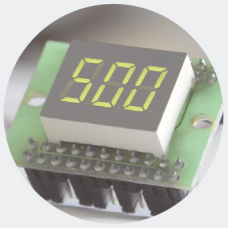
INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES






Hyperspectral Imaging for Real-Time Unmanned Aerial Vehicle Maritime Target Detection

SARA COSTA FREITAS
novembro de 2016

POLITÉCNICO
DO PORTO

Hyperspectral Imaging for Real-Time Unmanned Aerial Vehicle Maritime Target Detection

Master Thesis

To obtain the degree of master at the
Instituto Superior de Engenharia do Porto,
public defend on November 2016 by

Sara Costa Freitas
Nº 1110176

Supervisor: José Miguel Soares de Almeida

Co-Supervisor: Hugo Miguel Gomes da Silva

Degree in Electronics and Computer Science - Autonomous Systems
Porto, Portugal.

Aos meus pais, avós e irmã...

Abstract

The hyperspectral cameras use has been increasing over the past years, driven by the exponential growth of the computational systems power. The capability of acquiring multiple spectre wavelengths benefits the increase of the hyperspectral systems range of applications. However, until now, most hyperspectral systems are used in pos-processing and do not allow to take full advantage of the system capabilities.

There is a recent trend to be able to use hyperspectral systems in real-time.

Given the recent problems in European Union borders due to irregular immigration and drug smuggling, there is the need to develop novel autonomous surveillance systems that can work on these scenarios.

This thesis addresses the scenario of using hyperspectral imaging systems for maritime target detection using unmanned aerial vehicles. Specifically, by working in the creation of a hyperspectral real-time data processing system pipeline.

In our work, we develop a boresight calibration method that allows to calibrate the position of the navigation sensor related to the camera imaging sensor, and improve substantially the accuracy of the target geo-reference. We also develop a novel method of distinguish targets (boats) from their dominant background. With this application our system is able to only select relevant information to send to a remote station on the ground, thus making it suitable to be installed in an actual unmanned maritime surveillance system.

Keywords: hyperspectral, real-time, boresight calibration, geo-referencing, push-broom, target detection, unmanned aerial vehicle

Resumo

A utilização de câmaras hiperespectrais tem vindo a aumentar nos últimos anos, motivada pelo crescimento exponencial da capacidade de processamento dos mais recentes sistemas computacionais. A sua aptidão para observar múltiplos comprimentos de onda beneficia aplicações em diferentes campos de atividade. No entanto, a maior parte das aplicações com câmaras hiperespectrais são realizadas em pós-processamento, não aproveitando totalmente as capacidades destes sistemas.

Existe uma necessidade emergente de detetar mais caracteristicas sobre o cenário que está a ser observado, incentivando o desenvolvimento de sistemas hiperespectrais capazes de adquirir e processar informação em tempo-real.

Face aos mais recentes problemas de emigração e contrabando ilegal na União Europeia, surge a necessidade da realização de vigilancia autónoma capaz de adquirir o máximo de informação possível sobre os meios envolventes presentes num dado percurso. É neste contexto que se insere a dissertação que visa a criação e implementação de um sistema hiperespectral em tempo-real.

Para construir o sistema, foi necessário dividir o problema em diferentes etapas. Iniciou-se por um estudo detalhado dos sistemas hiperespectrais, desenvolvendo um método de calibração dos ângulos de boresight, que permitiu calibrar a relação entre o sistema de posicionamento e navegação da câmara hiperespectral e o sensor imagem. Esta calibração, permite numa fase posterior geo-referenciar os alvos com maior precisão.

Posteriormente, foi criada uma pipeline de processamento, que permite analisar os espectros obtidos, distinguindo os alvos do cenário onde estão inseridos.

Após a deteção dos alvos, procede-se à sua geo-referenciação, de forma a obter as coordenadas UTM do alvo. Toda a informação obtida sobre o alvo e a sua posição é enviada para uma estação em terra, de forma a ser validada por um humano. Para tal, foi também desenvolvida a metodologia de envio, para selecionar a informação a enviar apenas à mais relevante.

Palavras-Chave: hiperespectral, tempo-real, calibração boresight, geo-referenciação, pushbroom, deteção de alvos, veiculos aéreos não tripulados

Agradecimentos

Primeiro, gostaria de agradecer ao meu orientador, Eng. José Almeida, pelo desafio proporcionado, assim como toda ajuda disponibilizada ao longo deste projeto.

Queria agradecer ao Eng. Hugo Silva por toda a preocupação, apoio e acompanhamento constante ao longo de todo este percurso, assim como por todo o conhecimento que me conseguiu transmitir.

À familia do Laboratório de Sistemas Autónomos (LSA), com quem tanto tenho aprendido ao longo destes últimos dois anos.

Aos meus colegas de trabalho, Flávio Lopes, João Ribeiro, Guilherme Amaral e Carlos Almeida, por tudo o que me ensinaram, pelo incentivo e suporte incessante ao longo deste ciclo.

A todos os meus amigos e colegas de curso que me acompanharam no decorrer desta longa jornada, em especial à Sofia Teixeira.

Ao meu padrasto, Nelson Cunha, por todo o apoio e motivação ao longo desta etapa e das anteriores.

Aos meus avós, pelo amor, incentivo e interesse ao longo de toda a minha jornada académica.

À minha irmã, Catarina Cunha, pela motivação e carinho que me deu ao longo desta etapa.

Ao Carlos Ferrás, por todo o apoio, carinho, força e compreensão ao longo destes últimos cinco anos.

Aos meus pais, a quem irei estar eternamente grata por desde criança me desafiarem a querer saber sempre mais, e também pela motivação, apoio e carinho ao longo do meu percurso académico. Espero que a conclusão desta etapa possa, de alguma forma, retribuir todas as oportunidades que tanto se esforçaram para me dar.

Contents

List of Figures

List of Tables

Acronyms

AHRS	Attitude and Heading Reference System
AIS	Automatic Identification System
BIL	Band-interleaved-by-line
BIP	Band-interleaved-by-pixel
BSQ	Band Sequential
CCD	Charge-Coupled Device
CMOS	Complementary Metal-Oxide Semiconductor
CPU	Central Processing Unit
DEM	Digital Elevation Model
DOF	Degrees Of Freedom
ENVI	ENvironment for Visualizing Images
EO	Electro-Optic
FPA	Focal Plane Array
FPS	Frames Per Second
GigE	Gigabit Ethernet
GPS	Global Positioning System
GPU	Graphics Processing Unit
HSDSP	HyperSpectral Data Streamer Protocol
IMU	Inertial Measurement Unit
IR	InfraRed
LSA	Autonomous Systems Laboratory
NED	North-East-Down
RGB	Red Green Blue
ROS	Robot Operating System
SNR	Signal-to-Noise Ratio

SUNNY Smart UNattended airborne sensor Network for detection of vessels used for cross border crime and irregular entrY

SWIR Short-wave infrared

UAV Unmanned Aerial Vehicle

VNIR Visible and Near-Infrared

1 Introduction

1.1 Outline of the Thesis

The importance of hyperspectral imaging systems has grown significantly in the last several years. The huge increase in the hyperspectral computational systems capacity, together with the reduction of its dimensions and weight, has encouraged its use in several domains such as: agriculture, industry, inspection and surveillance. The development of novel computational methods, some of them enabling parallel hardware/software implementations, therefore more capable of processing the vast amount of generated data, also contributed to motivate and increase hyperspectral cameras range of applications.

In this thesis we address the problem of acquiring and processing hyperspectral imaging data in real time, using unmanned aerial vehicles (UAV) for maritime surveillance applications. We develop and implement novel methods that allow to separate the target (boat) from the background. The methods are useful in order to decrease the huge amount of data that needs to be transmitted from the Unmanned Aerial Vehicle to a remote station on the ground for further processing.

Hyperspectral imaging systems are able to obtain information about the constituent elements of a scene, by acquiring spectral band information from each pixel. The integration of hyperspectral imaging in robotics applications is increasing due to this capability, namely in remote sensing applications using manned/unmanned aerial vehicles.

Up until now, hyperspectral imaging systems were mostly used in offline applications. The collected information was simply too much for the data to be acquired and processed in real time. Neither were the hyperspectral imaging acquisition systems ready to provide to other systems the hyperspectral information in real time. However, there is an emerging need to detect characteristics of the scenarios in real-time and also to reduce the cost of knowing if the hyperspectral systems information is useful or not.

1.2 Motivation

The Autonomous Systems Laboratory (LSA)[1] has been actively involved in cutting edge projects over the past years. These projects are applied to different areas of field robotics applications such as: marine robotics (with surface and underwater vehicles), aerial robotics and terrestrial robotics.

(a) TIGRE (b) ROAZ II (c) OTUS

Figure 1.1: *Some of the robots developed in LSA.*

In addition to the projects carried out based on robots developed at the laboratory, there are other projects carried out with other entities, namely with: Portuguese Air Force, Portuguese Navy, and other research laboratories and enterprises, national and foreign. Some of these projects are national, others exist at European level.

In this context arises the Smart UNattended airborne sensor Network for detection of vessels used for cross border crime and irregular entrY (SUNNY)[2] project, which aims to create and develop novel sensors and algorithms for autonomous aerial surveillance of maritime borders.

The SUNNY project represents a step beyond existing research projects due to the following main features:

A two-tier intelligent heterogeneous Unmanned Aerial Vehicle (UAV) sensor network will be considered in order to provide both large field and focused surveillance capabilities, where the first-tier sensors, carried by medium altitude, long-endurance

[1] http://www.lsa.isep.ipp.pt/
[2] http://www.sunnyproject.eu/

autonomous UAVs, are used to patrol large border areas to detect suspicious targets and provide global situation awareness. Fed with the information collected by the first-tier sensors, the second-tier sensors will be deployed to provide more focused surveillance capability by tracking the targets and collecting further evidence for more accurate target recognition and threat evaluation. Novel algorithms will be developed to analyse the data collected by the sensors for robust and accurate target identification and event detection;

Novel sensors and on-board processing generation, integrated on UAV system, will be focus on low weight, low cost, high resolution that can operate under variable conditions such as: darkness, snow, and rain. In particular, SUNNY will develop sensors that generate both RGB image, Near Infrared (NIR) image and hyperspectral image and that use radar information to detect, discriminate and track objects of interest inside complex environment, over land and sea. Alloying to couple sensor processing and preliminary detection results (on-board) with local UAV control, leading to innovative active sensing techniques, replacing low-level sensor data communication by a higher abstraction level of information communication.

Figure 1.2: *Fixed-wing UAV example.*

Among other objectives, the air surveillance of maritime borders aims to detecting illegal entry of vessels for drug traffic, illegal immigration, etc. Accordingly, traditional Electro-Optic (EO) and InfraRed (IR) cameras information is used to detect the existence of a boat, as well as, its shape and dimensions.

But what if we want to know what is the boat material? Or if we want to identify its cargo materials?

To be able to do so, hyperspectral cameras are used for providing information on the boat materials. Using hyperspectral information, we are able to detect the target, and identify the target to gain more relevant information. However, hyperspectral sensors are not commonly used in real time due to the amount of data obtained in a short period of time. Posing a complex problem, since the amount of data is much

higher than what typically is obtained with visible or infrared cameras.

In this thesis we tackle the problem of obtaining hyperspectral data in real-time due to the need of creating a real time processing pipeline of a hyperspectral camera, which allows to obtain information about the observed scene for robotics applications.

The whole processing must take place on board the UAV. However, this type of vehicle imposes some restrictions in terms of payload weight, as well as, energy consumption. This issue introduces additional challenges, whether inherent to the type of camera used, either to the real-time processing level.

1.3 Application Scenarios

The main application of the proposed system is to perform surveillance of maritime borders with unmanned aerial vehicles.

The proposed system solution operates in real time, and can cover different situations. Some possibilities are:

- Analysis of vegetation or minerals;
- Surveillance;
- Inspection tasks;
- Agriculture - monitoring the development and health of crops;
- Chemical imaging;
- Environment continuous monitoring.

1.4 Objectives

This thesis addresses the issue of hyperspectral imaging for real-time unmanned aerial vehicles maritime target detection. To accomplish this task, there are several objectives that need to be fulfilled, namely:

- Study of hyperspectral camera, and its data collection. Initially we examine this type of cameras operation, in order to understand the obtained data. Only after this preliminary study we started to explore the potential use for hyperspectral imaging data;

- Explore and understand the use of specific tools to analyse hyperspectral data. By exploiting the existing tools to process this data becomes simpler to study the options. Furthermore, these tools are used to confirm the obtained results;
- Development of a data acquisition pipeline in real-time. The first step to create a real-time processing system is to acquire data. As already mentioned, the amount of data is quite high, therefore it poses some challenges to the creation of a processing hyperspectral data pipeline;
- Data processing pipeline creation and system adaptation for real-time. Process the data obtained using the developed pipeline. This pipeline is based on the knowledge acquired during the execution of the first two objectives;
- Boresight angles calibration. Necessary to calibrate and relate the GPS/IMU position with the one of the image sensor. This calibration is is important for obtaining a good geo-reference of the image data;
- Detect targets (boats) in maritime environment, using hyperspectral data. Processing in the spectral domain is a complex task, especially in real time. One of the most innovative aspects of this thesis is the detection of targets in a maritime environment in real time;
- Perform data collection in real time to validate the developed approach.

1.5 Contributions

The work performed during this thesis, contributed to the development of a novel hyperspectral data processing system. More specifically:

- Creation of a real-time processing pipeline. Therefore, it becomes possible to detect maritime targets in real time, as well as, perform geo-referencing of the target;
- Detect boats against its background;
- Correction of the non-uniformities of a hyperspectral camera in flight;
- Boresight calibration for a significant reduction of the error associated to geo-rectification;
- Adaptation of a multispectral setup (visible, infrared and hyperspectral camera) in a fixed wing air vehicle. This allowed to obtain multispectral data, time synchronized between all the cameras, with inertial navigation data.

1.6 Thesis Overview and Organization

In the next chapter, the preliminary study on thesis topic is presented. It analyses existing methods for the different stages of processing, as well as the related work.

Chapter three presents the basic principle concepts for a good understanding of the developed work. Moreover, it also introduces the underlying concepts of hyperspectral cameras technologies.

The conceptual approach in chapter four discusses the architecture of the proposed system, in terms of hardware and software. All the different components are detailed, as well as, the developed pipeline.

The fifth chapter shows the develop system. Details its modules and all hardware/software implementations.

In chapter six, the boresight calibration results are presented, as well as, the detection of maritime targets using captured data in real time environment.

Finally, in the last chapter we discuss some conclusions on the hyperspectral imaging system results, and present future work directions.

2
Related Work

We will start this chapter, with a brief introduction to hyperspectral camera and hyperspectral imaging concepts. Subsequently, is displayed work inherent with this type of cameras such as the atmospheric correction and boresight calibration. Existing applications, as well as other complementary information necessary for the development work is also presented.

2.1 Hyperspectral cameras

In order to understand the working principle of a hyperspectral camera, as well as the different technologies involved, we conducted a small study in order to know the existing hyperspectral systems applications.

The use of an hyperspectral camera is becoming ubiquitous in remote sensing operations [1] [2]. Its use in data collection campaigns is becoming frequent in applications that range from agriculture, landscaping geo-referencing, food processing, mineralogy to others such as surveillance and inspection tasks [3] [4].

In our maritime target detection, recognition and identification application, hyperspectral data allows us to acquire image data of a target scene over the electromagnetic spectrum, from ultraviolet to long-infrared. Based on this information, we are able to obtain detailed information about the scene, which facilitates the detection of the vessel building material e.g fibre, aluminium, wood, rubber, persons on-board and also the cargo carried by the vessel. When all image data is associated,

it generates a hyperspectral image cube, consisting of a set of images layered on top of one another, as display in Figure 2.1. Each image corresponds to one wavelength. Therefore, the hyperspectral cube has a spectral dimension equal to the number of data wavelengths that the hyperspectral camera is able to acquire. Given that an image corresponds to one wavelength (in spectral dimension λ), each pixel (in spatial dimension XY) consists of a spectrum over the different wavelengths observed by the camera. This characteristic is named "spectral signature". If one compares the spectral signature to a previously known material one, can identify the type of vessel building material.

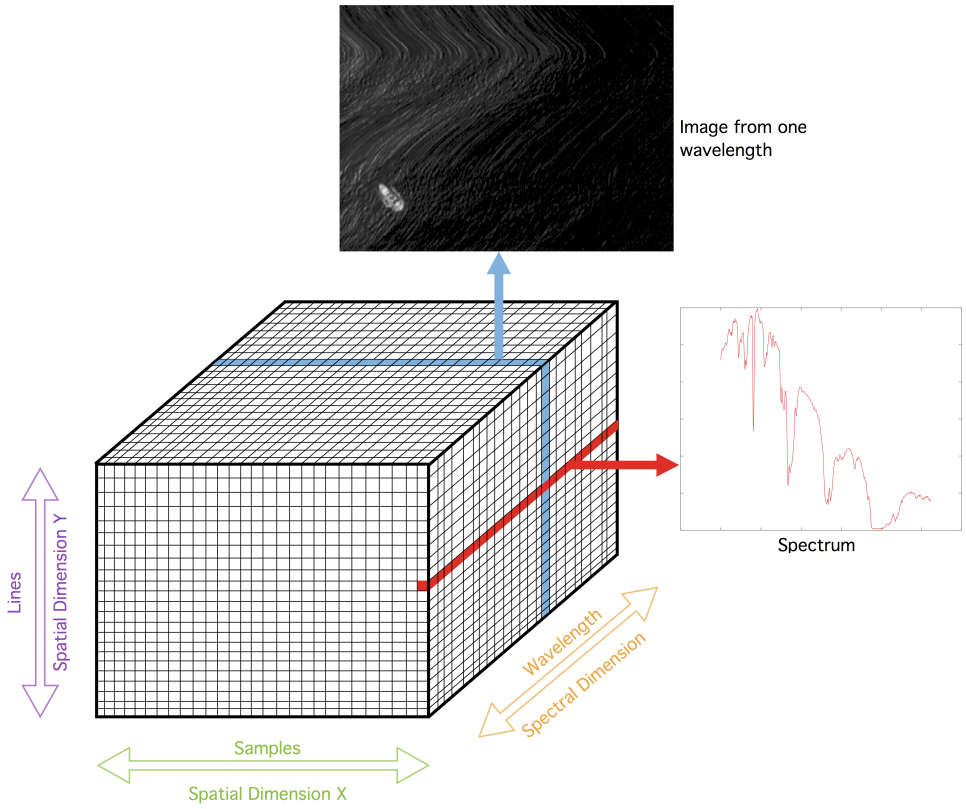


Figure 2.1: *Hyperspectral data cube.*

There are two main approachs for detecting a target using a hyperspectral data: using a pushbroom camera or by using a tunable spectral filter. In this work, we used a hyperspectral pushbroom camera.

2.1.1 Pushbroom cameras

This method allows full spectral data simultaneously, with spatial line scanning over time. With this, the camera acquires all spectral information exactly at the

same time, being insensitive to instrument/sample movement. Also, there is no moving parts in the instrument, and can collect light from sample to camera 5 to 20 times more efficiently than tunable spectral filter. Furthermore, only a line across the sample needs to be illuminated, which leads to 10 to 30 times more light and consequently more speed. So, the data is received line by line, for all wavelengths at the same time. It means that for each line, we obtain the spectrum for each pixel. Figure 2.2 shows the data acquisition scheme for the pushbroom method.

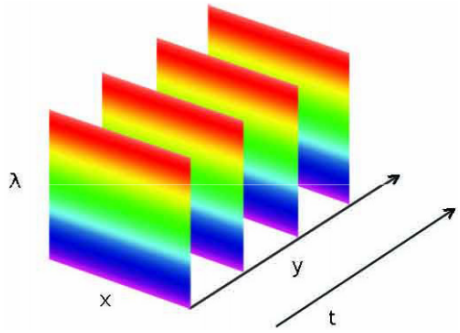


Figure 2.2: *Pushbroom aquisition method, adapted from* [1].

Figure 2.3 displays the UAV trajectory and scan line acquisition for a hyperspectral pushbroom sensor.

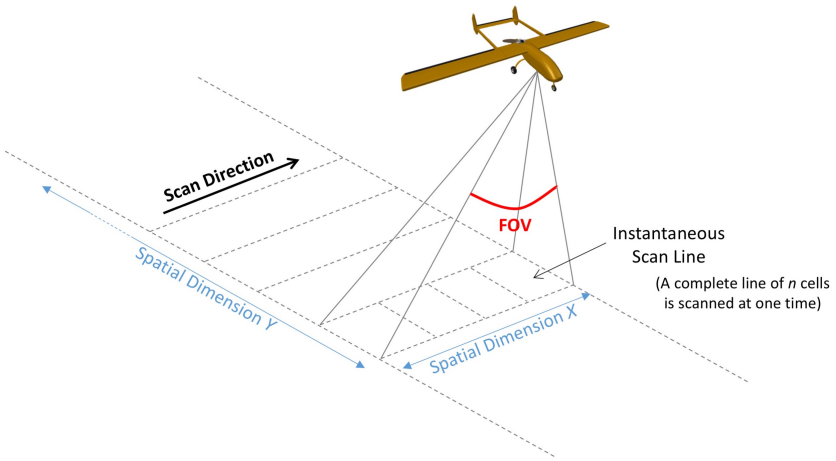


Figure 2.3: *UAV pushbroom scan methodology.*

[1] http://www.cosch.info/documents/14030/52106/Hyv%C3%A4rinen_presentation.pdf/021725c5-a6af-4003-8bb2-4cfd2c70ccc4 accessed on 17/02/2016

2.1.2 Tunable spectral filter cameras

The tunable spectral filter allows 2D image at a time, wherein each 2D image corresponds to a wavelength. So, only when all wavelength (2D image) are acquired, it becomes possible to obtain the spectrum of each pixel.

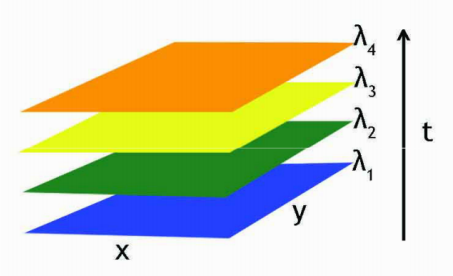


Figure 2.4: *Tunable spectral filter aquisition method, adapted from* [1].

2.2 Atmospheric correction

Now that we introduced the concept of hyperspectral cameras and its acquisition methods, we can start to analyze the hyperspectral data.

The hyperspectral data is captured at different wavelengths. So, there are numerous interferences associated with the acquisition of such information. These interferences are mainly due to the effects of the atmosphere, that, among others affect the hyperspectral data collection. Figure 2.5 represents these effects and illustrats how not all reflected light captured by the camera comes from the object. There is light directly from the light source, as well as other reflections in the soil. Therefore, it is necessary to use atmospheric correction algorithms.

However, the pushbroom sensors contain some nonuniformities (NUC). These derive mainly from the variability in the response of the Focal Plane Array (FPA) elements and the small-scale roughness of the camera's entrance slit. The presence of dirt or condensation on the entrance slit can make this nonuniformities most meaningful. For linear sensors, the NUC may consist of an offset and a single multiplicative factor for each element on the FPA. For nonlinear sensors, the NUC may require a higher-order relationship. The offset value is commonly computed in-flight by collecting data with the camera aperture closed and then averaging that data to obtain what is called in literature as a "dark frame". Visible and Near-Infrared (VNIR) sensors are usually very stable, so, its assumed that the values obtained in laboratory calibration will work well on subsequent data collects in the field.

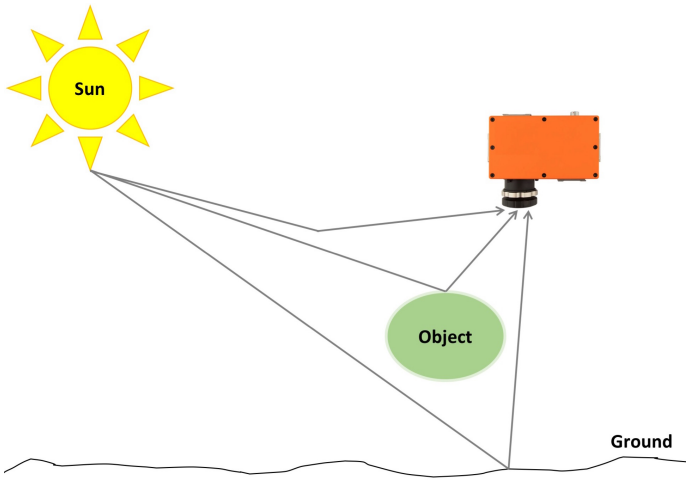


Figure 2.5: *Reflections captured by hyperspectral camera while observing an object.*

Then if there is any dust or condensation in the slit, or a physical shift in the alignment between the entrance slit and the focal plane array, will provoke differences between the system in the time of the laboratory calibration and at the end of the field campaign. However, these differences tend to be more common for Short-wave infrared (SWIR) sensors, and the FPA response tends to be more unstable. If the NUC is calculated as part of the calibration, it gives the conversion of digital number to radiance; separately, only gives a relative calibration.

There are several methods for performing atmospheric correction, including model-based and empirical-based methods [1].

In the first case, model-based method, the radiance at the sensor is modeled using radiative transfer models and data from detailed atmospheric and sun information archives (e.g., MODTRAN [5]). So, field measurement is not necessary, and the only required information is: site height and location, flight altitude, local visibility and acquisition times. There are multiple techniques that are based on this method, such as: ATREM [6], ATCOR [7], ACORN [8], FLAASH [9], CAM5S [10], QUAC [11] and HATCH [12]. In the case of empirical-based methods, these only require scene information and do not rely on physical information as in the model-based methods. Within this type of method, there are two distinct applications:

1. Require raw scene data without ground reference information;

2. Require raw scene data together with ground reference information.

In the first category, the two common approaches are: flat field method [13] and the internal apparent relative reflectance methods [14]. In both methods, the raw spectral data of each pixel is divided by a reference spectrum (in flat field, a

homogenous bright target and in internal apparent relative reflectance, an average scene spectrum). The disadvantage of using these methods is that they rely too much on the scene.

Empirical Line [15] method belongs to second category. In this type of approach two (or more) targets in the scene are linearly correlated and compared with the spectral signature obtained in laboratory.

Another empirical method is dark subtraction[1]. The method removes the effects of atmospheric scattering from an image by subtracting a pixel value that represents a background signature from each band.

Both methods, model-based and empirical, have disadvantages. While the model-based methods require additional computational resources. The empirical methods require to have additional ground information to be effective, otherwise their results output is deply affected.

2.3 Georectification

Unlike frame photography, each line of airborne pushbroom images is collected at a different time instants. Perspective geometry varies with each pushbroom line. Therefore, each line has different 6-Degrees Of Freedom (DOF) parameters (e.g. position, latitude, longitude, altitude and roll, pitch, heading of an image at the time of exposure), leading to the displacement of each line different and thus the rectification process much more difficult. However, one of the purposes of acquiring data from a hyperspectral camera is to find targets or certain spectral signatures. Therefore, it is necessary to geo-reference the hyperspectral data, and detection positions on the ground.. For this it is necessary to perform a task called geo-rectification, which is the projection of the point from the image reference frame to the world (ground) reference frame.

To perform this step, there are three different approaches:

- Direct georetification;
- Indirect georetification;
- Integrate georetification.

The first implies that the air vehicle is equipped with navigation sensors. Its output are exterior orientation parameters set for each line obtained by the hyperspectral camera [16] [17] [18] [2]. The exterior orientation parameters consists of

[1] http://www.harrisgeospatial.com/docs/AtmosphericCorrection.html#Using acessed on 04/04/2016

parameters such as the orientation of the camera, as well as its position in the world. With these parameters, the lines can be geometrically projected to the ground. This method works regardless of the existence of control points on the ground, being however quite susceptible to errors between the positioning of the camera and the navigation sensor. These errors are commonly called boresight errors, and are discussed in detail in the following chapter.

In the case of indirect georectification, features from a reference image are used and matched with features of the acquired lines to project them on the ground [19] [20] [21] . Therefore, its not required the navigation sensors use, such as Global Positioning System (GPS)/Inertial Measurement Unit (IMU). The most used methodology is to collect ground control points, which can later be identified in the reference image, or in the scan line, using a bundle adjustment process. The method requires human intervention to collect the ground control points and match them from one dataset to another.

The final approach, already performs the georectification using the combination of direct/indirect methods [22].

2.4 Boresight calibration

When using a direct georeferencing method. Its required to have the camera position and attitude. These can be obtained using a GPS and an IMU. Normally, the two are on the same system being attached to the camera. However, there are misalignments between the imaging sensor and an GPS/IMU, which makes it necessary to calculate the angle difference between them, denoted as boresight angles. The errors that these angles correct, increase with the UAV altitude. Usually, the error lead to misalignment in all 3-axis. Since the GPS/IMU is not released from the chamber, the obtained values of angles remains constant. Their values are usually low, typically being less than 1°.

The most common method to determine the boresight angles uses control points on the ground, then, using bundle adjustment (such as iterative least squares method) can determine the value of boresight angles [23]. Some authors suggest the use of information on the topography of the land for projecting the points and find the error [24].

There are also approaches that converts the hyperspectral data to Red Green Blue (RGB) images. In addition to these data, they also acquire RGB images from a EO camera. Later, using bundle adjustment, they combine the two RGB images [25].

2.5 Describe materials with the spectrum

Hyperspectral imaging sensors are advanced digital color cameras with fine spectral resolution at given wavelengths of illumination. Instead of measuring the three colors - red, green and blue - these sensors measure the radiation reflected by each pixel at a large number of visible or invisible frequency (or wavelength) bands. Such an instrument is called an imaging spectrometer. The spectrometers are instruments that divide the impinging electromagnetic radiation into a number of finite bands and measure the energy in each band. Depending on the design of the spectrometer, the bands may be contiguous or not, and with equal width. Therefore, the measured discrete spectrum is usually not a uniformly sampled version of the continuous one.

In Table 2.1 we display the differences between spatial and spectral processing. The first is usually used with EO cameras, and the second takes advantage of the capabilities of hyperspectral camera.

Table 2.1: *Spatial processing versus spectral processing, adapted from [26].*

Spatial Processing	**Spectral Processing**
Information is embedded in the spatial arrangement of pixels in every spectral band (two-dimensional image)	Each pixel has an associated spectrum that can be used to identify the materials in the corresponding ground-resolution cell
Image processing exploits geometrical shape information	Processing can be done one pixel at a time
Very high spatial resolution required to identify objects by shape (many pixels on the target)	No need for high spatial resolution (one pixel on the target)
High spatial resolution requires large apertures and leads to low signal-to-noise ratio	Spectral resolution more important than spatial resolution
Data volume grows with the square of the spatial resolution	Data volume increases linearly with the number of spectral bands
Limited success in developing fully automated spatial-feature exploitation algorithms	Fully automated algorithms for spectral-feature exploitation have been successfully developed for selected applications

There are two main types of identification targets using the spectral information:

- Spectral Anomaly Detection Algorithms [27]: This type of algorithms do not require spectral information from targets of interest. Therefore, all pixels whose spectrum does not fit the background model are marked as targets. This means that these algorithms don't allow to distinguish between spectral

anomalies generated by man-made objects, natural objects or targets of interest. Consequently, the atmospheric correction is not a prerequisite for the application of anomaly detection algorithms;

- Spectral Matching Detection Algorithms [27]: Need spectral information on targets of interest, and attempts to identify pixels whose spectrum display a high degree of correlation with the desired spectral signature. The spectral signature of the target can be obtained from a spectral library or from an identified scene in-target pixel.

An example of first type of methods is the RX algorithm [28]. The method identifies a spectral zone or with different colors within a region of tests which may be the whole dataset or only a portion thereof. The RX algorithm uses the pixel being analyzed as signal matched, since it uses the sample covariance matrix to take into account the sample spectral correlation. It performs the same tasks of the Mahalanobis distance, it is ideal for situations where the targets are small in comparison to the background. In addition to this, there is a variation called KRX [29]. This involves applying the same context as the previous algorithm, but using a sliding window. However, its computationally very expensive because it requires to calculate the covariance matrix and its inverse for each pixel of the image.

For second type algorithms, there are different options, such as: ACE [30], GLRT [31] and JSD [32]. These make use of probabilistic methods to model the target spectrum, determining what is the probability of this being in a given pixel.

2.6 Real-time algorithms

The creation of a real-time system based on the combination of the different techniques presented, allows the construction of a single system with sufficient processing capacity to detect the targets in real time. Up until now, there are not many approaches for hyperspectral real time data processing.

The work presented by Trym Vegard Haavardsholm et all [33], describes a target detection system in real time with GPU. It uses a hyperspectral system able to detect 160 wavelengths, and also able to set a background model. It detects possible targets that don't match the background. Although it was tested in a UAV, the processing of CPU was not on board.

In the work developed by Yuliya Tarabalka et all [34], a model background by fitting normal multivariate mix to a spatial subset of the image was used, then a background probability for each given pixel was computed in order to detect and es-

timated value that was afterwards use to detect anomalous pixels based on empirical values.

2.7 Hyperspectral Imaging Applications

In addition to surveillance, there are other fields where hyperspectral cameras are widely used. One of this fields is precision agriculture.

Using hyperspectral imaging to collect data from seeds, allows to determinate the germination of plant seeds [35]. With this, the scientists where able to detect significant changes in seed coat, responsible for the loss of germination. This technology can also be used to detect internal defects in food products [36].

Another application is in biotechnology. There are coral [37] and cells [38] studies using hyperspectral systems. It is also widely used in bacteria analisys [39] [40].

A different use for this systems is environmental monitoring, allowing to measure surface CO_2 emissions, map hydrological formations, track pollution levels, and more [41] [42].

2.8 Summary

Initially, there was a brief overview about the hyperspectral cameras. This allows to contextualize the work, as well to provide a better understanding of the algorithms presented.

This chapter also presents some underlying methods to the hyperspectral data processing, and relates their use for real time applications using UAVs.

3
Fundamentals

In this chapter, we will introduce the fundamental concepts for a better understanding of the work principles of a pushbroom hyperspectral camera.

3.1 Document Notation

Vector and matrices are represented by lowercase and uppercase letters, respectively. X^{-1} denotes the inverse of a matrix and X^T the transpose of a matrix. The Greek letters ψ, θ and ϕ represent the roll, pitch and heading angles, respectively. The letter f will be used for the focal length. Homogeneous vectors and matrix are represented by $\widehat{x}$ and $\widehat{X}$, respectively.

3.2 Vision

Visual sensing and perception of the environment is one of the most utilized means for a robot to be able to complete their tasks.

Today, there are several sensors capable of detecting various characteristics of the environment, most notably the distance sensors, sonar and optical sensors. However, optical sensors are those which represent an important role, since they are able to provide a large amount of information substantially similar to the information provided by the human eye.

For example, in the human eye, the cornea is the transparent coating surrounding the eye , while the pupil defines the opening of the iris in order to regulate the amount of light that strikes on the retina, performing the optical sensor function. The crystalline acts as the a lens, allowing focus the image on the retina. In the Figure 3.1 shows some of the constituents of the human eye to facilitate their identification.

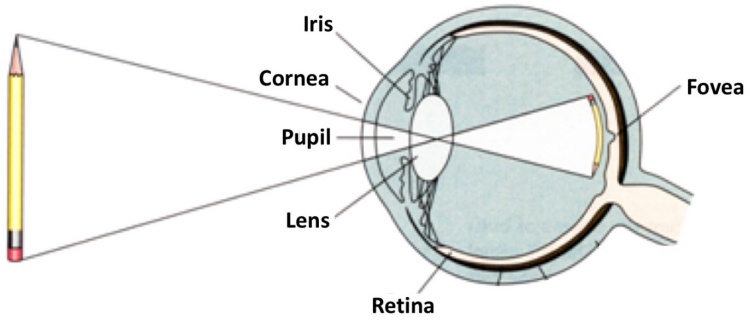


Figure 3.1: *Eye anatomy, adapted from* [1].

In the case of cameras, this may have an optical sensor of the Charge-Coupled Device (CCD) or Complementary Metal-Oxide Semiconductor (CMOS) type, which in comparison to the human eye, performs the role of the retina. These systems also have a mechanism for controlling the amount of light striking on the sensor, called the shutter, which acts similarly to the pupil of the human eye [43].

However, there are some differences between the cameras and the human eye. One of these is the fact that the human retina is concave unlike the optical sensors, which are flat. The case of the focus, standard lenses are focused to see far or near. In the case of the human eye, it is able to change the shape of the lens to focus on objects at different distances.

3.2.1 Pinhole model

In the case of normal visible cameras[44], the modelization of the image sensor is performed through the pinhole model. This describes the mathematical relationship between the coordinates of a 3D point in the world and the projection of this point on the image plane. However, it does not include geometric distortion or blurred objects through the lens, and the errors of the method depends on the sensor quality. This error will propagate from the center of the image to the edges.

The functioning of this model consists in placing all the light rays emitted and reflected to pass through a small hole, the pinhole of the camera, forming an inverted

[1]https://dundeemedstudentnotes.wordpress.com/2012/04/10/eye-physiology-optics/ accessed on 12/09/2016

image of the observed object in the image plane.

In a monocular system ruled by the pinhole model, the coordinate of each point can be deduced from the geometry of similar triangles, as defined in equation 3.1.

$$\frac{-y^I}{f} = \frac{y^C}{z^C} \Leftrightarrow -y^I = f \cdot \frac{y^C}{z^C} \tag{3.1}$$

Where f is the focal length, z^c is the distance between the camera and the object, y^c is the length of the object and y^i is the object length in the image plane. In Figure 3.2 can be observed each one of these elements.

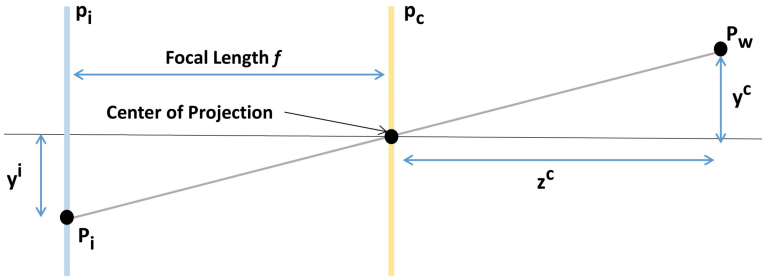


Figure 3.2: *Pinhole model.*

With the existence of a simple geometric relationship between the two planes, p^i e p^c, perspective projection can be described by a linear equation 3.2.

$$\begin{bmatrix} x^I \\ y^I \end{bmatrix} = \frac{f}{z^C} \begin{bmatrix} x^C \\ y^C \end{bmatrix} \tag{3.2}$$

The relationship between the points in the world (x^W,y^W,z^W) and points in the camera frame (x^W,y^W,z^W) can be obtained by a transformation comprising a rotation and a translation, in homogeneous coordinates:

$$z^C \begin{bmatrix} x^I \\ y^I \\ 1 \end{bmatrix} = PPM \begin{bmatrix} x^W \\ y^W \\ z^W \\ 1 \end{bmatrix} \tag{3.3}$$

The matrix PPM contains information of intrinsic and extrinsic parameters, being represented by:

$$PPM = A \begin{bmatrix} R|t \end{bmatrix} \tag{3.4}$$

In Figure 3.3 is represented the projection of an object in the world plane and the image plane.

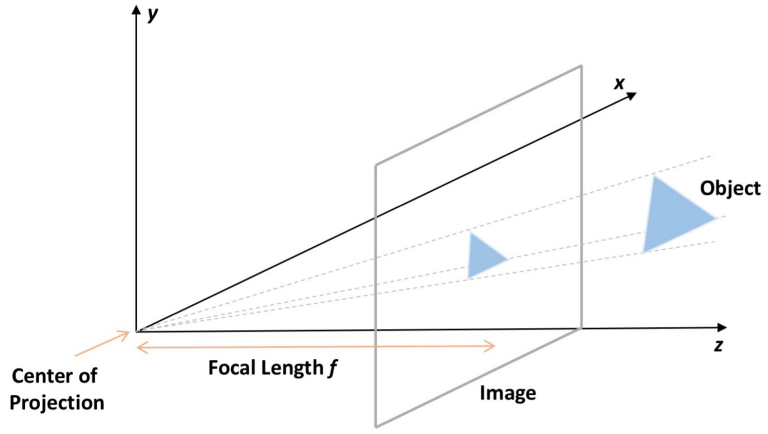


Figure 3.3: *Pinhole model: object in the world reference frame and in the image reference frame.*

Where A is the intrinsic matrix, and $\left[R|t\right]$ is the rotation and translation associated to the extrinsic matrix.

3.2.2 Intrinsic Parameters

The intrinsic parameters of a camera depend exclusively on their physical characteristics, including the internal geometry of the camera and lens type. In addition, it includes information relative to the distortion of the image sensor alignment and associated coefficients to image distortion.

These parameters need to be calibrated using specific techniques for this purpose, and the calibration returns the matrix A and the vector k.

$$A = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.5}$$

$$k = (k_1, k_2, k_3, k_4, k_5) \tag{3.6}$$

Where f_x and f_y represent the focal lenght, c_x and c_y the coordinates of the optical center, and the vector k contains the coefficients associated to image distortion.

So, the A matrix relates the image plane coordinates with the camera plane, i.e, relate the image reference (coordinate pixels) with the camera reference frame (standard coordinate).

3.2.3 Extrinsic Parameters

The extrinsic parameters are used to convert tridimensional coordinate in the world reference frame to the camera reference frame [45]. However, to get the coordinate of a 3D point of one reference frame to another reference frame, it's necessary to find the relationship between reference frames: rotation and translation. So, we need to define the rotation matrix R and the translation vector t. In the Figure 3.4 shows the rotation, and the translation of a point Q on the object reference frame for the camera reference frame.

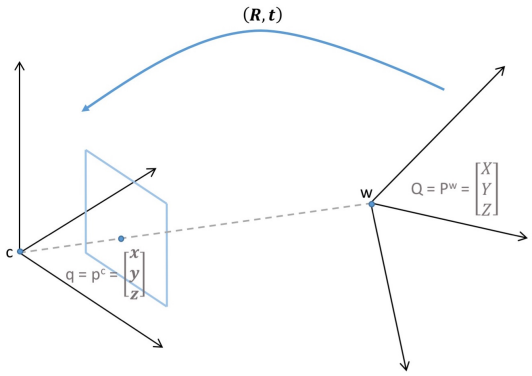


Figure 3.4: *Projection of the object coordinates (object reference frame) to the camera coordinates (camera reference frame).*

The rotation matrix can be represented in terms of Euler angles (*roll* (ψ), *pitch* (θ) and *yaw* (ϕ)) according to equation 3.7 [46]. The Figure 3.5 illustrates the application of these angles to a UAV (body reference frame).

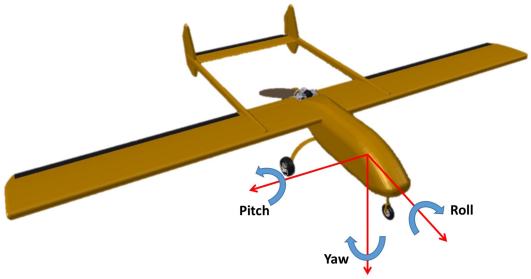


Figure 3.5: *Example of Euler angles applied to a UAV.*

$$R_{IMU}^{UTM} = R_x(\psi)\ R_y(\theta)\ R_z(\phi) \tag{3.7}$$

A rotation of ψ radians in the X-axis is defined by equation 3.8.

$$R_x(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\,\psi & -sin\,\psi \\ 0 & sin\,\psi & cos\,\psi \end{bmatrix} \tag{3.8}$$

In a similar way, a rotation of θ radians in the Y-axis described in the equation 3.9.

$$R_y(\theta) = \begin{bmatrix} cos\,\theta & 0 & sin\,\theta \\ 0 & 1 & 0 \\ -sin\,\theta & 0 & cos\,\theta \end{bmatrix} \tag{3.9}$$

Finally, in the equation 3.10 is defined the rotation of ϕ radians in the Z-axis.

$$R_z(\phi) = \begin{bmatrix} cos\,\phi & -sin\,\phi & 0 \\ sin\,\phi & cos\,\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.10}$$

3.3 Hyperspectral cameras

A hyperspectral camera is basically a advanced digital color camera with fine spectral resolution at given wavelengths of illumination. Therefore, it can measure the radiation reflected by each pixel at a large number of visible and infrared frequency (or wavelength) bands.

Consequently, the information obtained increases with the number of wavelengths that the camera captures. In the case of EO cameras, the number of bands observed is only three, while in the case of a multispectral system can vary between 10-20 bands. However, in the case of hyperspectral camera, this number may range between 100 to 500 bands.

3.3.1 Calibration

Before the camera is sent by the manufacturer, it was spectrally, radiometrically and geometrically calibrated. In addition, the sensor was characterized for the key parameters performance: spectral resolution, smile distortion, spatial resolution, keystone distortion, spectral sensitivity and signal-to-noise ratio.

In this case, the camera used was Aisa KESTREL 16 SPECIM system. Thus, the calibrations are presented as well as the characterizations made by SPECIM to this camera.

These calibrations and characterizations, as well as their interpretation are provided in a document created by the manufacturer.

Spectral calibration and resolution

Spectral (wavelength scale) calibration was done using spectral lamps (Argon, Mercury, Neon and Xenon). Spectral lamps were placed inside an integrating sphere and sensor was positioned in such a way that its entire FOV fits inside the exit aperture of the sphere. Afterwards 50 dark and image frames were collected and averaged, respectively. The averaged image frame was dark corrected. Spectral peaks were located from the image data and also Gaussian function was fitted to the located peaks. The centre position of the peak (in pixels) was obtained from the fitted data. Finally 2 to 4-degree polynomial was fitted to the peak positions to create the spectral calibration. Spectral calibration is a text file and it includes information about the spectral range, number of spectral bands, and centre wavelength and sampling (difference between adjacent rows) for each pixel row in the detector.

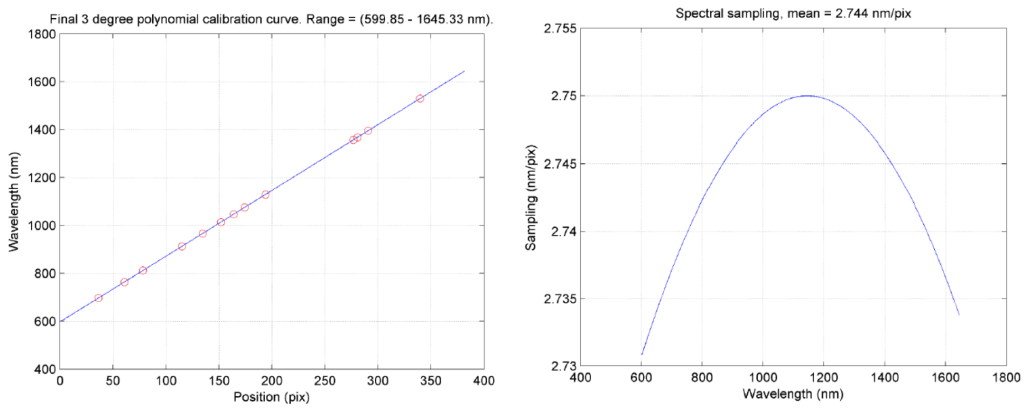


Figure 3.6: *Spectral calibration for the AisaKESTREL16 sensor (left) and spectral sampling determined from the spectral calibration (right).*

The dispersion of the KESTREL16 spectrometer is very linear and sampling per pixel is nearly constant (2.75 nm), as seen from chapter 3.6. For the determination of spectral resolution, the spectrometer with front optics was illuminated from a scanning monochromator with spectral resolution sharper than the sampling per pixel in the sensor. The monochromator was scanned with steps less than the sampling per pixel in the sensor. Light from the monochromator shall fill the sensor's numerical aperture. Spectral response was recorded for each step. Spectral resolution was determined as Full Width Half Maximum (distance between points on the curve at which the function reaches half its maximum value) from the Gaussian curve fitted

to the spectral response.

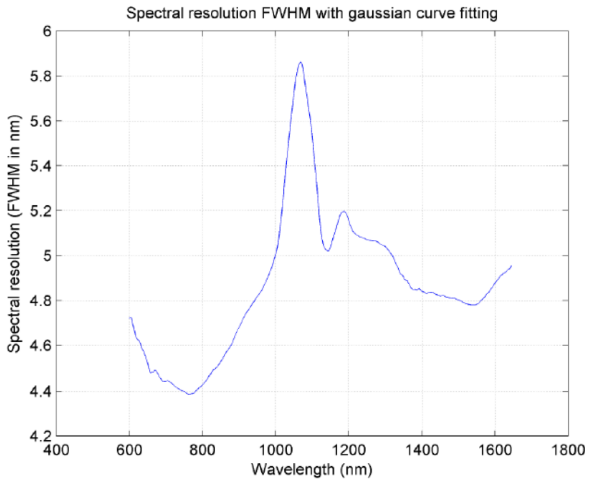


Figure 3.7: *AisaKESTREL16 spectral resolution (FWHM) measured with a monochromator.*

Smile

Smile was determined from the spectral calibration measurements by analyzing the spectral peaks position across the spatial dimension at five wavelengths across the spectral dimension. Smile is the maximum difference in the spectral peak position along the spatial axis. Smile was less than +/- 0.3 nm ($<$ +/$-$ 0.11 pix) which is below the design target specification.

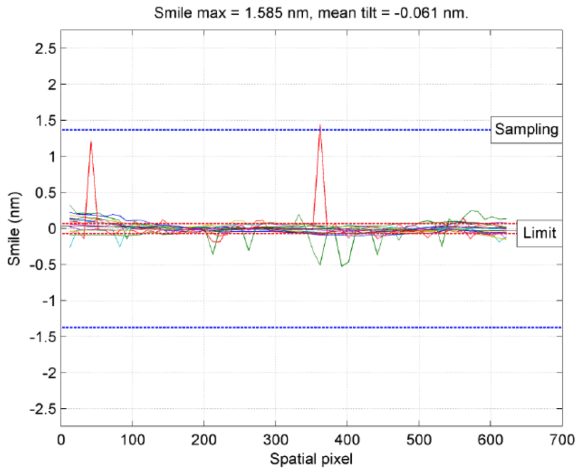


Figure 3.8: *AisaKESTREKL16 measured smile behaviour (the peaks are bad pixel related artifacts).*

Geometric Calibration and Spatial resolution

For the geometric calibration, the sensor was mounted on a 2-axial goniometer (allows an object to be rotated to a precise angular position) where it was illuminated with white light, through a collimator, from a pin hole. The pin hole image on the sensor's detector is a fraction of the pixel size. The angle position of the pin hole source/collimator can be precisely adjusted within the sensor's FOV with a motorized rotational actuator (Figure 3.9).

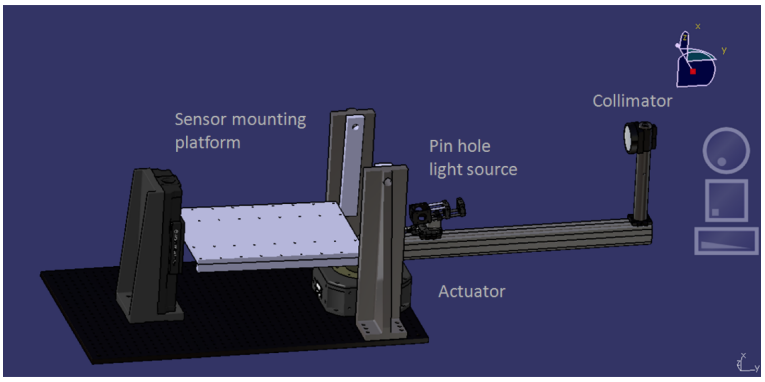


Figure 3.9: *Goniometric system for geometrical calibration.*

Twenty dark and image frames were acquired at 21 field angle positions within the sensors FOV. Illuminated spots were located from the image data and Gaussian function was fitted to the located spots. The centre position of the spots (in pixels) was obtained from the fitted data. Finally 2 to 4-degree polynomial was fitted to the spot positions to create the geometrical calibration. calibration results are shown in Figure 3.9. The actual geometrical calibration is a text file and includes information about the angular position in the FOV for each pixel column in the detector.

Spatial resolution was determined as Full Width Half Maximum from the Gaussian fitted spatial responses. Spatial resolution is well below the design target specification of 2 pixels.

Keystone

For Keystone, the spatial spot position along the spectral dimension was analysed at five spatial locations across the spatial dimension. Keystone is the maximum difference in the spot position along the spectral axis. Keystone was less than +/- 0.2 pixels and well below the design target specification of +/- 0.5 pixels.

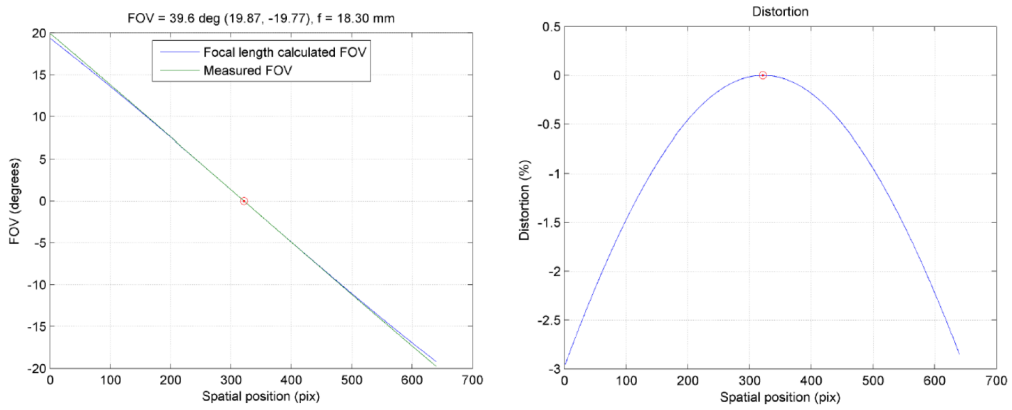


Figure 3.10: *Measured FOV with respect to linear (distortion free) FOV (left), and measured distortion in per cent (right).*

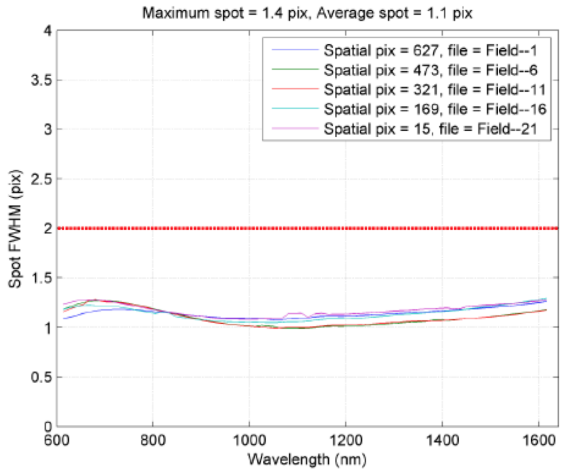


Figure 3.11: *AisaKESTREL16 spatial resolution measured in 5 field positions across the FOV.*

Radiometric calibration (Spectral response)

Sensor with front lens was positioned in front of a radiometrically calibrated integrating sphere in such a way that sensor's entire FOV fits inside the exit aperture of the sphere. 100 dark and image frames (with max signal at ca 82% of saturation) were collected and averaged, respectively. The averaged image frame was dark corrected. The known spectral radiance value at the sphere's output was divided, pixel by pixel, by measured dark corrected DN values and integration time. Final result is a spectral response (=radiometric calibration coefficient) array for each pixel in the image frame.

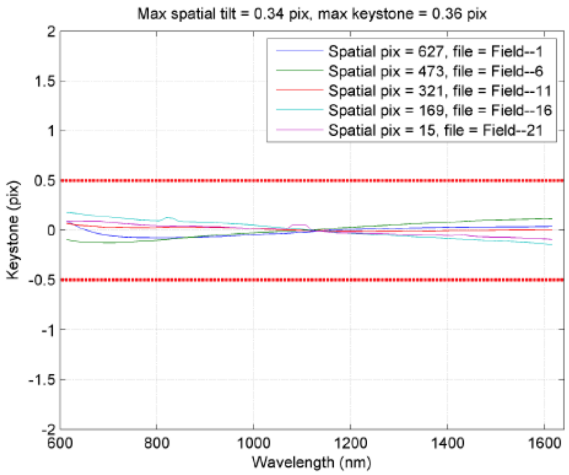


Figure 3.12: *AisaKESTREL16 keystone measured at 5 filed positions across the FOV.*

Calibration coefficient C for a pixel at wavelength λ and in spatial position of j and normalized to integration time is:

$$C_{\lambda j} = R_{\lambda}/(S_{\lambda j} - D_{\lambda j})/t_{int} \tag{3.11}$$

Where:

- R_{λ} is the spectral radiance at sphere output at wavelength λ;
- $S_{\lambda j}$ corresponds to the raw signal at wavelength λ in spatial pixel j;
- $D_{\lambda j}$ is the dark signal at wavelength λ in spatial pixel j;
- t_{int} corresponds to the integration time in the calibration measurement.

Signal-to-Noise Ratio (SNR)

SNR was determined from the 100 dark and images frames acquired for the radiometric calibration. SNR was determined as:

$$SNR = Signal/Noise \tag{3.12}$$

Where signal is the average of 100 dark subtracted signals, and noise is the standard deviation in the 100 signal frames.

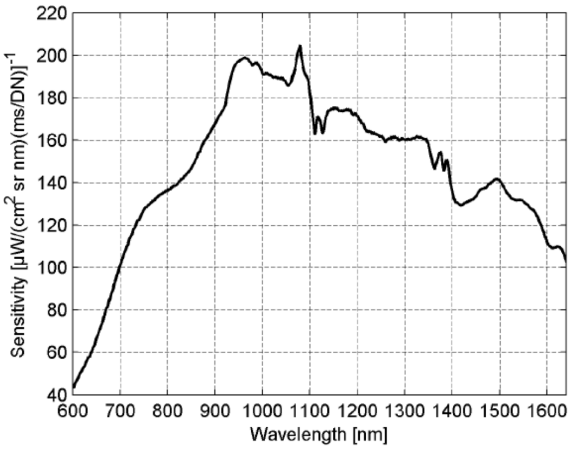


Figure 3.13: *AisaKESTREL16 measured spectral response, normalized by integration time.*

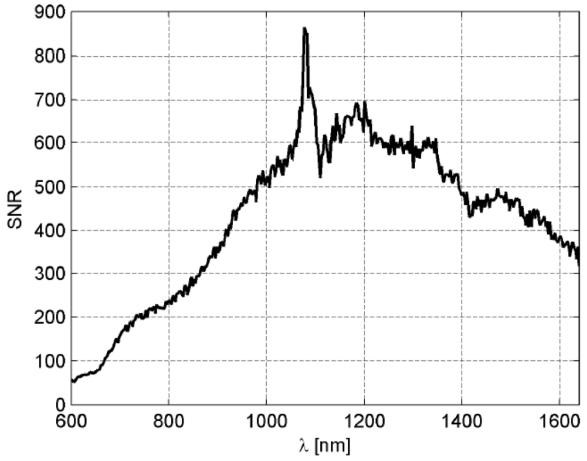


Figure 3.14: *AisaKESTREL16 SNR measured with a halogen light source, with max signal level 82% of saturation.*

3.3.2 Data obtained

Once all required parameters are calibrated, and performance tests are conducted, the camera is ready for use. As previously stated, the larger the number of wavelengths, the greater the amount of received data. So, it's necessary to take into account the data storage. Therefore, three distinct forms of storing data are defined in memory as:

- Band Sequential (BSQ): is the simplest format, where each line of the data is followed immediately by the next line in the same spectral band. This is optimal for spatial (x,y) access of any part of a single spectral band;

- Band-interleaved-by-pixel (BIP): stores the first pixel for all bands in sequential order, followed by the second pixel for all bands, followed by the third pixel for all bands, and so forth, interleaved up to the number of pixels. This format provides optimum performance for spectral (z) access of the image data;

- Band-interleaved-by-line (BIL): stores the first line of the first band, followed by the first line of the second band, followed by the first line of the third band, interleaved up to the number of bands. Subsequent lines for each band are interleaved in similar fashion. This format provides a compromise in performance between spatial and spectral processing.

For the case of a pushbroom hyperspectral cameras, using the BIL type, it will allow to receive the data from all the bands related to a line, forming all the spectral signatures observed in that line. This is usually the most used type, since it offers a compromise between spatial and spectral processing.

Every time we complete line, we can add it to the hyperspectral data cube.

The three types of data organization were defined by ENvironment for Visualizing Images (ENVI) [1]. Envi software is the most commom software for viewing and processing the hyperspectral data. In this software are implemented different processing algorithms, that are used for validation of results throughout this thesis.

3.3.3 Data type

The data returned by the hyperspectral sensor is raw data, i.e., digital numbers. This measures are commonly used to describe pixel values that have not yet been calibrated into physically meaningful units. This means that if we want a material spectrum, this values need to be corrected in order to obtain a measurement of radiance or reflectance. In hyperspectral imaging, raw data values, still contain influence from different factors such as: light source, sensor, atmosphere and surface material. Therefore, in order to extract only the surface material reflectance we need to remove all the other factors.

Radiance refers to the amount of radiation coming from an area. In order to derive a radiance value from a digital number, a gain and offset must be applied. In hyperspectral imaging, this calculation is performed for all pixels, and the gain and offset to be applied depends on the wavelength. This values are typically retrieved from the images metadata or received from the data provider. Radiance values normally have units of $w/(m^2\ SR\ nm)$. Transformation of the digital numbers to

[1] http://www.harrisgeospatial.com/ProductsandSolutions/GeospatialProducts/ENVI.aspx

radiance is called calibration. In order to perform this step, is necessary to calculate the radiance values for each pixel, as show in equation 3.13.

$$L_\lambda = Gain \times Pixel\ Value + Offset \tag{3.13}$$

With radiation, in addition to surface material, we have influence of light source and atmosphere. In some applications, this value may be sufficient for the desired implementation. However, if we want to obtain the spectrum of the material, it's necessary to calculate the reflectance.

Reflectance is the proportion of the radiation striking a surface to the radiation reflected of it. For some materials, it can be identified with the reflectance spectra, so it's common to correct an image to reflectance as a first step in order to try to identify the materials in the scene. There is also top-of-atmosphere reflectance (or TOA reflectance), being the reflectance measured by a space-based sensor flying higher than the earth's atmosphere. Although, this reflectance values will include influences from clouds and atmospheric aerosols and gases.

There is also another type of reflectance, called apparent reflectance. This one describes the combination between earth/atmosphere behavior with respect to the reflected solar irradiation.

The process of converting radiance into reflectance is called atmospheric correction.

Figure 3.15 shows the pipeline that allows to process raw hyperspectral data into reflectance data.

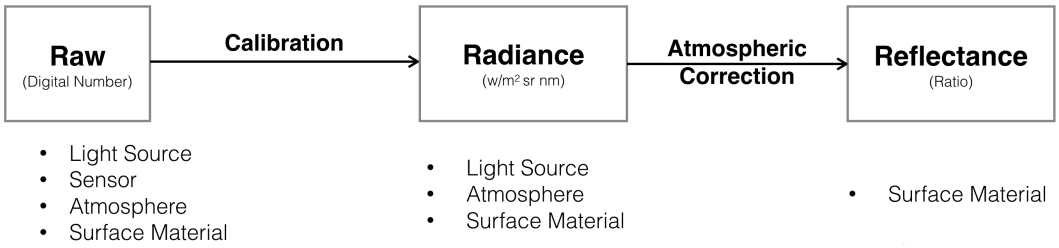


Figure 3.15: *Hyperspectral data processing pipeline.*

3.4 Point Projection - Pushbroom cameras

For our application, we need to obtain the target's position in the world. For this, we need to geo-reference the point, based on the information returned by the hyperspectral camera. In order to be able to perform the calculations, we need to know the system's position, as well as its attitude. That is the reason, why is common

to use an Inertial Navigation System coupled to the imaging sensor. We obtain the camera motion, thus its pose (position and attitude).

In chapter 3.2.1 we presented the case of the pinhole model. However, this model cannot be used in the case of pushbroom cameras.

3.4.1 Pushbroom model

In this thesis, we use a pushbroom hyperspectral camera. This type of cameras are electronic scanners that use a line sensor (CCD) for performing a scan along a two dimention scene.

First, the image/sensor reference frame is established. This is shown in Figure 3.16. The X-axis corresponds to the direction of flight. Regarding the Y-axis, it depends on the scanning direction of the camera, and can be defined to the right or to the left.

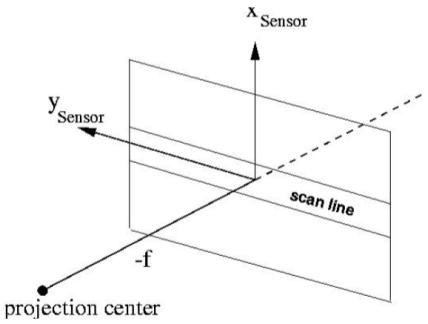


Figure 3.16: *Image reference frame, adapted from [2].*

Since this type of cameras performs the scanning of just one line, the x value will always be zero. Only y will have a variation corresponding to a constant spacing between the successive pixels. The focal distance in pixels can be calculated using the maximum scan angle, θ_{max}, and the total number of pixels, N, in a scan line. In equations 3.14, 3.15, 3.16 and figure 3.17 these relationships are presented.

$$x_i = 0 \tag{3.14}$$

$$y_i = -\frac{N-1}{2} + i, \qquad with\ i = 0, \ldots, N-1 \tag{3.15}$$

$$f = \frac{\frac{N-1}{2}}{tan\frac{\theta_{max}}{2}} \tag{3.16}$$

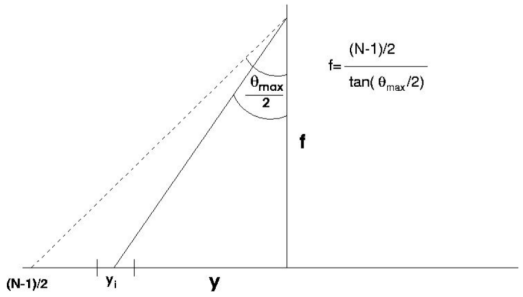


Figure 3.17: *Pushbroom model, adapted from [2].*

3.4.2 Direct geo-referencing

The use of direct measurements obtained by a GPS/IMU system to perform image rectification is called Direct geo-referencing. This method allows fast and automatic geo-rectification of the data. The concept of this method is shown in Figure 3.18.

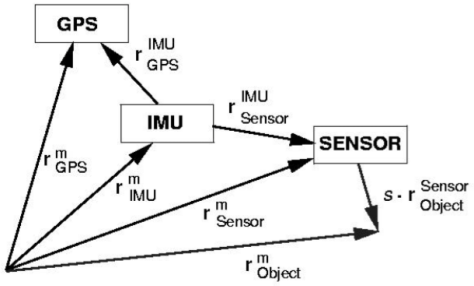


Figure 3.18: *Concept of direct geo-retification method, adapted from [2].*

The concept of collinearity is the basis for all the formulas applied to this method. The relationship between an object point r^{Sensor}_{Object} measured at the sensor reference frame is associated with the frame coordinates r^{m}_{Object} in the coordinate mapping as defined in equation 3.17.

$$r^{m}_{Object} = r^{m}_{Sensor} + s \cdot R^{m}_{IMU}\, R^{IMU}_{Sensor}\, r^{Sensor}_{Object} \tag{3.17}$$

The current position of the sensor centre of projection is given by the equation 3.18.

$$r^{m}_{Sensor} = r^{m}_{GPS} - R^{m}_{IMU}\, r^{IMU}_{GPS} + R^{m}_{IMU}\, r^{IMU}_{Sensor} \tag{3.18}$$

In equations 3.17 and 3.18, r is a vector whose upper index indicates the coordinate frame in which the vector has been measured, while the lower index point

corresponds to the point position. For the R (rotation matrices) case, , the transformation direction is indicated where the lower index represents the source system, while the upper index the destination system, as defined by the following items:

- m: Index which represents the mapping coordinate frame;
- r^{m}_{Object}: Vector for the object point on the surface of the Earth, in the coordinate frame mapping, which has to be determined;
- r^{Sensor}_{Object}: Vector from the image point, which is measured by the sensor pixel location to the corresponding object point on the earth surface in the sensor frame;
- r^{m}_{Sensor}: Vector to the projection center of the sensor in the mapping frame. This is calculated using the GPS/IMU data and the lever arm correction values. Usually it is determined during post-processing phase;
- R^{IMU}_{Sensor}: Rotation matrix between the sensor and the IMU, and should consider the boresight error;
- R^{m}_{IMU}: Rotation matrix between the IMU and the mapping coordinate frame;
- s: Scale factor. This is usually determined using a Digital Elevation Model (DEM), or techniques of stereo processing;
- r^{m}_{GPS}: Measured vector to the GPS antenna given in mapping coordinate frame;
- r^{IMU}_{GPS}: Vector from the IMU to the GPS measured in the IMU coordinate frame;
- r^{IMU}_{Sensor}: Vector from the IMU to the centre of projection of the sensor, measured in the IMU coordinate frame;
- r^{m}_{IMU}: Vector for the IMU in mapping coordinate frame, calculated using data from GPS/IMU.

Considering the model of a pushbroom camera shown in equations 3.14, 3.15, 3.16 and in Figure 3.17, together with the relationship shown in 3.17, it is possible to deduce the relationship between the coordinate of the object measured in the image with the coordinate of the point in the world using the equation 3.19.

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}_{Object}^{UTM} = f_{geo}^{UTM} \begin{pmatrix} x \\ y \\ z \end{pmatrix}_{Sensor}^{geo} + s \cdot R_{IMU}\, R_{Sensor}^{IMU} \begin{pmatrix} 0 \\ y_i \\ -f \end{pmatrix}_{Object}^{Sensor} \tag{3.19}$$

Where:

- $\begin{pmatrix} 0 \\ y_i \\ -f \end{pmatrix}_{Object}^{Sensor}$ corresponds to the target position in the image. x is 0 due to the model of pushbroom camera, while y_i corresponds to the equation 3.15, where N is the total number of samples (pixels) of the image and i is the pixel to be analyzed. f is the focal length in pixels;
- R_{Sensor}^{IMU} is the Rotation matrix from the Sensor reference frame to the IMU reference frame;
- R_{IMU} is the matrix that implements the rotations obtained by the IMU. This matrix can be represented in terms of Euler angles (*roll* (ψ), *pitch* (θ) and *yaw* (ϕ)), according to the equation 3.7 [46].
- s corresponds to the scale factor.
- $f_{geo}^{UTM} \begin{pmatrix} x \\ y \\ z \end{pmatrix}_{Sensor}^{geo}$ is the vector containing the UAV position when the target has been detected, in UTM coordinates.

3.5 Summary

In the section we presented the concepts and fundamentals necessary to understand the work principles of a hyperspectral camera. We presented the pinhole model, which is the most commom model used in robotics and computer vision applications, and then extended the pushbroom model which is the one used for the hyperspectral camera.

We also define the geo-referencing mode for a pushbroom sensor.

The camera calibrations and characterizations realized by the manufacturer are also described.

4
System Design

One of the main objectives of this thesis, is to develop a hyperspectral imaging system capable of being integrated into an UAV for maritime target detection. This implies that the overall system must takes into account certain requirements such as: weight, dimensions and limited power. So it is necessary to build a hardware architecture that allows implementation in a little airy and small environment. In addition, the software should run correctly in a low-power processor.

4.1 System Architecture

In this section, we will define the hardware and software architecture, and present their diagrams and data processing pipelines.

4.1.1 Hyperspectral Hardware System

The hyperspectral system comprises the following systems:

- SPECIM System: It comprises the SPECIM PC and Control Electronics, hyperspectral camera and the GPS/IMU. Performs acquisition of hyperspectral camera data and generates the timestamp message to geo-rectification;
- Data Processing System: is composed by the Data Processing CPU and an EO camera. It is responsible for performing the processing of the hyperspectral camera data, and data acquisition of an EO camera.

In Figure 4.1 is shown the configuration of the hyperspectral imaging system.

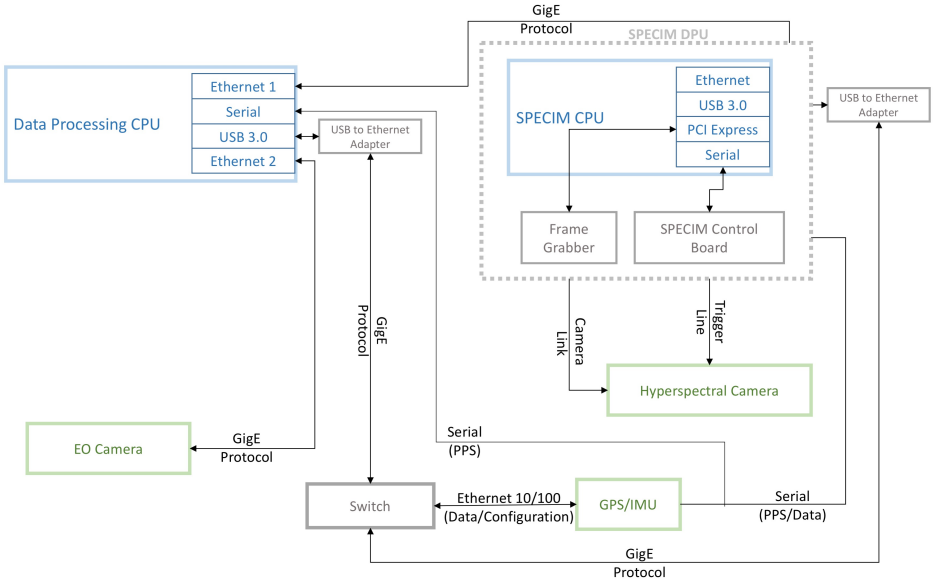


Figure 4.1: *Hyperspectral hardware system*

In the architecture, the hyperspectral camera sends data by the Camera Link Interface, so it is necessary to use a frame grabber to perform the interconnection of the sensor with SPECIM CPU. In addition, for being able to acquire the images, this CPU is also connected to SPECIM Control Board. This board, receive the PPS and the GPS/IMU data, generating the trigger pulse to start the acquisition of frames. This board is also responsible for generating the timing message between frames and inertial data, which then sends it to the SPECIM CPU. Moreover, it is necessary to consider that the GPS/IMU has an Ethernet connection to a switch, which is connected to both CPUs. This connection sends the GPS/IMU data streams formatted into a NCOM message, while still allowing the use of a specific software manufacturer in the SPECIM CPU to make the necessary settings (antenna position, etc.).

The other CPU presented in the system is the Data Processing CPU. It is responsible for acquiring data from the EO camera via Gigabit Ethernet protocol. In addition, the CPU receives the data from the hyperspectral camera and GPS/IMU through the Gigabit Ethernet protocol and is responsible for its processing.

Furthermore, there is still a PPS connection: this is a pulse which appears once per second, returned by the GPS/IMU. This is also connected to both CPUs and allows the synchronism of the two CPUs, using chrony software[1].

[1]https://chrony.tuxfamily.org accessed on 15/03/2016

4.1.2 Hyperspectral Software System

Previously we visualize the system's hardware architecture. As previously mentioned, the SPECIM system is a closed system that only returns the data obtained. The Data Processing CPU is where the data processing is performed. In Figure 4.2 is shown the processing pipeline accomplished in this CPU.

The main objective of this pipeline is to receive and process hyperspectral data in real-time. In order to accomplish this, first we need to perform some actions in offline, and then we are able to receive and process the hyperspectral data in real-time. The hyperspectral processing allows to detect targets and determine their position in the world, by perfoming the target geo-retification. Finally, we have the results ready to send to the ground station.

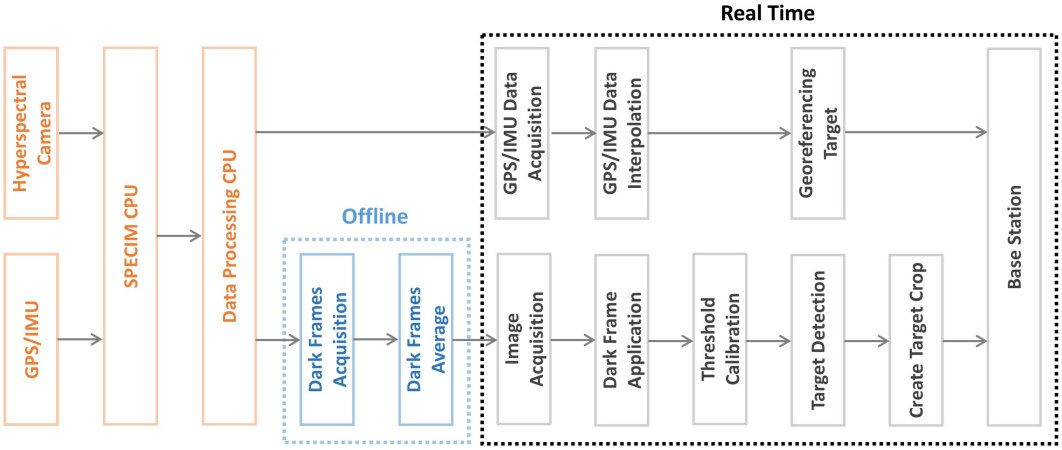


Figure 4.2: *Hyperspectral Software System - Processing pipeline.*

In this diagram, the left blocks correspond to the hardware, while the others blocks comprise the implemented software. As is identified in the diagram, there are two steps, one performed at the beginning of the flight (offline), and the other that corresponds to the real time processing. The first is responsible for calculating the offsets associated to the dark frames, that are later applied to all pixels. These values vary from flight to flight, and should always be calculated at its beginning. The second phase, which concerns the real-time processing, starts with the image acquisition and GPS/IMU data. Afterwards, the offset values can be applied to all pixels in the acquired image.

As soon as the procedure is implemented, the threshold calibration can be performed, which allows to detect targets in flight. However, in parallel with these calculations, there is the need to perform the interpolation from the obtained GPS/IMU data. Each line will have a position and an attitude, enabling the geo-rectification

of the point as soon as the target is detected.

After having detected the target, as well as its position, we perform a crop of the target, containing all spectral signatures. The crop of the hyperspectral cube is then sent to the remote station located on the ground, for further observation by the person in charge and processing of the required data.

4.2 Components Description - SPECIM System

Throughout this section, we address the SPECIM System hardware and software features. The system consists of three main hardware components, and a specific software developed by SPECIM.

4.2.1 SPECIM System Hardware

The hyperspectral camera is manufactured by SPECIM, and is a pushbroom hyperspectral camera.

The camera was built and idealized for SUNNY project, together with another model, and is now a commercial model (Aisa KESTREL[2]) of the brand. The models suggested by SPECIM were analyzed at the beginning of the project, and it was decided that the system would use is the SPECIM Aisa KESTREL 16.

The hyperspectral system has different hardware components, namely:

- Hyperspectral camera: It comprises the sensor, spectrometer and lens necessary for data acquisition in the frequency range. The sensor is a Bobcat 640 CL-V [3], manufactured by Xenics. It collect data by Camera Link interface, while still allowing the use of an external trigger that triggers the acquisition of the image. Camera Link interface for data acquisition guarantees sufficient transfer capacity even at the maximum data rates from the camera;
- PC and Control Electronics:
  - Computer motherboard (CPU) - With Windows 7 operating system;
  - SSD in a swappable slot;
  - Framelink Express frame grabber (Camera Link) - Device which makes the connection between the sensor and the CPU, for acquire the data;
  - SPECIM Control Board;
  - Power regulator.

[2]http://www.specim.fi/products/aisakestrel-hyperspectral-imaging-system/
[3]http://www.xenics.com/pt/camera/bobcat-640-cl

- GPS/IMU: GPS receiver and inertial system. It allows to detect the position and attitude of the UAV when the data was obtained. It is physical coupled to the camera and it's necessary to perform a calibration in order to eliminate alignment errors between the GPS/IMU and the sensor, called boresight calibration. If this is not performed, it can lead to significant errors during the geo-rectification of the collected points. The inertial system selected by SPECIM was the OXTS xNAV550 due to its small size and low weight.

As for the data, it is stored in a SSD which is the only hard drive option that provides the required speed, reliability and robustness. The SSD is exchangeable on a carriage slot. Storage capacity is 480 GB which is sufficient for approx 2.5 hours data collection at the highest image rate.

The SPECIM Control Board handles:

- Synchronization between image data and GNSS/IMU data acquisition with 0.1 ms accuracy;
- Shutter control in the hyperspectral camera.

The DC converter (regulator) has input range of 12 – 19 VDC.

In Table 4.1 is displayed the SPECIM AisaKESTREL 16 characteristics, and Table 4.2 shows the mechanical and power characteristics of the system.

Table 4.1: *SPECIM AisaKESTREL 16 characteristics.*

Parameter	**KESTREL 16**
Spectral Range	600-1650 nm
Spectral Sampling	4 nm
F/#	F2.4
Smile/Keystone	<0.15 pixels
Polariation Sensitivity	$< \pm 2\%$
SNR (Peak)	800
Spatial Resolution	320 or 640 pixels
Frame Time	100 to 350 Hz
Integration Time	Adjustable within frame period
FOV	21.7 or 40 degrees
Electro-mechanical Shutter	Yes
Data Interface	Cameralink 14-bit

4.2.2 SPECIM System - Software

The hyperspectral camera software allows to control the sensor parameters and start data acquisition. This program is called Lumo Recorder and is already installed on

Table 4.2: *SPECIM AisaKESTREL16 mechanical and power characteristics.*

Hardware	**Pushbroom Hyperspectral Camera**	**GPS/IMU**	**PC and Control Electronics**
Component	KESTREL16 	Oxford xNAV550 	Stacked card set 
Weight	2.3 kg	0.4 kg	<2 kg with cables
Power	<10 W	<6 W	<30 W
Dimensions	99 x 215 x 240 mm with front lens	120 x 70 x 40 mm	146 x 102 x 100 mm

the hyperspectral camera CPU.

Using this software, we are able to set parameters as the trigger (internal or external), the frame rate, exposure time, among others. In addition, this saves the data in different files as described in the next section.

The software also receives the GPS/IMU data, saving it in a specific format in conjunction with a timestamp message.

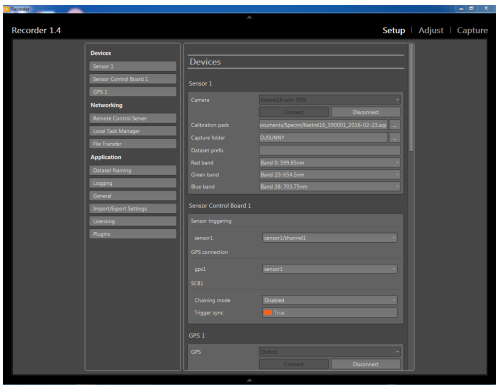
Using the software, we can see in real time the waterfall image of the camera being acquired, as well as the parameters of the sensor. It also allows to control the shutter and select the desired workflow. The workflow refers to the type of data acquisition performed, namely:

- Image;
- Dark frames;
- Image with GPS;
- Image, GPS and dark frames.

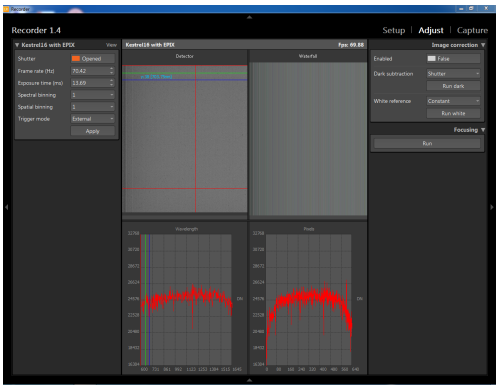
Figure 4.3 displays three screens of Lumo Recorder, and shows its main features.

4.2.3 Data acquisition using the Lumo Recorder Software

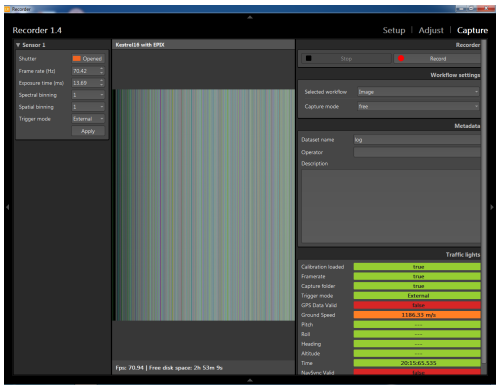
The data stored using the Lume Recorder is in a specific format defined by SPECIM. The Lumo Recorder saves different files for each log:



(a) Setup: where it is possible to connect the camera and the GPS / IMU, and set the folder where the data will be stored.



(b) Adjust: allows to adjust several parameters of the camera, such as frame rate, exposure time, spatial and spectral binning and control the shutter.



(c) Capture: responsible for presentation of the data in real time, as well as the start and stop of the record.

Figure 4.3: *Lumo Recorder*

- A png image with the *waterfall*: It corresponds to a graphical representation of data. Each line of the image corresponds to an line acquired, and does not contain all the acquired information. Usually, it's used for a quick check of the data;

- *Raw* file: It contains the data from the hyperspectral camera;

- *Header* file (.hdr): Indicates how the data is disposed in the raw file, in addition to the information it also holds the data size and in what wavelength data was obtained. This file follows the convention set by the ENVI program. It contains more information the one that is required for our application. Our system requires the following information:

  - *Samples*: Equal to the number of pixels of the image, which is part of its spatial dimension. In the case of camera used, is 640;
  - *Bands*: Corresponds to the number of wavelengths which data was obtained. Equivalent to the spectral dimension of the data. For our system, the Lumo Recorder enables the implementation of two values: 191 and 382. However, for our application we only acquired data using 382 wavelengths;
  - *Lines*: Are the number of lines captured in the dataset. This varies according to the log size;
  - *Interleave*: This parameter can be of three types: BSQ, BIP, or BIL. These three formats have been discussed in detail in section 3.3.2;
  - *Data type*: Type of data stored, where - *1=8-bit byte*, *2=16-bit signed integer*, *3=32-bit signed long integer*, *4=32-bit floating point*, *5=64-bit double-precision floating point*, *6=2x32-bit complex*, *9=2x64-bit double-precision complex*, *12=16-bit unsigned integer*, *13=32-bit unsigned long integer*, *14=64-bit signed long integer* and *15=64-bit unsigned long integer*. In this case, is used the type number 12, which corresponds to*16-bit unsigned integer*;
  - *Byte order*: Order of bytes, and it can be of two types, 0 or 1. The first corresponds to the least significant byte first (*least significant byte first - LSF*), while the second type is the most significant bit first (*most significant byte first - MSF*). In this case, the first type is used;
  - *Autodarkstartline*: As mentioned in section 2.2, the pushbroom sensors are subject to some nonuniformities. In this camera, it is necessary to capture dark frames, and then calculate their average. This value will correspond to an offset which must be applied to all pixels. The autodarkstartline parameter indicates from which captured line are the dark frames;

- *nav* file: This file contains the GPS/IMU data, as well as synchronized data between these and the hyperspectral camera.

4.3 Components Description - Data Processing System

Now, we will start describing the data processing system.

4.3.1 Data Processing System Hardware

There are two main components in this system: the CPU and the EO camera. We started by choosing the CPU, taking into account some limitations due to the platform that would transport the systems (UAV):

- Intel i5 or i7 processor: to allow the use of Hyper-Threading;
- Allow 8Gb of RAM;
- Have support for an SSD and a mSata simultaneously: the SSD to store the data obtained, and the mSATA for the operating system;
- Have at least one serial port;
- Two Gigabit Ethernet ports;
- 3 or 4 USB ports;
- Maximum size of 15 x 15 cm or equivalent in area;
- Low power consumption;
- 12V supply voltage. So only a DC/DC is needed.

After a detailed analysis of the products available in the market, we choose the CAPA 881 of Axiomtek. In Figure 4.4 is displayed the choosen CPU board.

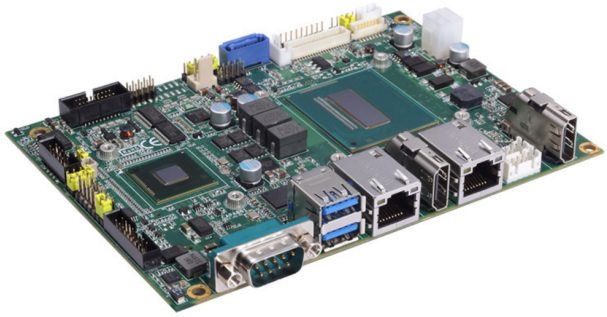


Figure 4.4: *CPU used: Axiomtek CAPA 881.*

The other element of this system was an EO camera, and the purpose of this sensor is to visually confirm the data obtained with the hyperspectral camera. Therefore, would have to fulfill the following requirements:

- Communication protocol: USB 3.0 or Gigabit Ethernet (GigE);
- 12V supply voltage;

- Low weight and reduced size.

The selected camera was the Blackfly 3.2 MP Color GigE PoE (Sony IMX265) model, of Point Grey. Figure 4.5 shows the selected camera.

Figure 4.5: *Camera used: Point Grey Blackfly 3.2 MP Color GigE PoE (Sony IMX265).*

4.3.2 Data Processing System Software

The Pointgrey camera, requires the use of a proprietary SDK, denominated *FlyCapture SDK*. In addition to providing a graphical interface for data acquisition and camera setup, this software also contain the libraries needed for the development of other tools of acquisition. Therefore, it becomes possible to use ROS[4], which allows EO and Hyperspectral image acquisition in real-time.

Additionally, this system also receives data from the hyperspectral camera and from the GPS/IMU. The first is received via GigE from SPECIM CPU, while the second message is received in two ways: directly from the GPS/IMU via ethernet (NCOM messages), and via GigE from SPECIM CPU. However, in the second connection we also receive one synchronization message for assigning a timestamp to each frame obtained by the hyperspectral camera. The reception of this data is performed using ROS, and explained in detail in the following chapter.

For synchronizing both the processing units, the PPS signal generated by GPS/IMU system is received in both units. We used the gpsd service daemon [5] for monitoring and receive the gps data. With the data received in the gpsd and the pps signal, we then use the Chrony clock synchronization daemon in order to synchronize internal clock of the processing unit to GPS time, assuring precise timing information.

4.4 Summary

In this chapter, we detail the individual hardware and software componentes, and match it to the corresponding hyperspectral imaging processing pipeline that we develop.

[4]http://www.ros.org/
[5]http://www.catb.org/gpsd/ accessed on 15/03/2016

5 Implementation

In this chapter, the various steps for performing the real-time processing are described.

5.1 Data Acquisition

This section presents the image acquisition mode. As previously mentioned, frames and GPS/IMU data obtained is sent by Gigabit Ethernet from SPECIM CPU to Data Processing CPU.

In order to connect both systems, it is necessary to define a communication protocol. So, it becomes necessary to define a communication protocol. The protocol was developed by SPECIM, with the name *Hyperspectral Data Streamer Protocol*.

5.1.1 Hyperspectral Data Streamer Protocol Specification

HyperSpectral Data Streamer Protocol (HSDSP) and Lumo Recorder are used to stream hyperspectral imaging data from a Lumo Recorder application to a remote host. There are two plugins that can be used to remotely get the Lumo Recorder data: Simple Remote Control plugin and Data streamer plugin. Simple Remote Control is not mandatory for data streaming, but it can be used for controlling the recorder functionality such as setting the frame rate, controlling the shutter or starting a workflow. Data Streamer plugin is needed to forward the recorded

hyperspectral imaging and GPS/IMU data over Ethernet to a remote host. Data streamer plugin can use either TCP or UDP as the transfer protocol, and creates two data streaming sockets, one for GPS/IMU data and other for captured hyperspectral data. The GPS/IMU data comes from the streamer plugin in the same format as it is stored in the Recorder. Captured hyperspectral data comes from the streamer plugin frame by frame as it is stored in the Recorder, but those frames are encapsulated in a header specified for the HSDSP.

Image Streamer Socket

The message associated to the image streamer socket consists of a segment header, data, and optional metadata sections. The HSDSP header contains 8 fields, followed by the data field, which is followed by the metadata field if present.

Figure 5.1 shows the message fields for the HSDSP protocol.

Offsets Octet	0	1	2	3
Octet	Bit	0| 1| 2| 3| 4| 5| 6| 7| 8| 9|10|11|12|13|14|15|16|17|18|19|20|21|22|23|24|25|26|27|28|29|30|31
0	0	[Start sequence (8bytes)			
4	32				]|
8	64	[Frame type ID (1byte)][Frame number (8bytes)			
12	96				
16	128	][Width (4bytes)			
20	160	][Height (4bytes)			
24	192	][Data type (1byte)	][Metadata size in bytes (4bytes)		
28	224		[Header checksum (2bytes)		]|
		[Data ...			
x	x	[Metadata ...	[Data checksum (2bytes)		]|

Figure 5.1: *Image streamer socket message.*

Where:

- Header start sequence (8 bytes - 64-bit unsigned integer): Identifies the start of a new data;
- Frame type ID (1 byte - unsigned char): Identifies the frame type. Enumeration, with the following values:
  0: Raw frame
  1: Dark frame
  2: Timestamped frame
- Frame number (8 bytes - 64-bit unsigned integer): Accumulated sequence number of the frame. Starts from 1 for each new workflow;
- Width (4 bytes - 32-bit unsigned integer): Defines the amount of horizontal pixels;

- Height (4 bytes - 32-bit unsigned integer): Defines the amount of vertical pixels;

- Data type (1 byte - unsigned char): Data type is in ENVI-format. It is an enumeration, with the following values: 1 = Byte: 8-bit unsigned integer
  2 = Integer: 16-bit signed integer
  3 = Long: 32-bit signed integer
  4 = Floating-point: 32-bit single-precision
  5 = Double-precision: 64-bit double-precision floating-point
  6 = Complex: Real-imaginary pair of single-precision floating-point
  9 = Double-precision complex: Real-imaginary pair of double precision floating-point
  12 = Unsigned integer: 16-bit
  13 = Unsigned long integer: 32-bit
  14 = 64-bit long integer (signed)
  15 = 64-bit unsigned long integer (unsigned)

- Metadata size in bytes (4 bytes - 32-bit unsigned integer): Defines the amount bytes used for metadata. If zero, then there is no metadata in the HSDSP segment. If greater than zero, then this field defines the size in bytes of the metadata that follows immediately after the data section as the last section of the HSDSP segment;

- Header checksum (2 bytes - 16-bit unsigned integer): 16-bit Fletcher's checksum calculated over header bytes. The check field of 16-bit Fletcher's is composed by two 8-bit values, initialised with the value 0. As each byte of the message is processed, its value is added to the first check value, and the remainder on division by 255 is saved as the new first check value. For the second value, the process is similar: its value is initialized to 0, and updated as each byte in the message is processed. However, it is updated by adding the current value of the first check value. The remainder on division by 255 is saved as the new second check value[1];

- Data (Width * Height * SizeOf(Data type)) bytes): The actual payload in the HSDSP segment. This is the same data that is stored in image acquisition;

- Metadata (Metadata size bytes): Optional section. Whether or not there is metadata in the HSDSP segment is defined with the Metadata rows - field.

[1] http://www.drdobbs.com/database/fletchers-checksum/184408761 acessed on 14/06/2016

If there is metadata, the length of this field is calculated from the Metadata rows, Width, and Data type values;

- Data checksum (2 bytes - 16-bit unsigned integer): 16-bit Fletcher's checksum calculated over data and metadata bytes.

GPS/IMU Streamer Socket

The Lumo Recorder creates a nav file with the position and attitude returned by GPS/IMU. The data is sent in a NCOM message, which is defined by the manufacturer of GPS/IMU, OXTS. This is a 72 byte message containing all position and attitude data, among others. The manufacturer offers its own decoder in C language [2] for this message, which facilitates reading. Using the decoder, we can extract data from NCOM message to a structure where are attitude and position data - necessary for the implementation of the system.

In addition, this file also contains a synchronization message created by SPECIM. This is synchronized with the PPS signal from the GPS/IMU, being added to the nav file once per second, as shown in Figure 5.2. The data contained in this synchronization message, allows to assign a timestamp to each frame captured by hyperspectral camera.

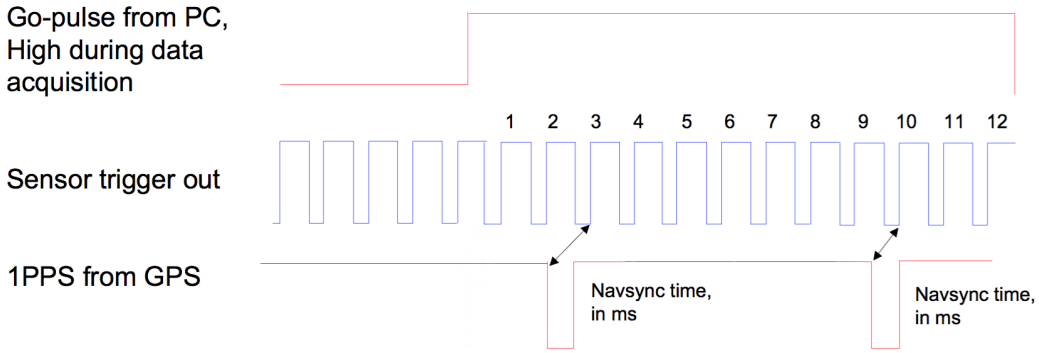


Figure 5.2: *Synchronization message insertion Scheme.*

Figure 5.3 shows the synchronization message fields.

This message allows access to various information relative to the frame with the value corresponding to *WORD7* (relative to the frame counter value). However, the value allows to calculate the different timestamps is contained in the *WORD6*: delay value. This corresponds to the delay, between the external trigger and PPS.

[2] http://www.oxts.com/new-ncom-decoders/ accessed on 01/03/2016

```
WORD1:  0x81FF          (sync)
WORD2:  0x03E6          (message id #998)
WORD3:  0x0002          (data word count)
WORD4:  0x8000          (flags, assuming no errors)
WORD5:  0xFA19          (header checksum)
WORD6:  0xtttt          (16-bit delay value, tick resolution 0.1 ms)
WORD7:  0xnnnn          (16-bit frame counter value)
WORD8:  0xaaaa          (data checksum)

Header checksum is WORD1 XOR WORD2 XOR WORD3 XOR WORD4.

Data checksum should be calculated according to time and counter
value.
         WORD8   = WORD6 XOR WORD7
```

Figure 5.3: *Synchronization message fields.*

The frame time is calculated using the NCOM messages. When we receive a synchronization message, the next navigation message tells the GPS time in full seconds (leave out the decimals) that the synchronization message is referring to. Frame counter tells the frame number, and delay value gives the decimals to be added to the GPS time in full seconds. The following steps, describe the procedure:

1. Take the GPS time in full seconds (*ssss*) from the next navigation data message;

2. GPS time for frame *nnnn* is *ssss,tttt* (note that *nnnn* and *tttt* have to be converted from hex into decimal numbers);

3. Interpolate GPS time for the frames between two last sync messages: Time between the frames is:

$$T = \frac{(tttt\_current - tttt\_previous + 1000)ms}{nnnn\_current - nnnn\_previous} \tag{5.1}$$

   So time for frame N is:

$$timestamp_N = ssss\_previous.tttt\_previous + T * (N - nnnn\_previous) \tag{5.2}$$

Based on this procedure, a acquistion time for each frame is obtained. However, the hyperspectral camera and the GPS/IMU are not acquiring data at the same frequency. This means that the probability of receiving a GPS/IMU message with the same time frame is very low. So, it's necessary to interpolate the data received from GPS/IMU, in order to obtain an approximate attitude and position for each frame.

5.2 Boresight Calibration

The boresight error is due to the misalignment error between GPS/IMU and the sensor, and must be calibrated whenever these two elements are separated. Without this calibration, georeferencing is influenced negatively.

One way to determine the boresight angles is to collect control points on the ground, and then relate this ground control points with the hyperspectral points, in camera reference frame. The matching between the ground control points and image point is performed in 2D. However, the ground control points collected are acquired in 3D, so there is the need to re-project this points to the camera reference frame, obtaining a 2D ground control point.

The ground control points are obtained with GPS receivers, and the image points are obtained by performing a calibration flight with the hyperspectral camera.

The difference between the image re-projection of the GCP to the image, and the image points acquired in flight is due to the boresight error. The equation that relates this procedure is described in equation 5.3.

However, is necessary to retify the pixel value according to the camera field of view. For this, the manufacturer provides a document where which pixel as a angle. Comparing this values with the obtained using the calculated field of view, we obtain a new pixel value to analyse.

$$\widehat{p}_{image} = \widehat{R}_{IMU}^{Sensor} \cdot \widehat{R}_{\psi' \theta' \phi'} \cdot \widehat{R}_{IMU}{}^{T} \cdot \widehat{R}_{UTM}^{IMU} \cdot (\widehat{p}_{world} - \widehat{p}_{UAV}) \tag{5.3}$$

Being:

- $\widehat{p}_{image}$ is a vector with the result, $\widehat{p}_{image} = [p_{image_1}, p_{image_2}, p_{image_3}, 0]$ in homogeneous coordinates. However, in order to obtain the image coordinates of the point, this result must be normalized. The image coordinates x_i and y_i are given by:

$$x_i = \frac{p_{image_1}}{p_{image_3}} \cdot (-f) \tag{5.4}$$

$$y_i = \left(\frac{p_{image_2}}{p_{image_3}} \cdot (-f) \right) + 320 \tag{5.5}$$

  Where f corresponds to the focal length in pixels. This value is given by:

$$f = \frac{Number\ of\ pixels \cdot Focal\ Length\ (mm)}{CCD\ Width\ (mm)} \tag{5.6}$$

- $\widehat{R}_{IMU}^{Sensor}$ is the rotation matrix from the IMU reference frame to camera reference frame;
- $\widehat{R}_{\psi' \theta' \phi'}$ corresponds to the matrix that contains the angle between the sensor and IMU in homogeneous coordinates (boresight error). In an ideal situation, this would be a identity matrix. This matrix is equal to equation 3.7, differing only in the angles - in this, case represent the boresight angles;
- $\widehat{R}_{IMU}{}^{T}$ is the matrix that apply the rotations (roll, pitch and yaw) obtained by the IMU. In this case, this matrix is equal to equation 3.7;
- $\widehat{R}_{UTM}^{IMU}$ corresponds to the rotation matrix from the World reference frame to the IMU reference frame;
- $\widehat{p}_{world}$ is the ground control point in UTM coordinates;
- $\widehat{p}_{UAV}$ is the UAV position in UTM coordinates.

This solution returns multiple boresight angle solutions. In order to minimize the value of the obtained angles we implement a Nonlinear least-squares solver, which finds the coefficients that minimized the re-projection error. The equation 5.7 represents this relationship.

$$min_x ||F(x, xdata) - ydata||_2^2 = min_x \sum_i (F(x, xdata_i) - ydata_i)^2 \tag{5.7}$$

Where $xdata$ is the input data and $ydata$ is the observed output data. x will be the vector with the variables that we want to minimize. In the end of this process, we should have minimized values for ψ' , θ' and ϕ'. This values acts as a calibration angles between IMU and sensor, denominated as boresight angles.

5.3 Spectrum Processing

As soon as the calibration is performed, the system can start with the spectral processing. Therefore, as previously stated, it is necessary first to subtract all pixels in the average value of the corresponding dark frame. Only after this step the pixel spectrum can be created, and the data values that are raw data can be converted to radiance spectrum. To do so, the gain and offset values provided by the manufacturer must be applied. The gain must be applied in accordance with the equation 3.13, and for this system the offset is zero.

Now, the data is in radiance and needs to be transformed into reflectance by removing the atmospheric effects. This procedure allows the elimination of almost all atmospheric effects, leaving only the surface materials and some residual amounts of the others components. Figures 5.4 and 5.5 shows the differences between the spectrum of a car in radiance and reflectance, respectively.

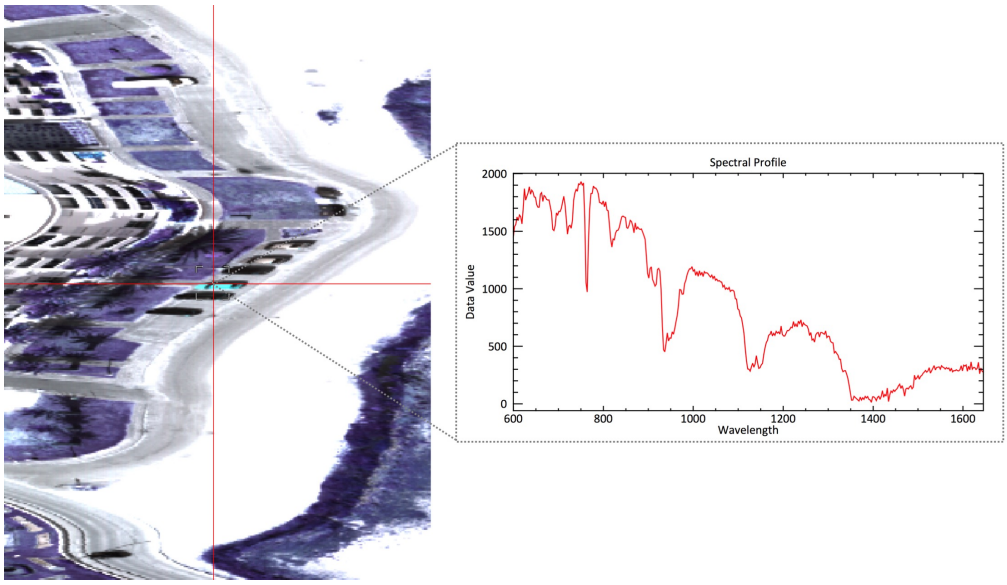


Figure 5.4: *Car spectrum - radiance.*

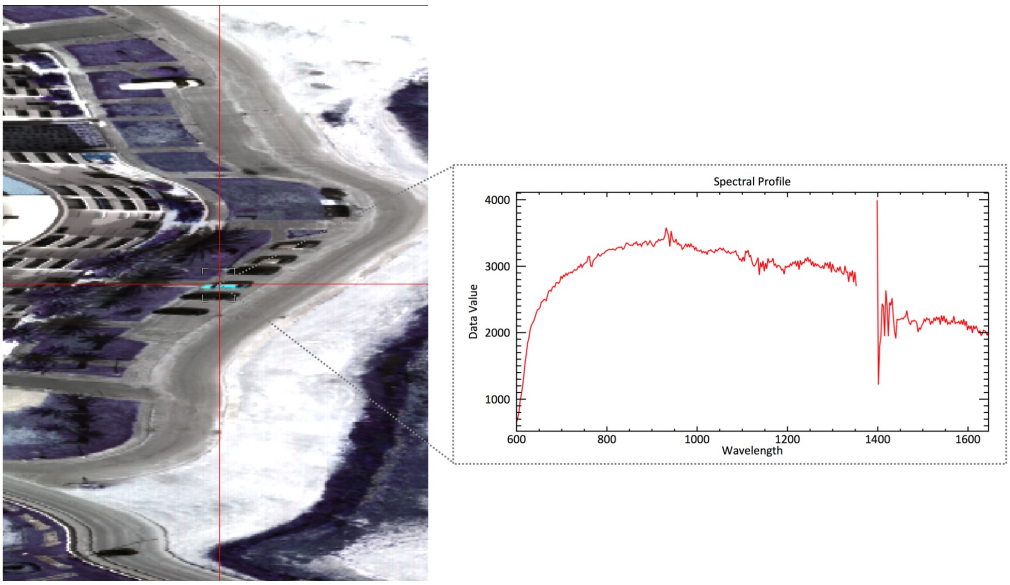


Figure 5.5: *Car spectrum - reflectance.*

However, the atmospheric correction required to transform the radiance into

reflectance requires a high computational power. Moreover, commercial algorithms are required to run in specific operating systems. Our system has to be implemented on a UAV, which due to power limitations reduces the computacional processing available. Therefore, it was necessary to consider other options that allow detecting targets in a maritime environment.

One of these options is to consider only the radiance spectrum. Despite the fact that the radiance spectrum still holds several factors in addition to material, this will vary according to the material spectrum. This makes the X material radiance spectrum different of the Y material radiance spectrum.

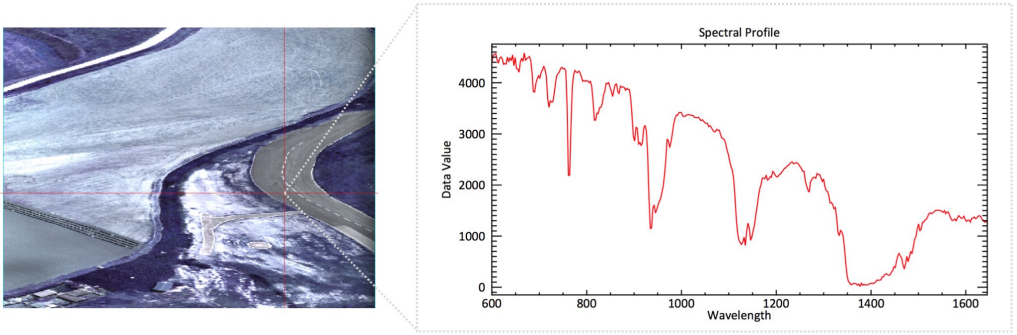


Figure 5.6: *Tar spectrum - radiance.*

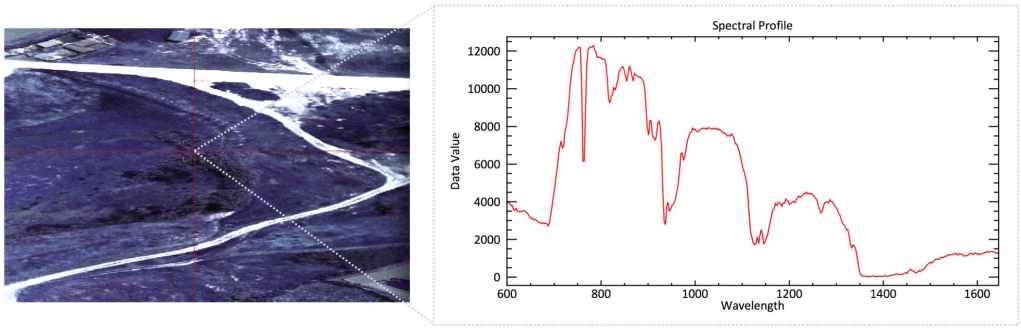


Figure 5.7: *Vegetation spectrum - radiance.*

Figures 5.6 and 5.7 show the spectrum in the field of radiance for tar and vegetation, respectively. There is a significant difference between the two spectra represented in the figures. However, since these are radiance spectrum, they are under atmospheric effects influence. This influence is more pronounced at certain wavelengths, leading to minimum values in this wavelengths, called absorptions. These correspond to the effects of certain gases, which occurs always at the same wavelength of the radiance spectrum. Figure 5.8 shows the effects in three different radiance spectrum.

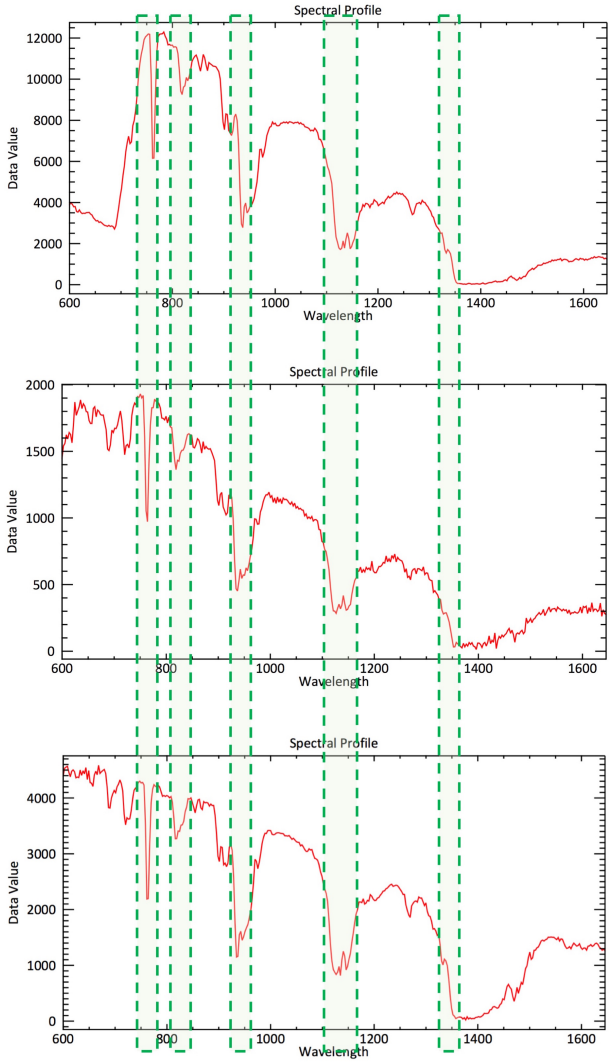


Figure 5.8: *Absorption by atmospheric gases, marked in green.*

However, despite the absorptions, there are enough differences between the spectrum. This proves that it is possible to use radiance spectrum to distinguish between two different materials. Moreover, it is necessary to take into consideration the purpose of this system: the detection of targets in maritime environment. Under these conditions, there will be not much different materials, leaving only the background and the target. Although the water has different phenomena of light reflection, it will be quite different from the target, making it possible to process the radiance spectrum and segment the target from the background.

The implemented method is based on the analysis of the transitions/existing peaks in the spectrum graph. Its based on first and second derivate analysis. First,

it determines number of times that the derivative exceeds a given value. Then, analyzes if the number of times that exceeds this value is greater than a given threshold. If the requirement is meet for both derivatives, it is considered target, otherwise belongs to background. These calculations and comparisons are made to the first and second derivative separately. Figure 5.9 shows a example of the first and second derivative of the background and of the target. In the figures below, is possible to observe the differences between the spectrum of the target and of the background.

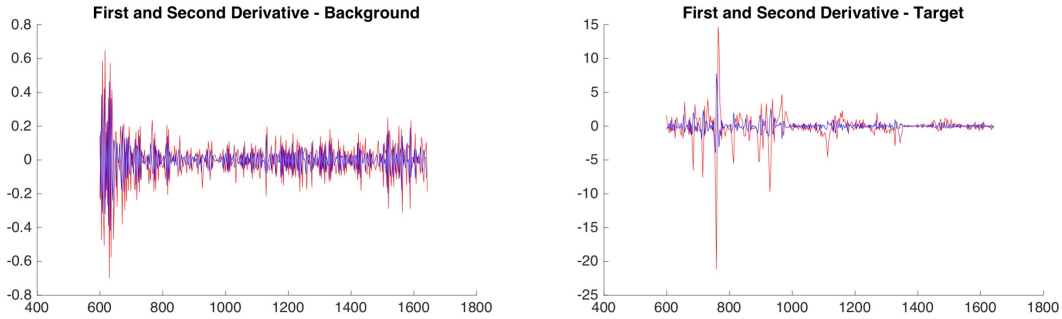


Figure 5.9: *First and second derivative of the background and the target, respectively.*

5.4 Geo-referencing

With this procedures, we can detect the target. However, it is necessary to determine its position in the world, in order to inform the ground station. To obtain its position on the world, is necessary to perform the geo-referenciation of the point obtained.

First, in order to geo-referencing the points, it is necessary to analyze all reference frames involved in the different components of the system.

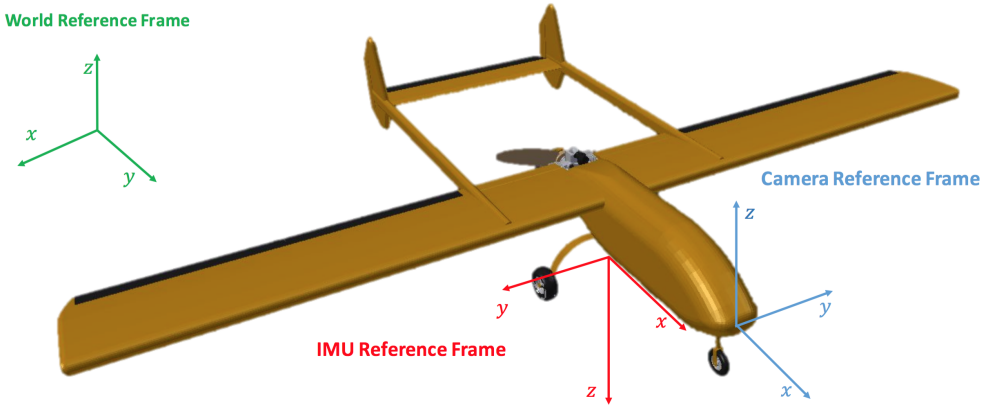


Figure 5.10: *World, camera and IMU reference frames.*

As shown in Figure 5.10, the world coordinates are represented in UTM coordinates, as illustrated in the image by the World Reference Frame. In the case of the IMU, the reference frame considered for roll, pitch and heading angles is the IMU reference frame, which corresponds to the North-East-Down (NED) reference frame. For the sensor, the reference frame is the Camera Reference Frame. Knowing all reference frames involved, and considering the type of camera (pushbroom), we are able to obtain the relation between the object coordinates measured in the image and the real world coordinates, as described in equation 3.19.

However, taking into account the given reference frames, it is necessary to re-set the equation 3.19, as described in the following relation:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}^{UTM}_{Object} = f_{geo}^{UTM} \begin{pmatrix} x \\ y \\ z \end{pmatrix}^{geo}_{Sensor} + s \cdot R_{IMU}^{World} \cdot R_{IMU}\ R_{Sensor}^{IMU} \begin{pmatrix} 0 \\ y_i \\ -f \end{pmatrix}^{Sensor}_{Object} \tag{5.8}$$

Where y_i corresponds to equation 3.15. The difference is the insertion of the R_{IMU}^{World} matrix. This is responsible for the rotation of IMU reference frame to World reference frame, as shown in equation 5.9.

$$R_{IMU}^{World} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{5.9}$$

Another matrix that we didn't have defined previously was R_{Sensor}^{IMU}. This matrix corresponds to the rotation from the sensor reference frame to the IMU reference frame, and is shown in equation 5.10.

$$R_{Sensor}^{IMU} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{5.10}$$

Moreover, it is also necessary to establish the value to assign to the scale, s. This can be deduced using the Ground sampling distance, which corresponds to the resolution that is on the floor. The Ground sampling distance in y is given by the equation 5.11, where δy is the pixel size, and $\frac{h}{f}$ is considered an approximation for the scale (h is the altitude of the UAV, and f is the focal length in pixels);

$$GSP_y = \Delta y \cdot m_b = \Delta y \cdot \frac{h}{f} \tag{5.11}$$

5.5 Data Processing System Software

In Figure 4.2 the processing pipeline starts with the blocks corresponding to the hardware. These represent the data sending from the sensors, which are received in the Data Processing CPU according to the messages defined in chapter 5.1.1.

The next step is the dark frame acquisition in an offline procedure. As already mentioned, it is necessary to acquire some dark frames (typically between 100 and 200 frames), and subsequently calculate the mean of all dark frames obtained for each pixel. With this, we obtain a value of dark frame gain for each pixel, which can be used in real time. This step must always be performed at the beginning of the flight, in order to capture the characteristics of the atmosphere in that day.

Now we can start the real-time processing, and it begins with the application of the dark frame gain. Each value received is subtracted from the corresponding dark frame gain, for each pixel. After this step, the detection of targets begins. For this, it is necessary to calibrate the threshold. In parallel, the interpolation of the GPS/IMU data is performed, so that each line has a position and attitude.

After detecting an target, the information is sent to a ground station, so it can be reviewed by a human. However, the amount of data to be transmitted is considerable. Therefore, it's necessary to perform a crop of the target. This crop is based on a mini hyperspectral cube, with all the spectral signatures of the target and some of the background. Figure 5.11 shows a target crop example.

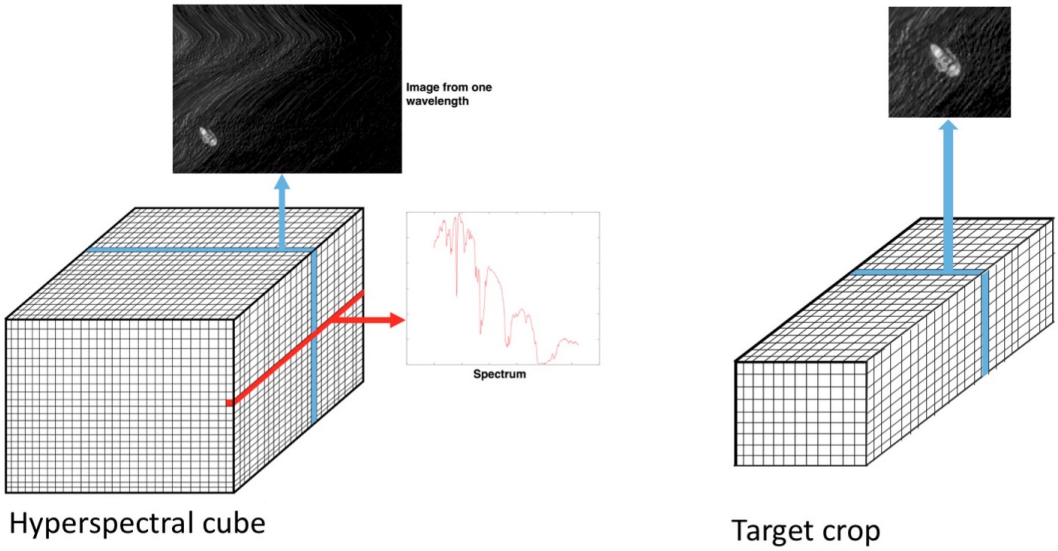


Figure 5.11: *Target crop example.*

The algorithm is described in the algorithm 1 flow.

Algorithm 1: Hyperspectral real-time processing

Input : 1 Hyperspectral frame and GPS/IMU data

Output: Target world position and crop

Step 1. Interpolate the GPS/IMU data

Step 2. Apply the dark frame gain to each pixel

Step 3. Convert the raw data into radiance spectrum

Step 4. Calculate the first and second derivative of the radiance spectrum

Step 5. Threshold calibration

Step 6. Target detection

Step 7. Target geo-referencing, Equation 5.8

Step 8. Create a crop of the target

Step 9. Send information to ground station

This algorithm runs in Data processing CPU, with the Ubuntu 14.04 LTS operating system. We use ROS, to create the acquisition and processing nodes. There are two different processing types: image processing and GPS/IMU processing that are divided into different nodes that run independently.

5.6 Summary

In this chapter, we present the detail implementation of our system, containing the data acquisition modules, the boresight calibration, the spectrum processing analysis and the target geo-reference.

Finally, all blocks are incorporated into a single system, creating the Data Processing System Software. Its allows the hyperspectral real-time processing, in order to detect targets and obtain their position in the world.

6

Results

In this chapter, we will present the results for the hyperspectral imaging for real-time maritime target detection using an unmanned aerial vehicle. The procedure implies to determine the boresight angles to relate the Unmanned Aerial Vehicle navigation data obtain from the GPS/IMU with the imaging data. Afterwards, we have obtain the boresight angles, we can based on signal processing analysis to determine non-water/water segments and keep the crop of the boats.

To validate our proposed solution, we installed the experimental setup, described in 6.1 into a Portuguese Air Force Unmanned Aerial Vehicle. Afterwards, we perform several flight tests at OTA Air Force Base for collecting synchronized data, and perform boresight calibration using the previously described offline procedure.

The boresight calibration procedure consists on the use of multiple ground control points, that were obtained using a precision RTK/GPS Septentrio receiver. For this procedure we used the cross-walk at the beginning of the OTA Air Force runway.

After the boresight calibration, the system was tested in an actual maritime surveillance scenario in Santa Cruz, Peniche. The objective of these flight trials was to test the detection and geo-referencing maritime targets using hyperspectral imaging data.

6.1 Experimental Setup

The experimental setup was installed in a fixed-wing Portuguese Aur Force Unmanned Aerial Vehicle, called ANTEX display in Figure 6.1.

Figure 6.1: *ANTEX UAV.*

The setup used was composed by the following components:

- CPU unit (Intel NUC ML100G-30 i5 2.30 GHz): used to record the E/O camera images, IR images and AIS transmitted messages;
- E/O camera (Point Grey Blackfly 3.2 MP Color GigE PoE Sony IMX265): acquires 24-bit RGB images, with synchronized timestamp;
- IR camera (Xenics Gobi-640-CL): acquires images in the infrared spectrum, 640x480 resolution, with synchronized timestamp;
- Frame Grabber (Pleora CL-U3): acquires frames from the infrared camera via Camera Link interface, making them accessible to the processing unit by USB 3.0;
- AIS (Yacht AIS100PRO): receive data from the transmitter AIS installed on vessels;
- Processing unit (SPECIM DPU): for data acquisition of the hyperspectral camera, inertial navigation and GPS information;
- Hyperspectral camera (SPECIM AisaKESTREL 16): acquires data in the electromagnetic spectrum from ultraviolet to long-infrared;
- Inertial and GPS (OXTS xNav550): dual GNSS receivers and a Inertial measurement unit;
- Wi-fi communications: 700 MHz module, for ground UAV payload communications. Used for start/stop recording and debug purposes;

- External antennas (antenna Wi-Fi, antenna GPS and antenna AIS);
- Power supply: LifePO4 battery and DC/DC for regulated voltage.

In Figure 6.2 is possible to see the payload wiring diagram where all sensors and equipments wiring connections are illustrated. Our system consists of two processing units. The NUC PC runs a Ubuntu 14.04 operating system environment with the Robotic Operating System(ROS) as middleware framework. This equipment is used to acquire and process images from the E/O camera, IR Camera, and also from the AIS receiver. It also records Inertial Measurements from the IMU unit via Ethernet connection for geo-reference all the data to the UAV position. In parallel with this setup, there is the SPECIM system.

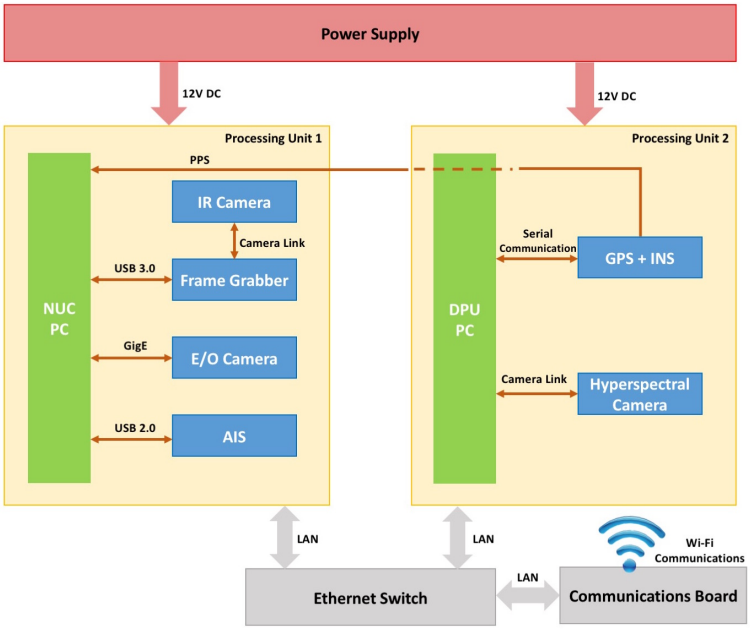


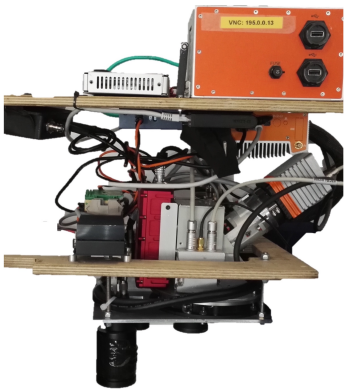
Figure 6.2: *Payload Wiring diagram.*

Figure 6.3, shows the payload as it is assembled and mounted inside the UAV. The sensors are all placed in the bottom structure so that all the cameras can be placed near/outside the bottom fuselage of the UAV. While the data recording CPUs are placed on top layer.

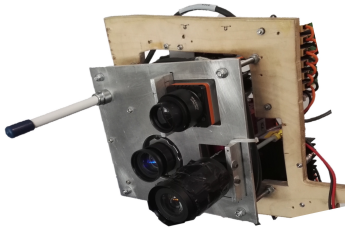
6.2 Dataset I

The first dataset campaign occur in OTA Air Force Base, Alenquer. OTA Air Force Base is the primary center for Portuguese Air Force flight trials. The ANTEX UAV was the aircraft used to carry the sensor payload in the field tests.

(a) Side view of the payload, with imaging sensors and IMU unit.



(b) Side view of the full payload.



(c) Bottom view of the payload, showing all imaging sensors and the AIS antenna.

Figure 6.3: *UAV payload used in the data collection trials.*

The Antex UAV performed a pre-determined flight trajectory, with the objective of flying in different directions over the crosswalk at the end of the runway, to collect image data of the crosswalk. In Figure 6.4 is represented the UAV flight path.

Figure 6.4: *UAV ANTEX flight path during dataset I.*

In addition, we also collected the control points with a RTK GPS sensor. These points were collected at the runway. In Figure 6.5 the control points are displayed.

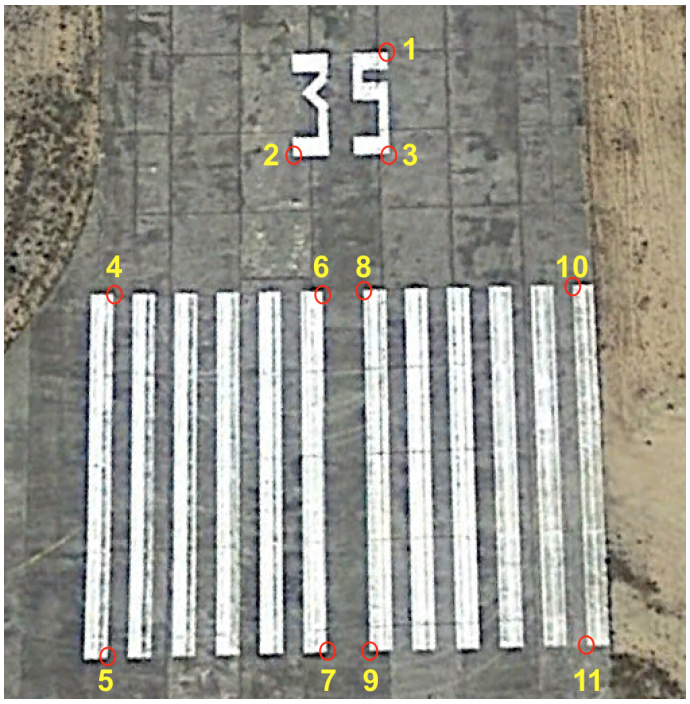


Figure 6.5: *Ground control points collected at OTA Air Force Base runway.*

6.2.1 Results

Since it was the first time we obtained data with hyperspectral camera, this was mainly used for calibration purposes.

In this dataset, we used four passages over the runway. Figure 6.6 shows the results for the first, second, third and fourth passage, without considering the boresight angles calibration.

Finally, Figure 6.6 show the results considering the boresight angles. The boresight angles have the following values: ψ' = 0.0401°, θ' = 0.6474° and ϕ' = -1.7074°.

In the Table 6.1 we find the mean and standard deviation for the results (considering absolute values), summarizing the results for each pass, with and without the application of the boresight angles.

As can be seen, the difference between the results with and without the boresight angles is more obvious in the case of the first two passages. In the case of the latter two, the results become worse slightly. However, the boresight error in heading is higher than 1°, not being the expected value. This is reflected in reported errors, which are still slightly higher than expected. The reason for this error may be due

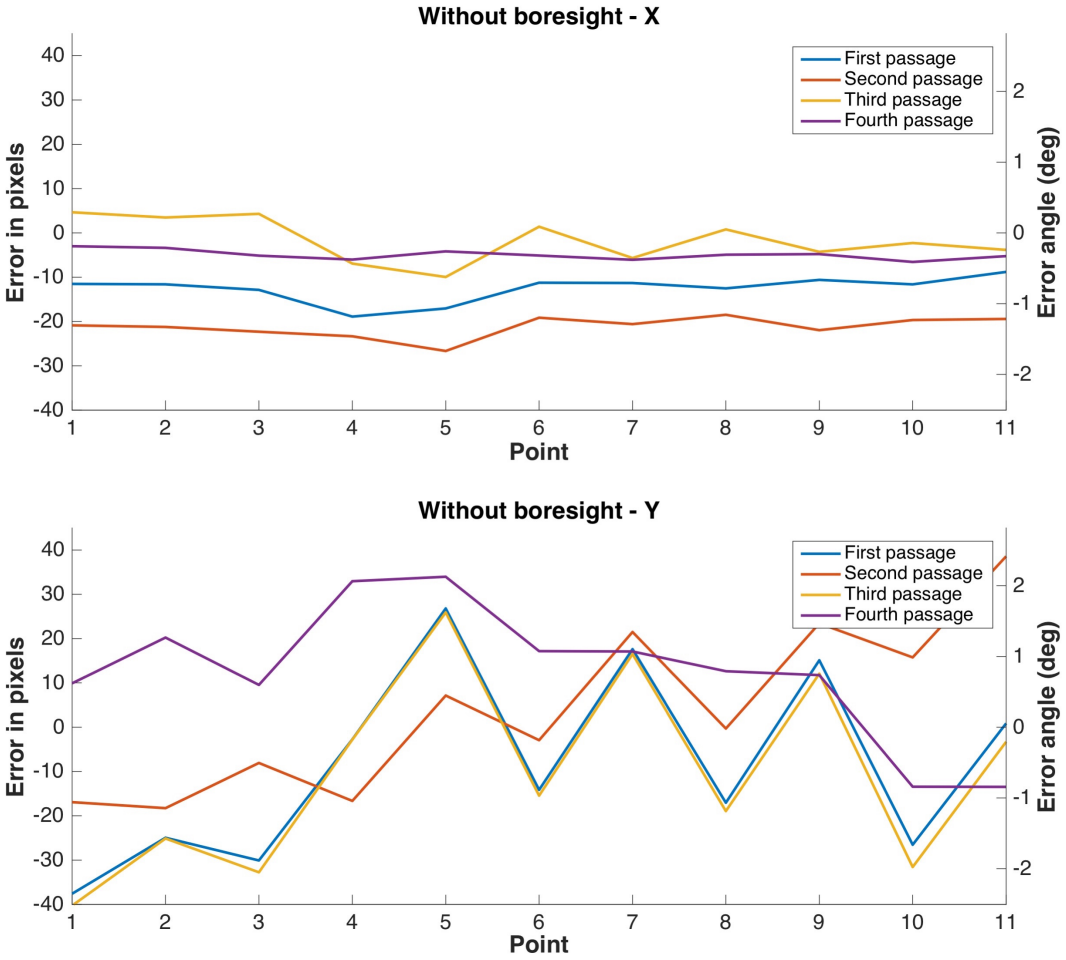


Figure 6.6: *Error in X and Y for all passages, without considering the boresight error.*

Table 6.1: *Mean and standard deviation in pixels for the dataset I results.*

		Without Boresight		**With Boresight**	
		X	Y	X	Y
Mean	1	12.5411	19.4099	4.8105	19.2936
	2	21.2426	15.4158	8.7672	15.4060
	3	4.3220	20.4284	7.3475	20.0487
	4	4.9372	17.4698	6.2404	18.0224
Standard Deviation	1	2.9102	11.2134	2.7170	11.0087
	2	2.3179	10.7503	4.1913	10.9720
	3	2.5777	11.9913	3.2750	12.0073
	4	1.0974	8.5323	2.5254	8.7150

to problems with the GPS/IMU, since the antenna can't be positioned outside of the UAV during this flight.

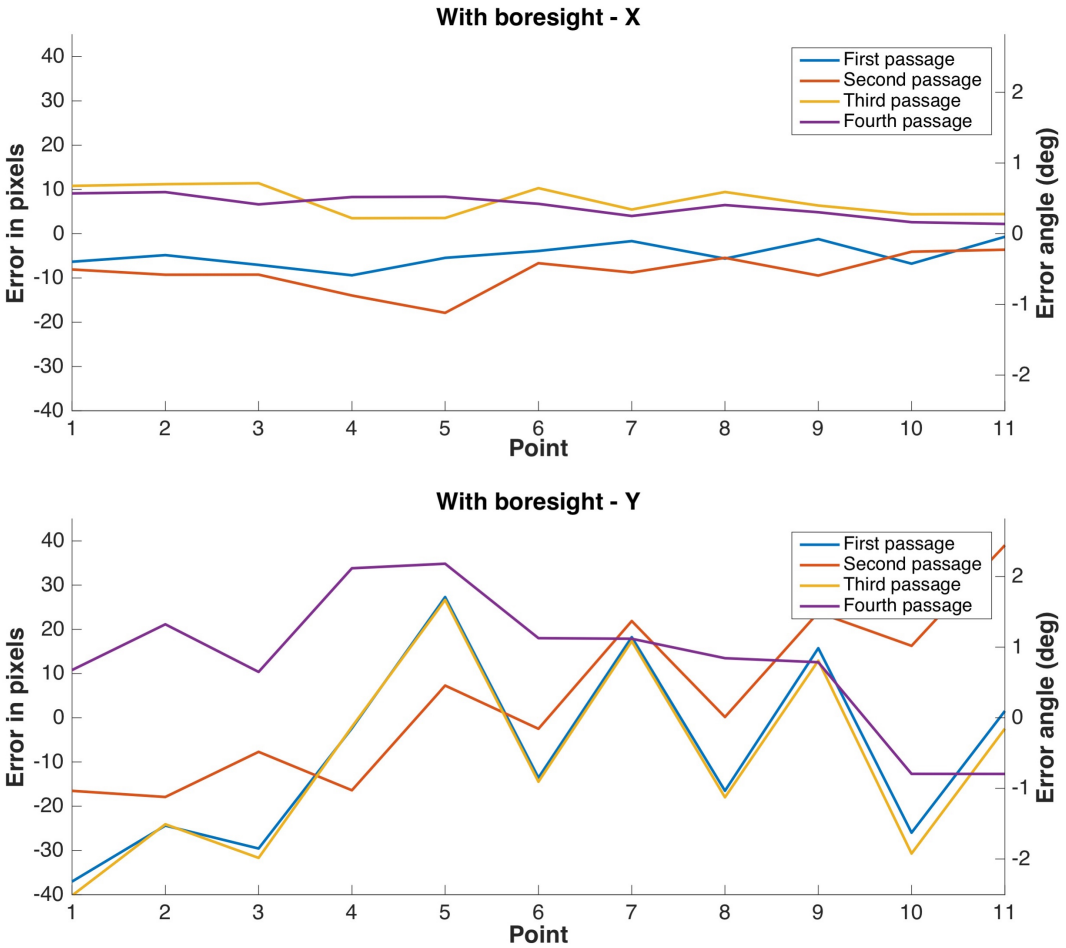


Figure 6.7: *Error in X and Y for all passages, considering the boresight error.*

6.3 Dataset II

Given the results of the first flight, there was a need for one more flight to the boresight calibration. The second dataset was also held at the OTA Air Force Base, with a trajectory similar to the previous. The flight path is shown in Figure 6.8.

Figure 6.8: *UAV ANTEX flight path during dataset II.*

The purpose of this flight was only the boresight calibration. Thus, there were two tests: one with the data from the GPS/IMU obtained during the flight, and other with post-processed data, obtained using a manufacturer's software, called RT Post-process[1]. This software which allows to process, analyze and explore data. The only input required for RT Post-process is a file extracted directly from the GPS / IMU.

For both types of data (coming directly from the GPS/IMU and post-processed) we measured the error between the GPS/IMU and camera. It followed the process already shown in chapter 5.2. The ground control points used are shown in Figure 6.5.

6.3.1 Results: Data GPS/IMU obtained during flight

First we analyze the results using the data obtained during the flight. These is the data available in real time, in order to perform georeferencing of points.

Figure 6.9 shows the heading, roll and pitch angles variation during the flight.

Figure 6.10 show the error in X and Y when the angles of boresight are not considered. The results are always analyzed for each pass through the runway.

Then, we consider the boresight angles. These were calculated with the data obtained having the following values:

[1]http://www.oxts.com/rt-post-process-support-for-multiple-base-stations-improves-productivity/ accessed on 08/03/2016

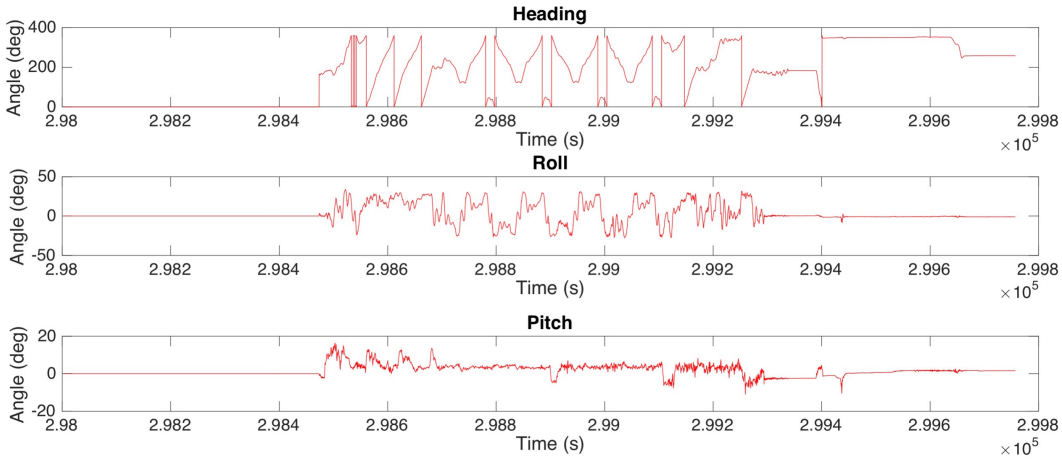


Figure 6.9: *Heading, roll and pitch angles variation during flight - GPS/IMU obtained during flight.*

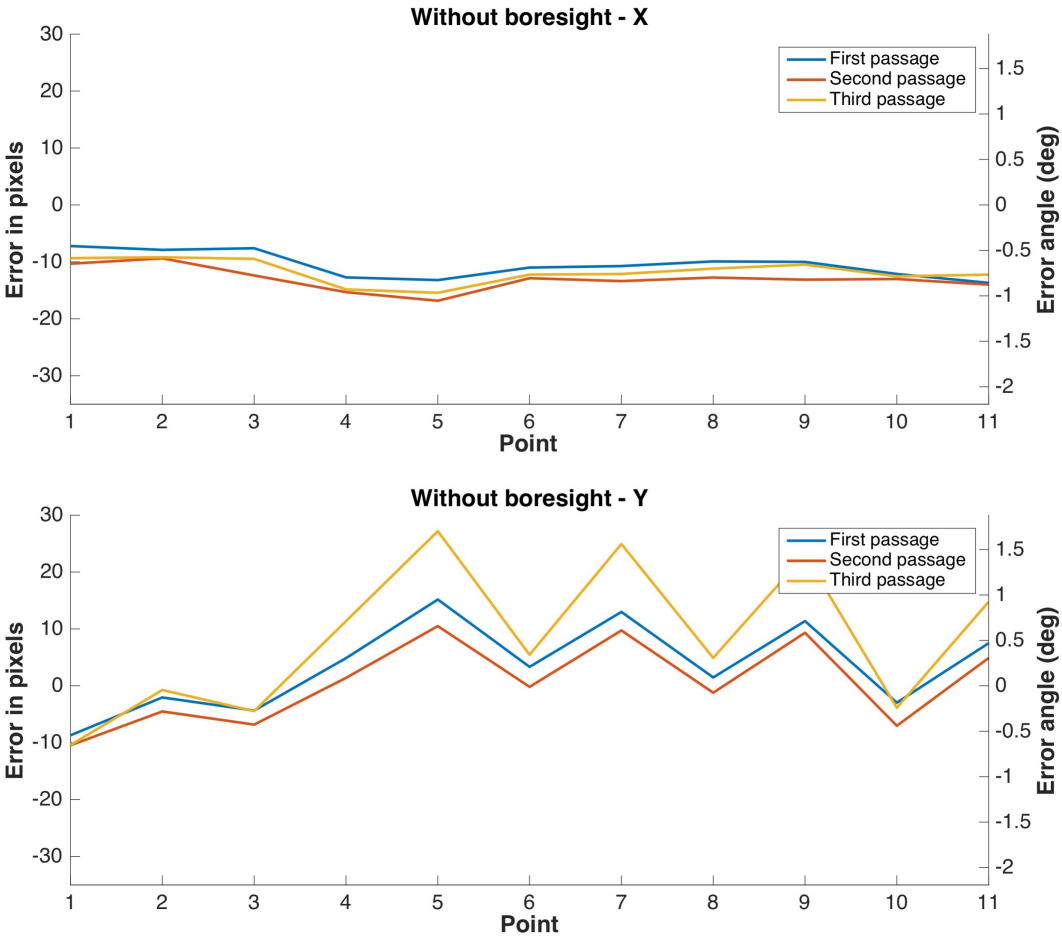


Figure 6.10: *Error in X and Y for all passages, without considering the boresight error.*

- ψ' = -0.2416°
- θ' = 0.6660°
- ϕ' = -1.3151°

Thus, in Figure 6.11 is possible to observe the errors in X and Y considering the boresight values.

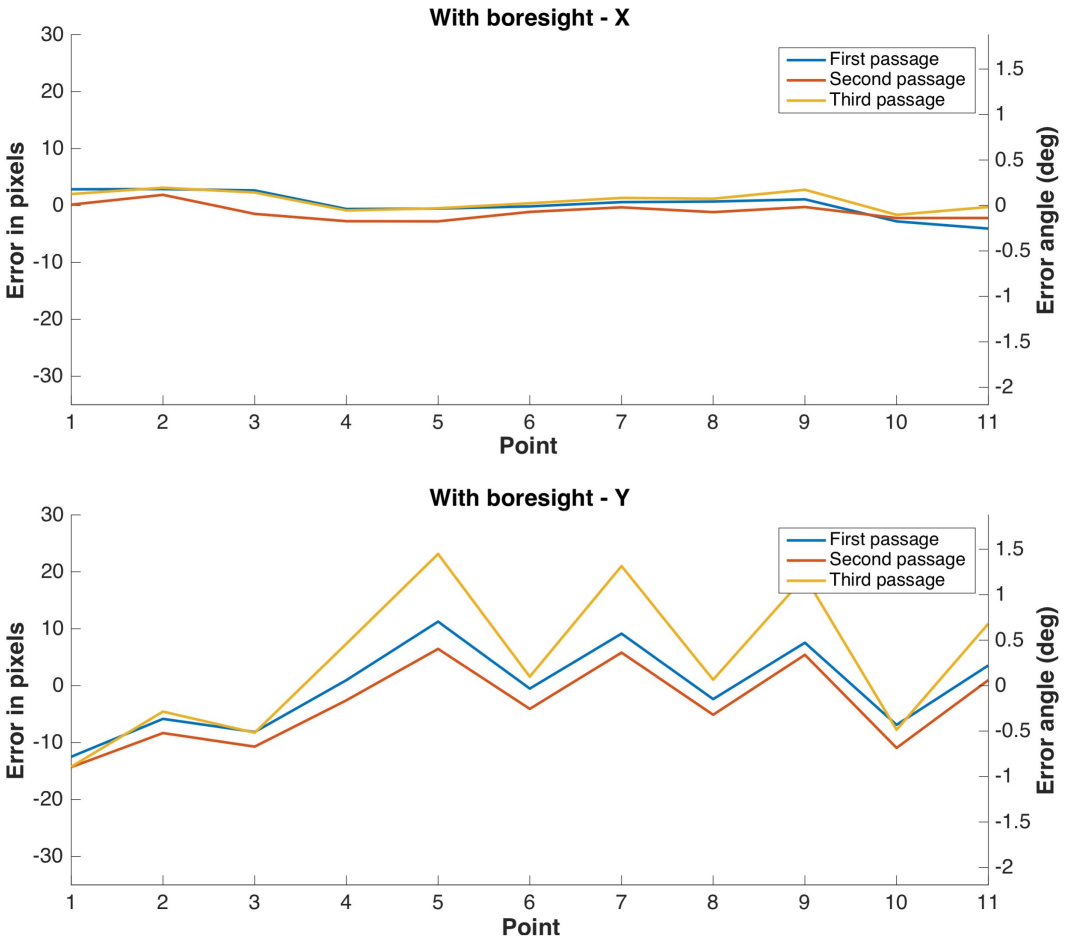


Figure 6.11: *Error in X and Y for all passages, considering the boresight error.*

Thus, in the Table 6.2 is possible to observe the mean and standard deviation (considering absolute value) of the error for each pass, considering that the boresight angles are zero, and considering the values obtained for boresight error. It is noteworthy that, for all passages the error values are lower when the angles of boresight are considered.

However, as can be seen in Figure 6.11, even with the boresight application, the error in Y had no significant improvements compared with the results without

Table 6.2: *Mean and standard deviation in pixels of the results with the data obtained during flight.*

		Without Boresight		With Boresight	
		X	Y	X	Y
Mean	1	10.5526	6.7979	1.7189	6.2458
	2	13.0293	6.0064	1.4862	6.7934
	3	11.7305	11.9177	1.4862	10.8155
Standard Deviation	1	2.2756	4.7016	1.3362	4.0150
	2	2.0468	3.8516	0.9663	3.9529
	3	2.0916	9.3422	0.9693	7.6452

boresight. This may be of timing errors due to the synchronization method between the hyperspectral camera and the GPS/IMU, the error means 1.5°, which is the expected value for this system. The boresight calibration represents a 86.72% improvement in X and 3.55% in Y, compared with the results without the application of the calibration.

6.3.2 Results: Data GPS/IMU post processed

After the flight, the data obtained by the GPS/IMU was processed, allowing a new analysis of the boresight angles. Figure 6.12 shows the heading, roll and pitch angles variation during the flight.

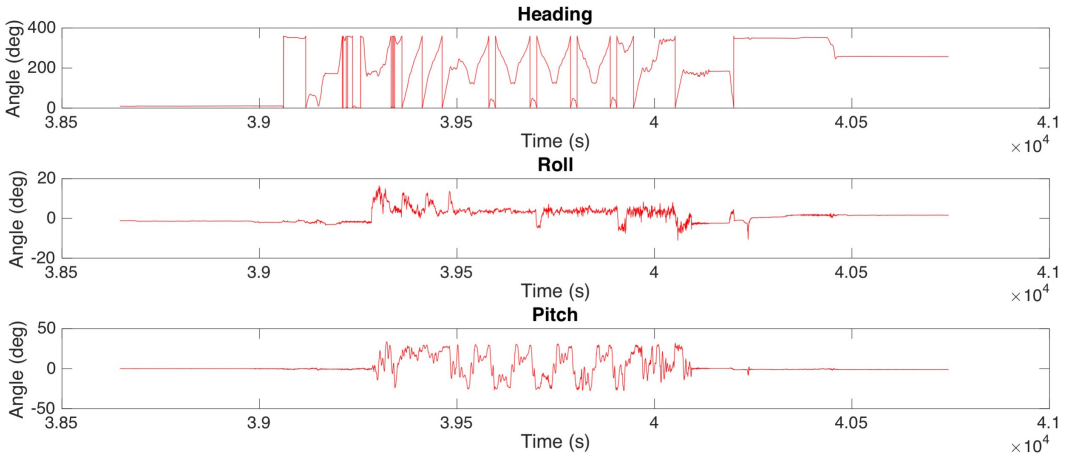


Figure 6.12: *Heading, roll and pitch angles variation during flight - GPS/IMU post processed.*

First it was considered that these had a value of 0, as shown in Figure 6.13 for X and Y.

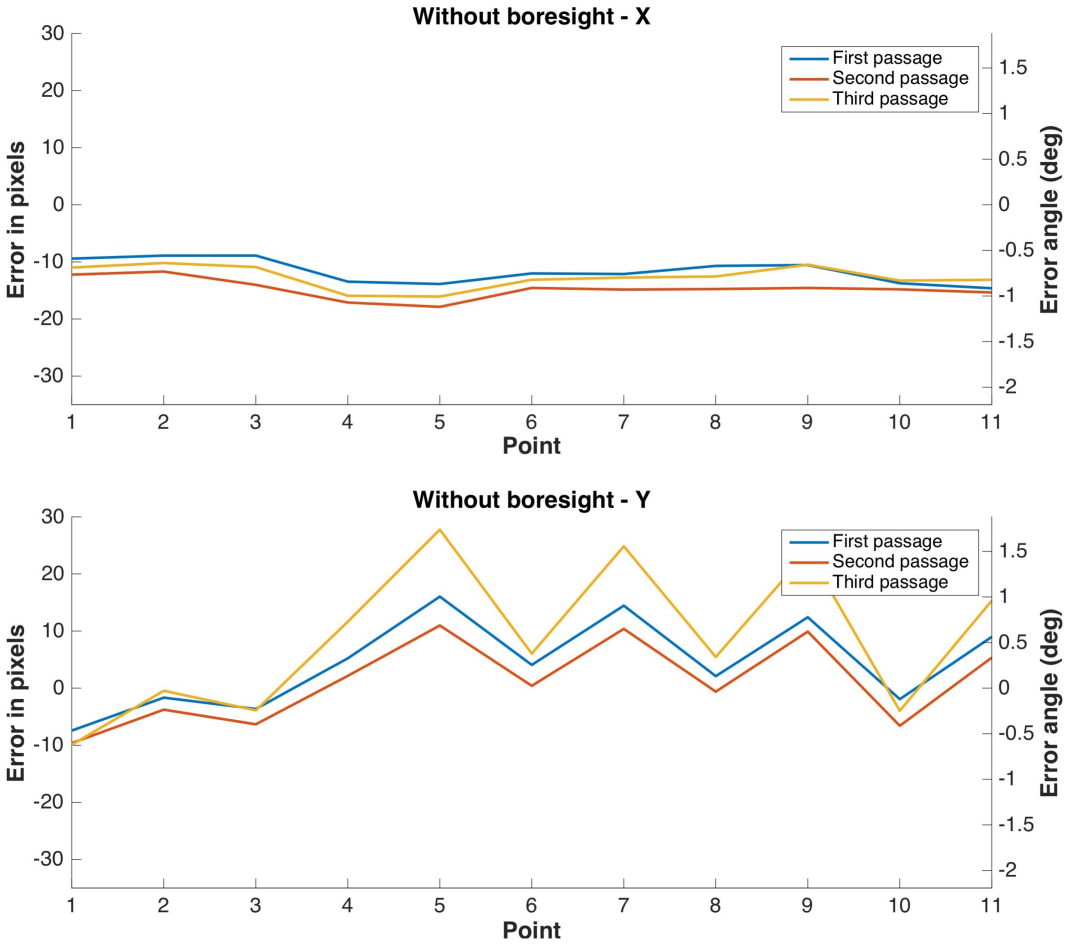


Figure 6.13: *Error in X and Y for all passages, without considering the boresight error.*

Then, we computed the boresight angles, obtaining the following values:

- ψ' = -0.2815°
- θ' = 0.7567°
- ϕ' = -1.0762°

In Figure 6.14 is possible to observe the errors in X and Y considering the boresight angles.

Finally, in Table 6.3 are presented the means and standard deviations (considering absolute values) for all passes without and with boresight angles. As in the previous case, there was a significant improvement in almost all passages when applying the obtained boresight angles.

As can be seen in Tables 6.2 and 6.3, there is a significant difference between the results obtained with the GPS/IMU data acquired during flight and post processed

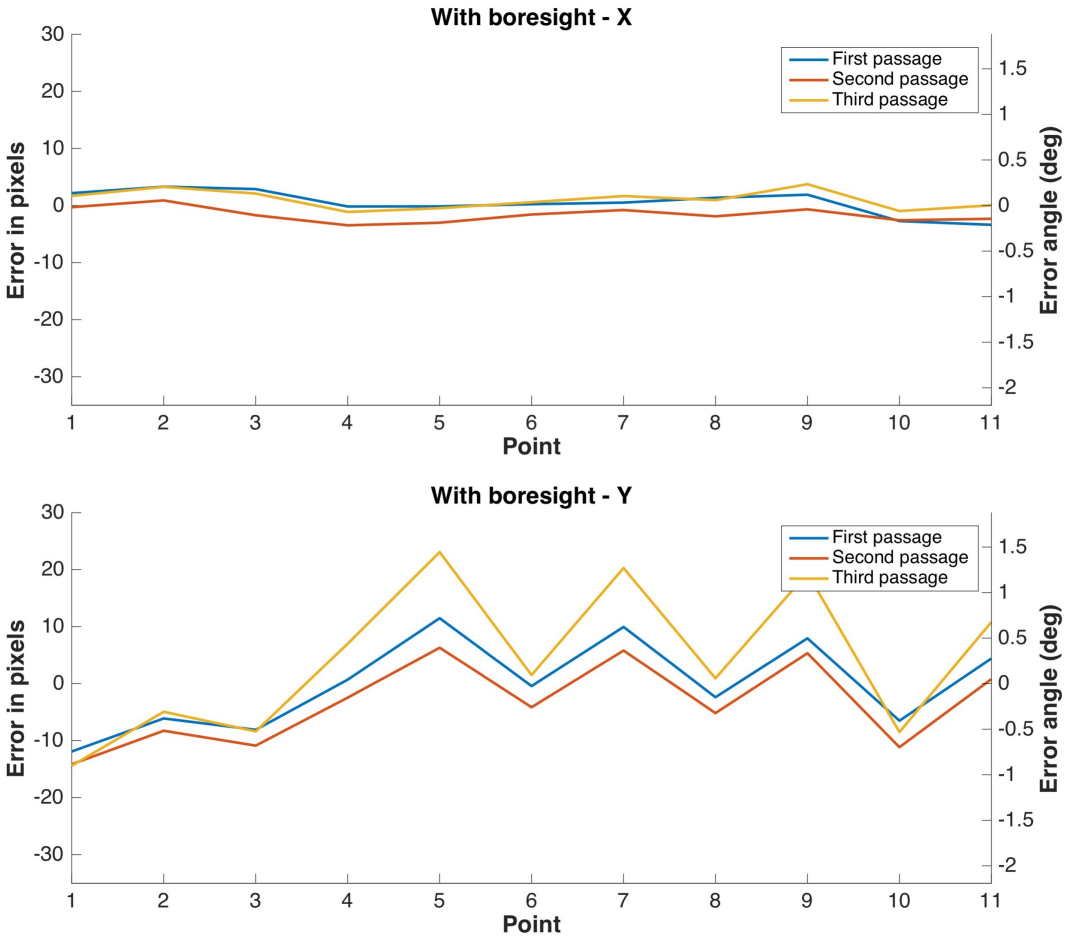


Figure 6.14: *Error in X and Y for all passages, considering the boresight error.*

Table 6.3: *Mean and standard deviation in pixels of the results with post processed data.*

		Without Boresight		**With Boresight**	
		X	Y	X	Y
Mean	1	11.6609	7.0872	1.7063	6.3552
	2	14.7075	6.0099	1.7435	6.7627
	3	12.6692	12.1100	1.5054	10.8360
Standard Deviation	1	2.0963	5.2117	1.2861	4.0287
	2	1.7905	3.9152	1.0282	3.9904
	3	1.9962	9.5496	1.1544	7.5547

for the heading. This is because the sensor used is Attitude and Heading Reference System (AHRS) type. This means that the returned attitude is a fusion of data from gyroscopes, accelerometers and Magnetometers, being applied to a Kalman filter to predict the evolution of the system. So, the GPS/IMU may return incorrect data,

especially in the heading. The boresight calibration represents a 87.31% in X and 4.97% in Y improvement compared with the results without the application of the calibration

Finally, we present the points projected in the ground, without and with the boresight angles. This is shown in Figure 6.15

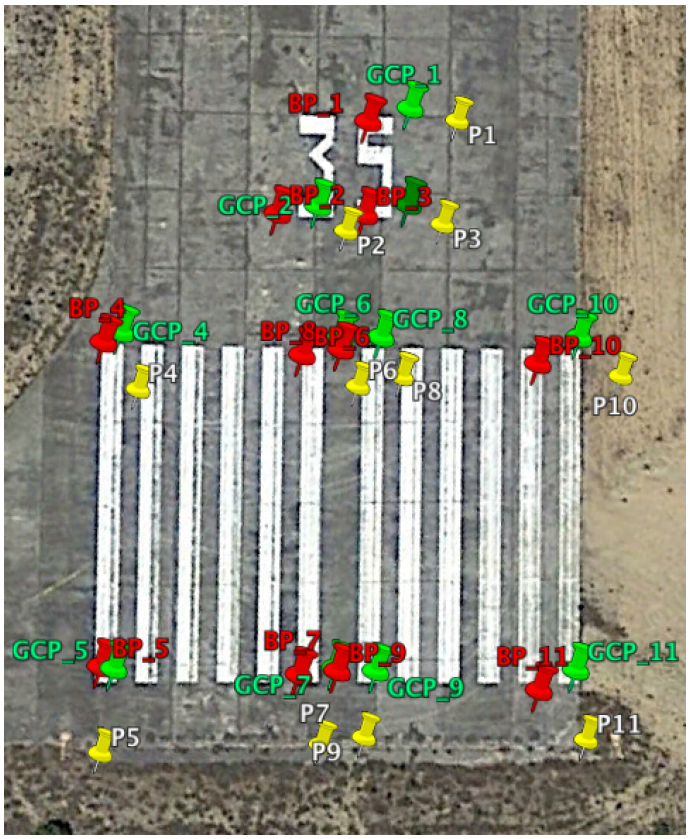


Figure 6.15: *Points obtained without and with boresight application projected to the ground. Green represents the ground control points, yellow are the points without boresight. In red we have the points projected considering the boresights angles.*

6.4 Dataset III

The third dataset campaign was Performed at Santa Cruz Airfield, Peniche. This airfield is located in a coastal area, allowing the UAV quickly head out to sea and can fly over boats. Therefore, we used the vessel lying in Figure 6.16. This carry a RTK GPS sensor on board, allowing to save the path over time. Comparing the timestamp of each boat location with the timestamp of each image, we are able to get the ground-truth of the boat. Thus, it is possible to determine the error of the determined position to the target in a real scenario.

Figure 6.16: *ISN Vigilante rescue vessel.*

The ANTEX UAV flight trajectory and the vessel ground-truth position information are displayed in Figure 6.17.

In Figure 6.18 is a part of the perform scan during the flight by the hyperspectral camera on the ground. These points are projected into the ground, allowing to demonstrate the scanning area projected on the ground.

6.4.1 Results

Since these datasets was very close to real scenario where the system must operate, the collected data was used to validate the steps performed in the processing pipeline. So, for this dataset are shown two types of results:

- Boat detection and spectrum analysis of this against the background;
- Target georeferencing and comparison with the boat's position.

The UAV flew over the boat on three separate occasions. Initially, the spectrum obtained was analyzed, together with the EO camera image. To analyze the spectrum, we choose to compare two pixels for each detection: a target pixel, and one

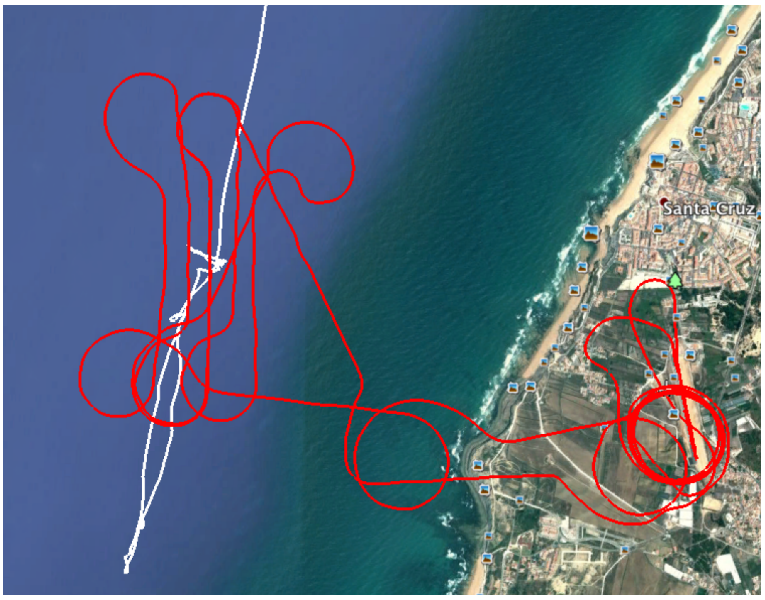


Figure 6.17: *In red we can see ANTEX UAV flight trajectory during dataset II, and in white is the vessel ground-truth position information.*

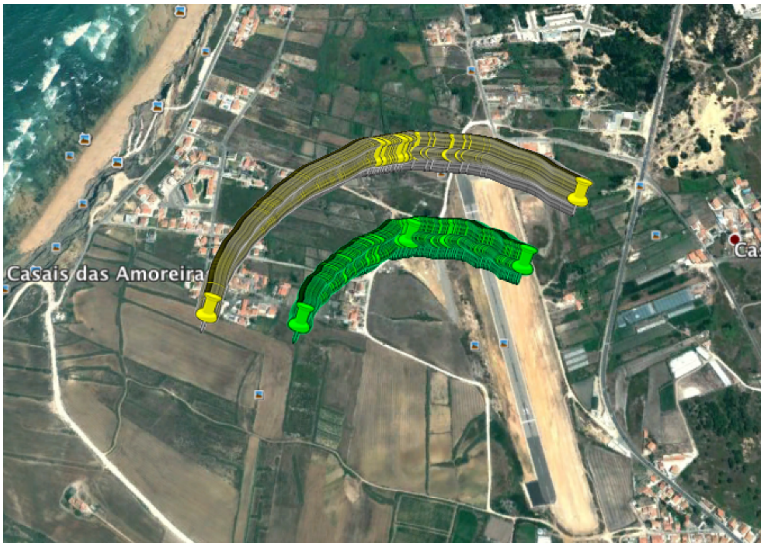


Figure 6.18: *Hyperspectral camera scan.*

pixel in the background. Thus, it is possible to see the differences between these spectra, and prove that it is possible to distinguish a target from the background in the system operating environment.

In order to verify the results, we used the ENVI software. The acquired data was displayed with this software, which led to the spectra analysis for the two situations. We also present a comparison with the EO camera image.

The results for the three detections are presented in the following order:

1. Comparison between the image obtained with hyperspectral camera and with the image obtained by the EO camera. In the image obtained by hyperspectral it is also shown the detection target against the background;
2. Spectrum analysis - background *versus* target;
3. First and second derivate of the spectrum analysis;
4. Presentation of the spectra obtained using ENVI;
5. georeferencing of the targets pixels.

Detection 1

Following the defined presentation, we started by showing a picture obtained with hyperspectral imaging and other by EO camera in Figure 6.19. The hyperspectral camera image was obtained using only a band located at a 651.77 nm wavelength.

This image also shows the direction of the boat, with all pixels being considered as targets represented in red.

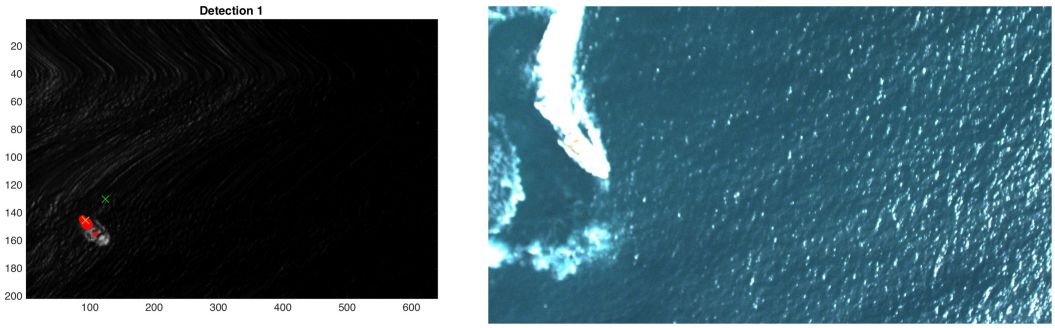


Figure 6.19: *Detection 1: Hyperspectral camera versus EO Camera. The yellow cross represents the point analysed to obtain the spectrum of the boat, while the green cross mark the point at what was obtained the spectrum of water. The points in red represent the target detection in relation to the background.*

With the data it was possible to analyze their radiance spectra. To do this, we used the MATLAB, to obtain a water spectrum and compare it with boat spectrum, as shown in Figure 6.20. Both spectra were obtained from the previously relationship, which allow passing the data obtained to radiance.

Then in Figure 6.21, the first and second derivatives of the two spectra obtained were analyzed, to check the differences between them, especially regarding their maximum value. In the case of the water spectrum first derivative, the maximum

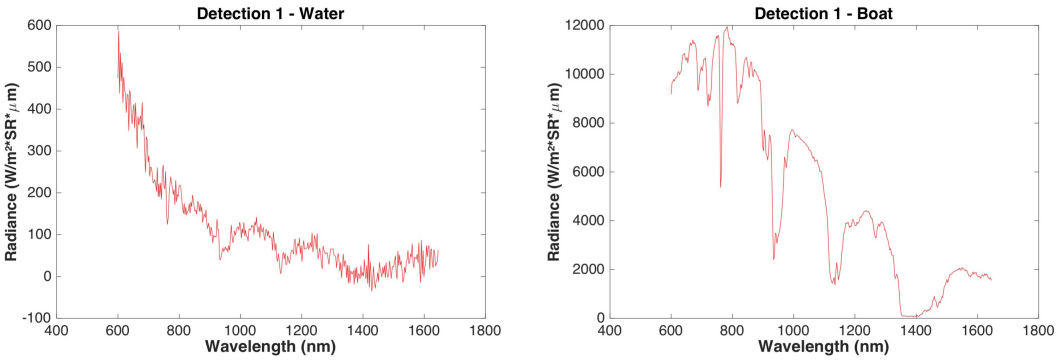


Figure 6.20: *Detection 1: Matlab radiance spectrum.*

value is about 0.4, and the minimum -0.5. However, in the case of target, the maximum value is greater than 10, and its minimum is very close to -15.

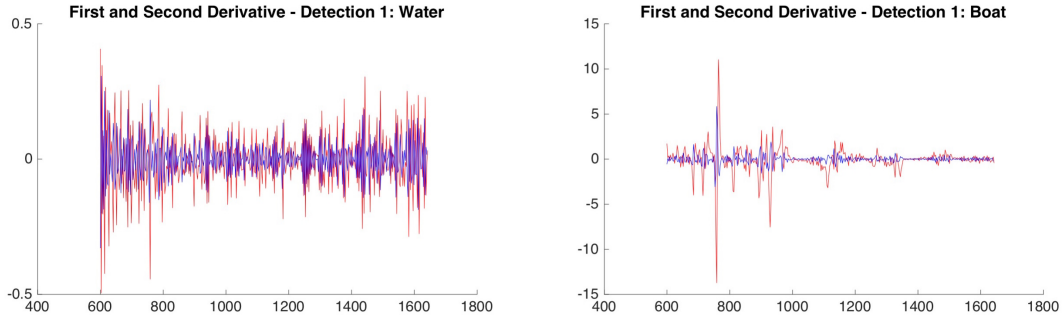


Figure 6.21: *Detection 1: First and second derivative of the spectra obtained.*

Finally, we present the spectra obtained with ENVI software, which serves as validation of the results obtained with Matlab. These spectra are shown in Figure 6.22, allowing to verify the obtained results.

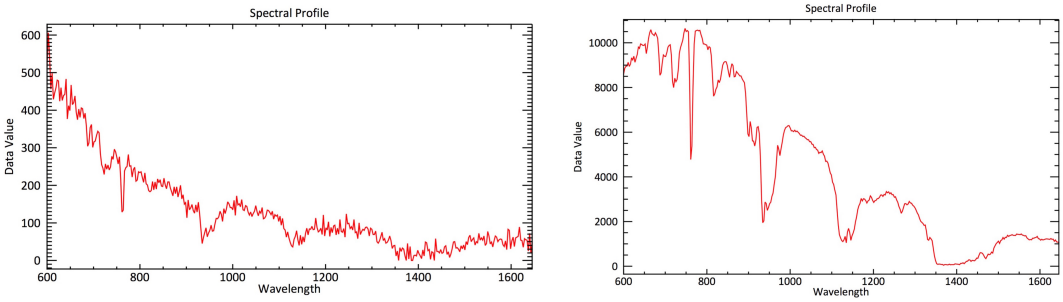


Figure 6.22: *Detection 1: Spectra obtained with ENVI software.*

Moreover, with the results obtained, it is still possible to observe that the difference between the target spectrum and the background is evident, especially as

regards the maximum that each of them can reach.

Finally, it is necessary to analyze the target projection, in order to obtain the point in the world. In the case of this detection, the difference between the point obtained by RTK/GPS on the boat and the projected point is about 13 meters.

Detection 2

For the detection 2 analysis, we followed the same steps applied to the first detection. Thus, in Figure 6.23 is shown a picture of the hyperspectral camera and the EO camera. As previously, in red are identified all pixels that are considered as targets. We also represented the pixels where the spectra were obtained (target and background).

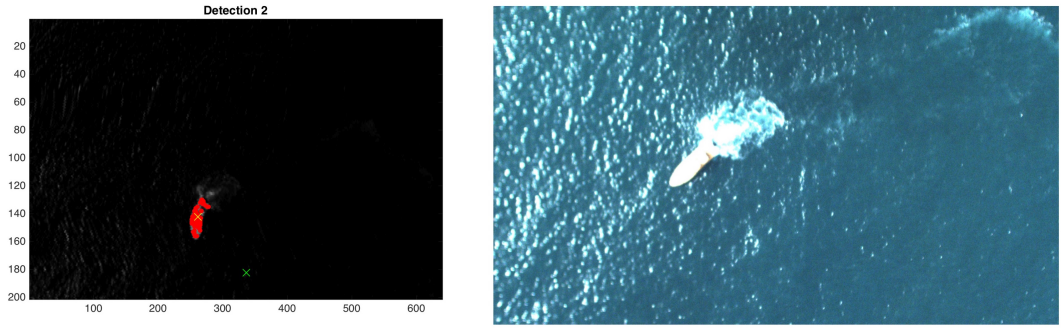


Figure 6.23: *Detection 2: Hyperspectral camera versus EO Camera. The yellow cross represents the point analysed to obtain the spectrum of the boat, while the green cross mark the point where was obtained the spectrum of the water.*

Figure 6.24 presents the spectral radiance obtained for the previously selected two points corresponding to the background and target. These were obtained using MATLAB, demonstrating the obvious differences between the radiance spectra of the background and target.

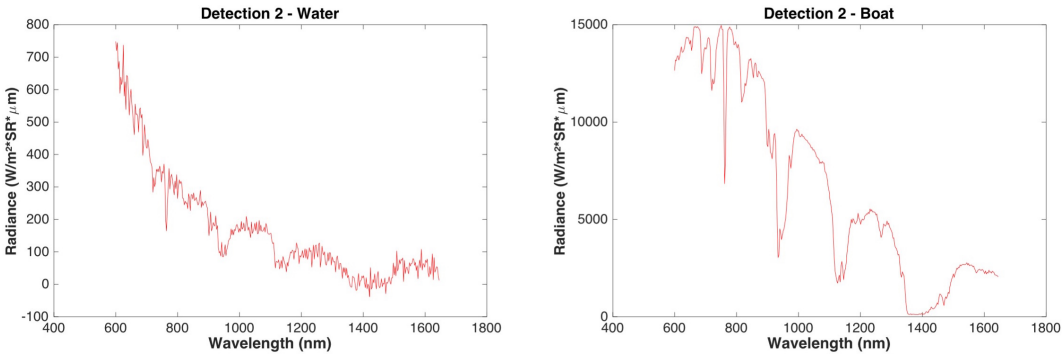


Figure 6.24: *Detection 2: Radiance spectra obtained with Matlab.*

Figure 6.25 shows the first and second derivative of the target and background specta. As in the previous situation, they are quite different, especially with regard to their maximum and minimum value. In the case of the background, considering the first derivative, its maximum value is 0.4, and the minimum -0.5. In the case of the target spectrum, the first derivative has it maximum in a value greater than 20, while its minimum is very close to -30.

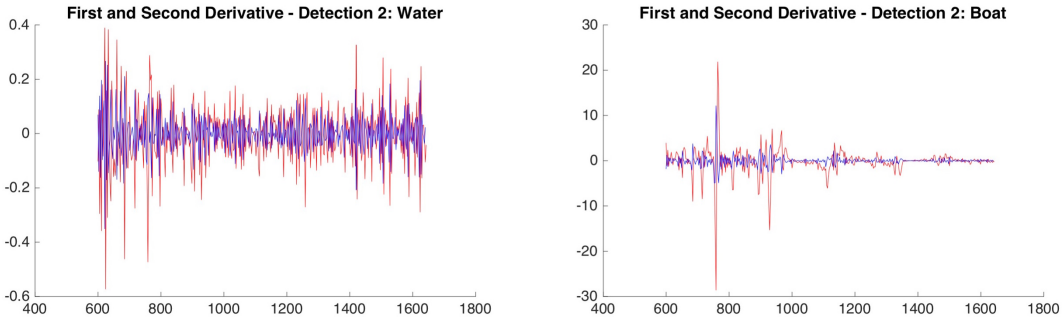


Figure 6.25: *Detection 2: First and second derivative of the spectra obtained.*

In Figure 6.26 the results with ENVI software are presented, in order to validate the difference between the target and background radiance spectra.

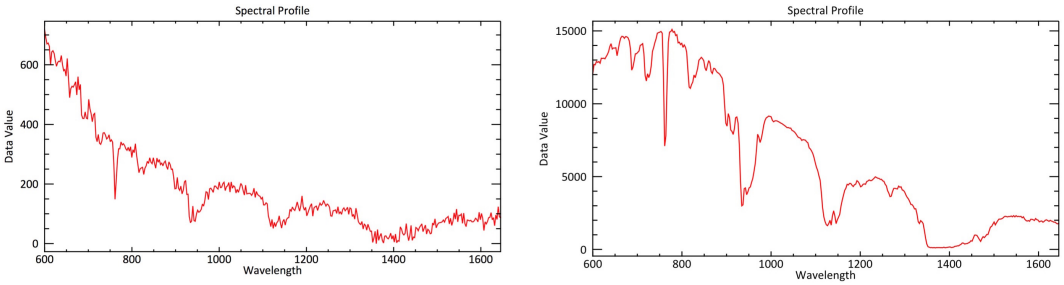


Figure 6.26: *Detection 2: Target and background radiance spectra obtained with ENVI software.*

The result of the projection shows a difference of about 8 meters between the boat's position obtained with RTK/GPS and the projected point.

Detection 3

As for the previous detections, Figure 6.27 shows the target detection together with the EO camera information.

Figure 6.28 shows the radiance spectra for the background and the target. Again, their differences are evident, which allows to correctly detect the target.

The first and second derivatives of the background and target are presented in

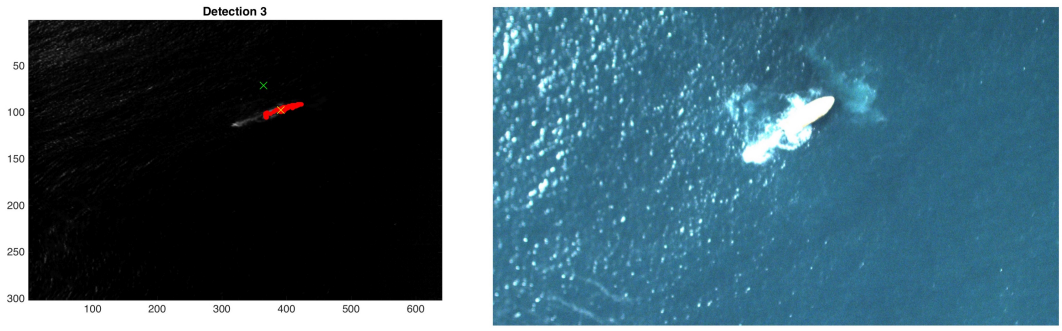


Figure 6.27: *Detection 3: Hyperspectral camera versus EO camera. The yellow cross represents the point analysed to obtain the spectrum of the boat, while the green cross mark the point at what was obtained the spectrum of the water. The points in red represent the target detection in relation to the background.*

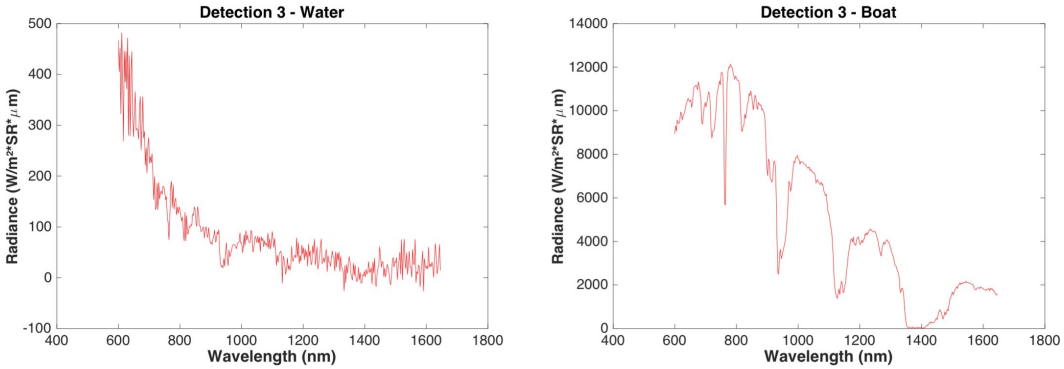


Figure 6.28: *Detection 3: Radiance spectra obtained with Matlab.*

Figure 6.29, and is possible to observe the differences between them, especially at the maximum and minimum values.

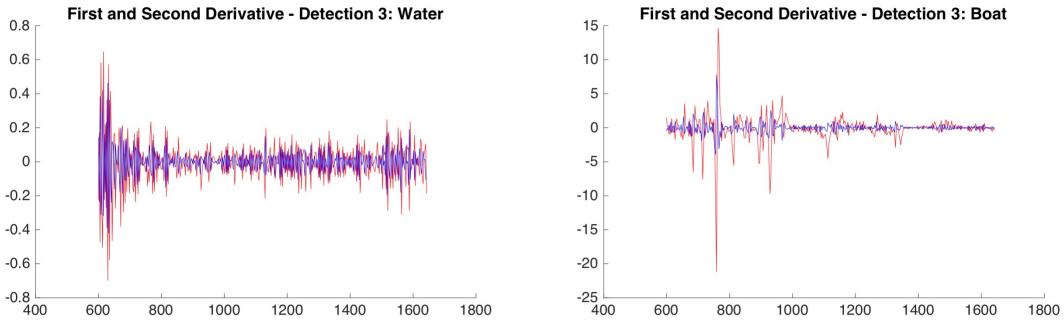


Figure 6.29: *Detection 3: First and second derivative of the spectra obtained.*

Finally, we used the ENVI software to compare the radiance spectra results. In Figure 6.30 the ENVI results are shown. Is possible to observe the difference between the target and its background.

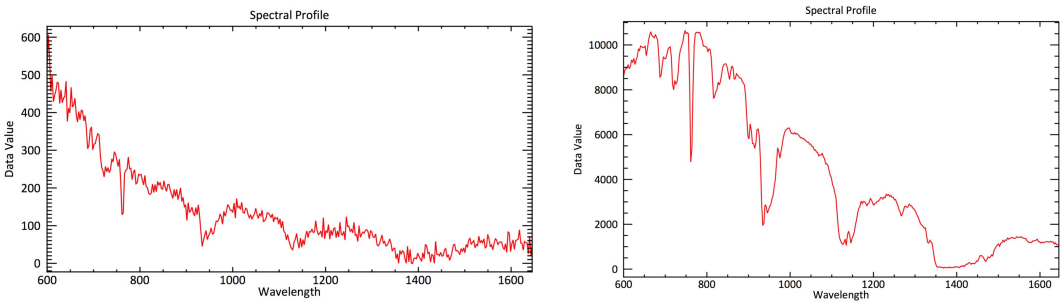


Figure 6.30: *Detection 3: Radiance spectra obtained with ENVI.*

The result of the georeference of the image point compared to the ground-truth position for this situation was about 5 meters error.

6.5 Summary

Throughout this chapter we presented the results obtained for the different points to consider. These points were based on boresight calibration, targets detection and its georeferencing.

So we started by the boresight calibration, using the data from two different datasets. We proved that by calibrating the boresight angles, we were able to improve the geo-reference of the image points.

Later, it was possible to analyze the spectral radiance from the target or the background, which helped to check the proper functioning of created algorithm.

Finally, we calculated the target's position, based on detection performed. This position was compared with the value obtained using the RTK / GPS inserted into the boat. We were able to geo-reference targets within 5 meters, which is accurate considering the intended maritime surveillance application.

7

Conclusions and Future Work

In this thesis, our work focused on the development of hyperspectral imaging techniques for real-time maritime target detection and recognition. We developed the following methods and applications:

- First, we co-develop and implemented an hyperspectral image acquisition solution, that allows to extract synchronized image and navigation data of the hyperspectral camera in near real-time.
- Second, we develop a novel boresight calibration procedure, that allows to calibrate the physical relation between the hyperspectral inertial navigation system and the imaging sensor. The boresight calibration procedure consists on the development of a bundle adjustment method that optimized the re-projection error between control points on the ground and the observed image points.
- Third, we develop a novel method that based on the real-time data acquisition hyperspectral setup, detects based on the first and second derivative analysis of the radiance spectrum the presence of targets (boats) on a pre-defined background (water). Allowing to reduce the amount of generated data that is required to be transmitted to a remote station. The crop of image data is geo-referenced to provide accurate location of the target.
- Finally, we integrated all previously mentioned contributions, into a unmanned

aerial vehicle solution, that was validate in real flights for maritime border surveillance scenario with target ground-truth position information.

The Hyperspectral experimental setup was already published in the International Oceans 2016 conference proceedings, under the title: **"UAV Trials for Multi-Spectral Imaging Target Detection and Recognition in Maritime Environment", IEEE Oceans 2016, September Monterey USA**.

Our future work efforts will be based on the following procedures:

- Conduct further testing to implement the hyperspectral data processing in an actual border surveillance UAV. There will be further tests in Portugal and in Greece. The system will also be adopted to other types of aerial vehicles, such as a Vertical Take-Off Vehicle;
- Further develop the data processing steps in order to develop methods that can identify the type of material based on the radiance spectrum. The separation of the target spectrum from the one in the background is already a improvement to current systems. Extend to actually identify the spectre response will further extend the range of applications of the current solution;
- Further improve the boresight calibration and geo-referencing methods, to diminish the errors. We can use EO information to complement the hyperspectral data and help reduce the bundle-adjustment errors.
- Implement a dual target identification mechanism using Electro-Optical information and Hyperspectral data information;
- Create a data spectrum library for maritime target identification;
- Extend the range of applications of the develop system to other scenarios.

Bibliography

[1] E. Ben-Dor, B. Kindel, A.F.H Goetzb, "Quality assessment of several methods to recover surface reflectance using synthetic imaging spectroscopy data", Remote Sensing of Environment, Volume 90, Pages 389 - 404, ISSN 0034-4257,(2004).

[2] R. Muller, M. Lehner, Rainer Muller, P. Reinartz, M. Schroeder, B. Vollmer, "A program for direct georeferencing of airborne and spaceborne line scanner images", ISPRS Commision I WGI/5, Pages 1 - 3, (2002).

[3] Doneus, M.; Verhoeven, G.; Atzberger, C.; Wess, M. Rus, M. "New ways to extract archaeological information from hyperspectral pixels" Journal of Archaeological Science , 2014, 52, 84 - 96.

[4] Lagueux, P.; Puckrin, E.; Turcotte, C. S.; Gagnon, M.A.; Bastedo, J.; Farley, V. Chamberland, M. "Airborne infrared hyperspectral imager for intelligence, surveillance and reconnaissance applications" Proc. SPIE, 2012, 8542, 854226-854226-10.

[5] Berk, A.; Conforti, P.; Kennett, R.; Perkins, T.; Hawes, F.; Bosch, J. "MODTRAN6: a major upgrade of the MODTRAN radiative transfer code," Proc. SPIE 9088, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XX, 90880H 2014.

[6] Gao, B. -C.; Heidebrecht, K. B.; Goetz, A. F. H. "Derivation of scaled surface reflectance from AVIRIS data" Remote Sensing of Environment, 1993, 44, 145–163.

[7] Richter, R. "A spatially adaptive fast atmosphere correction algorithm" International Journal of Remote Sensing, 1996, 11, 159–166.

[8] ACORNk "Atmospheric correction now", Analytical imaging and geophysics LLC version 3.12, 2001 Boulder, CO.

[9] Adler-Golden, S.; Berk, A.; Bernstein, L. S.; Richtsmeier, S. "FLAASH, A MODTRAN4 atmospheric correction package for hyper- spectral data retrievals and simulations", Summaries of the seventh JPL airborne earth science workshop, 1998, JPL-Pub., vol. 97-21, 1–9.

[10] O'Neil, N.T.; Royer, A.; Nguyyen, M.N. "Canadian Advanced Modified 5S (CAM5S)", Internal Report, CARTEL-1996-0202 Centre d'applications et de recherches en teledetection (CARTELl), Universitie de Sherbrooke, Sherbrooke, Quebec, Canada.

[11] Bernstein, L. S.; Jin, X.; Gregor, B.; Adler-Golden, S. "Quick Atmospheric Correction Code: Algorithm Description and Recent Upgrades." Optical Engineering 2012, vol. 51, No. 11 111719-1 to 111719-11.

[12] Qu, Z.; Goetz, A. F. H.; Heidbrecht, K. B. "High accuracy atmo- sphere correction for hyperspectral data (HATCH)" Proceedings of the ninth JPL airborne earth science workshop, 2000, JPL-Pub., vol. 00-18, 373–381.

[13] Goetz, A. F. H., Srivastava, V. "Mineralogical mapping in the Cuprite mining district, Nevada", Proceedings of the airborne imaging spectrometer data, 1985 JPL Publication, vol. 85-41, 22-29.

[14] Roberts, D. A.; Yamaguchi, Y.; Lyon, R. J. P. "Comparison of various techniques for calibration of AIS data" Proceedings of the 2nd AIS workshop, 1986, JPL Publication, vol. 86-35, 21–30.

[15] Conel, J. E.; Green, R. O.; Vane, G.; Bruegge, C. J.; Alley, R. E.; Curtiss, B. J. "Airborne imaging spectrometer-2: Radiometric spectral characteristics and comparison of ways to compensate for the atmosphere" Proceedings of SPIE, 1987, 834, 140–157.

[16] Haala, N.; Stallmann, D.; Cramer, M. "Calibration of Directly Measured Position and Attitude by Aerotriangulation of Three-line Airborne Imagery" ISPRS 1998, pp. 28–30.

[17] Cramer, M.; Stallmann, D.; Haala, N. "Direct Georeferencing using GPS/Inertial Exterior Orientations for Photogrammmetric Applications" IAPRS 2000, vol. 33.

[18] Mostafa, M.M.R.; Schwarz, K. "A multi-sensor system for airborne image capture and georeferencing. Photogramm" Photogrammetric Engineering & Remote Sensing 2000, vol. 66, No. 12, 1417–1423.

[19] Rzhanov, Y.; PE'ERI, S. "Pushbroom-Frame Imagery Co-Registration" Marine Geodesy 2012, vol. 35, 141–157.

[20] Tuo, H.; Liu, Y. "A new coarse-to-fine rectification algorithm for airborne push-broom hyperspectral images" Pattern Recognition Letters 2005, vol. 26, 1782–1791

[21] Toth, C. K.; Oh, J.H.; Grejner-Brzezinska, D. A. "Airborne Hyperspectral Image Georeferencing Aided by High-Resolution Satellite Images" ISPRS TC VII Symposium 2010, Vol. 38 (7B).

[22] Bethel, J. S.; Lee, C.; Landgrebe, D. A. "Geometric Registration and Classification of Hyperspectral Airborne Pushbroom Data" Int. Arch. Photogramm. Remote Sen. 2000, vol. 33 (B7), 183–188.

[23] Zhang, A.; Hu, S.; Meng, X.; Yang, L.; Li, H. "Toward High Altitude Airship Ground-Based Boresight Calibration of Hyperspectral Pushbroom Imaging Sensors" Remote Sensing, 2015, vol. 7, 17297–17311.

[24] Lenz, A.; Schilling, H.; Perpeet, D.; Wuttke, S. "Automatic in-flight boresight calibration considering topography for hyperspectral pushbroom sensors" Proceedings of the Geoscience and Remote Sensing Symposium (IGARSS) 2014.

[25] Barbieuxa, K.; Constantina, D. ; Merminoda, B. "Correction of Airborne Pushbroom Images Orientation using Bundle Adjustment of Frame Images", ICWG III/I.

[26] Manolakis, D.; Marden, D.; Shaw, G. A. "Hyperspectral Image Processing for Automatic Target Detection Applications" Lincoln Laboratory Journal 2003, vol. 14, No 1.

[27] Manolakis, D.; Lockwooda, R.; Cooleyb, T.; Jacobsonc, J. "Is There a Best Hyperspectral Detection Algorithm?" The International Society for Optical Engineering XV. Ed 2009.

[28] Chang, C.; Chiang, S. "Anomaly detection and classification for hyperspectral imagery" IEEE Transactions on Geoscience and Remote Sensing 2002, vol. 40, No. 6, 1314-1325.

[29] Taitano, Y. P.; Geier, B. A.; Bauer, K. W. "A locally adaptable iterative rx detector" EURASIP J. Adv. Signal Process 2010.

[30] Plaza, A.; Benediktsson, J. A; Boardman, J. W.; Brazile, J.; Bruzzone, L.; Camps-Valls,G.; Chanussot, J.; Fauvel, M.; Gamba, P.; Gualtieri, A.; Marconcinim, M.; Tilton, J. C.; Trianni, G. "Recent advances in techniques for hyperspectral image processing" Remote Sensing of Environment 2009 vol. 113, S110–S122.

[31] Manolakis, D. G.; Ingle, V. K.; Kogon, S. M. "Statistical and Adaptive Signal Processing: Spectral Estimation, Signal Modeling, Adaptive Filtering and Array Processing" McGraw-Hill Companies, Inc., Boston, 2000.

[32] Schaum, A. "Hyperspectral detection algorithms: from old ideas to operational concepts to next generation" Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XII 2006 vol. 6233, 623305, SPIE.

[33] Haavardsholm, T. V.; Arisholm, G.; Kavara A.; Skauli, T. "Architecture of the real-time target detection processing in an airborne hyperspectral demonstrator system" 2nd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing, 2010, 1-4.

[34] Tarabalka, Y.; Haavardsholm, T. V.; Kasen I.; Skauli, T. "Real-time anomaly detection in hyperspectral images using multivariate normal mixture models and GPU processing" J Real-Time Image Proc, 2009, vol. 4, 287–300.

[35] Nansena, C.; Zhaoc, G.; Dakind, N.; Zhaoc, C.; Turnerd, S. R. "Using hyperspectral imaging to determine germination of native Australian plant seeds" Journal of Photochemistry and Photobiology B: Biology 2015, vol. 145 19-24.

[36] Zhang, X; Nansen, C; Aryamanesh, N; Yan, G; Boussaid, F. "Importance of spatial and spectral data reduction in the detection of internal defects in food products" Applied Spectroscopy 2015, vol. 69(4).

[37] Barott, K.; Smith, J.; Dinsdale, E.; Hatay, M.; Sandin, S.; Rohwer, F. "Hyperspectral and Physiological Analyses of Coral Algal Interactions" PloS ONE 2004, vol. 4

[38] Polerecky, L.; Bissett, A.; Al-Najjar, M.; Faerber, P.; Osmers, H.; Suci, P. A.; Stoodley, P.; Beer, D. "Modular spectral imaging system for discrimination of pigments in cells and microbial communities" Applied and Environmental Microbiology 2009, vol. 75.

[39] Kuhl, M.; Polerecky, L. "Functional and structural imaging of phototrophic microbial communities and symbioses" Aquatic Microbial Ecology 2008, vol. 53, 99.

[40] Bachar, A.; Polerecky, L.; Fischer, J. P.; Vamvakopoulos, K.; Beer, D.; Jonkers, H. M. "Two-dimensional mapping of photopigment distribution and activity of Chloroflexus-like bacteria in a hypersaline microbial mat" FEMS Microbial Ecology 2008, vol. 65, 434.

[41] Keith, C. J.; Repasky, K. S.; Lawrence, R.L.; Jay, S. C.; Carlsten, J. L. "Monitoring effects of a controlled subsurface carbon dioxide release on vegetation using a hyperspectral imager" Int. J. Greenhouse Gas Control 2009, vol. 3, 626.

[42] Spangler L. H. et al. "A shallow subsurface controlled release facility in Bozeman, Montana, USA, for testing near surface CO2 detection techniques and transport models" Environmental Earth Sciences 2010, vol. 60, 227.

[43] Lopes, F. "Sistema de visão e laser para perceção em ambientes subaquáticos" Instituto Superior de Engenharia do Porto, 2014.

[44] Hartley, R. I; Zisserman, A. "Multiple View Geometry in Computer Vision" Cambridge University Press 2000.

[45] Ribeiro, J. "Sistema de auto calibração visual para robots do ISePorto, baseado em EKF" Instituto Superior de Engenharia do Porto, 2013.

[46] Slabaugh, G. G. "Computing Euler angles from a rotation matrix"