2015

Microwave Radiometer (MWR) Evaluation of Multi-Beam Satellite Antenna Boresight Pointing Using Land-Water Crossings, for the Aquarius/SAC-D Mission

Bradley Clymer
University of Central Florida

MICROWAVE RADIOMETER (MWR) EVALUATION OF MULTI-BEAM SATELLITE ANTENNA BORESIGHT POINTING USING LAND-WATER CROSSINGS, FOR THE AQUARIUS/SAC-D MISSION

BRADLEY CLYMER
B.S. University of Central Florida, 2012

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science
in the Department of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2015

Major Professor: W. Linwood Jones

ABSTRACT

This research concerns the CONAE Microwave Radiometer (MWR), on board the Aquarius/SAC-D platform. MWR's main purpose is to provide measurements that are simultaneous and spatially collocated with those of NASA's Aquarius radiometer/scatterometer. For this reason, knowledge of the MWR antenna beam footprint geolocation is crucial to mission success.

In particular, this thesis addresses an on-orbit validation of the MWR antenna beam pointing, using calculated MWR instantaneous field of view (IFOV) centers, provided in the CONAE L-1B science data product. This procedure compares L-1B MWR IFOV centers at land/water crossings against high-resolution coastline maps. MWR IFOV locations versus time are computed from knowledge of the satellite's instantaneous location relative to an earth-centric coordinate system (provided by on-board GPS receivers), and a priori measurements of antenna gain patterns and mounting geometry.

Previous conical scanning microwave radiometer missions (e.g., SSM/I) have utilized observation of rapid change in brightness temperatures (T_B) to estimate the location of land/water boundaries, and subsequently to determine the antenna beam-pointing accuracy. In this thesis, results of an algorithm to quantify the geolocation error of MWR beam center are presented, based upon two-dimensional convolution between each beam's gain pattern and land-water transition. The analysis procedures have been applied to on-orbit datasets that represent land-water boundaries bearing specific desirable criteria, which are also detailed herein. The goal of this research is to gain a better understanding of satellite radiometer beam-pointing error and thereby to improve the geolocation accuracy for MWR science data products.

ACKNOWLEDGMENTS

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CONAE ....Comisión Nacional de Actividades Espaciales

HPBW..Half-Power Beam Width

IFOV...Instantaneous Field of View

MWR...Microwave Radiometer

NASA ....National Aeronautics and Space Administration

SAC-D ....Satelite de Aplicaciones Cientificas-D

T_B ....Brightness Temperature

CHAPTER 1: INTRODUCTION

Aquarius/SAC-D is a collaborative earth science mission between the National Aeronautics and Space Administration (NASA), and the Argentine Space Agency CONAE (Comisión Nacional de Actividades Espaciales). The aim of the mission overall is to provide global ocean maps of Sea Surface Salinity (SSS) with high spatial resolution (150km). Comprehensive descriptions of the mission science objectives, the Aquarius/SAC-D system design, and the implementation of the remote sensing of SSS are found in [1], [2], and [3].

For this mission, there are two key sensors associated with the SSS measurement. The prime salinity measurement instrument is the Aquarius (AQ) instrument (provided by NASA), which is a combined radiometer/scatterometer operated at L-band (1.41 GHz). The second instrument, the CONAE Microwave Radiometer (MWR), which operates at K-band (23.8 GHz) and K_a-band (36.5 GHz), provides simultaneous and spatially collocated ancillary environmental measurements; such as, columnar water vapor, ocean surface wind speed, and sea ice concentration.

The AQ/SAC-D mission's polar, sun-synchronous, low earth orbit was designed to satisfy the science requirements of global ocean sampling. Further, the selection of a terminator orbit (sun-rise ascending node) and the satellite bus/AQ instrument geometry are largely driven by the remote sensing geometry constraints. These require that the AQ instrument antenna beams are consistently pointed away from the sun, to minimize direct interception of solar radiation. Finally, the AQ/SAC-D orbit with a 380 km AQ measurement

swath results in 100% sampling of the earth surface, with a repeating ground track, every 103 orbits (approximately one-week).

1.1: Past Work and Current Problem

1.1.1:   Geolocation Errors for the Special Sensor Microwave/Imager

In 1987, the first SSM/I instrument was launched as part of the Defense Meteorological Satellite Program (DMSP) Block 50-2 F8 spacecraft. The instrument's primary purposes were to provide a variety of near-real time global maps: those of cloud water; rain rates; water vapor over ocean; marine wind speed, sea ice location, age, and concentration; snow water content; and land surface type, moisture, and temperatures [4]. Its highest-resolution channel was required to maintain a geolocation accuracy of roughly 7km; however, geolocation errors in excess of 20-30km were routinely observed. To mitigate this, several steps were implemented.

First, the satellite's ephemeris model was corrected to more precisely locate the instrument itself, a crucial step in geolocation. This is shown in the center panel of Figure 1. Next, the geolocation algorithm was altered to improve errors, which resulted from some numeric approximations. A fixed set of attitude adjustments were able to reduce geolocation error significantly; these yaw, pitch, and roll corrections could not be conclusively attributed to any one source (i.e. misalignment of antenna boresight, or actual satellite yaw, pitch, and roll attitude biases correction), but it was found that algorithmically, corrections could be made to bring errors to within an acceptable range as shown in the right panel of Figure 1.

```
FNOC
PREDICT      20-30 KM
EPHEMERIS    ERROR

SATELLITE    10-12 KM
EPHEMERIS    ERROR

SATELLITE
EPHEMERIS +
BORESIGHT    < 5 KM
CORRECTION   ERROR
```
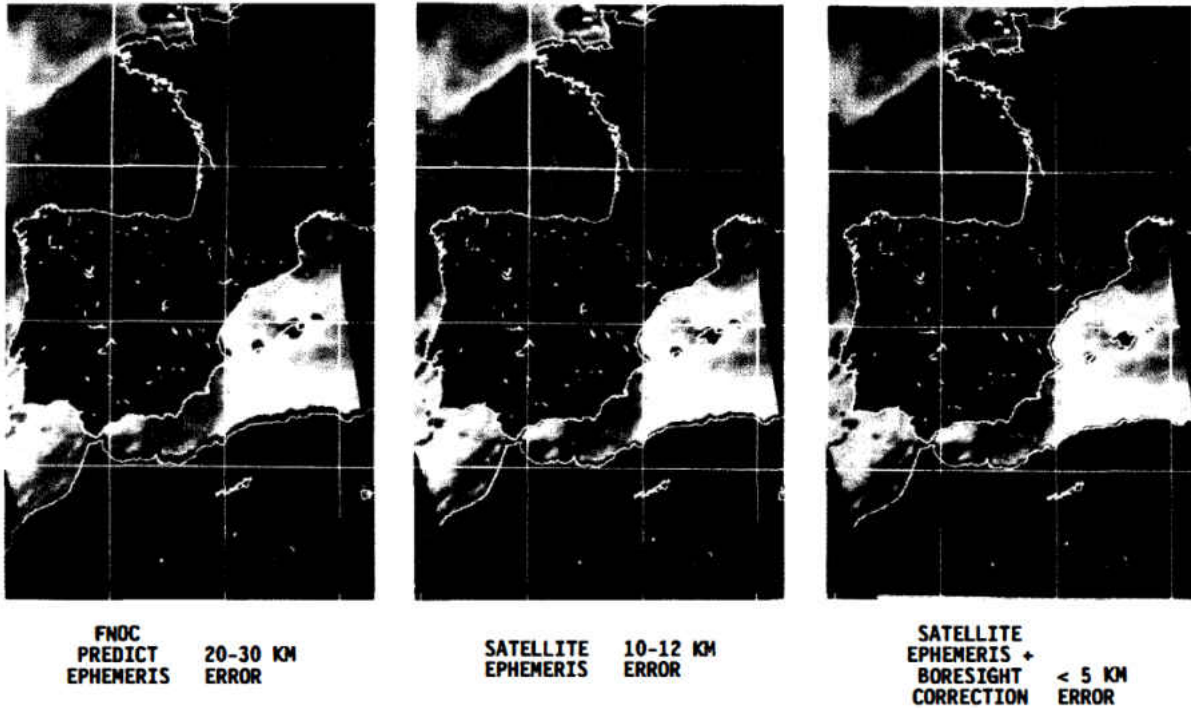
Figure 1: Raw geolocation results for SSM/I (left panel), results with ephemeris correction (middle panel), and results with both ephemeris and boresight correction (right panel).

1.1.2: Geolocation Errors for the WindSat

The WindSat polarimetric radiometer is the primary payload on the Coriolis satellite, which is the US Navy's "proof of concept mission" to demonstrate the passive microwave measurement the ocean surface wind vector from space [5]. The sensor was launched in 2003 and is still fully operational.

To validate the WindSat beam-pointing accuracy, brightness temperature (T_B) images of the Earth were used to establish the sensor derived location of land/water boundaries that were compared to high resolution map coordinates. To select the best antenna footprint location for this boundary, a technique was developed to determine the maximum rate of change of T_B data as the transition between regions of low brightness

3

temperature and high brightness temperature – ocean and land, respectively – these transition regions can be observed to indicate the most likely regions of coast.

Figure 2 (from Purdy et al. [5]) illustrates the use of this gradient technique, where the right-hand panel shows results before adjustment and the left-hand panel shows results after adjustment. With this technique the WindSat pointing errors have been reduced to be within error budget specifications of 0.05°, and the corresponding geolocation accuracy is below the required 5 km.
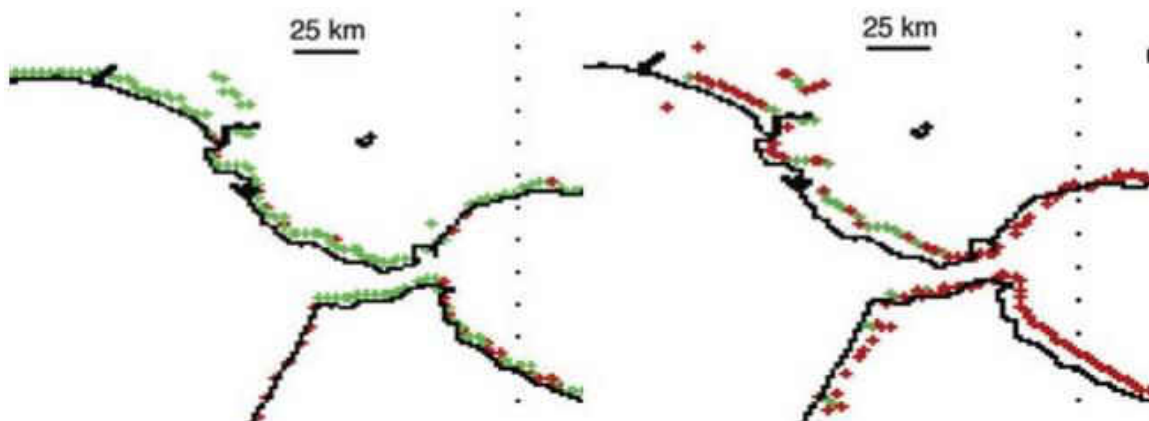
Figure 2: WindSat along-scan local maxima (green) and cross-scan local maxima (red), before correction (right) and after correction (left) from [5].

1.1.3: Geolocation Errors for the Special Sensor Microwave Imager/Sounder

As the next generation of the passive microwave imager on the DMSP, the first Special Sensor Microwave Imager/Sounder was launched in 2003, aboard the DMSP F-16 spacecraft [6]. This instrument featured 24 channels ranging from as 19 GHz to 183 GHz, with individual frequencies chosen to image specific parts of Earth's atmosphere and surface. Like SSM/I, SSMIS routinely exhibited 20-30 km of geolocation error at launch, and required calibration

to achieve the required geolocation accuracy, which is as small as 7 km for the surface-imaging beams.

As SSMIS is a conically-scanning microwave radiometer, it produces imagery from a portion of its circular scan swath as it passes over a region, as depicted in Figure 3. Clearly, errors in yaw calculation, beam central angle pointing, or start-time of scan (among other parameters) could cause erroneous geolocation.
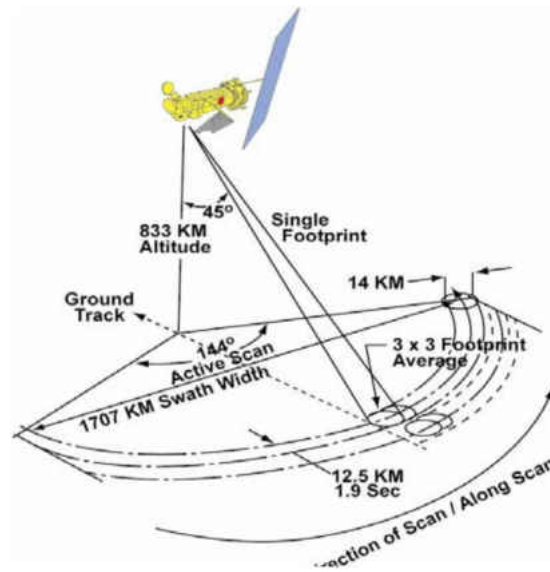
Figure 3: Scan geometry for SSMIS [7].

The technique adopted to combat these errors was an evolution of the technique used in the calibration of SSM/I and that of WindSat; specifically, partial derivatives of T_B in the "along scan" direction as well as the "along track" direction were taken to give a time series estimate of the points at which the instrument's field of observation crossed coast lines. In this context, "along scan" is the momentary direction of travel of the Instantaneous Field of View (IFOV) – a tangent line along the rotation – and "along track" is the direction of the satellite's forward travel.

After these partial derivatives were taken, the resulting ΔT_Bs were thresholded to determine areas of interest; regions with ΔT_Bs below a prescribed threshold were removed from consideration, and the areas of interest were then fit with a cubic spline. This spline was used to calculate the location of the time series maximum, and in turn to project it to latitude and longitude on the Earth's surface. An example of this result is shown in Figure 4 (from Poe et al. [6]), where the red points indicate locations of computed maxima, before and after correction.
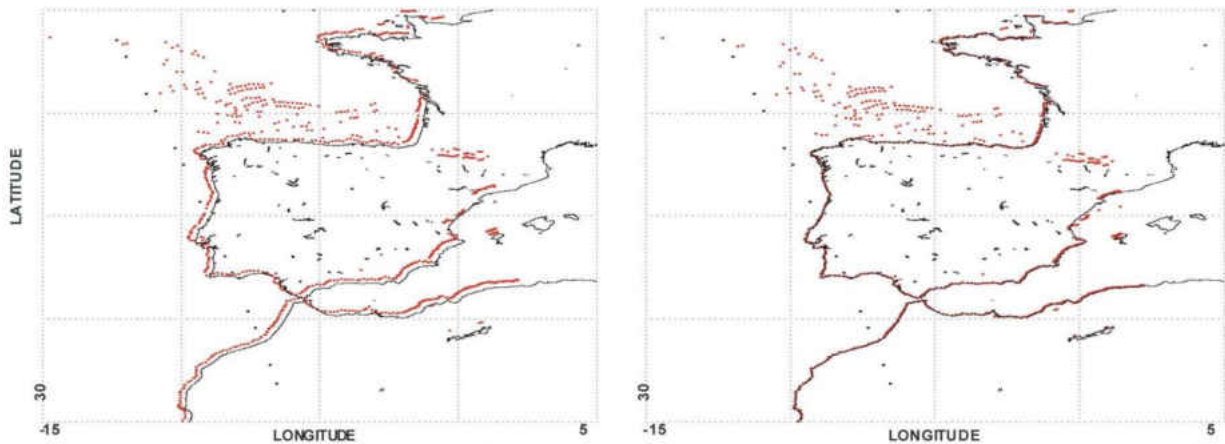
Figure 4: Time series derivative maxima before (left panel) and after (right panel) geolocation correction from Poe et al. [6].

After extensive study, it was determined that in the case of SSMIS, errors causing improper geolocation could be corrected by offsetting geolocation calculations with simple additive constants; the additions were applied to the cone angle of the feed-horns, the yaw of the spacecraft, and the scan start- and stop-times. Though the original sources of the errors were not definitively determined, the corrective constants mitigated errors to a range around 4-5 km. A sample result from the coasts of Portugal and Spain is shown in Figure 5, below. Examination of the black coastlines compared with the image of brightness

6

temperatures shows an obvious error in the left panel, and a significantly better agreement in the right panel.
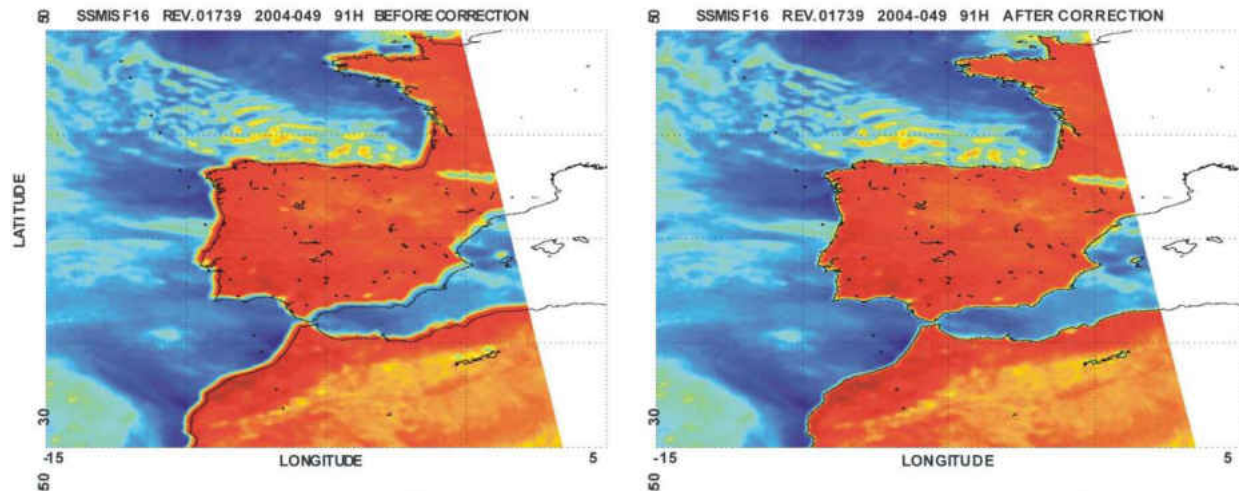
Figure 5: SSMIS Geolocation before (left panel) and after (right panel) geolocation correction [6].

1.1.4: Engineering Evaluation of MWR Multi-beam Satellite Antenna Boresight Pointing Using Land/Water Crossings

Prior to this current thesis, Catherine May [8] performed the initial evaluation of the MWR beam-pointing on AQ/SAC-D satellite. In her thesis, she used the maximum slope of T_B at land/water crossings and determine mispointing of MWR beams.

The first step of her analysis was to demonstrate a theoretical proof of concept: specifically, that a 1-Dimensional Gaussian antenna gain pattern, when convolved with an ideal "step function" brightness temperature at the coastline, would exhibit a maximum slope on the coastline, where the beam is filled exactly 50% by the ideal temperatures

7

representing land, and 50% by those representing water. A graphical 1-D illustration from her thesis [8] is shown in Figure 6.
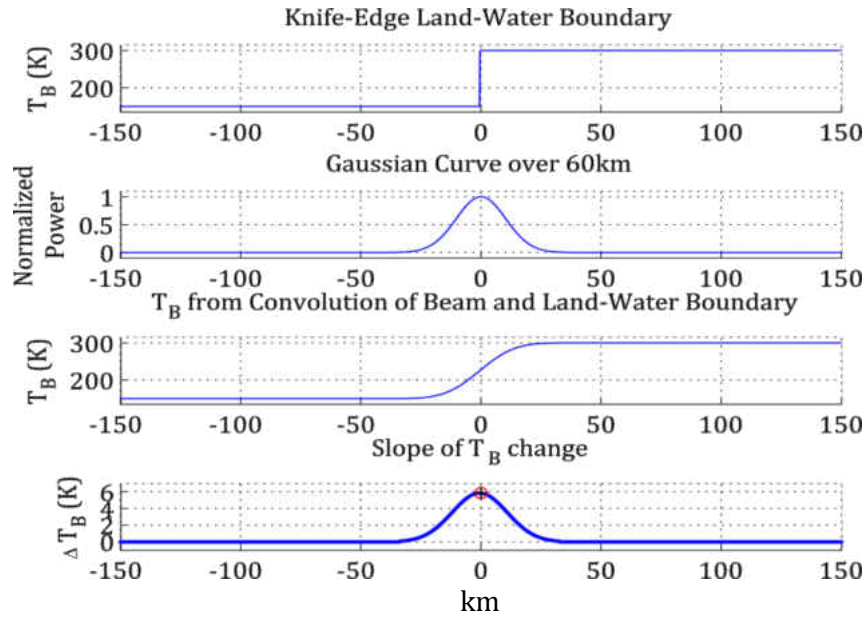
Figure 6: Convolution of a theoretical "knife-edge" land/water boundary with an ideal 1-Dimensional Gaussian antenna pattern from [8].

Following this, May [8] also evaluated several time series of data to find moments at which there were brightness temperature slopes of high magnitude, and investigated the corresponding locations to evaluate the similarity of the locations to ideal land/water crossings. Upon selecting a small number of these sites, she performed a fit procedure on orbits, which crossed over those locations, within a selected 14-week time period. An example of the slope plots for a 37GHz channel is shown in Figure 7.

Once these slopes were found, each was fit with a parabola; the vertex of each parabola was used to calculate a land/water crossing projection, in a manner similar to [6], though parabolic, rather than a cubic spline. Once each projection was geolocated, the along-

8

track distance from each parabola vertex to its nearest coast was calculated, resulting a

distribution of points similar to that shown in Figure 8.
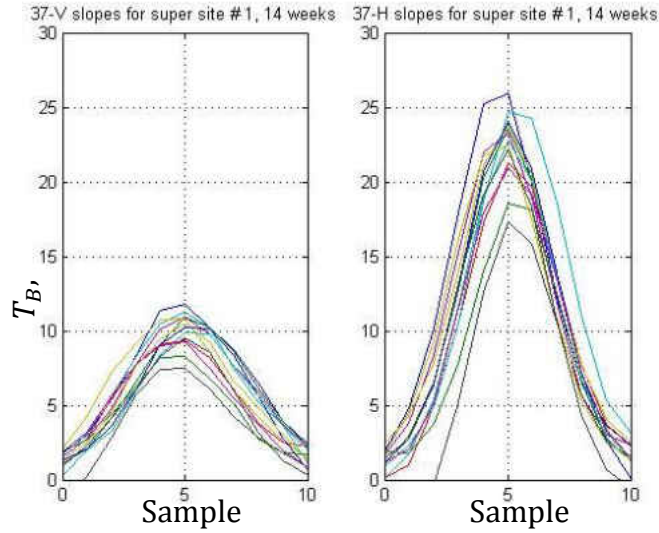
Figure 7: Initial slope plots for a single beam, vertical (left) and horizontal (right) polarizations from [8]
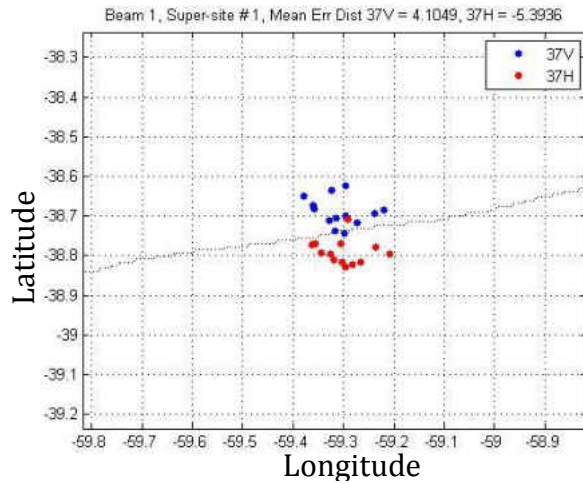
Figure 8: Maximum slope projections for a single site, corresponding to the slopes in **Figure 7** from [8].

9

1.1.5: Present Contribution and Research Objectives

The purpose of this thesis is to provide research that supports the post-launch AQ/SAC-D Calibration/Validation (CAL/VAL) activity, providing quantitative estimates of the antenna boresight pointing accuracy for the two MWR multi-beam antennas (24 individual beam geolocations).

To provide this quantitative assessment, an algorithm has been developed to quantify geolocation errors. The algorithm does so by comparing MWR observed land/water boundaries with a high-precision coastline land map. MWR observed boundaries are determined by the point of maximum brightness temperature slope, during the transition from a radiometrically cold scene to the radiometrically hot scene (and vice versa). As shown above, this method has been used in the past for WindSat and SSMI, and has been proven to be quite successful [4, 5].

WindSat, SSM/I, and SSMIS share the advantage of data density; because they each scan across-track at short intervals, construction of full images for an entire orbital cycle are possible, whereas they are not for any single MWR beam, and combining data from multiple beams would prevent identification of any single beam's biases. This makes attempts at the imaging of MWR brightness temperatures difficult, and direct use of the imaging technique impossible. An image of one week's worth of data in the vicinity of Florida illustrates this problem, shown in Figure 9. Because of this, Catherine May's modified approach was implemented, and a one-dimensional simulation was created to show that the numerical convolution of a Gaussian pattern resulted in a maximum slope when the beam was 50% filled.
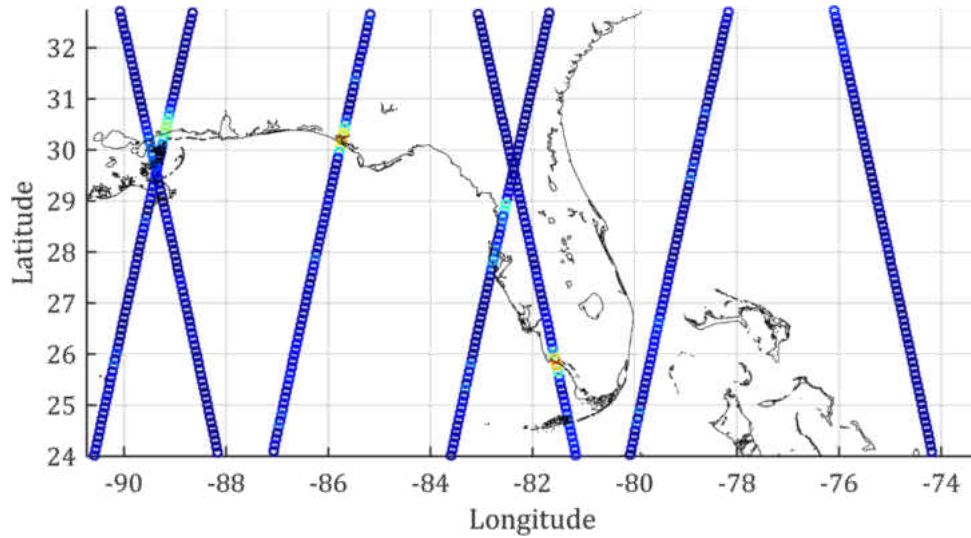
Figure 9: One week of 23.8 GHz MWR data over Florida, from beam 1.

As a further validation, the present work includes the two-dimensional convolution of an ideal Gaussian beam pattern with a simulated coastline. In this technique, incremental beam efficiency is numerically convolved with a two-dimensional brightness temperature map of a realistic coastal transition, to generate individual apparent brightness temperatures. The maximum-slope technique - which is the nucleus of the current work - is applied to these resulting temperatures to validate its efficacy in predicting coastal transitions.

Finally, the deliverable to CONAE will be a set of statistical results, which summarize the errors present in each MWR beam and polarization. To produce these results, the full three-year data record for each beam and polarization has been processed, which allow descriptive statistics to be produced. These results are intended to allow CONAE to take corrective action in future data processing to improve accuracy.

The organization of this thesis is as follows: the description of the MWR system and measurement geometry are presented in CHAPTER 2. In CHAPTER 3, the procedure for the

assessment of the observed antenna beam pointing (geolocation on the surface of the earth) is described. Simulations are presented that demonstrate the validity of using the rapid increase of the measured brightness temperature (T_B) at land/water transitions, to determine the beam geolocation (latitude/longitude). In CHAPTER 4, results are given for MWR antenna boresight pointing errors at selected land/water boundaries, which are analyzed separately by MWR channel (frequency/polarization combinations) and by individual beams. Finally, the summary and recommendations are presented in CHAPTER 5.

CHAPTER 2: AQUARIUS/SAC-D MISSION

The Aquarius/SAC-D mission was developed as a partnership between the United States (NASA) and Argentina (CONAE). The principal NASA contributions are the Aquarius salinity instrument, the Aquarius salinity data processing system, and the launch into orbit [1]. For its part, CONAE provided the satellite platform (SAC-D, shown in Figure 10), telecommunications control & command, science data acquisition and several instruments; one of these instruments is the Microwave Radiometer (MWR), which is the primary subject herein.

**Service Platform
& Science Instruments
(CONAE)**

**Aquarius Instrument
(NASA)**
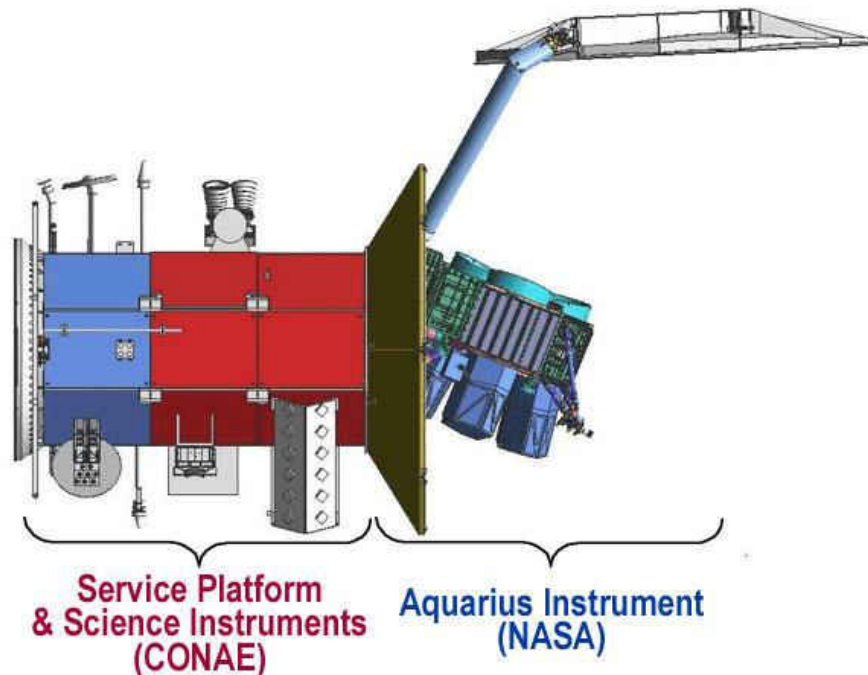
Figure 10: Satellite SAC-D overview.

The AQ/SAC-D satellite is positioned in a 98° sun-synchronous polar orbit at 657 km of altitude which crosses the equator northward (ascending) at 6 p.m., with the primary sensors (AQ and MWR) always facing away from the sun. The platform operates in a precise

seven-day repeating orbit to ensure parity between consecutive measurements of the same location.

The design of the AQ/SAC-D system is to provide global weekly (seven-day) SSS maps using a polar-orbiting space-borne, and an active/passive remote sensor. This seven-day observations are combined to produce global measurements of SSS on a monthly basis with an accuracy of 0.2 psu (practical salinity unit) at a spatial resolution of 150 km.

<p style="text-align:center"><u>**2.1: Aquarius Instrument**</u></p>

The Aquarius is an active/passive microwave remote sensor, which simultaneously measures ocean emitted brightness temperature (T_B) and radar backscatter at L-band [9]. The passive microwave radiometer operates at 1.4 GHz, with two Dicke receivers per feed to capture the linearly polarized ocean T_Bs. The active part is a single scatterometer (radar) that operates at 1.26 GHz, to capture the ocean normalized radar cross section.
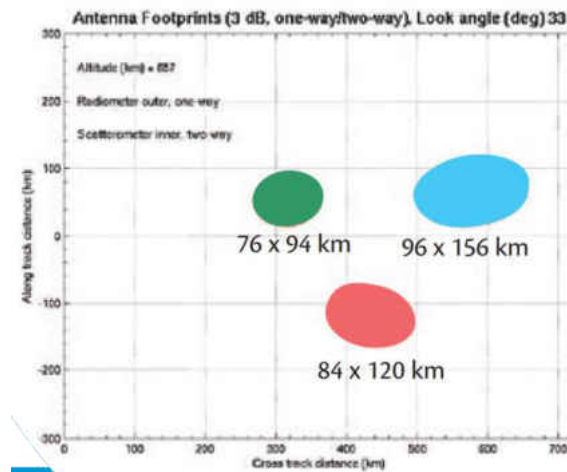
Figure 11: Footprint of Aquarius [1].

The antenna system uses a parabolic reflector producing three-beam measurements in a push-broom fashion. These beams point at incidence angles $29.3°$, $38.4°$ and $46.3°$ for the inner, middle and outer beams respectively [9], creating three instantaneous fields of view (IFOVs) with a resolution of 79x94 km for the inner beam, 84x120 km for the middle beam, and 96x156 km for the outer beam, resulting in a swath of approximately 390 km [9], as shown in Figure 11.

The Aquarius raw data is processed by the NASA Goddard Spaceflight Center, which provides the salinity data product to the science community, through NASA's Physical Oceanography Distributed Active Archive Center at the Jet Propulsion Laboratory.

2.2: Microwave Radiometer Instrument

2.2.1: MWR System and Measurement Geometry

The MWR is a three-channel, push-broom, Dicke radiometer, which is similar to the AQ L-band radiometer except in the operating frequencies and in the push-broom antenna configuration, as shown in Figure 12. Whereas the AQ provides 3 pushbroom beams (elliptical surface footprints in the cross-track plane), the MWR has two sets of 8 pushbroom beams that look both forward and aft of the cross-track plane.

The MWR antenna subsystem comprises two separate parabolic torus reflectors and a set of eight feed-horns, arranged in two rows [6] This configuration results in two pushbroom antennas, one dual polarized (horizontal and vertical polarizations) looking forward (K_a-band, 36.5 GHz) and one (horizontally polarized) looking aft (K-band, 23.8 GHz). By using separate frequency-scaled reflectors, the spot-beam antenna patterns for

individual feeds and frequencies are the same, which results in a common MWR instantaneous field of view (IFOV) of approximately 50 Km. The MWR antenna coverage is designed to match the AQ measurement swath (380 Km), formed by three AQ beams in the cross-track direction to the right hand side of the sub-satellite ground track. The geometry of the MWR/spacecraft ensures that eight of the horns are pointed forward of those of the Aquarius and that eight are pointed aft, covering the same 380km swath on the earth.

In this manner, the eight MWR beams exceed the Nyquist requirement for the Aquarius SSS spatial resolution of 150 Km. Also, as seen in Figure 12, the individual MWR beam IFOVs lie on two conical arcs, with odd beams (the closest to the satellite sub-track) at earth incidence angle (EIA) of $52°$ and even beams with EIA of $58°$.
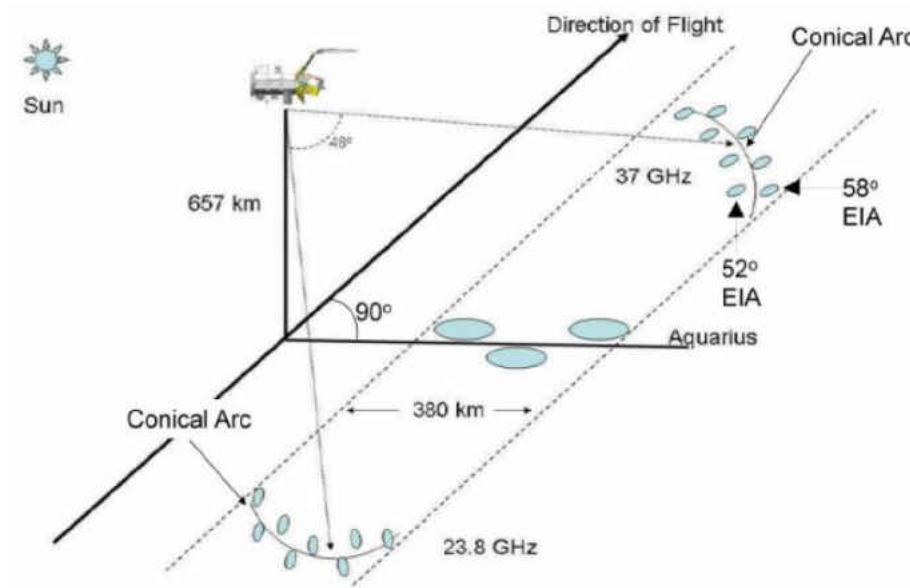
Figure 12: Measurement geometry for the AQ and MWR instruments.
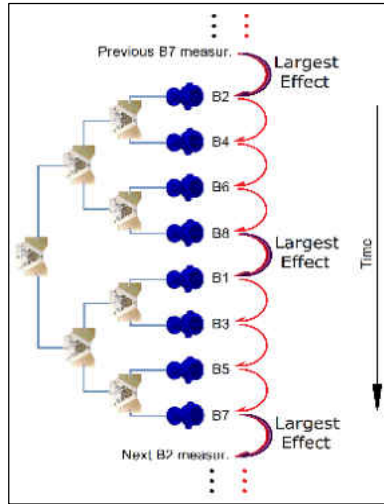
Figure 13: Antenna switch matrix temporal sampling of horns (right) for 23.8 GHz.

MWR provides three separate receivers and a 1x8 antenna switch matrix (as seen in Figure 13 for 23.8GHz) to sequentially sample each of the MWR spot beams. Each beam has an integration time of 240ms, thereby producing an overall cycle period of 1.92 s/MWR channel (as shown in the diagram in Figure 14: MWR Timing Diagram.), which results in a 13.1 km advancement along-track between the IFOVs of any beam's consecutive samples. Over the period of several minutes, the forward MWR measurements are collocated with the AQ IFOV's, which given the AQ IFOV size, is equivalent to being simultaneous; and the same is true for the aft MWR beams.
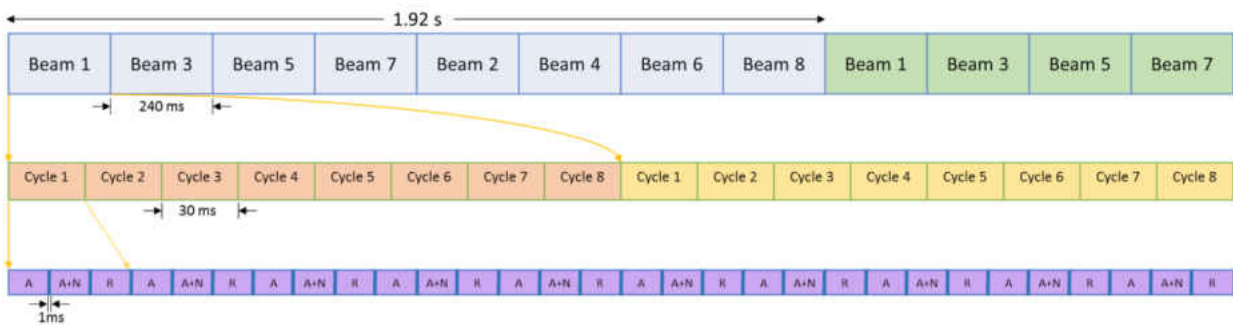
Figure 14: MWR Timing Diagram.

2.2.2: MWR Geolocation Calculation

The generation of the geodesic latitude and longitude (Earth location) for each MWR beam observation is provided by CONAE. This is based on a standard method of intersection of the line of sight of a sensor with the WGS-84 ellipsoid earth model, which is considered standard in current practices. This model, being an ellipsoid, does not take into account the elevations usually accounted for in Digital Elevation Models (DEMs), which are used in so-called orthorectification algorithms. As such, MWR Geolocation does not account for the elevation above sea level; moreover, the mean error due to altitude variations does not have a significant impact on overall geolocation performance, as it is most important over ocean.

The MWR calculations occur in post-processing at the CONAE ground facilities, during the production of the MWR L1B product [10], [11]. Given the definitive AQ/SAC-D orbit, the best estimate of satellite position, velocity and attitude are used in the line-of-sight intersection calculation that involves the following steps. First, a function is generated to calculate the spacecraft position/velocity for the time of the MWR observation (synchronized for all MWR channels). Next, based on the measured spacecraft attitude (roll, pitch and yaw orientation), the attitude quaternion is generated for the time of the observation. Following this, and based on pre-launch geometric calibration, the software uses measured alignment angles between the spacecraft coordinate system and the MWR beam boresight alignment angles (elevation and azimuth angles to produce the desired earth location relative to the satellite sub-point). Note that these angles were based upon pre-launch antenna pattern measurements (boresight calibration for the 3x8 MWR beams). Using all of this information, and coordinate transformations, the software calculates a line

18

of sight, pointing to earth, in an Earth-Centered Earth-Fixed (ECEF) coordinate system, for all MWR beams at their respective observation times. The final step is the intersection of this line of sight with the earth, giving the desired ECEF x, y, and z position of the measurement, and then geodesic latitude/longitude associated with it.

It should be noted that the assumed boresight angles for each K_a-band beam involved an analysis of separate measured antenna patterns for the vertical-polarization (V-pol) and horizontal-polarization (H-pol). Usually for a conical horn design, the boresight for the two polarization are co-aligned; however, since MWR boresight directions were determined independently from measured antenna beam patterns, they result in slightly different boresight angles (elevation and azimuth). The result is that V-pol and H-pol IFOV centers are systematically displaced, and whether this is true or an artifact of pattern measurement errors will be addressed in CHAPTER 4.

CHAPTER 3: GEOLOCATION ALGORITHM DESCRIPTION

3.1: Algorithm Overview

The initial period selected for this analysis was from October 29th 2012 to October 27th 2013. This period was based largely upon the necessity of the instrument reaching a stable radiometric calibration state. Freedom from short-term brightness temperature (T_B) radiometric calibration drift is crucial since the estimation of land/water boundaries is based upon a polynomial fit of the time series of T_Bs at the land/water crossing. The existence of spurious T_B data points within the fit range can vastly alter results, thereby ruining the parabolic approximation of a roughly Gaussian curve. Processing was performed on the L1B, Beta v3.0 version of MWR data, in MATLAB .mat format.

To effectively evaluate land-water crossings, certain sites are selected based upon a defined quality control (QC) criteria (see below), which captures results that frequently approximate ideal evaluation conditions. Once sites are selected for each beam and polarization, the data record is searched for locations at which MWR crosses each site. After such a time series is found, calculations are performed to project a point at which the land/water crossing is likely to have occurred (see Section 3.5:); this process is repeated for the entire data record under evaluation, and the results are aggregated with specific attention to disregarding outliers.

3.2: MWR Two-Dimensional Convolution

In the previous work, Catherine May [8] provided a 1D theoretical justification for the max-slope-point method of estimating the sensor derived land/water crossings. Recall that from above, this maximum slope of brightness temperatures corresponded to be the point at which the IFOV is 50% filled by a land, and 50% by water, which is the definition of the land/water boundary. In this thesis, we extend this theoretical analysis to a 2D convolution of an ideal Gaussian beam, which is a realistic approximation of the actual MWR brightness temperature measurement for each beam position, as described below.

Ulaby, Moore, and Fung [12] provide a continuous integral, which represents the convolution of a 2D scene brightness temperature, T_{ap}, with the observing antenna's power radiation pattern, F_n:

$$T_A = \frac{\iint_{4\pi} T_{ap}(\theta, \varphi) * F_n(\theta, \varphi) * \sin\theta \, d\theta \, d\varphi}{\iint_{4\pi} F_n(\theta, \varphi) * \sin\theta \, d\theta \, d\varphi} \tag{1}$$

In equation (1) a spherical coordinate system is assumed, where the antenna boresight lies on the is the +Z axis, θ represents the antenna pattern elevation angle, φ represents the pattern azimuth angle, and the result, T_A, is the output brightness temperature of the antenna.

For the current contribution, a modified version of this continuous integral has been implemented as a discrete convolution.

$$T_A = N/D \tag{2}$$

21

Consider, first, the numerator of equation:

$$N = \iint_{4\pi} T_{ap}(\theta, \varphi) * F_n(\theta, \varphi) * \sin \theta \, d\theta \, d\varphi \tag{3}$$

$$= 2\pi \int_0^{\theta_{main\ beam}} T_{ap}(\theta) * F_n(\theta) * \sin(\theta) \, d\theta \tag{4}$$

$$= \int_0^{\theta_{main\ beam}} T_{ap}(\theta) * F_n(\theta) * \sin(\theta) \, d\theta \tag{5}$$

$$\approx T_{ap}(\theta_1) * F_n(\theta_1) * \int_0^{\theta_1} \sin(\theta_1) \, d\theta + T_{ap}(\theta_2) * F_n(\theta_2)$$
$$* \int_{\theta_1}^{\theta_2} \sin(\theta_2) \, d\theta + \cdots + T_{ap}(\theta_{mb}) * F_n(\theta_{mb})$$
$$* \int_{mb-1}^{mb} \sin(\theta_{mb}) \, d\theta \tag{6}$$

$$\approx \sum_{i=1}^{main\ beam} T_{ap}(\theta_i) F_n(\theta_i) [-\cos\theta]_{\theta_{i-1}}^{\theta_i}, \tag{7}$$

where θ_i is the differential elevation angle, away from boresight, under consideration; θ_{mb} is the main-beam extent, which is assumed $= 2.5*(HPBW/2)$. This equation gives an approximation of the numerator of equation (1) in small, discrete steps over the main beam of a circularly-symmetric radiation pattern – in this case a Gaussian.

Considering next the denominator of equation (1), and performing a similar approximation:

$$D = \int_0^{\pi} F_n(\theta, \varphi) * \sin \theta \, d\theta \, d\varphi \tag{8}$$

$$\approx \sum_{i=1}^{N} F_n(\theta_i)[-cos\theta]_{\theta_{i-1}}^{\theta_i}, \tag{9}$$

Note here that the summation is to N, where N represents the incremental angle, which corresponds to π. Combining the numerator and denominator of this discrete expression, the result is:

$$\frac{\sum_{i=1}^{main\ beam} T_{ap}(\theta_i)F_n(\theta_i)[-cos\theta]_{\theta_{i-1}}^{\theta_i}}{\sum_{i=1}^{N} F_n(\theta_i)[-cos\theta]_{\theta_{i-1}}^{\theta_i}} \tag{10}$$

Equation (10) explains that the incremental radiation pattern of a circularly-symmetric antenna may be approximated by the sum of a series of incremental products between the pattern, and the apparent brightness temperature scene.

An illustration of such a convolution is shown in Figure 15. The top panel shows a time series of IFOVs as they progress from water (on the left side) to land (on the right). The water is radiometrically cool ‑ in this case modeled as 180 Kelvin ‑ and the land is modeled as a hot 300 Kelvin. The progression in color of the elliptical IFOVs in the top panel represents that each differential ellipse within the IFOV ‑ the θ_is in (10) - is receiving more radiometric energy as the IFOV transitions from ocean to land. The bottom panel shows the numerical results of this series of convolutions. The blue curve represents the set of raw convolutions of an ideal 2-D Gaussian beam with this brightness temperature scene, while the red curve represents the differences of these temperatures. Note that for ease of viewing, the red curve has been shifted and scaled; actual peak values are approximately 51.5 Kelvin.
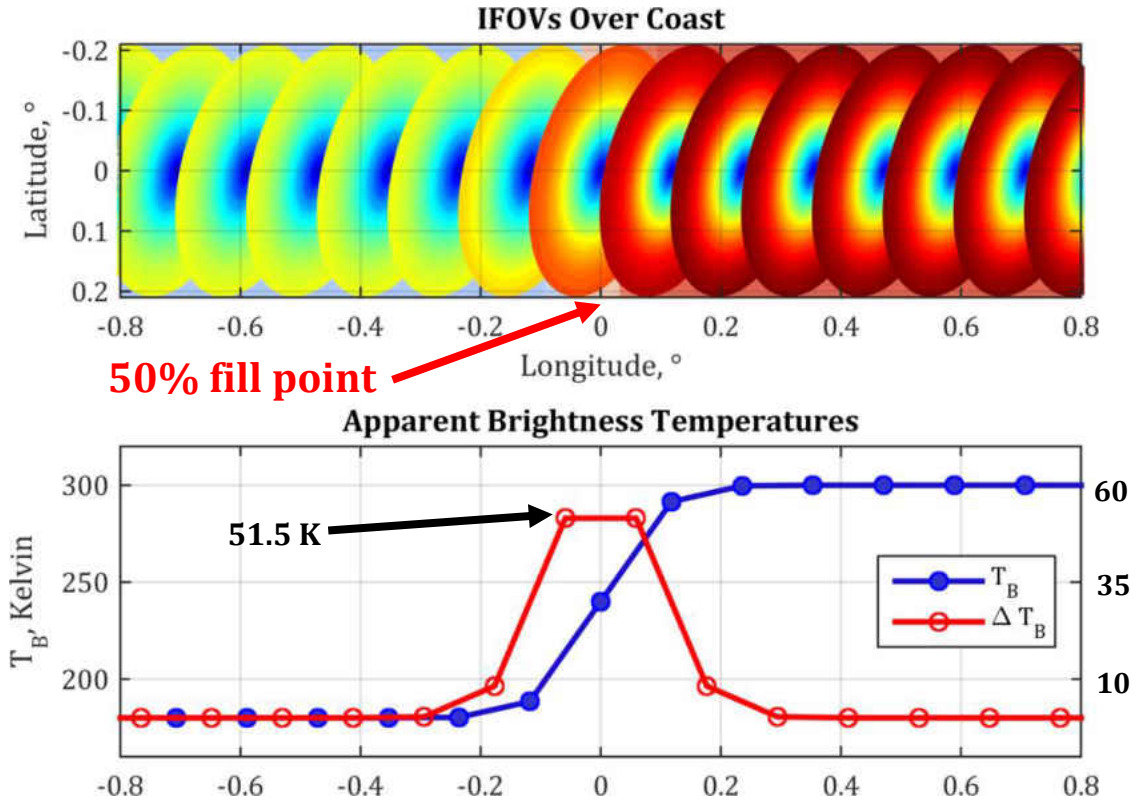
Figure 15: Top panel: antenna IFOVs progressing from radiometrically cool ocean (left) to hot land (right). Bottom panel: brightness temperature (blue) and slope (deltas) of brightness temperature (red).

As is visible from this example, the highest slopes of T_B occur in the vicinity of land/water crossings, for radiometric instruments with 2D circularly symmetric antenna patterns, when observing coasts with an idealized "step–function" transition. This serves as a motivation for finding locations on the earth that exhibit properties similar to the ideal step-function coast, which will be shown in 3.3:.

3.3: Site Selection

As it was mentioned in Section 1.1.5:, Figure 6 illustrates a simplified, 1-D step-function coast, transitioning from ocean temperatures on the left, to land temperatures on the right. This simple model shows that the point of maximum T_B slope is collocated with the

true coast. Its accuracy is due to the transitional nature of the coast: an ideally flat, unit step change from ocean T_B to land T_B. Clearly this does not exist in nature, however the closer a land/water crossing is to this situation, the more accurate the crossing location estimation will be.

Since rapid transition that approximates this step-function represents a good coastal interface (in an accuracy-of-estimation sense), searching for high T_B slope within a time series naturally suggests itself as a likely method of finding ideal analysis locations.

3.3.1: Supersite Definition

Upon finding points in T_B time series data, which exhibit strong slope features, it is necessary to further investigate the locations at which these features occur. Using visible imagery from Google Earth, each potential site is examined (quality controlled) to eliminate the following cases:

1. the area must NOT contain any significant area % of man-made structures; roads, bridges, buildings, etc. This is important because these structures have different radiometric properties from those of natural land.

2. the area of land must NOT contain free water is the form of: rivers, lakes, bays, channels, swamps or other. Only land free from significant sources of water is acceptable, but surface vegetation of any type is not a concern.

3. the coastline of interest should be relatively straight for a length significantly longer than the MWR IFOV. The reason for this is that the 50% beam-fill is difficult to define for

complex shapes and also the location of successive passes over the same area will occur in slightly different locations due to changes in the satellite ground track.

4. Finally, there is one other consideration, which cannot be mitigated by quality control and this is the occurrence of weather. Rain over oceans can significantly increase the ocean T_B at random locations, which could produce significantly distortion of the T_B slope over a land/water crossing. The only effective means is to consider rain as a transient error source that can produce outliers that can be removed by statistical means.

Thus the motivation for investigating areas of land/water crossing heuristically using Google Earth images results in areas that meet the above criteria are referred to as "supersites."

Madagascar poses an example of one such site in Figure 16. Note the straight, vegetative coast in the southeast, and the relative size of the IFOV in comparison. This location exhibits all of the characteristics of an excellent supersite. It is important to note, however, that supersites differ by beam and polarization. This is because of the significant variation in crossing angle and location for each beam; since a large number of sites are needed for evaluation, in the next section a procedure for rapidly identifying sites is shown.

3.4: Procedure for Finding Supersites

A novel contribution of this thesis is an automated method of finding these sites. The procedure starts by classifying a given orbital dataset according to the longitude of its ascending node; since there are 103 orbits in a single cycle, there are naturally 103 bins for orbital datasets. After all orbits in the period of interest are classified, all the datasets are aligned in a 2-D matrix, such that the ascending nodes occur in the same column; then, the

mean of the resulting matrix is taken along its columns, giving a single representative time series of orbits that fall into this ascending node bin.
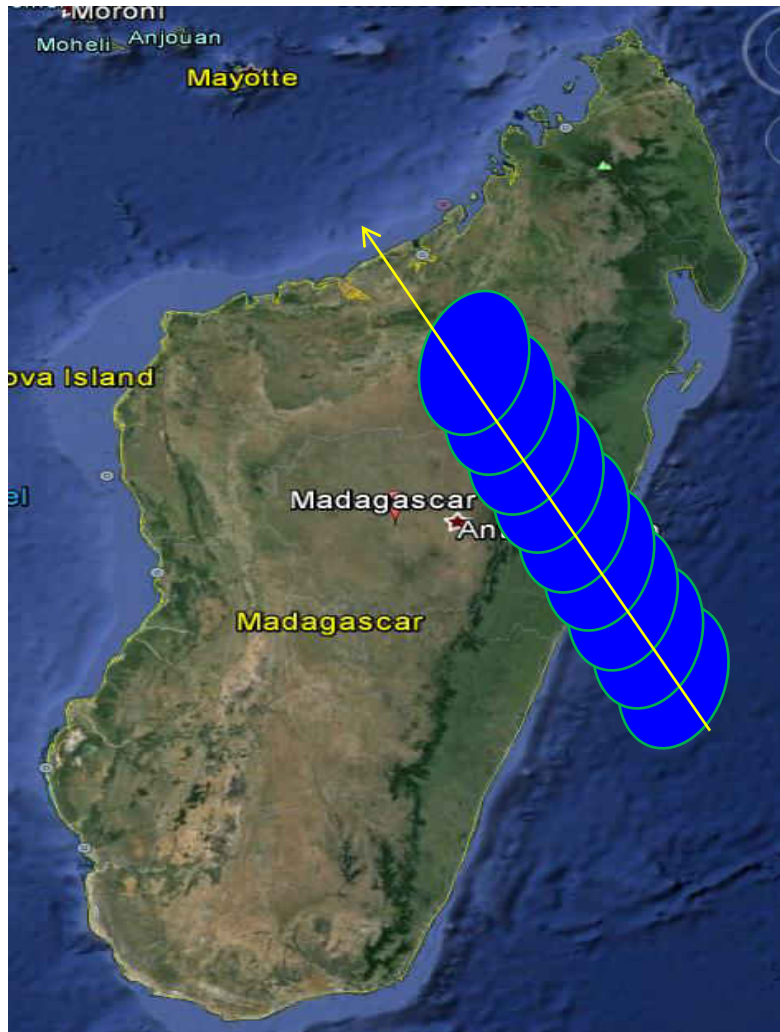
Figure 16: IFOVs crossing Madagascar on an ascending pass. [13].

After orbit classification, the next step is to calculate the numerical difference of the resulting averaged time series, with the aim of finding locations with T_B slopes that exceed a desired threshold. These candidate sites are bounded between $\pm 60°$ latitude, to ensure avoidance of polar ice. Finally, through Google Earth's kml interface, the sites that meet

threshold criteria for qualitative investigation are displayed, and the sites that exhibit ideal qualities are stored for later evaluation. A flow chart of this process is exhibited in Figure 17.

Examination of the averaged time series – representing all orbits falling into an ascending node bin – is done by simultaneously observing the symmetry and maximum slope point of the ΔT_B plot as shown in Figure 18, and the linked nadir view of the corresponding Google Earth image, shown in Figure 19. Symmetry of the slopes around the maximum slope point has been found to enhance the accuracy of curve fitting, and correspondingly, the calculation of coastal crossing.
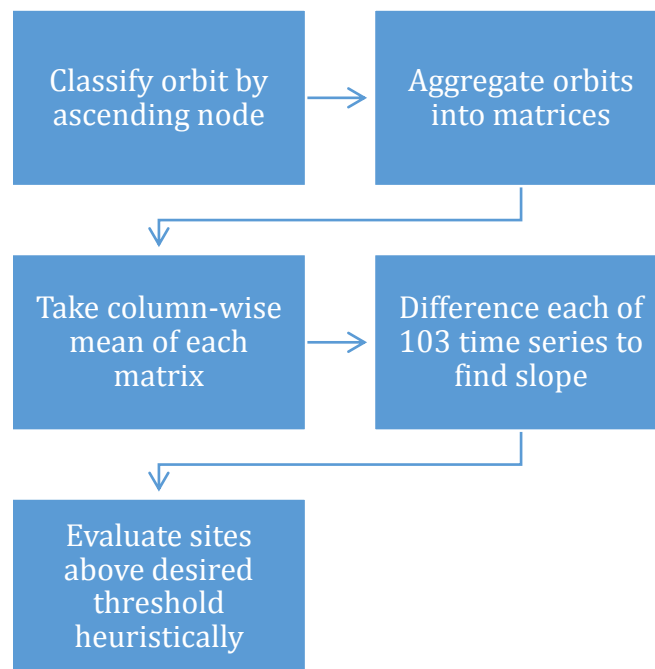
Figure 17: Supersite-evaluation flow chart.
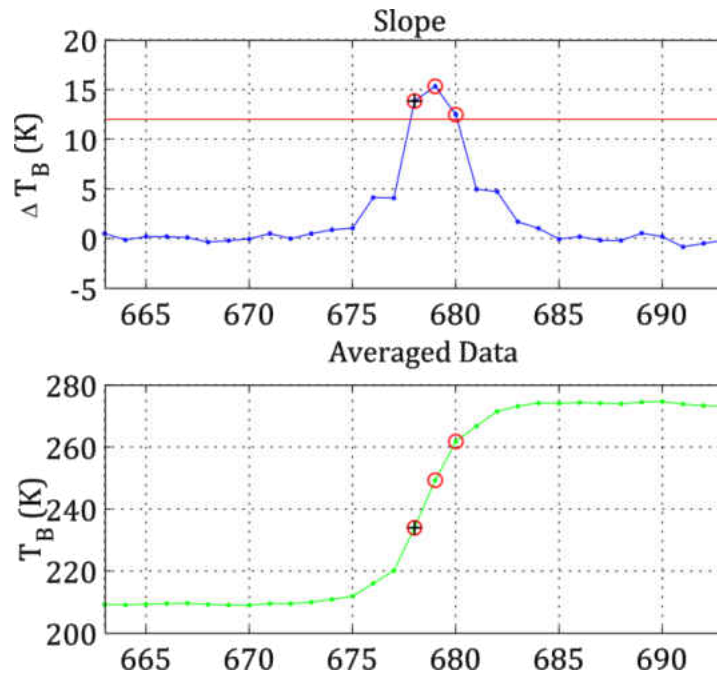
Figure 18: Inspection of averaged data for site fitness determination.
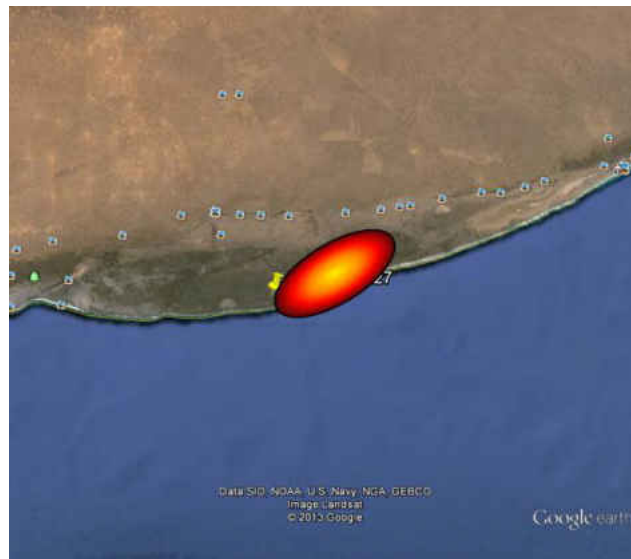
Figure 19: Google Earth image of ideal site and 3dB IFOV.

3.4.1: Supersite Statistics

The location of supersites depends upon the MWR beam ground-tracks and the intersection of favorable coastlines. There is no set criterion for the number of supersites required; however, since statistics derived from the analysis of these land/water crossings

29

at supersites, then more is better. The statistical results are clearly dependent upon the number of passes over these points, it is necessary to view the individual land/water passes as-well-as the collection of passes of such sites per beam and polarization. Table 1 summarizes this information and a global supersite map combining all MWR beams is given in Figure 20. Overall there appears to be an adequate number of supersites to allow a subdivision of statistics to examine the mean and standard deviations of collocation errors e.g., comparing collocation errors separately for ascending versus descending passes.
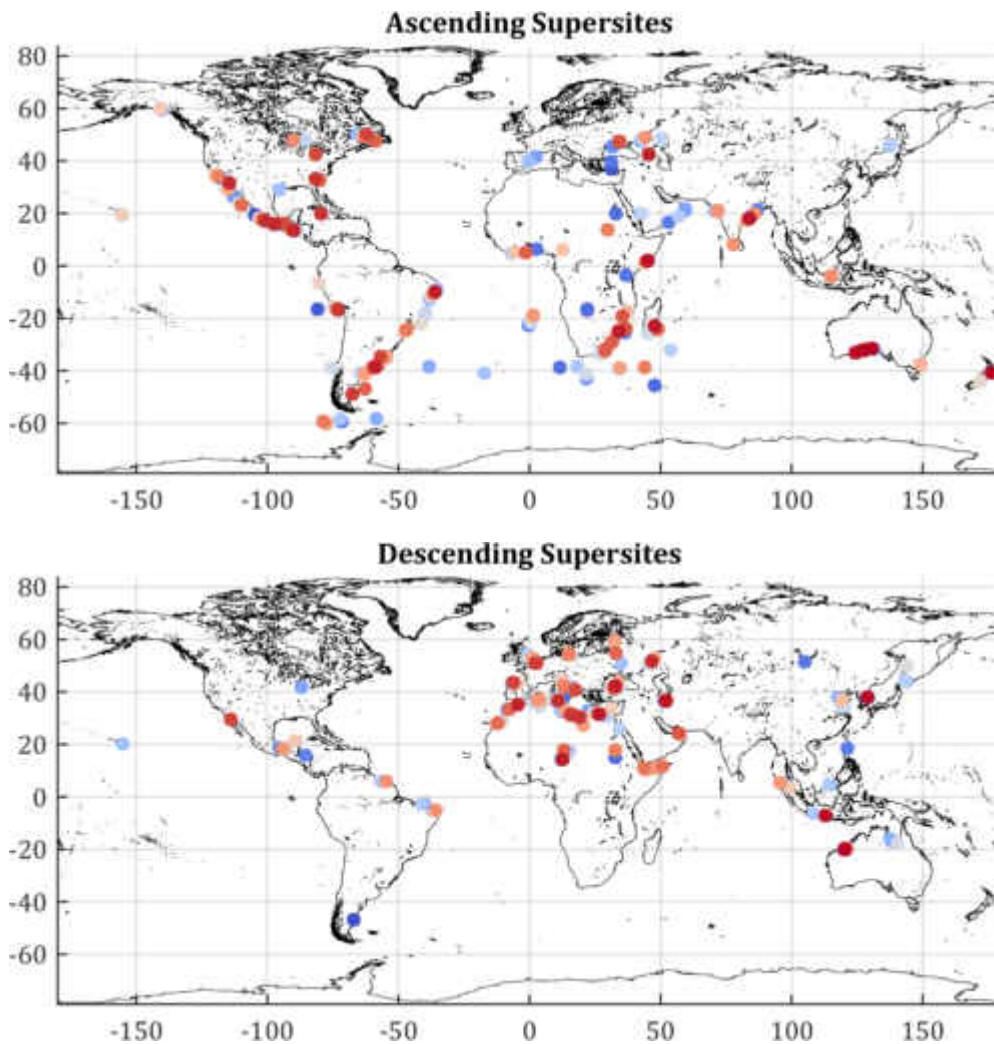
Figure 20: Ascending and descending supersites, all beams

Table 1 – Number of Supersites per Beam and Polarization

	36.5V	36.5H	23.8H
Beam 1			
Ascending Supersites	13	25	10
Descending Supersites	8	19	5
Total Supersites	21	44	15
Beam 2			
Ascending Supersites	9	21	8
Descending Supersites	7	8	4
Total Supersites	16	29	12
Beam 3			
Ascending Supersites	20	26	14
Descending Supersites	14	11	9
Total Supersites	34	37	23
Beam 4			
Ascending Supersites	16	20	10
Descending Supersites	7	14	6
Total Supersites	23	34	16
Beam 5			
Ascending Supersites	15	20	13
Descending Supersites	8	14	7
Total Supersites	23	34	20
Beam 6			
Ascending Supersites	15	22	9
Descending Supersites	13	21	7
Total Supersites	28	43	16
Beam 7			
Ascending Supersites	17	19	7
Descending Supersites	16	14	2
Total Supersites	33	33	9
Beam 8			
Ascending Supersites	9	16	10
Descending Supersites	9	8	7
Total Supersites	18	24	17
Global Totals			
Total Ascending	114	169	81
Total Descending	82	109	47
TOTAL	196	278	128

The 36.5 GHz channel's horizontal polarization accumulated the largest number of supersites through the search procedure, while the 23.8 GHz channel accumulated the least. This could be a result of ΔT_B threshold values that need further refinement; the threshold is shown as a red horizontal line in the top panel of Figure 18, and is a manually-set parameter, determined prior to performing the supersite search procedure.

For nearly all polarizations and beams, a significantly larger number of supersites were found for ascending revolutions than those of descending revolutions. No algorithmic feature explains this bias: points under consideration are aligned by ascending node, and bounded to encompass an equal portion of ascending and descending time within each revolution.

The 23.8 GHz channel shows far fewer supersites than either of the 36.5 GHz channel's polarizations. Examinations of T_B distributions around land/water crossings for the three beams could potentially offer some insight, as it can be seen Figure 21, and Figure 22. In particular, Figure 21 shows that the 23.8 GHz channel has a larger fraction of intermediate data points – those between clearly delineated ocean and land – than either of the 36.5 GHz channels Figure 22 and Figure 23. It is possible that this is an indication of poorly defined boundaries for this channel, though is out of the scope of this research and therefore this is a conclusion left to CONAE to make.
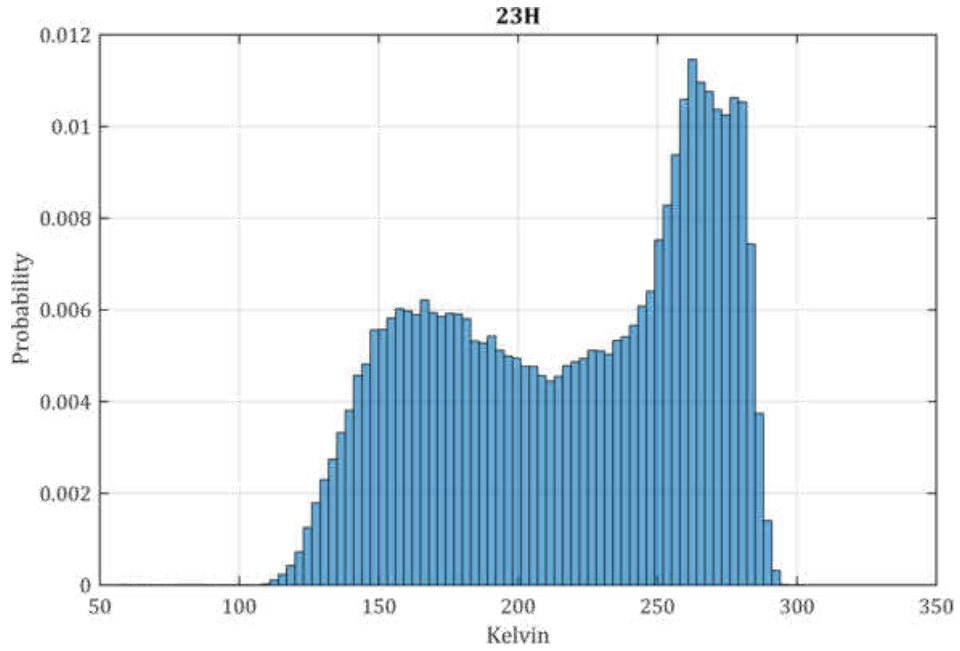
Figure 21: Brightness temperature distributions near land/water crossings, for 23.8 GHz.
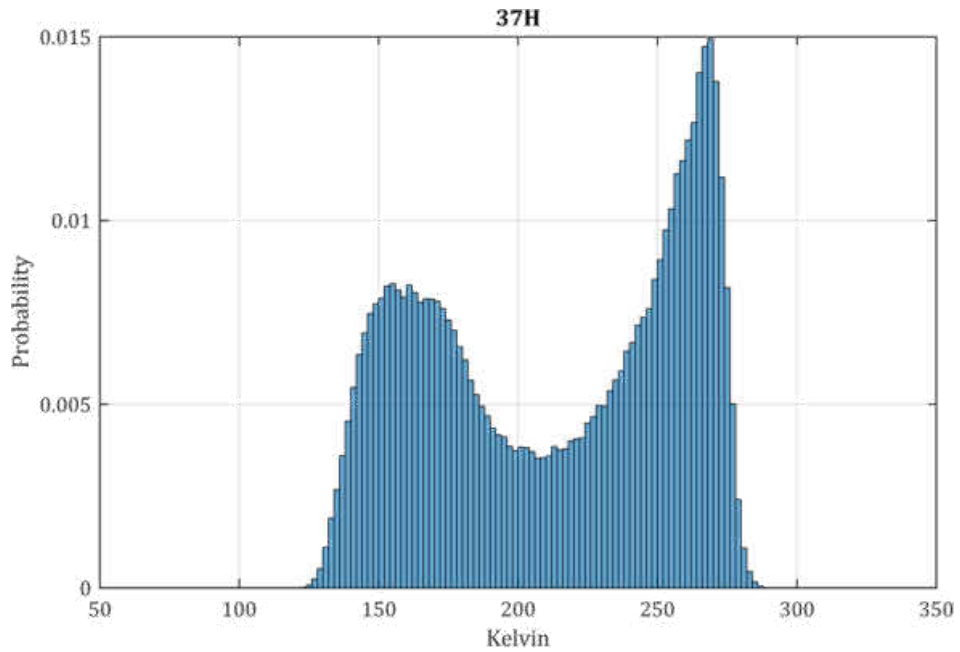
Figure 22: Brightness temperature distributions near land/water crossings, for 36.5 GHz Horizontal Polarization.

33
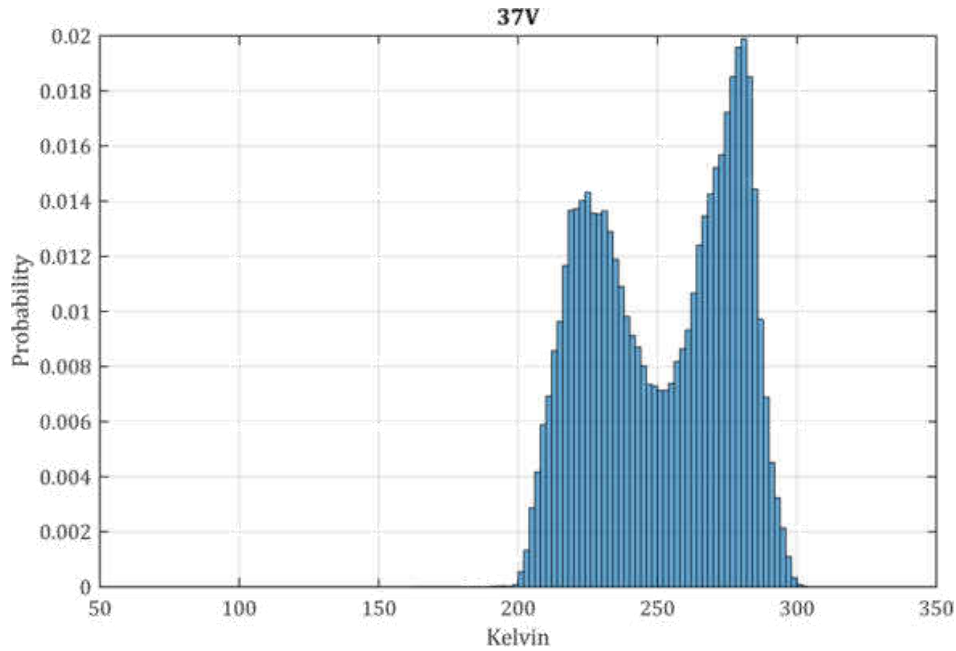
Figure 23: Brightness temperature distributions near land/water crossings, for 36.5 GHz Vertical Polarization.

3.5: Land/Water Crossing Projection

To calculate the instrument-observed coast, the previously determined supersites are compared with time series T_B data to determine if a particular data record contains points, which pass near the center of any designated supersites. If such points are found, their T_Bs are then differenced to determine the rate of change between each sample. Those differences are then fit to a parabola, and the parabola's vertex is used to determine a projected latitude and longitude of where the IFOV likely crossed. This projected point is then compared to the nearby high-resolution coastline to determine the coastal point nearest to it. The distance from the coastal point to the projected point is the error recorded for that supersite crossing. These results are then aggregated into a master data structure which contains their geolocation information, as well as several details about the coastal crossing of each. This

34

process is illustrated pictorially in Figure 24: Algorithm for calculating instrument-observed coast which presents the algorithm for instrument-observed coast calculation.
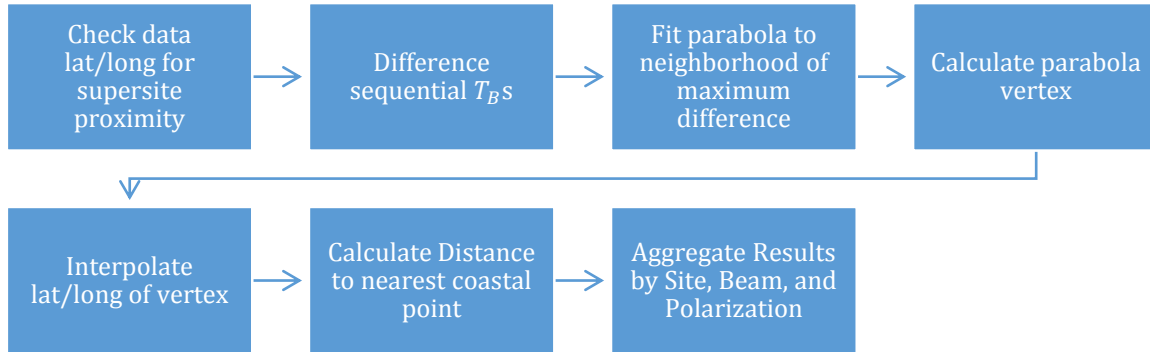
Figure 24: Algorithm for calculating instrument-observed coast.

3.5.1: Step 1: Check Data Latitude and Longitude for Supersite Proximity

Supersites are stored in a data structure which is organized by beam and polarization, as well as ascending and descending status; the MWR data record is organized similarly, so when a file is loaded, the combinations of beam and polarization are traversed correspondingly in the supersite record and the file. A window is constructed 100 km in each direction from a supersite, and translated to lat/long; points in the data record are then compared to this window, as well as a check to ensure that an ascending match is found near an ascending supersite, and vice versa. If at least seven points are found near an appropriate supersite, calculations for that crossing commence.

3.5.2: Step 2: Difference Sequential T_Bs

Data points within each window are differenced to find the numerical derivative of brightness temperatures in the neighborhood of a supersite; since these points are slopes between points, they are each assigned a lat/long that is midway between those of the

35

samples used to generate them. This process is illustrated in Figure 25. In this figure, green arrows indicate later points, and yellow arrows indicate earlier points which are subtracted from them to generate differences.
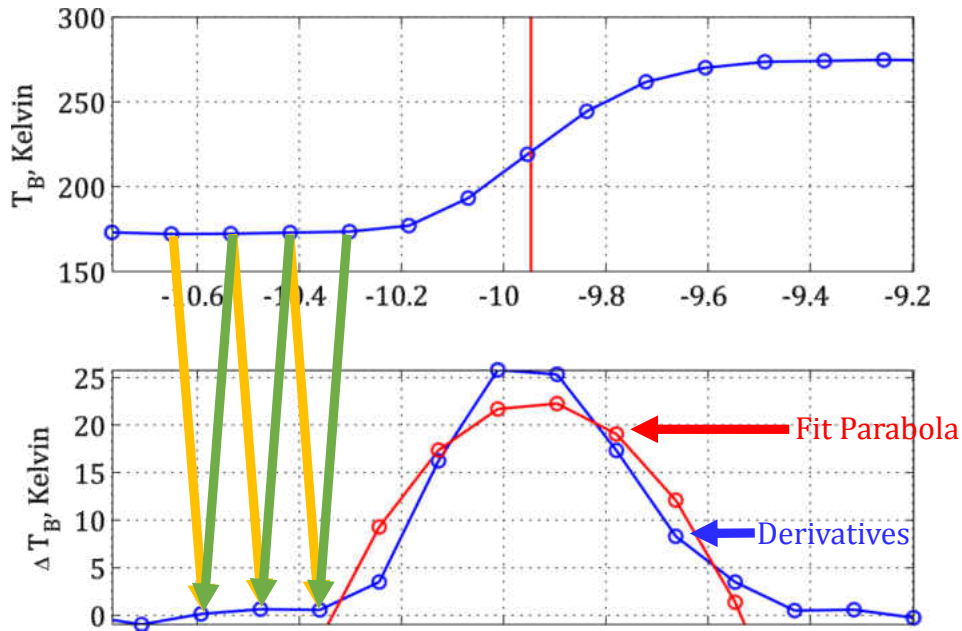
Figure 25: Differencing of sequential points to generate slopes between brightness temperatures.

3.5.3: Step 3: Fit Parabola for Neighborhood of Maximum Difference

Upon generating a time series of brightness temperature differences, those ΔT_Bs are fit with a parabola. This is done with MATLAB's polyfit command, which uses a standard Vandermonde matrix to perform the fitting, and results in a set of three coefficients A, B, and C, representing the quadratic, linear, and constant terms of the parabola, P:

$$P = Ax^2 + Bx + C, \tag{11}$$

36

where x is simply the sample number in the current window, as shown in Figure 26.

3.5.4: Step 4: Calculate Parabola Vertex

The vertex of the fit parabola calculated in the previous step is also calculated in a well-known manner, as:

$$x_{Vertex} = -\frac{B}{2A.}.$$ (12)

This x – again in unitless sample number – is rarely an integer; note the position of the red triangle which represents it in Figure 26. This triangle is placed on the red parabola, which has been fit to the green time series of brightness temperature slopes; the slopes are generated from the blue brightness temperatures, and placed halfway between them in sample position.

Thus, the conceptual position of the vertex of a parabola, which has been fit to brightness temperature numerical differences, lies somewhere between samples and must be interpolated.

3.5.5: Step 5: Interpolate latitude and longitude of vertex

The interpolation is straightforward: it is treated as a linear weight, which is applied to the difference between the coordinates of the samples, which surround it. In the case shown in Figure 26, the x which corresponds to the sample number of the fit parabola's vertex is 4.37; this is interpreted as 37% of the distance between the coordinates of point 4 and point 5, and is geolocated there.
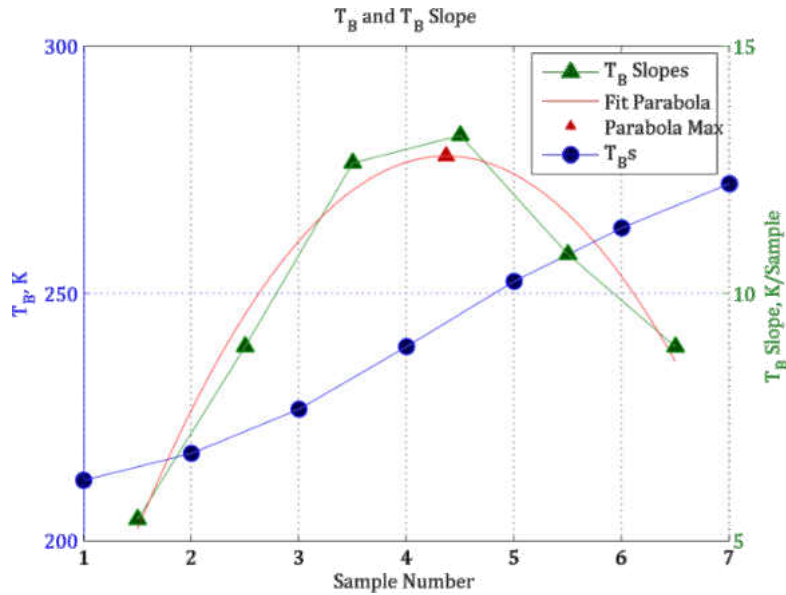
Figure 26: T$_B$s and slopes, with fit parabola and its maximum at sample 4.37.

3.5.6: Step 6: Calculate Distance to Nearest Coastal Point

The mispointing error present in each assessment is simply the vector norm of the distance between the projected point, and the nearest coastal point, combined with a logical assessment of whether the point leads or lags the actual crossing; if the point follows the crossing, it is deemed to be a positive error, and if it precedes the crossing, it is deemed a negative error. Utilizing this, the error may be quantified as

$$error = norm\left(\begin{bmatrix} projected_{lat} - coast_{lat} \\ projected_{lon} - coast_{lon} \end{bmatrix}\right) * error\ sign \qquad (13)$$

where the *norm* is the familiar "Pythagorean" ℓ_2 vector norm, which gives the magnitude of the error vector. This is illustrated in Figure 27 where the ocean is pictured at the top of the figure, land at the bottom, a dotted black coast separating them, a black arrow representing

ground-track direction, an open circle representing the projected land/water crossing point, and finally, a blue double-ended arrow illustrating the distance between projected radiometric slope maximum, and the nearest coastal point. It is worth observing that the error distance is not constrained to be parallel to ground track; this is a result of the coastal point nearest to the projection being used.
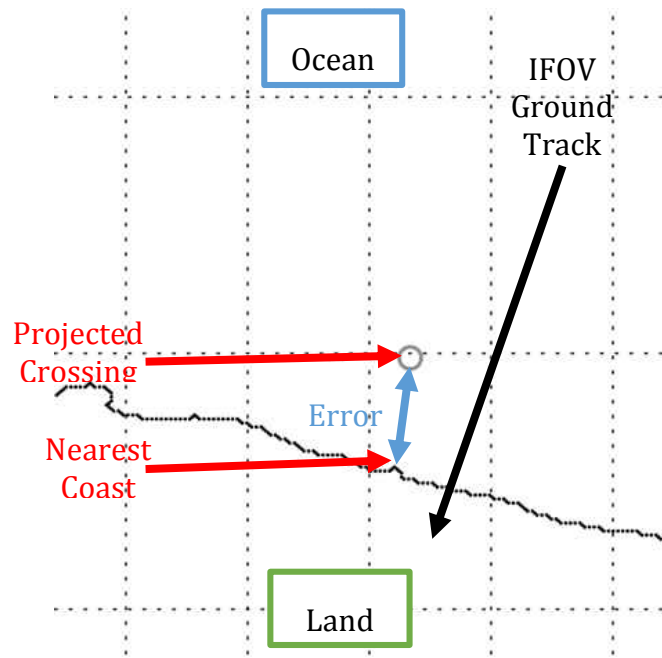
Figure 27: Calculation of mispointing error.

Note that the conversion between lat/long and earth-projected distance has been omitted from equation (13) for clarity; the calculations are simple, and treated as linear within the small regions under consideration. The justification for this lies in the constant nature of latitude calculations, and the linearity of longitude calculations in a region. For example, within six IFOVs, the instrument's ground track traverses 5 * 13.1 km, or 65.5 km;

since latitude calculations are constant, this amount will always be equal to a fixed number

of latitude degrees, i.e.:

$$Deg_{Lat} = \frac{Dist_{km}}{2\pi * Radius_{Earth}} * 360°. \tag{14}$$

In the 65.5 km case, this is equal to .58°. The calculation of longitude at this distance

is simply the product of the above number and the cosine of the latitude. Obviously the cosine

function is most sensitive to changes in the neighborhood of 90°; since the area of

investigation is bounded by $\pm 60°$ latitude, that angle is representative of the worse-case

scenario of angle sensitivity. Below, Figure 28 illustrates that in this scenario, less than 0.3

km of error can be experienced in geolocation, far less than both the size of the IFOV, and the

gap between consecutive IFOVs.
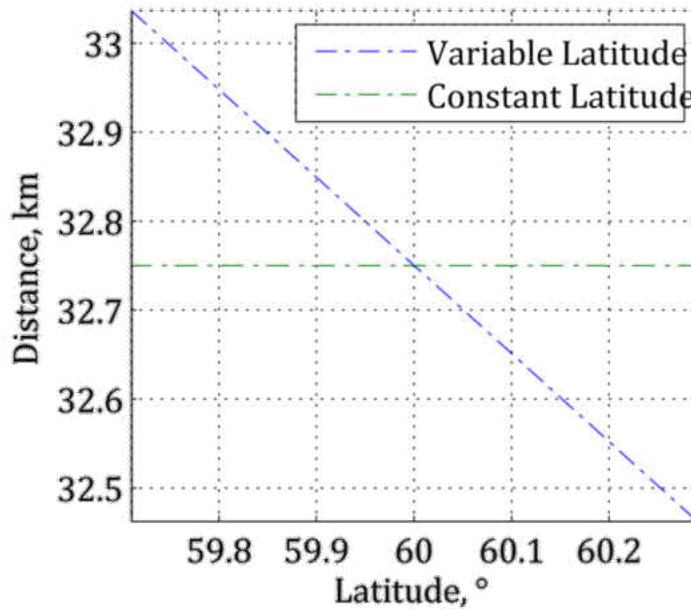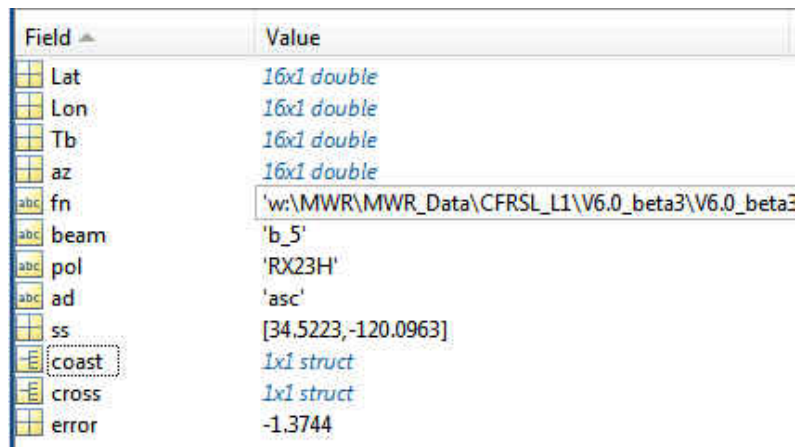
Figure 28: Geolocation error in IFOV calculation window.

3.5.7: Step 7: Aggregate Results by Site, Beam, and Polarization

The preceding steps are performed for each supersite, and each data file. Upon their completion, the procedure returns a data structure which consists of several results, depicted in Figure 29. The primary results are the signed error in kilometers, and its accompanying identification information: beam, polarization, ascending or descending flag, and projected crossing point. To ease any later analysis which might need to be done, several other data are captured: latitude, longitude, T_B, azimuth angle for each point in the vicinity of the supersite, as well as the coordinates of the local coastline, the coordinates of the designated supersite, and the name of the original data file from which each result was derived. The results and identification are then easily parsed to generate statistics, and can also be used to reconstruct the crossing, in case of any potential error, or for ease of peer review.

Field ▲	Value
Lat	16x1 double
Lon	16x1 double
Tb	16x1 double
az	16x1 double
fn	'w:\MWR\MWR_Data\CFRSL_L1\V6.0_beta3\V6.0_beta3
beam	'b_5'
pol	'RX23H'
ad	'asc'
ss	[34.5223,-120.0963]
coast	1x1 struct
cross	1x1 struct
error	-1.3744

Figure 29: Sample data structure of a single land/water crossing; the completeness of this result enables reconstruction of the conditions which created it quickly and easily.

3.6: <u>**Error Correction**</u>

Initial error assessments were performed with ideal simulations, as referenced in Figure 6. In such simulations, where a coastline is modeled as a perfect step function, the maximum slope location is found to exist exactly where the -3dB beam footprint is half-filled by water and land. However in real measurements, radiometric aberrations, such as inland water and human development, distort T_B measurements in ways, which are difficult to define; this is the reason for choosing straight, well-defined coastlines for site evaluation. After ideal sites were found, statistics were generated from the mispointing results, giving an overall view of the errors present in each beam. These results are presented in CHAPTER 4.

CHAPTER 4: RESULTS AND VALIDATION

4.1: Coastal Boundaries

Analysis of the results indicates largely consistent mis-pointing for each beam, with a deviation between the vertical and horizontal polarizations of several kilometers for the 36.5 GHz channels. An example of this mispointing error for the Beam 1 - 36.5GHz, V- and H-Pol is shown Figure 30 below. In this figure, red points indicate the coastline location of the projected points in vertical polarization, while blue points represent the ones in horizontal polarization. From the simulation of the 2D Gaussian antenna pattern convolution with the WindSat observed T_Bs is believed that the sensor derived land/water boundary should be independent of the polarization used (see Appendix-C). Thus, since these comparisons are calculated from data provided by the same MWR horn, they should lie atop one another. This disparity is probably attributed to the fact that CONAE uses different boresight angles for each polarization. This is a finding of our analysis, which need to be addressed by CONAE during future MWR reprocessing.

4.2: Results Analysis

Within each radiometer channel, the processed data exhibit central trends stemming from the clustering, which is illustrated in Figure 31, Figure 32 and Figure 33, for 23.8 GHz, 36.5 GHz H-pol and 36.5 GHz V-pol, respectively. In particular, the left panel of Figure 32 the ascending runs – those in which the instrument crosses the equator from south to north – are shown. Note that a normal distribution fits these data bins reasonably well, however the descending results are shown in the right panel, and no simple distribution seems to fits these well. Determination of the source(s) of these differences is out of the scope of this

thesis and therefore it is left to CONAE. The remainder of these error distribution figures by beam and polarization are shown in APPENDIX A: FULL HISTOGRAMS BY BEAM, and depicted in stem-and-leaf plot in Figure 34.
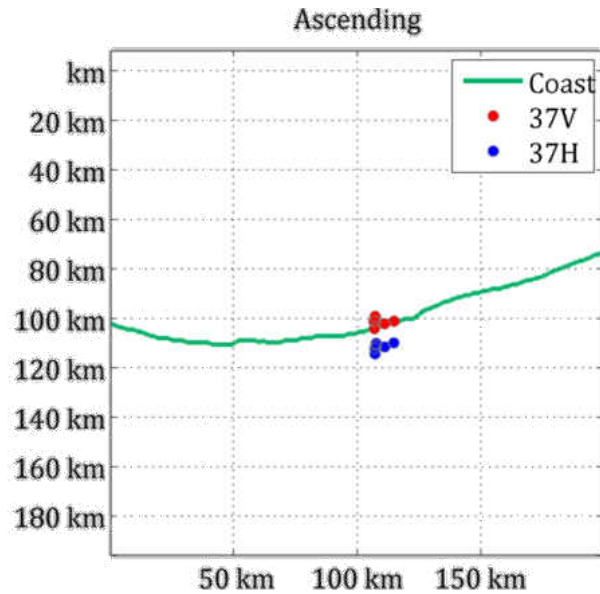
Figure 30: Errors in land-water crossing for beam one over a site in southern Australia. The flight direction in this case is north, which is oriented to the top of the figure.

As with any observations, outliers occurred in the collection brightness temperatures, and calculation of maximum crossings and error distances. These outliers were removed with a simple procedure. First, the data from the inner 60% of all values were taken; to these, a normal distribution was fit; then, based upon the derived normal distribution, a z-score was derived for each value. Measurements were deemed outliers if their z-score was three or higher; that is, if they equal to or more than three standard deviations away from the mean. As a point of clarity, note that in this case, the "measurement" is the mispointing error.

44
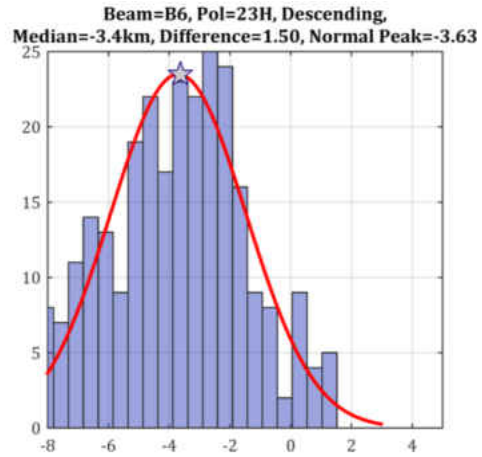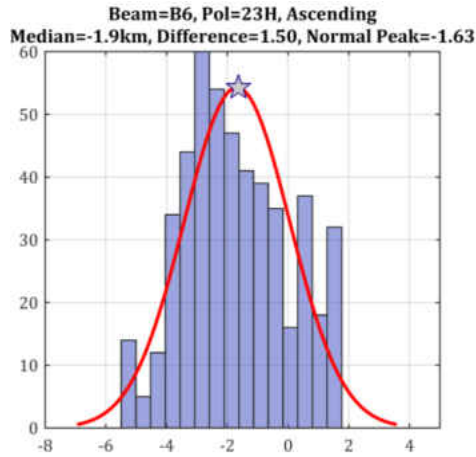
Figure 31: Error distribution for Beam 7 of the 23.8 GHz channel, horizontal polarization, ascending (left panel) and descending (right panel).
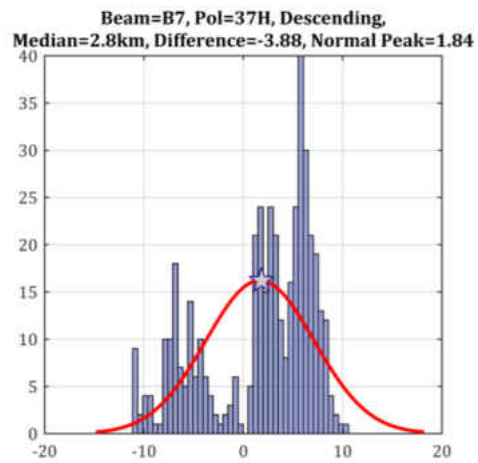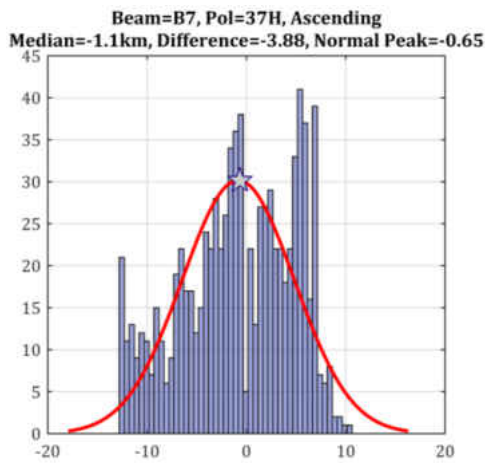
Figure 32: Error distribution for Beam 6 of the 36.5 GHz channel, horizontal polarization, ascending (left panel) and descending (right panel).
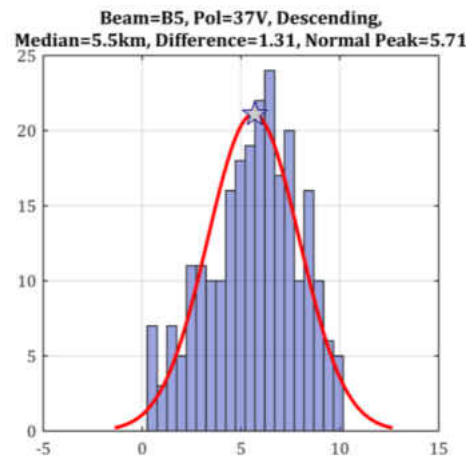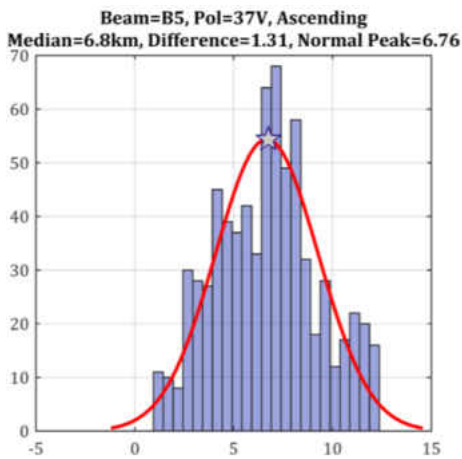
Figure 33: Error distribution for Beam 5 of the 36.5 GHz channel, vertical polarization, ascending (left panel) and descending (right panel).

In the top panel of Figure 34, the 23.8 GHz channel's mispointing error is depicted beam-by-beam. This is the most "well-behaved" of all channels, tending to have a negative error for all channels, with high variation only on beams one and four. The panel below shows the error and variation in the 36.5 GHz channel's vertical polarization. Wide variations in error from beam to beam are visible, and variation larger than that of the 23.8 GHz channel is evident on average. The bottom panel displays these same statistics for the 36.5 GHz channel's horizontal polarization.

As discussed above (section 4.1), there are coastline location differences between the vertical and horizontal polarizations of the 36.5 GHz channel, and Figure 35, Figure 36 and Figure 37 highlight these. The top panel overlays the two polarizations, and the bottom panel depicts the difference. From this figure it is evident all beams and pols have different collocation mean errors. Since the boresight angles are derived separately by beam and polarization, this result is not surprising, and it indicates that CONAE should derive a consistent set of boresight angles that causes V- and H-pols to overlap and that removes the mean mispoint angle by beam number.

Additionally, a side-by-side comparison of the polarizations is available in Table 2 – Error for Each Beam, where each beam exhibits its own individual mispointing error. Table 2 also shows a summary of mispointing error statistics, by beam and polarization, for the sample period.
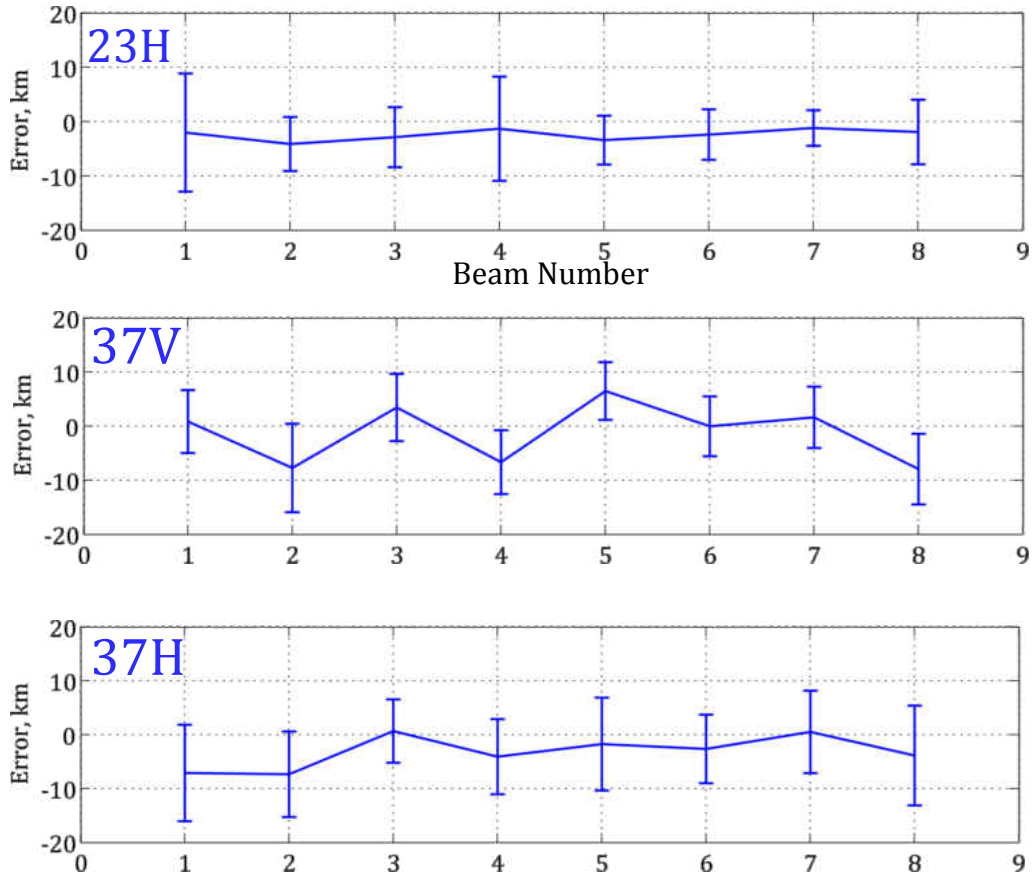
Figure 34: Errors and deviations for each beam and polarization.
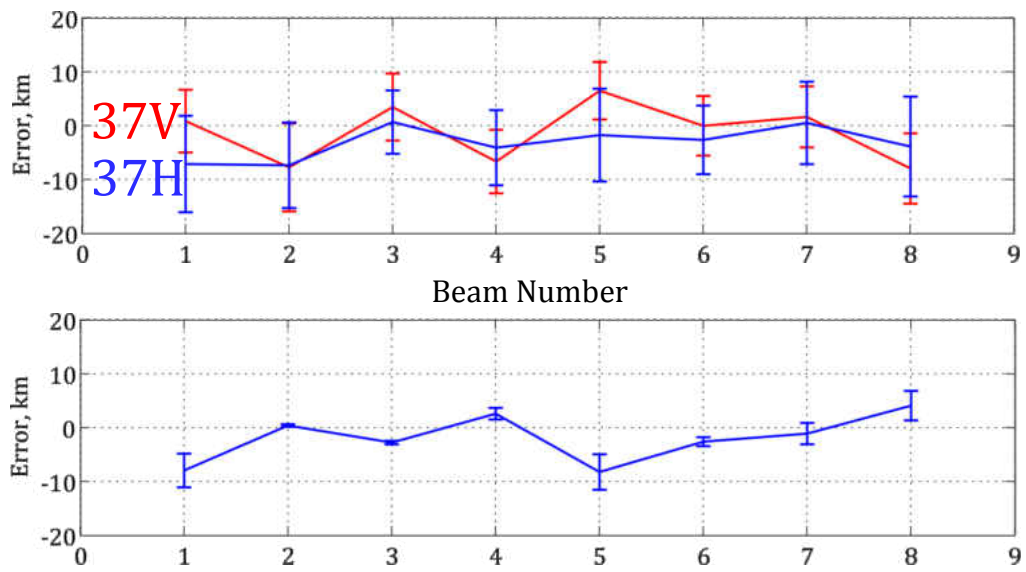
Figure 35: Comparison of the two polarizations the 36.5 GHz channel (top) and computed difference between these polarizations (bottom).

Local Coast -17.388°lat 37.681°lon
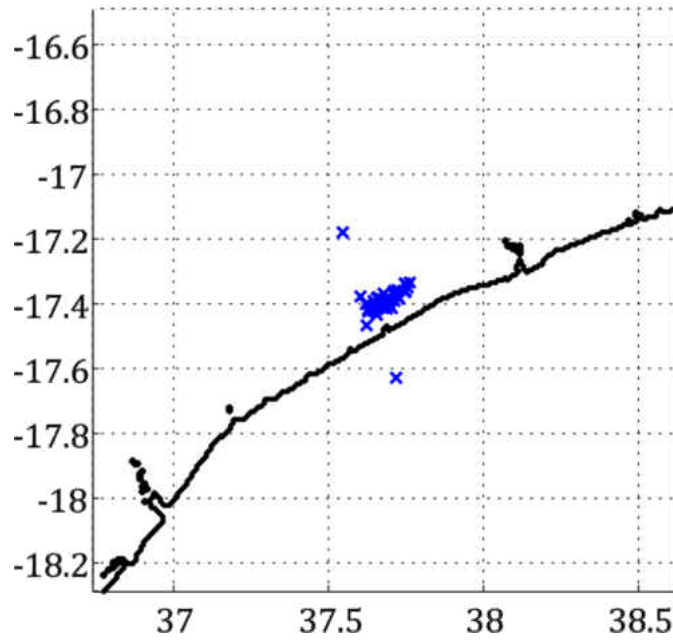
Figure 36: Estimated land/water crossings of 36.5GHz Beam 5, vertical polarization.

Local Coast -17.504°lat 37.661°lon
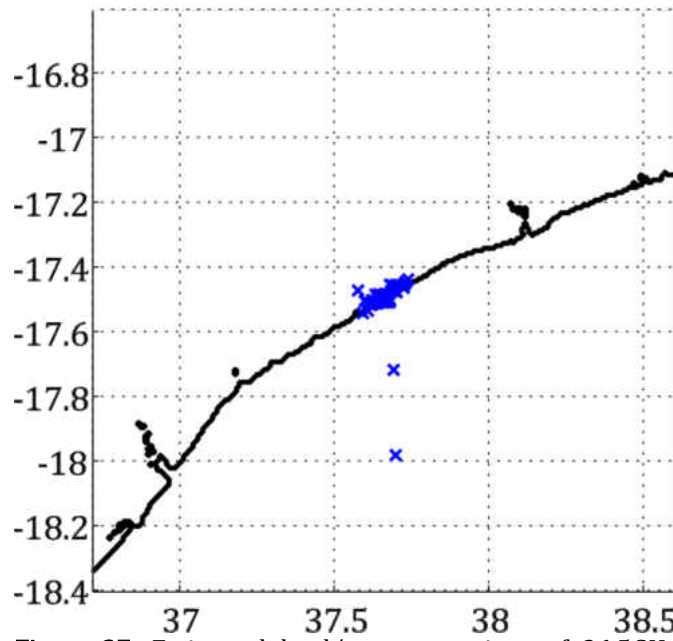
Figure 37: Estimated land/water crossings of 36.5GHz Beam 5, horizontal polarization.

48

Table 2 – Error for Each Beam

	36.5V	36.5H	23.8H
Beam 1			
Median Error, km	M = 1.98	M = 4.70	M = 1.62
Error Std. Dev, km	σ = 4.31	σ = 4.52	σ = 3.72
Number of Measurements	n = 699	n = 1193	n = 511
Beam 2			
Median Error, km	M = 6.82	M = 5.33	M = −0.15
Error Std. Dev, km	σ = 2.06	σ = 5.49	σ = 5.55
Number of Measurements	n = 483	n = 802	n = 441
Beam 3			
Median Error, km	M = 4.27	M = 3.08	M = −1.10
Error Std. Dev, km	σ = 3.75	σ = 2.46	σ = 3.58
Number of Measurements	n = 1100	n = 1020	n = 700
Beam 4			
Median Error, km	M = 2.47	M = 3.25	M = −0.94
Error Std. Dev, km	σ = 1.93	σ = 4.19	σ = 2.26
Number of Measurements	n = 722	n = 989	n = 536
Beam 5			
Median Error, km	M = 1.29	M = 2.78	M = 1.62
Error Std. Dev, km	σ = 2.48	σ = 3.23	σ = 5.65
Number of Measurements	n = 790	n = 1076	n = 712
Beam 6			
Median Error, km	M = 2.70	M = 1.97	M = −.30
Error Std. Dev, km	σ = 2.09	σ = 4.25	σ = 6.34
Number of Measurements	n = 745	n = 1196	n = 561
Beam 7			
Median Error, km	M = 5.29	M = 4.68	M = 1.05
Error Std. Dev, km	σ = 3.95	σ = 4.61	σ = 1.91
Number of Measurements	n = 920	n = 1032	n = 316
Beam 8			
Median Error, km	M = 3.84	M = 4.85	M = .87
Error Std. Dev, km	σ = 4.15	σ = 2.47	σ = 3.96
Number of Measurements	n = 486	n = 763	n = 522

4.3: Ascending and Descending Characteristics

The coastline collocation data indicate difference in ascending and descending revs, and certainly satellite attitude errors could be different for ascending and descending. However, similar beam point analysis for the AQ sensor beams has not indicated this issue. Therefore the cause is more likely the error associated with this thesis approach. Taking the difference of ascending and descending means, a nonzero disparity is evident, as shown in Table 3. Figure 31, Figure 32, and Figure 33 give some insight as to the numerical source of these differences: there are clear differences in ascending and descending error distributions for several beams and polarizations. The cause of this disparity remains unknown, however the data suggest that the 36.5 GHz, horizontal polarization suffers from this problem more than the other channels. One possible explanation for this variation is seasonal; since MWR passes ascending nodes at around 5:00 a.m. local time (close to sunrise) it is possible that seasonal affects have contributed to this change.

Table 3 –Ascending/Descending Difference (km)

	36.5V	36.5H	23.8H
Beam 1	-0.65	-2.99	1.49
Beam 2	7.92	-4.98	4.23
Beam 3	3.91	2.96	0.57
Beam 4	0.85	-1.69	-1.54
Beam 5	1.31	1.11	1.54
Beam 6	3.95	3.69	1.50
Beam 7	-0.44	-3.88	0.43
Beam 8	2.45	5.24	3.76

Another explanation may lie in Figure 38. This figure shows an aggregation of the 36.5 GHz channel's vertical polarization; each row is one orbit, and is separated from adjacent rows by one week; each column is a single sample. Rows are aligned by ascending node. The first 36 weeks share similar features most of the time, but the nature of the data changed at week 37, and from then on showed much greater variation from orbit to orbit.
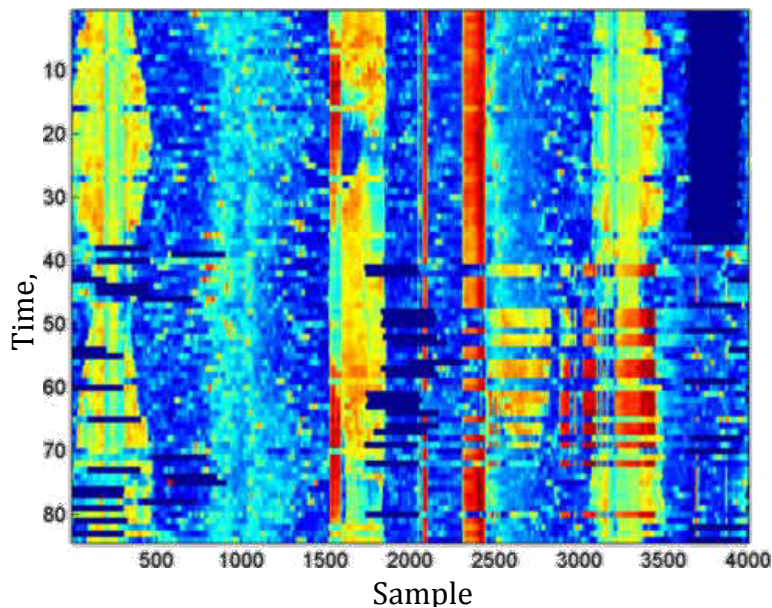
Figure 38: Vertically stacked time series of a single orbit. Note apparent, unexplained change around week 37.

CHAPTER 5: CONCLUSIONS AND FUTURE WORK

From the analysis conducted, it appears that the MWR instrument does exhibit some mispointing error. Referring back to Table 2 – Error for Each Beam, some corrective quantifications are suggested. These numbers have yet to be applied to the MWR data record and re-processed. Currently, the MWR instrument is in a non-functioning state on an indefinite basis; indications from the Aquarius team suggest that this is unlikely to change. Thus the correction must be applied to the existing data record from launch until MWR was powered off on 2014/08/24, and because of this there is no opportunity to see how the correction values impact new data.

In the present work, an algorithm has been created which allows the calculation of MWR geolocation error from the time series of brightness temperature data; this error is expressed in kilometers, which is expected to admit correction angles after further work from CONAE is performed. Furthermore, a simulated two-dimensional convolution of an ideal Gaussian antenna pattern has been implemented to validate the approach; this technique concurs with past work, and with theory.

To implement the designed algorithm on the existing three-year data record of MWR, a software package in MATLAB has been produced, and refined: initial efforts took several days on a high-powered server to process a year of data, but the in its current state ‑ shown in APPENDIX B: MATLAB CODE ‑ the package of MATLAB functions and scripts together is able process the full three-year data record from CONAE in three hours, on a simple dual-core PC. The original 22,000 lines of code have been reduced to just over 2,000 lines of

legible, functional code which may be handed off to any external group, to process their T_B with minimal alteration.

The beginning processing goal was to extract metrics from one full year of MWR data, representing one full seasonal cycle; this goal has been met, and extended to the entire three-year data record.

The deliverable to CONAE is the record of geolocation error distances, which are intended to allow that organization to continue investigation of the sources existing geolocation error; this goal has been met, in the form of a single HDF5 file.

In summary, all of the existing goals for this research have been met, and the resulting code, documentation, and data have been delivered to CONAE.

5.1: **Future Work**

Though the MWR mission is stalled, the technique herein may be applied to any T_B data. Because of this, any refinements in accuracy, as well as analytical and empirical results may enhance the analysis of beam-pointing in the future.

There are two specific points, which are recommended for further investigation:

1. Monte Carlo simulations should be performed to ascertain the effect of coast shape, land/water T_B difference, and most importantly, actual coastline location versus distance from nearest point on calculated crossings.

2. Secondly, the use of a sigmoid curve should be investigated to find the maximum slope point. Heuristic evaluations indicate that a parabola can only be fit well to five

53

points of MWR data, which risks placing too high an emphasis on any one data point; for well-chosen sites, data suggest that a sigmoid or hyperbolic tangent function would permit the use of many more data points, and possibly improve robustness of this technique.

3. Investigate the relative alignment of the IFOV ellipse major axis relative to the coastline. When the major axis of the ellipse is approximately perpendicular, there seems to be more consistent collocation errors with smaller standard deviations.

4. Develop an error model to characterize collocation error estimates uncertainties associated with the algorithmic assumptions and MWR T_B error characteristics.

5. Perform an analysis to establish the covariance matrix between collocation errors for the various beams and pols.

APPENDIX A: FULL HISTOGRAMS BY BEAM

In the present appendix, histograms are presented which characterize the differences found between coastal boundaries, and computed maximum T_B slope. Histograms are separated by beam, polarization, and whether they are ascending or descending. In each case, an attempt was made to fit a normal curve to the histogram (red line). The location of the mean of this curve is marked (blue star), and the title information contains both the median of the results (for outlier robustness), and the peak of the normal curve.
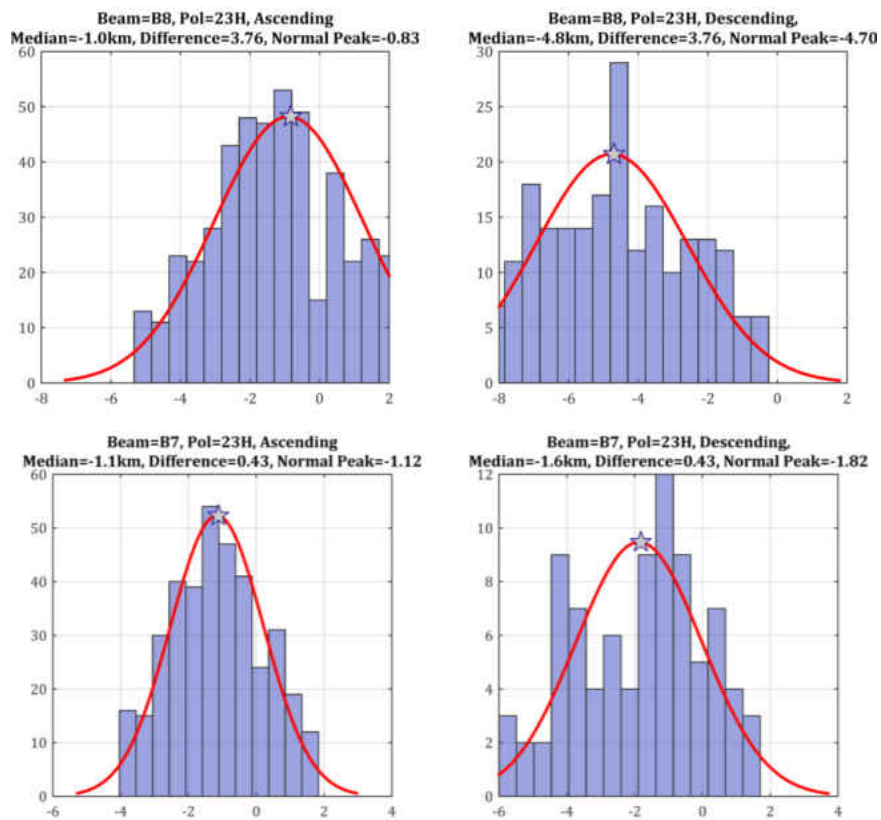
Beam=B6, Pol=23H, Ascending
Median=-1.9km, Difference=1.50, Normal Peak=-1.63
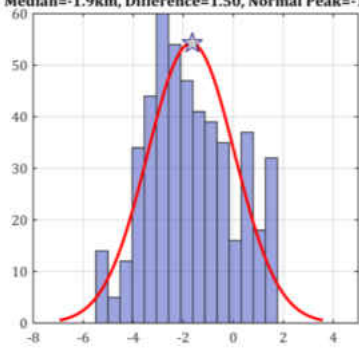
Beam=B6, Pol=23H, Descending,
Median=-3.4km, Difference=1.50, Normal Peak=-3.63
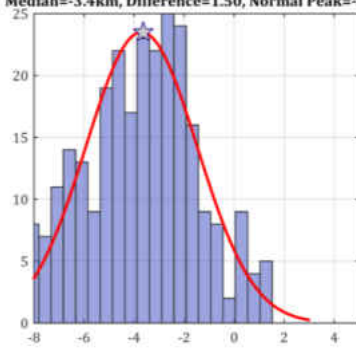
Beam=B5, Pol=23H, Ascending
Median=-3.1km, Difference=1.54, Normal Peak=-3.09
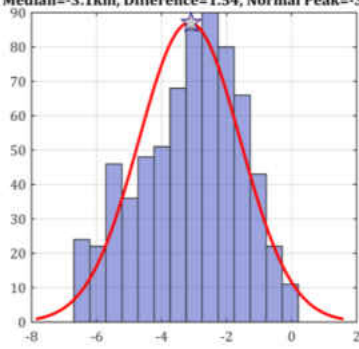
Beam=B5, Pol=23H, Descending,
Median=-4.6km, Difference=1.54, Normal Peak=-4.58
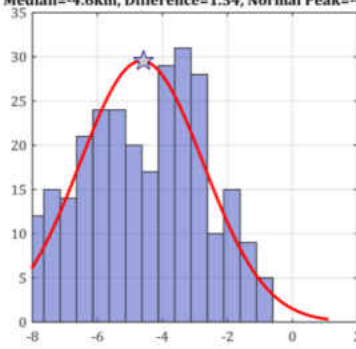
Beam=B4, Pol=23H, Ascending
Median=-2.0km, Difference=-1.54, Normal Peak=-1.51
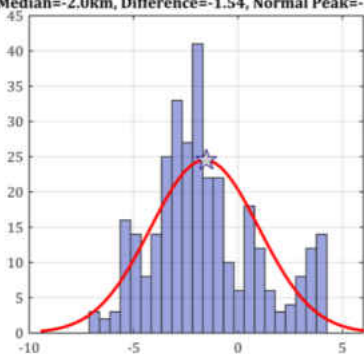
Beam=B4, Pol=23H, Descending,
Median=-0.4km, Difference=-1.54, Normal Peak=-0.41
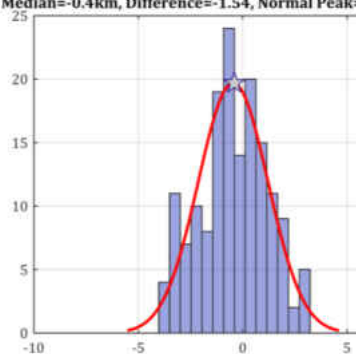
Beam=B3, Pol=23H, Ascending
Median=-2.7km, Difference=0.57, Normal Peak=-2.29
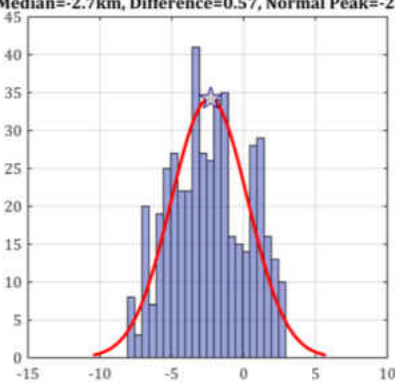
Beam=B3, Pol=23H, Descending,
Median=-3.2km, Difference=0.57, Normal Peak=-3.11

57
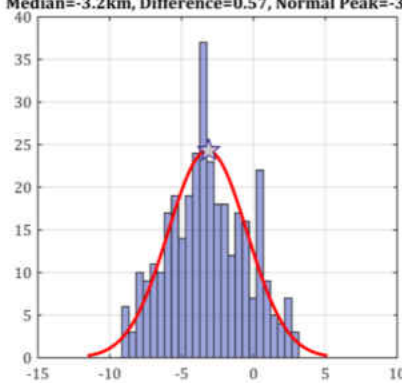
Beam=B2, Pol=23H, Ascending
Median=-3.9km, Difference=4.23, Normal Peak=-3.90
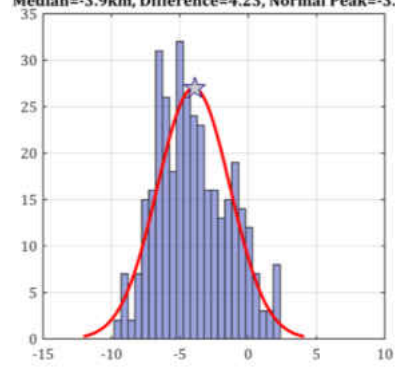
Beam=B2, Pol=23H, Descending,
Median=-8.1km, Difference=4.23, Normal Peak=-6.51
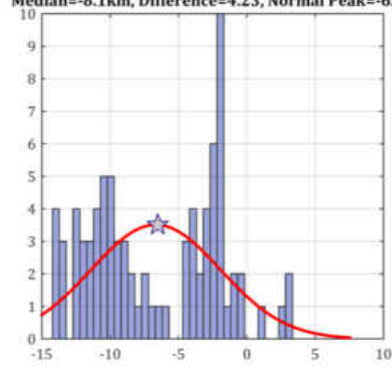
Beam=B1, Pol=23H, Ascending
Median=-1.6km, Difference=1.49, Normal Peak=-1.65
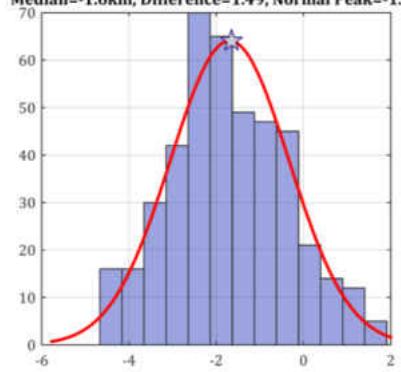
Beam=B1, Pol=23H, Descending,
Median=-3.1km, Difference=1.49, Normal Peak=-3.18

58
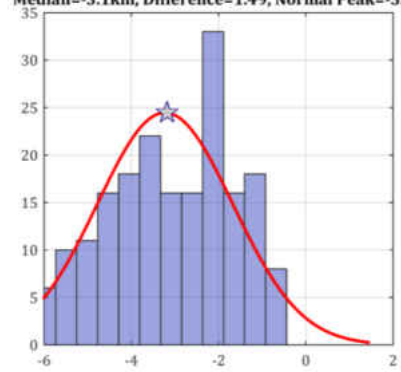
Beam=B8, Pol=37V, Ascending
Median=-6.8km, Difference=2.45, Normal Peak=-6.42
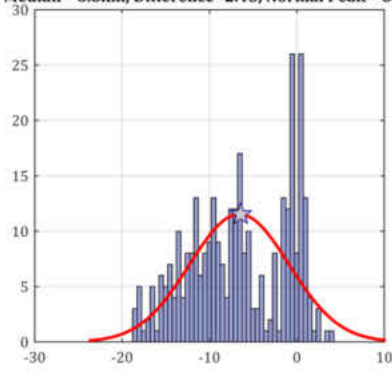
Beam=B8, Pol=37V, Descending,
Median=-9.2km, Difference=2.45, Normal Peak=-6.97
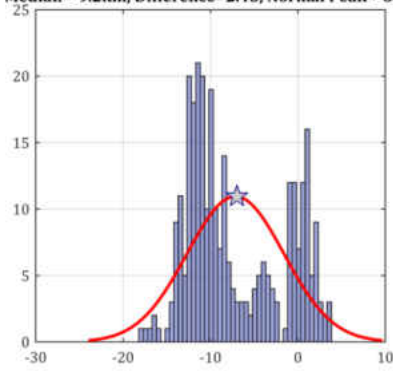
Beam=B7, Pol=37V, Ascending
Median=1.4km, Difference=-0.44, Normal Peak=1.79
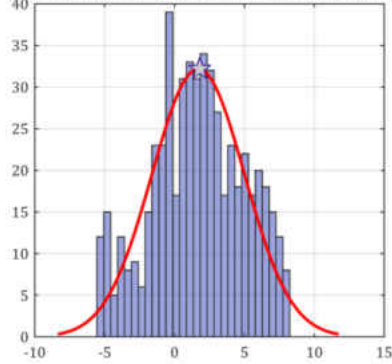
Beam=B7, Pol=37V, Descending,
Median=1.9km, Difference=-0.44, Normal Peak=2.09
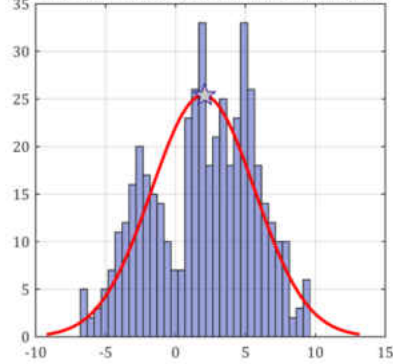
Beam=B6, Pol=37V, Ascending
Median=1.4km, Difference=3.95, Normal Peak=1.84
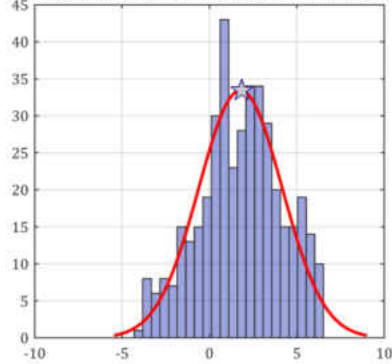
Beam=B6, Pol=37V, Descending,
Median=-2.5km, Difference=3.95, Normal Peak=-2.95
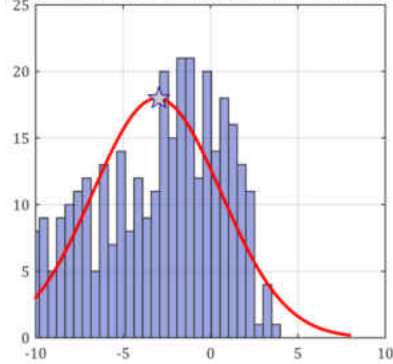
Beam=B5, Pol=37V, Ascending
Median=6.8km, Difference=1.31, Normal Peak=6.76
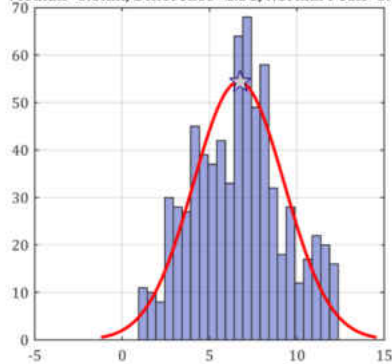
Beam=B5, Pol=37V, Descending,
Median=5.5km, Difference=1.31, Normal Peak=5.71
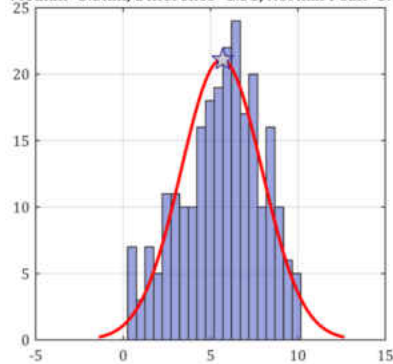
Beam=B4, Pol=37V, Ascending
Median=-6.2km, Difference=0.85, Normal Peak=-4.23
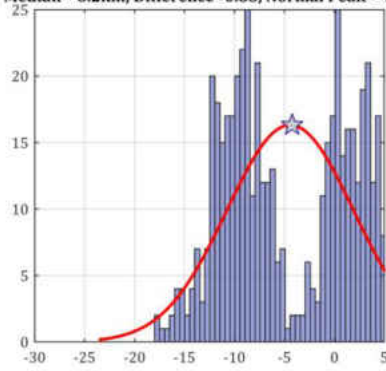
Beam=B4, Pol=37V, Descending,
Median=-7.0km, Difference=0.85, Normal Peak=-7.49
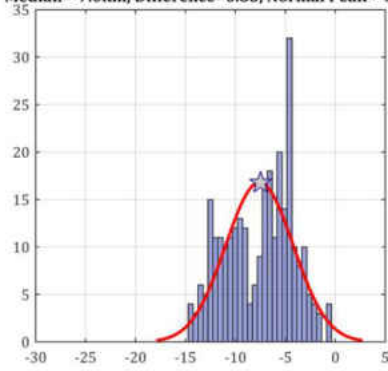
Beam=B3, Pol=37V, Ascending
Median=4.6km, Difference=3.91, Normal Peak=4.80
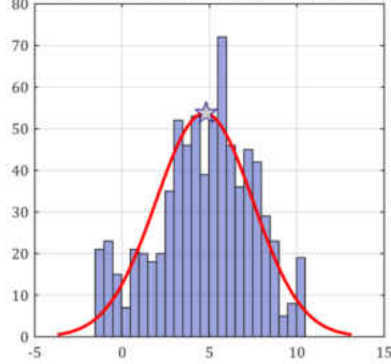
Beam=B3, Pol=37V, Descending,
Median=0.7km, Difference=3.91, Normal Peak=0.67
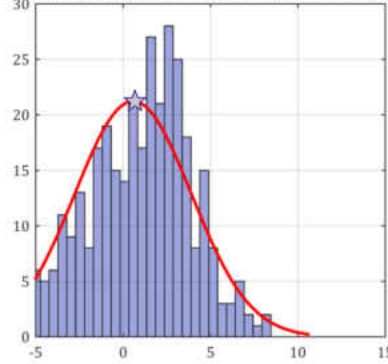
Beam=B2, Pol=37V, Ascending
Median=-4.7km, Difference=7.92, Normal Peak=-3.97
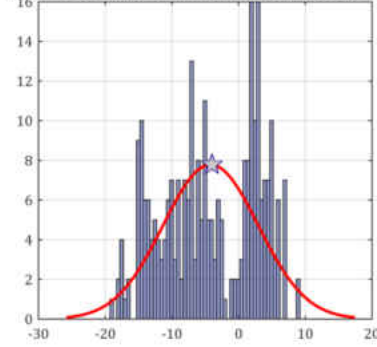
Beam=B2, Pol=37V, Descending,
Median=-12.6km, Difference=7.92, Normal Peak=-9.85
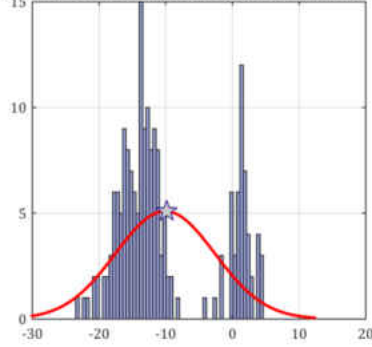
Beam=B1, Pol=37V, Ascending
Median=0.5km, Difference=-0.65, Normal Peak=0.58
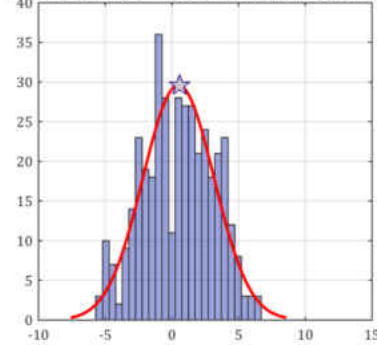
Beam=B1, Pol=37V, Descending,
Median=1.1km, Difference=-0.65, Normal Peak=1.59
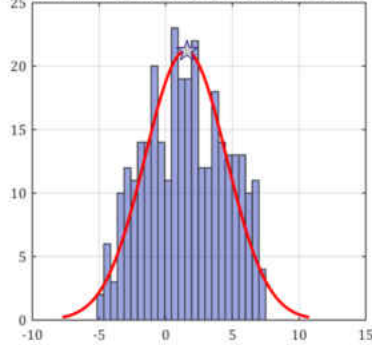
Beam=B8, Pol=37H, Ascending
Median=-3.1km, Difference=5.24, Normal Peak=-2.91
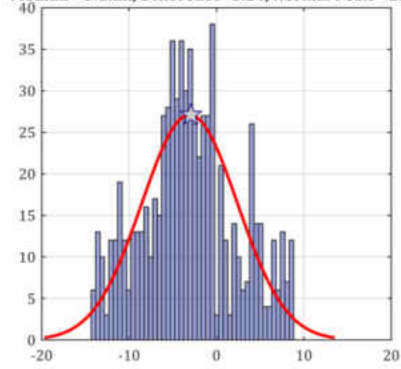
Beam=B8, Pol=37H, Descending,
Median=-8.4km, Difference=5.24, Normal Peak=-6.07
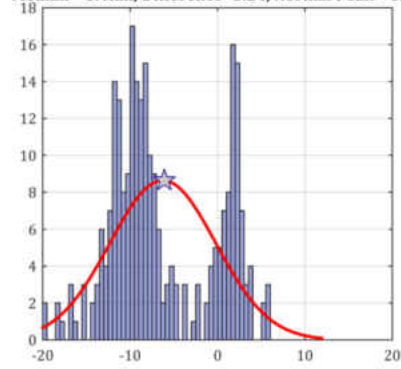
Beam=B7, Pol=37H, Ascending
Median=-1.1km, Difference=-3.88, Normal Peak=-0.65
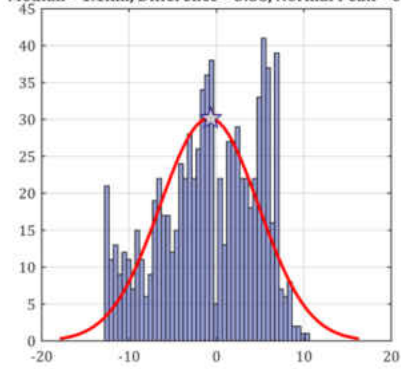
Beam=B7, Pol=37H, Descending,
Median=2.8km, Difference=-3.88, Normal Peak=1.84
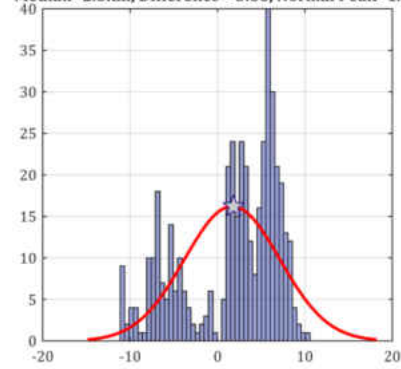
Beam=B6, Pol=37H, Ascending
Median=-1.7km, Difference=3.69, Normal Peak=-1.62
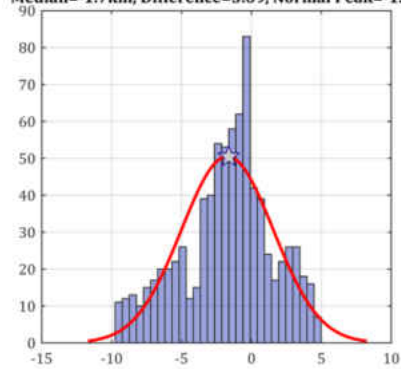
Beam=B6, Pol=37H, Descending,
Median=-5.4km, Difference=3.69, Normal Peak=-4.94
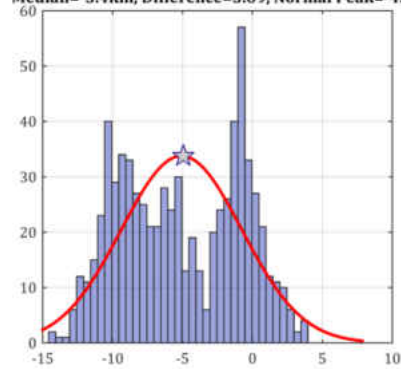
Beam=B5, Pol=37H, Ascending
Median=-1.4km, Difference=1.11, Normal Peak=-1.36
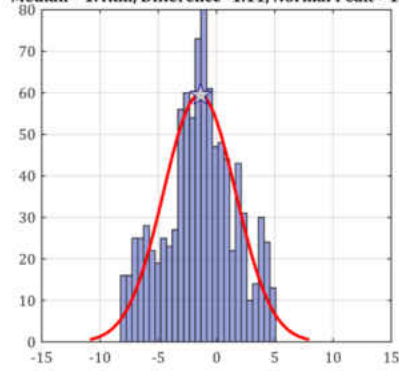
Beam=B5, Pol=37H, Descending,
Median=-2.5km, Difference=1.11, Normal Peak=-1.88
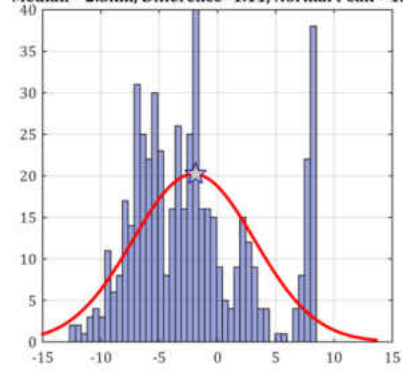
Beam=B4, Pol=37H, Ascending
Median=-4.6km, Difference=-1.69, Normal Peak=-4.78
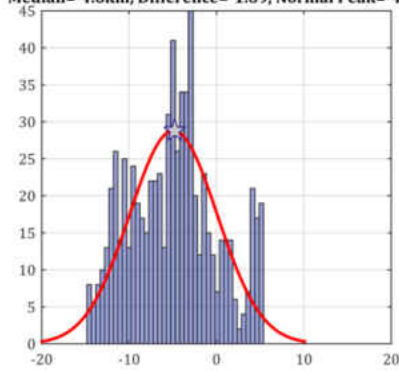
Beam=B4, Pol=37H, Descending,
Median=-2.9km, Difference=-1.69, Normal Peak=-2.86
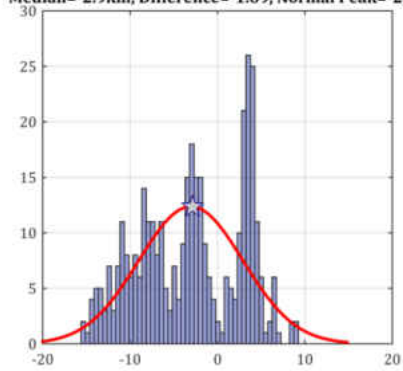
Beam=B3, Pol=37H, Ascending
Median=1.4km, Difference=2.96, Normal Peak=2.47
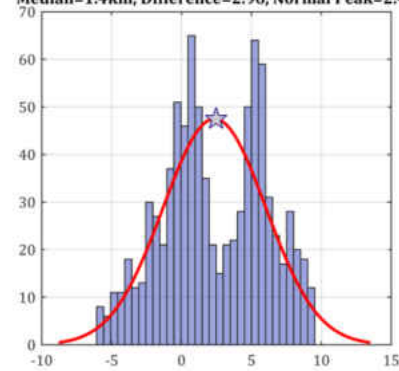
Beam=B3, Pol=37H, Descending,
Median=-1.6km, Difference=2.96, Normal Peak=-1.95

62
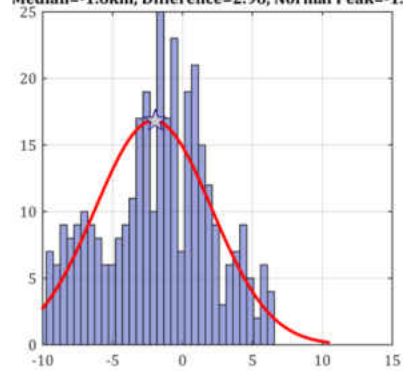
Beam=B2, Pol=37H, Ascending
Median=-8.1km, Difference=-4.98, Normal Peak=-6.81
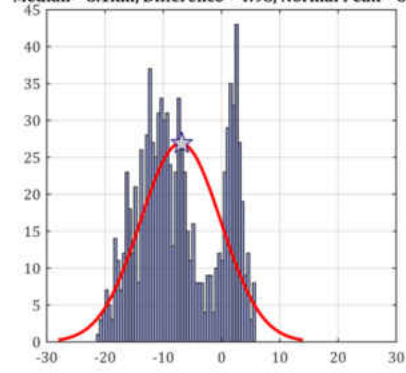
Beam=B2, Pol=37H, Descending,
Median=-3.1km, Difference=-4.98, Normal Peak=-5.05
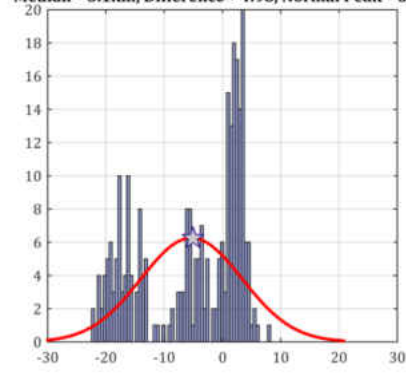
Beam=B1, Pol=37H, Ascending
Median=-8.3km, Difference=-2.99, Normal Peak=-7.31
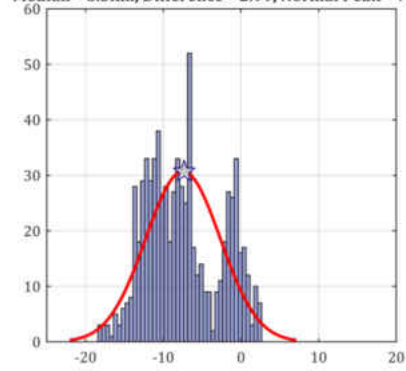
Beam=B1, Pol=37H, Descending,
Median=-5.3km, Difference=-2.99, Normal Peak=-4.74
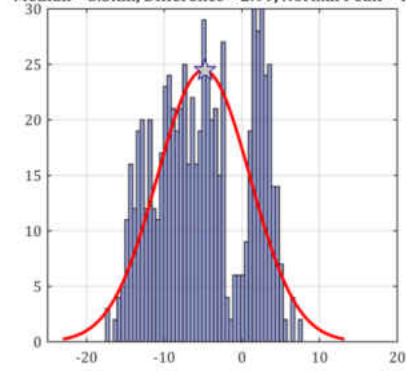
APPENDIX B: MATLAB CODE

In the following code, a MATLAB package is presented which can be used to process MWR data, from raw .mat files. Scripts are included to classify supersites from raw Beta v3.0 files, group them by beam and polarization, search for these supersites in Beta v3.0 files, compute and project the maximum slope point onto a 1km coastline map, and finally record these results in a single structure, results.

Classify_Orbit.m

```matlab
function output_struct = classify_orbit( data )

if                          nargin                    ==                    0,
   clc
   data     =      evalin(      'base'      ,      'current_file.data'      )
end
```

Pre-Process: Generate a list of fields to grab, set up

```matlab
data_fields              = fieldnames( data )                          ;
empty_cells              = cell( size( data_fields ) )                 ;
zero_cells               = repmat( { zeros( 4000 , 1 ) } , 3 , 1 )     ;
output_fields                    = vertcat( data_fields' , empty_cells' )
output_struct            = struct( output_fields{ : } )                ;
for      i_pols              =      1      :      numel(      data_fields     )
   output_struct.( data_fields{ i_pols } ) = process_pol( data.( data_fields{
i_pols         }         )         )                                   ;
   if     isempty(     output_struct.(     data_fields{     i_pols     }     )     )
       return
   end
end

   end
```

65

Main Processing: pad input data with NaNs so it can be processed rectangularly

```matlab
function pol_info = process_pol( pol_struct )
graph = false ;
for i_beam = 1 : 8
    BF = [ 'B' num2str( i_beam ) ] ;
    in_size = size( pol_struct.( BF ).Lat ) ;
    pad_mat = NaN( [ 4001 2 ] - in_size ) ;
    pol_info.( BF ).Lat = [ pol_struct.( BF ).Lat ; pad_mat ] ;
    pol_info.( BF ).Lon = [ pol_struct.( BF ).Lon ; pad_mat ] ;
    pol_info.( BF ).Tb = [ pol_struct.( BF ).Tb ; pad_mat ] ;

    [ mins , maxes ] = minmax( pol_info.( BF ).Lat ) ;
    equator_indeces = find_zero_crossings( pol_info.( BF ).Lat ) ;
    if isempty( equator_indeces.asc )
        pol_info.( BF ).Lat = [ ] ;
        pol_info.( BF ).Lon = [ ] ;
        pol_info.( BF ).Tb = [ ] ;
        pol_info.( BF ).asc_node= [ ] ;
        return
    end

    if graph
        close all hidden
        figure
        plot( pol_info.( BF ).Lon , pol_info.( BF ).Lat , 'LineSmoothing' , 'on' )
        grid on
        xlim( [ -182 182 ] )
        ylim( [ -92 92 ] )
        hold on
        scatter( pol_info.( BF ).Lon( equator_indeces.asc ) , pol_info.( BF ).Lat( equator_indeces.asc ) , 'r' )
        scatter( pol_info.( BF ).Lon( equator_indeces.des ) , pol_info.( BF ).Lat( equator_indeces.des ) , 'g' )
        legend( { 'Flight Path' , 'Ascending Node' , 'Descending Node' } )
```

```matlab
        end

%       We  have  a  padded  4001  x  1  vector,  this  shifts  the  ascending
%           node  to  be  at  the  quartile  for  use  with  imagesc.
        shift_mat                 = [ 1001 - equator_indeces.asc( 1 ) , 0 ]       ;
        pol_info.( BF ).Lat       = circshift( pol_info.( BF ).Lat , shift_mat )  ;
        pol_info.( BF ).Lon       = circshift( pol_info.( BF ).Lon , shift_mat )  ;
        pol_info.( BF ).Tb        = circshift( pol_info.( BF ).Tb  , shift_mat )  ;
        pol_info.( BF ).asc_node  = pol_info.( BF ).Lon( 1001 )                   ;
        if                   i_beam                        ==                   1
        fprintf( 'Ascending Note for beam %d at %0.3f Longitude\n' , i_beam , pol_info.(
BF                                    ).asc_node                                  )
        end
    end
end
```

Convenience function for finding min and max in one function call

```matlab
    function  [    minimums    ,    maximums    ]    =    minmax(    x    )
    x(  x  <  -180  )     =  NaN                                              ;
    min_val             =  min(  x  )                                         ;
    max_val             =  max(  x  )                                         ;
    minimums             =  find(  x  ==  min_val  )                          ;
    maximums             =  find(  x  ==  max_val  )                          ;
end
```

Function to find the zero crossings, used to find equator on latitude

```matlab
    function    crossings         =       find_zero_crossings(    x    )
    x(  x  <  -180  )     =  NaN                                              ;
    raw                   =  [  0  ;  diff(  sign(  x  )  )  ]               ;
    crossings.asc  =  find(  raw  ==  2  )                                    ;
    crossings.des  =  find(  raw  ==  -2  )                                   ;
    plot(                                x                                    )
    drawnow
end
```

Find SuperSite from bulk tbs.m

```matlab
        clc
clear
close all hidden
% profile on
% camera      = 'ge'
addpath( genpath( 'C:\Users\Student\Dropbox\CFRSL Projects\MWR GeoLocation\Brad''s
Code\Analysis' ) )

if ~exist( 'ge' , 'var' ) & ( camera( 1:2 ) == 'ge' )
    addpath( genpath( 'C:\Users\Student\Dropbox\CFRSL Projects\MWR
GeoLocation\Grunt_Scripts' ) )
    ge = GEserver      ;
end

if ~( exist( 'Lat' , 'var' ) && exist( 'Lon' , 'var' ) )
    coast_file  = 'C:\Users\Student\Dropbox\CFRSL Projects\MWR GeoLocation\Brad''s
Code\coast_1km.mat'
    load( coast_file )
end

lat_min             = -40
;
lat_max             = 60
;
pole_cut            = ( lat > lat_max ) | ( lat < lat_min )
;
fprintf( '%d coast points cut, out of %d points. %0.2f reduction.\n' , sum( pole_cut )
, numel( lat ) , sum( pole_cut ) / numel( lat ) )
lat( pole_cut )     = [] ;
lon( pole_cut )     = [] ;
% asc_slope         = [ -39.17 -39.43 ] , [ -17.58 -16.54 ]
asc_slope           = atand( diff( [ -17.58 -16.54 ] ) / diff( [ -39.17 -39.43 ] )
) + 180
des_slope           = atand( diff( [ -17.58 -16.54 ] ) / diff( [ -39.17 -39.43 ] )
) * -1
```

```matlab
circum                  = 40075
desired_rad             = 65
ref_marker              = @( a , o ) 30 * 360 * [ -1 1 ] * 1 / ( circum * cosd( a ) ) +
o


set( 0 , 'DefaultAxesXgrid' , 'on' , ...
        'DefaultAxesYgrid' , 'on' , ...
        'DefaultAxesZgrid' , 'on' , ...
        'DefaultLineLineSmoothing' , 'off' , ...
        'DefaultFigureUnits' , 'Normalized' , ...
        'DefaultFigurePosition' , [ 0.1 0.1 0.8 0.8 ] )



% if ~exist( 'bulk_tbs' , 'var' )
%     in_file              = 'W:\MWR\MWR_Beam_Pointing_Analysis\Brad''s Code\Supersite
Files\003\003.mat'  ;
%     load( in_file )
%     disp( 'Bulk_tbs loaded from default' )
% end

root_direc              = 'W:\MWR\MWR_Beam_Pointing_Analysis\Brad''s Code\Supersite
Files'  ;
folder_direc            = dir( fullfile( root_direc , '*.' ) )
folder_direc( 1:2 )     = []
progressbar( 'Orbits (out of 103)' , 'Channels (out of 3)' , 'Beams (out of 8)' ,
'Ascending Sites' , 'Descending Sites' )

pb                      = struct( 'orbit' , 0 , 'channel' , 0 , 'beam' , 0 , 'asc' , 0
, 'des' , 0 )

for i_orbit = 64 : 64 %numel( folder_direc )
subfolder           = folder_direc( i_orbit ).name
;
stem                = [ subfolder filesep subfolder '.mat' ]
;
in_file             = fullfile( root_direc , stem )
load( in_file )
clear s
```

```matlab
[ p,f,~ ]                = fileparts( in_file )
channels                 = { 'RX37V' , 'RX37H' , 'RX23H' }
;
beams                    = { 'B1' , 'B2' , 'B3' , 'B4' , 'B5' , 'B6' , 'B7' , 'B8' }
;
supersite_mat_beams      = { 'b_1' , 'b_2' , 'b_3' , 'b_4' , 'b_5' , 'b_6' , 'b_7' ,
'b_8' } ;
offset_angles.asc        = [ linspace(  16 ,  60 , 8 ) ;
...
                             linspace(  16 ,  60 , 8 ) ;
...
                             linspace( -60 , -16 , 8 ) ] + asc_slope
;
offset_angles.des        = [ linspace(  16 ,  60 , 8 ) ;
...
                             linspace(  16 ,  60 , 8 ) ;
...
                             linspace( -60 , -16 , 8 ) ] + des_slope
;
orbit                    = f
;


try
    close( coast.fig )
    disp( 'Old Coast Figure Deleted.' )
catch coast_figure_error
    disp( 'Unable to close coast figure.' )
end

coast.fig      = figure( 'Position' , [ 1.1000    0.1000    0.4000    0.6511 ] )
coast.ax       = axes
coast.xlim     = [ -83.212295  , -80.317325    ]
coast.ylim     = [  27.937851  ,  29.458046    ]
ecc            = axes2ecc( 30 , 15 )



coast.plot     = plot( coast.ax , lon , lat , '.k' )
set( coast.ax , 'Clipping' , 'on' )
```

```matlab
% xlim( coast.ax ,   )
% ylim( coast.ax ,   )
axis( coast.ax , [ coast.xlim coast.ylim ] )
axis square
drawnow
hold on
run.plot          = plot( 30 , 30 , 'ow' )
run.poss          = plot( 30 , 30 , 'LineStyle' , 'none' , 'Marker' , 'o' ,
'MarkerFaceColor' , 'k' , 'MarkerEdgeColor' , [ 0 1 0 ] )
ref.line          = plot( 30 , 30 , 'LineWidth' , 1 , 'Color' , [ 0.8 0.8 0.8 ] )


for chan_num = 1 : numel( channels )
    for beam_num = 1 : numel( beams )
%        close all hidden
        chan                    = channels{ chan_num }
;
        beam                    = beams{ beam_num }
;
        supersite_beam          = supersite_mat_beams{ beam_num }
;
        slope_threshold         = 18
              ;
        if chan == 'RX37V'
            slope_threshold = 6
        elseif chan == 'RX37H'
            slope_threshold = 13    ;
        end
                                                          ;
        current_img             = images.( chan ).( beam )
;
        tb_median               = nanmedian( current_img )
;
        tb_diff                 = [ 0 diff( tb_median ) ]
;
        current_lats            = lats.( chan ).( beam )
;
        current_lons            = lons.( chan ).( beam )
```

71

```matlab
;
        lat_median              = nanmedian( current_lats )
;
        lon_median              = nanmedian( current_lons )
;
        smooth_lat              = medfilt2( lat_median , [ 1 30 ] )
;
        smooth_lon              = medfilt2( lon_median , [ 1 30 ] )
;
        signal_index            = 1 : numel( smooth_lat )
;


        %   Logical Test Values
        peak                    = diff( [ 0 sign( [ diff( tb_diff ) 0 ] ) ] ) == -2
;
        peak                    = true( size( peak ) )
;
        trough                  = diff( [ 0 sign( diff( [ 0 tb_median ] ) ) ] ) == 2
;
        good_signal             = ( signal_index > 500 ) & ( signal_index < 3000 )
;
        good_lat                = ( smooth_lat > lat_min ) & ( smooth_lat < lat_max )
;
        asc                     = [ 0 ( diff( smooth_lat ) > 0 ) ]
;
        des                     = [ 0 ( diff( smooth_lat ) < 0 ) ]
;
        water_to_land_poss      = tb_diff > slope_threshold
;

        asc_des_poss            = good_signal & good_lat & water_to_land_poss & peak
;
        asc_des_num             = find( asc_des_poss )
;
        asc_poss                = asc_des_poss & asc
;
        asc_num                 = find( asc_poss )
;
```

72

```matlab
        des_poss                    = asc_des_poss & des
;
        des_num                     = find( des_poss )
;
        logic_mat                   = vertcat( peak , good_signal , good_lat , asc   ,
water_to_land_poss )  ;
        figure( 'Position' , [ 0 0 1 0.5 ] )
        imagesc( logic_mat )


        h.fig   = figure( 'Position' , [ 0.6097    0.0433    0.3917    0.9567 ] )
        sp( 1 ) = subplot( 211 )
        p_1     = plotyy( signal_index , tb_diff , signal_index( good_lat & good_signal
) , smooth_lat( good_lat & good_signal)  )
        tb_line = findobj( get( p_1( 1 ) , 'Children' ) , 'Type' , 'Line' )
        lat_lin = findobj( get( p_1( 2 ) , 'Children' ) , 'Type' , 'Line' )
        set( lat_lin , 'Marker' , '.' , 'LineStyle' , '-' , 'MarkerFaceColor' , 'k' )
        set( tb_line , 'Marker' , 'x' , 'LineStyle' , '-' )
        set( get( p_1( 1 ) , 'Children' ) , 'LineStyle' , '.-' , 'MarkerFaceColor' ,
'k' )
        hold on
        plot( signal_index( asc_des_poss & good_signal ) , tb_diff( asc_des_poss &
good_signal ) , 'or' )
        hold on
        plot( [ 1 numel( signal_index ) ] , slope_threshold * [ 1 1 ] , 'r' )
        title( 'Derivatives' )
        grid on
        marker( 1 )                 = plot( 1 , 1 , 'k+' )
;


        sp( 3 ) = subplot( 212 )
;
        plot( signal_index , tb_median , '.-g' , 'MarkerEdgeColor' , 'k' )
        hold on
        plot( signal_index( asc_des_poss ) , tb_median( asc_des_poss ) , 'or' )
        title( 'Averaged Data' )
        grid on
        linkaxes( sp , 'x' )
```

```matlab
        drawnow
        marker( 3 )               = plot( 1 , 1 , 'k+' )
;

        condition_string         = sprintf( '%s, %s' , chan , beam )
        set( gcf , 'NumberTitle' , 'off' , 'Name' , condition_string )

        asc_supersites            = { [] }
;


        set( run.plot , 'XData' , smooth_lon , 'YData' , smooth_lat )
        set( run.poss , 'XData' , smooth_lon( asc_des_poss ) , 'YData' , smooth_lat(
asc_des_poss ) )

        set( coast.ax , 'Color' , [ 1 0.2 0.2 ] )
        set( run.poss , 'MarkerEdgeColor' , [ 0 1 0 ] )
        axis( coast.ax , 'square' )

%        pause
        for i = 1 : numel( asc_num )
            title( coast.ax , [ condition_string ' Ascending' ] )

            xlim( sp( 1 )  , asc_num( i ) + [ -15 15 ] )
            xlim( p_1( 2 ) , asc_num( i ) + [ -15 15 ] )
            current_latlon    = [ lat_median( asc_num( i ) ) lon_median( asc_num( i )
) ] ;
            cc                = num2cell( current_latlon )          ;
            determine_box_limits
            axis( coast.ax , [ lonlim latlim ] )

            set( marker( 1 ) , 'XData' , asc_num( i ) , 'YData' , tb_diff( asc_num( i )
) )
            set( marker( 2 ) , 'XData' , asc_num( i ) , 'YData' , smooth_lat( asc_num(
i ) ) )
            set( marker( 3 ) , 'XData' , asc_num( i ) , 'YData' , tb_median( asc_num( i
) ) )
            rot               = offset_angles.asc( chan_num , beam_num ) * -1
;
```

```matlab
            [elat,elon]          = ellipse1(cc{ : },[30 ecc],rot,[],earthRadius('km'));
            set( ref.line , 'XData' , elon , 'YData' , elat )
            scene_radius_km      = 65
;
            one_km_alt            = 864.3388
;
            total_alt            = 2 * scene_radius_km * one_km_alt
;
            GEcamera( ge , [ current_latlon( 1 ) , current_latlon( 2 ) , 0 ] ,
        ...
                        [ total_alt 0 0 ] )

            [point]              = GEcamera(ge)
;

            clipboard( 'copy' , sprintf( '%0.8f , %0.8f' , current_latlon ) )
            fprintf( 'Current Lat: %0.2f, Current Lon: %0.2f\n' , current_latlon )
            fprintf( 'Orbit %d, channel %s, beam %s, \n' , i_orbit , chan , beam , i )
            user_choice      = input( [ sprintf( 'Ascending site %d of %d. Keep result?
(y/n, x to exit) --> ' , i , numel( asc_num ) ) ] , 's' )     ;
            if lower( user_choice ) == 'y'
                asc_supersites{ end+1 , 1 }     = current_latlon
;
            end
            if lower( user_choice ) == 'x'
                 return
            end
        pb.asc = i / numel( asc_num )
;

        end
        s.( supersite_beam ).( chan ).asc = [ asc_supersites{ : } ]

        des_supersites              = { [] }
;
        set( coast.ax , 'Color' , [ 0.2 0.2 1 ] )
        set( run.poss , 'MarkerEdgeColor' , [ 1.0000    0.6445        0 ] )
        axis( coast.ax , 'square' )
```

```matlab
        for i = 1 : numel( des_num )
            title( coast.ax , [ condition_string ' Descending' ] )

            xlim( sp( 1 )   ,  des_num( i ) + [ -15 15 ] )
            xlim( p_1( 2 ) ,  des_num( i ) + [ -15 15 ] )
            current_latlon  = [ lat_median( des_num( i ) ) lon_median( des_num( i ) ) ]
;
            cc              = num2cell( current_latlon )         ;
            determine_box_limits
%           set( ref.line , 'XData' , ref_marker( cc{ : } ) , 'YData' , [ 1 1 ] *
current_latlon( 1 ) )
%           ylim( coast.ax , latlim )
%           xlim( coast.ax , lonlim )
            axis( coast.ax , [ lonlim latlim ] )

            set( marker( 1 ) , 'XData' , des_num( i ) , 'YData' , tb_diff( des_num( i )
) )
%           set( marker( 2 ) , 'XData' , des_num( i ) , 'YData' , smooth_lat(
des_num( i ) ) )
            set( marker( 3 ) , 'XData' , des_num( i ) , 'YData' , tb_median( des_num( i
) ) )
            rot             = offset_angles.des( chan_num , beam_num ) * -1
;

            [elat,elon]     = ellipse1(cc{ : },[30 ecc],rot,[],earthRadius('km'));
            set( ref.line , 'XData' , elon , 'YData' , elat )
            scene_radius_km   = 65
;
            one_km_alt        = 864.3388
;
            total_alt         = 2 * scene_radius_km * one_km_alt
;
            GEcamera( ge , [ current_latlon( 1 ) , current_latlon( 2 ) , 0 ] ,
        ...
                    [ total_alt 0 0 ] )

            [point]         = GEcamera(ge)
```

```matlab
                clipboard( 'copy' , sprintf( '%0.8f , %0.8f' , current_latlon ) )
                fprintf( 'Current Lat: %0.2f, Current Lon: %0.2f\n' , current_latlon )
                fprintf( 'Orbit %d, channel %s, beam %s\n' , i_orbit , chan , beam )
                user_choice     = input( [ sprintf( 'Descending site %d of %d. Keep result?
(y/n, x to exit) --> ' , i , numel( des_num ) ) ] , 's' )   ;
                if lower( user_choice ) == 'y'
                    des_supersites{ end+1 , 1 }     = current_latlon
;
                end
                if lower( user_choice ) == 'x'
                    return
                end
                pb.desc = i / numel( des_num )
;

            end
            try
                close( h.fig )
            catch err
            end
            s.( supersite_beam ).( chan ).des = [ des_supersites{ : } ]
            pb.beam = beam_num / numel( beams )
        end
        pb.channel = chan_num / numel( channels )
        progressbar( pb.orbit , pb.channel , pb.beam , pb.asc , pb.des )
    end
    save( [ p filesep 'ss_' f ] , 's' )
    pb.orbit    = i_orbit / numel( folder_direc )   ;

end
```

Process_Beta3_files.m

```matlab
    clc
%   Grab a path of useful folders; machine-insensitive, designed to set up
%   a file list on the lab machine or the server.
add_necessary_folders

direc.grids        = dir( folders.mas_grid )
;
direc.grids( 1:2 )  = []
;
direc.asc          = dir( fullfile( folders.mas_grid , direc.grids( 1 ).name ,
'*.mat' ) )   ;
direc.des          = dir( fullfile( folders.mas_grid , direc.grids( 2 ).name ,
'*.mat' ) )   ;
direc.beta3        = load( fullfile( folders.brads , 'file_list.mat' ) )
;
addpath( genpath( 'helpers' ) )
options.convolve   = false
;
results            = {}
;
```

Begin Processing

```matlab
    % We start by stripping the datenum from each grid file's name.
    for i_grids = 1 : numel( direc.asc )
    grid_dates.asc( i_grids )  = datenum( direc.asc( i_grids ).name( 7:14 ),
'yyyymmdd' )                    ;
end

for i_grids = 1 : numel( direc.des )
    grid_dates.des( i_grids )  = datenum( direc.des( i_grids ).name( 7:14 ),
'yyyymmdd' )                    ;
end
```

```matlab
if ~exist( 'loaded_grid_file' , 'var' )
loaded_grid_file.asc        = 'none'
;
loaded_grid_file.des        = 'none'
;
end

min_date_num                = str2num( '2012302' )
;
max_date_num                = str2num( '2013300' )
;
all_date_strings            = { direc.beta3.file_list.name }
;
all_date_nums               = cellfun( @( x ) str2double( x( 2:8 ) ) ,
all_date_strings )'                  ;
good_date_indeces           = find( ( all_date_nums > min_date_num ) & (
all_date_nums < max_date_num ) )    ;
coast_file                  = ( 'coast_1km.mat' )
;
coast                       = load( coast_file )
;
load( 'ss_master.mat' )
tic
```

File Matching Section

```matlab
    for i_beta = good_date_indeces'%( good_date_indeces >= 4015 )'
    current_datafile            = fullfile( folders.beta3 , direc.beta3.file_list(
i_beta ).name )          ;
sprintf( 'Current file: %s\n%0.2f%% complete\n.' ,current_datafile , i_beta / numel(
good_date_indeces * 100 ) )
current_date_string         = direc.beta3.file_list( i_beta ).name( 2:8 )
;
[ asc_match , des_match ]   = match_grid_to_data_file( current_datafile , grid_dates
)          ;
closest_grid                = struct( 'asc' , fullfile( folders.mas_grid ,
...
```

79

```matlab
                                                    'Ascending' ,
...
                                                    direc.asc( asc_match ).name )
,                          ...
                              'des' , fullfile( folders.mas_grid ,
...
                                                    'Descending' ,
...
                                                    direc.des( des_match ).name )
)                      ;

if options.convolve
    fprintf( 'Grid file name:\t\t\t\t\t\t\t%s\n\n' , direc.asc( asc_match ).name )
    if ~strcmp( loaded_grid_file.asc , closest_grid.asc )
        load( closest_grid.asc )                              ;
        loaded_grid_file.asc = closest_grid.asc               ;
        disp( 'New ascending grid file loaded.' )
    else
        disp( 'Existing ascending grid file used.' )
    end


    if ~strcmp( loaded_grid_file.des , closest_grid.des )
        load( closest_grid.des )                              ;
        loaded_grid_file.des = closest_grid.des               ;
        disp( 'New descending grid file loaded.' )
    else
        disp( 'Existing descending grid file used.' )
    end
end


%   Load the current data in .mat format. New variable is 'data.'
disp( current_datafile )
load( current_datafile )
data.filename       = current_datafile                                  ;
```

Actual Processing

```matlab
    disp(     [    'i_Beta    =    '    num2str(    i_beta    )    ]    )
r_new          = find_crossing( data , coast , ss_master )        ;
results              = vertcat( results , r_new )                          ;
drawnow

% ellipse_gen( center , ellipse_major_km , ellipse_minor_km , num_ellipses ,
offset_angle , plot_points , num_points )
```
```matlab
    end
toc
```

Find_crossing.m

```matlab
        function results = find_crossing( data , c , ss_master )
if nargin == 0
    load( 'test_state.mat' )
end
beams               = { 'b_1' , 'b_2' , 'b_3' , 'b_4' , 'b_5' , 'b_6' , 'b_7' ,
'b_8' }'    ;
beams_data          = { 'B1' , 'B2' , 'B3' , 'B4' , 'B5' , 'B6' , 'B7' , 'B8' }'
;
pols                = { 'RX23H' , 'RX37V' , 'RX37H' }'
;
ascdesc             = { 'asc' , 'des' }
;
radius              = 100
;
min_points          = 7
;
asc_sign            = struct( 'asc' , 1 , 'des' , -1 )
;
results             = { }
;

for i_beam = 1 : 8
    beam            = beams{ i_beam }                           ;
    B               = beams_data{ i_beam }                      ;
    for i_pol = 1 : 3
        pol         = pols{ i_pol }                             ;

        for i_ascdesc = 1 : 2
            ad      = ascdesc{ i_ascdesc }                      ;

            if isfield( ss_master.( beam ) , pol )
                if ~isempty( ss_master.( beam ).( pol ).( ad ) )
                    for i_site = 1 : size( ss_master.( beam ).( pol ).( ad ) , 1 )
                        this_site   = ss_master.( beam ).( pol ).( ad )( i_site ,
: ) ;
```

```matlab
                        [ n, w, s, e ]  = calc_geo_square( this_site( 1 ) ,
this_site( 2 ) , radius )   ;
                        d               = data.( pol ).( B )
     ;
                        d.asc           = sign( [ 0 ; diff( d.Lat ) ] ) > 0
     ;
                        d.des           = sign( [ 0 ; diff( d.Lat ) ] ) <= 0
     ;
                        d.good          = ( d.Lat > s ) & ( d.Lat < n ) &
...
                                          ( d.Lon > w ) & ( d.Lon < e )
     ;
                        if ~logical( i_ascdesc - 1 ) % ascending
                            all_good    = d.good & d.asc
     ;
                        else % descending
                            all_good    = d.good & d.des
     ;
                        end

                        if sum( all_good ) >= min_points
                            display_name = data.filename( 1:28 )
;
                            disp( fprintf( 'Processing %s at %0.2f Lat %0.2f Lon' ,
display_name , this_site ) )
                            current_pts = struct( 'Lat'     , d.Lat( all_good ) ,
...
                                                  'Lon'     , d.Lon( all_good ) ,
...
                                                  'Tb'      , d.Tb(  all_good ) ,
...
                                                  'az'      , d.az(  all_good ) ,
...
                                                  'fn'      , data.filename     ,
...
                                                  'beam'    , beam              ,
...
                                                  'pol'     , pol               ,
```

```matlab
...
                                                'ad'      , ad                ,
...
                                                'ss'      , this_site         ,
...
                                                'coast'   , []                ,
...
                                                'cross'   , []                )   ;
%                         save( 'test_state' )
                          [ Lat_fraction , mu ]   = project_max( current_pts )
;
                          if isnan( Lat_fraction )
                              disp( 'Points invalid. Continuing.' )
                              continue
                          end

                          crossing_point.Lat    = interp1( current_pts.Lat ,
current_pts.Lat , mu )        ;
                          crossing_point.Lon    = interp1( current_pts.Lat ,
current_pts.Lon , mu )        ;

                          c.good                = ( c.Lat > s ) & ( c.Lat < n ) &
      ...
                                                    ( c.Lon > w ) & ( c.Lon < e )
      ;
                          if ~any( c.good )
                              disp( 'No coastal points.' )
                              continue
                          end
                          c_raw                 = [ c.Lat( c.good ) c.Lon( c.good ) ]
;
                          c_mat                 = [ ( c.Lat( c.good )-
crossing_point.Lat ) * cosd( crossing_point.Lat )            ...
                                                    c.Lon( c.good )-
crossing_point.Lon ]'                                         ;
                          cross_mat             = repmat( [ crossing_point.Lat
crossing_point.Lon ] ,                                        ...
                                                        size( c.Lat( c.good ) , 1
```

84

```matlab
                    ) , [] )                                    ;
                            distances           = distance( cross_mat , c_raw ,
...
                                                referenceEllipsoid(
'earth' ) )          ;

                            min_ind             = distances == min( distances )
;
                            km_dist             = distances( min_ind )
;
                            closest_coast       = c_raw( min_ind , : )
;
                            current_pts.coast   = struct( 'Lat' , closest_coast( 1 )
,  ...
                                                'Lon' , closest_coast( 2 )
)  ;
                            current_pts.cross   = crossing_point
;

%                              km_dist              = angle_dist / 360 * 2 * pi *
earthRadius ;
                            error_sign          = asc_sign.( ad )
;
                            current_pts.error   = sign( crossing_point.Lat -
closest_coast( 1 ) ) * error_sign * km_dist / 1000 ;
                            %   NEGATIVE ERROR SIGN MEANS CROSSING
                            %   CALCULATED BEFORE COAST
                            results{ end+1, 1 } = current_pts
;
                    else
                        continue
                    end
                end
            end
        end
    end
end
```

Project max.m

```matlab
function [ lat_fraction , mu ] = project_max( current_pts )

    if nargin == 0
      plot_pts = true          ;
    else
        plot_pts = false         ;
    end

    close all
    mu               = nan
;

    current_max      = max( current_pts.Tb )
;

    current_min      = min( current_pts.Tb )
;

    current_range    = current_max - current_min
;

    if current_range < 40
        lat_fraction = nan
;

        return
    end
    current_pts.Tb     = ( current_pts.Tb - current_min ) / current_range
;

    lat_diffs          = diff( current_pts.Lat )
;

    input_std          = std( current_pts.Tb )
;

    avg_lat_diff       = mean( lat_diffs )
;

    diff_points        = current_pts.Lat  - avg_lat_diff / 2
;

    tb_diffs           = diff( current_pts.Tb )
;

    max_loc            = find( tb_diffs == max( tb_diffs ) , 1 , 'first' )
      ;
```

```
    eval_range           = max_loc-1 : max_loc+1
;

    if ( min( eval_range ) < 1 ) || ( max( eval_range ) > numel( tb_diffs ) )
        lat_fraction = nan
;

        return
    end
    max_eval             = diff( tb_diffs( eval_range ) )
;

    eval_quotient        = max( abs( max_eval ) ) / min( abs( max_eval ) )
;

    if eval_quotient > 1.5
        if diff( abs( max_eval ) ) < 0
            center_point = mean( diff_points( max_loc+1 : ( max_loc+2 ) ) )
;

        else
            center_point = mean( diff_points( max_loc   : ( max_loc+1 ) ) )
;

        end
    else
        center_point = diff_points( max_loc+1 )
;

    end
    fit_bounds           = center_point + 3.8 * avg_lat_diff * [ -1 1 ]
;

    fit_inds             = diff_points > min( fit_bounds ) & diff_points < max(
fit_bounds )    ;
    fit_inds( 1 )        = false
;

    fit_x                = diff_points( fit_inds )
;

    fit_ind_shift        = find( fit_inds ) - 1
;

    fit_y                = tb_diffs( fit_ind_shift )
;

    fit_max              = max( tb_diffs )
;

    [ s , mu , A ]       = mygaussfit( fit_x , fit_y )
```

87

```matlab
;
    if ~isreal( s )
        lat_fraction = nan;
        return
    end
    max_tb_point        = mu
;
        fit_curve           = A / sqrt( 2*pi*s ) * exp( -( ( diff_points - mu ) / (
2*s ) ) .^2 )   ;
    parabola            = polyfit( fit_x , fit_y , 2 )
;

    parab_vertex        = -parabola( 2 ) / ( 2*parabola( 1 ) )
;

    parab_val           = polyval( parabola , diff_points )
;

    lat_fraction        = ( mu - current_pts.Lat( 1 ) ) /
...
                             ( current_pts.Lat( end ) - current_pts.Lat( 1 ) )
;
    if abs( lat_fraction ) > 1 | ( lat_fraction < 0 )
        plot_pts = true
    end

    error_g             = sum( ( [ 0 ; tb_diffs ] - fit_curve ) .^2 )
;
    if plot_pts
    sp( 1 )             = subplot( 211 )
    plot( current_pts.Lat , current_pts.Tb , '.-' , 'LineSmoothing' , 'on' )
    grid on
    hold on
    plot( max_tb_point * [ 1 1 ] , [ min( current_pts.Tb ) max( current_pts.Tb ) ] ,
'r' )
    legend( { 'T_Bs' , 'Max' } , 'Location' , 'SouthEast')
    title( sprintf( '%s, \\sigma_{input} = %0.2g' , current_pts.fn , std(
current_pts.Tb ) ) )

    sp( 2 )             = subplot( 212 )
    plot( diff_points ,[ 0 ; tb_diffs ] , '.-' , diff_points , fit_curve , '.-' ,
```

```matlab
    diff_points , parab_val , '.-' , 'LineSmoothing' , 'on' )
    ylim( [ min( tb_diffs ) max( tb_diffs ) ] )
    grid on
    legend( { 'Input Diff' , 'Gaussian Fit' , 'Parabolic Fit' } , 'Location' ,
'NorthEast' )
    linkaxes( sp , 'x' )
    title( sprintf( 'Error of Gaussian = %0.2g' , error_g ) )
    drawnow
    end
    if abs( lat_fraction ) > 1 || ( lat_fraction < 0 )
        lat_fraction = nan;
    end

        end
```

# APPENDIX C: WINDSAT CROSS-VALIDATION

The geolocation analysis was performed using WindSat T_B data for the period of October 29th-December 9th, 2012. The WindSat was selected because the wide swath provides frequent overlap with MWR observations within a time window of ± 45 minutes. The 2012 WindSat sensor data records (SDR), obtained from the United States Naval Research Laboratory (NRL) (http://www.nrl.navy.mil/WindSat/), were used to produce brightness temperature images of the MWR supersites for 23.8 GHz-H-pol; 37 GHz V- & H-pol T_B's gridded on a 0.125°Lat/Lng earth-grid.

When evaluating separate tracts of the same geographical area (e.g., Southern Australia) within any week of time, T_B differences between 15-20°K were observed on a pixel by pixel basis. Since it is not possible to have simultaneous WindSat data for every pass, it was decided to use weekly averages. Further, to minimize T_B changes associated with the diurnal cycle, the WindSat data were separated into ascending and descending tracts to correspond to the individual collocations being considered.

It was important that the selected supersites have contiguous gridded brightness temperatures within the antenna pattern surface footprint; therefore to fill missing pixels average brightness temperature resulting from values of the surrounding cells were inserted.

Two other significant characteristic played an important factor in a supersite selection; first, no large bodies of water on land near the coast and secondly, no islands or land masses around the coast close to land were permitted. These two cases would bring anomalous temperature issues. In a pure sense, we were trying to measure a water/land boundary crossing to validate satellite antenna boresight geolocation.

A sample site is shown on the southern coast of Australia is shown in Figure 39. Figure 40 and Figure 41 are images displaying the same boxed-in supersite in South Australia, between latitudes [-31.16, -32.96] & longitudes [126.31, 128.44]. The supersite figures are a snapshot of a 7 day mean brightness temperature of an ascending orbit, of a 37 GHz V-pol. Channel. Figure 40's image denotes 5 locations which previously contained nan (not a number) values, originally gridded from SDR files, with the modification that a paint program was used to remove/replace the nans with average temperature values of the surrounding cells. Figure 41 shows the antenna pattern overlay used to provide support of the geolocation scheme mentioned above.
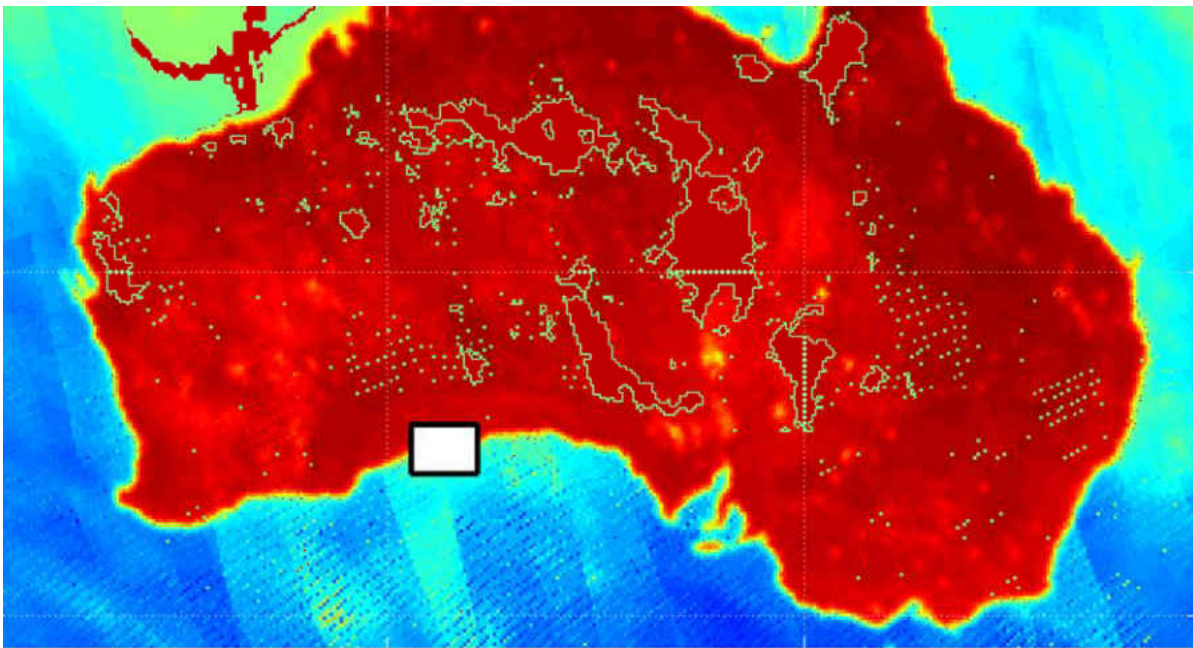
Figure 39: Southern Australia, with a box indicating a sample supersite location.
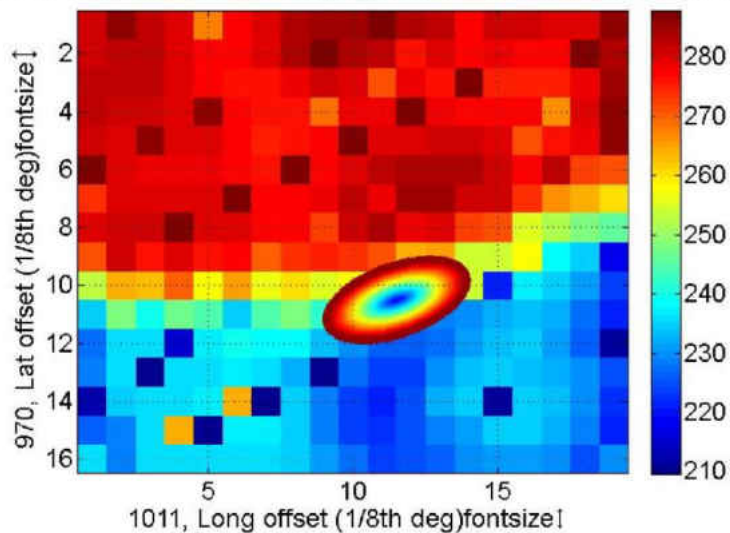
S. Australia, 7 Day Temperature in Ascending orbit 37v touched (Wk. 48, Nov. 26, 2012)
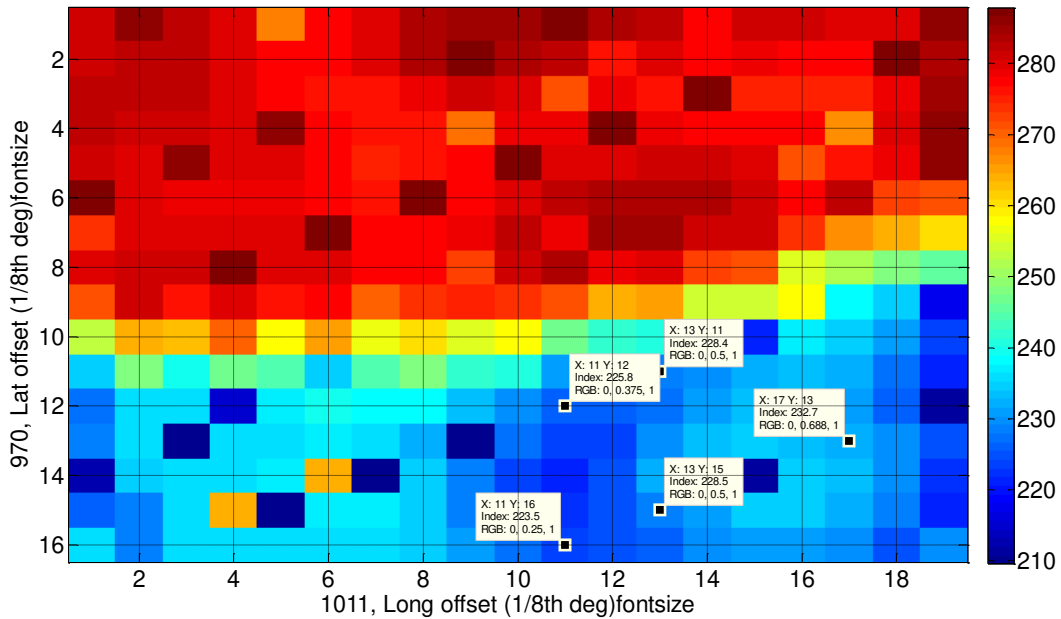
Figure 40: South Australia, gridded raw data, touched

7 Day Temperature in Ascending orbit 37v touched (Wk. 48, Nov.

Figure 41: South Australia, gridded with antenna pattern superimposed.

These gridded T_Bs are used as the map of apparent brightness temperatures (T_{AP}) for

comparison with measured T_Bs via numerical convolution (see 3.2: MWR Two-Dimensional

Convolution, page 21); this has been done in the present work to validate that the maximum slope point of a series of convolved T_Bs is collocated with that of measured T_Bs. As a proof of principal, this was done prior to full automation on a large series of land/water crossings, and the results are shown in Figure 42 and Figure 43 on the following page.
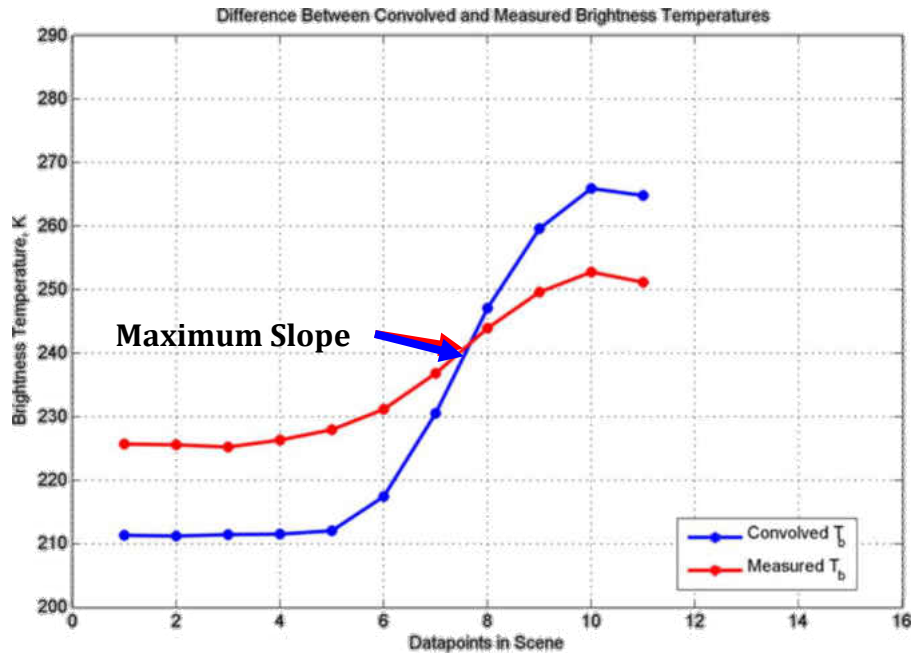
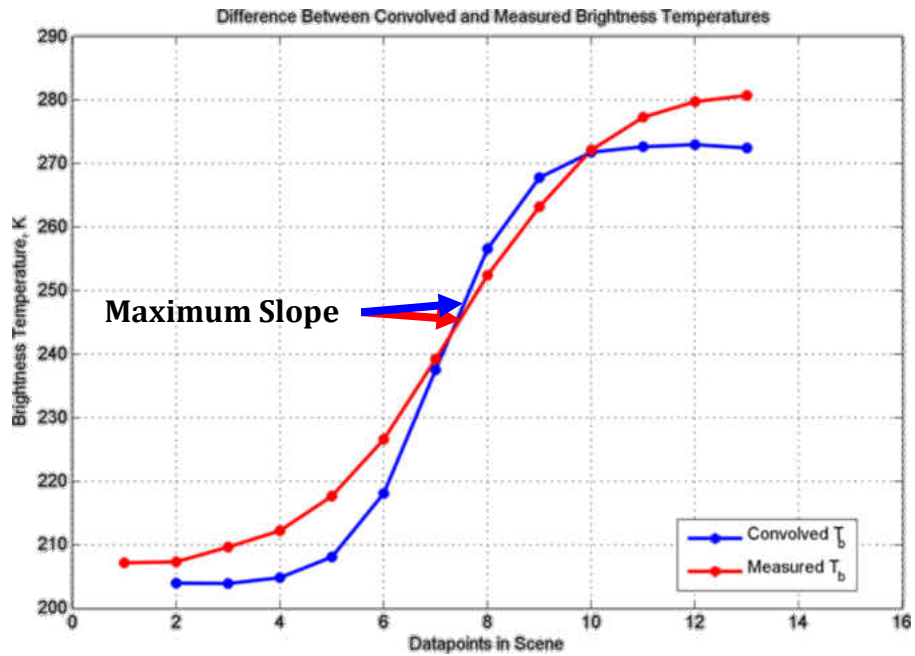**Figure 42**: Comparison of measured vs. convolved T_Bs, 36.5GHz v-pol

Figure 43: Comparison of measured vs. convolved T_Bs, 36.5GHz h-pol

REFERENCES

[1] G. Lagerloef, F. R. Clomb, D. Le Vine, F. Wentz, S. Yueh, C. Ruf and et al, ""THE AQUARIUS/SAC-D MISSION: DESIGNED TO MEET THE SALINITY REMOTE-SENSING CHALLENGE"," *OCEANOGRAPHY,* vol. 21, pp. 68-81, 2008.

[2] D. Le Vine, G. S. E. Lagerloef, F. R. Colomb, S. H. Yueh and F. A. Pellerano, ""Aquarius: An Instrument to Monitor Sea Surface Salinity From Space"," *Geoscience and Remote Sensing, IEEE Transactions on,* vol. 45, pp. 2040-2050, 2007.

[3] D. M. Le Vine, G. S. E. Lagerloef, F. Pellerano and F. R. Colomb, ""The Aquarius/SAC-D mission and status of the Aquarius instrument","" *Microwave Radiometry and Remote Sensing of the Environment,* pp. 1-4, 2008.

[4] G. Poe and R. Conway, "A Study of the Geolocation Errors of the Special Sensor Microwave/Imager (SSM/I)," *IEEE Trans. Geosci. Remote Sensing,* vol. 28, no. 5, pp. 791-799, 1990.

[5] W. E. Purdy, P. W. Gaiser, G. A. Poe, E. A. Uliana, T. Meissner and F. J. Wentz, "Geolocation and Pointing Accuracy Analysis for the WindSat Sensor," *IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING,* vol. 44, no. 3, pp. 496-505, 2006.

[6] G. A. Poe, E. A. Uliana, B. A. Gardiner, T. E. vonRentzell and D. B. Kunkee, "Geolocation Error Analysis of the Special Sensor Microwave Imager/Sounder," *IEEE Transatcions on Geoscience and Remote Sensing,* vol. 46, no. 4, pp. 913-922, 2008.

[7]   G. Poe, K. St. Germain, J. Bobak and et al, "DMSP Calibration/Validation Plan for the Special Sensor Microwave Imager Sounder (SSMIS)," Naval Research Laboratory, Washington, DC, USA, 2001.

[8]   C. May, Engineering Evaluation of Multi-Beam Satellite Antenna Boresight Pointing Using Land/Water Crossings, Orlando, FL: University of Central Florida, 2012.

[9]   F. A. Pellerano, J. Piepmeier, M. Triesky, K. Horgan, J. Forgione and J. Caldwell, "The Aquarius Ocean Salinity Mission High Stability L-Band Radiometer," in *Geocsience and Remote Sensing Symposium, 2006, IGARSS 2006, IEEE International Conference*, 2006.

[10] F. Madero, "MWR Processor Geometric ATBD--DS, Tech. Report GS-PMW-DSN-DS-00400-A," CONAE, Cordoba, Argentina, 2010.

[11] J. Galvan, "Posicionamiento absoluto de las bocinas de la Antena MWR"," IA, Buenos Aires, Argentina, SD – MWR – A – DVAG – A001 – A, 2008.

[12] B. Tieu and M. Suarez, "Mission Plan, Tech. Report AS-213-0097, 2009," Buenos Aires, Argentina, 2009.

[13] F. Ulaby, R. K. Moore and A. K. Fung, Microwave Remote Sensing, Active and Passive, Vol I: Microwave Remote Sensing Fundamentals and Radiometry, Addison Wesley Publishing Company, 1981.

[14] Google, "Google Earth," [Online]. Available: http://www.google.com/earth/. [Accessed 05 02 2015].

[15] M. M. Jacob, ""MWR User's Guide - English Version"," CONAE, Córdoba, Argentina, GS-OIV-OPS-GU-00500-A, 2010.

[16] G. Lagerloef, "The Aquarius/SAC-D Mission: Designed to Meet the Remote Sensing Challenge," *Oceanography,* vol. 21, no. 1, pp. 68-81, 2008.