# A Hyperspectral Imager for a Cubesat to Identify Ocean Ship Parameters

Tabitha Koehn

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Masters of Science
in
Electrical Engineering

Jonathan T. Black, Chair
Scott M. Bailey
Bob C. Clauer

June 23, 2017
Blacksburg, Virginia

Keywords: Hyperspectral Imagery, Ship Detection, Instrumentation

# A Hyperspectral Imager for a Cubesat to Identify Ocean Ship Parameters

Tabitha Koehn

ABSTRACT

A Hyperspectral imager aboard a cubesat would be able to provide images which could be used to identify ships and determine the ship's length and breadth and heading. Depending on the size of the ship, the speed the ship is traveling can be determined as well; however the speed and size determination is limited by the spatial resolution of 100 meters. The spectral signature of the boat is dramatically different from the spectral signature of the open Ocean especially within the range of 400 to 1000 nanometers, and this threshold is the basis for extracting ship data. Hyperspectral Imagers are ideal for minimization with few optical errors introduced, and designs range in durability making them useful on board small satellites especially in the visible and near infrared region. Placing an imager on a satellite allows for consistent observation over a region to identify patterns in ship movement over time.

# A Hyperspectral Imager for a Cubesat to Identify Ocean Ship Parameters

Tabitha Koehn


## GENERAL AUDIENCE ABSTRACT

A camera capable of taking pictures at a higher resolution than humans can see and slightly beyond the visible range could be put on a small satellite and the images from it could be used to find ships and determine the size and direction of the ship. If the camera was closer than on a satellite, the images could be used to also determine how fast the ship is traveling. In the images the ship stands out from the water and software program can use this difference to pick out a ship and run calculations on it. The camera described in this paper is one that can easily be made smaller and are more durable than other kinds of camera. A camera that could go into a small satellite could take frequent images of the same area so that ships in the area could be tracked and shipping traffic observed for gathering research on things like illegal fishing or drug transporting.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

A small sized hyperspectral imager can be designed for use on board a cubesat with the smallest possible optical errors, and as small of a spatial resolution as reasonable, around 100 meters. The data gathered by this instrument could be used to identify ships within the image and determine the size and heading of those ships from low earth orbit. The use of a satellite rather than aircraft observations ensures consistent observation to track shipping patterns and identify irregularities.

Hyperspectral Imaging is an imaging technique used to analyze the spectral emissions over an area. The imager creates a three dimensional hypercube of data with two spatial dimensions and one spectral dimension which measures the intensity of each wavelength at each point. The instrument takes in a row of data in the spatial dimension perpendicular to the direction of movement and disperses the image into its spectral components. This process fills two of the hypercube dimension. As the satellite moves, the third dimension accumulates with the cross track and spectral information at each point along the path, building up the hypercube. The dimensions of the hypercube can be described as the along track dimension, the cross track dimension, and the spectral dimension. Hyperspectral imaging can be used in many different applications and can fill many different purposes within a single application, and the instrument can be specialized based on what the application requires. Applications can range across agricultural, aquatic, geological, manufacturing, medical and more. Using hyperspectral imaging for ocean and lake observation, for example, can reveal chemical and metal traces, harmful algae blooms, ecological conditions, submerged objects, tracks where boats have previously been, and overall lake health.

The basics design of a hyperspectral imager necessitates a camera as well as some sort of dispersive element to achieve the hypercube, both of which partially dictate the resolution of the imager. Hyperspectral imaging evolved out of multispectral imaging as gratings became finer. Multispectral imaging creates a discrete spread of wavelengths whereas hyperspectral imaging records a more continuous wavelength band which can pick up more detail. Hyperspectral imagers are designed with specific small steps between wavelengths along the lines of 5 or 10 nanometers, whereas multispectral imagers tend to look at 4 or 5 specific wavelengths. It is possible to reach a spectral resolution of 1 nanometer which is about as continuous as possible, but this is not really necessary. Using a step size of around 5 nanometers gives a very clear spectral profile of the sea and makes anomalies very easy to identify.

Hyperspectral Imagers vary in usage from being mounted on satellites or in aircrafts down to use in laboratories although the basic instrument design does not change very much whether using a telescope or not. Aircraft and satellite instruments vary greatly in size depending on the parameters of bandwidth, spectral and spatial resolution. This paper deals with the design of a small sized imager, capable of use in a cubesat and with a range of 400 to 1000 nanometers. This wavelength range is the ideal range for identifying water conditions because water has very high absorption across most of the spectrum. This small range allows the design to be small while maintaining a high spectral resolution. Land observation, on the other hand, generally requires measurements up to 2500 nanometers to be able to differentiate the spectra of different plants or plants in different stages of life. When the instrument measures further into the infrared range than around 1000 nanometers, it needs additional cooling systems to maintain the accuracy of the measurement and to prevent ambient heat from the satellite electronics from affecting the data. In addition to needing a bigger instrument for the wider range, the cooling system increases the size, making it difficult to use in a cubesat. Existing hyperspectral imagers on board satellites capable of measuring both visible and infrared wavelengths tend to use two spectrometers. One to measure the visible which does not require extra cooling, and the second for the infrared region with a cooling system, essentially tripling the size of the spectrometer. Laboratory and Aircraft mounted instruments have much less restrictions on size and weight but the parameters for a water observation, cubesat design lend themselves to a simple, small design.

Boat wakes and ship detection have been studied using a myriad of sensing techniques although rarely spectral data. Ship detection can be done with a few different methods, varying from image segmentation, detecting salinity levels, transforming wavelets generated by the ship, or detecting anomalies observed in the image [34]. Some of these methods identify the ship itself and attempt to distinguish the ship from clouds and small islands while others identify the ships wake and use the information from that to extract ship information. The hyperspectral data can be used to extract ship data required for image processing and this paper describes an algorithm to calculate size and heading of the ship. This paper also looks into the requirements to analyze ship speed from the data. The Python code used to process and analyze the data is attached.

## 1.1 Background

### 1.1.1 Hyperspectral Imagery

Hyperspectral imagers have been used in a wide variety of applications including analyzing mining runoff, ocean conditions, crop production, drinking water contamination, and even Mars observations. The data gathered from the imagers can be compared to other data to study the correlation between what can be measured visually over time and what happens on the Earth. Some missions focus on chemical levels in water to estimate growths of biohazards, determine the pollution levels in an area, or track chemical runoff. Different chemicals have different spectral profiles, making them identifiable. Water quality evaluations often involve the clarity of the water by calculation of turbidity or Secchi depth as well as chemical or biochemical identification. Previous missions have dealt with some of the same obstacles when it came to water observation and the methods to overcome them can be integrated into the design described in this paper.

#### Instruments

**Hyperion** Hyperion was an early spacecraft-based hyperspectral imager acquiring high quality data from aboard the satellite Earth Observer 1 (EO1) and has provided a basis for imaging spectrometers since. It has provided data for the majority of studies requiring visible and infrared data, due to the high coverage orbit and has data on both land and coastal regions. EO1 covers the same area as Landsat 7 [9] but a minute behind, so that results can be compared and verified. Pearlman et al.[25] gives a detailed description of the Hyperion instrument. It is a push broom imager with a single telescope and a light splitter sending light to two spectrometers, one for visible and near infrared (VNIR) wavelengths and the second for shortwave infrared (SWIR) wavelengths. The second spectrometer requires cooling through use of a cryocooler. Both spectrometers made use of the Offner optical configuration which uses three reflectors, one of which being a convex grating. The VNIR spectrometer was passively cooled and operated at 283 K while the SWIR spectrometer was kept at 110 K. The satellite orbited at 705 km and captured an image of 30 km in the direction the satellite is moving by 7.7 km in the perpendicular direction. To achieve these numbers the fore-optic telescope uses a three-mirror astigmate design with a 12-cm primary aperture and an effective f-number of 11 [25]. A field of view of 0.624 degrees gave the 7.7 km cross-track area. These values create a design much larger than one capable of fitting into a cubesat, but the basics of the design are relevant. The swath is actually smaller than would be useful in this application because it does not cover enough area to see many ships. Hyperion does have a good spatial resolution for ship analysis but that increases the size of the instrument.

Hyperions VNIR spectrometer uses a 60 (spectral) by 250 (spatial) pixel array [9] and the SWIR spectrometer has a wider spectral range of 160 channels by the same spatial and used an HgCdTe detector which was kept cooled. Across the entire range, there is a 10 nanometer spectral resolution. Pearlman et al. [25] examined the signal to noise ratio across the spectrum which caused some pixels to be ignored. From 550 to 700 nanometers the SNR varies between 140 and 190, which results in the VNIR spectral channels being reduced down to 50. Hyperion's primary purpose is not water observation, so lower SNR is not as big of a problem as it would be for coastal or open water observing satellite.

The instrument includes an on board calibration system involving solar, lunar, and earth-surface-observing [25] measurements to verify the performance of the spectrometers. Hyperion was designed with two internal calibrating lamps to be references for a radiometric standard; however after launch the lamp's output increased, making their use as references uncertain. Vicarious calibration of the instrument directly observed a test area of the Earth and then compared the image to Landsat-7. Similar techniques can be implemented to a cubesat design given further research. [19] [25][9]

**MARTE** The Mars Astrobiology Research and Technology Experiment (MARTE) is focused on the search for life on Mars by drilling ground samples and analyzing the samples by remote sensing, including a visible and near infrared imaging spectrometer, a panoramic context imager, and a microscopic imager. Brown, Sutter and Dunagan [4] evaluated the performance of the instrument in Rio Tinto, Spain, an area with similar chemistry to that of Mars. The soil is high in Iron, the Rio Tinto

River appears red, and the whole area is made up of subvolcanic materials. The experiment involved the instruments mounted on a drilling rig remotely analyzing cores.

The VNIR Spectrometer is a pushbroom imager with a spectral range of 400 to 1000 nanometers. The instrument makes use of controlled illumination and vapor emission lamps for spectral calibration. It incorporates a Dichronic order-blocking filter which prevents the CCD from absorbing light from higher order diffractions which might cause errors and longer dimension of the detector measures the spectral dimension, maximizing spectral over spatial. There are 782 x 582 effective pixels which have a pixel size of 8.3 x 8.3 micrometers. The instrument included an amethyst filter to suppress a peak in the green wavelength region in order to increase integration times for improvement in the blue and infrared and prevent saturation in the green.

Rather than a typical Offner spectrometer design, the MARTE hyperspectral imager makes use of a collimator, a couple of prisms, a grating and a focus to direct the image onto the CCD. The light passes through a slit to the collimator, which straightens out the rays of light. From the collimator the light goes to a wedge prism and then to a diffraction grating and onto another prism and then to the focus. There is no telescoping lens involved because it is mounted on a drill rig and it was unnecessary. [29][4]

**PROBA/CHRIS** Barnsley et al. [3] describes the Project for On-Board Autonomy (PROBA) Satellite which carries the Compact High Resolution Imaging Spectrometer (CHRIS). PROBA/CHRIS is a project to test the capabilities of a spacecraft with as little intervention from the earth as possible. PROBA is unique in many ways but most applicable to hyperspectral imaging is its high maneuverability due to four reaction wheels, allowing the satellite to be positioned off nadir in any direction. This maneuverability makes it possible to acquire five images for a single pass as well as maximizing chances of images without sun glint. With a slow pitch, the integration times for the imager can also be lengthened, which improves the signal to noise ratio. The images take an angular sampling from 55, 35, and 0 degrees. Over a couple of days, its possible for the satellite to take multiple sets of images of the same area.

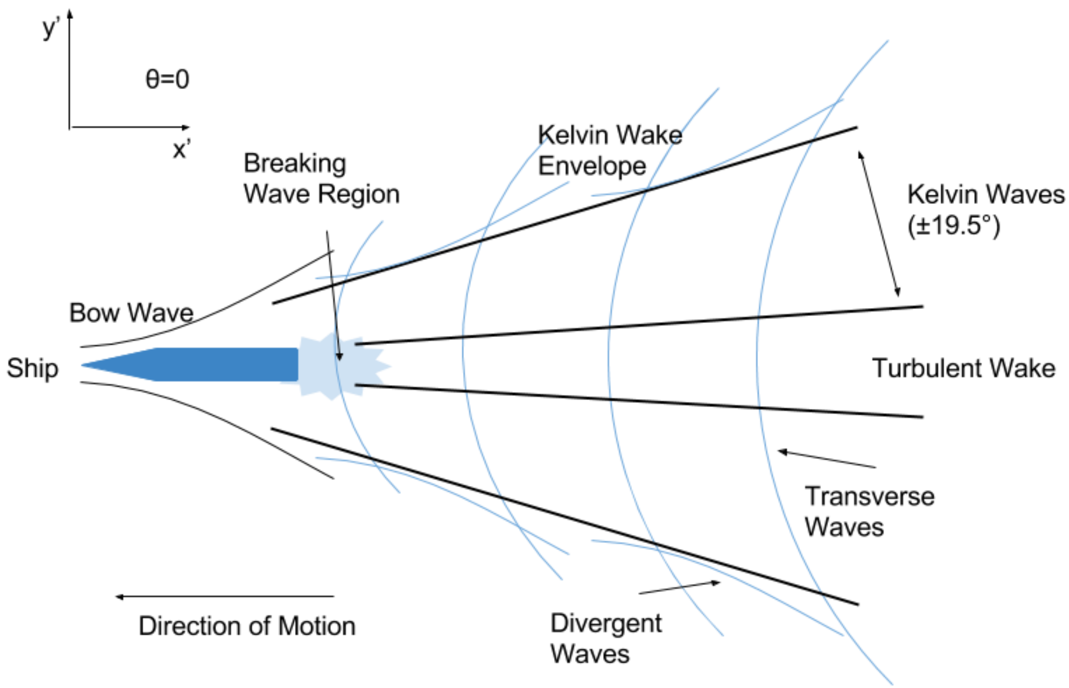
The imaging instrument itself is a 14 kg device, 0.79 meters by 0.26 meters by 0.20 meters with a spectral range of 400 to 1050 nanometers. CHRIS includes a telescope, a spectrometer, and a CCD, and is capable of five operational modes. The telescope is a two mirror configuration which operates at f/6 and has a diameter of 120 millimeters. It uses only spherical surfaces and a large meniscus lens at the entrance in order to reduce spherical aberration. The lenses used in the telescope are coated with a multilayer dielectric to achieve 98% reflection [3]. The telescope has a 746 millimeter focal length which achieves a 15 kilometer by 20 meter instantaneous image when the satellite is at apogee and 13 kilometer by 17 meter image at perigee. While this is far too large of a design for a cubesat, the satellite has an extremely high range of capabilities some of which were worth consideration to work into a smaller design.

The detector used has 770 columns and 576 rows with increased efficiency in the deep blue region and high flexibility in read-out, which is appropriate given the range of the device. The CCD is a thinned, back lit, single layer detector, where each detector has a dump gate to a register to ensure fast data processing speeds. The detector used has a nearly linear relationship between wavelength and bandwidth, so with a range of 400 to 1050 nanometers, the bandwidth varies from 1.25 to 11.25 nanometers. While this is an interesting method to increase blue region efficiency, it is not exactly necessary and relies on the spectrometer configuration which is too large. PROBA/CHRIS is extremely flexible but beyond what is necessary for the purposes of this design. [3][1]

**HyspIRI** The NASA Hyperspectral Infrared Imager (HyspIRI) uses a Dyson spectrometer design rather than the typical Offner configuration for observation of the ecosystem and natural hazards in terrestrial and aquatic systems, biogeochemical, and atmospheric geology fields. The Dyson design minimizes polarization sensitivity and scattered light for improved coastal observation. The signal-to- noise ratio is very high, and the spatial resolution over open ocean averages to around 1 kilometer. The Dyson design is a very small design without need for minimization. The satellite contains both a visible to short wave infrared (VSWIR) imaging spectrometer and a thermal infrared multispectral imager (TIR) and uses an intelligent payload module (IPM) for on board processing. The VSWIR spectrometer uses 10 nanometer bandwidth with a range of 380 to 2500 nanometers and a 30 meter spatial resolution at low earth orbit. To maintain an accuracy of greater than 95% in both spectral and spatial directions, the device goes through monthly lunar calibrations and occasional surface calibration as determined by the IPM. The Dyson design minimizes polarization sensitivity and scattered light for improved coastal observation. The signal-to-noise ratio is very high, and the spatial resolution over open ocean averages to around 1 kilometer. The Dyson design is a very small design without need for minimization. [17][7]

### 1.1.2 Ship Wakes

Figure 1.1: Diagram of Ship Wakes [11]




As a ship moves through water it leaves behind a trail made up of a turbulent wake directly behind the ship and Kelvin wakes which radiate out at the arcsin (1/3) or 19.47 degrees off both sides of the ship [11]. The frequency and amplitude of the waves in the Kelvin wake is dependent on how fast the boat is traveling. The turbulent wake trails behind for 10s of kilometers and the wake spreads out to a width of 60 meters at the maximum before becoming indistinct [33]. The bubbles within the wake increase the radiance reflected, and in open ocean the wake appears more green than the surrounding water. The more turbid the water observed is, the more difficult to detect ships because the water has more bright spots and can not be assumed to be a roughly uniform area [30]. Sea conditions and weather conditions do affect the persistence of the wakes.

Using multispectral data from Sentinel-2, Heiselberg [11] determined that the background sea is generally dark and mainly reflects in the blue spectral region while the ship, wakes, clouds, and any land masses have high radiances within the visible range. Large, bright land masses can oversaturate detectors designed for water observation, so in the image these portions are easy to eliminate for ship identification. Cutting out large sections of over saturation does not quite eliminate all land in the image as small islands and sandbars will not saturate the detector. These land masses will appear bright in the image but do not have characteristics specific to ships making the islands identifiable. Comparing data to other images of the area would make it possible distinguish land from ships.

Most ship analysis has been achieved by radar or lidar or by capturing multiple images over a short period of time in order to extract speed more directly than wake analysis but these methods are not typically used over long periods of time.

# Chapter 2

# Instrument

## 2.1 Introduction

The hyperspectral imager design requires a telescope, a spectrometer optical system and a detector capable of measuring frequencies from 400 to 1000 nanometers with a spatial resolution of at least 100 meters. The spectrometer is designed to image the earth as it passes over and disperse the image into the spectral components it is made up of to record the intensity of the reflected light. As the satellite passes over the earth, a three-dimensional image is created, two dimensions spacial and the third spectral. The third dimension accumulates as the cross track and spectral data for each step along the track dimension is saved. The orbit of the satellite would determines the frequency necessary for the detector operations. For the purposes of water observation, the signal-to-noise ratio (SNR) has to be as high as possible.

Choosing the method of obtaining images, whether by whisk broom or push broom scanning, is one of the first determinations which occurs outside of the spectrometer design. Whisk broom spectrometers record the spectra at each pixel by the same detector array while push broom requires a linear array for each spatial pixel. The push broom system is a more mechanically simplistic system than the whisk broom because it does not have any moving parts. Push broom scanners also offer better SNR than whisk boom scanners although there is a trade-off. The push broom technique is more difficult to calibrate, introducing more error in smile and keystone distortion [16] especially over time. Over time, shifts in the image begin to appear as the satellite's altitude slowly decreases, varying non-linearly across the track which generates smile [10]. This means that the instrument should have smile reduced to less than 5% initially. The cubesat has a relatively short lifetime, so the smile will not increase too dramatically and what does appear can be removed using smoothing techniques. The advantages of the push broom design win out overall.

The image distortions which need to be minimized for high accuracy include smile, keystone, coma, astigmatism, and vignetting although completely eliminating all error is impossible. Smile distortion is when the monochromatic image is curved, while keystone distortion is when there is unequal magnification across the spectral range due to beam angle and the focus. This mostly affects pixels with sharp boundaries or when a target is only a few pixels large. Keystone distortion can be calculated using the following proportion where $\epsilon$ is the angle between object and image and $\alpha$ is the width of the focus [32]. Keystone tolerance is more difficult to predict because it depends on the spectra observed and the point spread function [23] but can be determined through simulation.

$$Keystone\ Distortion = \frac{\cos(\epsilon - \frac{\alpha}{2})}{\cos(\epsilon + \frac{\alpha}{2})} \tag{2.1}$$

Coma is an aberration in which imperfections in the lens cause variation in magnification over the entrance pupil across the spectral data. Astigmatism is where the light rays have different focuses so the image fails to resolve properly on the detector. Vignetting is when the edges of the image are darker due to the lenses interfering with the passage of the rays.

All of these distortions can result in unusable data. The best design eliminates as much possibility of distortion as possible and limits what cannot be eliminated. Some designs eliminate certain aberrations simply because of the physics of the lenses, but no design eliminate every problem. Even beyond the optical errors, there will be spread in the pixels so any bright spot will spread to the surrounding pixels in the detector.

## 2.2 Fore-Optics

The fore-optics has to be secure enough to block stray light that might enter the device from a different direction than the target and then focus the light from the target onto the slit with as much transmission as possible. The lenses also have to be minimized in size as much as possible without affecting how much of the image makes it to the detector. The optics contribute to the field of view of the satellite, which determines the ground spot size and the field angle, as well as the focal length and from that the f-number. The f-number is the ratio of the focal length of a lens to the aperture diameter and is an indicator of how quickly the sensor can process data. The fore-optics have the potential to introduce additional errors which would continue through the system. In order to eliminate vignetting errors, spectrometer and fore-optics should have pupil matching [13] which can be designed in conjunction with spectrometer design. The exit pupil of the telescope and the entrance pupil of spectrometer do not have to be exactly the same but close.

For water observations, there is potential for a huge variation in reflectance, from the dark water with low reflectance to land which has very high reflectance, appearing bright in images. While attempting to identify objects on the water, the areas with low reflectance have a higher priority and require a very good signal to noise ratio. The most simplistic way to achieve this is by using a large high collection aperture [22] which has the problem of over saturating the detector when the reflectance is higher than the majority of the image. Ensuring that enough light is passing into the instrument is extremely important and can limit the times when it is actually usable. A large aperture diameter relative to the size of the optics lowers the f-number which is ideal for minimization and increases the probability that there will be enough light in the spectrometer. It is still likely that the usability of the instrument would be limited to times when the sun is high in the sky and the maximum amount of light is reflected.

The field of view of the telescope determines the cross track observation area. To maximize spatial resolution, the scan angle is kept small on the order of around $2.5°(\pm 1.25°)$. This does depend on the altitude that the satellite was set to, and for these purposes was chosen to be 600 km. The calculation of the scan angle was based on the following equation [24].

$$Field\ of\ View = 2\ (Height)\ \tan(scan\ angle) \tag{2.2}$$

The spatial resolution and field of view will change over time as the satellite slowly descends, but this is a good starting point. Fore optics can be built specifically to meet these requirements, with a telecentric focus next to the slit and chromatic correcting lenses [8]. This simplifies the design and keeps it small compared to the larger telescoping designs. This puts the exit pupil at infinity which limits the angle of incident light into the instrument. Fisher describes fore optic designs fitting these requirements, either a modified double Gauss or an inverted Huygens eyepiece with the addition of field lenses to optimize pupil matching [8]. The double Gauss design is a compound design with two opposite facing Gauss lenses and similarly, the Huygens eyepiece is two lenses but one side is a plane and the other convex. The curved side of the lenses face in the same direction. There are commercially available lenses by brands such as Fujinon or Schneider optics, but if the spectrometer is being custom fabricated, the fore optics could be as well.

The slit length and width depends on the pixel size of the detector and the number of pixels of the detector. In this case, the slit length is 8.2 mm with a width of 16 $\mu$m. Due to the push broom design choice, the slit has to be as wide as the detector because it fills the cross track dimension all at once for each along track step.

## 2.3 Spectrometer Designs

### 2.3.1 Offner Spectrometer

Offner spectrometers are often used in hyperspectral imagers due to their high spectral and spacial resolutions. The Offner spectrometer uses concentric lens with a 1:1 magnification, to eliminate typical optical errors such as axial aberration, coma, and sagittal field curvature however is affected by astigmatism and vignetting. The Offner design requires a low f-number and can process a large field size [8]. The light passes through a slit to a concave mirror and then reflects onto a dispersing grating. From there it is reflected off another concave mirror to focus upon the CCD.

The Offner spectrometer is simple, inexpensive, compact and is flexible to design requirements. Changing the range of wavelengths is achieved by altering the geometry of the mirror and grating. Kim [16] describes the property of Offner spectrometers referred to as the ring of zero aberration which derives a ring field where there are no obstructions. A more basic design uses only a grating and a mirror where the single mirror fills the roll of the primary and tertiary surfaces. The ring removes aberrations by balancing the third and fifth order aberrations and by splitting the primary mirror into two mirrors, a spherical primary mirror and a spherical tertiary mirror with a shared center of curvature. The three surface design makes manufacturing easier and slightly increases the performance of the instrument. It does not affect the size of the spectrometer but is more flexible to match the pupil to the fore optics to reduce error [13].

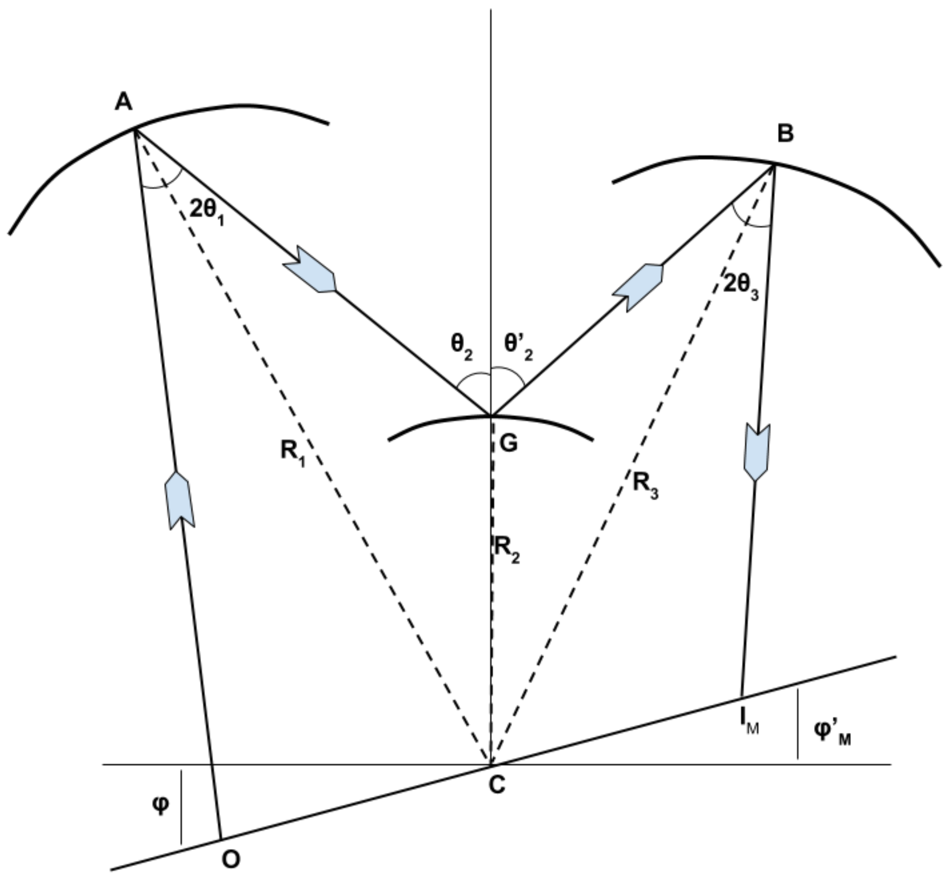
The Offner design is very susceptible to astigmatism, so minimizing this aberration is a high priority. Chrisp [6] and Prieto-Blanco et al. [27] have both addressed resolving astigmatism either for a single wavelength or across the range of wavelengths. The vignetting can be addressed by analyzing the ray trajectories around the optics, however the intensity of the image around the edges is still slightly diminished. Applying Prieto-Blanco's method to determine the geometry of the spectrometer should lead to design with low astigmatism, however some is permitted to ensure minimization and limiting other errors and in actual fabrication it is impossible to completely eliminate. Prieto-Blanco et al. [27] evaluates astigmatism based on the sagittal and meridional image locations and the incidence and reflection angles on the grating. The following equations summarize the conclusions which give a geometric basis for the Offner design. Using the conditions of the Rowland circle to determine the geometry, astigmatism is eliminated for a single wavelength. The Rowland circle has a diameter containing the center of curvature and the incidence point where the light ray hits the grating or mirror and implies that the image will have no coma. Each wavelength implies another Rowland circle which would change the geometry of the device but can be resolved with the concentric design. It is a circle with the radius equal to the radius of curvature of the grating and tangent to the grating at the chief incidence ray and exists on the plane orthogonal to the grooves of the grating. When this concept is fulfilled, the meridional image is on the same circle and the coma aberration is eliminated. The sagittal image is not as simple as the meridional image though, and the difference between those images is the astigmatism. This can be calculated with the following equation [27].

$$\Delta r = R_l \sin\theta_l' \tan\delta \tag{2.3}$$

Where $\Delta$r is the longitudinal astigmatism, $\theta_l'$ the angle of refraction, $\delta$ is the angle of incidence, and $R_l$ is the radius of the lens. When the astigmatism is greater than zero the meridional image is beyond the detector. Using the concept of the Rowland circle, the geometry of an Offner Spectrometer can be calculated.

The schematic of the design (Figure 2.1) can be overlapped with the Rowland circle for each surface to calculate the relation between the angles of the optical elements, the radii of curvatures for the various elements and the location of the sagittal and meridional images. The properties of gratings give the Bragg relation [27] for incident and diffracted angles of the secondary surface based on the number of grooves per unit length (p), the wavelength ($\lambda$), the order of diffraction (m), and the incidence ($\theta$) and diffraction ($\theta'$) angles.

Figure 2.1: Schematic of Offner three-lens geometry. [27]



$$\sin(\theta) + \sin(\theta') = m\lambda p \tag{2.4}$$

By definition the location of the meridional image and the sagittal image of an object due to a reflective grating is as follows [27].

$$\frac{\cos^2(\theta)}{r} + \frac{\cos^2(\theta')}{r'_M} = \frac{\cos(\theta) + \cos(\theta')}{R} \tag{2.5}$$

$$\frac{1}{r} + \frac{1}{r'_S} = \frac{\cos(\theta) + \cos(\theta')}{R} \tag{2.6}$$

For the best final image, the center of slit and the final and intermediate images are along the three Rowland circles so that all measured wavelengths are passed through and fit to the equation developed and reinforce the idealism of the concentric design. An actual instrument would not be so ideal and would have some minor levels of astigmatism. The incident, reflection, and diffraction angles are all easily simplified by the simple geometry of the conditions, which implies that when the line between the object and the center of curvature (CO) and the reference ray is 90 degrees, then the object is on the first Rowland circle. The angles formed using this assumption are below [27].

$$\frac{\pi}{2} - 2\theta_1 + (\pi - \theta_2) + \frac{\pi}{2} + \phi = 2\pi \quad Simplified\ to\ \ \phi = \theta_2 + 2\theta_1 \tag{2.7}$$

$$CO = R_1 \sin(\theta_1) = R_2 \sin(\theta_2) \tag{2.8}$$

The Rowland circle for the tertiary surface results in similar equations for the angles involved with the third surface and the location of the meridional image ($CI_M$). The primary and tertiary surfaces have nearly identical geometry, so the equations [27] to determine the angles between surfaces are nearly the same.

$$\frac{\pi}{2} - 2\theta_3 + (\pi - \theta'_2) + \frac{\pi}{2} + \phi'_M = 2\pi \quad Simplified\ to\ \ \phi'_M = \theta'_2 + 2\theta_3 \tag{2.9}$$

$$CI_M = R_2 \sin(\theta'_2) = R_3 \sin(\theta_3) \tag{2.10}$$

These relations give a basis for designing a spectrometer with minimal astigmatism which can be calculated by the difference in where the sagittal and meridional images focus. If the difference between the angles is zero, the astigmatism will be zero, so equating the angles ensures the least optical error. The line between the center of curvature and the sagittal image location ($CI_S$) can be related to center of curvature to meridional image location ($CI_M$) to calculate astigmatism [27].

$$CI_S = \frac{CI_M}{\cos(\psi'_M - \psi'_S)} \tag{2.11}$$

$$Astigmatism = CI_S \sin(\psi'_M - \psi'_S) \tag{2.12}$$

$$\psi'_M = \psi'_S = \bar{\theta}'_2 - 2\bar{\theta}_3 \tag{2.13}$$

The size of the eventual image is dependent upon the geometry of the grating and radius from the center point. This size relates to the size of the detector necessary to ensure accuracy so that all rays reach the detector [27].

$$R_2 = \frac{h_{spec}}{mp\Delta\lambda} \tag{2.14}$$

These equations give the basis for a design with minimal astigmatism although do not address vignetting within the spectrometer which depends on the trajectory and spread of the rays. Vignetting is most likely to occur near the edges of the grating or where the two mirrors overlap. There are mathematical equations to eliminate these errors, but the complexity of the equations makes the analysis impractical. Simulation of the instrument is a more accurate method to evaluate optical errors as well as providing the opportunity to test different lengths and angles to minimize the space necessary to achieve the imaging required.

Using OSLO EDU, educational optics simulation software, the hyperspectal imager design could be evaluated and the smallest design with reasonable error can be determined. Mouroulis has multiple papers [13] [12] [23] regarding the minimization of Offner spectrometers with as few errors as possible. Using these papers as a guideline, the spectrometer can be minimized as much as possible and compared to the astigmatism minimization. The aberrations and spot sizes can be evaluated using this software with the intention of keeping spot size small and aberrations limited.

This geometry works with an f number of 2.6 and was evaluated with a wavelength range of 300 to 1000 nanometers to ensure that the part of the spectrum of interest is as accurate as possible. The radius dimension refers directly to the curvature of each lens. Thickness describes the distance between the lens in question and the next lens. The aperture radius describes how large the lens itself actually is. The aperture radius can be adjusted based on the geometry of how the lenses are set up.

Table 2.1: Offner Spectrometer lens data.

| Surface | Radius | Thickness | Aperture Radius |
|---|---|---|---|
| Object | | 136.0 | $1e^{-6}$ |
| 1 | -136.0 | -67.0 | 25.0 |
| Grating | -68.5 | 61.5 | 12.5 |
| 3 | -130.0 | -130.0 | 34.0 |
| Image | | | 130.0 |

The size and positioning can also affect spot size and aberrations as the angles of the lenses and where the rays hit on the curvature of the mirror affect where the image forms.

While varying the thicknesses and angles of the surfaces, there was an evident trade-off between vignetting and astigmatism. Minimizing one caused more of the other and so the dimensions described are simply the best option while not entirely eliminating either aberration.

The three lens design is simply based off the two lens design, and so the first and third surfaces share the same center of curvature. The software makes this difficult to achieve because every movement of the lens is based off the lenses around it. Every alteration affects the aberration and how many of the rays actually reach the detector. The build of this requires angling and offsetting the lens while attempting to maintain high image integrity.

Table 2.2: Offner Spectrometer lens Geometry.

| Surface | Decentrations | Rotation Angle (Degrees) |
|---|---|---|
| Object | 20.0 | 0.0 |
| 1 | 20.0 | 12.0 |
| Grating | -14.0 | 0.0 |
| 3 | -63.0 | -25.0 |

The spot diagram in figure 2.3, shows the light would spread and not simply be absorbed by a single pixel. Spread is expected because of the nature of light but can be reduced so that it doesn't create too much error in the image. Using a grating with a higher number of slits increases the spectral resolution which can then be

Figure 2.2: Offner Lens Drawing.

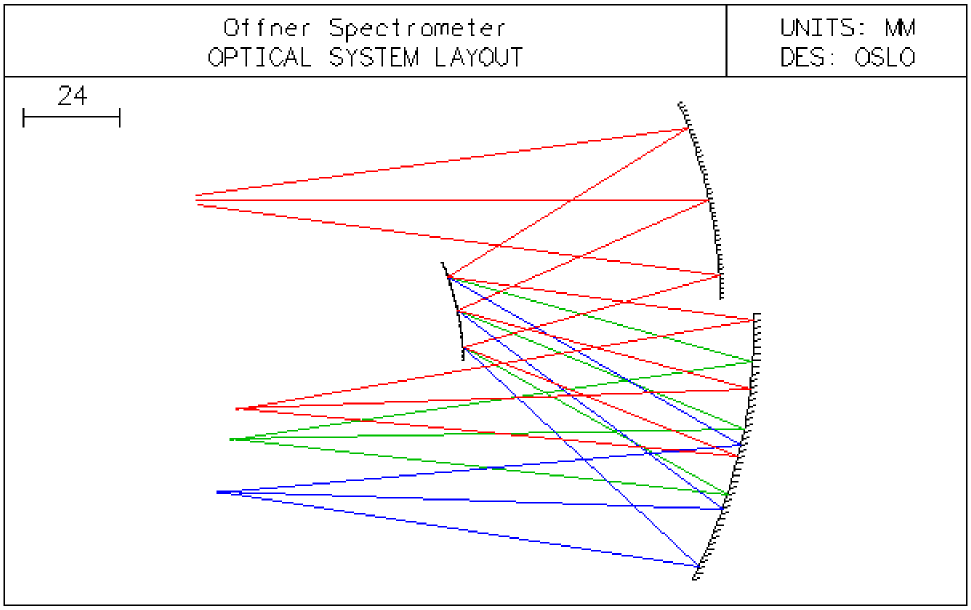
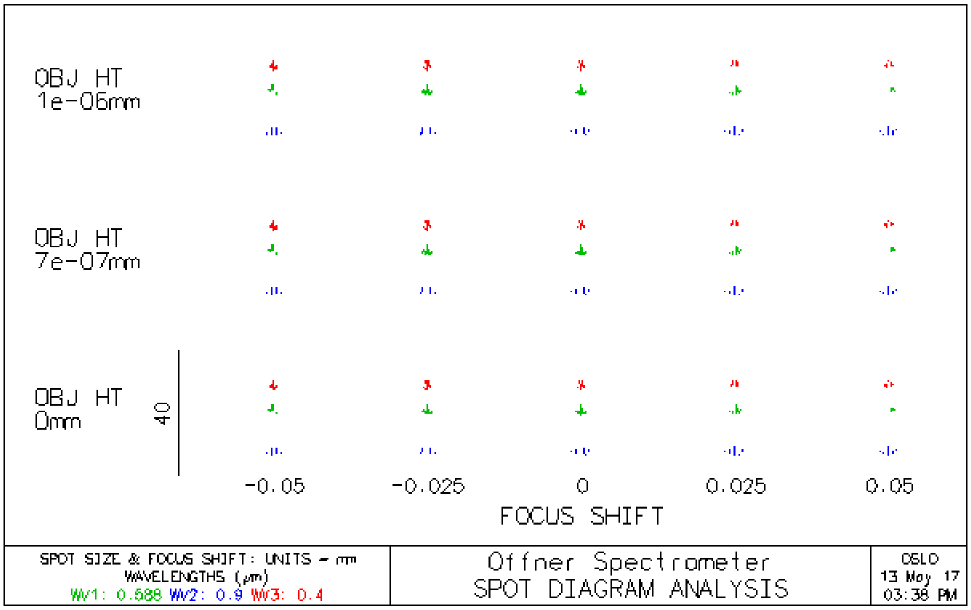



Figure 2.3: Spot Diagram of Offner three-lens geometry.



averaged to reduce the effect of the spread. Designing the spectrometer to have spectral resolution a third of what is actually desired would mean that every three pixels in the spectral dimension can be combined to reduce the spread. The spread is also highest where there is the least detail near the near infrared so the more significant data is preserved. The simulated design would be small enough to fit within the cubesat and easy enough to fabricate at a relatively low cost because the mirrors do not need to be specialized.

### 2.3.2 Dyson Spectrometer

The Dyson spectrometer is a much more compact although more expensive design requiring more complexity to eliminate optical errors. Despite this, the Dyson spectrometer was the second best option. Like the Offner spectrometer its a concentric design with a 1:1 magnification. The basic design involves a long Dyson lens and a grating which, while simpler than the Offner form, has a few drawbacks which is part of the reason the Offner design has been much more commonly used. In more recent years instrument designers have tried to make use of the Dyson design because of its compact size.

Montero-Orille et al. [21] explain the conceptual design of the Dyson imaging spectrometer using the Rowland circle concept and generalize the design for the removal of astigmatism. The equations derived from the Rowland circle hold for eliminating astigmatism for two wavelengths but not across the whole desired spectrum, so by using the wavelengths at the edge of the spectrum measured can minimize the error similar to the Offner analysis. It can be found that the total dispersion is the change in the wavelength divided by the distance between the images at the extreme wavelength. The ratio of the radii of the lens ($R_l$) and the grating ($R_g$) is based on the [21]. The angle of diffraction is $\theta'_g$ and $n_l$ is the len's refractive index.

$$\frac{R_g}{R_l} = \rho = \frac{n_l}{\sqrt{1 - 2n_l \cos(\theta'_g) + n_l^2}} \tag{2.15}$$

For any given wavelength, the angle of refraction is a function of the grating density. The relative difference increases as the grating density increases. The grating density does have to be fairly high to ensure a high enough spectral resolution so the angle has to be large as well.

While the Dyson imaging spectrometer is a more compact design than the Offner configuration, the drawbacks due to astigmatism and other optical errors complicate the design. There have been many attempts to modify the design to fix the aberrations. The primary errors come from the large size of the lens as well as the proximity of the slit and focal plane to each other and to the plano rear surface of the Dyson lens [15]. Increasing these distances result in spherical aberrations. Warren et al. and Pei et al. [15] [26] have both attempted designs to correct these errors by adding lens or replacing lens with mirrors. However while fixing some problems, these create more problems, resulting in the need to extend the design and making it much more complex, thus more difficult to fabricate and more expensive.

### 2.3.3 Chromatographic Hyperspectral Camera

Among the classes of hyperpectral imaging is a technique called tomographic measuring, which eliminates the use of a slit that other techniques require, however requires more signal and image processing. Murguia et al. [14] designed one such sensor using a telescope, a two prism set and a framing video camera. The prism set is the dispersive element which rotates to project the image onto the focal plane array of the camera, one angle per camera frame. The prism is designed to pass one wavelength through directly while the others are dispersed along the axis. This type of prism is referred to as a direct vision prism. In a single camera frame the area is imaged and as the prism rotates and frames progress, the spectral response is multiplexed over the focal plane array. During the processing of the data, it has to be de-multiplexed to extract the integrated data and reconstruct the hypercube. There are a few conditions that do not require complete reconstruction, but those are the cases where the light is monochromatic.

While this design adheres to many of the requirements of the device for a cubesat, and Murguias design has room for minimization, the rotation of the interior prism makes it less than suitable for space operation, without making the design larger to increase the ruggedness of the sensor to be able to withstand launch. The rotation of the prism could also effect the speed of the data accumulation resulting in the loss of data. The complexity of the data is problematic as the hypercube is such a large amount of data to transmit without having to worry about preprocessing.

### 2.3.4 Interferometer

Another technique for hyperspectral imaging that was considered was interferometry where the light waves are superimposed and makes use of the phase of the wave to identify the wavelengths present.

**Fabry-Perot Interferometer** This particular type of interferometer makes use of two partially silvered surfaces to create a series of reflections. At each collision with the second surface, part of the light is transmitted through where it is focused through a lens onto a detector array. The light interferes and focuses into concentric circles based on the order of the interferences. The geometry of the interferometer and the order of interference allows for the calculation of the wavelength.

The spectral resolution is controlled by the mirror reflectivity, surface roughness, the degree to which the mirrors are parallel and the degree of collimation of light on the filter. All of these factors mean that the spectral resolution is limited more than the conditions where the spectral resolution is determined by the grating. The free spectral range (FSR) is the wavelength spacing between successive transmission orders and divided by the finesse of the instrument to become a rough maximum for the spectral resolution. This makes it more ideal for longer wavelengths than the visible and near infrared of interest.

**Michelson Interferometer** The Michelson interferometer uses a beam splitter to divide the incident light into two directions and then recombines the light after phase shifting one of the beams causing interference. The intensity of the beams depends on the difference in the optical path length and when plotted demonstrate the Fourier transform of the observed spectrum. The second mirror moves to change the path difference and when moving at a constant velocity creates a pattern of constructive and destructive interference based on the wavelength. When the mirrors are equidistant, the path difference is zero, so there is complete constructive interference. When the second mirror is set a distance of half a wavelength different, there will be complete destructive interference.

Both Interferometer designs are more ideal for Doppler observations which could be considered to determine ship speeds, but have many moving parts and high sensitivity making it less than ideal for cubesat operations.

## 2.4 CCD

Observing water rather than land requires a greater sensitivity and a slightly more complex camera. These conditions require a thinned, back-lit detector array with a high signal to noise ratio. With the desired resolution of the spectral component, there is only needs to be 110 pixels in that dimension. Accuracy in the spectral dimension can be improved by averaging two or three spectral pixels together [18] so the instrument can accumulate data with a spectral resolution of 2 nanometers and average every three pixels so that the data has a resolution of 6 nanometers. One of the smallest pixel arrays with enough detail to achieve this resolution with pixels to spare has one length dimension of 512 pixels. The spatial dimension does not have to be very wide because the along track accumulates so much data. The wider the cross track dimension is, the more difficult it is to transfer the data for each frame. Limiting this dimension to 512 pixels as well keeps the detector efficient and inexpensive. Saving the data at each along track point is faster when not all of the data needs to be saved such as it would be in this case.

Princeton Imaging produces the ProEM Professional Grade EMCCD Camera [28] which fits all requirements. It has good absorption between 200 to 1050 nanometers and was designed to have as low noise as possible, at less than 5 $e^-$ rms. The full-frame resolution speed is 100kHz; however, because not all of the pixels are needed for the spectral dimension, faster speeds can be achieved. This means that the detector could detect a wider range of wavelengths than of interest but only save those in the visible and near infrared to preserve the integrity of the data while reaching higher speeds.

The pixel size is 16 x 16 $\mu m^2$ which with a pixel array of 512 x 512 determines the size of the imaging area to be 8.2 x 8.2 $mm^2$. This combined with a scan angle of 2.5 °and a spatial resolution of 100 meters means that the field of view of the instrument would be 51.2 km in the cross track dimension.

## 2.5 Instrument Summary

The spectrometer used for this application needs to be easily minimized to fit within a cubesat but have low optical aberrations within the range of 400 to 1000 nanometers. A pushbroom spectrometer in the Offner design is the most easily and inexpensively fabricated to the purposes of being mounted on-board a small satellite. This design eliminates a few optical aberrations and sets some requirements for the fore optics. Fore optics can range in complexity and size, however a basic telecentric focus with chromatic correction is enough to achieve the throughput needed to image ocean water. Water is more difficult to image due to the low reflectivity but potentially contains details which cannot be ignored. Using a back lit, thinned detector is the best way to achieve accurate image values. Using a 512 x 512 pixel array gives more than enough values in the spectral dimension and a wide range in the cross track dimension. These values set the size of the slit between the fore optics and the spectrometer. The width of the slit is the size of a single pixel, and the length is the length of the detector. The detector width as well as the desired spatial resolution and altitude can determine the field of view. This means that the cross track dimension will image slightly over 50 km in each swath.

# Chapter 3

# Algorithm

## 3.1 Introduction

The algorithm designed uses a hypercube of data to identify ships in the image and calculate the size of the ship and the heading, as well as the speed when possible. Real data is used to test the algorithm as well as a test file generated in the format of the real data. Written in Python, the code converts the data to usable arrays and differentiates the pixels containing information about the ship from those containing information about the background water based on a threshold reflectivity value. The threshold value changes with wavelength, so determining the threshold accurately across the spectrum is essential for accurate calculation of size. The pixels of the ship are weighted to determine center of mass and the covariance to calculate the length and breadth. The variances can be used to calculate the angle off of the image axis, which in terms of the image is the heading. Length and breadth multiplied together will give the size of the ship.

The code contains commands to analyze the image to find the speed the ship is traveling. This speed calculation is based on the Kelvin wake which trails out from the ship at an angle. The length of the waves in the Kelvin wake have to be twice the spatial resolution for the wake to be visible in the data and able to be analyzed. The spatial resolution of the data analyzed was too large for any but very fast ships to have been visible. The wake pattern follows a predictable shape which can be replicated to evaluate the speed analysis of the algorithm. The main code itself is contained in Appendix A.

## 3.2 Data Structure

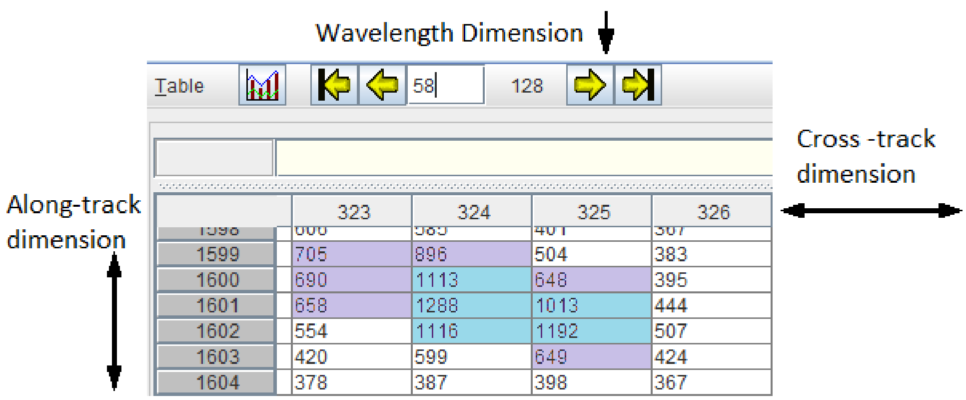
Images sampled from the Hyperspectral Imager for the Coastal Ocean (HICO) [2] were used to derive an algorithm for ship detection and determination of ship parameters. The images used were from regions with long periods of direct sunlight and which have an identifiable ship present. In addition to examining real data, test data was developed in the same format to evaluate the accuracy of the algorithm. The HICO database saves data in two formats, one that is only modified to remove atmospheric effects and some optical errors and the other which also has reduced that wavelength scale to 400 to 900 nanometers, where the instrument is most accurate. The file set used was that with the wider spectral range and fewer modifications. This file set is in an HDF5 format which can be accessed and processed using Python. The HDF5 files are saved as datasets within a single file and include a description of the image and the hypercube, as well as the true color image and details of sensor and solar positioning. Each piece of information is a large scale array and varies between 1 dimensional to 3 dimensional. Python can open these files and sort through and save and read each array individually as necessary.

The Hyperspectral Imager for the Coastal Ocean (HICO) resides on the International Space Station and images swaths 42 by 192 kilometers with around 90 to 100 meter spatial resolution. The spectral resolution of the raw data is 1.9 nanometers, but every three is combined to produce 5.7 nanometer spectral resolution with a higher signal to noise ratio than the raw data alone. The data is scaled in order to condense the size of the files which has to be reversed before working with the data. The data is scaled up so that each data point is an integer to reduce the size of the hypercube.

The HDF5 data include measurements across the wavelength range 350 to 1080 nanometers, so that needs to be cropped although not by much to fit the 400 to 1000 nanometer requirements. The hyper cube is made up of a 2000 x 512 x 128 array of spatial and spectral dimensions and each point is a reflectance of that area. Cropping the spectral dimension to only the values between positions 8 and 115 eliminates the extra values. This cuts the data down to the wavelength range 398.352 nm to 1005.520 nm. The data used had been trimmed to eliminate optical errors along the edges of the image [2]. This left four blank columns and four blank rows at the beginning of the data but does not affect the data all that much besides trimming the spatial dimensions down to 1996 x 508. If there was a ship within that section, it would be cut out, but getting a reliable evaluation of that would be difficult anyway.

To view the HDF5 data more easily and edit the data a program called HDFView can be used to open the files, sort through them, and edit as necessary. Figure 3.1 shows a clip from the program with the hypercube opened and reflectivity values highlighted. The spatial dimensions can be scrolled through as a typical spreadsheet and the wavelength dimension appearing as pages of arrays compiled together. The data is unmodified beyond having atmospheric and optical effects removed, but the values themselves are not the same as will actually be worked with in the code. The blue highlighted pixels are those that are likely to entirely contain image of the ship. The purple highlighted pixels have higher values than the very low values of the majority of the pixels but do not have anywhere near as large of values as the blue highlighted pixels, so they probably contain some of the ship image and some wake and water image. This information is important for getting accurate results on the size of the ship. The light does not perfectly hit a single pixel at a time, but rather spreads over the surrounding pixels in small amounts. The bright spot of the light spreads out to the blue highlighted pixels making the ship appear larger than in reality.

Figure 3.1: HDF5 file with pixels definitely belonging to a ship highlighted in blue and pixels which probably contain some ship and some wake information highlighted in purple




Choosing images with clearly visible ships simplified the identification process. The images below in figures 3.2 and 3.3 are those used to test the algorithm due to the low cloud cover and easily identified land masses. In these images, the ships are bright points set farther out from any other bright areas, with a turbulent wake following after them, although they are very small points.

Figure 3.2: HICO image over the Arabian Sea used to evaluate algorithm [2]

Figure 3.3: HICO image over the Black Sea used to evaluate algorithm [2]

## 3.3 Ship Identification

Before any ships can be analyzed to determine size and speed, they must be identified in the image. Mirghasemi [20] describes a method to analyze the colors present in the image while processing by transforming the basic RGB to a YCbCr color space where the three components are the luminance, the blue difference, and red difference components respectively. This system takes advantage of the patterns that emerge and eases the process of object identification. This is effective but doesn't take advantage of the wide spectral range of information because it is only focused on the data at those three wavelengths. Another method to identifying a ship is by segmenting the image to extract areas which are not a nearly uniform gradient which can be identified as open sea. The ocean does not have much variability, so anomalies are easy to identify. Once the anomalies are detected, ships have to be distinguished from clouds and islands. The hyperspectral image contains very thorough data, making the anomaly detection simple. No prior knowledge of the ship is needed and the algorithm can easily sort pixels which match the radiance of the background water from pixels with different radiance, which can be used to determine ship parameters.

Atmospheric conditions can cause errors in detection which is why satellite data often corrects for atmospheric haze before analysis can occur. The most basic atmospheric correction subtracts a constant digital number across the whole image which assumes a constant level of haze [5]. This is just a first order correction and not quite enough to ensure clear, accurate data. The HICO data used is level 1B pre-corrected, having already had first and second order corrections, including dark-current correction, detector smear correction, and radiance calibration [2].

Some segmentation techniques are unsupervised but they still require some human involvement to ensure that the anomaly is a ship. Rather than segmenting unsupervised, cropping was done entirely manually. The images contain mainly the visible spectrum so the ships which are large enough to be analyzed by the algorithm can be seen on the color image. On the color image, land is identifiable and can be cropped from analysis. Very large clusters of bright spots are unlikely to be ships, although there might be ships among what may be clouds or small islands. To simplify analysis, this part of the data can be cropped until the only visible bright point is the boat itself. From there, the data goes through an anomaly detection process to extract ship data from the ocean data.

## 3.4 Determination of Ship Parameters

After an image of a ship has been verified, the image can be analyzed to determine the heading and speed. To quicken that process, a smaller section of the data which includes the ship is examined rather than processing such a large structure every time. Because the ship does not affect anything past a few kilometers at most around itself, cropping the image is perfectly reasonable and ensures that the determination of ship size is only of the ship rather than any other bright points on the image.

Figure 3.4 shows a 40 by 45 pixel cropping of what appears to be a traveling ship and eliminates the scattered bright points which were evident in the full image. Figure 3.5 is much narrower than the image over the Arabian Sea in order to capture just the values from the single ship. In the larger image, it appears that there is a second ship traveling next to it which adds into the calculations, skewing the center of mass of the bright spot and messing up the results. The algorithm relies on the number of pixels above the threshold, so there is no need to make the cropped image larger because the number of useful pixels does not change.

Using the combination of spatial resolution and pixel size, the length and width of the ship can be estimated. Accuracy is difficult as the white water area around the boat disguises the length, however the bow length is roughly half the ships breadth. Half of the bow length should be added to the estimated length to find the true length. Multiplying the breadth and length together roughly gives the area of the ship. This can be compared to the number of pixels with reflectance values above a minimum threshold which is somewhere between the reflectance of the ship and the reflectance of the background ocean to confirm the size of the ship. There is some difficulty though in determining the values of threshold especially as it changes based on the wavelength examined. To this end, the accurate determination of threshold is essential to accurately determine ship size.

Figure 3.4: Arabian Sea RBG Image cropped down to just the area around the visible ship with wake seen following the ship.

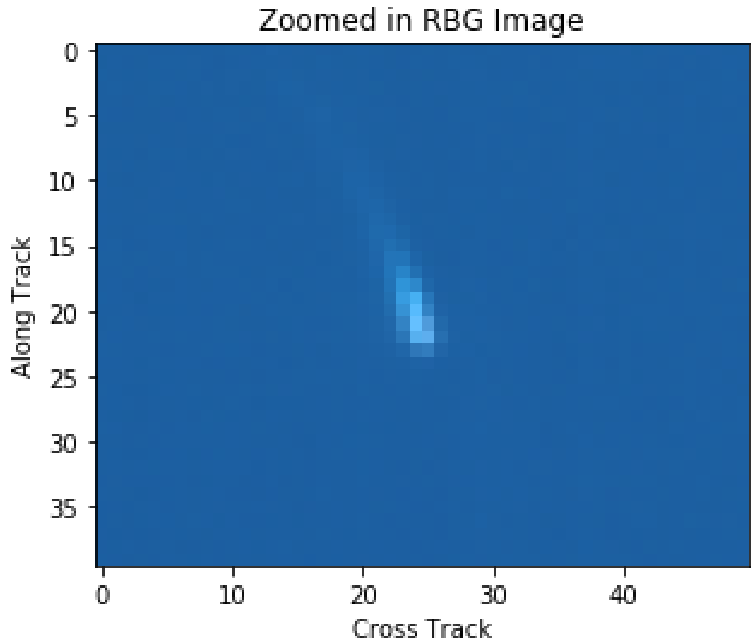
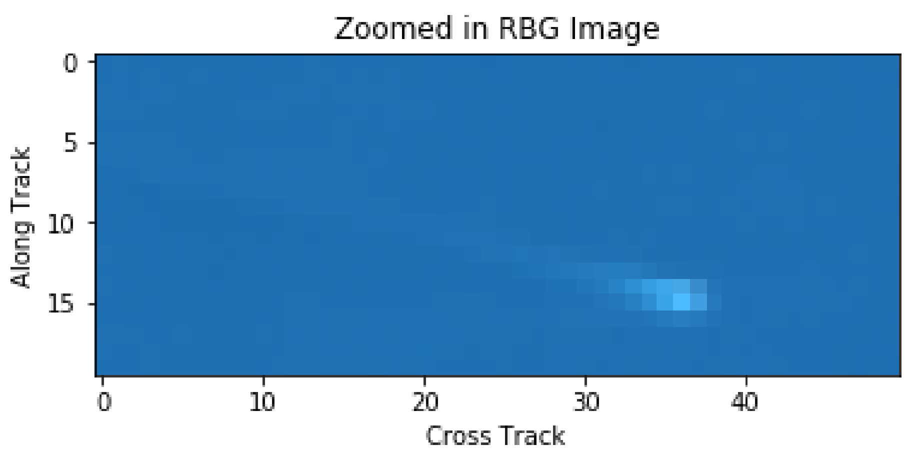


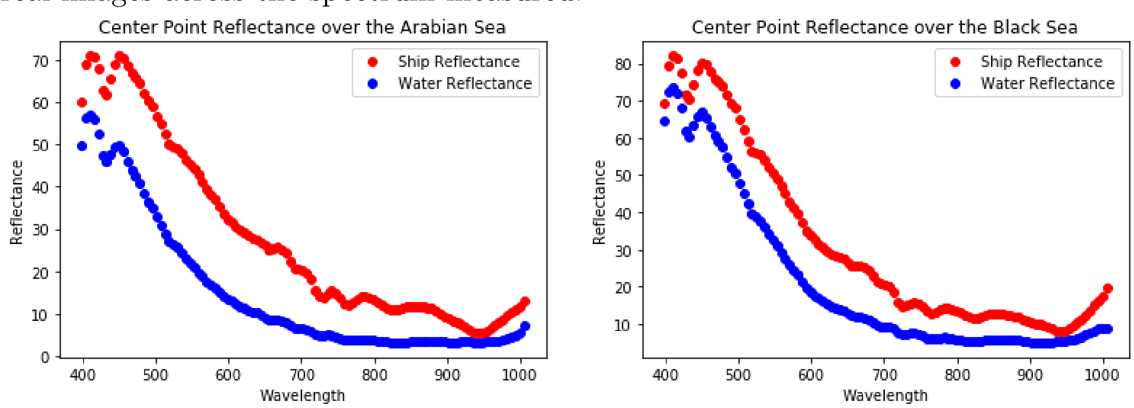

Figure 3.5: Black Sea RBG Image cropped down to just the area around the visible ship with wake seen following the ship.



### 3.4.1 Threshold

When comparing the reflectance of the dark water to the reflectance of the boat across the spectrum, the reflectance of the boat is almost always higher except at very large wavelengths. The threshold for any wavelength could be determined by looking at the number of points which have reflectance greater than a value. Plotting a range of reflectance values against the number of points which exceed that value shows a drop off from the total number of points in the image to the number of pixels which make up the ship and wake to only that of the ship to zero. The difference between the ship and the wake is not always clear around the red wavelength region and into the very near infrared. Closer to the edge of the recorded spectrum, the boat is hardly evident at all. This process gave a good estimate of the point for which wavelengths would more accurately describe the threshold of the ship as well as the threshold of the ship plus wake area. In figure 3.6 the differences between the brightness of the ship and the darkness of the water is apparent, and the threshold is a value somewhere between those values. The point where the ship reflectance is measured is the brightest point on the ship. Setting the threshold to that value excludes pixels which are not purely ship reflectance and therefore not as bright as the center points.

Figure 3.6: Comparison between reflectance of the boat and the dark water in both real images across the spectrum measured.




In the determination of the threshold, the data is analyzed at a specific wavelength and the number of pixels with a reflectance value greater than a value is calculated. Due to the way Python works, the set value iterates by integer values, yet the threshold is still simple to determine. The threshold is a single value, but often there are steps in the data where a few pixels are over a certain threshold and then even fewer of those pixels are over a threshold a few numbers larger. These steps are indicative of the ship and wake. Appendix C contains the code to determine the threshold at any given wavelength. The code looks at a range of threshold choices and returns the number of pixels over each threshold which can be plotted to show the drop off from the total number of pixels down to the number of pixels of the boat, down to nothing.

When evaluating the threshold at low wavelengths, all of the pixels in the detector image have lower reflectance values and then reach a drop off. The drop off is fairly quick as it eliminates the pixels which image the dark water although is not instantaneous. Waves and wakes create white water which raise the reflectance values although not as significantly as the boat itself. The effect of the waves continues to decrease every increment until the threshold of the wake and ship is found. That often forms the first step seen and then a linear decrease to the next step which generally indicates the reflectance of the ship. These steps are not always evident. At certain wavelengths, especially just outside of the visible, there is simply a slow decrease in number of pixels over the reflectance values because the ship and wake are not emitting much differently from the water around it. Fortunately, around 700 nanometers the step for the ship is very obvious and the size of the ship based on the number of pixels can be seen and from that, the threshold at other wavelengths.

The following figures, figure 3.7 show the changes in the threshold over a range of wavelengths. Within the violet to blue region the threshold changes somewhat irregularly and the reflectance is very high overall. The reflectance of the boat is still much higher than that of the surrounding water, resulting in small slopes with long sections at the same number of pixels or only the loss of one or two pixels at a time. The threshold does gradually go down as the wavelength increases but not regularly. The next step of a wavelength could suddenly have a slightly increased threshold but it will not remain high. This area of the spectrum is particularly structured and varies between consecutive wavelengths. Overall though the reflectance slowly starts to drop and with it the threshold.

Across the green, yellow and orange regions, the threshold consistently and evenly decreases as seen in figure 3.8. The threshold of the boat is no longer quite as significantly higher and the slope from the threshold down to zero pixels is not as gradual as in the blue and violet regions.

Within the red and infrared regions, the threshold is nearly constant although in the red region, the threshold for the ship is clear while in the very near infrared, there is very little indication of a boat. Using the red region for the determination

Figure 3.7: Threshold Evaluation in Violet and Blue Wavelengths.

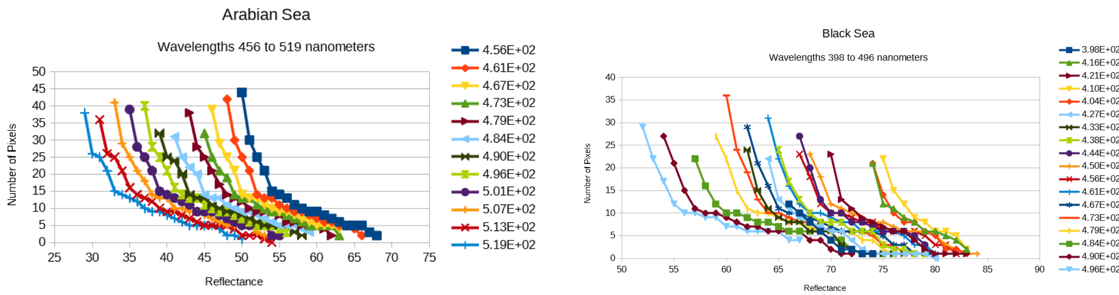
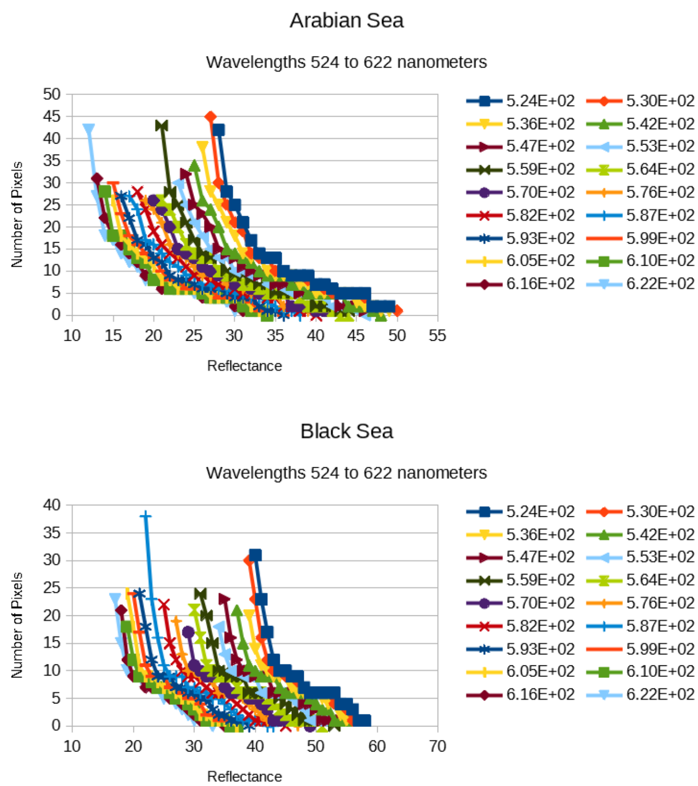



Figure 3.8: Threshold Evaluation in Green, Yellow and Orange Wavelengths.




of ship parameters is very effective because of this clear threshold. The water has a very low reflectance at this point while the ship is still relatively bright. The very near infrared quickly drops from the total number of pixels to none in only a few reflectance values. Neither the water nor the ship emit in this region, however the normalization of data boost the values so that there is some indication of reflectance.

Figure 3.9: Threshold Evaluation in Red Wavelengths.

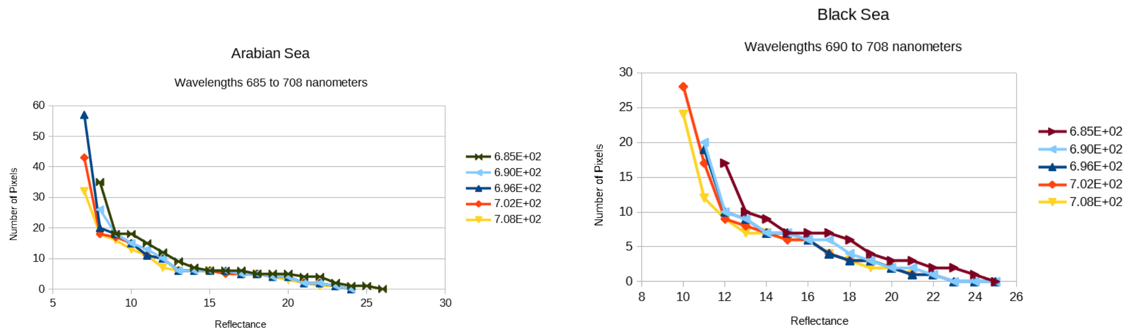



Figure 3.10: Threshold Evaluation in Very Near Infrared Wavelengths.

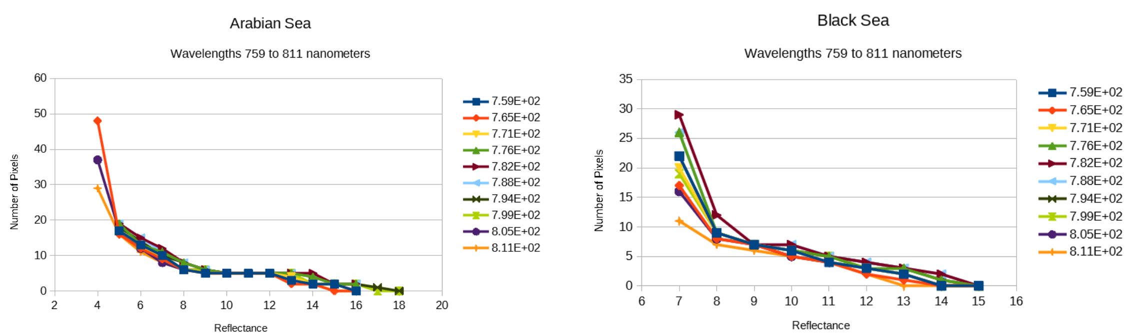
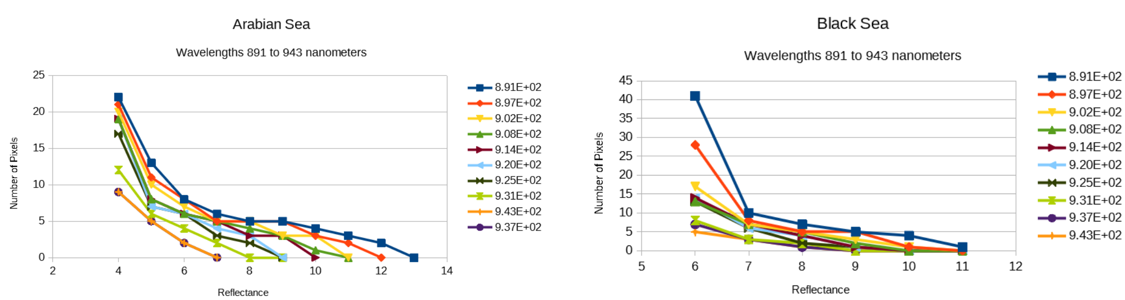


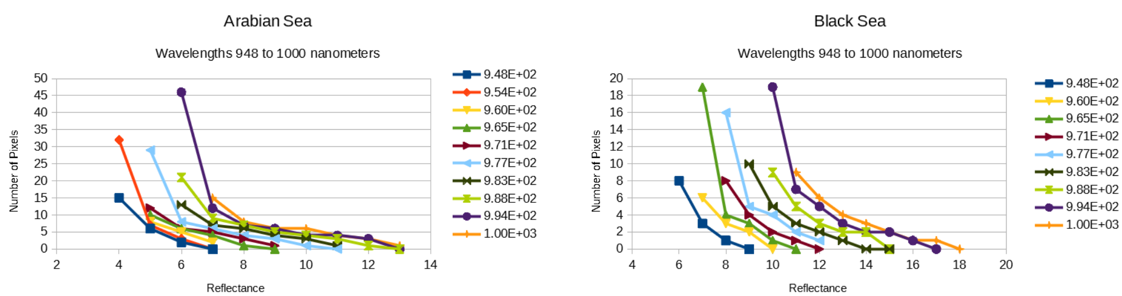

Figure 3.11: Threshold Evaluation in Near Infrared Wavelengths.




Further into the infrared at the edges of the recorded data, the threshold begins to increase again, indicating that there is higher reflectance as the infrared increases.

This edge of the data however is not quite as reliable as it is subject to optical errors. The consistent way the threshold increases appears to be deliberate, so it seems there is some more reflectance in this region although not much. For a better idea of the significance of this region, tests at higher wavelength would be required. It can be assumed that in the thermal region there would be much higher reflectance from the ship, but the evaluation works fine without that information.

Figure 3.12: Threshold Evaluation at the edge of the Data.




### 3.4.2 Area and Heading

Finding only the points which were above the threshold at a certain wavelength required looping through the spatial dimensions to find the pixel coordinates of the boat. These pixel coordinates can be related to physical coordinates by the spatial resolution represented by l. The spatial resolution is not a very stable value and changes during the satellite lifetime and is effected by the spread of the light in the detector. However the accuracy of the calculation does depend on the accuracy of the spatial resolution so if that can be determined it can improve the algorithm. The most basic way to determine the size of the ship is by adding together the number of points over the threshold and multiplying them by the spatial resolution squared [11]. This number can be visually confirmed simply by zooming into the image of the ship and counting bright pixels. The code is flexible to zooming in, simply by setting the boundaries of the space to be analyzed.

$$A_\lambda = l^2 \sum_{i,j}^{>T} 1 \tag{3.1}$$

This method gives a rough idea of the size of the area which is brighter than the surrounding ocean but fails to take into account the pixels which have a mix of ship and water as well as the wake areas. When running the algorithm, this value was always larger than the ship area determined by more in depth means. Because the spatial resolution is so high, each pixel is capable of containing much more than the ship while still reflecting enough to cross the threshold. Using a good threshold value for analysis will ensure that the area due to pixel count is accurate to the rounded up number of pixels. A ship might be even smaller than a pixel, however at that point, detection and analysis of the ship would be nearly impossible and most likely inaccurate even using more intensive evaluation. A smaller spatial resolution would make the simplistic area determination more accurate as well as the advanced determination however would be difficult to achieve using the system described.

To find the total ship reflectance, the total reflectivity has to be calculated by summing the reflectance value of the pixels over the threshold [11]. The reflectance values are not all the same, so this factor minimizes the effect of the mixed pixels so that the algorithm is focused on the ship.

$$I_\lambda = \sum_{i,j}^{>T} I_\lambda(i,j) \tag{3.2}$$

The total ship reflectivity can be used to find the center of mass coordinates based on the pixel coordinates multiplied by the pixel number itself [11]. These values breach the gap between the image and the real ship. In the determination of threshold, the pixel coordinates of the points with high enough reflectivity but not the values. This means that rather than using a built in summing function, code loops through the address values to find the reflectance at each point and adds them together iterating through the addresses. All of the following equations executed the summations in the same loop but require the value for the total ship reflectivity. Once the loop is exited, the rest of the equation is computed using the summation values.

$$\bar{x} = \frac{l}{I_\lambda} \sum_{i,j}^{>T} i * I_\lambda(i,j) \qquad \bar{y} = \frac{l}{I_\lambda} \sum_{i,j}^{>T} j * I_\lambda(i,j) \tag{3.3}$$

Using all of these values, the covariance for each moment can be found which can confirm the length and breadth [11].

$$\sigma_{xx}^2 = \frac{l^2}{I_\lambda} \sum_{i,j}^{>T} i^2 * I_\lambda(i,j) - \bar{x}^2 \tag{3.4}$$

$$\sigma_{yy}^2 = \frac{l^2}{I_\lambda} \sum_{i,j}^{>T} j^2 * I_\lambda(i,j) - \bar{y}^2 \tag{3.5}$$

$$\sigma_{xy}^2 = \frac{l^2}{I_\lambda} \sum_{i,j}^{>T} i * j * I_\lambda(i,j) - \bar{x}\bar{y} \tag{3.6}$$

At this point, the image of the ship needs to be oriented so that the length of the ship lays along the x axis. The degree the ship is rotated is $\theta$. The equation to derive the rotation can be described as follows [11], first by establishing the relationship between the coordinates and the center of mass and then making a few assumptions about the ship. The Python library "numpy" has functions to easily rotate an array or an image by a set angle. The transformed data was not necessary for the calculation of the length and breadth of the ship but came in useful in speed determination.

$$x - \bar{x} = x' \cos\theta - y' \sin\theta \tag{3.7}$$

$$y - \bar{y} = x' \sin\theta + y' \cos\theta \tag{3.8}$$

Assuming the ship is a rectangle implies that $-L/2 < x < L/2$ and $-B/2 < y < B/2$ and the covariances for the rotated $(x, y)$ coordinate system are as follows which can simplify the determination of the heading and ellipticity [11].

$$\sigma_{x'x'}^2 = \frac{L^2}{12} \qquad \sigma_{y'y'}^2 = \frac{B^2}{12} \qquad \sigma_{x'y'}^2 = 0 \tag{3.9}$$

The pixel coordinates need to be adjusted as well to connect the original ship covariances to the oriented covariance, so $i = x/l$ and $j = y/l$, although this can be performed with a simple python command and the pixels over the threshold reestablished. The simplified covariance for each moment allows for the determination of ship size by putting them in term of B, L, and $\theta$. The values of the covariance are calculated using pixels. Solving these equations for the unknowns will result in the desired parameters [11]. Due to rotating the pixel coordinates, the values for the covariances are the same as previously. These assumptions also set the rough size of the ship so that evaluating of the length and breadth of the ship is based on this as well as the ellipticity.

$$\sigma_{xx}^2 = \frac{1}{12}(L^2\cos(\theta)^2 + B^2\sin(\theta)^2) \tag{3.10}$$

$$\sigma_{yy}^2 = \frac{1}{12}(B^2\cos(\theta)^2 + L^2\sin(\theta)^2) \tag{3.11}$$

$$\sigma_{xy}^2 = \frac{1}{12}(L^2 + B^2)(\sin\theta * \cos\theta) \tag{3.12}$$

The heading angle needs to be determined first to find the ellipticity and can be found by solving the three covariance equations for $\theta$ [11]. The angle can be added on to the angle of the satellite swath on the earth in order to determine the direction the ship is actually traveling in reference to north but this was not included in the algorithm.

$$\tan(2\theta) = \frac{2\sigma_{xy}^2}{\sigma_{xx}^2 - \sigma_{yy}^2} \tag{3.13}$$

The ellipticity ($\varepsilon$) classifies the length and breadth ratio and helps identify ships from ocean objects which might not be so elongated, and can be expressed using the covariances already found [11].

$$\varepsilon = \frac{(L^2 - B^2)}{(L^2 + B^2)} = \frac{\sigma_{xx}^2 - \sigma_{yy}^2}{\sigma_{xx}^2 + \sigma_{yy}^2}\frac{1}{\cos(2\theta)} \tag{3.14}$$

This relation means that the length and breadth can be directly solved more accurately than the simple pixel count [11].

$$L^2 = 6(1 + \varepsilon)(\sigma_{xx}^2 + \sigma_{yy}^2) \tag{3.15}$$

$$B^2 = 6(1 - \varepsilon)(\sigma_{xx}^2 + \sigma_{yy}^2) \tag{3.16}$$

Naturally from these equations, the area is the product of the breadth with the length and adjusted to units of $km^2$.

These size values are still only an estimate because the wakes do have an effect on the reflectance and some points will pass the threshold. It also relies entirely upon the given spatial resolution which is idealistic. These values can all be evaluated with a simple Python script and the size of the ship found. Averaging the values across the spectrum gives a much more accurate calculation of the size of the ship in the image. For each evaluation the threshold based on the wavelength used has to be determined. The more wavelengths used, the more accurate the ship area calculation will be. Appendix B contains the code used to average the values because the threshold lists complicates the process. Using a separate code keeps the main analysis concise. The code averages every five wavelength values so there are only 22 values to evaluate rather than 110. It would be possible to simplify this so that all 110 values are considered but not easily.

### 3.4.3 Velocity

Most evaluations of ship speed are based off observation of the Kelvin wake, require previous knowledge of the ship, or are derived from multiple images of the ship over time. Without knowing anything about the ship and having only a single swath image, the only way to calculate ship speed was through the Kelvin wakes. The Kelvin wake occurs regardless of ship type, even including sailboats without a motor. They curve off the boat at an angle of 19.5°and are not affected by strong winds which can change most wave patterns. The Kelvin wave length can be related to the speed of the ship with $g = 9.98\frac{m}{s^2}$, the acceleration due to gravity [11]. The speed cannot always be determined using the spatial resolution of 100 meters which has been set for this instrument and which is difficult to beat using a space borne instrument.

$$\lambda_K = \frac{2\pi V^2}{g} \tag{3.17}$$

The Kelvin wake can be a good way to evaluate the speed only if the wake is visible. On the images used for size calculation the turbulent wake can sometimes be seen but not the Kelvin wake. The turbulent wake is more of an indicator of ocean conditions than anything else although it does leave the trail of where the boat has been. It does not indicate how fast the ship was traveling. To be able to calculate Kelvin wavelength, the spatial resolution has to fall within the Nyquist frequency [31] based on the shortest wavelength the signal contains depending on the dimension.

$$l \leq \frac{1}{2}\lambda_{min} \tag{3.18}$$

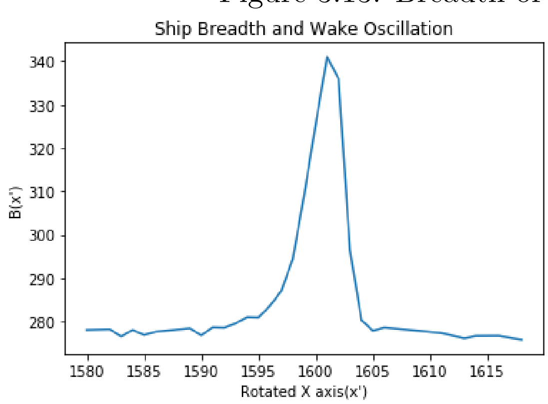
The minimum wavelengths in either direction correspond to the maximum frequencies in the x and y directions. This sets the conditions of whether the speed is possible to determine from the image. Kelvin wakes can get long enough to be visible with the spatial resolution used but would not be larger than one or two pixels. If the boat is moving too slowly or not at all, there would not be much indication although if the boat is still moving, the turbulent wake should be identifiable. Turbulent wake cannot be used to determine speed.

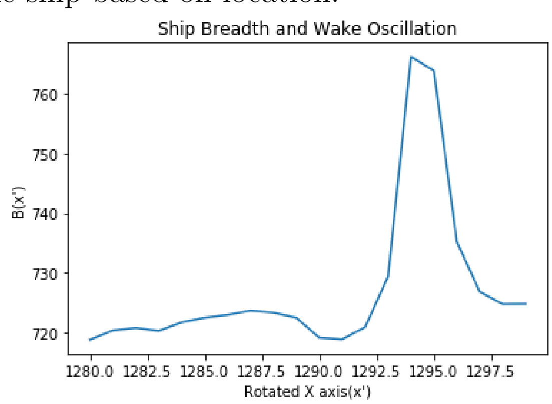
When the Kelvin wavelength is large enough, the breadth can be calculated as a function of the coordinates. Doing this, rather than summing over both the x and y, the reflectance values are summed over only the rotated y directions and calculate the covariance based on the moment in the y direction. With a visible wake, this would make the high points of the wake all the more obvious. The covariance multiplied by the square root of 12 gives a value for the breadth of the ship based on the location in the image. For these calculations, a lower threshold value is used so that white water from the wake can be observed. Plotting the data as in figure 3.13 shows general oscillations and then a large bump indicating the ship itself.

$$\sigma(i)^2 = \frac{l^2}{I_\lambda} \sum_{j}^{>T} j^2 * I_\lambda(i,j) - \bar{y}^2 \tag{3.19}$$

$$B(x') = \sqrt{12} * \sigma(i) \tag{3.20}$$

Figure 3.13: Breadth of the ship based on location.




If there was a visible wake, the oscillation beside the boat would be more defined and a Fourier transform would reveal the frequency of the wake. That spatial frequency is the wavenumber (k) of the wake which can be inverted to find the wavelength of the Kelvin wake. Once the wavenumber is found, the velocity of the boat is easy to calculate.

$$V = \sqrt{\frac{g\lambda}{2\pi}} = \sqrt{\frac{g}{2k\pi}} \tag{3.21}$$

With the spatial resolution of the imager, the ship would have to be traveling at least 17.66 $\frac{m}{s}$ or 34.33 knots which is not impossible but is not common, especially for large ships. For calculation of speed especially, an imager design with a smaller spatial resolution would be desired, or a consideration of the possibility of mounting the instrument on an aircraft.

## 3.5 Results

The part of the algorithm to determine size and heading was evaluated with both the actual hypercubes from the HICO data as well as a test file. The actual hyper cubes do not have usable data to test the part of the algorithm which determines velocity, and attempts to create a test file were mostly unsuccessful. The wake pattern is predictable, so using the algorithm with a function which would mimic the wake results shows the limitations of the velocity calculation.

### 3.5.1 Size and Heading Evaluation

The algorithm can be tested with data directly from the HICO Hypercube but there is no actual information on the visible ships available. The results show the difference between the area estimation from pixel counting and the algorithm calculation. This part of the algorithm was also evaluated with a test file with clear threshold differences between ship and water which is not always as clear in actual images.

The test file used to verify this algorithm has the majority of the pixels following the reflectance spectrum of water and a section of the array with pixels set to the reflectance of the ship across the spectrum. This method ignores the effects of the wake of the calculation of area but can be assumed to be a boat at rest. The previous assumptions used to determine the equations to find length and breadth, means that the test file has to be designed carefully to maintain accuracy. The test file has a rectangle, 5 by 3 pixels, completely straight in the array so there is no rotation necessary.

The length and breadth calculation are based entirely on the ellipticity value because despite being calculated from the covariance values, is a ratio of length to breadth. This value ensures that the object has the dimensions that a ship usually has. The ellipticity of a ship is usually between 4 and 7 [11] which makes the length and breadth calculations more reasonable to a ship.

The heading angle adjusts the image of the ship to be in-line with the x-axis of the image although has not been related to the earth. Each of the HICO HDF5 files contains an array of the latitude and longitude points as well as positioning of the sun and sensor for every spatial coordinate. It would not be a reach to access these files to determine the angle the satellite was traveling over the earth and adding in the calculated heading to determine where the ship is traveling in terms of the earth.

Table 3.1: Ship Size Evaluation

| | Test Case | Arabian Sea | Black Sea |
|---|---|---|---|
| Size based on Pixels | 1.6 $km^2$ | 0.30 $km^2$ | 0.07 $km^2$ |
| Average Size | 1.575 $km^2$ | 0.231 $km^2$ | 0.063 $km^2$ |
| Heading | 0.00° | 44.56° | -0.94° |

The pixel size is never quite the same as the average size, which is to be expected, but is within a pixel of being accurate, verifying the algorithm. Some of the brightness around the ship will be due to the white water that forms around the ship, but using a good value for the threshold will eliminate some of this. The size due to pixels is always rounded up to the full pixels number rather than down because the algorithm takes this full number of pixels and trims that down to the more accurate number. Setting the threshold correctly is what keeps the error low although actually calculating the error is not useful. The pixel calculation will only ever be a full number of pixels and the average size something less than that but more than the the pixels size minus one. So long as this is the case, the algorithm is working.

It is interesting that the test case is not exactly the same, but to keep the size small like a ship would actually be, the ellipticity does not result in accurate breadth and length calculations which alters the size calculation across the entire spectrum.

### 3.5.2 Velocity Evaluation

The algorithm does compute the velocity from the data if the Kelvin wake is visible, however most of the time, it is not. The wake oscillates and slowly decreases in wake amplitude as it gets further from the ship. The wave amplitude slightly changes the reflectivity of the high points of the wake distinguishing them from the low point, leaving a clear path.

Attempting to perform a Fourier transform on the data available shows that there is no distinct frequency in the ripples next to the boat. Testing this across a wide variety of thresholds and wavelengths showed that frequency of the maximum value changed based on the number of pixels greater than the wavelength rather than having any connection to the oscillation in the data. The Python code includes the speed analysis, and with visible Kelvin wakes, should be able to output a value for the velocity of the ship. When given a wavelength of 200 meters which is the minimum wavelength the code outputs the correct speed for that wavelength. This does not test the accuracy of the Fourier transform however.

The plot of breadth across the space should look similar to the absolute value of a damped cosine, though with some noise. Testing the Fast Fourier Transform analysis in Python using this function with the frequency set to 0.0025, half of the wavenumber, the maximum frequency feasible with this instrument results with the appropriate wavelength and speed. A bit of modification is necessary to remove a spike at zero Hz due to the exponential decay and is limited to accuracy in wavenumber to $10^{-4}$ which does affect the accuracy of the velocity calculation, although not by as much at higher frequencies. These are the frequencies that a ship is more likely to be traveling at, if they are going fast enough for the Kelvin wake to be observable. At frequencies below 0.001 Hz the Fourier transform does not present with any significant spikes besides the spike from zero, although the speeds at that wavelength are unrealistic so this is not really a concern.

The real data is not likely to be as clear of an image but would still have an identifiable frequency spike. As the ship speed increases to unreasonable speeds the frequency nears zero and becomes less reliable. This means that with a smaller spatial resolution, the frequency spike will be very visible at a large enough value

Table 3.2: Fourier Analysis Outputs.

| Set Frequency (Hz) | Wavelength (m) | Velocity ($\frac{m}{s}$) | Velocity (knots) |
|---|---|---|---|
| 0.0025 | 200 | 17.82339 | 34.64590 |
| 0.0023 | 217.39 | 18.58216 | 36.12084 |
| 0.0020 | 250 | 19.92715 | 38.73529 |
| 0.0017 | 294.11 | 21.61403 | 42.014325 |
| 0.0015 | 333.33 | 23.00989 | 44.72766 |
| 0.0013 | 384.61 | 24.71659 | 48.04521 |
| 0.0010 | 500 | 28.18125 | 54.77997 |

Figure 3.14: Expected Breadth based on location

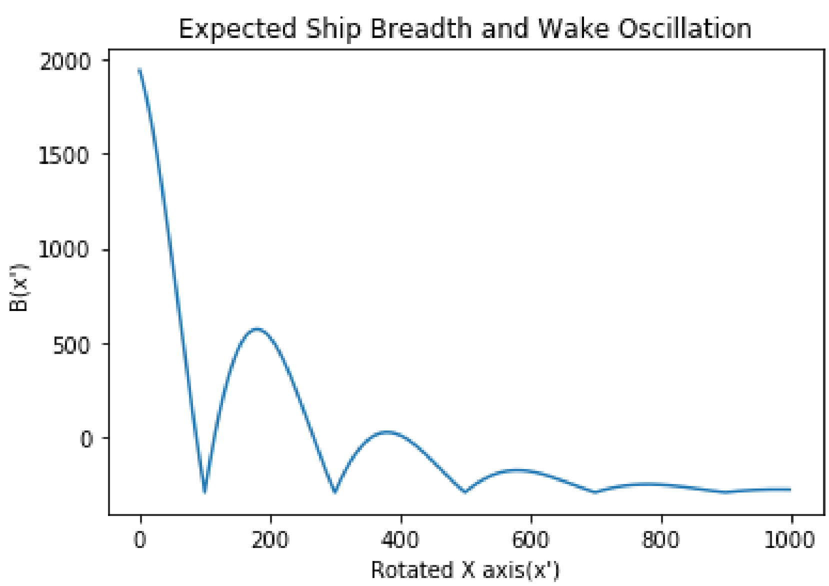
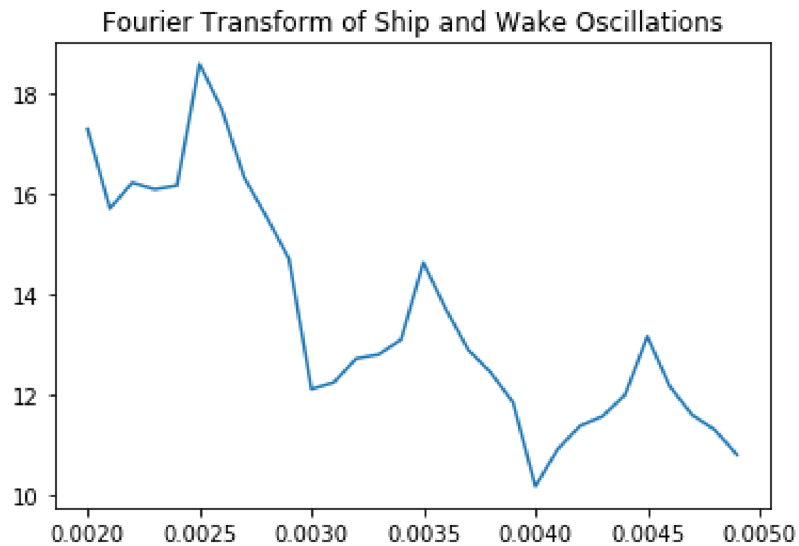


Figure 3.15: Expected Fourier transform results



# Chapter 4

# Discussion

A Hyperspectral Imager for a cubesat can be designed with relative ease using an Offner spectrometer, minimized to take up as little space as possible. The images gathered from sweeps over large bodies of water can be used to identify larger ships and determine how large they are, the direction they are heading, and, if they are going fast enough, determine how fast they are traveling. The algorithm can determine the size based on the number of pixels with a high enough reflectance value. The algorithm weighs the reflectance of each pixel to eliminate the parts of the pixel which contains water rather than purely ship.

Hyperspectral observation of water requires a high spectral resolution and a thinned, back-lit detector array with high sensitivity. There is a lot of possibility for over saturation of the detector, but this detector design avoids much of this error. The difference in reflection of water and land is extreme but the detector has to be able to measure the reflectance of the ship.

Different types of spectrometer were compared to find the best option for minimization and for space applications. While some of the choices were ideal for minimization, they required more modification to ensure an aberration free image which made them larger and more difficult to work with. Others required moving pieces with a high enough sensitivity that could not necessarily be maintained through a rocket launch. Some spectrometers simply could not be minimized to the size necessary for a cubesat and still measure in the visible region.

The algorithm determines which pixels within the hypercube contain the higher reflectance values which belong to the ship. These pixels are summed and weighted to find center of mass of the object and the covariance of the moments in all directions. These values can be used to calculate the length and breadth of the object and from that the size. The length and breadth calculation is most accurate when the object is the shape of a ship because the algorithm contains certain assumptions about the ratio of length to breadth and uses those assumptions in the calculation. Under conditions where there is a visible Kelvin wake, the algorithm can determine the speed. Looking at all of the pixels containing the wake, the pixels can be compiled to emphasize the wave structure. A Fourier transform of this reveals the frequency of the wake and from that the velocity the boat is traveling.

## 4.1 Further Research

The next steps with the cubesat are to actually fabricate the spectrometer and test its performance. Throughout that process, further cubesat details can be determined and the full cubesat assembled and launched. Reasearch into improving the spatial resolution while maintaining the minimal size would further the usefulness of the images for these purposes.

A smaller spatial resolution would increase the accuracy of the algorithm and increase the number of ships that could be identified and analyzed. This would also make the calculation of the dimensions more precise and more reasonable ship speeds measurable. The algorithm is flexible enough to work with any spatial resolution that it is given but is restricted to the data available. Whether that is achieved by larger fore-optics or placing the instrument on board an aircraft, there are methods to increase this parameter.

Besides improving spatial resolution, one of the most basic next steps is to look within the latitude and longitude files to determine which way the ship is heading on the earth itself rather than just relative to the swath of the satellite.

There are methods to make the algorithm more autonomous with the identifying of ships and distinguishing ships from clouds or small islands which would simplify the user involvement. This would be especially helpful when dealing with images with many visible ships. It is possible to do with the algorithm the way it is, but could be made faster if they were identified with less supervision. If the data is accumulated daily, the analysis time could become a major factor.

This algorithm also is working with preprocessed data which may not always be the case, and so adding in a check for whether or not there is some level of processing could be useful. In the case of the data not being processed in anyway, it would not be difficult to add in a dark current subtraction. Some of the other processing that the data had already undergone would require more knowledge about the status of the instrument itself to determine calibration and corrections due to aberrations. The Offner spectrometer eliminates most errors but not all are avoidable. The algorithm could also be used with any set of data so there is all range of possibilities for error, but a lot of opportunity for use.

# Appendix A

# Main Code

```
# -*- coding: utf-8 -*-
"""
Created on Wed Apr 26 08:34:25 2017

@author: Tabitha Koehn
"""

import matplotlib.pyplot as plt
import numpy as np
import h5py
from scipy import ndimage
import math

def threshold(L,k,x1,x2,y1,y2,array): # Distinguish pixels
    above the threshold
    i = x1
    j = y1
    savexy=[i,j]
    for i in range(x1,x2):
        for j in range(y1,y2):
            if array[i,j,k] >= L:      # Array point greater
                than or equal to threshold
                savexy.extend([i,j])  # Saves the
                    coordinates of the pixel
```

```
    savexy = savexy[2:]      # First two values were set to
       the first location coordinates
    x = savexy[::2]          # Separate out x coordinates
    y = savexy[1::2]         # Separate out y coordinates
    return (x,y)

filename = "C:\Temp\Test_file.hdf5"        # Test File
f = h5py.File(filename, 'r+')              # Open HDF5 file
data = f['data']                           # File has multiple
    databases within main file
header = data['header']                    # Sort data within
   database

images = f['images']
true_color = images['true_color']    #2000x512x3 array

navigation = f['navigation']
latitudes = navigation['latitudes']    # 2000x512 array
longitudes = navigation['latitudes']    # 2000x512 array
sensor_azimuth = navigation['sensor_azimuth']    # 2000x512
   array
sensor_zenith = navigation['sensor_zenith']    # 2000x512
   array
solar_azimuth = navigation['solar_azimuth']    # 2000x512
   array
solar_zenith = navigation['solar_zenith']    # 2000x512 array

products = f['products']
lt = products['Lt']      # 2000x512x128 HYPERCUBE array
quality = f['quality']
flags = quality['flags']    # 2000x512 array
slope = 0.02
ltfix = slope * lt[:,:,:]    # 2000x512x128

lamda =np.array
   ([352.528,358.256,363.98398,369.712,375.44,381.168,386.89603,392.624,39
          # Wavelength values corresponding to spectrum
   dimension
```

```
L = 11        # Threshold value
L2 = 3        # To include wake for speed calculation
k = 58        # Pixel number in Spectral dimension

# To determine wavelength of chosen pixel, uncomment line
    below
#print("The wavelength of the pixel is ", lamda[k])

x1 = 1280   # Top boundary of crop
x2 = 1320   # Bottom boundary of crop
y1 = 200    # Left boundary of crop
y2 = 230    # Right boundary of crop

(x,y) = threshold(L,k,x1,x2,y1,y2,ltfix)      # Determine
    threshold coordinates

rbg = np.array(ltfix[:,:,[58,32,22]])          # View RBG image
rbg1= np.array(ltfix[x1:x2, y1:y2, [58,32,22]])  # Cropped
    RBG image
maxrbg=np.amax(rbg)    # scale factor of maximum value
maxrbg1=np.amax(rbg1)

plt.figure(1)
plt.imshow(rbg/maxrbg)  #Full RBG Image
plt.title("Full RBG Image")
plt.xlabel("Cross Track")
plt.ylabel("Along Track")

plt.figure(2)
plt.imshow(rbg1/maxrbg1)     #Zoomed in RBG
plt.title("Zoomed in RBG Image")
plt.xlabel("Cross Track")
plt.ylabel("Along Track")

'''
Calculation of Ship Parameters
'''
```

```
spat_res = .1    # Spatial resolution in kilometers (100m)
# Initialize Parameters
p = 0
shipsum = 0
sigxx1 = 0
sigyy1 = 0
sigxy1 = 0
xcms1 = 0
ycms1 = 0

for p in range(len(x)):      # Loop through pixels over the
    threshold for summations
     XX = x[p]          #pixel coordinate
     YY = y[p]          #pixel coordinate
     shipsum=shipsum+ltfix[XX,YY,k]        # Sum of reflectance
          of all pixels over threshold
     xcms1 = xcms1 + ltfix[XX,YY,k]*XX         # Partial
        equation to find ship cms coordinates (Sum)
     ycms1 = ycms1 + ltfix[XX,YY,k]*YY         # Partial
        equation to find ship cms coordinates (Sum)
     sigxx1 = sigxx1 + ltfix[XX,YY,k]*XX**2     # Partial
        equation to find covariance (Sum)
     sigyy1 = sigyy1 + ltfix[XX,YY,k]*YY**2     # Partial
        equation to find covariance (Sum)
     sigxy1 = sigxy1 + ltfix[XX,YY,k]*XX*YY     # Partial
        equation to find covariance (Sum)


areaship = (spat_res**2) * (len(x)) * 10       #Estimated
    area of the ship in kilometers squared

xcms = spat_res/ shipsum * xcms1          #Ship center of
    mass coordinate x
ycms = spat_res/ shipsum * ycms1          #Ship center of
    mass coordinate y
sigxx = ((spat_res**2)/ shipsum) * sigxx1 - xcms**2          #
    covariance in xx moment
```

```
sigyy = ((spat_res**2)/ shipsum) * sigyy1 - ycms**2          #
    covariance in yy moment
sigxy = ((spat_res**2)/ shipsum) * sigxy1 - (xcms*ycms)      #
    covariance in xy moment

angle = 0.5*math.atan((2*sigxy**2)/(sigxx**2-sigyy**2))
    # Caclulation of the angle of ship off the axis (Heading)
degrees = math.degrees(angle)        # Switch heading angle
    to degrees
rotate_ship = ndimage.rotate(rbg1 / maxrbg1, angle, reshape
    = False)   # Rotate the RBG image to see aligned ship
rotate_image = ndimage.rotate(ltfix, angle, reshape = False)
        # Rotate entire hypercube for velocity calculation
    summing in only one direction

plt.figure(3)         # Plot the aligned ship
plt.imshow(rotate_ship)
plt.title("Image rotated so Ship lays along x axis")
plt.xlabel("x")
plt.ylabel("y")

epsilon = ((sigxx**2 - sigyy**2) / (sigxx**2 + sigyy**2)) /
    math.cos(2 * angle)  # Calculate ellipticity (Length to
    Breadth ratio)

B = math.sqrt(6 * math.fabs(1 - epsilon) * (sigxx**2 + sigyy
    **2))  # Calculate Breadth
Length = math.sqrt(6 * math.fabs(1 + epsilon) * (sigxx**2 +
    sigyy**2))  # Calculate Length

Area = B*Length *1000      #Calculated area of the ship in
    kilometers squared

# Print important results
print(B, "breadth")
print(Length, "length")
print(degrees, "heading angle")
print(Area, "kilometers squared")
```

```
print(areaship, "area based on pixels")

'''
Calculation of Velocity
'''

(x_rot, y_rot) = threshold(L2,k,x1,x2,y1,y2,rotate_image)    #
    Reevaluate threshold with lower threshold to include wake
   , in the rotated hypercube

# Initalization
u=0
v=0
sigy_rot = [0]*len(y_rot)

for u in range(len(y_rot)):  # Loop through for summation
    YYY = y_rot[u]
    sumK=0
    for v in range(len(y_rot)):
        XXX=x_rot[v]
        K=ltfix[XXX,YYY,k]
        sumK=sumK + (K*XXX**2)
        v=v+1
    sigy_rot[u]=sumK*((spat_res**2)/ shipsum) - xcms**2
       # Create list of covariance dependant on x coordinate,
        summed over the y dimension

Bb=[math.sqrt(math.fabs(i*12)) for i in sigy_rot]    #
   Calculate the Breadth across the y dimension

plt.figure(4)         # Plot the Breadth based on location
plt.plot(y_rot, Bb)
plt.title("Ship Breadth and Wake Oscillation")
plt.xlabel("Rotated X axis(x')")
plt.ylabel("B(x')")

# Fourier Transform with given function to replicate wake
   pattern
```

```
Fs = .1      # Sampling Rate
Ts = 1/Fs     # Sampling Interval
t = np.arange(0,1000,Ts)     # Time interval
Bfft =abs(44.5 * np.exp(-t * 0.005) *np.cos(2 * np.pi *
    .0025 * t))  # Expected wake pattern
Bfft = np.subtract(Bfft,np.average(Bfft))  # Remove Zero
    point spikes
fft = np.fft.fft(Bfft)/len(Bfft)     # FFT computation and
    normalization
T = len(Bfft) * Ts
K = np.arange(len(Bfft))
freq = K/(10*T)
freq = freq[range(len(Bfft)//2)]  # One side frequency range
fft = fft[range(len(Bfft)//2)]    # One side fft
fft = fft[2:]    # Eliminate zero spike
freq = freq[2:]

plt.figure(5)          # Plot expected wake
plt.plot(t, Bfft)
plt.title("Expected Ship Breadth and Wake Oscillation")
plt.xlabel("Rotated X axis(x')")
plt.ylabel("B(x')")

plt.figure(6)          # Plot fft
plt.plot(freq, fft.real)
plt.title("Fourier Transform of Ship and Wake Oscillations")

frequency = freq[fft.argmax()]  # Find frequency of the
    maximum value
print("The frequency is ", frequency)
wavenumber = 2 * freq[fft.argmax()]  # Calculate wavenumber
print("Wavenumber is ", wavenumber)
#wavelength = 200
wavelength = 1/wavenumber    # Calculate wavelength
print("Wavelength is", wavelength)

g=9.98  # Acceleration due to gravity
velocity = math.sqrt(g/(math.fabs(wavenumber)*2*math.pi))
```

```
    # Calculation of Velocity
print("The velocity is ", velocity, "m/s")
knots = 1.94384449244 * velocity     # Convert speed to knots
print("Velocity in knots is ", knots, "knots")

f.close()          # Close the hdf5 file
```

# Appendix B

# Average Size

```
import matplotlib.pyplot as plt
import numpy as np
import h5py
from scipy import ndimage
import math

def threshold(L,k,x1,x2,y1,y2,array):
    i = x1
    j = y1
    savexy=[i,j]

    for i in range(x1,x2):
        for j in range(y1,y2):
            if array[i,j,k] >= L:
                savexy.extend([i,j])
            j=j+1
        i=i+1
    savexy = savexy[2:]
    x = savexy[::2]
    y = savexy[1::2]
    return (x,y)

def parameters(x,y,k):  # Calculate parameters based on
   information
```

```
spat_res = .1    # Spatial Resolution in kilometers (100m
    )
# Initalize variables
p = 0
shipsum = 0
sigxx = 0
sigyy = 0
sigxy = 0
xcms = 0
ycms = 0

for p in range(len(x)):
    XX = x[p]          #pixel coordinate
    YY = y[p]          #pixel coordinate
    shipsum=shipsum+ltfix[XX,YY,k]        #reflectance of
        all pixels over threshold
    xcms = xcms + ltfix[XX,YY,k]*XX         #partial
        equation to find     ship cms coordinates
    ycms = ycms + ltfix[XX,YY,k]*YY         #partial
        equation to find ship cms coordinates
    sigxx = sigxx + ltfix[XX,YY,k]*XX**2     #partial
        equation to find covariance
    sigyy = sigyy + ltfix[XX,YY,k]*YY**2     #partial
        equation to find covariance
    sigxy = sigxy + ltfix[XX,YY,k]*XX*YY

xcms = spat_res/ shipsum * xcms          #Ship center of
    mass coordinate x
ycms = spat_res/ shipsum * ycms          #Ship center of
    mass coordinate y
sigxx = (spat_res**2)/ shipsum * sigxx - xcms**2
    #covariance
sigyy = (spat_res**2)/ shipsum * sigyy - ycms**2
    #covariance
sigxy = (spat_res**2)/ shipsum * sigxy - (xcms*ycms)
    #covariance
angle = 0.5*math.atan((2*sigxy**2)/(sigxx**2-sigyy**2))
#degrees = math.degrees(angle)
```

```
    epsilon = ((sigxx**2 - sigyy**2) / (sigxx**2 + sigyy**2)
       ) / math.cos(2 * angle)
    B = math.sqrt(6 * math.fabs(1 - epsilon) * (sigxx**2 +
       sigyy**2))
    Length = math.sqrt(6 * math.fabs(1 + epsilon) * (sigxx
       **2 + sigyy**2))
    return(B,Length)

#filename = "C:\\Temp\\H2011321100743.L1B_ISS.hdf5"        #
   Arabian Sea
filename = "C:\\Temp\\H2014175104231.L1B_ISS.HDF5"     #Black
   Sea
#filename = "C:\Temp\Test_file_2.h5"         #Test File
f = h5py.File(filename, 'r+')
products = f['products']
lt = products['Lt']   #2000x512x128
slope = 0.02
ltfix = slope * lt[:,:,:]     #2000x512x128

# Chosen Wavelength Pixels
k =
   [10,15,20,25,30,35,40,45,50,55,60,65,70,75,80,85,90,95,100,105,110,115]

# The thresholds are not quite the same across the spectrum,
    probably due to quality of water or location of the sun.
   This means that each file would need to have the a list of
    thresholds as well as the coordinates of the ship for
   each file
if filename == "C:\\Temp\\H2011321100743.L1B_ISS.hdf5":   #
   Choose which file
    name = "Arabian Sea"
    x1 = 1580  #Coordinates of Ship per file
    x2 = 1620
    y1 = 300
    y2 = 350
    # Threshold corresponding to each wavelength per file
    L=
       [65,58,60,53,45,38,25,22,19,18,14,12,12,12,13,13,10,8,6,6,7,10]
```

```
elif filename == "C:\\Temp\\H2014175104231.L1B_ISS.HDF5":
    name = "Black Sea"
    x1 = 1280
    x2 = 1300
    y1 = 180
    y2 = 230
    L =
        [81,73,72,63,52,44,35,27,25,23,21,15,14,14,12,12,12,9,8,8,13,17]

elif filename == "C:\\Temp\\Test_file.hdf5":
    name = "Test file"
    x1 = 0
    x2 = 1999
    y1 = 0
    y2 = 511
    L =
        [81,73,72,63,52,44,35,27,25,23,21,15,14,14,12,12,12,9,8,8,13,17]

else:
    print('Find Coordinates and Thresholds for new file and 
        add to the if statement')

# Calculate the pixels above the threshold for each
    wavelength. Becuase it is pulling values from a list, each
    must be done individually
(x00,y00) = threshold(L[0],k[0],x1,x2,y1,y2,ltfix)
(x01,y01) = threshold(L[1],k[1],x1,x2,y1,y2,ltfix)
(x02,y02) = threshold(L[2],k[2],x1,x2,y1,y2,ltfix)
(x03,y03) = threshold(L[3],k[3],x1,x2,y1,y2,ltfix)
(x04,y04) = threshold(L[4],k[4],x1,x2,y1,y2,ltfix)
(x05,y05) = threshold(L[5],k[5],x1,x2,y1,y2,ltfix)
(x06,y06) = threshold(L[6],k[6],x1,x2,y1,y2,ltfix)
(x07,y07) = threshold(L[7],k[7],x1,x2,y1,y2,ltfix)
(x08,y08) = threshold(L[8],k[8],x1,x2,y1,y2,ltfix)
(x09,y09) = threshold(L[9],k[9],x1,x2,y1,y2,ltfix)
(x10,y10) = threshold(L[10],k[10],x1,x2,y1,y2,ltfix)
(x11,y11) = threshold(L[11],k[11],x1,x2,y1,y2,ltfix)
```

```
(x12,y12) = threshold(L[12],k[12],x1,x2,y1,y2,ltfix)
(x13,y13) = threshold(L[13],k[13],x1,x2,y1,y2,ltfix)
(x14,y14) = threshold(L[14],k[14],x1,x2,y1,y2,ltfix)
(x15,y15) = threshold(L[15],k[15],x1,x2,y1,y2,ltfix)
(x16,y16) = threshold(L[16],k[16],x1,x2,y1,y2,ltfix)
(x17,y17) = threshold(L[17],k[17],x1,x2,y1,y2,ltfix)
(x18,y18) = threshold(L[18],k[18],x1,x2,y1,y2,ltfix)
(x19,y19) = threshold(L[19],k[19],x1,x2,y1,y2,ltfix)
(x20,y20) = threshold(L[20],k[20],x1,x2,y1,y2,ltfix)
(x21,y21) = threshold(L[21],k[21],x1,x2,y1,y2,ltfix)

#Calculate the Breadth and Length for each wavelength
(B00,L00) = parameters(x00,y00,k[0])
(B01,L01) = parameters(x01,y01,k[1])
(B02,L02) = parameters(x02,y02,k[2])
(B03,L03) = parameters(x03,y03,k[3])
(B04,L04) = parameters(x04,y04,k[4])
(B05,L05) = parameters(x05,y05,k[5])
(B06,L06) = parameters(x06,y06,k[6])
(B07,L07) = parameters(x07,y07,k[7])
(B08,L08) = parameters(x08,y08,k[8])
(B09,L09) = parameters(x09,y09,k[9])
(B10,L10) = parameters(x10,y10,k[10])
(B11,L11) = parameters(x11,y11,k[11])
(B12,L12) = parameters(x12,y12,k[12])
(B13,L13) = parameters(x13,y13,k[13])
(B14,L14) = parameters(x14,y14,k[14])
(B15,L15) = parameters(x15,y15,k[15])
(B16,L16) = parameters(x16,y16,k[16])
(B17,L17) = parameters(x17,y17,k[17])
(B18,L18) = parameters(x18,y18,k[18])
(B19,L19) = parameters(x19,y19,k[19])
(B20,L20) = parameters(x20,y20,k[20])
(B21,L21) = parameters(x21,y21,k[21])

# Average the values together to find Average Breadth and
    Length
B=(B01+B02+B03+B04+B05+B06+B07+B08+B09+B10+B11+B12+B13+B14+
```

```
    B15+B16+B17+B18+B19+B20+B21)/len(k)
Length=(L01+L02+L03+L04+L05+L06+L07+L08+L09+L10+L11+L12+L13+
    L14+L15+L16+L17+L18+L19+L20+L21)/len(k)

print(name)
print(B, "breadth")
print(Length, "length")
Area = B*Length       #Calculated area of the ship in
    kilometers squared
print(Area, "kilometers squared")

f.close()
```

# Appendix C

# Determination of Threshold Value

```
import h5py
import matplotlib.pyplot as plt

def threshold(L,k,x1,x2,y1,y2,array):
    i = x1
    j = y1
    savexy=[i,j]
    for i in range(x1,x2):
        for j in range(y1,y2):
            if array[i,j,k] >= L:  #Check if pixel value is
               greater than or equal to Threshold
                savexy.extend([i,j])
    savexy = savexy[2:]    # Skip intial setup values
    x = savexy[::2]          # Sort out x coordinate values
    y = savexy[1::2]         # Sort out y coordinate values
    return (len(x),len(y))

filename = "C:\\Temp\\H2011321100743.L1B_ISS.hdf5"         #
   Arabian Sea
#filename = "C:\\Temp\\H2014175104231.L1B_ISS.HDF5"  #Black
   Sea
f = h5py.File(filename, 'r+')
products = f['products']
lt = products['Lt']  #2000x512x128
```

```
slope = 0.02
ltfix = slope * lt[:,:,:]    #2000x512x128

k=114     #Choose wavelength
#Initalize parameters
laxis=[]
lenxaxis=[]
lenyaxis=[]
#Coordinates based on file
if filename == "C:\\Temp\\H2011321100743.L1B_ISS.hdf5":
    name = "Arabian Sea"
    x1 = 1580
    x2 = 1620
    y1 = 300
    y2 = 350
elif filename == "C:\\Temp\\H2014175104231.L1B_ISS.HDF5":
    name = "Black Sea"
    x1 = 1280
    x2 = 1300
    y1 = 180
    y2 = 230
else:
    print('Find Coordinates for ship in new file and add to
       the if statement')

for L in range(8,17):     #Create range of thresholds to
   examine
    (x,y) = threshold(L,k,x1,x2,y1,y2,ltfix)
    laxis.append(L)
    lenxaxis.append(x)
    lenyaxis.append(y)

plt.plot(laxis, lenxaxis,'bo')   #Plot the number of pixels
   at each threshold value
print('wavelength number',k)
plt.title("Threshold Determination for " + name)
plt.xlabel("Reflectance")
plt.ylabel("Number of points")
```

# Bibliography


[1] *The Compact High Resolution Imaging Spectrometer (CHRIS): the future of hyperspectral satellite sensors. Imagery of Oostende coastal and inland waters*, 2004.

[2] Hyperspectral imager for the coastal ocean. Oregon State University, 2009.

[3] M. J. Barnsley, J. J. Settle, M. A. Cutter, D. R. Lobb, and F. Teston. The proba/chris mission: a low-cost smallsat for hyperspectral multiangle observations of the earth surface and atmosphere. *IEEE Transactions on Geoscience and Remote Sensing*, 42(7):1512–1520, July 2004.

[4] Adrian J Brown, Brad Sutter, and Stephen Dunagan. The marte vnir imaging spectrometer experiment: Design and analysis. *Astrobiology*, 8(5):1001–1011, 2008.

[5] Pat S. Chavez. An improved dark-object subtraction technique for atmospheric scattering correction of multispectral data. *Remote Sensing of Environment*, 24(3):459 – 479, 1988.

[6] Michael P Chrisp. Convex diffraction grating imaging spectrometer, March 9 1999. US Patent 5,880,834.

[7] Emmanuel Devred, Kevin R Turpie, Wesley Moses, Victor V Klemas, Tiffany Moisan, Marcel Babin, Gerardo Toro-Farmer, Marie-Hélène Forget, and Young-Heon Jo. Future retrievals of water column bio-optical properties using the hyperspectral infrared imager (hyspiri). *Remote Sensing*, 5(12):6812–6837, 2013.

[8] John Fisher and Wayne C Welch. Survey and analysis of fore-optics for hyperspectral imaging systems. In *Defense and Security Symposium*, pages 62062R–62062R. International Society for Optics and Photonics, 2006.

[9] Mark A. Folkman, Jay Pearlman, Lushalan B. Liao, and Peter J. Jarecke. Eo-1/hyperion hyperspectral imager design, development, characterization, and calibration. *Proc. SPIE*, 4151:40–51, 2001.

[10] Luis Guanter, Vctor Estells, and Jose Moreno. Spectral calibration and atmospheric correction of ultra-fine spectral and spatial resolution remote sensing data. application to casi-1500 data. *Remote Sensing of Environment*, 109(1):54 – 65, 2007.

[11] Henning Heiselberg. A direct and fast methodology for ship recognition in sentinel-2 multispectral imagery. *Remote Sensing*, 8(12):1033, 2016.

[12] International Society for Optics and Photonics. *Compact low-distortion imaging spectrometer for remote sensing*, 1998.

[13] International Society for Optics and Photonics. *Low-distortion imaging spectrometer designs utilizing convex gratings*, 1998.

[14] International Society for Optics and Photonics. *Compact visible/near-infrared hyperspectral imager*, 2000.

[15] International Society for Optics and Photonics. *Dyson spectrometers for infrared earth remote sensing*, 2008.

[16] Seo Hyun Kim, Hong Jin Kong, Jong Ung Lee, Jun Ho Lee, and Jai Hoon Lee. Design and construction of an offner spectrometer based on geometrical analysis of ring fields. *Review of Scientific Instruments*, 85(8):083108, 2014.

[17] Christine M. Lee, Morgan L. Cable, Simon J. Hook, Robert O. Green, Susan L. Ustin, Daniel J. Mandl, and Elizabeth M. Middleton. An introduction to the {NASA} hyperspectral infrared imager (hyspiri) mission and preparatory activities. *Remote Sensing of Environment*, 167:6 – 19, 2015. Special Issue on the Hyperspectral Infrared Imager (HyspIRI).

[18] Robert L. Lucke, Michael Corson, Norman R. McGlothlin, Steve D. Butcher, Daniel L. Wood, Daniel R. Korwan, Rong R. Li, Willliam A. Snyder, Curt O. Davis, and Davidson T. Chen. Hyperspectral imager for the coastal ocean: instrument description and first images. *Appl. Opt.*, 50(11):1501–1516, Apr 2011.

[19] Michael Marshall and Prasad Thenkabail. Advantage of hyperspectral eo-1 hyperion over multispectral ikonos, geoeye-1, worldview-2, landsat etm+, and modis vegetation indices in crop biomass estimation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 108:205–218, 2015.

[20] Saeed Mirghasemi, Hadi Sadoghi Yazdi, and Mojtaba Lotfizad. A target-based color space for sea target detection. *Applied Intelligence*, 36(4):960–978, 2012.

[21] Carlos Montero-Orille, Xesús Prieto-Blanco, Héctor González-Núñez, and Raúl de La Fuente. Design of dyson imaging spectrometers based on the rowland circle concept. *Applied optics*, 50(35):6487–6494, 2011.

[22] Pantazis Mouroulis, Robert O. Green, and Daniel W. Wilson. Optical design of a coastal ocean imaging spectrometer. *Opt. Express*, 16(12):9087–9096, Jun 2008.

[23] Pantazis Mouroulis and Michael M McKerns. Pushbroom imaging spectrometer with high spectroscopic data fidelity: experimental demonstration. *Optical Engineering*, 39(3):808–816, 2000.

[24] Anne Nolin. Satellites and sensors: How they work. University Lecture, 2003.

[25] Jay S Pearlman, Pamela S Barry, Carol C Segal, John Shepanski, Debra Beiso, and Stephen L Carman. Hyperion, a space-based imaging spectrometer. *IEEE Transactions on Geoscience and Remote Sensing*, 41(6):1160–1173, 2003.

[26] Linlin Pei, Bin Xiangli, Qunbo Lv, and Xiaopeng Shao. Optical system design of the dyson imaging spectrometer based on the fery prism. *Optical Review*, 23(4):695–702, 2016.

[27] X Prieto-Blanco, C Montero-Orille, B Couce, and R De la Fuente. Analytical design of an offner imaging spectrometer. *Optics Express*, 14(20):9156–9168, 2006.

[28] Princeton Instruments. *HProEM-HS:512BX3*, 2017. Rev. P4.

[29] Brad Sutter, Adrian J Brown, and Carol R Stoker. Visible–near infrared point spectrometry of drill core samples from río tinto, spain: Results from the 2005 mars astrobiology research and technology experiment (marte) drilling exercise. *Astrobiology*, 8(5):1049–1060, 2008.

[30] Guofeng Wu, Jan de Leeuw, Andrew K. Skidmore, Yaolin Liu, and Herbert H.T. Prins. Performance of landsat {TM} in ship detection in turbid waters. *International Journal of Applied Earth Observation and Geoinformation*, 11(1):54 – 61, 2009.

[31] Zhijian Wu. On the estimation of a moving ship's velocity and hull geometry information from its wave spectra. Technical report, DTIC Document, 1991.

[32] Mr Devvrat Yadav and Mr Shashikant Agrawal. Keystone error correction method in camera-projector system to correct the projected image on planar surface and tilted projector. *International Journal of Computer Science & Engineering Technology*, 4(2):142–146, 2013.

[33] Xiaodong Zhang, Marlon Lewis, W Paul Bissett, Bruce Johnson, and Dave Kohler. Optical influence of ship wakes. *Applied optics*, 43(15):3122–3132, 2004.

[34] Z. Zou and Z. Shi. Ship detection in spaceborne optical image with svd networks. *IEEE Transactions on Geoscience and Remote Sensing*, 54(10):5832–5845, Oct 2016.