Rodríguez, S. López, R. Sarmiento, J. Fco. López, R. Lattarulo, J. Pérez, J. Matute, Z. Slavik, D. Pereira, M. Pusenius and T. Leppalampi. (2018). Towards improved validation of autonomous systems for smart farming. In *Workshop on Smart Farming*, CPS Week, Porto, Portugal, 2018

References

[1] Michael Theodore Eismann. Hyperspectral remote sensing. SPIE Bellingham, 2012.

[2] Zhongping Lee and Kendall L Carder. Hyperspectral remote sensing. In *Remote Sensing of Coastal Aquatic Environments*, pages 181–204. Springer, 2007.

[3] Éva Papp and Thomas Cudahy. Hyperspectral remote sensing. *Geophysical and Remote Sensing Methods for Regolith Exploration*, 144:13–21, 2002.

[4] Laura M Dale, André Thewis, Christelle Boudry, Ioan Rotar, Pierre Dardenne, Vincent Baeten, and Juan A Fernández Pierna. Hyperspectral imaging applications in agriculture and agro-food product quality and safety control: a review. *Applied Spectroscopy Reviews*, 48(2):142–159, 2013.

[5] G. Vane, A. F. H. Goetz, and J. B. Wellman. Airborne imaging spectrometer: A new tool for remote sensing. *IEEE Transactions on Geoscience and Remote Sensing*, GE-22(6):546–549, Nov 1984. ISSN 0196-2892. doi: 10.1109/TGRS.1984.6499168.

[6] Gregg Vane, Robert O Green, Thomas G Chrien, Harry T Enmark, Earl G Hansen, and Wallace M Porter. The airborne visible/infrared imaging spectrometer (aviris). *Remote Sensing of Environment*, 44(2):127–143, 1993.

[7] U.S. Geological Survey and NASA. Earth observing 1, hyperion website. [online]. URL `https://eo1.usgs.gov/sensors/hyperion`.

[8] Q. Lv, Y. Dou, X. Niu, J. Xu, and J. Xu. Classification of tiangong-1 hyperspectral remote sensing image via contextual sparse coding. In *2015 International Conference on Machine Learning and Cybernetics (ICMLC)*, volume 1, pages 128–133, July 2015. doi: 10.1109/ICMLC.2015.7340910.

[9] N. Pahlevan, S. Sarkar, B.A. Franz, S.V. Balasubramanian, and J. He. Sentinel-2 multispectral instrument (msi) data processing for aquatic science applications: Demonstrations and validations. *Remote Sensing of Environment*, 201:47 – 56, 2017. ISSN 0034-4257. doi: https://doi.org/10.1016/j.rse.2017.08.033. URL `http://www.sciencedirect.com/science/article/pii/S0034425717303991`.

[10] PRISMA (Hyperspectral Precursor and Application Mission). `https://earth.esa.int/web/eoportal/satellite-missions/p/prisma-hyperspectral`. [Online; accessed 7-August-2019].

[11] Rupert Müller, Martin Bachmann, Christian Chlebek, Harald Krawczyk, Amaia de Miguel, Gintautas Palubinskas, Mathias Schneider, Peter Schwind, Tobias Storch, Valerie Mogulsky, and Bernhard Sang. The enmap hyperspectral satellite mission an overview and selected concepts. 2012.

[12] AA Gowen, CPo O'Donnell, PJ Cullen, G Downey, and JM Frias. Hyperspectral imaging–an emerging process analytical tool for food quality and safety control. *Trends in Food Science & Technology*, 18(12):590–598, 2007.

[13] D Lorente, N Aleixos, J Gómez-Sanchis, S Cubero, Or L García-Navarrete, and J Blasco. Recent advances and applications of hyperspectral imaging for fruit and vegetable quality assessment. *Food and Bioprocess Technology*, 5(4):1121–1142, 2012.

[14] Christian Fischer and Ioanna Kakoulli. Multispectral and hyperspectral imaging technologies in conservation: current research and potential applications. *Studies in Conservation*, 51(sup1):3–16, 2006.

[15] Guolan Lu and Baowei Fei. Medical hyperspectral imaging: a review. *Journal of biomedical optics*, 19(1):010901–010901, 2014.

[16] J. A. J. Berni, P. J. Zarco-Tejada, L. Suarez, and E. Fereres. Thermal and narrowband multispectral remote sensing for vegetation monitoring from an unmanned aerial vehicle. *IEEE Transactions on Geoscience and Remote Sensing*, 47(3):722–738, March 2009. ISSN 0196-2892. doi: 10.1109/TGRS.2008.2010457.

[17] J. Yue, G. Yang, C. Li, Z. Li, Y. Wang, H. Feng, and B. Xu. Estimation of Winter Wheat Above-Ground Biomass Using Unmanned Aerial Vehicle-Based Snapshot Hyperspectral Sensor and Crop Height Improved Models. *Remote Sensing*, 9(708), 2017. doi: 10.3390/rs9070708. URL `https://www.mdpi.com/2072-4292/9/7/708`.

[18] H. Aasen, E. Honkavaara, A. Lucieer, and P.J. Zarco-Tejada. Quantitative Remote Sensing at Ultra-High Resolution with UAV Spectroscopy: A Review of Sensor Technology, Measurement Procedures, and Data Correction Workflows. *Remote Sensing*, 10(1091), 2018. doi: 10.3390/rs10071091. URL `https://www.mdpi.com/2072-4292/10/7/1091`.

[19] Martin Kanning, Insa Kühling, Dieter Trautz, and Thomas Jarmer. High-resolution uav-based hyperspectral imagery for lai and chlorophyll estimations from wheat for yield prediction. *Remote Sensing*, 10(12), 2018. ISSN 2072-4292. doi: 10.3390/rs10122000. URL `http://www.mdpi.com/2072-4292/10/12/2000`.

[20] Juha Suomalainen, Niels Anders, Shahzad Iqbal, Gerbert Roerink, Jappe Franke, Philip Wenting, Dirk Hünniger, Harm Bartholomeus, Rolf Becker, and Lammert Kooistra. A lightweight hyperspectral mapping system and photogrammetric processing chain for unmanned aerial vehicles. *Remote Sensing*, 6(11):11013–11030, 2014. ISSN 2072-4292. doi: 10.3390/rs61111013. URL `http://www.mdpi.com/2072-4292/6/11/11013`.

[21] Zhengrong Li, Yuee Liu, Rodney Walker, Ross Hayward, and Jinglan Zhang. Towards automatic power line detection for a uav surveillance system using pulse coupled neural filter and an improved hough transform. *Machine Vision and Applications*, 21(5):677–686, 2010.

[22] Andreas Birk, Burkhard Wiggerich, Heiko Bülow, Max Pfingsthorn, and Sören Schwertfeger. Safety, security, and rescue missions with an unmanned aerial vehicle (uav). *Journal of Intelligent & Robotic Systems*, 64(1):57–76, 2011.

[23] Kai Daniel and Christian Wietfeld. Using public network infrastructures for uav remote sensing in civilian security operations. Technical report, DORTMUND UNIV (GERMANY FR), 2011.

[24] Telmo Adão, Jonáš Hruška, Luís Pádua, José Bessa, Emanuel Peres, Raul Morais, and Joaquim João Sousa. Hyperspectral imaging: A review on uav-based sensors, data processing and applications for agriculture and forestry. *Remote Sensing*, 9 (11):1110, 2017.

[25] J. Pablo Arroyo-Mora, Margaret Kalacska, Deep Inamdar, Raymond Soffer, Oliver Lucanus, Janine Gorman, Tomas Naprstek, Erica Skye Schaaf, Gabriela Ifimov, Kathryn Elmer, and George Leblanc. Implementation of a uav–hyperspectral pushbroom imager for ecological monitoring. *Drones*, 3(1), 2019. ISSN 2504-446X. doi: 10.3390/drones3010012. URL `https://www.mdpi.com/2504-446X/3/1/12`.

[26] D. Turner, A. Lucieer, M. McCabe, S. Parkes, and I. Clarke. Pushbroom Hyperspectral Imaging from an Unmanned Aircraft System (uas) - Geometric Processing workflow and Accuracy Assessment. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W6:379–384, August 2017. doi: 10.5194/isprs-archives-XLII-2-W6-379-2017.

[27] Seon Jin Kim and Gino J. Lim. A hybrid battery charging approach for drone-aided border surveillance scheduling. *Drones*, 2(4), 2018. ISSN 2504-446X. doi: 10.3390/drones2040038. URL `http://www.mdpi.com/2504-446X/2/4/38`.

[28] Miroslaw Adamski. Analysis of propulsion systems of unmanned aerial vehicles. *Journal of Marine Engineering & Technology*, 16(4):291–297, 2017. doi: 10.1080/20464177.2017.1383337. URL `https://doi.org/10.1080/20464177.2017.1383337`.

[29] Óscar González-Espasandín, Teresa J. Leo, and Emilio Navarro-Arévalo. Fuel cells: A real option for unmanned aerial vehicles propulsion. *The Scientific World Journal*, 2014, Article ID 497642:12, 2014. URL `https://doi.org/10.1155/2014/497642`.

[30] W. Lu, D. Zhang, J. Zhang, T. Li, and T. Hu. Design and implementation of a gasoline-electric hybrid propulsion system for a micro triple tilt-rotor vtol uav. In *2017 6th Data Driven Control and Learning Systems (DDCLS)*, pages 433–438, May 2017. doi: 10.1109/DDCLS.2017.8068112.

[31] FAO. The future of food and agriculture – Alternative pathways to 2050, 2018. URL http://www.fao.org/3/I8429EN/i8429en.pdf.

[32] Kirtan Jha, Aalap Doshi, Poojan Patel, and Manan Shah. A comprehensive review on automation in agriculture using artificial intelligence. *Artificial Intelligence in Agriculture*, 2:1 – 12, 2019. ISSN 2589-7217. doi: https://doi.org/10.1016/j.aiia.2019.05.004. URL http://www.sciencedirect.com/science/article/pii/S2589721719300182.

[33] E. Raymond Hunt Jr. and Craig S. T. Daughtry. What good are unmanned aircraft systems for agricultural remote sensing and precision agriculture? *International Journal of Remote Sensing*, 39(15-16):5345–5376, 2018. doi: 10.1080/01431161.2017.1410300. URL https://doi.org/10.1080/01431161.2017.1410300.

[34] J. Huang, H. Wang, Q. Dai, and D. Han. Analysis of ndvi data for crop identification and yield estimation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(11):4374–4384, Nov 2014. ISSN 1939-1404. doi: 10.1109/JSTARS.2014.2334332.

[35] AVL List GmbH. ENABLE-S3 project, 2016. URL https://www.enable-s3.eu/. [Online; accessed 06-Oct-2019].

[36] M. Rooker, P. Horstrand, A.S. Rodríguez, S. López, R. Sarmiento, J.F. López, R. Lattarulo, J. Pérez, J. Matute, Z. Slavik, D. Pereira, M. Pusenius, and T. Leppalampi. Towards improved validation of autonomous systems for smart farming. In *Workshop on Smart Farming, CPS Week, Porto, Portugal*, April 2018.

[37] M. Holder, P. Rosenberger, H. Winner, T. D'hondt, V. P. Makkapati, M. Maier, H. Schreiber, Z. Magosi, Z. Slavik, O. Bringmann, and W. Rosenstiel. Measurements revealing challenges in radar sensor modeling for virtual validation of autonomous driving. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2616–2622, Nov 2018. doi: 10.1109/ITSC.2018.8569423.

[38] Fred Sigernes, Mikko Syrjäsuo, Rune Storvold, Jo ao Fortuna, Mariusz Eivind Grøtte, and Tor Arne Johansen. Do it yourself hyperspectral imager for handheld

to airborne operations. *Opt. Express*, 26(5):6021–6035, Mar 2018. doi: 10.1364/OE.26.006021. URL http://www.opticsexpress.org/abstract.cfm?URI=oe-26-5-6021.

[39] Ryan Hruska, Jessica Mitchell, Matthew Anderson, and Nancy F. Glenn. Radiometric and geometric analysis of hyperspectral imagery acquired from an unmanned aerial vehicle. *Remote Sensing*, 4(9):2736–2752, 2012. ISSN 2072-4292. doi: 10.3390/rs4092736. URL http://www.mdpi.com/2072-4292/4/9/2736.

[40] Ayman Habib, Youkyung Han, Weifeng Xiong, Fangning He, Zhou Zhang, and Melba Crawford. Automated ortho-rectification of uav-based hyperspectral data over an agricultural field using frame rgb imagery. *Remote Sensing*, 8(796), 2016. doi: 10.3390/rs8100796. URL http://www.mdpi.com/2072-4292/4/9/2736.

[41] Irving S Reed and Xiaoli Yu. Adaptive multiple-band cfar detection of an optical pattern with unknown spectral distribution. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(10):1760–1770, 1990.

[42] A Schaum. Joint subspace detection of hyperspectral targets. In *Aerospace Conference, 2004. Proceedings. 2004 IEEE*, volume 3, page 1824. IEEE, 2004.

[43] Chein-I Chang. Orthogonal subspace projection (osp) revisited: a comprehensive study and analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 43(3): 502–518, March 2005. ISSN 0196-2892. doi: 10.1109/TGRS.2004.839543.

[44] S. Matteoli, M. Diani, and G. Corsini. A tutorial overview of anomaly detection in hyperspectral images. *IEEE Aerospace and Electronic Systems Magazine*, 25(7): 5–28, July 2010. doi: 10.1109/MAES.2010.5546306.

[45] Qiandong Guo, Ruiliang Pu, and Jun Cheng. Anomaly detection from hyperspectral remote sensing imagery. *Geosciences*, 6(4), 2016. ISSN 2076-3263. doi: 10.3390/geosciences6040056. URL https://www.mdpi.com/2076-3263/6/4/56.

[46] M. Ben Salem, K. S. Ettabaa, and M. A. Hamdi. Anomaly detection in hyperspectral imagery: an overview. In *International Image Processing, Applications and Systems Conference*, pages 1–6, Nov 2014. doi: 10.1109/IPAS.2014.7043320.

[47] Jan Hodicky. Autonomous systems operationalization gaps overcome by modelling and simulation. In Jan Hodicky, editor, *Modelling and Simulation for Autonomous Systems*, pages 40–47, Cham, 2016. Springer International Publishing. ISBN 978-3-319-47605-6.

[48] Justin Yapp, Remzi Seker, and Radu Babiceanu. Uav as a service: A network simulation environment to identify performance and security issues for commercial uavs in a coordinated, cooperative environment. In Jan Hodicky, editor, *Modelling and Simulation for Autonomous Systems*, pages 347–355, Cham, 2016. Springer International Publishing. ISBN 978-3-319-47605-6.

[49] C. Coopmans, M. Podhradský, and N. V. Hoffer. Software- and hardware-in-the-loop verification of flight dynamics model and flight control simulation of a fixed-wing unmanned aerial vehicle. In *2015 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, pages 115–122, Nov 2015. doi: 10.1109/RED-UAS.2015.7440998.

[50] Prasad S Thenkabail, Ronald B Smith, and Eddy De Pauw. Hyperspectral vegetation indices and their relationships with agricultural crop characteristics. *Remote Sensing of Environment*, 71(2):158 – 182, 2000. ISSN 0034-4257. doi: https://doi.org/10.1016/S0034-4257(99)00067-X. URL `http://www.sciencedirect.com/science/article/pii/S003442579900067X`.

[51] Clara Rey-Caramés, María P. Diago, M. Pilar Martín, Agustín Lobo, and Javier Tardaguila. Using rpas multi-spectral imagery to characterise vigour, leaf development, yield components and berry composition variability within a vineyard. *Remote Sensing*, 7(11):14458–14481, 2015. ISSN 2072-4292. doi: 10.3390/rs71114458. URL `http://www.mdpi.com/2072-4292/7/11/14458`.

[52] Y. J. Kaufman and D. Tanre. Atmospherically resistant vegetation index (arvi) for eos-modis. *IEEE Transactions on Geoscience and Remote Sensing*, 30(2):261–270, March 1992. doi: 10.1109/36.134076.

[53] A. J. Richardson and C. L. Wiegand. Distinguishing vegetation from soil background information. Technical Report Photogrammetric Engineering and Remote Sensing, NASA; Washington, DC, United States, 1977.

[54] Moon S. Kim, C. S. T. Daughtry, E. W. Chappelle, J. E. Mcmurtrey, and C. L. Walthall. The use of high spectral resolution bands for estimating absorbed photosynthetically active radiation (a par). Technical Report In CNES, Proceedings of 6th International Symposium on Physical Measurements and Signatures in Remote Sensing, NASA; Washington, DC, United States, 1994.

[55] P. Ghamisi, N. Yokoya, J. Li, W. Liao, S. Liu, J. Plaza, B. Rasti, and A. Plaza. Advances in hyperspectral image and signal processing: A comprehensive overview of the state of the art. *IEEE Geoscience and Remote Sensing Magazine*, 5(4):37–78, Dec 2017. ISSN 2168-6831. doi: 10.1109/MGRS.2017.2762087.

[56] Y. Zhong and L. Zhang. An adaptive artificial immune network for supervised classification of multi-/hyperspectral remote sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 50(3):894–909, March 2012. ISSN 0196-2892. doi: 10.1109/TGRS.2011.2162589.

[57] F. Melgani and L. Bruzzone. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Transactions on Geoscience and Remote Sensing*, 42(8):1778–1790, Aug 2004. ISSN 0196-2892. doi: 10.1109/TGRS.2004.831865.

[58] J. Li, J. M. Bioucas-Dias, and A. Plaza. Semisupervised hyperspectral image segmentation using multinomial logistic regression with active learning. *IEEE Transactions on Geoscience and Remote Sensing*, 48(11):4085–4098, Nov 2010. ISSN 0196-2892. doi: 10.1109/TGRS.2010.2060550.

[59] J. Li, J. M. Bioucas-Dias, and A. Plaza. Spectral–spatial hyperspectral image segmentation using subspace multinomial logistic regression and markov random fields. *IEEE Transactions on Geoscience and Remote Sensing*, 50(3):809–823, March 2012. ISSN 0196-2892. doi: 10.1109/TGRS.2011.2162649.

[60] B. Du and L. Zhang. Random-selection-based anomaly detector for hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 49(5):1578–1589, May 2011. ISSN 0196-2892. doi: 10.1109/TGRS.2010.2081677.

[61] G. Licciardi, P. R. Marpu, J. Chanussot, and J. A. Benediktsson. Linear versus nonlinear pca for the classification of hyperspectral data based on the extended morphological profiles. *IEEE Geoscience and Remote Sensing Letters*, 9(3):447–451, May 2012. ISSN 1545-598X. doi: 10.1109/LGRS.2011.2172185.

[62] S. Prasad and L. M. Bruce. Limitations of principal components analysis for hyperspectral target recognition. *IEEE Geoscience and Remote Sensing Letters*, 5(4):625–629, Oct 2008. ISSN 1545-598X. doi: 10.1109/LGRS.2008.2001282.

[63] A. Villa, J. A. Benediktsson, J. Chanussot, and C. Jutten. Hyperspectral image classification with independent component discriminant analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 49(12):4865–4876, Dec 2011. ISSN 0196-2892. doi: 10.1109/TGRS.2011.2153861.

[64] L. He, J. Li, C. Liu, and S. Li. Recent advances on spectral–spatial hyperspectral image classification: An overview and new guidelines. *IEEE Transactions on Geoscience and Remote Sensing*, 56(3):1579–1597, March 2018. ISSN 0196-2892. doi: 10.1109/TGRS.2017.2765364.

[65] P. Ghamisi, E. Maggiori, S. Li, R. Souza, Y. Tarablaka, G. Moser, A. De Giorgi, L. Fang, Y. Chen, M. Chi, S. B. Serpico, and J. A. Benediktsson. New frontiers in spectral-spatial hyperspectral image classification: The latest advances based on mathematical morphology, markov random fields, segmentation, sparse representation, and deep learning. *IEEE Geoscience and Remote Sensing Magazine*, 6(3):10–43, Sep. 2018. ISSN 2168-6831. doi: 10.1109/MGRS.2018.2854840.

[66] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. A. Benediktsson. Deep learning for hyperspectral image classification: An overview. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–20, 2019. ISSN 0196-2892. doi: 10.1109/TGRS.2019.2907932.

[67] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot. Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 5(2):354–379, April 2012. ISSN 1939-1404. doi: 10.1109/JSTARS.2012.2194696.

[68] N. Dobigeon, Y. Altmann, N. Brun, and S. Moussaoui. Chapter 6 - linear and nonlinear unmixing in hyperspectral imaging. In Cyril Ruckebusch, editor, *Resolving Spectral Mixtures*, volume 30 of *Data Handling in Science and Technology*, pages 185 – 224. Elsevier, 2016. doi: https://doi.org/10.1016/B978-0-444-63638-6.00006-1. URL `http://www.sciencedirect.com/science/article/pii/B9780444636386000061`.

[69] Chein-I Chang and Qian Du. Estimation of number of spectrally distinct signal sources in hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 42(3):608–619, March 2004. doi: 10.1109/TGRS.2003.819189.

[70] J. M. Bioucas-Dias and J. M. P. Nascimento. Hyperspectral subspace identification. *IEEE Transactions on Geoscience and Remote Sensing*, 46(8):2435–2445, Aug 2008. doi: 10.1109/TGRS.2008.918089.

[71] J. M. P. Nascimento and J. M. B. Dias. Vertex component analysis: a fast algorithm to unmix hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 43(4):898–910, April 2005. doi: 10.1109/TGRS.2005.844293.

[72] Michael E. Winter. N-findr: an algorithm for fast autonomous spectral end-member determination in hyperspectral data, 1999. URL `https://doi.org/10.1117/12.366289`.

[73] D. C. Heinz and Chein-I-Chang. Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 39(3):529–545, March 2001. doi: 10.1109/36.911111.

[74] Raul Guerra, Lucana Santos, Sebastian Lopez, and Roberto Sarmiento. A new fast algorithm for linearly unmixing hyperspectral images. *Geoscience and Remote Sensing, IEEE Transactions on*, 53(12):6752–6765, 2015.

[75] Ernestina Martel, Raúl Guerra, Sebastián López, and Roberto Sarmiento. A gpu-based processing chain for linearly unmixing hyperspectral images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2016.

[76] Alexander F.H. Goetz, Gregg Vane, Jerry E. Solomon, and Barrett N. Rock. Imaging spectrometry for earth remote sensing. *Science*, 228(4704):1147–1153, 1985. ISSN 0036-8075. doi: 10.1126/science.228.4704.1147. URL `https://science.sciencemag.org/content/228/4704/1147`.

[77] Alexander F.H. Goetz. Three decades of hyperspectral remote sensing of the earth: A personal view. *Remote Sensing of Environment*, 113:S5 – S16, 2009. ISSN 0034-4257. doi: https://doi.org/10.1016/j.rse.2007.12.014. URL `http://www.sciencedirect.com/science/article/pii/S003442570900073X`. Imaging Spectroscopy Special Issue.

[78] Glenn D. Boreman R. Glenn Sellar. Classification of imaging spectrometers for remote sensing applications. *Optical Engineering*, 44(1):1 – 3 – 3, 2005. doi: 10.1117/1.1813441. URL `https://doi.org/10.1117/1.1813441`.

[79] Helge Aasen, Eija Honkavaara, Arko Lucieer, and Pablo J. Zarco-Tejada. Quantitative remote sensing at ultra-high resolution with uav spectroscopy: A review of sensor technology, measurement procedures, and data correction workflows. *Remote Sensing*, 10(7), 2018. ISSN 2072-4292. doi: 10.3390/rs10071091. URL `https://www.mdpi.com/2072-4292/10/7/1091`.

[80] Peter A. Mitchell. Hyperspectral digital imagery collection experiment (hydice), 1995. URL `https://doi.org/10.1117/12.226807`.

[81] Yuheng Chen Xinhua Chen Weimin Shen Jiankang Zhou, Yiqun Ji. Design and laboratory calibration of the compact pushbroom hyperspectral imaging system, 2009. URL `https://doi.org/10.1117/12.838159`.

[82] Parrot. Sequoia+ camera, 2019. URL https://www.parrot.com/soluciones-business/profesional/parrot-sequoia. [Online; accessed 06-Oct-2019].

[83] Micasense. Rededge-MX camera, 2019. URL https://www.micasense.com/rededge-mx. [Online; accessed 06-Oct-2019].

[84] MAIA the multispectral camera. MAIA WV model, 2019. URL https://www.spectralcam.com/maia-tech/. [Online; accessed 06-Oct-2019].

[85] Eija Honkavaara, Roman Arbiol, Lauri Markelin, Lucas Martinez, Michael Cramer, Stéphane Bovet, Laure Chandelier, Risto Ilves, Sascha Klonus, Paul Marshal, Daniel Schläpfer, Mark Tabor, Christian Thom, and Nikolaj Veje. Digital airborne photogrammetry—a new tool for quantitative remote sensing?—a state-of-the-art review on radiometric aspects of digital photogrammetric images. *Remote Sensing*, 1(3):577–605, 2009. ISSN 2072-4292. doi: 10.3390/rs1030577. URL https://www.mdpi.com/2072-4292/1/3/577.

[86] SENOP Optronics Hyperspectral. Hyperspetcral Imaging and OEM Modules, 2019. URL https://senop.fi/en/optronics-hyperspectral. [Online; accessed 06-Oct-2019].

[87] Heikki Saari Kai Ojala Tapani Antila Jussi Mäkynen, Christer Holmlund. Unmanned aerial vehicle (uav) operated megapixel spectral camera, 2011. URL https://doi.org/10.1117/12.897712.

[88] Pixelteq. SpectroCam Multispectral Wheel Cameras, 2019. URL https://pixelteq.com/spectrocam/. [Online; accessed 06-Oct-2019].

[89] R. Graser A. Jung, R. Michels. Hyperspectral camera with spatial and spectral resolution and method, 2015.

[90] Cubert Hyperspectral VNIR Cameras. Firefleye 185 model, 2019. URL https://cubert-gmbh.com/product/firefleye-185-hyperspectral/. [Online; accessed 06-Oct-2019].

[91] Helge Aasen, Andreas Burkart, Andreas Bolten, and Georg Bareth. Generating 3d hyperspectral information with lightweight uav snapshot cameras for vegetation monitoring: From camera calibration to quality assurance. *ISPRS Journal of Photogrammetry and Remote Sensing*, 108:245 – 259, 2015. ISSN 0924-2716. doi: https://doi.org/10.1016/j.isprsjprs.2015.08.002. URL `http://www.sciencedirect.com/science/article/pii/S0924271615001938`.

[92] A. Lambrechts, P. Gonzalez, B. Geelen, P. Soussan, K. Tack, and M. Jayapala. A cmos-compatible, integrated approach to hyper- and multispectral imaging. In *2014 IEEE International Electron Devices Meeting*, pages 10.5.1–10.5.4, Dec 2014. doi: 10.1109/IEDM.2014.7047025.

[93] IMEC Hyperspectral Imaging. Sensors, Cameras and Software , 2019. URL `https://www.imec-int.com/en/hyperspectral-imaging`. [Online; accessed 06-Oct-2019].

[94] N. Muchiri and S. Kimathi. A review of applications and potential applications of uav. *Proceedings of Sustainable Research and Innovation Conference*, 0(0):280–283, 2016.

[95] Esther Salamí, Cristina Barrado, and Enric Pastor. Uav flight experiments applied to the remote sensing of vegetated areas. *Remote Sensing*, 6(11):11051–11081, 2014. ISSN 2072-4292. doi: 10.3390/rs61111051. URL `https://www.mdpi.com/2072-4292/6/11/11051`.

[96] Alessandro Matese, Piero Toscano, Salvatore Filippo Di Gennaro, Lorenzo Genesio, Francesco Primo Vaccari, Jacopo Primicerio, Claudio Belli, Alessandro Zaldei, Roberto Bianconi, and Beniamino Gioli. Intercomparison of uav, aircraft and satellite remote sensing platforms for precision viticulture. *Remote Sensing*, 7 (3):2971–2990, 2015. ISSN 2072-4292. doi: 10.3390/rs70302971. URL `https://www.mdpi.com/2072-4292/7/3/2971`.

[97] Gonzalo Pajares. Overview and current status of remote sensing applications based on unmanned aerial vehicles (uavs). *Photogrammetric Engineering & Remote*

Sensing, 81(4):281 – 329, 2015. ISSN 0099-1112. doi: https://doi.org/10.14358/PERS.81.4.281. URL http://www.sciencedirect.com/science/article/pii/S0099111215300793.

[98] Huang Yao, Rongjun Qin, and Xiaoyu Chen. Unmanned aerial vehicle for remote sensing applications—a review. *Remote Sensing*, 11(12), 2019. ISSN 2072-4292. doi: 10.3390/rs11121443. URL https://www.mdpi.com/2072-4292/11/12/1443.

[99] Luís Pádua, Jakub Vanko, Jonáš Hruška, Telmo Adão, Joaquim J. Sousa, Emanuel Peres, and Raul Morais. Uas, sensors, and data processing in agroforestry: a review towards practical applications. *International Journal of Remote Sensing*, 38(8-10): 2349–2391, 2017. doi: 10.1080/01431161.2017.1297548. URL https://doi.org/10.1080/01431161.2017.1297548.

[100] Y. Zhong, X. Wang, Y. Xu, S. Wang, T. Jia, X. Hu, J. Zhao, L. Wei, and L. Zhang. Mini-uav-borne hyperspectral remote sensing: From observation and processing to applications. *IEEE Geoscience and Remote Sensing Magazine*, 6(4):46–62, Dec 2018. ISSN 2168-6831. doi: 10.1109/MGRS.2018.2867592.

[101] Lei Deng, Zhihui Mao, Xiaojuan Li, Zhuowei Hu, Fuzhou Duan, and Yanan Yan. Uav-based multispectral remote sensing for precision agriculture: A comparison between different cameras. *ISPRS Journal of Photogrammetry and Remote Sensing*, 146:124 – 136, 2018. ISSN 0924-2716. doi: https://doi.org/10.1016/j.isprsjprs.2018.09.008. URL http://www.sciencedirect.com/science/article/pii/S0924271618302533.

[102] R. Calderón, M. Montes-Borrego, B. B. Landa, J. A. Navas-Cortés, and P. J. Zarco-Tejada. Detection of downy mildew of opium poppy using high-resolution multi-spectral and thermal imagery acquired with an unmanned aerial vehicle. *Precision Agriculture*, 15(6):639–661, Dec 2014. ISSN 1573-1618. doi: 10.1007/s11119-014-9360-y. URL https://doi.org/10.1007/s11119-014-9360-y.

[103] S. Esposito, M. Mura, P. Fallavollita, M. Balsi, G. Chirici, A. Oradini, and M. Marchetti. Performance evaluation of lightweight lidar for uav applications.

In *2014 IEEE Geoscience and Remote Sensing Symposium*, pages 792–795, July 2014. doi: 10.1109/IGARSS.2014.6946543.

[104] Y. Lin, J. Hyyppa, and A. Jaakkola. Mini-uav-borne lidar for fine-scale mapping. *IEEE Geoscience and Remote Sensing Letters*, 8(3):426–430, May 2011. ISSN 1545-598X. doi: 10.1109/LGRS.2010.2079913.

[105] Luke Wallace, Arko Lucieer, Christopher Watson, and Darren Turner. Development of a uav-lidar system with application to forest inventory. *Remote Sensing*, 4(6): 1519–1543, 2012. ISSN 2072-4292. doi: 10.3390/rs4061519. URL `https://www.mdpi.com/2072-4292/4/6/1519`.

[106] Tetsuro Ishida, Junichi Kurihara, Fra Angelico Viray, Shielo Baes Namuco, Enrico C. Paringit, Gay Jane Perez, Yukihiro Takahashi, and Joel Joseph Marciano. A novel approach for vegetation classification using uav-based hyperspectral imaging. *Computers and Electronics in Agriculture*, 144:80 – 85, 2018. ISSN 0168-1699. doi: https://doi.org/10.1016/j.compag.2017.11.027. URL `http://www.sciencedirect.com/science/article/pii/S0168169917310499`.

[107] A. Burkart, S. Cogliati, A. Schickling, and U. Rascher. A novel uav-based ultra-light weight spectrometer for field spectroscopy. *IEEE Sensors Journal*, 14(1):62–67, Jan 2014. ISSN 1530-437X. doi: 10.1109/JSEN.2013.2279720.

[108] Sara Freitas, Hugo Silva, José Almeida, and Eduardo Silva. Hyperspectral imaging for real-time unmanned aerial vehicle maritime target detection. *Journal of Intelligent & Robotic Systems*, 90(3):551–570, Jun 2018. ISSN 1573-0409. doi: 10.1007/s10846-017-0689-0. URL `https://doi.org/10.1007/s10846-017-0689-0`.

[109] J. Pablo Arroyo-Mora, Margaret Kalacska, Deep Inamdar, Raymond Soffer, Oliver Lucanus, Janine Gorman, Tomas Naprstek, Erica Skye Schaaf, Gabriela Ifimov, Kathryn Elmer, and George Leblanc. Implementation of a uav–hyperspectral pushbroom imager for ecological monitoring. *Drones*, 3(1), 2019. ISSN 2504-446X. doi: 10.3390/drones3010012. URL `https://www.mdpi.com/2504-446X/3/1/12`.

[110] M. Teke, H. S. Deveci, O. Haliloğlu, S. Z. Gürbüz, and U. Sakarya. A short survey of hyperspectral remote sensing applications in agriculture. In *2013 6th International Conference on Recent Advances in Space Technologies (RAST)*, pages 171–176, June 2013. doi: 10.1109/RAST.2013.6581194.

[111] Lorenz Meier. Pixhawk Flight Controller, 2008. URL `http://pixhawk.org/`. [Online; accessed 06-Oct-2019].

[112] Chris Anderson & Jordi Munoz, 3D Robotics. Ardupilot copter site, 2009. URL `http://ardupilot.org/copter/index.html`. [Online; accessed 06-Oct-2019].

[113] Dji Products. Dji RTK GNSS System, 2019. URL `https://www.dji.com/d-rtk/info`. [Online; accessed 06-Oct-2019].

[114] Dji Products. Dji Gimbal Ronin-Mx, 2019. URL `https://www.dji.com/ronin-mx/info`. [Online; accessed 06-Oct-2019].

[115] Drok. Drocking LM2596 converter item, 2019. URL `https://www.droking.com/`. [Online; accessed 06-Oct-2019].

[116] Dji Developer site. Dji Mobile Software Development Kit, 2019. URL `https://developer.dji.com/mobile-sdk/`. [Online; accessed 06-Oct-2019].

[117] Dji Developer site. Dji Onboard Software Development Kit, 2019. URL `https://developer.dji.com/onboard-sdk/`. [Online; accessed 06-Oct-2019].

[118] Pleora Technologies. eBUS Software Development Kit, 2019. URL `https://www.pleora.com/products/ebus-sdk/`. [Online; accessed 06-Oct-2019].

[119] IDS Imaging Development Systems GmbH. IDS uEye SDK, 2019. URL `https://en.ids-imaging.com/ueye-software-archive.html`. [Online; accessed 06-Oct-2019].

[120] Dji Products. Dji Lightbridge, 2019. URL `https://www.dji.com/es/dji-lightbridge`. [Online; accessed 06-Oct-2019].

[121] GIS Geography. World Geodesic System (WGS84), 2019. URL https://gisgeography.com/wgs84-world-geodetic-system/. [Online; accessed 06-Oct-2019].

[122] John Hunter, Darren Dale, Eric Firing, Michael Droettboom and the Matplotlib development team. MatplotLib Python plotting library version 3.1.1, 2019. URL https://matplotlib.org/3.1.1/index.html. [Online; accessed 06-Oct-2019].

[123] SphereOptics. Zenith Polymer Diffusers, 2019. URL https://sphereoptics.de/en/product/zenith-polymer-diffusers/?c=79. [Online; accessed 06-Oct-2019].

[124] Christian Rogass, Christian Mielke, Daniel Scheffler, Nina K. Boesche, Angela Lausch, Christin Lubitz, Maximilian Brell, Daniel Spengler, Andreas Eisele, Karl Segl, and Luis Guanter. Reduction of uncorrelated striping noise—applications for hyperspectral pushbroom acquisitions. *Remote Sensing*, 6(11):11082–11106, 2014. ISSN 2072-4292. doi: 10.3390/rs61111082. URL http://www.mdpi.com/2072-4292/6/11/11082.

[125] Newport Corporation. Oriel Instruments Light Sources, 2019. URL https://www.newport.com/b/oriel-instruments. [Online; accessed 06-Oct-2019].

[126] Y. Chen, Y. Ji, J. Zhou, X. Chen, and W. Shen. Computation of signal-to-noise ratio of airborne hyperspectral imaging spectrometer. In *2012 International Conference on Systems and Informatics (ICSAI2012)*, pages 1046–1049, May 2012. doi: 10.1109/ICSAI.2012.6223191.

[127] Dong Xu, Lei Sun, and Jianshu Luo. Noise estimation of hyperspectral remote sensing image based on multiple linear regression and wavelet transform. *Boletim de Ciências Geodésicas*, 19:639 – 652, 12 2013. ISSN 1982-2170. URL http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1982-21702013000400008&nrm=iso.

[128] Marion Jaud, Nicolas Le Dantec, Jérôme Ammann, Philippe Grandjean, Dragos Constantin, Yosef Akhtman, Kevin Barbieux, Pascal Allemand, Christophe

Delacourt, and Bertrand Merminod. Direct georeferencing of a pushbroom, lightweight hyperspectral system for mini-uav applications. *Remote Sensing*, 10 (2), 2018. ISSN 2072-4292. doi: 10.3390/rs10020204. URL https://www.mdpi.com/2072-4292/10/2/204.

[129] Agisoft LLC. Photoscan Software Package, 2019. URL https://www.agisoft.com/. [Online; accessed 06-Oct-2019].

[130] Pau Gargallo. OpenSfM, 2017. URL https://www.opensfm.org/. [Online; accessed 06-Oct-2019].

[131] Avigdor Margalit, IS Reed, and RM Gagliardi. Adaptive optical target detection using correlated images. *IEEE Transactions on Aerospace and Electronic Systems*, AES-21(3):394–405, 1985.

[132] Liaoying Zhao, Weijun Lin, Yulei Wang, and Xiaorun Li. Recursive local summation of rx detection for hyperspectral image using sliding windows. *Remote Sensing*, 10 (1), 2018. ISSN 2072-4292. doi: 10.3390/rs10010103. URL https://www.mdpi.com/2072-4292/10/1/103.

[133] Heesung Kwon and N. M. Nasrabadi. Kernel rx-algorithm: a nonlinear anomaly detector for hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 43(2):388–397, Feb 2005. doi: 10.1109/TGRS.2004.841487.

[134] Chunhui Zhao, Jiawei Li, Meiling Meng, and Xifeng Yao. A weighted spatial-spectral kernel rx algorithm and efficient implementation on gpus. *Sensors*, 17(3), 2017. ISSN 1424-8220. doi: 10.3390/s17030441. URL https://www.mdpi.com/1424-8220/17/3/441.

[135] L. Zhu, G. Wen, S. Qiu, and X. Zhang. Improving hyperspectral anomaly detection with a simple weighting strategy. *IEEE Geoscience and Remote Sensing Letters*, 16 (1):95–99, Jan 2019. ISSN 1545-598X. doi: 10.1109/LGRS.2018.2869337.

[136] M. Vafadar and H. Ghassemian. Hyperspectral anomaly detection using combined similarity criteria. *IEEE Journal of Selected Topics in Applied Earth Observations*

and Remote Sensing, 11(11):4076–4085, Nov 2018. ISSN 1939-1404. doi: 10.1109/JSTARS.2018.2870123.

[137] H. Su, Z. Wu, Q. Du, and P. Du. Hyperspectral anomaly detection using collaborative representation with outlier removal. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(12):5029–5038, Dec 2018. ISSN 1939-1404. doi: 10.1109/JSTARS.2018.2880749.

[138] N. Ma, Y. Peng, and S. Wang. A fast recursive collaboration representation anomaly detector for hyperspectral image. *IEEE Geoscience and Remote Sensing Letters*, 16 (4):588–592, April 2019. ISSN 1545-598X. doi: 10.1109/LGRS.2018.2878869.

[139] N. Huyan, X. Zhang, H. Zhou, and L. Jiao. Hyperspectral anomaly detection via background and potential anomaly dictionaries construction. *IEEE Transactions on Geoscience and Remote Sensing*, 57(4):2263–2276, April 2019. ISSN 0196-2892. doi: 10.1109/TGRS.2018.2872590.

[140] Y. Xu, Z. Wu, J. Chanussot, and Z. Wei. Joint reconstruction and anomaly detection from compressive hyperspectral images using mahalanobis distance-regularized tensor rpca. *IEEE Transactions on Geoscience and Remote Sensing*, 56(5):2919–2930, May 2018. ISSN 0196-2892. doi: 10.1109/TGRS.2017.2786718.

[141] Chunhui Zhao, Weiwei Deng, Yiming Yan, and Xifeng Yao. Progressive line processing of kernel rx anomaly detection algorithm for hyperspectral imagery. *Sensors*, 17(8), 2017. URL `http://www.mdpi.com/1424-8220/17/8/1815`.

[142] Cornelius Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *J. Res. Natl. Bur. Stand. B*, 45:255–282, 1950. doi: 10.6028/jres.045.026.

[143] R. B. Lehoucq and D. C. Sorensen. Deflation techniques for an implicitly restarted arnoldi iteration. *SIAM Journal on Matrix Analysis and Applications*, 17(4):789–821, 1996. doi: 10.1137/S0895479895281484.

[144] MIT education website. Power Method tutorial, MATLAB Central File Exchange, 2007. URL http://web.mit.edu/18.06/www/Spring17/Power-Method.pdf. [Online; accessed 06-Oct-2019].

[145] USGS Spectroscopy Lab. Digital Spectral Library, 2019. URL http://speclab.cr.usgs.gov/spectral-lib.html. [Online; accessed 06-Oct-2019].

[146] Grupo de Inteligencia Computacional, Universidad del País Vasco/Euskal Herriko Unibertsitatea(UPV/EHU), Spain. Hyperspectral imagery synthesis toolbox, 2009. URL http://www.ehu.es/ccwintco/index.php/Hyperspectral_Imagery_Synthesis_tools_for_MATLAB. [Online; accessed 06-Oct-2019].

[147] Jared A Herweg, John P Kerekes, Oliver Weatherbee, David Messinger, Jan van Aardt, Emmett Ientilucci, Zoran Ninkov, Jason Faulring, Nina Raqueño, and Joseph Meola. Spectir hyperspectral airborne rochester experiment data collection campaign. In *SPIE Defense, Security, and Sensing*, pages 839028–839028. International Society for Optics and Photonics, 2012.

[148] Antonio Plaza, Qian Du, Yang-Lang Chang, and Roger L King. High performance computing for hyperspectral remote sensing. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 4(3):528–544, 2011.

[149] M. Diaz, R. Guerra, S. Lopez, and R. Sarmiento. An algorithm for an accurate detection of anomalies in hyperspectral images with a low computational complexity. *IEEE Transactions on Geoscience and Remote Sensing*, 56(2):1159–1176, Feb 2018. ISSN 0196-2892. doi: 10.1109/TGRS.2017.2761019.

[150] M. Díaz, R. Guerra, P. Horstrand, S. López, and R. Sarmiento. A line-by-line fast anomaly detector for hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–15, 2019. ISSN 0196-2892. doi: 10.1109/TGRS.2019.2923921.

[151] M. Díaz, R. Guerra, P. Horstrand, E. Martel, S. López, J. F. López, and R. Sarmiento. Real-time hyperspectral image compression onto embedded gpus.

IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, pages 1–18, 2019. doi: 10.1109/JSTARS.2019.2917088.

[152] Caglar Uyulan, Metin Gokasan, and Seta Bogosyan. Modeling, simulation and slip control of a railway vehicle integrated with traction power supply. *Cogent Engineering*, 4(1):1312680, 2017. doi: 10.1080/23311916.2017.1312680. URL `https://www.tandfonline.com/doi/abs/10.1080/23311916.2017.1312680`.

[153] Hazim Sadideen, Karim Hamaoui, Munir Saadeddin, and Roger Kneebone. Simulators and the simulation environment: Getting the balance right in simulation-based surgical education. *International Journal of Surgery*, 10(9):458 – 462, 2012. ISSN 1743-9191. doi: https://doi.org/10.1016/j.ijsu.2012.08.010. URL `http://www.sciencedirect.com/science/article/pii/S1743919112007558`.

[154] Redmond Ramin Shamshiri, Ibrahim A. Hameed, Lenka Pitonakova, Cornelia Weltzien, Siva K. Balasundram, Ian J. Yule, Tony E Grift, and Girish Chowdhary. Simulation software and virtual environments for acceleration of agricultural robotics: Features highlights and performance comparison. *International Journal of Agricultural and Biological Engineering*, 11(4):15–31, 1 2018. ISSN 1934-6344. doi: 10.25165/ijabe.v11i4.4032.

[155] M. A. F. Wagner, S. Sevinc, O. Tanir, P. L. Haigh, J. D. Arthur, R. E. Nance, and H. D. Schwetman. A standard simulation environment: a review of preliminary requirements. In *Proceedings of Winter Simulation Conference*, pages 664–672, Dec 1994. doi: 10.1109/WSC.1994.717406.

[156] Haiwen Yuan, Changshi Xiao, Wenqiang Zhan, Yanfeng Wang, Chen Shi, Huixian Ye, Kai Jiang, Zhenyi Ye, Chunhui Zhou, Yuanqiao Wen, and Qiliang Li. Target detection, positioning and tracking using new uav gas sensor systems: Simulation and analysis. *Journal of Intelligent & Robotic Systems*, 94(3):871–882, Jun 2019. ISSN 1573-0409. doi: 10.1007/s10846-018-0909-2. URL `https://doi.org/10.1007/s10846-018-0909-2`.

[157] Ankit A. Ravankar, Abhijeet Ravankar, Yukinori Kobayashi, and Takanori Emaru. Autonomous mapping and exploration with unmanned aerial vehicles using low cost

sensors. *Proceedings*, 4(1), 2019. ISSN 2504-3900. doi: 10.3390/ecsa-5-05753. URL https://www.mdpi.com/2504-3900/4/1/44.

[158] Creanex Oy. Simulator Solutions, 2019. URL https://creanex.fi/en/. [Online; accessed 06-Oct-2019].

[159] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot. Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 5(2):354–379, April 2012. ISSN 1939-1404. doi: 10.1109/JSTARS.2012.2194696.

[160] Boris Kozintsev. *Computations with Gaussian Random Fields*. PhD thesis, University of Maryland, 1999.

[161] Driss Haboudane, John R Miller, Elizabeth Pattey, Pablo J Zarco-Tejada, and Ian B Strachan. Hyperspectral vegetation indices and novel algorithms for predicting green lai of crop canopies: Modeling and validation in the context of precision agriculture. *Remote Sensing of Environment*, 90(3):337 – 352, 2004. ISSN 0034-4257. doi: https://doi.org/10.1016/j.rse.2003.12.013. URL http://www.sciencedirect.com/science/article/pii/S0034425704000264.

[162] Wouter H. Maes and Kathy Steppe. Perspectives for remote sensing with unmanned aerial vehicles in precision agriculture. *Trends in Plant Science*, 24(2):152 – 164, 2019. ISSN 1360-1385. doi: https://doi.org/10.1016/j.tplants.2018.11.007. URL http://www.sciencedirect.com/science/article/pii/S1360138518302693.

[163] UM Rao Mogili and B B V L Deepak. Review on application of drone systems in precision agriculture. *Procedia Computer Science*, 133:502 – 509, 2018. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2018.07.063. URL http://www.sciencedirect.com/science/article/pii/S1877050918310081. International Conference on Robotics and Smart Manufacturing (RoSMa2018).

[164] Chunhua Zhang and John M. Kovacs. The application of small unmanned aerial systems for precision agriculture: a review. *Precision Agriculture*, 13(6):693–712,

Dec 2012. ISSN 1573-1618. doi: 10.1007/s11119-012-9274-5. URL https://doi.org/10.1007/s11119-012-9274-5.

[165] Simon Measho, Baozhang Chen, Yongyut Trisurat, Petri Pellikka, Lifeng Guo, Sunsanee Arunyawat, Venus Tuankrua, Woldeselassie Ogbazghi, and Tecle Yemane. Spatio-temporal analysis of vegetation dynamics as a response to climate variability and drought patterns in the semiarid region, eritrea. *Remote Sensing*, 11(6), 2019. ISSN 2072-4292. doi: 10.3390/rs11060724. URL https://www.mdpi.com/2072-4292/11/6/724.

[166] Jimmy O. Adegoke and Andrew M. Carleton. Relations between soil moisture and satellite vegetation indices in the u.s. corn belt. *Journal of Hydrometeorology*, 3(4): 395–405, 2002. ISSN 1525755X, 15257541. URL http://www.jstor.org/stable/24909189.

[167] Rajen Bajgain, Xiangming Xiao, Pradeep Wagle, Jeffrey Basara, and Yuting Zhou. Sensitivity analysis of vegetation indices to drought over two tallgrass prairie sites. *ISPRS Journal of Photogrammetry and Remote Sensing*, 108:151 – 160, 2015. ISSN 0924-2716. doi: https://doi.org/10.1016/j.isprsjprs.2015.07.004. URL http://www.sciencedirect.com/science/article/pii/S0924271615001719.

[168] Tien Le Thuy Du, Duong Du Bui, Minh Duc Nguyen, and Hyongki Lee. Satellite-based, multi-indices for evaluation of agricultural droughts in a highly dynamic tropical catchment, central vietnam. *Water*, 10(5), 2018. ISSN 2073-4441. doi: 10.3390/w10050659. URL https://www.mdpi.com/2073-4441/10/5/659.

[169] M.P. González-Dugo and L. Mateos. Spectral vegetation indices for benchmarking water productivity of irrigated cotton and sugarbeet crops. *Agricultural Water Management*, 95(1):48 – 58, 2008. ISSN 0378-3774. doi: https://doi.org/10.1016/j.agwat.2007.09.001. URL http://www.sciencedirect.com/science/article/pii/S037837740700220X.

[170] Bruno Basso, Davide Cammarano, and Pasquale De Vita. Remotely sensed vegetation indices: Theory and applications for crop management. 2004.

[171] Jinru Xue and Baofeng Su. Significant remote sensing vegetation indices: A review of developments and applications. *Journal of Sensors*, (1353691):17, 2017. doi: 10.1155/2017/1353691. URL https://doi.org/10.1155/2017/1353691.

[172] Verena Henrich & Katharina Brüser, Institute of Crop Science & Resource Conservation (INRES). Vegetation Index Database, 2019. URL https://www.indexdatabase.de/. [Online; accessed 06-Oct-2019].

[173] Sebastian Pop, Luciana Cristea, Marius Cristian Luculescu, Sorin Constantin Zamfira, and Attila Laszlo Boer. Vegetation index estimation in precision farming using custom multispectral camera mounted on unmanned aerial vehicle. In Michael E. Auer and Kalyan Ram B., editors, *Cyber-physical Systems and Digital Twins*, pages 674–685, Cham, 2020. Springer International Publishing. ISBN 978-3-030-23162-0.

[174] John W. Rouse, Robert H. Haas, John A. Schell, and Donald W. Deering. Monitoring the vernal advancements and retrogradation of natural vegetation. Technical Report E75-10354, NASA-CR-144661, RSC-1978-4, NASA; Washington, DC, United States, 1974.

[175] Jing M. Chen. Evaluation of vegetation indices and a modified simple ratio for boreal applications. *Canadian Journal of Remote Sensing*, 22(3):229–242, 1996. doi: 10.1080/07038992.1996.10855178. URL https://doi.org/10.1080/07038992.1996.10855178.

[176] F. Baret, S. Jacquemoud, and J. F. Hanocq. The soil line concept in remote sensing. *Remote Sensing Reviews*, 7(1):65–82, 1993. doi: 10.1080/02757259309532166. URL https://doi.org/10.1080/02757259309532166.

[177] C.S.T Daughtry, C.L Walthall, M.S Kim, E.Brown de Colstoun, and J.E McMurtrey. Estimating corn leaf chlorophyll concentration from leaf and canopy reflectance. *Remote Sensing of Environment*, 74(2):229 – 239, 2000. ISSN 0034-4257. doi: https://doi.org/10.1016/S0034-4257(00)00113-9. URL http://www.sciencedirect.com/science/article/pii/S0034425700001139.

[178] M Möller, V Alchanatis, Y Cohen, M Meron, J Tsipris, A Naor, V Ostrovsky, M Sprintsin, and S Cohen. Use of thermal and visible imagery for estimating crop water status of irrigated grapevine*. *Journal of Experimental Botany*, 58(4):827–838, 09 2006. ISSN 0022-0957. doi: 10.1093/jxb/erl115. URL `https://doi.org/10.1093/jxb/erl115`.

[179] Carlos Ballester, James Brinkhoff, Wendy C. Quayle, and John Hornbuckle. Monitoring the effects of water stress in cotton using the green red vegetation index and red edge ratio. *Remote Sensing*, 11(7), 2019. ISSN 2072-4292. doi: 10.3390/rs11070873. URL `https://www.mdpi.com/2072-4292/11/7/873`.

[180] Elke Bauriegel and Werner B. Herppich. Hyperspectral and chlorophyll fluorescence imaging for early detection of plant diseases, with special reference to fusarium spec. infections on wheat. *Agriculture*, 4(1):32–57, 2014. ISSN 2077-0472. doi: 10.3390/agriculture4010032. URL `https://www.mdpi.com/2077-0472/4/1/32`.

[181] Anne-Katrin Mahlein, Elias Alisaac, Ali Al Masri, Jan Behmann, Heinz-Wilhelm Dehne, and Erich-Christian Oerke. Comparison and combination of thermal, fluorescence, and hyperspectral imaging for monitoring fusarium head blight of wheat on spikelet scale. *Sensors*, 19(10), 2019. ISSN 1424-8220. doi: 10.3390/s19102281. URL `https://www.mdpi.com/1424-8220/19/10/2281`.

[182] Jonathan S. West, Gail G. M. Canning, Sarah A. Perryman, and Kevin King. Novel technologies for the detection of fusarium head blight disease and airborne inoculum. *Tropical Plant Pathology*, 42(3):203–209, Jun 2017. ISSN 1983-2052. doi: 10.1007/s40858-017-0138-4. URL `https://doi.org/10.1007/s40858-017-0138-4`.

[183] Y. Wu, X. Yang, A. Plaza, F. Qiao, L. Gao, B. Zhang, and Y. Cui. Approximate computing of remotely sensed data: Svm hyperspectral image classification as a case study. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(12):5806–5818, Dec 2016. doi: 10.1109/JSTARS.2016.2539282.

[184] Shaojun Wang, Xinyu Niu, Ning Ma, Wayne Luk, Philip Leong, and Yu Peng. A scalable dataflow accelerator for real time onboard hyperspectral image

classification. In Vanderlei Bonato, Christos Bouganis, and Marek Gorgon, editors, *Applied Reconfigurable Computing*, pages 105–116, Cham, 2016. Springer International Publishing. ISBN 978-3-319-30481-6.

[185] J. López-Fandiño, P. Quesada-Barriuso, D. B. Heras, and F. Argüello. Efficient elm-based techniques for the classification of hyperspectral remote sensing images on commodity gpus. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(6):2884–2893, June 2015. doi: 10.1109/JSTARS.2014.2384133.

[186] Chih-Chung Chang and Chih-Jen Lin. Libsvm - A Library for Support Vector Machines, 2005. URL `https://www.csie.ntu.edu.tw/~cjlin/libsvm/`. [Online; accessed 06-Oct-2019].

[187] M. Khodadadzadeh, J. Li, A. Plaza, H. Ghassemian, J. M. Bioucas-Dias, and X. Li. Spectral–spatial classification of hyperspectral data using local and global probabilities for mixed pixel characterization. *IEEE Transactions on Geoscience and Remote Sensing*, 52(10):6298–6314, Oct 2014. doi: 10.1109/TGRS.2013.2296031.

[188] Péter Burai, Balázs Deák, Orsolya Valkó, and Tamás Tomor. Classification of herbaceous vegetation using airborne hyperspectral imagery. *Remote Sensing*, 7 (2):2046–2066, 2015. ISSN 2072-4292. doi: 10.3390/rs70202046. URL `https://www.mdpi.com/2072-4292/7/2/2046`.

[189] E. Ibarrola-Ulzurrun, L. Drumetz, J. Marcello, C. Gonzalo-Martín, and J. Chanussot. Hyperspectral classification through unmixing abundance maps addressing spectral variability. *IEEE Transactions on Geoscience and Remote Sensing*, 57(7):4775–4788, July 2019. doi: 10.1109/TGRS.2019.2892903.

[190] I. Dopido, A. Villa, A. Plaza, and P. Gamba. A quantitative and comparative assessment of unmixing-based feature extraction techniques for hyperspectral image classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 5(2):421–435, April 2012. doi: 10.1109/JSTARS.2011.2176721.

[191] S. Lopez, P. Horstrand, G. M. Callico, J. F. Lopez, and R. Sarmiento. A low-computational-complexity algorithm for hyperspectral endmember extraction: Modified vertex component analysis. *IEEE Geoscience and Remote Sensing Letters*, 9(3):502–506, May 2012. doi: 10.1109/LGRS.2011.2172771.

[192] Sebastian Lopez, Pablo Horstrand, Gustavo M Callico, Jose F Lopez, and Roberto Sarmiento. A novel architecture for hyperspectral endmember extraction by means of the modified vertex component analysis (MVCA) algorithm. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, 5(6):1837–1848, 2012.