NTNU – Trondheim
Norwegian University of
Science and Technology

On-board, Fourier-Based Image-Analysis System for Satellite Observation of Gravity Waves

Morten Hatlen

Master of Science in Physics and Mathematics
Submission date: June 2013
Supervisor: Patrick Joseph Espy, IFY

Norwegian University of Science and Technology
Department of Physics

Abstract

The main objective of this thesis is obtaining data on gravity waves from images taken by the NTNU Test Satellite (NUTS). The satellite's main payload consists of a camera intended to capture images of gravity waves in the atmosphere. A limitation of the satellite is its ability to transmit data. Two separate approaches have been investigated to limit the amount of data that needs to be sent. One is the degree of image and video compression that can be applied to a series of images before data is corrupted. The second is implementing a 3D-Fast Fourier Transform (FFT) method that obtains the relevant data in the image series directly and only sends these.

The NUTS-satellite is a project still in the making with a preliminary launch date in 2014. Currently there are no comparable, usable images of the wave phenomenon captured from space. Ground based images taken with fish-eye lenses are therefore used as a basis to make emulate satellite images. These are used to test how satellite images will be influenced by compression and how the 3D-FFT can extract the data.

We found the 3D-FFT able to extract wave parameters from the images, providing significant reduction in the need for transmission of data compared to sending compressed images. In addition we found that the compression-approach corrupts data before a considerable reduction in transmitted data is achieved.

Sammendrag

Fokusområdet til denne oppgaven er å hente ut informasjon om tyngdebølger fra bilder tatt av NTNUs Test Satellitt (NUTS). Nyttelasten til satellitten er et kamera som skal ta bilder av tyngdebølger i atmosfæren. En begrensning ved satellitten er evnen til å sende data. To forskjellige metoder har blitt undersøkt for åbegrense mengden med data som måsendes. En er graden av bilde- og videokompresjon som kan gjennomføres før dataene om bølgene blir forstyrret. Den andre er en 3D-Fast Fourier Transform (FFT) som henter ut de relevante dataene direkte fra bildene og sender kun disse.

NUTS-satellitten er et prosjekt som fremdeles på g år, med en foreløpig oppskytningsdato i 2014. For øyeblikket finnes det ingen tilsvarende bilder av bølgefenomenet tatt fra rommet. Bakkebaserte bilder tatt med en vidvinkellinse brukes derfor som et grunnlag for åetterligne satellittbilder. Disse brukes for åteste hvordan satellittbildene vil bli påvirket av kompresjon, samt hvordan 3D-FFT-metoden kan hente ut data.

Vi fant at 3D-FFT metoden var i stand til åhente ut bølgeparametre fra bildene, som vil gi en betraktelig reduksjon i mengden med data som måsendes sammenlignet med åsende komprimerte bilder. I tillegg fant vi at kompresjonsfremgangsmåten påvirker bølgeparameterne før en vesentlig reduksjon i sendt data blir oppnådd.

Preface

The work presented in this text is my master thesis, done as a part of the fifth and final year of a masters degree at the Department of Physics at the Norwegian University of Science and Technology. The work has been supervised by Patrick Joseph Espy, and has been carried out in the period August 2012 to June 2013. The work has been directed towards the NTNU Test Satellite-group, and the weekly progress was reported to the group and its supervisor Roger Birkeland.

I would like to thank Patrick Espy for good guidance and insightful discussions, both on- and off-topic.

16th June, 2013 Morten Hatlen

Contents

1 Airglow

The airglow is a phenomenon observed as a weak emission of radiation from the atmosphere. The airglow is easiest observed during nighttime and is then called nightglow.

The airglow presents itself as an approximately 8 km thick shell of light in the mesosphere. The radiation stem from chemical reactions that occur once the sun sets and stops splitting molecular oxygen to atomic oxygen. When the molecules are split, the products get a kinetic energy equal to the original binding energy. When the atoms recombine back to O2, or take part in other chemical reactions, this energy excites the product. This will later relax via radiation and is observed as nightglow. The nightglow consists of several spectrum of radiation from several different mechanisms. The one interesting for the NUTS-project is the OH-spectrum, created by hydrogen reacting with O3. This is a layer which radiates best at a height of approximately 87 km.

The satellite will be taking images of the airglow as it passes over the part of earth where it's currently dark, as the glow will be easier to observe when the much more intensive radiation from the sun is not present. The airglow itself can often be considered a disturbance to some measurements, as its radiation is masking radiation from deep space. This makes earth based observation of space more difficult, and gives rise to the need of satellites such as Hubble. In the NUTS' case though, the airglow will serve as a medium to observe a different phenomenon, gravity waves.

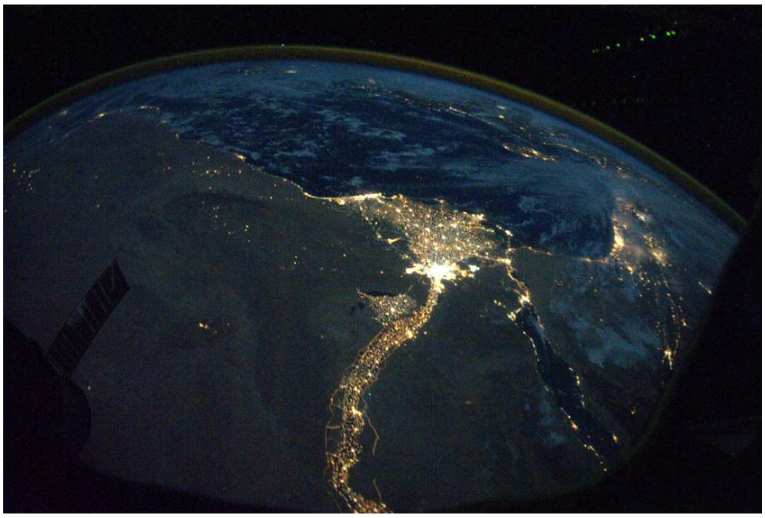


Figure 1: Picture taken from ISS [1]. Airglow showing as green sphere above horizon.

2 Gravity Waves

Atmospheric gravity waves are internal waves, and are in some ways similar to the waves known from the interface between water and air. When the density discontinuity between two layers of fluid is disturbed, the forces of gravity and buoyancy act to restore the equilibrium. Atmospheric waves differ from waves in water in the way that the density of air varies throughout the inspected height, as opposed to the sharply defined interface of water and air. Despite this, Archimedes' principle still applies, and less dense matter will rise over more dense matter. The forces overshoot and the resulting oscillation create propagating waves. The atmospheric gravity waves are named such because gravity and buoyancy are the restoring forces. These waves are important because they influence momentum and energy transport in the atmosphere. They can produce turbulence and trigger convection as well as significantly affect the atmospheric mean flow. The perturbations needed to initialize the oscillations can come in many forms, but wind passing over mountains is one of the most common sources.

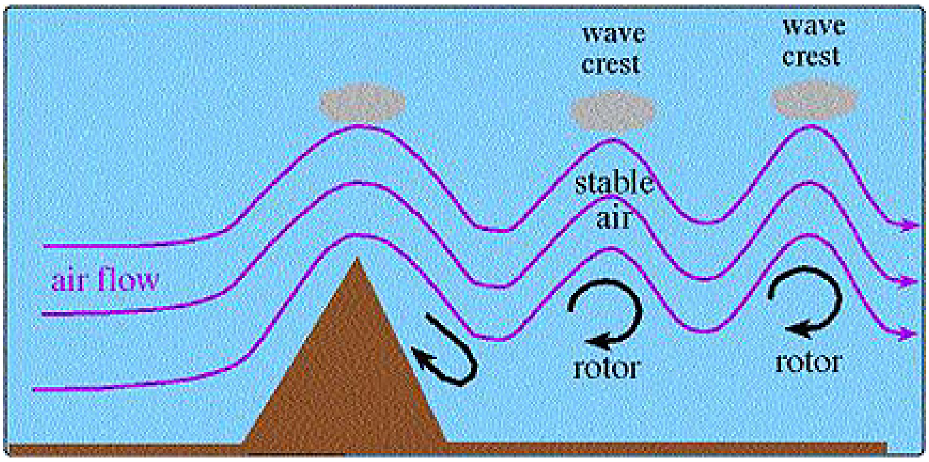


Figure 2: Simplified illustration of how gravity waves are made [2]

Observation of gravity waves is not trivial since they propagate through transparent air. However, as the waves propagate vertically and horizontally in the atmosphere, they will hit the airglow. When this happens the waves will perturb the relative uniform distribution in the layer. Some parts of the layer will ascend while others descend, in accordance with the wavelength. This creates a variation in the radiation intensity corresponding with the wavelength. By observing the varying intensity of the airglow, wavelengths can be found.

The currently available data on gravity waves are limited to localized measurements from the ground. Two examples of measuring techniques are radar tracking and fish-eye lens cameras with appropriate filters. These methods can provide good data on how the waves behave in a certain area over time. They are limited in only covering a fraction of the earth's atmosphere. This is why a satellite is needed, as it can orbit the earth and provide global coverage. This will be a requisite for learning more about momentum transfer in the atmosphere and will help construct more accurate models of climate and weather.

These waves should not be confused with the more elusive gravitational waves, ripples in the curvature of space-time predicted by Albert Einstein.

3 The NUTS Project

Currently, there is a group at The Norwegian University of Science and Technology dedicated to specifying, designing and building a small satellite following the CubeSat standard. The CubeSat standard for small satellites was developed by California Polytechnic State University and Stanford University. The work begun in 1999 and today the main requirement for a single CubeSat is that it should be of size $10x10x10cm^3$ and weigh no more than 1.33kg. The NUTS satellite will be a double CubeSat of size $20x10x10cm^3$ and have a weight limit of 2.66kg. One can use ordinary of-the-shelf electronics within the cube. The standard outer dimensions of the satellite makes launch from the carrier rocket much easier as only one standard ejection mechanism is needed. The small size makes is possible for the CubeSats to hitch a ride with considerably larger satellites being launched into space, making a much cheaper launch into orbit. Combined with this and an easy to follow standard, CubeSat makes space more accessible for smaller companies and universities. The main payload of the satellite, the camera, is intended to take pictures of the earth's atmosphere. More specifically the influence of atmospheric gravity waves on the airglow during night, the nightglow. If successful, the images will provide data previously unavailable on the same scale. There is a multitude of people from different fields involved in the project. The following text will mainly focus on the image analysis, processing and compression, although other aspects of the satellite are equally important to investigate in order to gain usable data. Some of these are; design of software, attitude control, hardware specification, and design of layout. The goal of the NUTS project is both educational and scientific. If the project is a success it will provide useful data on gravity waves and airglow. If this fails, the involved students have gained useful experience none the less.

3.1 Similar Projects

A similar project to the NUTS project is the SwissCube. This was also a student satellite project intended to take images of the atmosphere. The launch was successful and the satellite was able to transmit images taken with the onboard camera. Unfortunately, the images were not suitable to extract data. NASA planned to launch a satellite in 2007 intended to explore the different wave structures in the atmosphere. The project was however canceled due to lack of funds.

At the moment there are many ground based imagers who collect data on the gravity waves, but these do not provide a global coverage since they are stationary.

3.2 Limitations of the Satellite

There are limitations to the NUTS-satellite. Regarding the images, the main concern after they have been obtained is transmission back to earth. According to estimates in Marianne Bakken's master thesis [3],the satellite can transmit only 5Mb per day, which means that a series of 10 images can take up to 2 days to download. This is not sufficient to create a global set of data on the gravity waves. The limitations of the satellite's transfer rate give rise to the need for compression. A compression algorithm suited for the satellite has already been designed [3], but not tested and optimized for the pictures it will take. As there are no available, useful, images of the phenomenon taken from space, synthetic sine wave images has been used to test the compression algorithm.

3.3 Other Challenges

In addition to the challenges presented in this text, there are several others not concerned with the processing of images. The following is a summary of some of them.

3.3.1 ADCS

The Attitude and Determination Control System is essential to stabilizing the satellite once it has been ejected from the rocket carrying it to orbit. If the satellite was simply ejected, it would tumble its way around earth making it close to impossible to capture usable images. The satellite will be equipped with several electromagnetic coils in order to stabilize the satellite and keep the payload, the camera, pointed towards earth. The coils will serve as both sensors and actuators. The sensing is done by measuring the induced currents in the coils by the earth's magnetic field. With knowledge of the magnetic fields surrounding our planet attitude determination is possible. A voltage is imposed on the coils resulting in the coils establishing a magnetic field of their own. This interacts with the earth's magnetic field, resulting in a torque on the satellite and the attitude is possible to control.

3.3.2 BIT Flip

The satellite will not be shielded by the earth's atmosphere in the same way that our standard electronic equipment is. It is therefore exposed to the cosmic radiation of space. This can cause single bits in the satellite's on-board computer to flip, i.e. a zero being replaced by a one. This can be an insignificant change in the image files, or it could be a change which affects the operation of the satellite. There is therefore need for making scripts for detection, and protection against, bit flips.

3.3.3 Design

Designing the physical layout of the satellite is also a challenging task. The satellite is following the CUBESAT standard. This gives a list of specifications and standards to follow in order to ease the transport and launch from the rocket. This means that all components need to fit within a certain volume in a rectangular cuboid shape and not weigh more than a certain limit.

3.3.4 Gases

The satellite will probably be hitching a ride as a secondary payload on rockets carrying far bigger, more sophisticated and more expensive satellites. These could carry sensitive equipment and sensors that should not be contaminated.

As equipment is carried into space, the drop in pressure could cause them to start leaking gases otherwise contained within their material. The satellite will contain a lot of electronic equipment, so the various gases expected can be many. In order to prevent this emission of gases the satellite will be baked at high temperature in a low pressure chamber for a long time. This will remove most of the gases from the material.

4 On-board Image Processing

There are many challenges associated with the process of obtaining useful satellite images. Many of these are mentioned by Marianne Bakken in her master thesis[3]. In the thesis, simulations and discussions proposes a method that should be the one most suited for the needs of the NUTS satellite. In the NUTS case there are two main aspects to obtaining good satellite images, besides the purely technical challenges of designing and launching a satellite into orbit. The two aspects are getting clear images with a high signal to noise ratio, and then transmitting the data back to earth. When it comes to transmitting the data, two approaches is presented. The first is compressing the images in order to transmit as many as possible. The second is identifying the wave properties on-board, and just transmit this data. This is achieved through a 3D Fast Fourier Transform, and represents a great reduction in the amount of data that has to be transmitted.

4.1 Signal-to-noise-Ratio

Infrared sensors, such as the one intended to be used on-board the satellite are prone to influence from external sources which can introduce noise to the signal. A high signal to noise ratio (SNR) is important not only to the quality of the images themselves, but also the efficiency of the chosen compression algorithm.

4.2 Dark Current

Dark current is the small current that appears in the sensor of the camera, even when there are no photons hitting it. The source of this current is thermally generated charges. In order to eliminate the effects of the dark current, each series of images will start off with one image recorded with the shutter shut. This will give an image of only the effects of the dark current, shielded from the photons usually hitting the sensor. When the effects of dark current have been recorded, this image can be subtracted from the open shutter images. Thus the effects of dark current will be eliminated from the images the data will be extracted from.

4.3 Shot Noise

The nature of light is governed by statistical quantum fluctuations, which will become evident if the exposure time of an image is short. This can best be illustrated by an example. Consider tossing a coin a handful of times, the distribution of heads and tails will probably be shifted towards either heads or tails. If one then continues throwing a large number of tosses, this shift will eventually even out to a insignificant difference. The same applies in a camera if the exposure time is short. Only a relatively few photons are allowed to hit the sensor, and the resulting image will not give a correct representation of the actual distribution of the photons. A way to eliminate this problem is to increase the time each image is exposed to light. This will on the other hand give rise to a new problem, as described below.

4.4 Motion Blur and Image Averaging

Getting images with a good SNR is a trade-off between having a long exposure time and the consequent blur which occurs in the images. The blur is due to the fact that the satellite will move in its orbit while the shutter is open. Several methods to mitigate this balancing act were investigated by [3], before one was decided to be better than the others. Two methods investigated were to use the Lucy-Richardson algorithm on the image or do a deconvolution, based on knowledge of how fast the satellite would move in its orbit. It was however discovered that this method could be too sensitive to the variations that could occur in the speed of the satellite. If the speed was not correct, more blur could be introduced by the method. This was in addition to the fact that the methods would be computationally demanding and complex to implement. The method decided to be the most suited for the satellite's need is one based on motion compensation and image averaging. Instead of taking images with long exposure times, and then later process the images to remove the resulting blur, images with short exposure times taken at short intervals will be used. These images will individually have a low SNR, but if the images are averaged to combine one single image the SNR will increase. An important requirement for this to work is that the movement between each image is compensated. This method will not be as sensitive to inaccuracies in the estimated speed of the satellite and can therefore employ a simple scheme to "move" an image over the previous one. This short series of images will be combined to one image, which again will be a part of the series of images the satellite should send back down to earth.

4.5 Bit Binning

SNR is of importance when it comes to obtaining good images. In order to increase this ratio a process of binning together pixels, effectively reducing the resolution, has been proposed. The information held within a certain number of pixels will be combined to a smaller pixel, reducing the influence of random noise in each separate pixel. Binning the pixels together will also reduce the amount of data that has to be transmitted. The data transfer rate, as previously mentioned, is limited and binning will therefore mean that more images can be sent. One has to be careful not to reduce the resolution to such an extent that wave structures no longer will be distinguishable. A different way to reduce the amount of data that is to be sent is to utilize the previously mentioned compression algorithm.

4.6 The Compression Algorithm

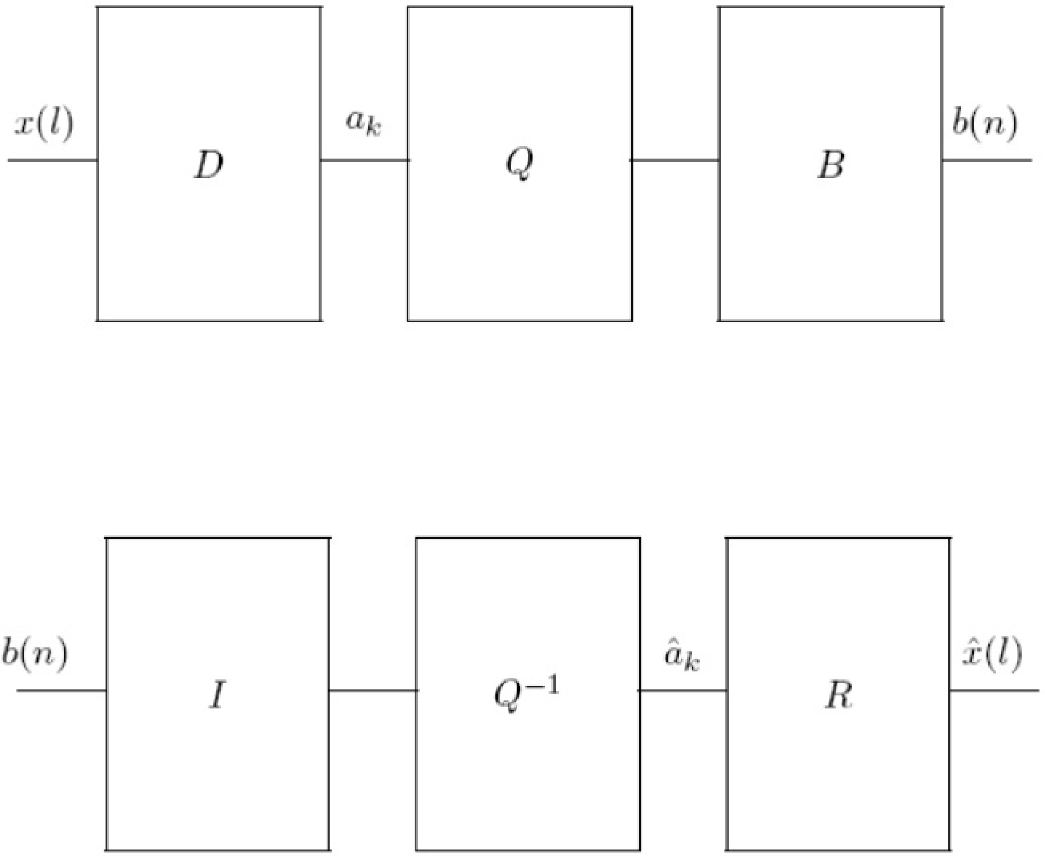
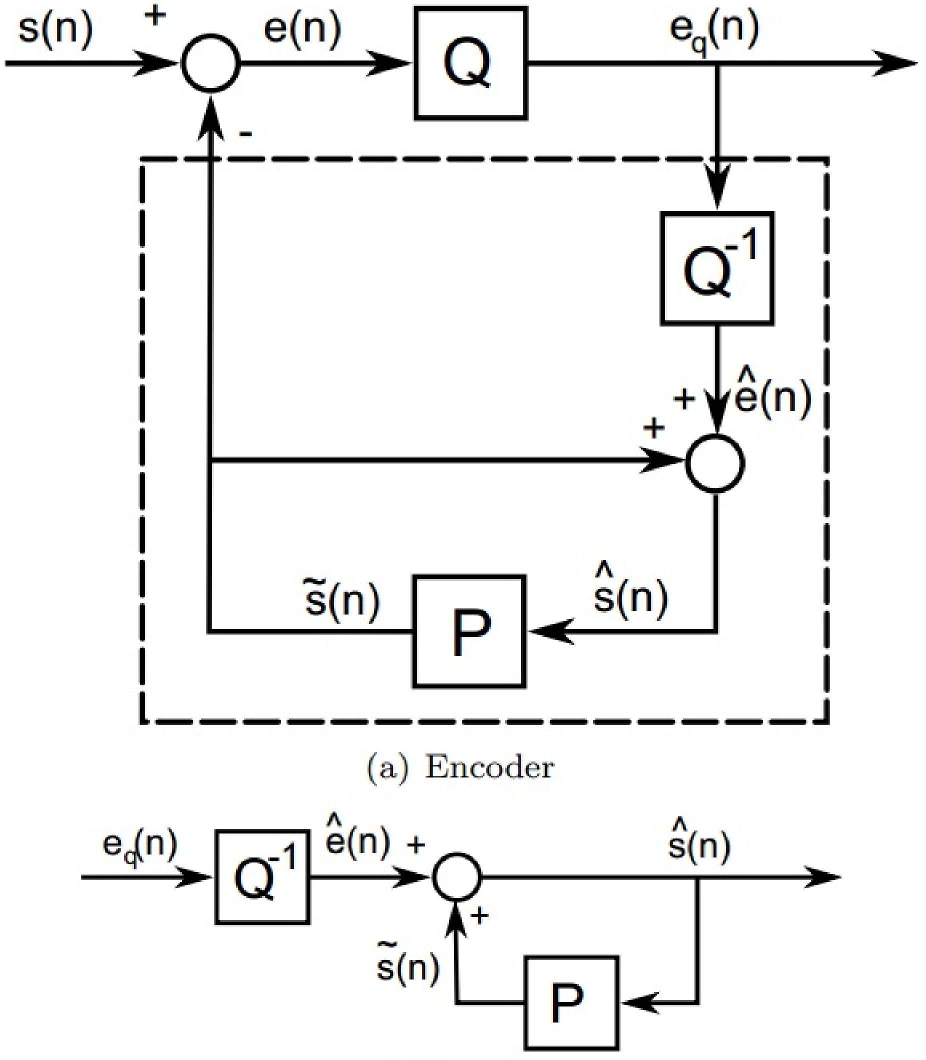


Figure 3: Schematic of a generic compression algorithm

The compression algorithm designed for the satellite is a 3DPCM method, a three dimensional differential pulse code modulation. The three dimensions in this case consists of the two dimensional images and the added dimension of successive images. The satellite will capture images in series, which can be considered a low rate video. Images taken at a sufficient rate will provide a good overview of the waves and their development. Hopefully there will be limited difference between the successive images, giving a better basis for compression.

The chosen method is a lossy one, meaning that some of the data will be lost in compression and recovery. In some applications this would not be acceptable, but since we are compressing images some loss is acceptable. Determining how much loss of image data can be tolerated before the extracted wavelengths are influenced is the main goal of the methods described in the following sections of this text.
For a more comprehensive explanation by the author of the compression algorithm, view [3]. In order to provide a simplified explanation of the compression algorithm used, one can start by investigating a simpler, generic compression



(a) Encoder

(b) Decoder

Figure 4: Schematic of DPCM compression algorithm, dashed line box represent D-block in figure 3

algorithm. This algorithm can be considered to consist of six blocks as seen in figure 3. First there is a block, D, which decomposes the data into a series of coefficients. This block also includes a prediction of the next data point. So if the data is highly correlated, prediction will be easier and the algorithm will work better. The next box, Q, is one which quantizes the data to some predetermined levels. This is where the loss of data occurs. Imagine a pixel which originally can display 16 different shades of grey. A quantization of its data to four levels means that four different shades are now represented by one single shade. The third box, B, is concerned with coding the bits in the signal in the most favorable way. The chosen bit coding for this application is stack run coding. It is outside the scope of this report to determine if others could be more efficient. The most efficient bit coding will be dependent on signal to noise ratio in the images used. The following three boxes, I, Q^{-1} and R, are inverses of the ones already mentioned and come in reversed order to reconstruct the original signal. The operation of the three first boxes will be carried out by the satellite and represent the compression. The three last boxes represent the conversion back to images from the transmitted bits, in our case carried out by the terrestrial receiver.

The suitable number of levels to quantize the output of the D-block in the Q-block remain as one of the unknowns in the implementation of the compression algorithm. Choosing the number of levels is one of the parameters determining how much the data will be compressed. Another parameter is the dead-zone loading, which decides how large the level that represents zero should be. At some level, the reconstructed image will be too distorted to recover any useful data. How quantizers with different parameters could look is illustrated in figure 5 and figure 6.

The supplied compression algorithm was written to handle square images only and consequently had to be altered in order to handle the resolution of the satellite.

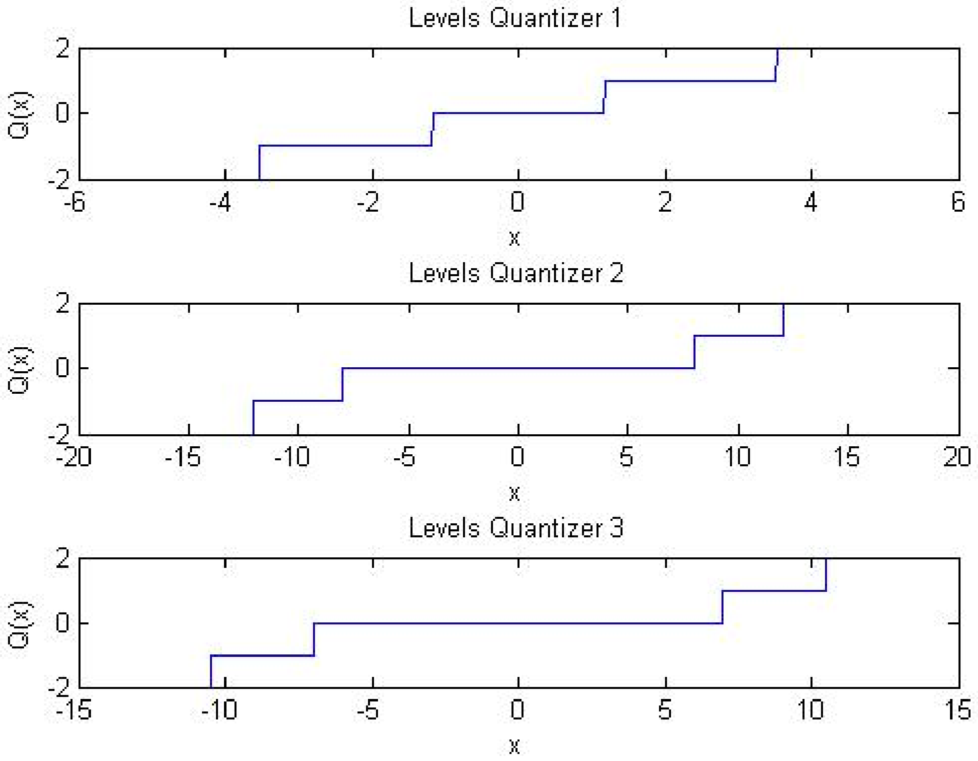


Figure 5: 5 levels and dead-zone loading equal 2.5

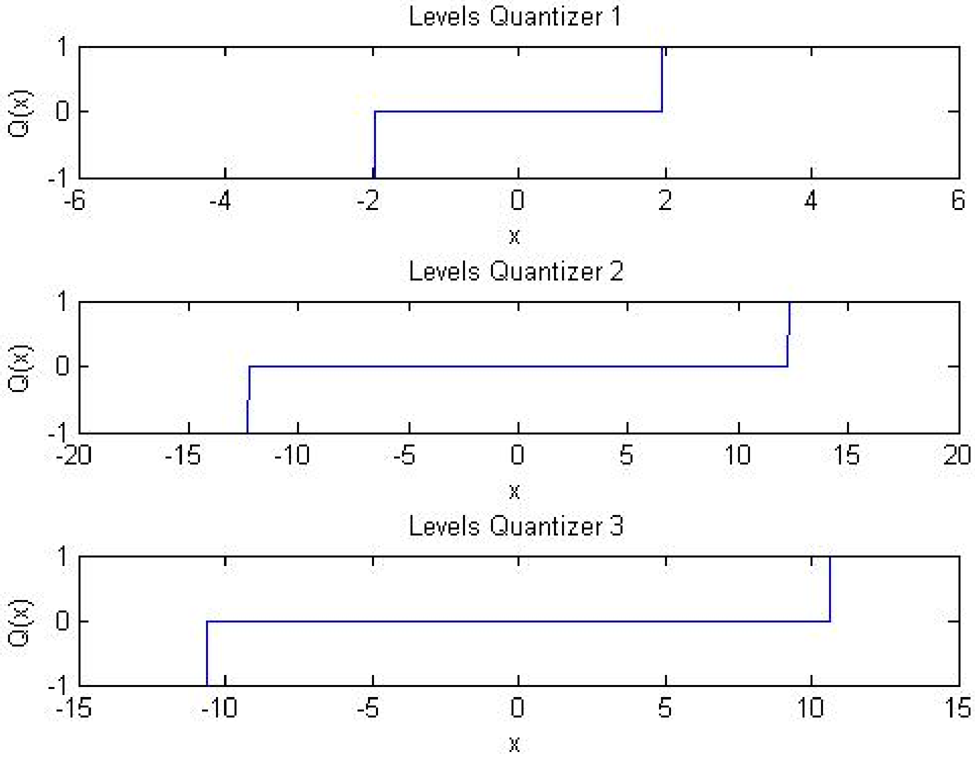


Figure 6: 5 levels and dead-zone loading equal 3.8

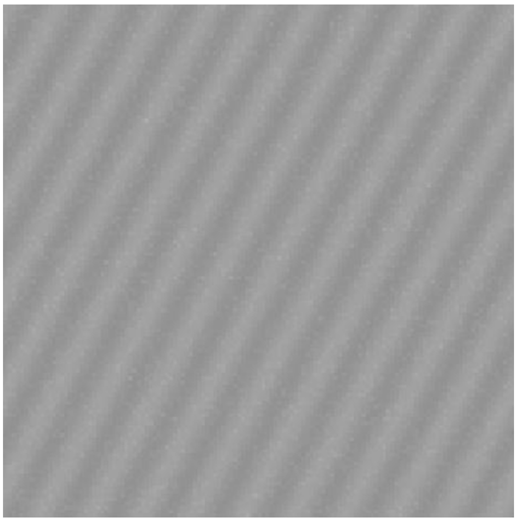


Figure 7: Sine wave test image, uncompressed

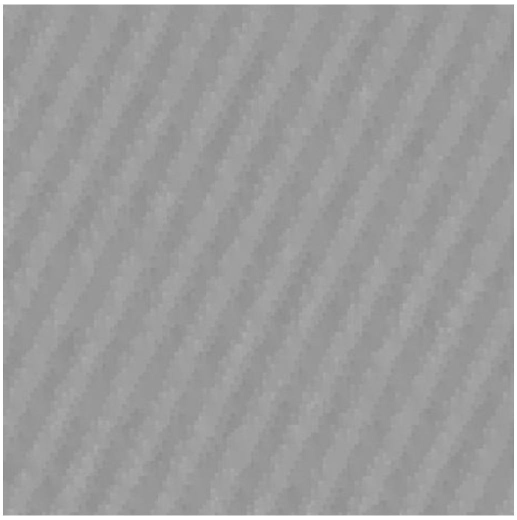


Figure 8: Sine wave test image, compressed

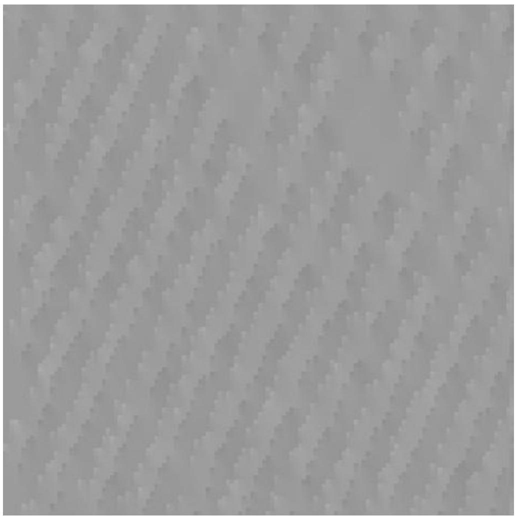


Figure 9: Sine wave test image, highly compressed

The images seen in figure 8 and 9 show how a simulated image of a wave structure is distorted with increasing level of compression, using the quantizers from figure 5 and figure 6 respectively. The wave structure is deteriorating, ultimately making recovering the wavelength more difficult.

5 Obtaining Test Images

The satellite's compression algorithm has previously been tested on simple synthetic images of a sine wave with some noise added. In order to establish a reasonable level of compression for the satellite, one should test the compression algorithm on more realistic images. As there are no pictures taken in space of the gravity waves' impact on the nightglow, ground based images will serve as a substitute. During the fall of 2012 a ground based imager was installed close to NTNU Dragvoll. This imager includes several filters and a fish-eye lens which enables the setup to take pictures of the nightglow over Trondheim and the surrounding horizon. These images, taken with a fish-eye lens, are distorted and will not give a correct presentation of the wave structures. Image processing is needed in order to obtain the correct data. The following steps will describe a method, inspired by Garcia *et al.* [4], to obtain a geographic projection of the images. This will make it possible to find the correct wavelength of the waves and then test how the waveforms stand up to compression.

5.1 Mapping to Geographical Coordinates

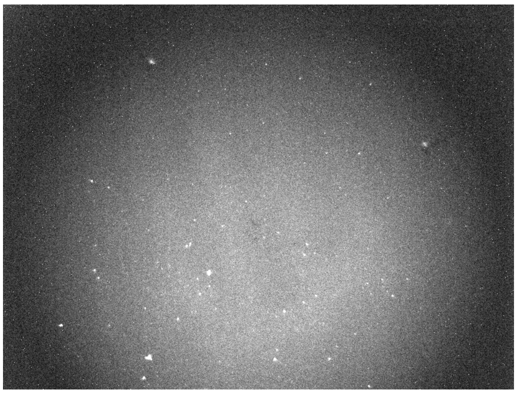


Figure 10: Fish-eye image of the sky above Dragvol, 90° field of view

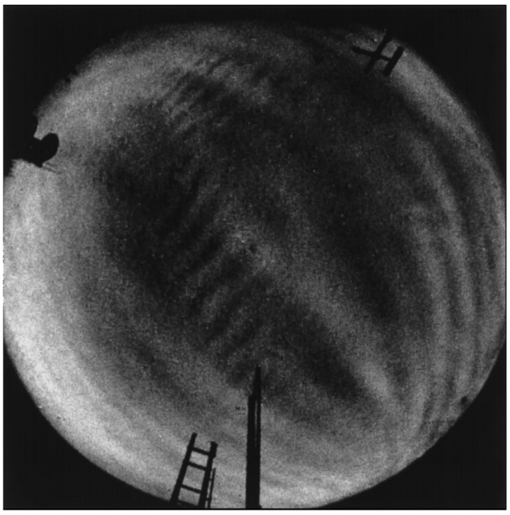


Figure 11: Fish-eye image of the sky above Halley, Antartica, 180° field of view

The use of a fish-eye lens introduces a distorting effect on the images and the correct wavelength can only be obtained after reversing this effect. Why the distorting effect appears is evident if the geometry of the problem is inspected.

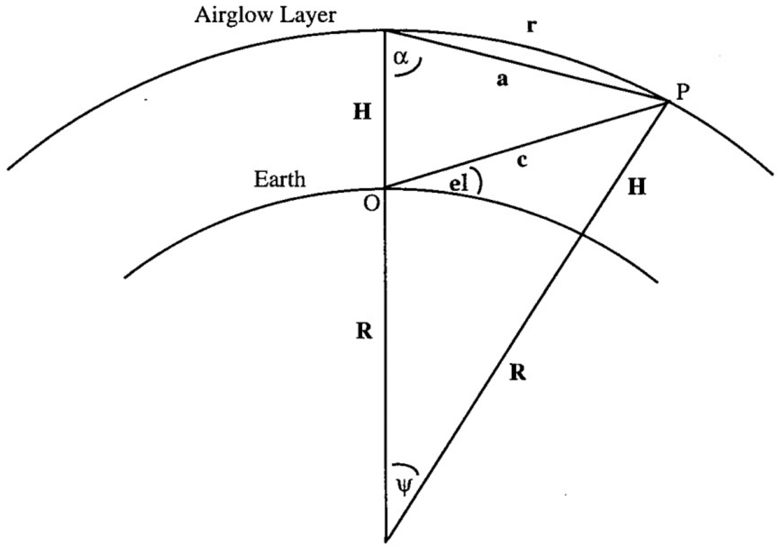


Figure 12: Cross section of earth and airglow layer, not to scale. Figure adapted from [4]

Figure 12 shows how the different parameters in the problem are related to each other. Please note that the height of the airglow layer is greatly exaggerated compared to the radius of the earth. The geometry presented in this figure will give the relationship between the elevation, which is the viewing angle between the horizon and a point in the airglow, and r, the length in the airglow traversed between zenith and the same point.

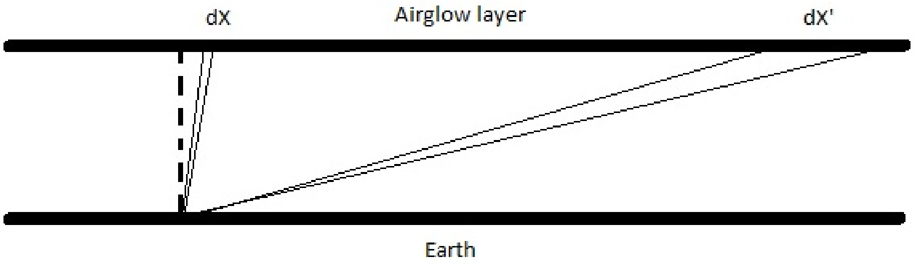


Figure 13: Cross section of earth and airglow layer, on a smaller scale

Although Figure 12 gives the correct relationship, it is not as efficient when it comes to visualizing the effect seen in the all-sky images taken with fish eye lenses. Figure 13 is drawn with the height of the layer and the curvature of earth more to scale. This better illustrates the effect seen in the images. At high elevation angles a change $d\theta$ in angle gives a relatively small dX, while the same $d\theta$ at lower angles gives a much larger dX'. What this translates to in the all-sky images is that pixels towards the edges of the data, which corresponds to low elevation angles, depict larger areas of the airglow layer than the pixels located towards the center of the image, which corresponds to higher elevation angles. The relationship between pixels in the image and degrees covered is assumed to be linear.

The exact relationship between the different parameters in figure 12 is found by [4] as the following:

$$r = \sqrt{x^2 + y^2} \tag{1}$$

$$\psi = r/(R + H) \tag{2}$$

$$a = 2(R + H)\sin(\psi/2) \tag{3}$$

$$c = \sqrt{(H^2 + a^2 - 2Ha\cos(\alpha)} \tag{4}$$

$$\alpha = (\pi - \psi)/2 \tag{5}$$

$$el = \arccos(\frac{a\sin\alpha}{c}) \tag{6}$$

A plot of radius from zenith in the airglow to a point the airglow as a function of the elevation to said point is included in figure 14. This relationship is the essence of what is needed to create a geographically mapped image from fish-eye images. In the image to be made, the geographically mapped image, the pixel radius and azimuth from the center pixel is readily available. Assuming a linear relationship between pixels and degrees covered in the fish-eye image one can find the corresponding pixel radius in the fish-eye image using the relationship illustrated in figure 14. Azimuth in the fish-eye image is also readily available. This will make it possible to iterate through the pixels in the image we want to make and find the corresponding indices in the fish-eye image. As this index not necessarily will be an integer an interpolation scheme is also included.

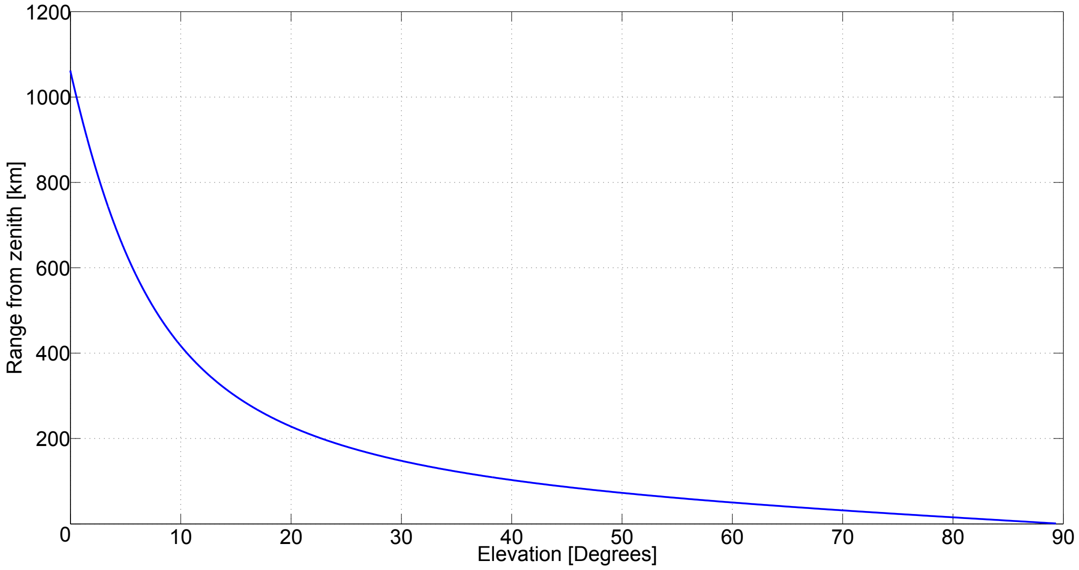


Figure 14: Radius in airglow image as a funtion of elevation

Figure 16 illustrate how the described method will fetch data from pixel within an area approximately equal to the the circle in figure 15 and place them in a new grid according to the geometrics described earlier.

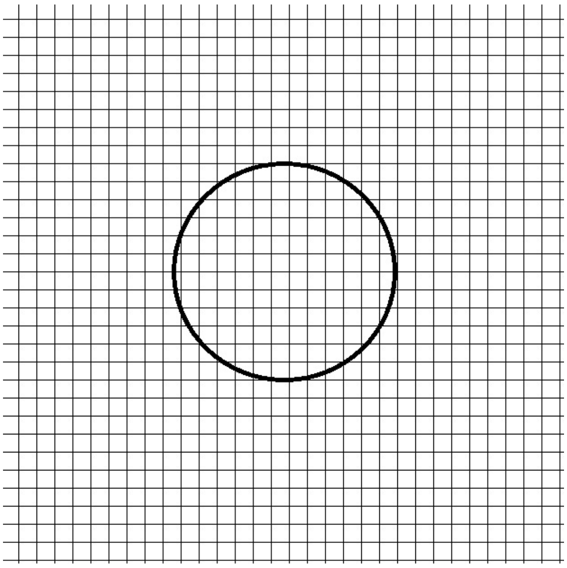


Figure 15: Grid representing pixels in ground based image. Circle representing approximate area from where data is collected

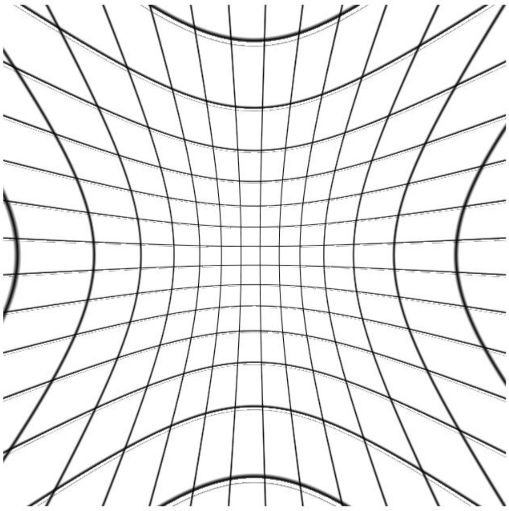


Figure 16: Grid representing what happens to pixels within circle in ground based images in figure 15 when mapped to geographical coordinates

5.2 Finding Zenith

The center in the mapped image will be defined to zenith. The radius from that pixel to the current pixel in the iteration gives us information about which pixel in the fish-eye image the values should be fetched from. However, as the ground based camera isn't perfectly centered the center pixel in the fish-eye image will not display true zenith. Several factors can cause this, but the main causes are believed to be that the hut containing the camera is not perfectly level and the fact that the weight of the camera setup bends the structure holding it in place. So the pixel that displays zenith needs to be found. This will give the needed shift in order to sample the correct values in the fish-eye image. The zenith-pixel is also needed to find the correct ratio between elevation and pixels covered in the fish-eye image. This is needed to find the correct relationship between pixel radiuses in the two images.

Once this calibration has been done, the relationship is fixed and can be used for all images taken with the setup. This calibration will be done with the aid of stars visible in an image taken with the lens in question. With the exact knowledge of location and time the image was taken, one can find a corresponding star map that will tell the location of the stars in the sky at that moment. This data and the location of the stars in image can then be compared, making it possible to make a relationship of placement of pixels in the image and their true location in the sky.

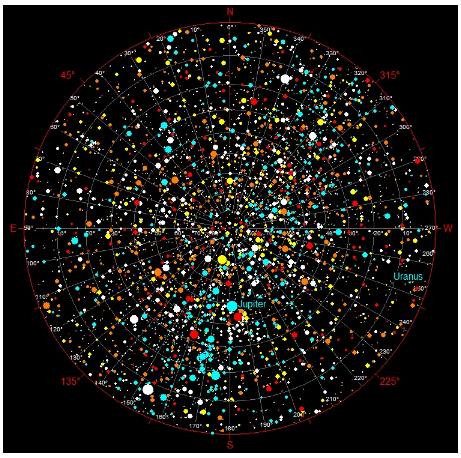


Figure 17: Star map at same time and location as in figure 10

A picture with a clear sky and distinct stars is selected. The exact time and location for this image is known. An interactive star map, StarCalc [9], is then used. Supplying the program with the same time and location it will provide a detailed view of how different celestial bodies should appear on the sky. As seen by comparing figure 17 and figure 18, finding the corresponding stars can be daunting task. However, the problem can be eased by knowing some extra parameters. The all-sky camera does not have axes fixed along the North-South-East-West directions. Rather the north in the image is rotated about 45 degrees clockwise compared to the "up" direction. The star map is rotated accordingly. The viewing angle in the ground based image is 90 degrees, which is less than the 180 degrees in figure 17. This allows us to zoom in on the desired area in the star map. Not all stars are bright enough to be visible in the ground based image, so the star map is set to display only stars above a certain threshold. Also, the OH-filter on the camera influences how bright stars are displayed according to the wavelength of the light they emit. In this instance this means that a very large, white star in the star map can be outshined by a much smaller red star next to it.

With all these factors accounted for it is now possible to compare the calibrating image in figure 18 with the adjusted star map in figure 19 . The indices of the stars and their corresponding azimuth and elevation angle for two different locations is presented in table 4.1 and table 4.2.

Figure 18: Calibration image, contrast adjusted to enhance stars

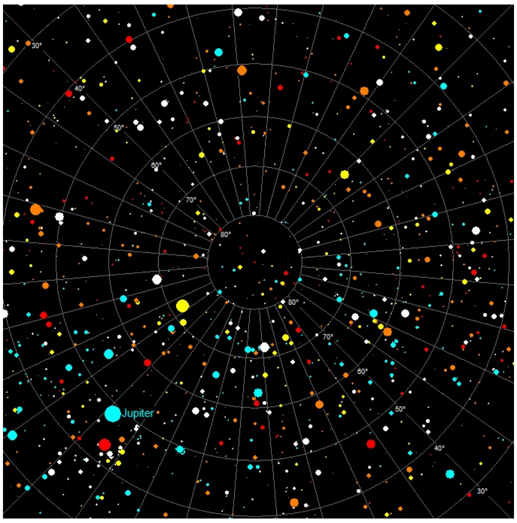


Figure 19: Adjusted viewing angle and star size from figure 17

Table 1: Corresponding coordinates for celestial bodies above Dragvoll at 18.11.12 22:13:13 UT. Note that the star names are not complete

Celestial body	Indices	Azimuth	Elevation
(-)	(i, j)	(degree)	(degree)
Polaris	488,1153	359.19	63.99
Camelopardalis	723,918	294.63	82.89
Capella	737,593	166.33	72.29
Jupiter	974,350	182.07	48.17
Shedir	957,1149	287.15	59.09
Camelopardis	613,805	90.29	85.77

Table 2: Corresponding coordinates for celestial bodies above Halley at 06.06.00 22:59:24 UT. Note that the star names are not complete

Celestial body	Indices	Azimuth	Elevation
(-)	(i, j)	(degree)	(degree)
Trianguli Tustrini	267,233	79.29	77.08
Gacrux	207,288	317.22	69.10
Rigel Kentaurus	226,246	10.66	75.10
Avior	974,350	248.2	56.90
Antares	190,146	38.39	38.17
Canopus	302,385	228.25	44.36

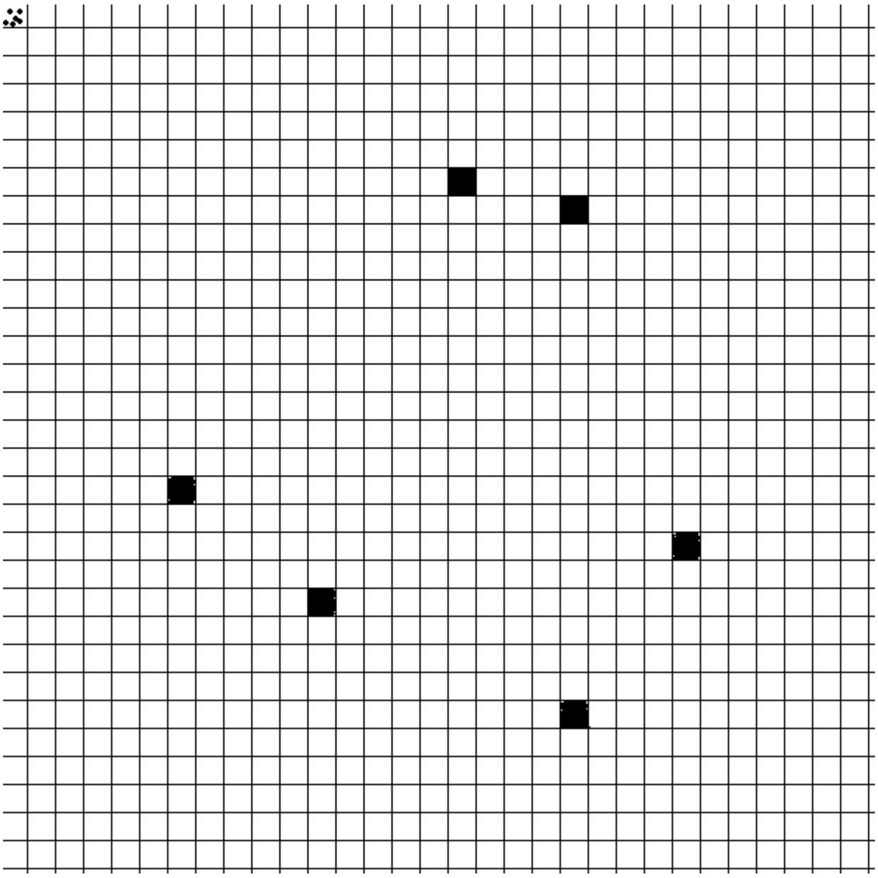


Figure 20: Illustration of difference in coordinates. Grid represents pixels in fish-eye image.

In the coordinates from the star map, azimuth and elevation, zenith is given by default. In order to find the location of zenith in terms of index coordinates in the ground based image, the conversion between the different coordinates needs to be known. This can be done automatically by a built in function in MATLAB, procrustes() [10]. The function takes on two sets of points with different X,Y-coordinates and then find the rotation, translation and scaling between the two sets. The indices of the stars in the ground based images are already in the desired format. The azimuth and elevation data from the star map is converted to points on a circle of unity radius, before their radius is scaled with the ratio of $\frac{elevation}{90^\circ}$. Then the two sets of data can be placed in the same coordinate system as illustrated in figure 20. Filled pixels illustrate pixel location of stars in the ground based image. The small dots in the upper left corner illustrate the converted coordinates from the star map. With these data sets, procrustes() can find the rotation, translation and scaling between them. The value of the translation will correspond to the indices of the zenith pixel, as the default value of zenith is zero in the converted coordinates from the star map.

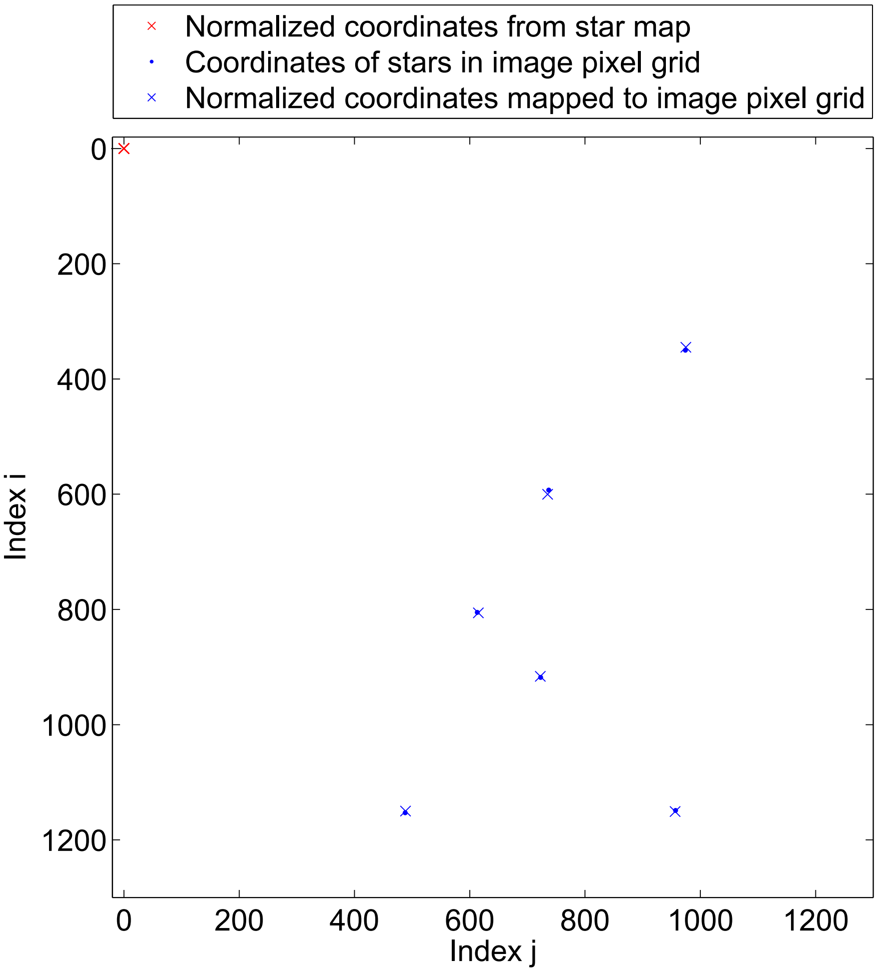


Figure 21: Search for zenith in Dragvoll images

Figure 21 shows how this will look for the Dragvoll case. The the normalized coordinates from the star map is placed in the upper left. On this scale they will appear as a single coordinate, while there actually are several in close proximity to each other. The other points show how the normalized coordinates can be transformed to fit with the pixel locations. The transform will give us the pixel value as described earlier.

5.3 Bit Binning

The resultant images after mapping to geographical coordinates will be made to have a square resolution, with side lengths equal to the smallest side length in the original image. These square images will be of a much larger resolution, 1253x1253, than the smaller oblong resolution the satellite images will have, 256x320. These images will contain information equal to a flat square plane located in the same height as the airglow layer.

In order to obtain useful test images for the compression algorithm, the geographically mapped images needs to be converted to images of resolution 256x320 as they would be seen from the satellite . In order to decide what parts of the geographically mapped images are covered by the satellites field of view, some extra calculations are needed. In our case $S_1 >> S_2$ and $f = F = S_2$.

$$FOV_i = 2tan^{-1}\left(\frac{CCD_i \cdot pp}{2 \cdot f}\right) \tag{7}$$

$$FOV_j = 2tan^{-1}\left(\frac{CCD_j \cdot pp}{2 \cdot f}\right) \tag{8}$$

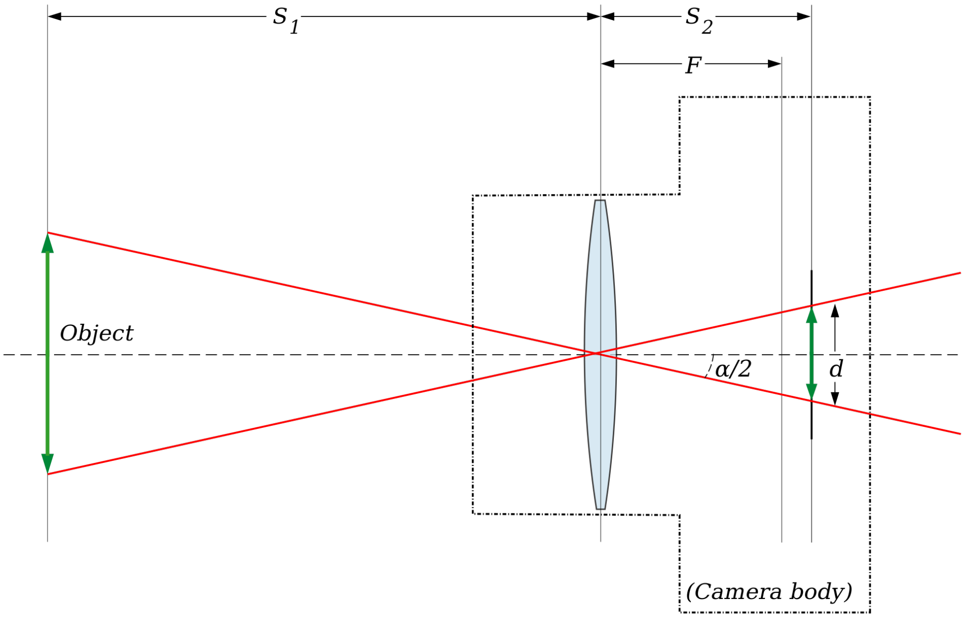


Figure 22: "Object" represents the geographically mapped image, while "d" represents the CCD-chip. Figure adapted from [8]

These equations give the respective angular field of view, based on the lengths of the two sides of the CCD-chip and the focal length l. The lengths are the product of the pixel count in each direction, CCD_i and CCD_j, and the size of each pixel, the pixel pitch (pp). The area that the satellite will cover is then given by the two lenghts the two FOV will see in the plane of the geographically mapped images. These lengths are given by the following relations between the FOV and the height from the layer to the satellite. $(H_{sat} - H_{layer})$.

$$D_i = 2tan(FOV_i)(H_{sat} - H_{layer}) \tag{9}$$

$$D_j = 2tan(FOV_j)(H_{sat} - H_{layer}) \tag{10}$$

The length covered and the pixel count in each direction is known in the geographically mapped image. From this one can find the length per pixel, lpp_{mapped}. The equivalent size from the image that the satellite will take can be found as $lpp_{sat} = D_i/CDD_i$. The ratio lpp_{mapped}/lpp_{sat} gives the necessary scale needed to find out how many pixels in the geographically mapped image to bin together in order to give it a *lpp* equal to the calculated *lpp* from the satellite images. Starting in the middle of a grid of size CDD_ixCDD_j one can fill the pixels with values from pixels in the geographically mapped image outwards according to the ratio lpp_{mapped}/lpp_{sat}. As this is not necessarily an integer a interpolation scheme is needed to find the correct values to fill the pixels.

After mapping to geographical coordinates and binning the bits of ground based images of the atmosphere, we have images which should be a better approximation of how the satellite images will appear. This gives more realistic images to test the compression algorithm, which in turn gives a better opportunity to find suiting levels of quantization in the compression algorithm.

6 Effects of Compression

6.1 Compressing Images

Two images, one from Halley and one from Dragvoll, was chosen to test the compression algorithm. The compression algorithm made by [3] was made to operate on square images. In order to test the simulated satellite images the compression algorithm was adapted to accept rectangle resolutions. The images were shifted in a direction orthogonal to the wavefronts in order to make a series of 7 images that could exploit the three dimensionality of the 3DPCM. Several different values for chosen for the number of levels and size of the dead-zone in the quantizer. Following is some examples of some of them.

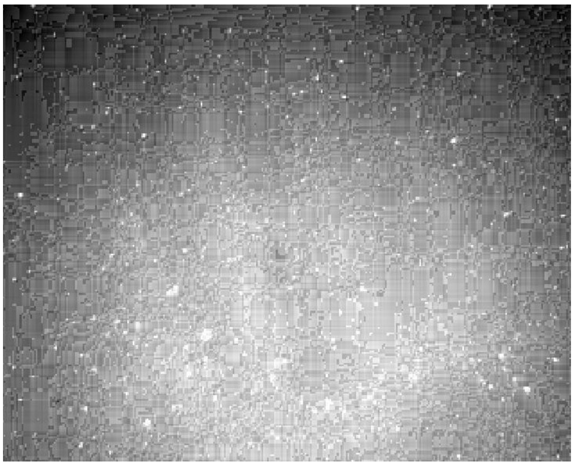


Figure 23: Compressed image from Dragvoll. 5 levels and dz equal 1.5

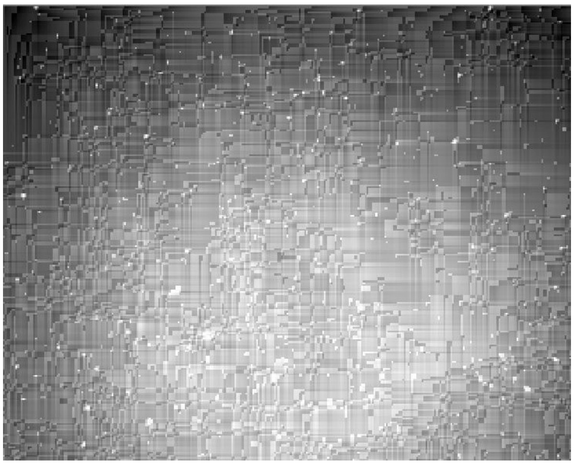


Figure 24: Compressed image from Dragvoll. 5 levels and dz equal 2.0

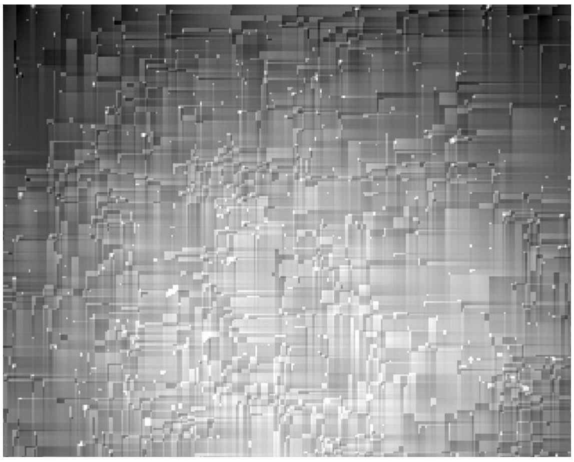


Figure 25: Compressed image from Dragvoll. 5 levels and dz equal 2.5

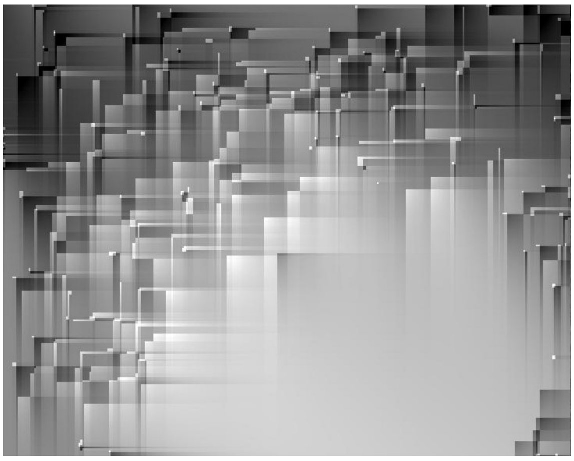


Figure 26: Compressed image from Dragvoll. 5 levels and dz equal 4.5

6.2 Difference Imaging

The wave structures in the images may not appear as distinct wavefronts. To mitigate this challenge a new image is made from the compressed image and the same image shifted a half wavelength orthogonal to the wavefront.. By subtracting these two images, only the difference between the images will be present. This will eliminate much of the general intensity variation and leave an image where the variations in intensity are due to the wave structure. This means that the resulting difference image will have a larger amplitude in intensity variance. This effect is illustrated if the two waves in 27 are subtracted from each other. How the real waves become more distinct is apparent if one compares figure 28 - 31

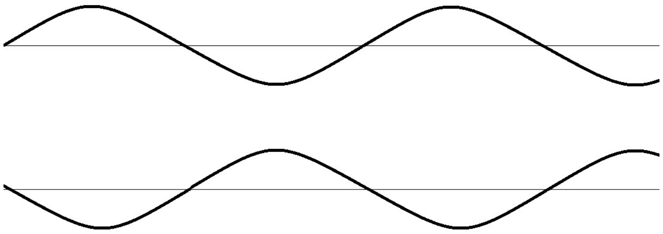


Figure 27: Phase-shifted sine waves

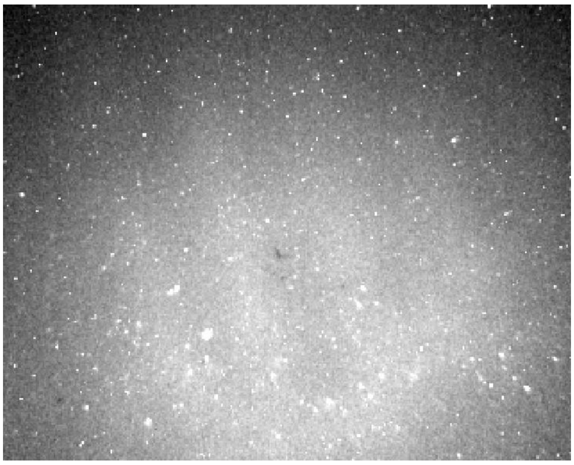


Figure 28: Uncompressed geographically mapped image from Dragvoll

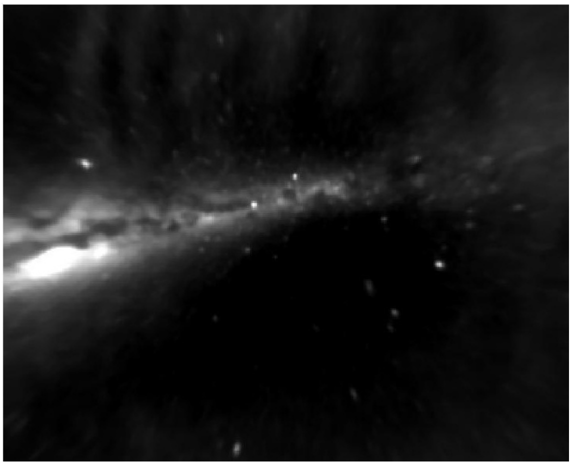


Figure 29: Uncompressed geographically mapped image from Halley

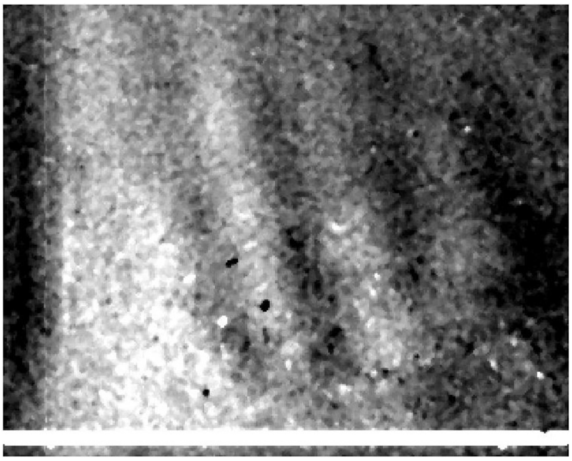


Figure 30: Figure 28 shifted and subtracted from itself

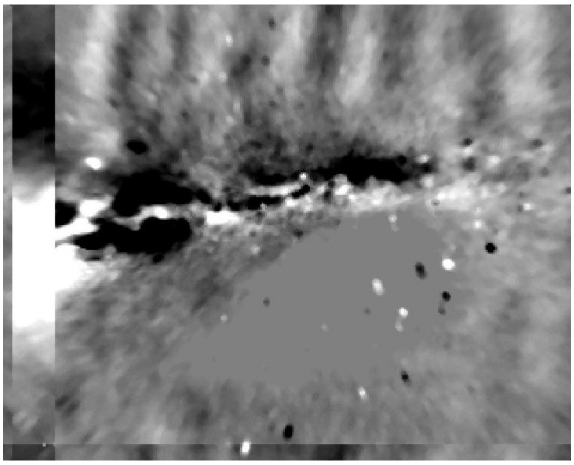


Figure 31: Figure 29 shifted and subtracted from itself

6.3 Fitting to Find Wavelength

Once the images has been run through the compression algorithm and differenced, it will be necessary to extract the wave parameters available within them. This has to be done in order to compare how the wave parameters change with varying degrees of compression. This done by extracting strips of pixels orthogonal to the wavefronts. By plotting the intensities in the respective pixels and fitting a sine waveform to them it should be possible to find amplitude, wavelength, phase shift and a constant DC-level. By comparing phase shift from image to image it will be possible to find how far the waves have moved between them, and if the time between subsequent images is known, the speed is readily available.

Once the difference images of the compressed images are ready, a strip of pixels is chosen along the path of wave propagation. This single strip can however contain some random noise that disturbs the waveform. In order to eliminate some of this noise, several strips defined by the original strip and parallel shifted orthogonally are also made. These values are then averaged, and should contain a better signal to noise ratio. The pixel values are then fitted to a sine using a built-in function from MATLAB, nlinfit. The process is then repeated for a recovered compressed image. The sampling points are chosen to be in the same indices as in the original image to provide a basis for comparison. Comparing the wavelengths from the fitted sines will then give a good representation of how much the chosen compression level distorts the data.

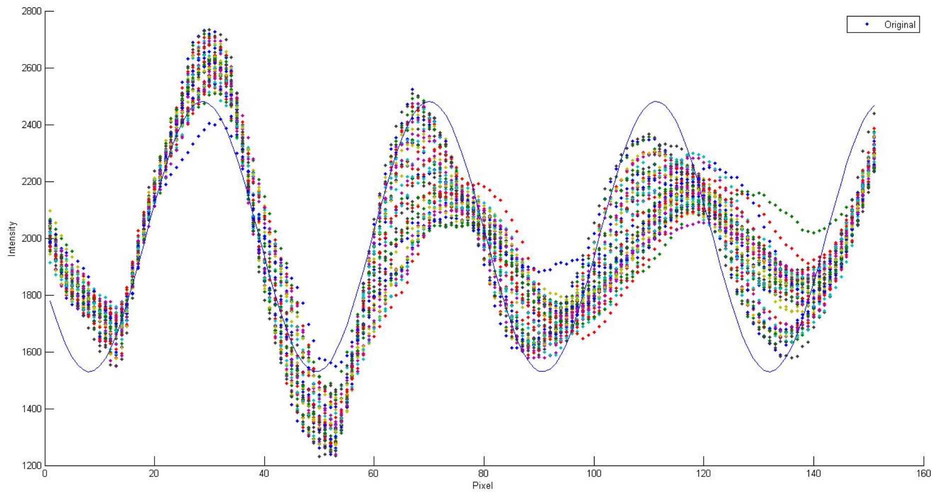


Figure 32: Samples from Halley images with different compression levels. Line represents nonlinear fit in uncompressed image

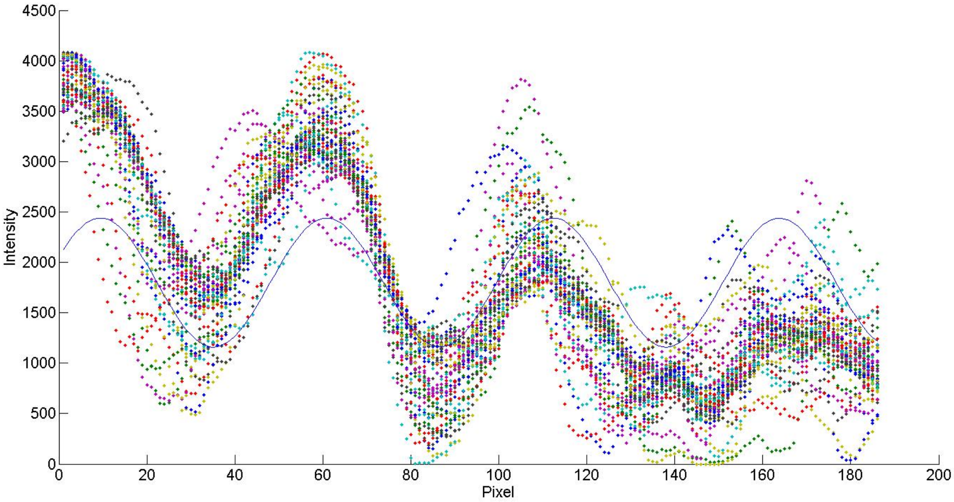


Figure 33: Samples from Dragvoll images with different compression levels. Line represents nonlinear fit in uncompressed image

Figure 32 and figure 33 show samples from different compression levels across the wavefronts in images from Halley and Dragvoll respectively. Already in this stage one can see trends that the images from Halley receive a systematic shift in the fitted wave while the shift in images from Dragvoll seem to of a more random nature.

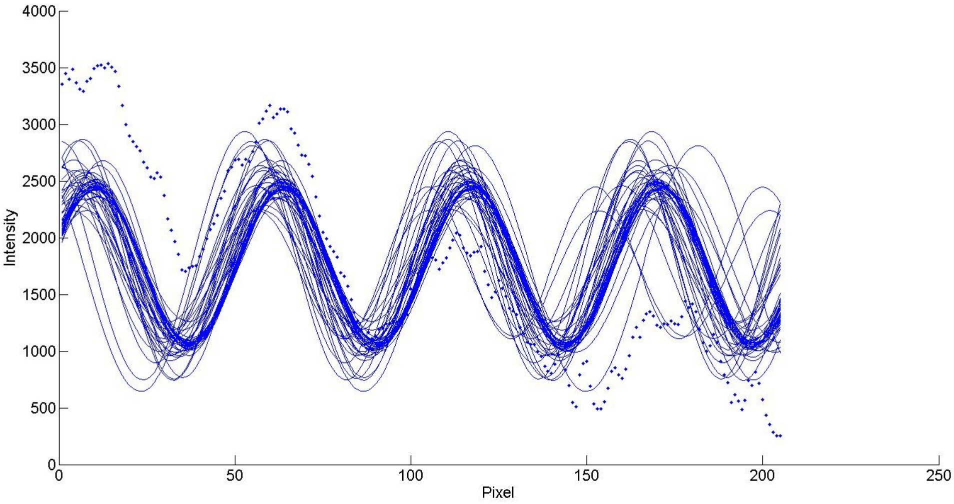


Figure 34: Fitted sine waves from Dragvoll images with different compression levels. Dots represents samples in uncompressed image

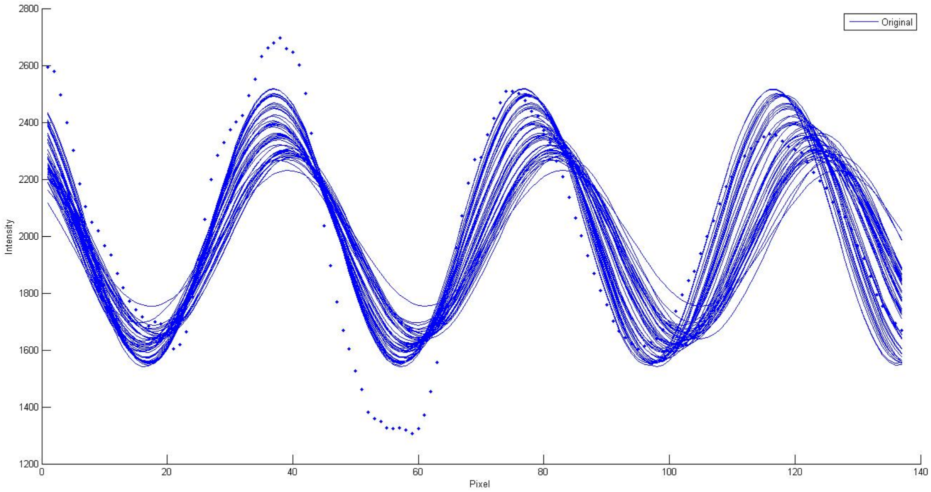


Figure 35: Fitted sine waves from Halley images with different compression levels. Dots represents samples in uncompressed image

Figure 34 and figure 35 show the fitted sine waves to the data from figure 32 and figure 33 Also here one see trends that the images from Halley receive a systematic shift in the fitted wave while the shift in images from Dragvoll seem to of a more random nature.

6.4 Results

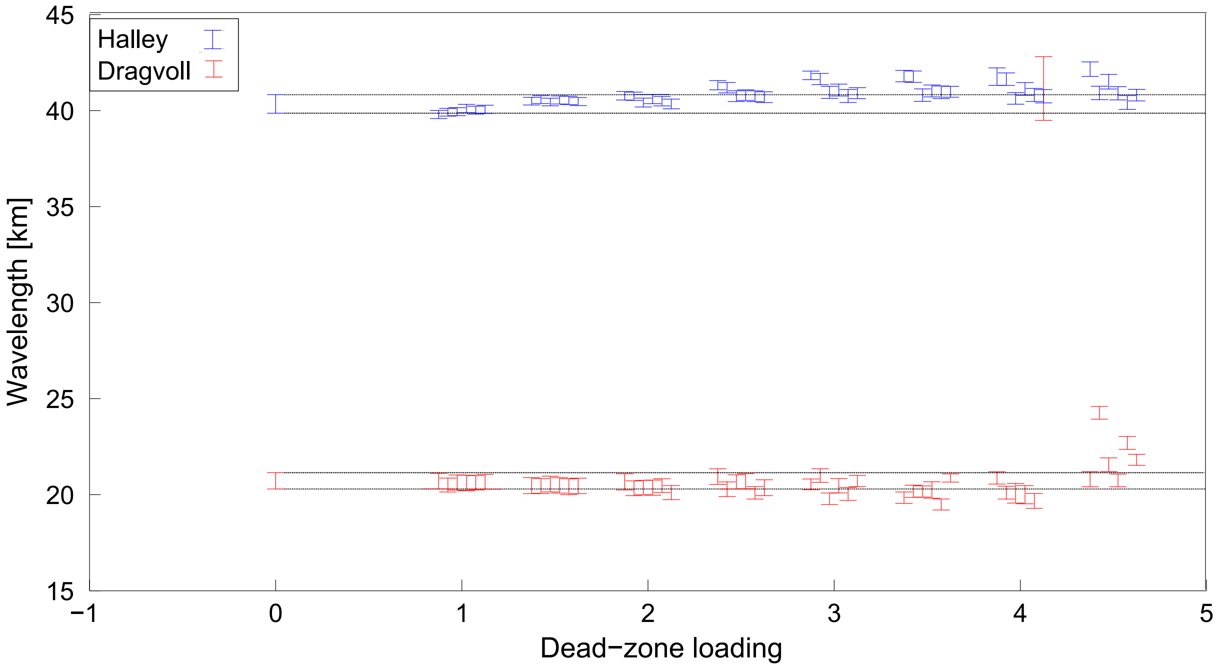


Figure 36: Each cluster of wavelengths represents the value of dead-zone loading that they approximately hover over. The actual values are in increments of 0.5 Within each cluster the variation is the number of levels in the compression algorithm. From left to right they are, within each cluster, 3, 5, 7, 9, 11 and 13. Larger dead-zone loading signals more compression, while larger values of levels signals less. Single value to the far left represents zero compression.

In this plot the far left wavelength stem from sampling several strips manually in the uncompressed image to find the wavelengths and the associated average and standard deviation. The other wavelengths stem from samples from one single selection of strip, on the exact same location, on all the different compression levels. The standard deviation plotted as error bars on these wavelengths comes from the fitting routine in MATLAB. The sampling to find the wave in the uncompressed image is based on manual input, which is naturally prone to some error. The fitting method will also have some uncertainty in its estimates. Finding the average value and standard deviation of several manually sampled wavelengths determines the accuracy. The uncertainty in fitting estimates of several compression levels in one sample determines the precision. The point where the precision becomes worse than the accuracy can be found. That point will in essence be the point where data is lost, and no more compression should be undertaken.The goal is to determine when the wavelengths from the compressed images no longer will be within the expected errors of our measurement of the uncompressed waves.

This point can be identified as the point where the wavelengths from the recovered images breach the dotted lines given by the estimate of the uncompressed wavelengths.

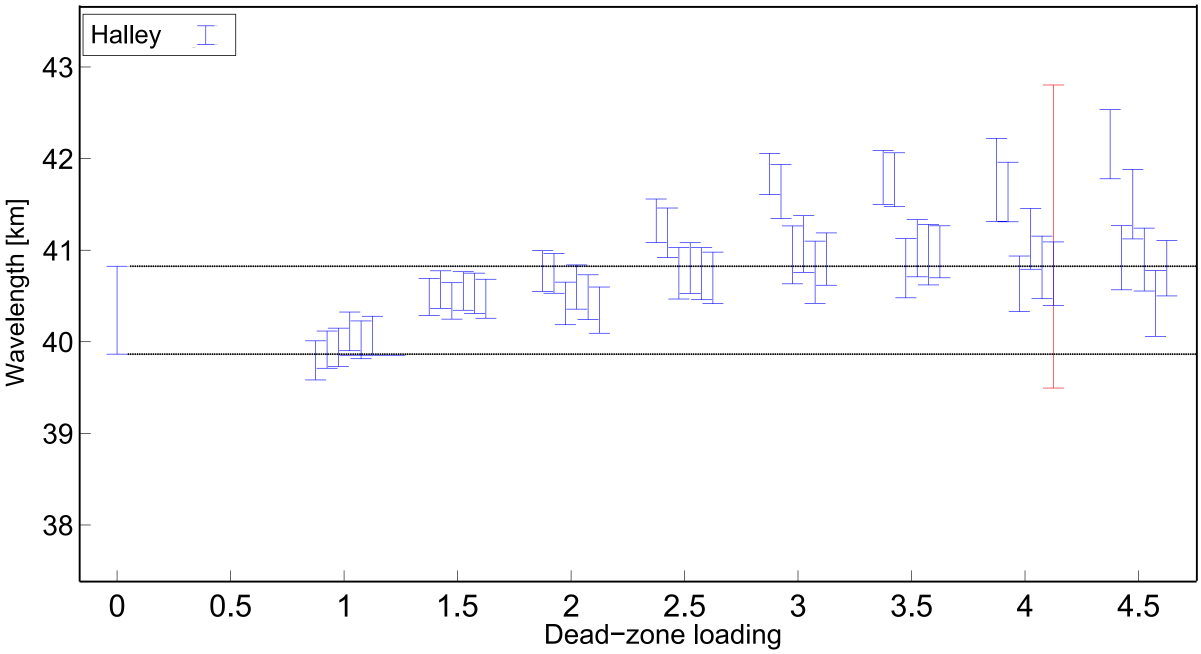


Figure 37: Wavelengths for different compression levels from Halley

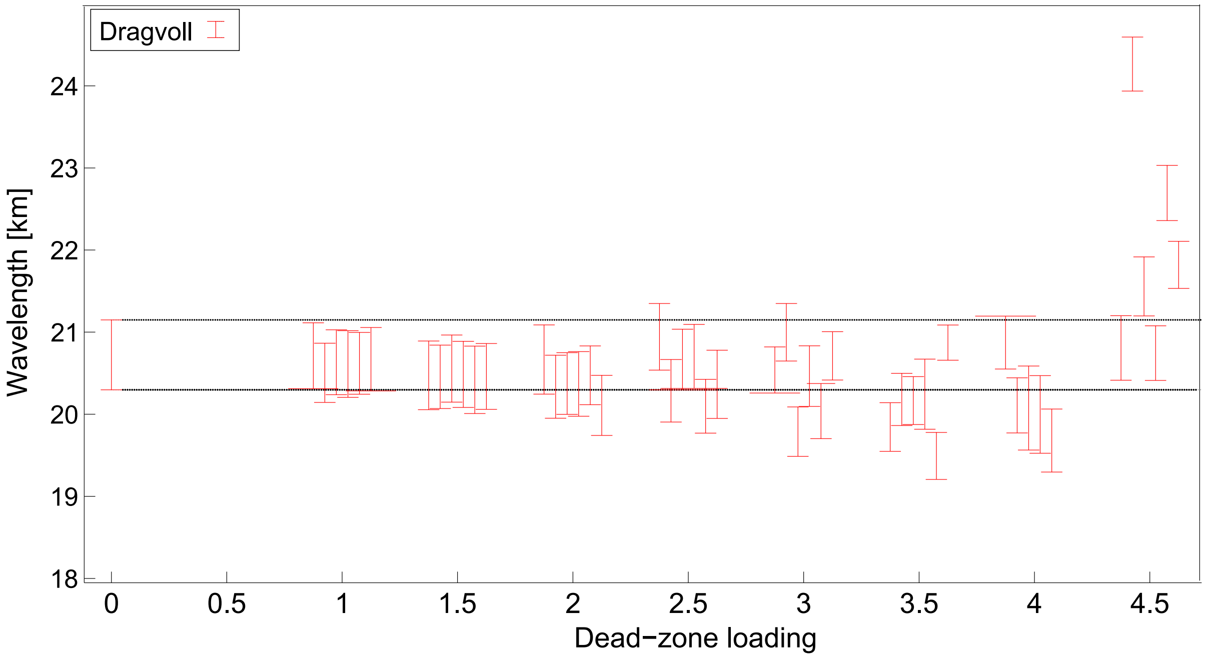


Figure 38: Wavelengths for different compression levels from Dragvoll

From this small selection of compression levels and two tested images it is possible to distinguish some trends. The Halley images seem to get a systematic shift towards higher wavelengths with increased dead-zone loading, but that this to some extent can be remedied by increasing number of levels. The Dragvoll images on the other hand seem to have a trend where number of quantization levels seems to have no effect in stabilizing the wavelengths and they assume a seemingly random distribution with varying degrees of compression. This can probably be attributed to the fact the image from Dragvoll has a lower SNR that the Halley images, meaning that different choices of quantization has a greater influence on the output of the compression algorithm. This will in essence mean that the error introduced by the user will be more dominant in the Dragvoll images with a lower SNR. With a larger selection of parameters these trends could be investigated further.

These two cases can be considered as best-case and worst-case scenarios. If the images resemble Halley images a choice of dead-zone loading equal 2 with number of levels larger than 11 could be seen as a safe choice. But dead-zone loading equal 1.5 would be safer with any number of levels. If the images resemble the images from Dragvoll more caution should be exercised. Here, a dead-zone loading equal 1.5 would be edgy, while dead-zone loading equal 1.0 could be seen as safer.

In order to determine in these trends actually are trends and not random results, more testing should be done. Currently, only two images has been tested, and the number of levels has been limited to 13. The dead-zone loading is limited in within the compression algorithm to be less than 5, this can be modified. If more images are compressed with more quantization levels and larger dead-zone loading values, the trends discovered here can be investigated further. That could provide more rigorous estimates on a suitable compression level.

7 Identifying Waves in Reconstructed Images

Once a suitable level of compression has been found it is of interest for further research to identify the waves properly. In the previous section, the wavelengths were found by a user visually determining a line to sample orthogonally to the wavefronts. In addition, samples where the fitting was successful for several compression levels was only found after several trials and errors. This results in inaccuracies in addition to the uncertainties in the fitting routine itself. In an attempt to eliminate some of the error introduced by the user an improved method was implemented. This was based on the user still determining a line orthogonally to the wavefronts, but now, after this was done the routine would take over. Based on that original sample it would make several new samples in a fan shape from one of the endpoints of the line. In theory the fitted wavelengths would then vary as a function of the angle between the sample and the actual wavelength. The shortest wavelength should correspond to the sample that is most orthogonal to the waves. This can be seen in an actual sample in figure 39

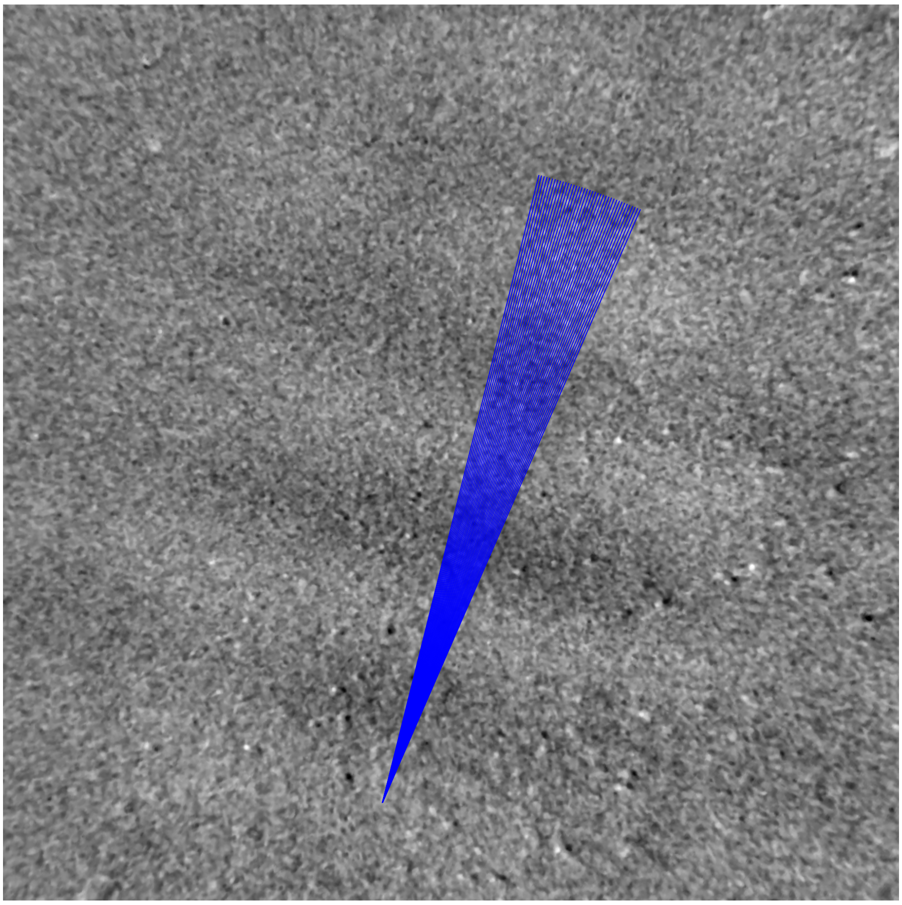


Figure 39: 40 samples distributed over different angles from user input sample

The idea is then to choose the shortest wavelength, and then also extract the amplitude and orientation from the fit for further processing. The method is however far from perfect since it does not produce easily reproducible results. Filtering would also be required for a better result. In figure 40 there is a plot of the associated wave numbers from the previous figure. As $\lambda = 2\pi/k$, the highest values in this plot corresponds to the shortest wavelengths. The trend in the picture is not typical as it varies for each separate try. Also evident is the spread between neighbouring values, which ideally should be smoother.

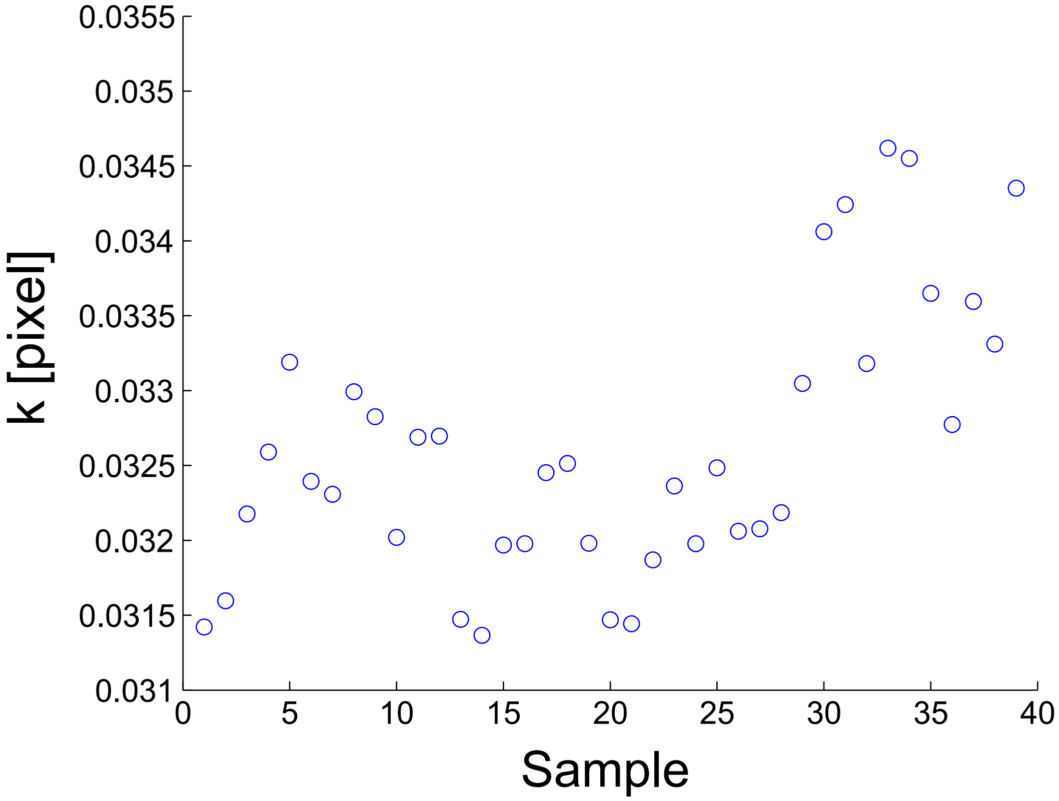


Figure 40: Identified wave number from the 40 samples in the previous figure

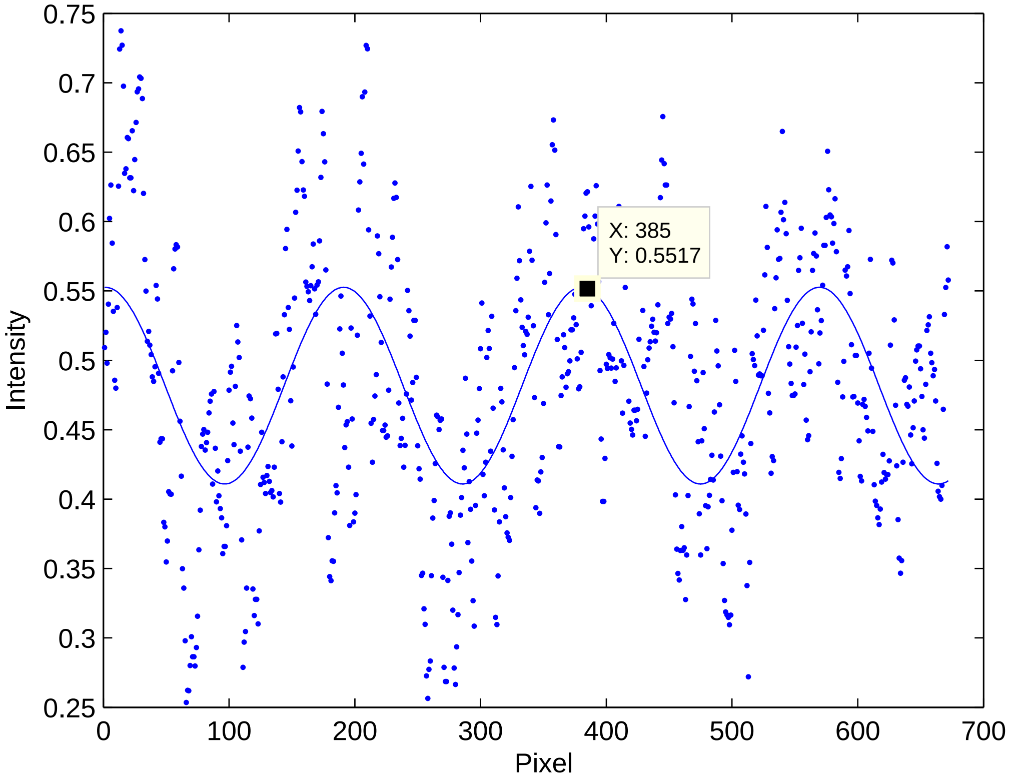


Figure 41: Arbitrary fit from the fan sampling

8 Preperation of Data for 3D-FFT

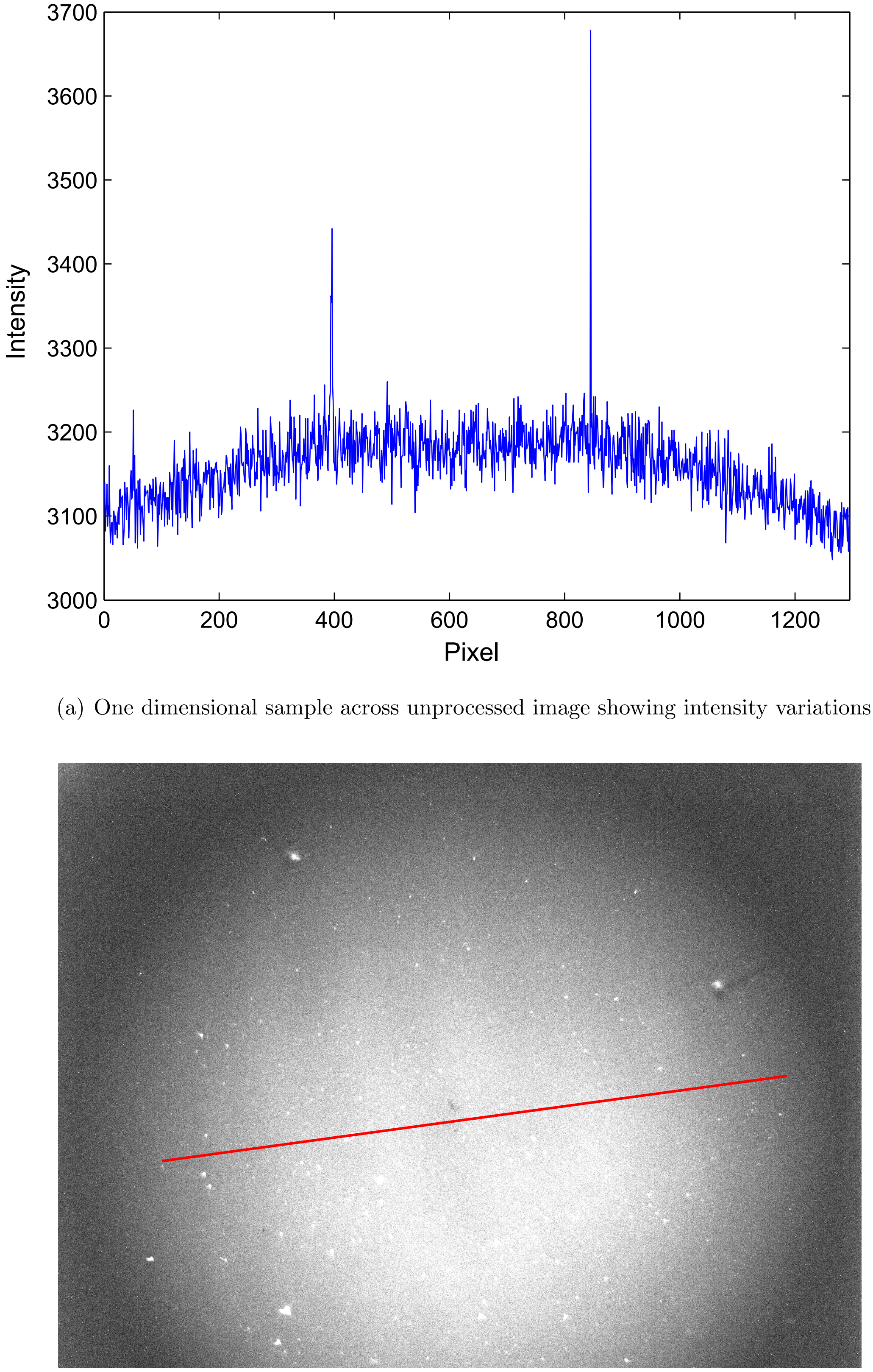
The approach of compressing satellite images and then process them is a proven one. A tool for testing how much compression can be tolerated has been made, as well as a way of identifying the relevant waveforms. The compression is computationally heavy, the fitting of wavefronts is tedious and has yet to provide consistent results. A different approach based on a 3D-Fast Fourier Transform (FFT), designed by Gardner *et al.* [11], has been implemented as an alternative to compression, transmitting, reconstruction and fitting. The 3D-FFT should be able to identify the relevant data directly, on-board the satellite. These data are the wavelengths, the orientation and amplitude. This is a method which will produce more consistent results, and will only require the data about the waves to be sent, as opposed to entire images.

As before, ground based images will be used to emulate satellite images for further testing. The ground based images are taken from ground level in Trondheim, looking at the sky with a wide angle lens.

Figure 42(a) shows a vector sample across a typical ground based image, where no processing has yet taken place. In figure 42(b) one can see where this data was sampled in the image. The only processing that has taken place in figure 42(b) is scaling of the pixel intensities to allow the image to be displayed in grayscales visible for the human eye. In figure 42(a) it is possible to see some of the typical traits of the ground based images:

- There is a general curve in the intensity. The intensity is weaker towards the edges, and stronger towards the middle. This is due to vignetting and atmospheric attenuation.
- At approximately sampled pixel number 400 and the immediate surrounding pixels there is a peak present which represent the added intensity from a star.
- At approximately sampled pixel number 50 and 850 respectively, there is one single pixel with as a spike in the intensity. This represents a hot pixel which stems from a malfunction in the CCD-chip used.
- Embedded in the general intensity variation it is possible to make out a weak wave structure across the sampled data.

The waveform is the relevant data for this text. The following sections are aimed at removing other traits from the images and enhancing the waves in order to investigate them further.



(a) One dimensional sample across unprocessed image showing intensity variations

(b) Red line represents sample across the image, shown in the previous figure

Figure 42: Intensity vector sample and source of sample

8.1 Removal of Stars and Hot Pixels

The ground based images will inherently contain stars as they are captured looking towards zenith. The stars will act as a disturbance of the wave signal and their presence could in some instances make sampling of the correct waveform less accurate. Removing the stars from the images and replacing their pixels with interpolated values from the surrounding area was therefore set as an objective. A robust method was developed, using many of the tools included in MATLAB's "Image Processing Toolbox". The inner workings of these tools will not be discussed further, as they have only been used as "black boxes". The following is rather an explanation of how they have been combined to achieve a satisfactory result. Consult MATLAB's own documentation for more reference on the tools used.

8.1.1 Hot pixel Removal

Given the easily recognizable nature of the hot pixels a simple yet effective method was applied to remove them. Iterating trough the pixels of the image, the average of the surrounding pixels was computed and compared to the pixel in question. If the value of the given pixel was larger than a certain percentage of the average value of the surrounding pixels, this pixel was deemed to contain a hot pixel with the associated spike in intensity. This intensity was then replaced with the average value. The percentage difference needed to distinguish a hot pixel from its neighbors was chosen as to give a safe margin from the variations naturally present by noise in the signal. The solitary spike in intensity is only found in hot pixels, as the stars stretch over several pixels with a gradient to the intensity variation. The stars are therefore untouched by this method and require a different approach to be removed.

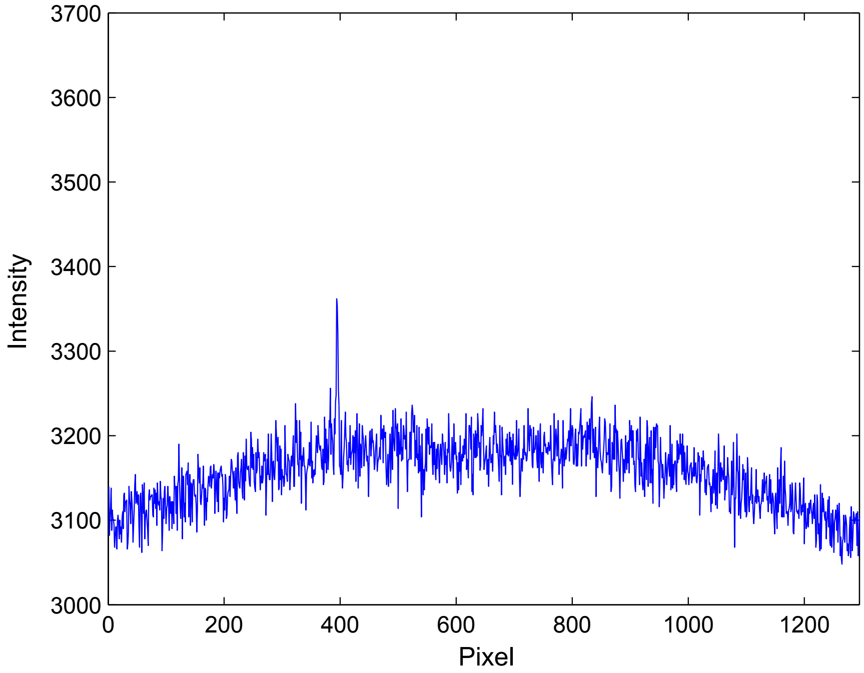
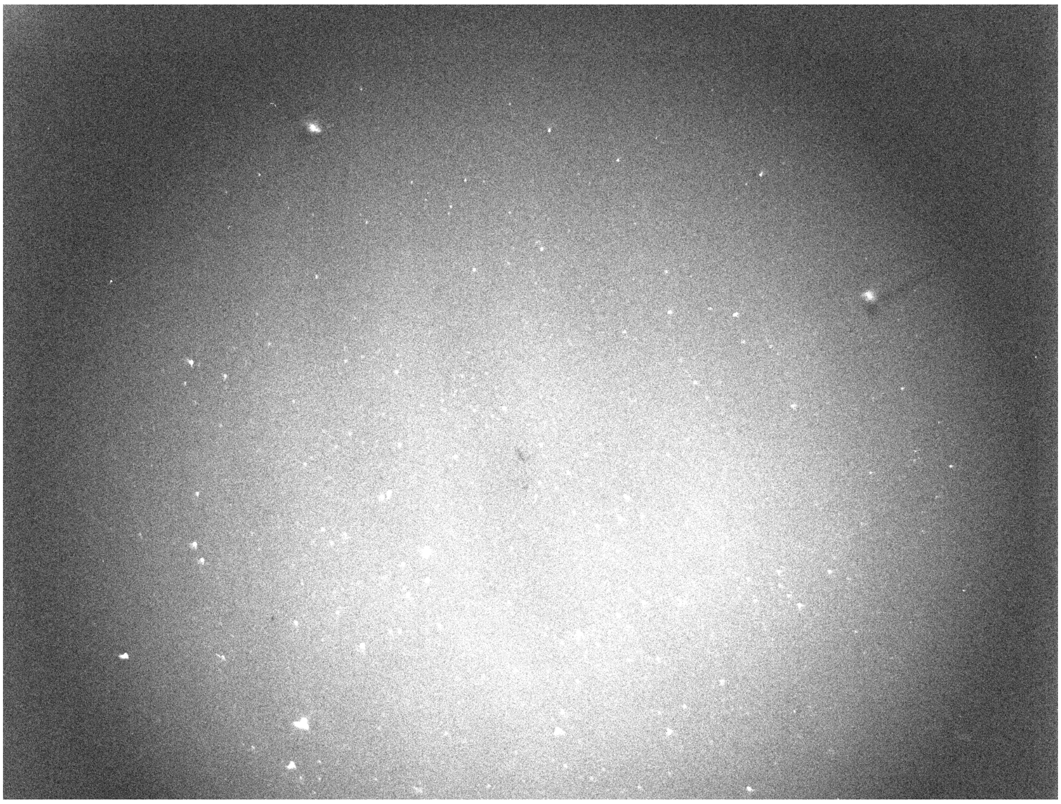


Figure 43: The same cross-section as figure 42, but now with the hot pixels removed. Close to sampled pixel number 400 a star remains.



(a) Hot pixels removed from image

(b) White pixels represents change from before hot pixels were removed, and after (Might only be visible on a computer screen, at a zoomed level)

Figure 44: Removal of hot pixels

8.1.2 Star Removal

In order to remove the stars, it is important to appreciate the difference between them and hot pixels. As can be seen in figure 45 the stars do represent a spike in the intensity, but not in the discrete way of the hot pixels. With stars, the intensity increases rapidly over a few pixels. Finding the correct border between star and background is now a more complicated task.

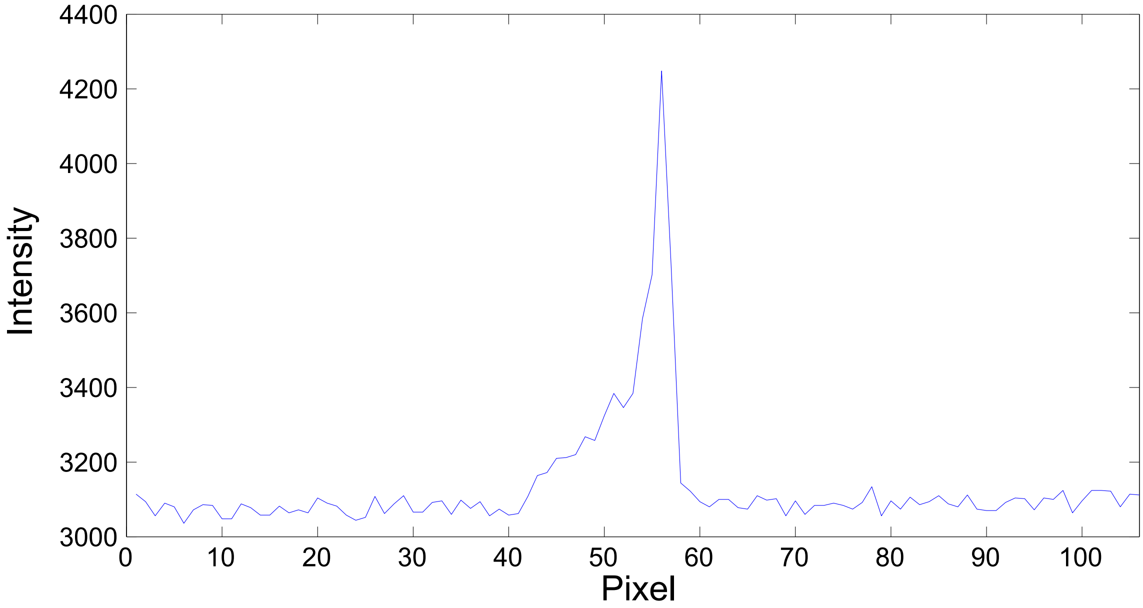


Figure 45: Close-up cross-section of intensity of a random star

To correctly identify the stars present in the signal, a built in MATLAB-function was used. BW = edge(p,'canny',thresh) returns a logical black and white image which identifies edges present in the image p, using the Canny edge detector developed by John F. Canny in 1986. 'thresh' defines the gradient needed to identify an edge.

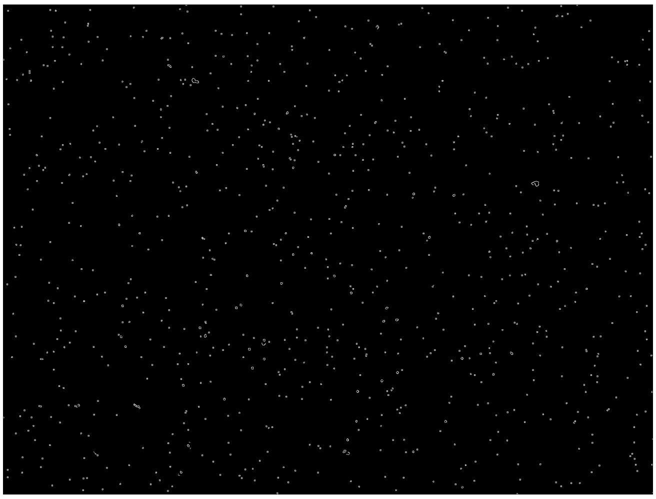


Figure 46: BW-image. White pixels represents identified edges from steep gradients in the pixel intensity still present after removal of hot pixels. 1252x1663 pixels

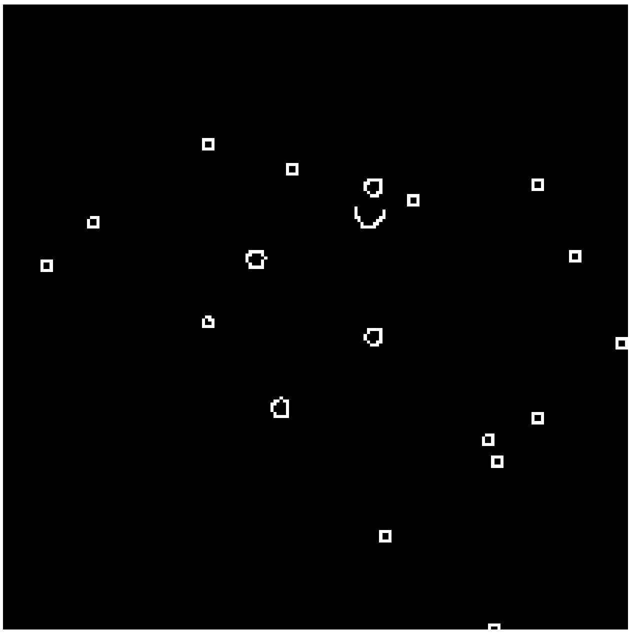


Figure 47: Zoomed view of previous figure. Showing pixels 800:1000x550:750

The presence of stars is marked by small pixelated rings, based on the steep intensity gradient associated with the relatively round stars. More processing is however needed to make a positive identification of actual stars, and how large area surrounding them that should be replaced by interpolated values.

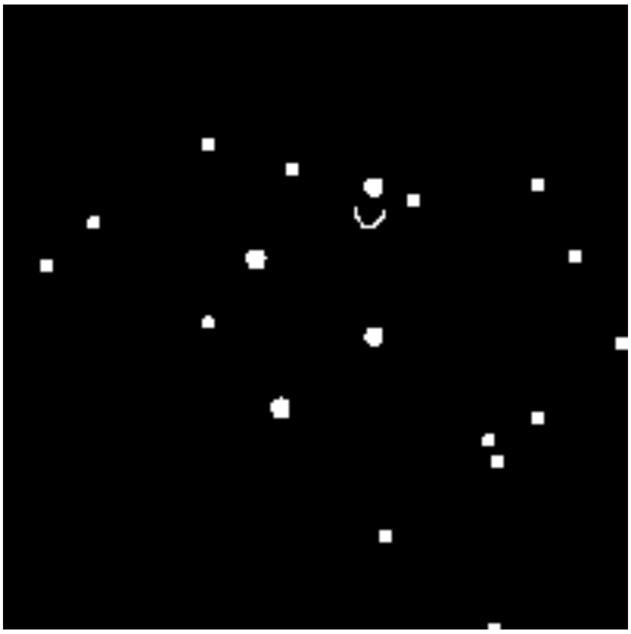


Figure 48: Circles in previous figure filled

By using the MATLAB-function BW = imfill(BW,'holes') on the BW image in figure 46 (and with that also figure 47) one can fill the logical black and white circles produced by BW = edge(I,'canny',thresh). The result can be seen in figure 48.

This image, BW, of logical ones and zeros is then used as a map for p = roifill(p, BW) which removes the pixels in the original image, p, with the same indices as the white parts of BW. BW is now the result of the edge detector and filling of the resulting circles. After p = roifill(p, BW) has removed the star pixels from the image, it will replace them with interpolated values from the immediately surrounding pixels. These interpolated values will present themselves as a smooth transition from edge to edge. The surrounding pixels however, do not represent a smooth signal. To emulate the noise that should be present in the signal, a small square of the top left part of original image, with the hot pixels removed, is sampled. In this area there is no signal present, as the circular filter used in the camera prevents light from hitting all parts of the CCD chip used. The standard deviation of the noise introduced by the camera itself is calculated from this square and added to the interpolated values previously mentioned. This will remove some of the stars.

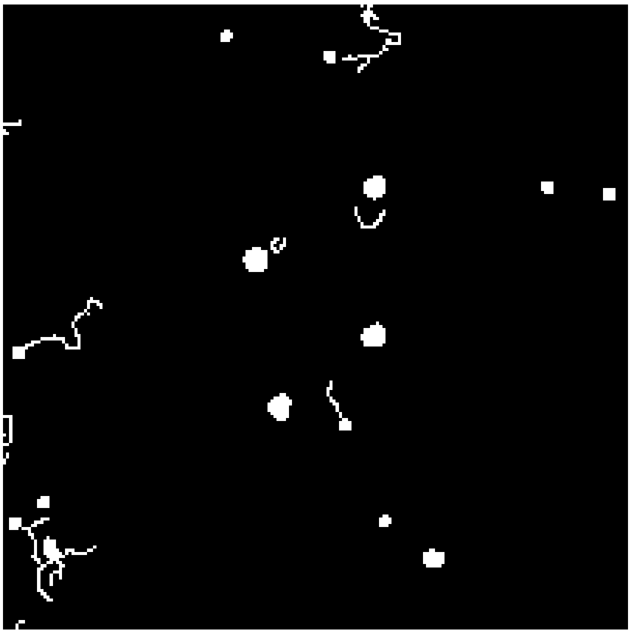


Figure 49: Areas identified as stars after previous areas identified as stars have been partly replaced by interpolated values of surrounding pixels

So far, so good. The approach for finding the edges is however not flawless. It requires a threshold, which in some parts of the image may make the detection too sensitive, while too unsensitive in some other areas. In order to mitigate this problem, the process described so far is repeated on the image that just had the areas detected as stars replaced. The result of this can be seen in figure 49. It is possible to see that some of the stars has survived the process, while some other features of the images has been detected as stars, eventhough they clearly do not have the shape of a star.

The other features of the images that have been detected as stars, but are not, are there because of the workings of BW = edge(I,'canny',thresh) which adapts to the image it is fed. If the gradient in the image is smaller, it becomes more sensitive and reacts to smaller gradients. This is evident as the "snakes" in figure 49.

The surviving stars are there because of the way the limits for the stars are set by the Canny edge finder. When encountering stars such as the one seen in figure 45, the edge may not be set exactly where the signal starts to deviate from its locally, relatively flat trend. Since the edge detector is expecting a certain gradient it may not detect intensity that has "bled" into other pixels. This is a trade-off in the chosen threshold for the gradient. To small and it will not detect enough stars and too big it will react to natural variations in the signal as seen in figure 49. For instance, if the edge detector detects the gradients around pixel 53 and pixel 56 in figure 45 as an edge, the interpolation from pixel 52 will still leave an intensity peak. Keep in mind that this is an illustrating 1D visualization of the actual 2D approach.

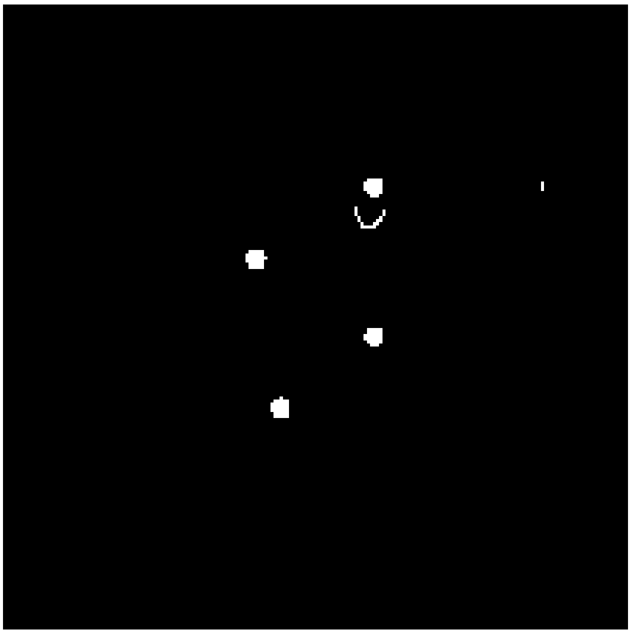


Figure 50: White regions show which pixels were white in both edge detections, marking which regions that needs a bigger mask of interpolating area surrounding the star

To distinguish the areas that still contain actual stars, figure 48 and figure 49 is compared. If a pixel is white in both figure 48 and figure 49 those pixels are set as white in a new logical image. This can be seen in figure 50. This figure shows only stars that have not been removed because they have "bled" into surrounding pixels. This "bleeding" has a lesser gradient than the one needed to remove "sharper" stars. Also, the features detected that are not stars are eliminated by comparing the two star identifications. The apparent "snake" still visible in figure 50 is not a misidentification, but rather two stars close to each other. This is also a feature that needs some attention.

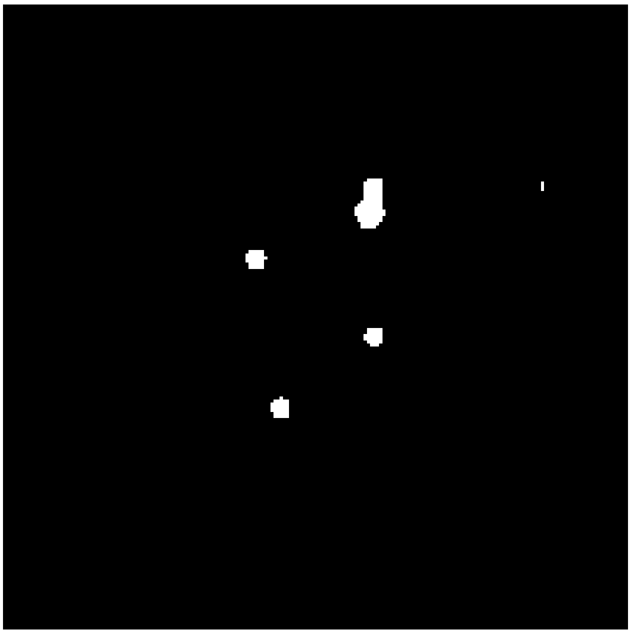


Figure 51: Morphing of regions in close proximity

In order to eliminate instances such as the "snake" in figure 50 a built in function from MATLAB called BW=imclose(BW,se); is utilized. Given a structuring element, in this case a disk with a radius of 8 pixels, the method will perform a morphological closing. This operation consists of a dilation followed by an erosion dictated by the structuring element. In essence this will connect identified star elements that are sufficiently close to each other. The effect of this can be seen in figure 51.

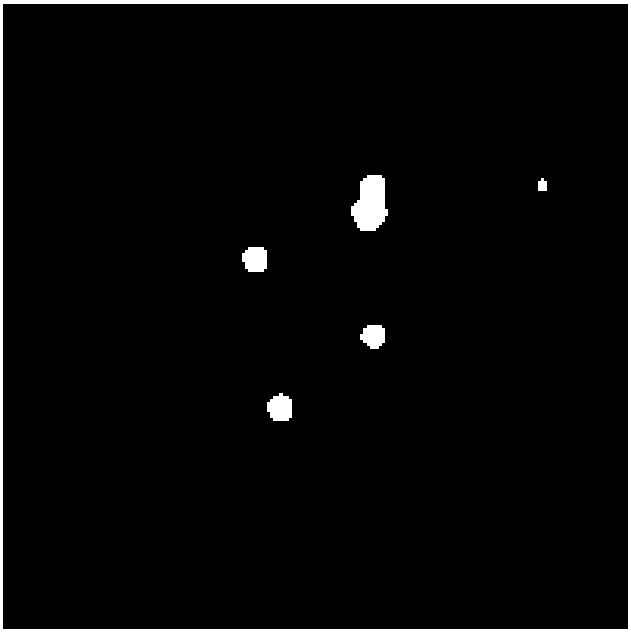


Figure 52: Increasing size of all white regions by adding pixels along the edges

BW = bwmorph(BW,'thicken',1); is then applied to make the structures now identified padded with some extra pixels. This will make it more probable that the area identified as stars cover the entire area influenced by the star, not only the area where the greatest gradients are present. This is the same effect as mentioned earlier. If the edge detector detects the gradients around pixel 53 and pixel 56 in figure 45 as an edge, the interpolation from pixel 52 will still leave an intensity peak. Still, this padding of the areas identified as stars is not enough to cover the entire area influenced by the stars. The result of the padding can be seen in figure 52.

In order to make sure that the area identified as stars is sufficiently large, the group of white pixels relating to each seperate star is given individual treatment. In order to give individual treatment, they need to be separated by size.

Choosing an upper limit for how many pixels there could possibly be within one single star, a 3D matrix is made. Two of the dimensions are of the same size as the original image, and the last from 1 to the largest possible star in pixel size. This is to sort the different identified stars by size. Iterating through starsize, 3DBW(:,:,starsize) = bwareaopen(BW, starsize); will remove any area in figure 52, here BW, that has a lesser connected pixel count than starsize. This is illustrated on the left side of figure 53.

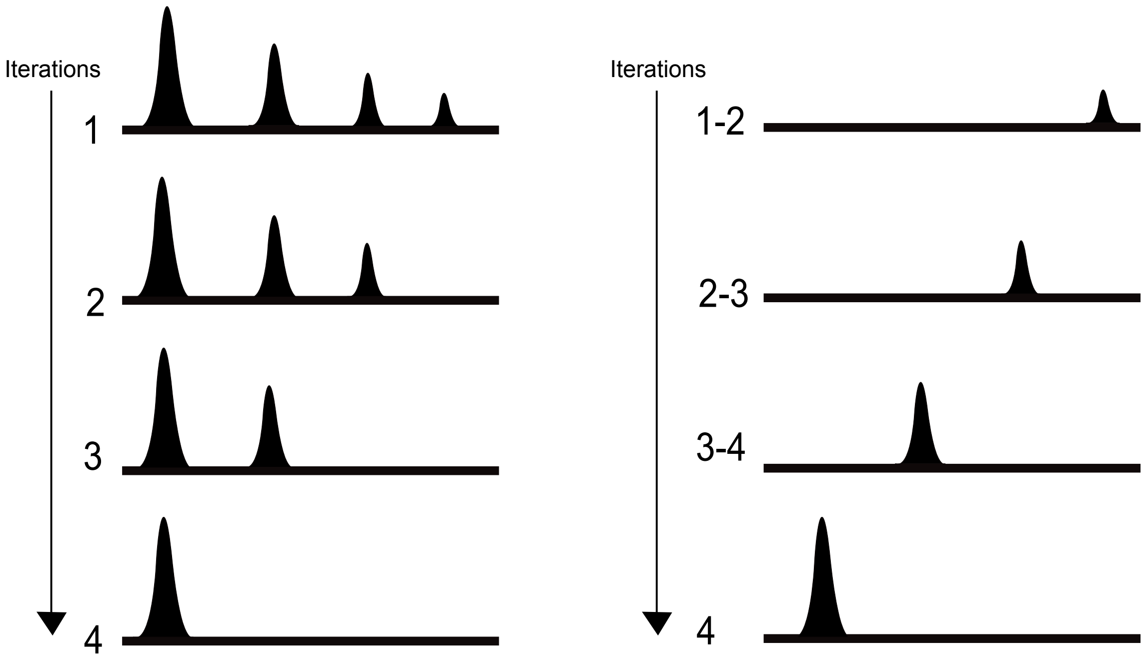


Figure 53: Left: 1: Shows imagined cross section across stars of different intensity and width. Width is the criteria for sorting. 2-4: the smaller connected areas are removed. Right: To positively identify stars of different sizes, the levels on the left containing areas with bigger areas is subtracted from the level with a smaller area still present.

After this is completed, the images presented in the 2D part of the 3D matrix is subtracted from each other along the starsize dimension. That is to say, subsequent images are subtracted from each other to find the difference between them, which is is area that has been removed. The result is illustrated on the right side of figure 53. The 2D images will now only contain connected areas of the size specified by the third dimension, which runs from 1 to the largest possible star size.

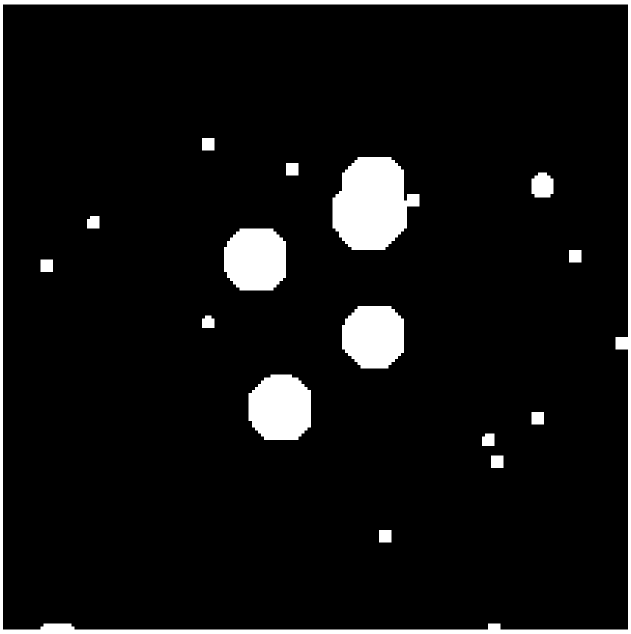
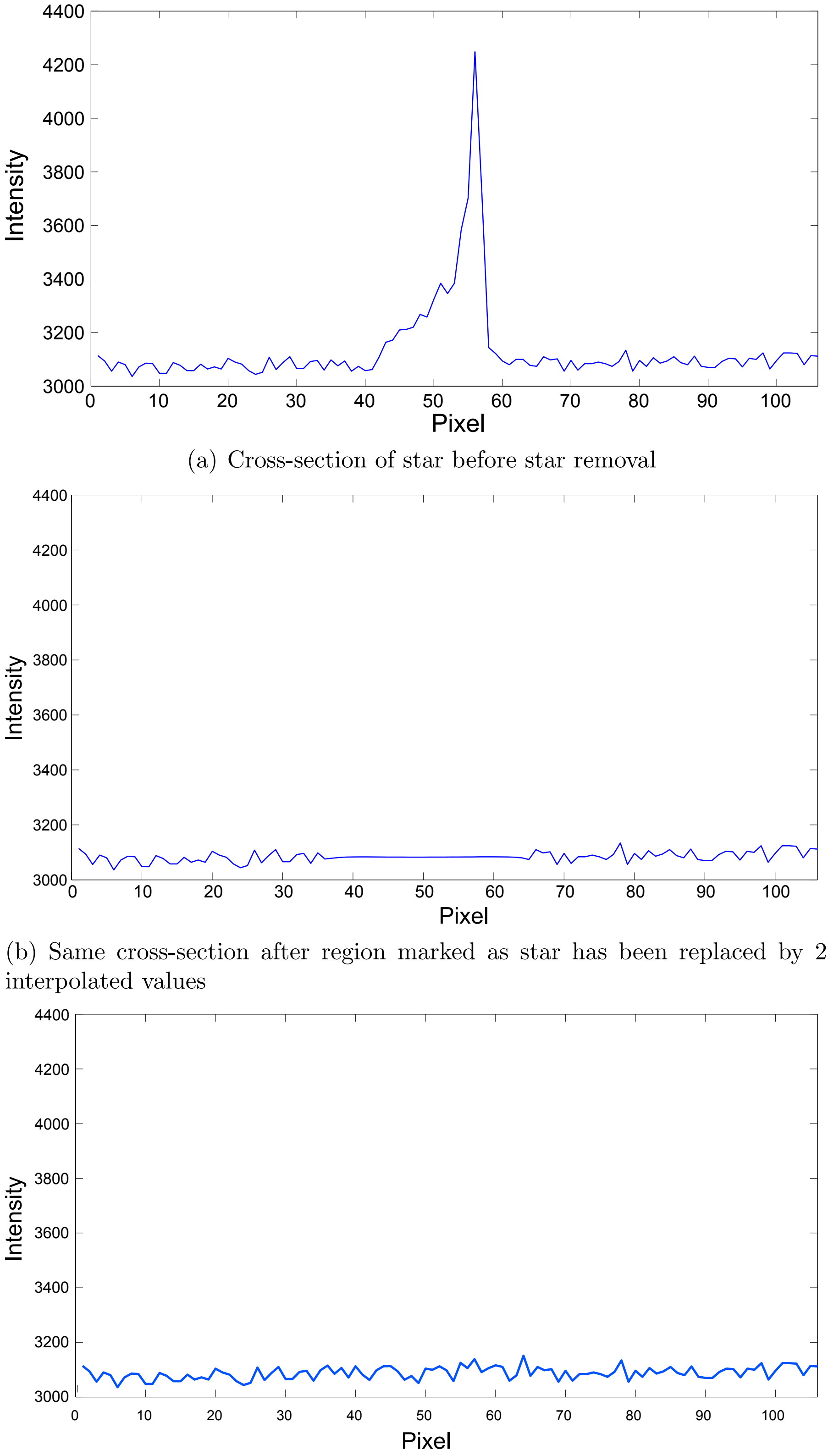


Figure 54: White regions mark which areas has been changed from the original image to the final image following removal of hot pixels and stars. Note how regions above a certain size have increased more in size more than others. This is to remove the effect of bright stars bleeding into surrounding pixels

Assuming the stars should be ideally circular, new structuring elements are made. These elements are disks with a radius based on the number of pixels present in each connected area, which is now a sorted size. The radius is initially based on this principle, for small identified stars. As the number of connected pixels in a region increase, a variable dependent on the number of connected pixel is added to the radius. Meaning that the larger a region is, the larger the ratio between the size of the region and the disk becomes. The disks are used to dilate the areas already identified to larger, circular areas. The image that has had it hot pixels, and initial stars removed is then subjected to p = roifill(p, BW); where BW is the new dilated logical image of star locations. The result of this is that the larger a star is, the larger the area surrounding it is selected for replacement of interpolated values. Then the noise previously sampled is added to the areas where new values has been interpolated to replace stars. This leaves a signal where the previous location of stars and hot pixels should be hard to identify.

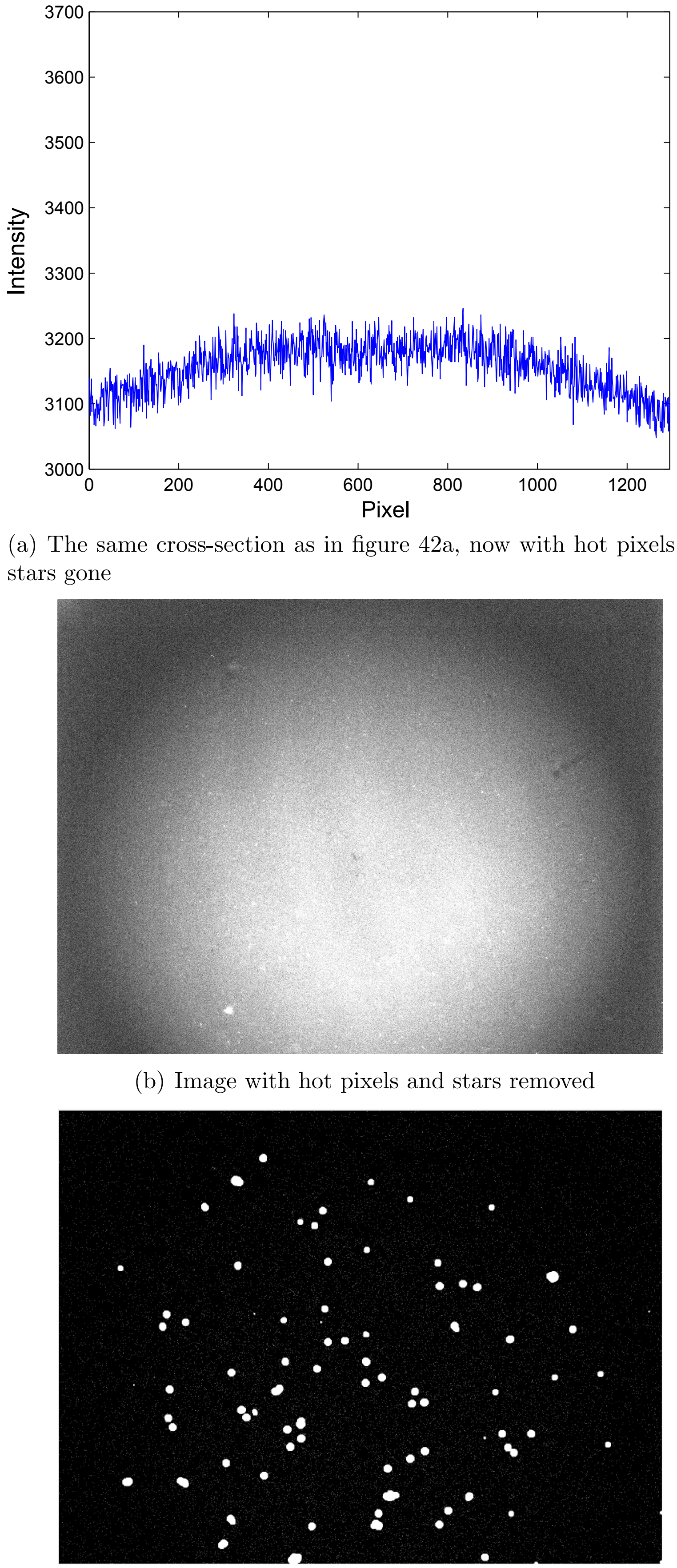


(a) Cross-section of star before star removal

(b) Same cross-section after region marked as star has been replaced by 2D interpolated values

(c) Same cross-section, now with random noise added to areas replaced by interpolated values.

Figure 55: Removal of star



(a) The same cross-section as in figure 42a, now with hot pixels and stars gone

(b) Image with hot pixels and stars removed

(c) White pixels represents change from before hot pixels and stars were removed, and after

Figure 56: Removal of star

8.2 Mapping to Geographical Coordinates

Just as the images used to test the image compression techniques had to be mapped to geographical coordinates, so must the images needed to test the 3D-FFT method. This is to obtain images were the distance between pixels corresponds to a certain geographical length. If the wavelength in pixels is found, the wavelength in km is easy to find. This gives a more correct representation of the waves themselves, as well as what the satellite will see. As opposed to the wide angle images taken with the ground based imager. The images are mapped in exactly the same manner as before.

8.3 Flatfielding by Image Subtraction

Once images have been mapped to geographical coordinates there will still be a trend present where the intensity seems to drop towards the edges. This can be seen in figure 57.

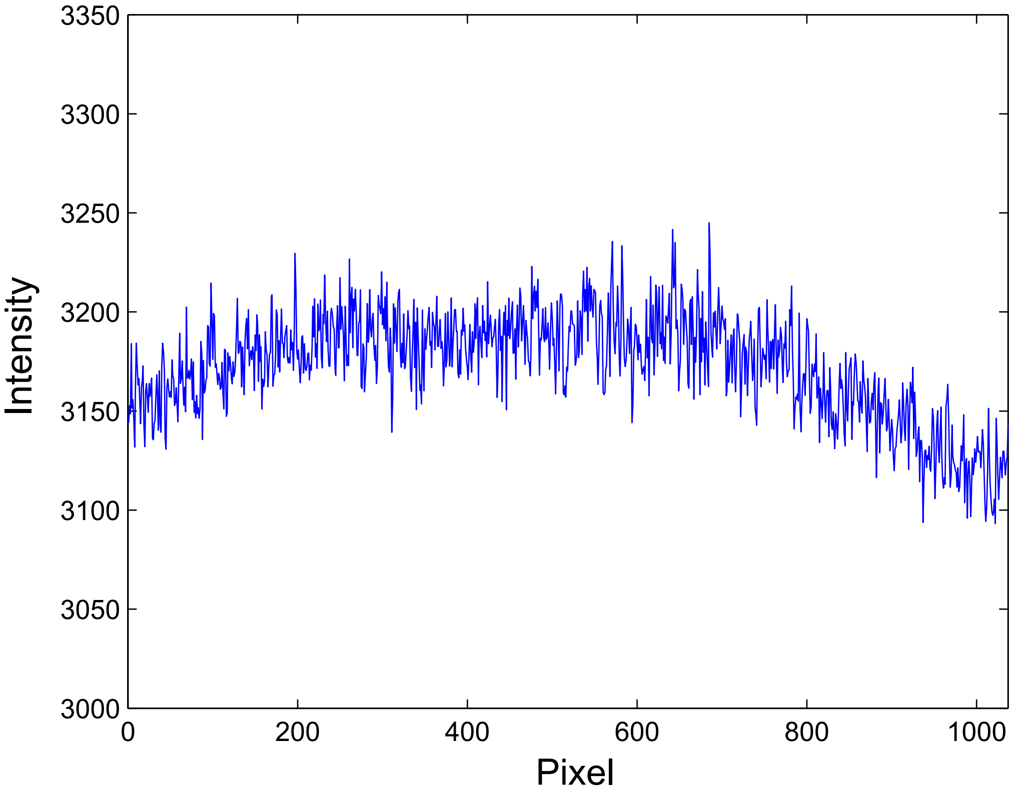
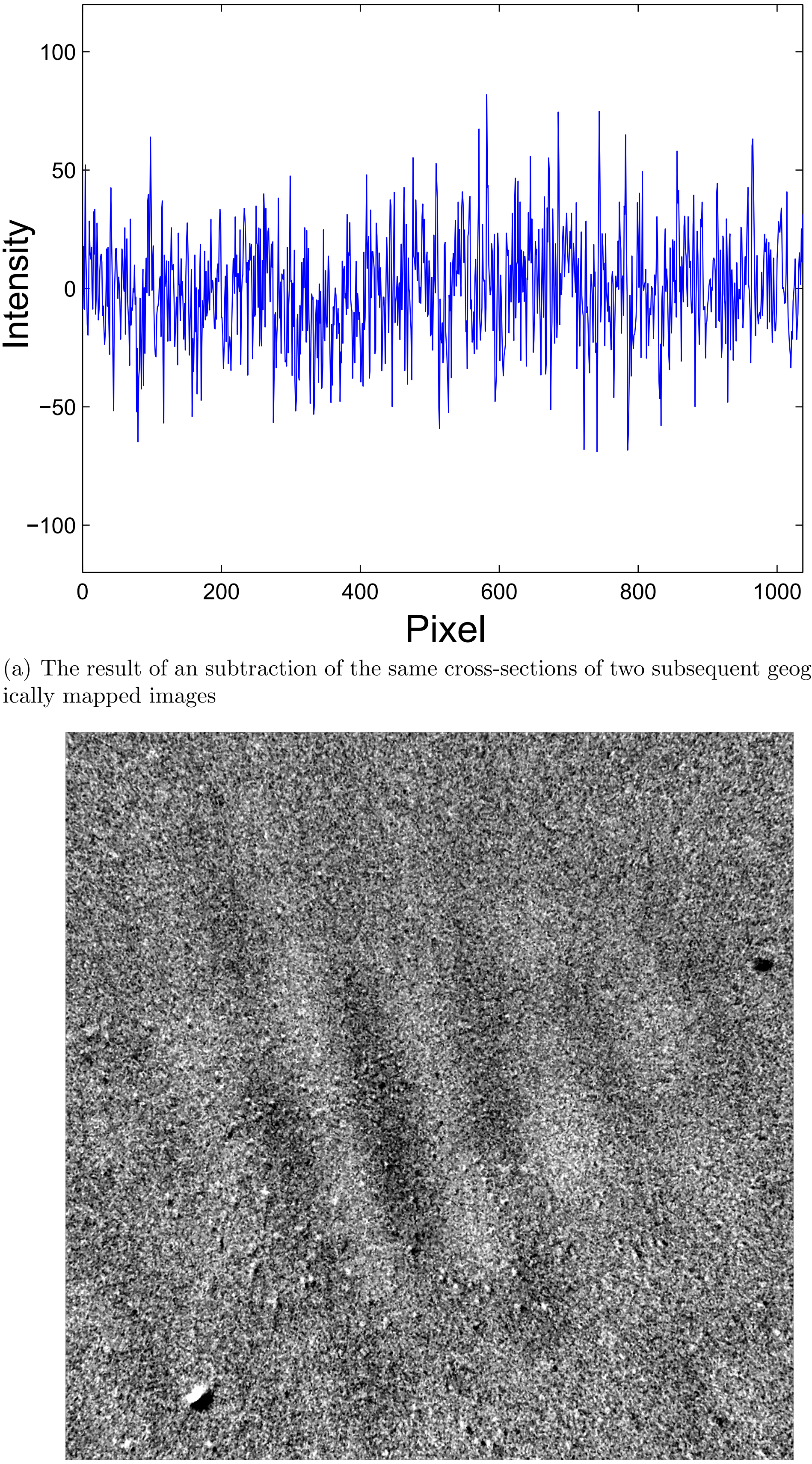


Figure 57: Cross-section of the image previously used, but after star removal and mapping to geographical coordinates

The main objective of this text is isolating the wave structure signal from any other signal present. To enhance the wave structure and at the same time eliminate the intensity variation introduced by the lens, two subsequent images are subtracted from each other. This will only leave the difference between them. This will consist of noise and difference in wave propagation that has taken place in the between the two frames. With some luck, given that the wave speed is suitable, the wave structure will be strengthened instead of eliminated. A subtraction of the profile in figure 57 from the next in the same image series is shown in figure 58(a).

The result of this subtraction show that the signal now has a slight wave structure, with an overall flat trend. The signal is however still quite noisy. Figure 58(b) shows how this operation has influenced the image itself.



(a) The result of an subtraction of the same cross-sections of two subsequent geographically mapped images

(b) End result of subtraction of two subsequent star-removed and mapped images

Figure 58: Removal of star

8.4 Orientation NSWE and Cropping to Square

An important objective for the satellite is identifying the orientation of the waves. This orientation will be given relative to the satellite, so in order to know the true orientation of the waves, the orientation of the satellite compared to earth is also required. In order to make it possible to test and verify the results of the following methods, the ground based images are rotated so that the axis of the image correspond to the geographical North-South axis. Zenith in the Dragvoll image series has already been identified, and the images has been mapped to geographical coordinates so that the center of the image corresponds to zenith. The polar star is then used as a reference to the location of north in the image. The image is then rotated about the center, so that the polar star will be located on a line directly above the center of the image. This rotated image will of course not fill all the space of a square image of the same resolution, so the result is cropped to the largest possible square that contains pixels with information in them. It is important to note that the images are taken looking from the ground up, so the left part of the image corresponds to east, while the right side corresponds to the west.

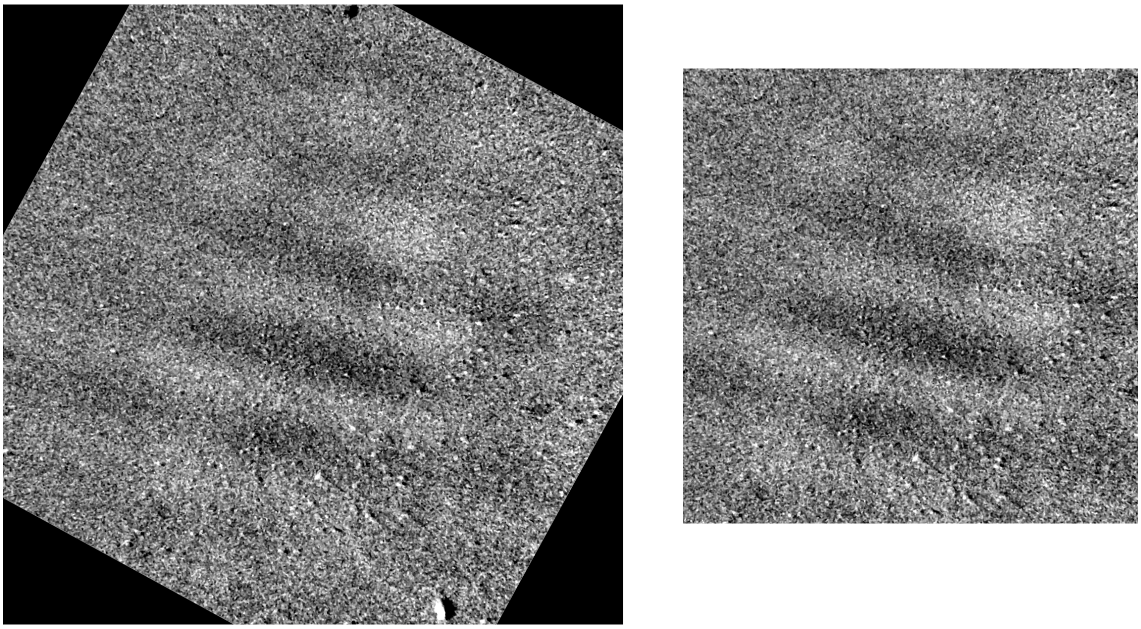


Figure 59: Image rotated to North-South orientation and then cropped to remove black areas

8.5 Windowing and Zero-Padding

In preparation for the following discrete Fourier transform that will be implemented by the 3D-Fast Fourier Transform-algorithm, there are some adjustments to the signal that are needed in order to obtain a usable result. These adjustments are needed because of the Fourier transform's workings.

Without going into details about the discrete Fast Fourier Transform, the following is stated after personal communication with Patrick Espy [5] and studies in "Numerical Recipes" [12] and "The Fast Fourier Transform and its Applications" [13]: The FFT will assume the signal it receives is infinitely repeating itself. This would would mean that the periodicity that is sought found in the image is corrupted by the possible periodicity that could be introduced by an infinitely repeating signal. This is mitigated by padding the signal with zeros in all dimensions. The 2D images effectively becomes larger with a "zero frame" around the original signal. The same is done in the time dimension.

The Fourier transform does not handle sudden changes in signal well, as many wave functions need to be added to create a sharp increase in signal strength. This will be the case when the images transition from signal to zero-padded areas. To mitigate this, a two dimensional Hanning-window is applied. This process consist of a 2D filter multiplied element by element with the signal. The filter is 1 in the central parts, and quickly drops to zero towards the edges. The same is done in the time dimension. The 2D-effect of this multiplication is seen in figure 60.

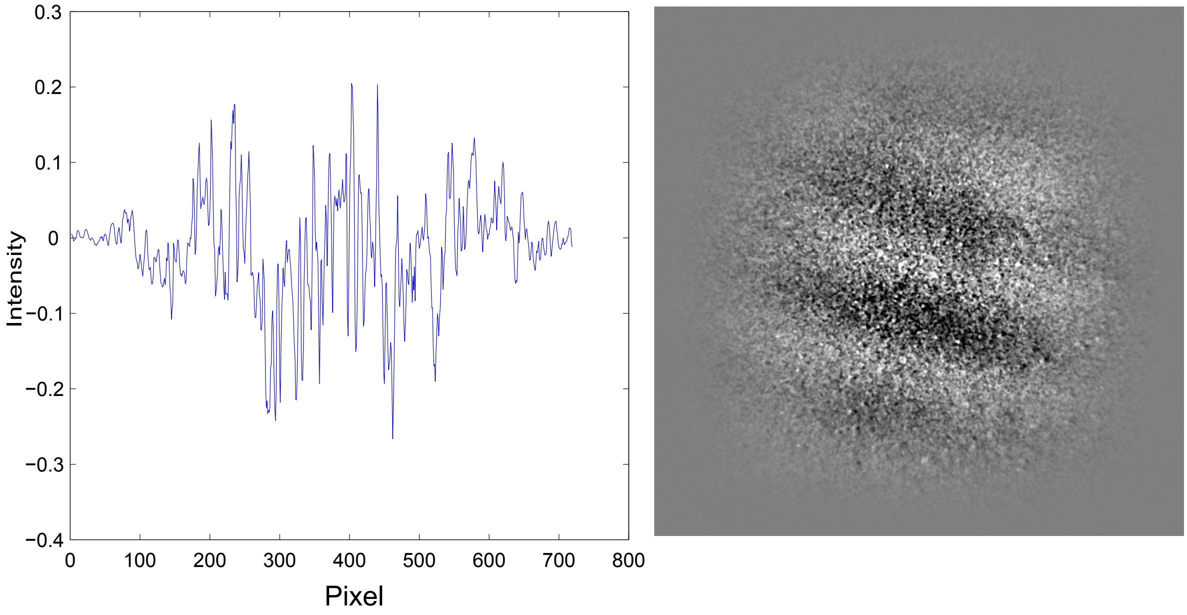


Figure 60: Spatial Hanning-window applied to rotated and cropped image and associated cross-section across waves. Image series is now ready for 3D-FFT

9 3D-FFT Method

This method is aimed at identifying the different waves that pass the satellite's field of view during a time series of images. This will mean that only the waves' parameters will be transmitted, as opposed to the series of images supplied by the video compression. This will represent a significant reduction of transmitted data.

The essence of this approach is a temporal 1D-FFT of a series of spatial 2D FFTs, combined to produce a final spectrum where a wave corresponds to one peak. As opposed to the the two symmetrical peaks that appears in the Fourier transform of a two-dimensional wave. The goal is a spectrum that is not symmetric and delivers unambiguous data about the waves' orientation, wavelength and direction over a time period. This method was designed by Gardner *et al.* [11].

9.1 1D Spatial and Temporal FFT

To properly explain how the aspects of the 3D-FFT will help unambiguously identify waves' wavelength and orientation it is practical to start with an ideal one-dimensional case. Imagine a cosine wave moving in positive x-direction; $y = cos(kx - \omega' t)$. Performing the spatial Fourier transform, it is possible to determine the location of peaks in the spectrum. The location is dependent on the wavenumber, which determines the wavelength. The complex numbers that the peaks will consist of will be dependent on $\omega' t$'s, which determines the propagation speed and direction.

$$\begin{aligned}\mathcal{F}_x[cos(kx - \omega' t] &= \mathcal{F}_x[cos(k(x - \frac{\omega' t}{k}))] \\ &= \sqrt{\frac{\pi}{2}} e^{-i\frac{\omega' t}{k}\omega}\big[\delta(\omega - k) + \delta(\omega + k)\big] \\ &= \sqrt{\frac{\pi}{2}}\big[e^{-i\omega' t}\delta(\omega - k) + e^{i\omega' t}\delta(\omega + k)\big]\end{aligned} \tag{11}$$

Equation (11) show that a wave will have two peaks of complex values in it's spectrum. Note how the magnitude of these are constant and equal, but that they rotate in opposite directions in the complex plane. The location of the peaks will be at $\omega = -k$ and $\omega = k$.

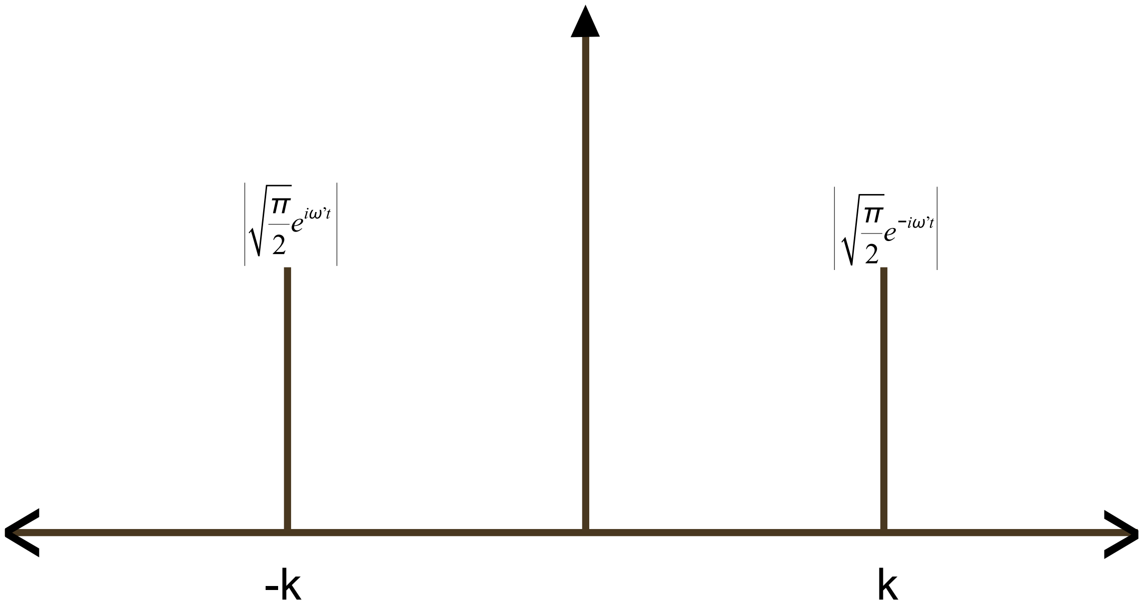

$\left|\sqrt{\frac{\pi}{2}}e^{i\omega' t}\right|$
$\left|\sqrt{\frac{\pi}{2}}e^{-i\omega' t}\right|$
-k
k

Figure 61: Spatial Fourier transform of time varying cosine-wave. The values at -k and k will oscillate between imaginary and real, the length of the vector will stay constant. The frequency of the oscillation depends the phase-speed of the wave

The next step is investigation of the temporal dimension. The spatial spectrum just obtained are snapshots of a wave passing the field of view. Sampling how each value along the spatial spectrum varies with time one will find that the values are zero, except at k and $-k$. Here the values will vary as $\sqrt{\frac{\pi}{2}}e^{i\omega' t}$ and $\sqrt{\frac{\pi}{2}}e^{-i\omega' t}$, respectively, as vectors along the time dimension.

The Fourier transform of these values will produce:

$$\mathcal{F}_t\left[\sqrt{\frac{\pi}{2}}e^{i\omega' t}\right] = \pi\delta(\xi - \omega') \tag{12}$$

$$\mathcal{F}_t\left[\sqrt{\frac{\pi}{2}}e^{-i\omega' t}\right] = \pi\delta(\xi + \omega') \tag{13}$$

So the Fourier transform of the time dimension vector on spatial spectrum position k and -k will have a peak π shifted ω' and $-\omega'$, respectively, from the origin.

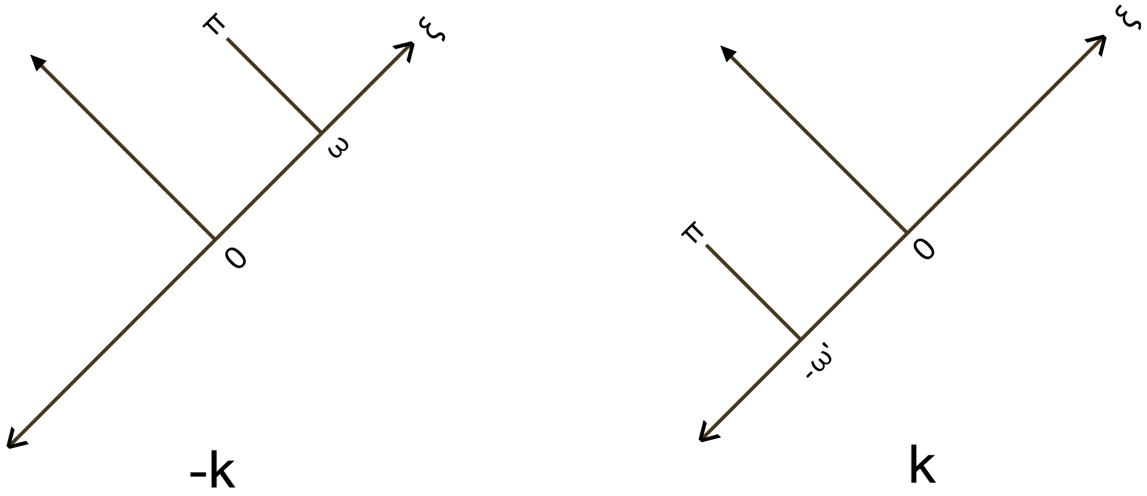


Figure 62: Shows the temporal Fourier transform of values sampled at position -k and k in the spatial Fourier transform. Tilted to illustrate that they move into the "paper plane" at k and -k in previous figure

9.2 Unambiguous Spectrum

So far the wavelength has been identified by determining the location of peaks in k-space. The single FFT of a wave produces an ambiguous spectrum of two symmetric peaks. With a temporal FFT of those peaks, new spectra have been made with peaks that are shifted depending on their location in k-space and the sign of ω'. These two peaks will be equally large and shifted equal length, but in different directions. The aim is a spatial spectrum where only the peak that corresponds to the waves' direction of propagation remains. In other words, a wave moving towards the right should only leave a peak on the right side of the spectrum, as opposed to the peaks symmetric about the axis that are produced by a standard spatial FFT.

To obtain this sort of spectrum the symmetric properties of the spectra are used. Looking at the temporal spectrum of the values sampled at k and -k in the spatial spectrum, one will see that they are anti-symmetric about the origin. This means that we can exclude one of the peaks in the further analysis with no loss of data. Looking only at the temporal spectrum sampled at k, the right hand side of figure 62, it is possible to split the spectrum into a positive and a negative half. Given that this example is a wave moving towards the right, we only want a peak at value k in the spatial spectrum. The next step is iterating through all values present in the positive part of the spatial spectrum and replacing them with a sum of the negative temporal spectrum of the respective values. This will leave a peak at k in the spatial spectrum, from the peak at $-\omega'$ in the temporal spectrum. The left part of the spatial spectrum, that contains a peak at -k is replaced similarly, but with a sum of the negative temporal spectrum from the positive spatial spectrum. The spatial indices are reversed when replacing the left part of the spatial spectrum with sums of the negative temporal spectrum from the positive spatial spectrum. All these values are zero in this example. This will leave a new spatial spectrum that only has a peak at positive k, as was sought.

Had the wave propagated to the left the value of ω' in equation (12) and (13) would be negative. The spectrum from equation (13) which corresponds to positive k, would be mirrored about the origin compared to its previous value. The way that this spectrum is sampled to create the unambiguous spectrum will leave a peak only on the left side, corresponding to a wave propagating to the left.

Figure 63 shows how the different steps in the process from image series of waves to one final spectrum showing the properties of the wave structures that has passed in the duration of the series. This is an expansion of the 1D wave just explained, to the 2D waves in question. The main principle still applies, with temporal FFTs of only the right half complex plane produced by the 2D spatial transform.

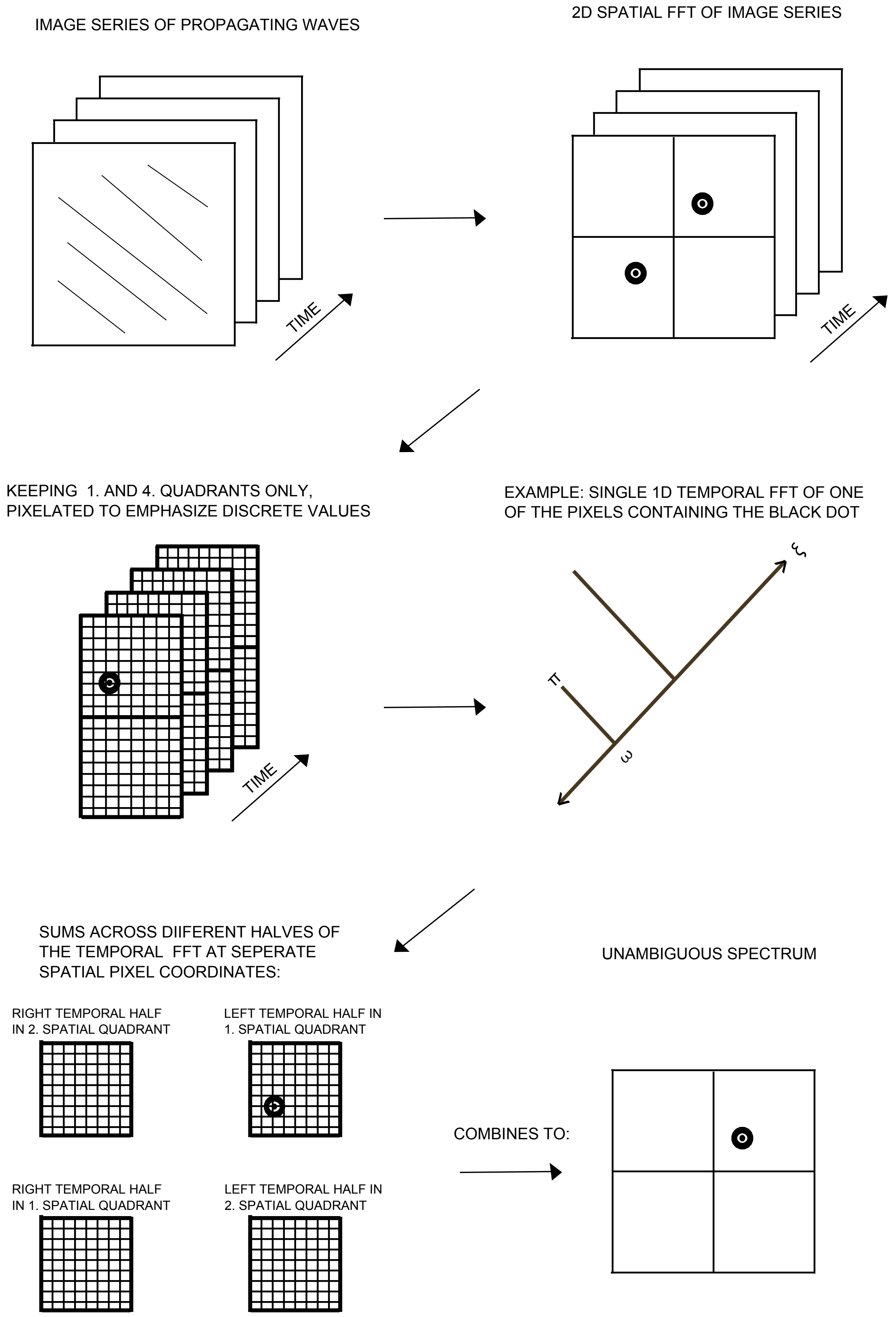


IMAGE SERIES OF PROPAGATING WAVES
TIME
2D SPATIAL FFT OF IMAGE SERIES
TIME
KEEPING 1. AND 4. QUADRANTS ONLY,
PIXELATED TO EMPHASIZE DISCRETE VALUES
TIME
EXAMPLE: SINGLE 1D TEMPORAL FFT OF ONE
OF THE PIXELS CONTAINING THE BLACK DOT
ξ
π
ω
SUMS ACROSS DIIFERENT HALVES OF
THE TEMPORAL FFT AT SEPERATE
SPATIAL PIXEL COORDINATES:
RIGHT TEMPORAL HALF
IN 2. SPATIAL QUADRANT
LEFT TEMPORAL HALF IN
1. SPATIAL QUADRANT
RIGHT TEMPORAL HALF
IN 1. SPATIAL QUADRANT
LEFT TEMPORAL HALF IN
2. SPATIAL QUADRANT
COMBINES TO:
UNAMBIGUOUS SPECTRUM

Figure 63: Simplified illustration of production of unambiguous spectrum

9.3 Spatial 2D-FFT

The data to be processed is presented as 2D-image series in time. So far, the only discussion of Fourier transform has been of 1D-waves. The waves in question is of course 2D, so the spatial transform must also be 2D. For this, the MATLAB-function fft2(); is used. This is basically a 1D transform along the rows of the image, followed by the transform of those complex values along the columns of the image. The difference from 1D to an expansion to 2D is that the peaks are located in a 2D spectrum, still centered about the origin, but rotated to match the orientation of the waves in the image. A cross-section in an arbitrary orientation of a 2D spectrum, will be equal to a 1D spectrum of values sampled in the same orientation in the real values. This is shown in figure 64.

Notice how the axis ”Pixel” is the number of pixels in the sampled line of the direct FFT. That means that the location of peaks shown here does not directly correspond to values in k-space. The conversion is done after the relevant peaks have been identified at the end of the full process. This is so that the conversion only has to be done on a few numbers in the satellite, as opposed to converting back and forth on all images in both the spatial and temporal domain.

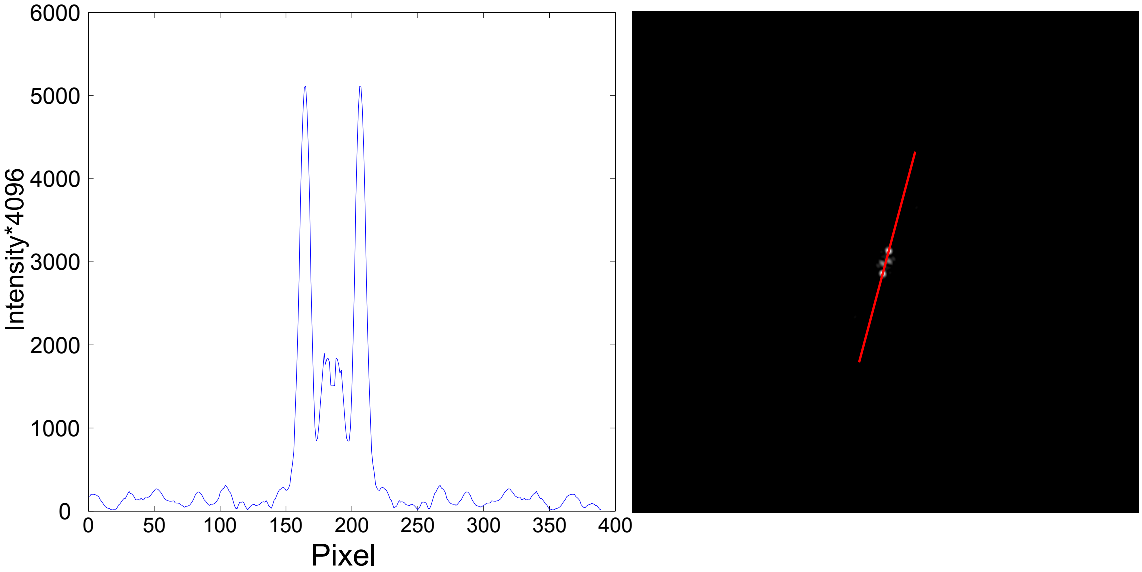


Figure 64: Cross-section of 2D-FFT

9.3.1 Optimal Filtering and Reconstruction of Waves

As can be seen in figure 64, the two-dimensional Fourier transform of a the ground based image gives two distinct peaks. This is as predicted from the theoretical treatment of a pure cosine wave show in figure 61. Figure 64 does however also contain more information than the one present in the two dominant peaks. These parts of the signal are the result of noise present in the image and a non-perfect Fourier transform. It is therefore of interest to isolate the peaks, do an inverse transform of them, and through that find the true wave signal, without noise.

A 2D version of the Optimal Wiener Filtering described in Numerical Recipes [12] is implemented to remove the signal parts not part of the wave associated peaks. The filter is presented as the following:

$$\Theta = \frac{|S(f)|^2}{|S(f)|^2 + |N(f)|^2} \tag{14}$$

This filter assumes that the signal in the spectrum can be split into actual signal and noise. $S(f)$ denotes the actual signal, while $N(f)$ denotes the noise. This means that equation (14) will give a filter that has a value close to 1 whenever $|S(f)|$ is different from zero and large compared to $|N(f)|$. Whenever $|S(f)|$ is different from zero it will be a lot larger than $|N(f)|$ giving a roughly equal number for the denominator and numerator. When $|S(f)|$ is zero, $|N(f)|$, which is modeled as a constant, will dominate and the fraction will be close to zero. When this filter is multiplied with the signal, element by element, it will leave the peaks relatively intact and leave everything else close to zero.

$|N(f)|$ is modeled as a constant, given by an average value from the flat areas surrounding the central peaks in figure 64.The location of the peaks of interest, their maximum and FWHM is gathered from the spectrum. This is used to create a modeled signal, as the peaks are assumed to be Gaussian. σ is given as FWHM/$2\sqrt{2\ln 2}$

$$f(x) = max \cdot e^{-(x-location)^2/2\sigma^2} \tag{15}$$

With the combination of the modeled $|N(f)|$ and $|S(f)|$, and the definition of the filter, equation (14), the 2D filter is created. A cross-section is showed in figure 65. The values of this filter will be multiplied with the values of figure 64 element by element. This will leave the dominant peaks intact, still containing the information that is sought. An inverse transformation and the reconstructed waves are shown in figure 66.

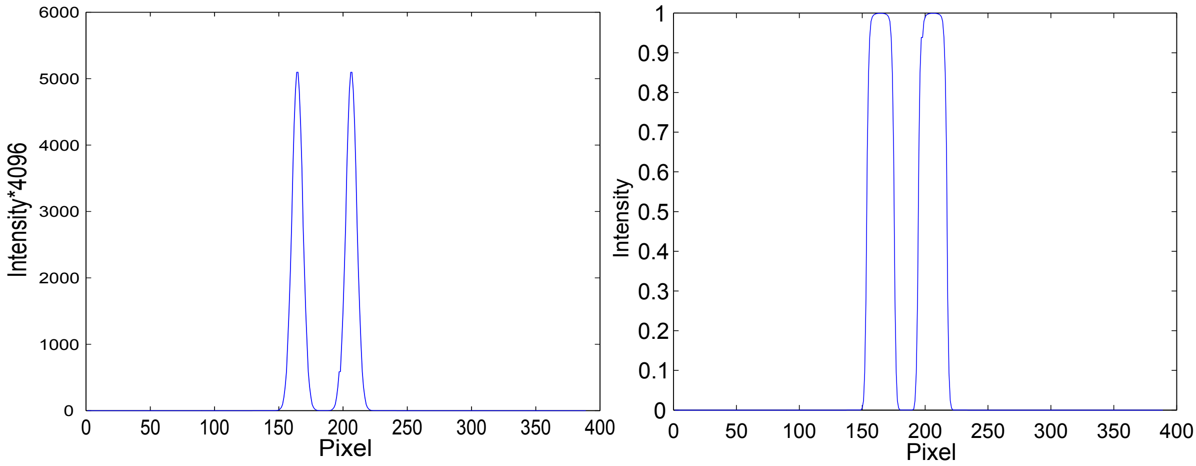


Figure 65: Cross-section of modeled signal $|S(f)|$ (left), and filter Θ (right)

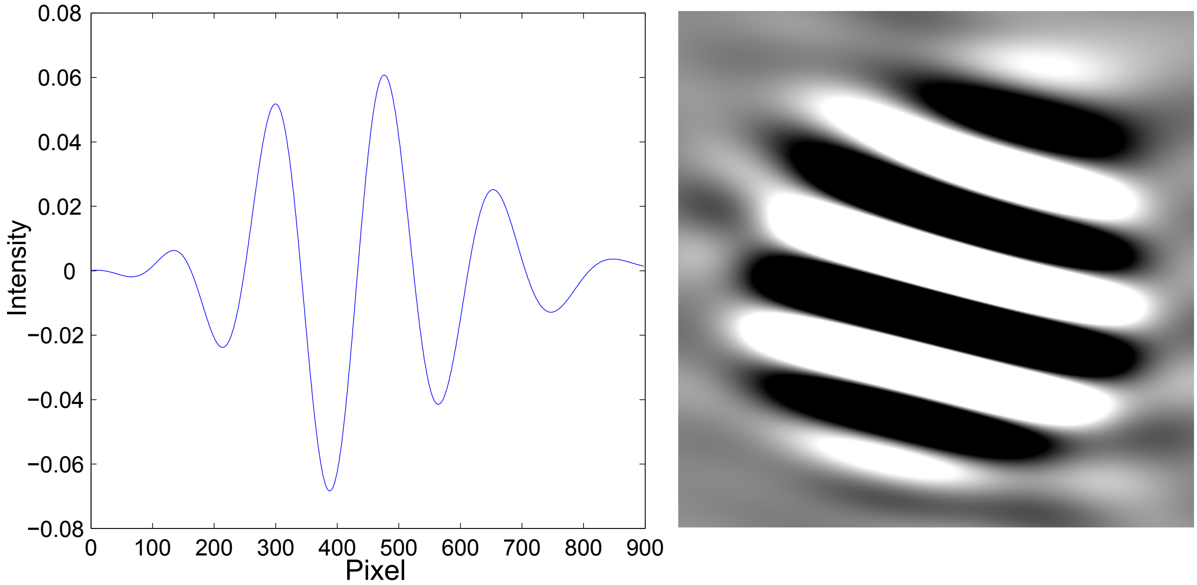


Figure 66: Cross-section of a reconstructed wave (left), and image of the same wave wave following the filtering of the 2D spectrum (right)

9.4 Identifying Waves in Time Series of Image Through 3D-FFT

To test the 3D-FFT approach, and the validity of its results, a synthetic image series was made. This way the input to the method can be controlled and compared to the output. The series included two waves with different wavelengths and orientations. One was propagating downwards, the other to the right. See figure 67. In figure 68 the unambiguous spectrum produced by the 3D-FFT method is shown.

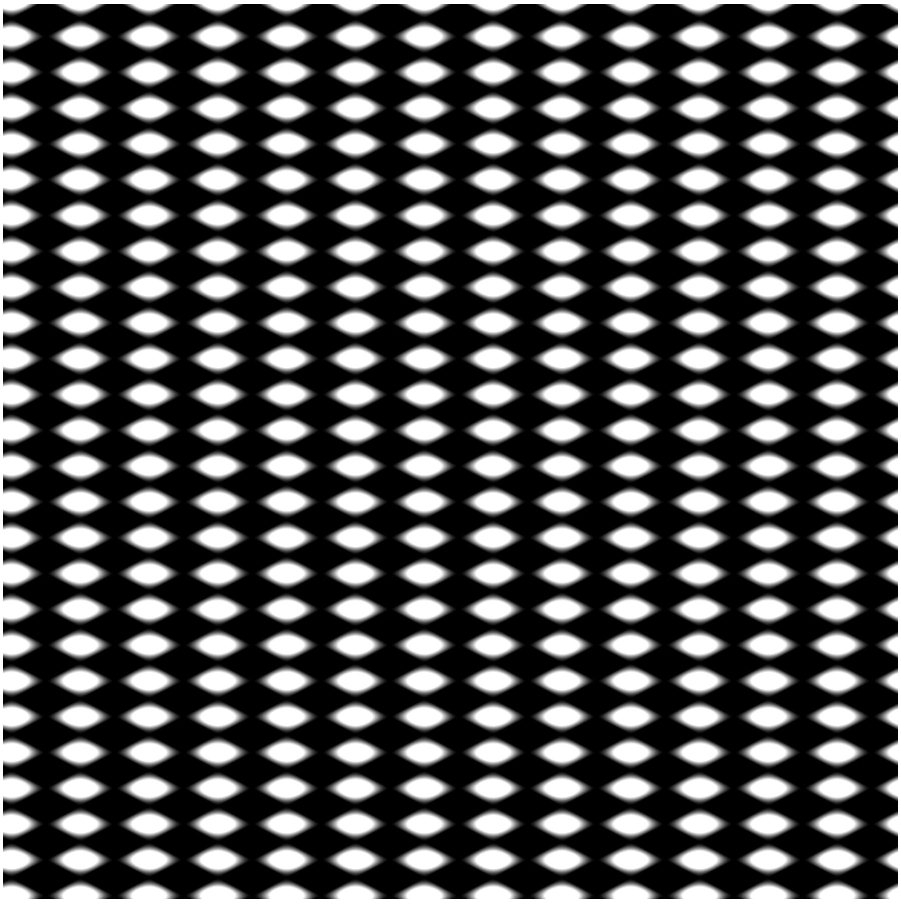


Figure 67: Test image of two waves of different wavelengths and orientations. One is propagating straight down, the other to the left

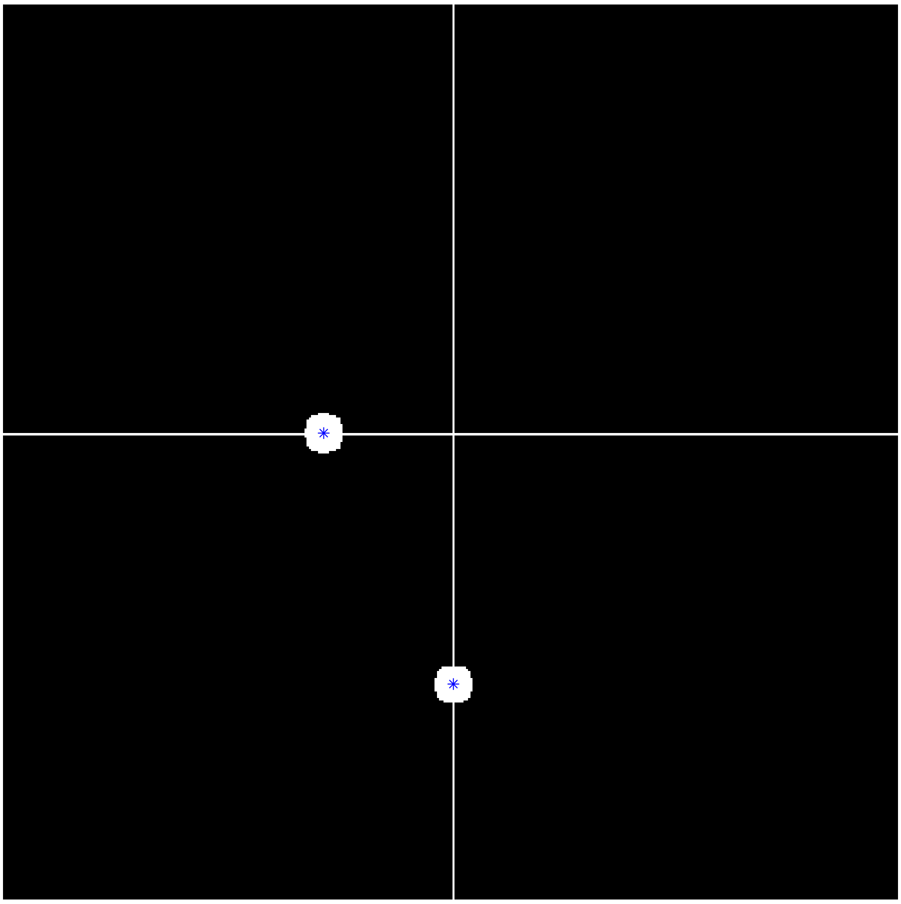


Figure 68: Zoomed view of unambiguous spectrum. Thresholded to find location of most intensive regions, and with that information about the waves. Cross hair added for reference to origin. Axis are in k-space with arbitrary units. Longer distance from origin means smaller wavelength

The identification of peaks is made by making a new matrix of the same size as the spectrum. Here, indices of pixels are marked off as white when the corresponding pixels in the spectrum is above a certain value. This creates white regions over at matching location to the peaks. The centroid of this region is then found. The pixel distance from the centroids to the origin and the azimuth angle is calculated. The pixel distance is then converted to distance in k-space and then to wavelength in pixels.

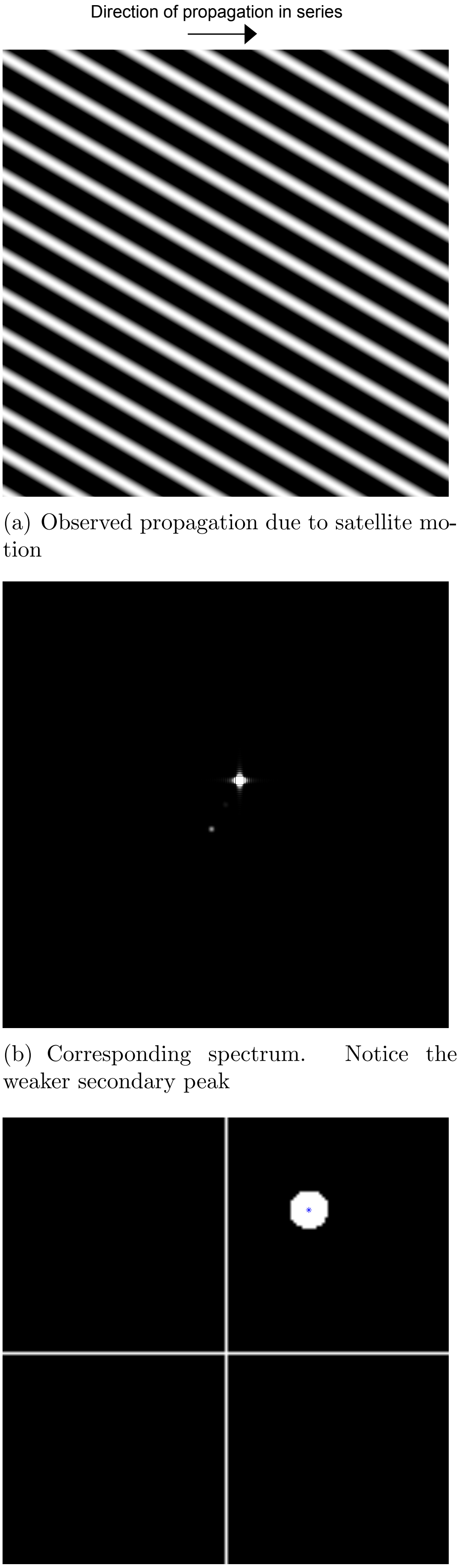
Table 3: Comparison of in- and output. The measured wavelength seem to converge towards the actuall values, dependent on the chosen threshold for identifying peaks.

Peak	Actuall azimuth	Measured azimuth	Actuall	Measured
’	of propagation	of propagation	wavelength	wavelength
(-)	(Deg)	(Deg)	(Pixel)	(Pixel)
1	270	270	72.7531	72.73
2	180	180.2563	36.97	36.92

9.4.1 Identifiyng Wave Series Seen by the Satellite

The satellite will be orbiting the earth at high speeds. This will influence the apparent propagation of waves seen by the satellite. The waves in the previous section was seen as they would have been from earth, where the waves propagate orthogonally to the wavefronts. Because of the satellite's high speed, the wave's relative slow motion can be dwarfed. Thus, waves can be seen propagating in other directions than their wavefronts. If the satellite's orbit is parallel to the wavefronts, the resulting spectrum will contain two symmetric peaks, as if it was made from a 2D-FFT. These peaks can be identified as before, with orientation and wavelength.Their symmetric properties will tell that one of them can be eliminated arbitrarily, as the direction of propagation is not defined.

The angles between observed orthogonal and parallel propagation will also give symmetric peaks. The ratio of their intensity will vary between 1 for wavefronts parallel to the satellite's orbit and close to zero for wavefronts orthogonal to the orbit. The ratio of intensity can be compared with the telemetric data of the satellite's orientation and speed to give more information. This is field that requires more attention.



(a) Observed propagation due to satellite motion

(b) Corresponding spectrum. Notice the weaker secondary peak

(c) Thresholding identifies only the stronger peak

Figure 69: Wave motion seen by the satellite

9.5 Identifying Real Waves

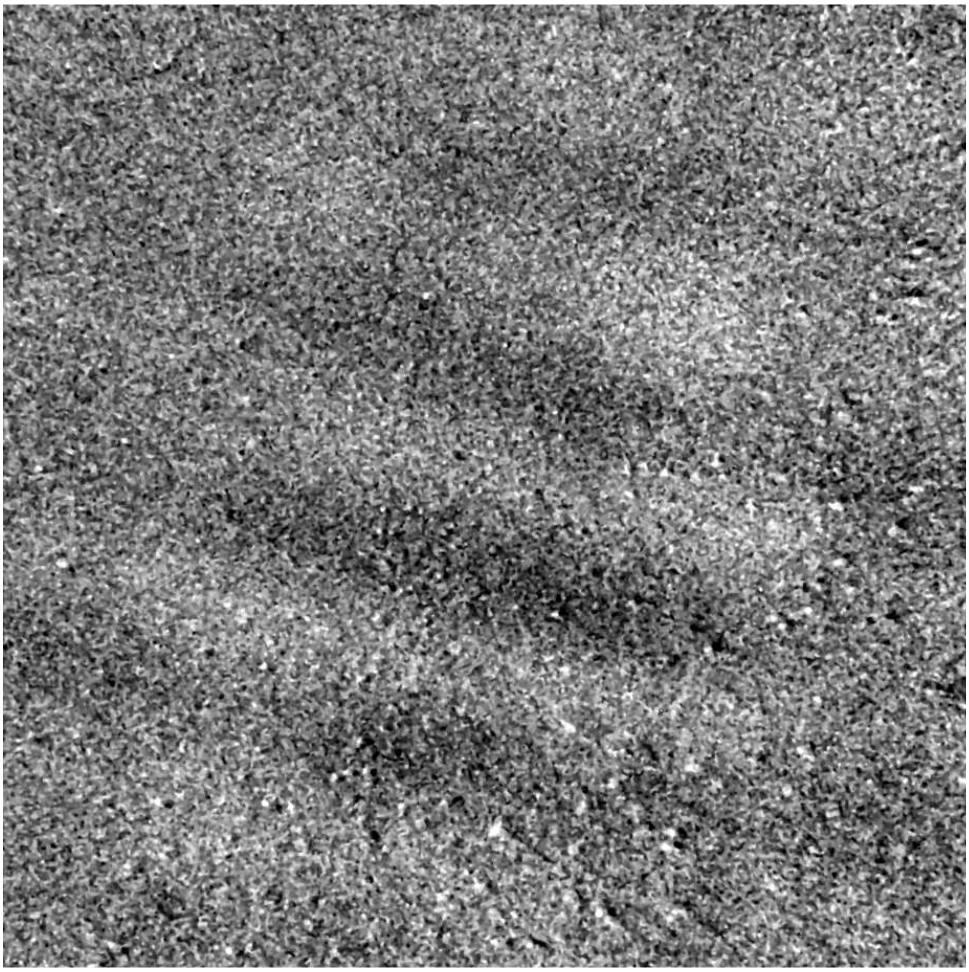
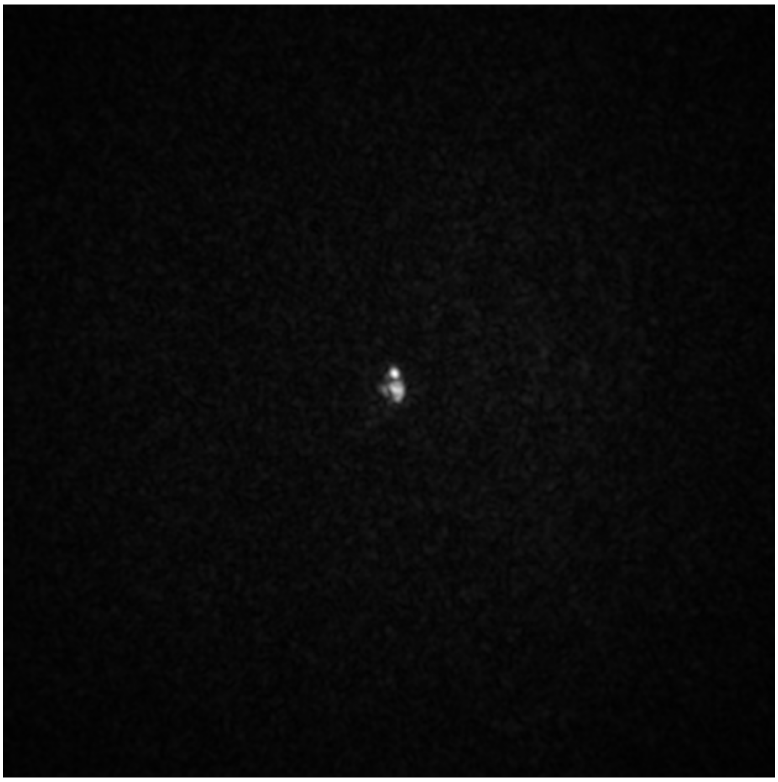
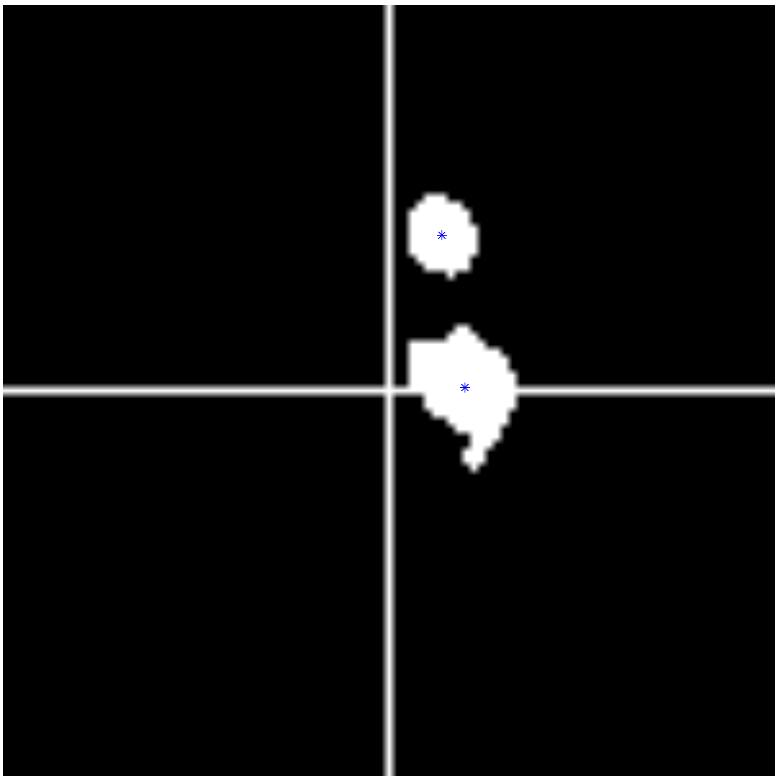


Figure 70: Image part of the series used to test the 3D-FFT method

The series of images that contain the image previously used as an example was used to test the 3D-FFT method on real data. In the start of this series there is a smaller wave moving in one direction, before this disappears and is replaced by a bigger wave moving in a different direction. Keep in mind that these images are North-South-East-West, as opposed to the more intuitive North-South-West-East as known from standard maps. The cause of this is that the images are taken from the ground up. This is compensated in the identification where the needed adjustments are made. The angles presented are calculated as azimuth clockwise from North.



(a) Unambiguous spectrum showing some noise and two distinct peaks implying that two waves in different directions have passed during the time of the image series



(b) Identifying pixels above a certain intensity and the centroid of the resulting blobs gives information about wavelengths and orientations. Shows a zoomed view relative to previous figure. Cross hair added for reference of relative distance from origin and orientation

Table 4: Extracted wave parameters

Peak ,	Azimuth ,	λ from (920x920)	λ from (920x920)	λ from (320x320)	λ from (320x320)
(-)	(Deg)	(Pixels)	(km)	(Pixels)	(km)
1	340	205.11	33.44	82.75	33.05
2	271	463.82	75.62	178.89	71.44

In table 4, 920x920 is from mapped ground images, while 320x320 is the same image processed to emulate satellite image throug bit binning. Pixel wavelength is different, due to different resolution. Km wavelength is based on knowledge of KMs covered by pixels in mapped image and simulated satellite image.

10 Comparison of Compression and 3D-FFT

A direct comparison of the two approaches is not straight forwards. There main reason to this is hot the efficiency of the compression algorithm depends on the SNR, and the not-yet-implemented motion compensation. Assuming a satellite-image resolution of 128x128 pixels, after on-board bit binning, an indication of the differences can still be made.

$$(128 \times 128) \cdot 8[\text{bytes}] \cdot (\text{Compression}) \cdot N_{\text{images}}$$

$$= 131072 \cdot (\text{Compression}) \cdot N_{\text{images}}[\text{bytes}] \tag{16}$$

is the the amount of data that has to be sent in order to obtain one image. The compression is based on video compression and thus needs to send several of these images to work. After this data is sent, the images can be reconstructed and data on the waves extracted. However, the algorithm can only use a certain compression rate before the wave data is corrupted.

The 3D-FFT is able to identify waves present in a series of images, with data on wavelength, orientation and intensity that otherwise would be extracted after image compression and reconstruction. The information about the waves could then consist of three numbers per identified wave structure present during the time of registration, yielding:

$$3 \cdot N_{\text{waves}} \cdot 8[\text{bytes}] = 24 \cdot N_{\text{waves}}[\text{bytes}] \tag{17}$$

This is a considerate reduction in the amount of data that has to be sent. The compression rate should be very high to compensate the efficiency of the 3D-FFT.

11 Conclusion

This thesis show how gravity wave data from images taken by the NTNU Test Satellite (NUTS) can be obtained. The restriction of limited data transfer rates is mitigated through two separate approaches. One approach is the degree of image and video compression that can be applied to a series of images before data is corrupted. The second approach is implementing a 3D-Fast Fourier Transform (FFT) method that obtains the relevant data from the image series directly, and only sends these.

We have made simulated satellite images by mapping ground based images to geographical coordinates and then mapping to the coordinates seen by the satellite, as well as downsizing the resolution. These images have been compressed with a small selection of compression levels. We have extracted and compared the wavelengths before and after compression. From this, some emerging trends have been spotted and a preliminary suitable level of compression has been found for both an image with low SNR and an image with high SNR.

The method of compressing the images is shown to quickly deteriorate the quality of the images, making it difficult to recover the wavelengths at high levels of compression. We found the manual sampling of profiles and fitting wavelengths to be unpredictable and tedious.

A 2D-FFT could be applied to the reconstructed compressed images to recover the wave information in separate images. However, we found that the 3D-FFT could operate on-board the satellite and identify waves in a series of images. This will greatly reduce the amount of data sent. The process of using FFTs to identify waves was also shown to be more consistent than manually sampling profiles and fitting them to sine waves, even with an attempted minimization of error through fan-shaped sampling.

In addition the FFT approach can be used to determine which images to transmit, as the images themselves still will be of interest even if the waves have been identified. The images with the most pronounced waves could then be transmitted.

12 Future Work

- The current version of the compression algorithm is based on the images being motion compensated before compression. This has yet to be implemented.

- The satellite height has been set to the highest possible number for the simulated images. The height is limited by the field of view of the lens used to take the ground based image in Dragvoll. The final height of the satellite is still not known, so the compression algorithm should be tested with the correct height once it is known.

- The tools to analyze the effects of compression has been developed, but a large scale analysis has not been done. The effects of different satellite heights, quantization levels, dead-zones, compressed rate, wavelengths, amplitude and SNR should be further investigated and compared in order to give a better estimate of a suitable compression level.

- Depending on the SNR in the images and the choice of quantizer, a suitable bitcoder must be found. The current, stack run-coding, is not necessarily best for all conditions.

- The the data compression rate, dependent on quantization and SNR, along with associated recovered wavelength should be found, to weigh transfer rate against loss of data.

- All details about the fusing of quadrants to an unambiguous spectrum, implemented by [11] , is not fully understood by the author and should be investigated further. It is suspected that the chosen method is to mitigate edge effects in the built-in FFT of MATLAB.

- Introduce intensity scaling to properly obtain the amplitude of the waves.

- The identification of peaks is based on thresholding and finding centroids in the resulting blobs. If the blobs are to small in terms of pixels, the centroid may not be accurate. Additional interpolation of the regions determined to be above the threshold might therefore be needed.

- If there are waves of similar orientation and wavelength, the identified peaks might coincide and give one average wave instead of the actual two. It should be investigated if this is a relevant issue.

- Flat fielding may have to be implemented in another fashion as subtraction of images could supress the wave signal. If the images are not eliminated, their amplitude will be dependent of the phase difference between images.

References

[1] NASA *http://eol.jsc.nasa.gov/*

[2] FAS *http://www.fas.org/*

[3] M. Bakken. *Master thesis, Signal Processing for Communicating Gravity Wave Images from the NTNU Test Satellite* 2012

[4] F.J. Garcia, M.J. Taylor, and M.C. Kelley *Two-dimensional spectral analysis of mesospheric airglow image data*, 1997

[5] P.J. Espy, personal communication.

[6] BAS, *http://www.antarctica.ac.uk/*

[7] Swisscube, *http://swisscube.epfl.ch/*

[8] *http://en.wikipedia.org/wiki/Angle_of_view*

[9] StarCalc, *http://www.m31.spb.ru/StarCalc/*

[10] MATLAB, *http://www.mathworks.se/help/stats/procrustes.html*

[11] C.S. Gardner, M. Coble, G.C. Papen, and G.R. Swenson. *Observations of the unambiguous 2-dimensional horizontal wave number spectrum of OH intensity perturbations* 1996, Geophys. Res. Lett., 23, 3739-3742.

[12] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery. *Numerical Recipes, Third Edition.* Cambridge University Press, 2007.

[13] E. Oran Brigham. *The Fast Fourier Transform and its Applications*, Prentice Hall, 1988.

A Discarded Methods

A.1 Star Removal

Before the current star removal was implemented, a different approach was tested. The way stars present themselves in the night sky was exploited. The stars represent a sudden change in brightness from the otherwise dark background. The images are converted to black and white images with black represented as zeros, white represented as ones and grayscales given by numbers between 0 and 1. A simple searching algorithm is implemented, following the instructions in [4]. The algorithm seeks along each row and then each column and identifies stars if the value of a pixel changes more than a certain threshold from the previous pixel. Once a star has been detected, the algorithm is instructed to continue to iterate until the value of the pixel return to a reasonable difference from the background and replace these values with interpolated values from surrounding pixels not defined to contain a star. If the value of the pixels does not return to a reasonable difference from the background within a certain number of pixels, nothing is done. This number can be set to be the pixel size of the largest star, measured manually. If this restriction applies the algorithm has obviously found something larger than a star, which is not intended to be removed.

This algorithm proved to be very sensitive, needing finely tuned parameters to work properly. It could be made to function on an image after some trial and error, but was abandoned due to time concerns. An example of a nearly successful tuning of this can be seen in figure 71 and figure 72. In addition it was discovered that stars would not influence the extraction of the wavelength to a great extent with a careful selection of sampling area.

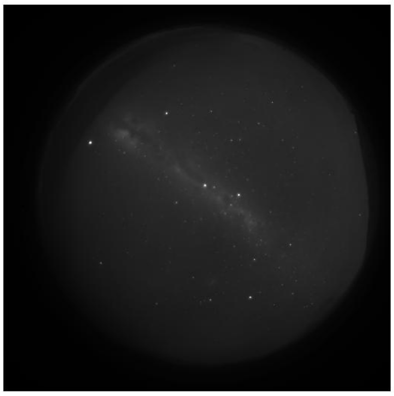


Figure 71: Image from Halley before star removal

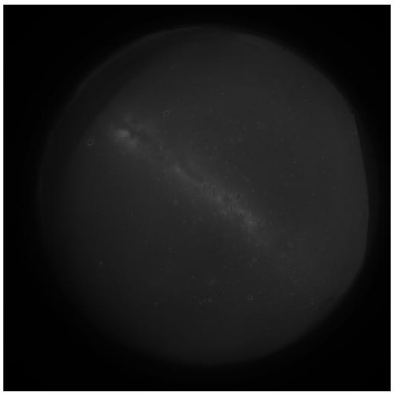


Figure 72: Image from Halley after star removal

A.2 Finding Zenith

Before the built in functions in MATLAB were used to find zenith in the all-sky images, another method made from the ground up was tested. It proved to supply unreliable results compared to the now preferred method. This scheme was based on iterating through all pixels in a grid of the same size as the ground based image. The grid had location of stars plotted, just like the filled pixels in figure 20. As the algorithm iterated though all the pixels it calculated the angle from each pixel to the plotted stars, comparing it to the known azimuth angles from zenith to the given star. The total difference in calculated angle and actual azimuth angle was stored, so that finally the pixel with the least difference should represent azimuth. The method was abandoned as it gave unreliable results with varying rotation of the grid, which should have no influence on the true location of zenith.

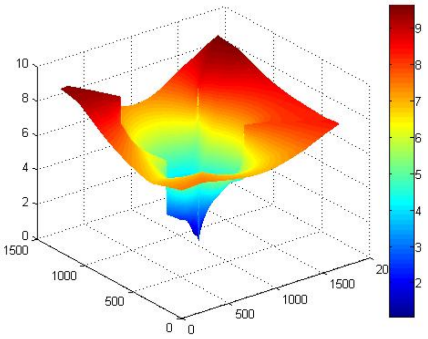


Figure 73: Color code representing total difference between known and calculated azimuth. xy-plane represents image grid.

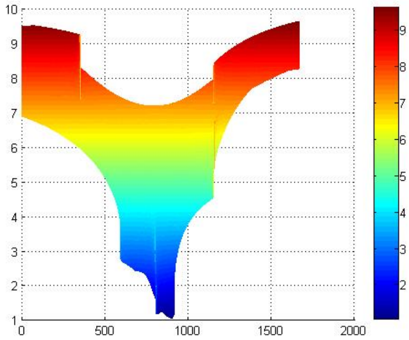


Figure 74: Color code representing total difference between known and calculated azimuth. xy-plane represents image grid.

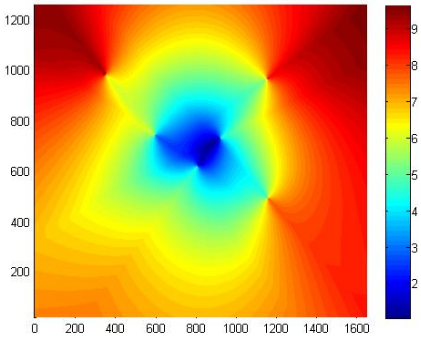


Figure 75: Color code representing total difference between known and calculated azimuth. xy-plane represents image grid.