# GAMES

## "Geolocation Assessment/validation Methods for EPS-SG ICI and MWI"

In response to: EUMETSAT ITT 19/218140 "Development of Geolocation Validation Methods for EPS-SG ICI and MWI"

# Final report

Deliverable document D13

prepared by

**F.S. Marzano, M. Papa**
**M. Montopoli, D. Casella**
**B. Rydberg**

**Sapienza University of Rome**
**Center for Research in AeroSpace**
**(SUR-CRAS)**

**Möller Data Workflow Systems AB**
**(Molflow)**

**National Research Council of Italy**
**Institute of Atmospheric Sciences and Climate**
**(CNR-ISAC)**

Date: 26 January 2021

## ABOUT

The GAMES study focuses on the development of a methodology and implementation of algorithms for the quantitative assessment of geolocation error related to the upcoming EPS-SG ICI and MWI spaceborne radiometric sensors.

The Ice Cloud Imager (ICI) is a millimetre and submillimetre-wave conical radiometer imager on board of the forthcoming European Polar System – Second Generation (EPS-SG) satellite. ICI observations are acquired at about 53° incidence angle and within ±65° in azimuth thus providing a swath of about 1700 km at the nominal orbit altitude. ICI has channels around 183 GHz, 243 GHz, 325 GHz, 448 GHz and 664 GHz. The primary objective of the ICI space mission is to support meteo-climate monitoring with a special focus on the retrieval of ice clouds in terms of columnar equivalent water content and effective particle diameter.

## OBJECTIVES

The primary goal of GAMES study is to develop a methodology for the quantitative assessment of geolocation of the ICI spaceborne radiometer. The work investigates ICI channels to look for optically detectable surface targets (e.g., landmarks) as well as and atmospheric features (e.g., water vapor gradients and deep convective clouds) to implement an algorithm for the geolocation error validation. The validation is based on a contour matching technique, using high altitude lakes, ice shelves and mountain chains as landmark targets and water vapour features and deep convective cloud as atmospheric targets.

## OVERVIEW

Geolocation of satellite data is a standard part of the post-launch calibration process. For the data to be of value, it is critical that the measured parameters are correctly mapped to the surface of the Earth. To validate the geolocation error for satellite-based microwave radiometers at 10-50 GHz (outside the absorption bands) it is possible to observe the surface, exploiting the strong difference in terms of surface emissivity between land and ocean. At this frequency the land surface emissivity is in the order of 0.8 (mainly depending on geographical coordinate) and sea surface is about 0.4 (mainly depending on salinity, surface temperature, wind speed and angle of view). The brightness temperature contrast along the coastline is used to extract the shoreline contours from satellite images. Correlating the reference shoreline with the extracted radiometric contour, it is possible to estimate the mutual shift along, obtaining the final geolocation error in km. An alternative way to obtain the coastline is to exploit the difference between ascending and descending swaths. At frequencies beyond 150 GHz, the gaseous absorption increases, and the atmospheric transmittance tends to be nearly zero so that the surface becomes optically invisible. However, some channels can still detect the surface, e.g. in subarctic-winter conditions and in regions with high topography with very low atmospheric water vapour. This is the case for the outermost 183 GHz channel (ICI-1), and for the channel at 243 GHz (ICI-4). As such, the first objective of GAMES is to develop a method for the validation of these channels, identifying specific landmark targets areas.
The further goal of the GAMES study is to quantify the errors in the field of view geolocation of ICI, exploiting meteorological targets. Unfortunately, most of the ICI channels, namely those at frequency equal or greater than 325 GHz, have no chance to sample the surface features due to the strong gas absorption at those frequencies, thus preventing any ground-target based geolocation method. Here is where the meteorological target-based geolocation methods come mainly into play. Two distinct geolocation methods making use of meteorological

targets were studied: absolute and relative geolocation methods. The goal of the absolute geolocation is to estimate the geolocation error of a pivot ICI channel (e.g., one of those around 183 GHz) with respect to external reference information. For the relative geolocation, it is a self- referenced method since it uses ICI channel only without resorting to external auxiliary information and it aims at finding the pointing error of the ICI channels (i.e., those at frequencies above 183 GHZ) with respect to the pivot one. The meteorological targets considered are the deep convective clouds and water vapor masses (or atmospheric rivers) since these two target typologies are expected to be sufficiently detectable by the investigated ICI channels. The effectiveness of the absolute geolocation method is assessed using actual observations from PMW (e.g. SSMI/S, MHS, ATMS) sensors and SEVIRI on board of MSG. GMI and radar information from DPR and Cloudsat radar are also used for verification. The rationale is to have the PMW 183 GHz channel that mimics the 183 GHz ICI channel that needs to be geolocated, whereas MSG channels in the infrared window act as reference. On the other hand, the relative geolocation is assessed using a simulated dataset of four ICI orbits.

A geolocation validation prototype tool is also generated as stand-alone tool, implementing the promising methods based on landmark targets as well as a Backus-Gilbert based relative geolocation assessment technique. The geolocation error assessment prototype is implemented in a standard language, implemented in Python and Cython for computational extensive algorithms. Cython is a superset of the Python language, and a compiled language designed to give C-like performance of code mostly written in Python. The coding is modular, as much generic as possible, clearly readable, commented, and following coding standards (e.g. PEP8). The GAMES tool is portable and is developed so that it can be run within a Docker container, and this container can be transferred to any Docker-enabled machine. Data can easily be mounted into this container. A representative test and validation dataset, over selected regions of the globe spanning a time period that should be equivalent to one month of data is also collected. A detailed Algorithm Theoretical Basis Document (ATBD) is finally prepared. The ATBD describes in detail the methodology and the selection procedures applied.

## CONTACT

**EUMETSAT**
<Dr. Vinia Mattioli>
< Remote Sensing Scientist - ICI Level 1 & 2 >
< Remote Sensing and Product Division >
<EUMETSAT>
<vinia.mattioli@eumetsat.int>

**PROJECT MANAGER (s)**
<Prof. Frank S. Marzano>
<Sapienza University of Rome, Italy>
<frank.marzano@uniroma1.it>

## OTULINE OF THE FINAL REPORT

**Task 1. Final report on landmark target assessment method**
**Task 2. Final report on atmospheric target assessment method**
**Task 3. Final report on assessment method implementation (ATBD, Algorithm Theoretical Baseline Document)**
**Quick guide to GAMES tool v1.1**

# GAMES

## "Geolocation Assessment/validation Methods for EPS-SG ICI and MWI"

In response to: EUMETSAT ITT 19/218140 "Development of Geolocation Validation Methods for EPS-SG ICI and MWI"

Deliverable document 03 – D03

# Final report – Task 1

prepared by

**M. Papa, F.S. Marzano**
**M. Montopoli, D. Casella, B. Rydberg**

**Sapienza University of Rome**
**Center for Research in AeroSpace**
**(SUR-CRAS)**

**Möller Data Workflow Systems AB**
**(Molflow)**

**National Research Council of Italy**
**Institute of Atmospheric Sciences and**
**Climate (CNR-ISAC)**

Date: 11 February 2020

# OUTLINE

## LIST OF ACRONYMS

| | |
|---|---|
| AMSU | Advanced Microwave Sounding Unit |
| ATMS | Advanced Technology Microwave Sounder |
| BT | Brightness Temperature |
| DCC | Deep Convective Clouds geolocation-error assessment method |
| DEM | Digital Elevation Model |
| FNC | Fast Normalized Cross-correlation |
| FR | Full Resolution |
| GRD | Ground Range Detected |
| GSHHG | Global Self-consistent, Hierarchical, High-resolution Geography database |
| HR | High Resolution |
| ICI | Ice Cloud Imager radiometer |
| MHS | Microwave Humidity Sounders |
| MR | Medium Resolution |
| NOAA | National Oceanic and Atmospheric Administration |
| RAOB | RAwinsonde OBservation |
| RFD | Registration in Frequency Domain |
| SAR | Synthetic Aperture Radar |
| SNAP | Sentinel Application Platform |
| SNPP | Suomi National Polar-orbiting Partnership |
| SSMIS | Special Sensor Microwave Imager Sounder |
| SUR | Sapienza University of Rome |
| TCM | Target Contour Matching |
| TELSEM2 | Tool to Estimate Land.Surface Emissivities at Microwave version 2 |
| TESSEM2 | Tool to Estimate Sea Surface Emissivities at Microwave version 2 |
| TOA | Top Of Atmosphere |
| WVS II | Vector Shoreline Data Bank II |

# 1. INTRODUCTION

The EPS-SG Ice Cloud Imager (ICI) is a sub-millimetre wave conical imager on board of the European Polar System – Second Generation (EPS-SG) and it will have 11 channels with frequencies around 183, 243, 325, 448 and 664 GHz, as shown in Tab. 1.1.

*Table 1.1: Summary of ICI channels*

| CHANNEL | FREQUENCY (GHz) | BANDWIDTH (MHz) | NEΔT (K) | BIAS (K) | POLARISATION | FOOTPRINT SIZE AT 3 dB |
|---|---|---|---|---|---|---|
| ICI-1 | 183.31±7.0 | 2x2000 MHz | 0.8 | 1.0 | V | 16 km |
| ICI-2 | 183.31±3.4 | 2x1500 MHz | 0.8 | 1.0 | V | 16 km |
| ICI-3 | 183.31±2.0 | 2x1500 MHz | 0.8 | 1.0 | V | 16 km |
| ICI-4 | 243.2±2.5 | 2x3000 MHz | 0.7 | 1.5 | V, H | 16 km |
| ICI-5 | 325.15±9.5 | 2x3000 MHz | 1.2 | 1.5 | V | 16 km |
| ICI-6 | 325.15±3.5 | 2x2400 MHz | 1.3 | 1.5 | V | 16 km |
| ICI-7 | 325.15±1.5 | 2x1600 MHz | 1.5 | 1.5 | V | 16 km |
| ICI-8 | 448±7.2 | 2x3000 MHz | 14 | 1.5 | V | 16 km |
| ICI-9 | 448±3.0 | 2x2000 MHz | 1.6 | 1.5 | V | 16 km |
| ICI-10 | 448±1.4 | 2x1200 MHz | 2.0 | 1.5 | V | 16 km |
| ICI-11 | 664±4.2 | 2x5000 MHz | 1.6 | 1.5 | V, H | 16 km |

These wavelengths allow to detect ice clouds, whereas the emission signal from the surface is predominantly masked by high water vapour opacity. The latter is a problem for the geolocation assessment because current methods, comparing coastlines in imagery data with the known geographic locations, are not readily applicable to ICI channels. However, in very dry atmospheric conditions the geolocation validation technique could be still based on the observations at 183.3±7 GHz and 243.2 GHz.

## 1.1 Literature review

A wide experience has been accumulated so far on the geolocation error validation for satellite-based microwave radiometers at lower microwave frequencies (10-50 GHz, outside the absorption bands) by exploiting their strong difference in terms of surface emissivity between land and ocean. Global-scale coastlines can be used as surface landmarks with a significant contrast in terms of measured brightness temperature (BT). Comparing the latter with a reference coastline database [1], it is possible to assess the spaceborne sensor geolocation error.

In [2] Purdy et al. the shoreline obtained from WindSat satellite imagery and the World Vector Shoreline Data Bank II (WVS II) is compared. The position of the coastline is obtained taking the peak of the first derivative of radiometric data along scan and cross scan direction, after a cubic spline interpolation to obtain a more smoothed curve. Poe et al. [3] apply a similar method on Special Sensor Microwave Imager/Sounder (SSMIS) using data provided by spacecraft F-16.

In **[4]** Heygster et al. exploit the fact that, when geolocation errors are present, the projected footprints have different shifts considering ascending or descending swaths. Since the brightness temperature (BT) differences between ascending or descending swaths are higher along coastlines, they evaluate the geolocation error using data from AMSR-E at 89 GHz. Berg et al. [5] use the BT difference between ascending and descending swaths to obtain the attitude error for SSM/I spacecraft. Finally, Moradi et al. **[6]** correct the pitch, yaw and roll angles for Advanced Microwave Sounding Unit (AMSU) and Microwave Humidity Sounders (MHS) minimizing the difference in brightness temperature between ascending and descending swathes.

Along the coastline, the measured signal consists of radiation received from both land and water surfaces and Bennartz [7] proposed to use a high-resolution land–sea mask to infer the fraction of water surface for each measurement. He has developed a method to validate the geolocation accuracy using the convolution of land-sea masks that is suitable to apply for channels that are sensitive to land/sea contrast. Han et al. [8] adapted this so-called "Land/sea Fraction Method" for the NOAA 16-18 satellites and also for ATMS on SNPP.

Several algorithms can be applied to extract contour from images, starting from the simplest and faster to the most sophisticated, but with higher computational costs. In the following work we have focused on the Canny edge detector [9], because it is a fast algorithm that is able to detect both strong and weak edges [10], whereas its accuracy is slightly better than other algorithms [11], [12]. In addition we have also used the Sobel filter [13] as it is a fast approximation of image gradient [14].

## 1.2 Organization of the report

This Task-1 report is organized as follows.

Section 2 contains general information about the proposed methodology, including the criteria regarding the search of landmark targets and the cloud-masking defuzzification step to filter the available dataset from cloud coverage contamination.

Section 3 lists the 9 selected targets, among the list of those which have been explored, by dividing the list for the northern and southern hemisphere in order to guarantee a good temporal coverage during the driest seasons. For each target this section resumes the results, using data from SSMIS channel at 183 GHz, in terms of mean value and standard deviation of the geolocation error both in northern hemisphere and southern hemisphere.

Section 4 approaches the problem of the sensitivity analysis of the proposed geolocation error assessment methodology to the most critical free parameters as a proxy to the error budget estimate. The latter, as a matter of fact, is not easily defined for the lack of an absolute reference (we are here estimating not the geolocation error but its accuracy or the error of the geolocation error correction procedures).

Section 5 aims at evaluating how the analysis, carried out using SSMIS channel at 183 GHz, can be extended to ICI channels at 183 GHz, 243 GHz and beyond if possible. The discussion uses both an analysis of SSMIS imagery at 150 GHz and 183 GHz and a radiative transfer simulation of brightness temperatures and slant-path attenuation from available radiosounding profiles and from ERA5 reanalysis atmospheric profiles near the selected targets.

Section 6 draws the main conclusions and paves the way to the geolocation error assessment for ICI channels using the landmark targets.

The six final appendixes discuss some details about the geolocation error assessment methodology and technical analysis of target classes, such as high-latitude lakes, mountain ranges and ice shelves. In particular, the appendixes are devoted to target contour matching (TCM) approach for high-altitude lake targets, TCM approach for mountain-chain targets, TCM approach for ice-shelf targets, Contour extraction and cross-correlation techniques for TCM, Parallax error correction and Threshold selection for cloud-masking fuzzy-logic algorithm.

# 2. ASSESSMENT METHODOLOGY USING LANDMARK TARGETS

The GAMES methodology for the validation of the geolocation error consists of some steps described in the following paragraphs.

## 2.1 Target-contour matching block diagram and data flow

Within the GAMES project, the proposed target-contour matching (TCM) algorithm is shown in the following block diagram. The proposed scheme generalizes the conventional geolocation assessment method because it has some differences according to the selected target and used reference.

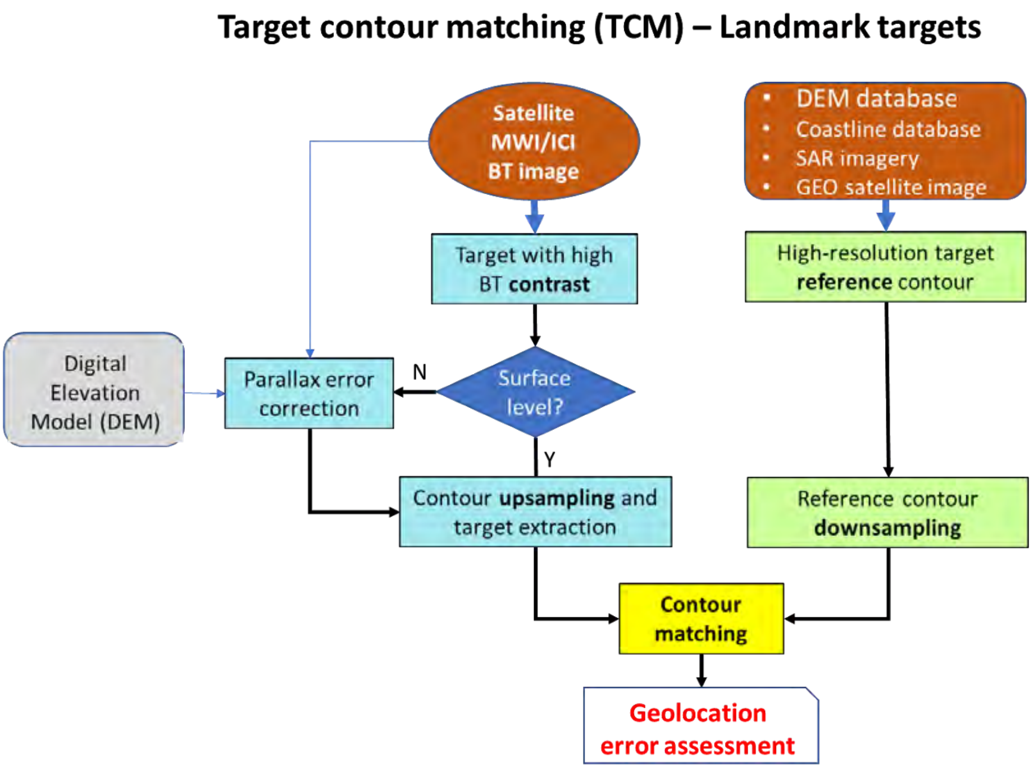



*Figure 2.1.1: Logical scheme of proposed methodology to validate the geolocation using landmark targets (target-contour matching algorithm).*

The inputs to the GAMES methodology are the satellite radiometric imagery containing the landmark target, highlighted by the blue ellipse. To select target images with a sufficient BT contrast to extract a contour, we use a fuzzy-logic approach, as described in subsection 2.3.

Following the left branch in Fig. 2.1.1, there is a control whether the target is at sea level or not, because the satellite data are projected on terrestrial ellipsoid, as shown by point A in Fig. 2.1.2. If the target is located above sea-level, the line of sight intercepts the Earth at point C so that the corrected coordinates are those of point B. This step represents the parallax error correction and the Digital Elevation Model (DEM) is used to find the intersection between the satellite line of sight and the orography.

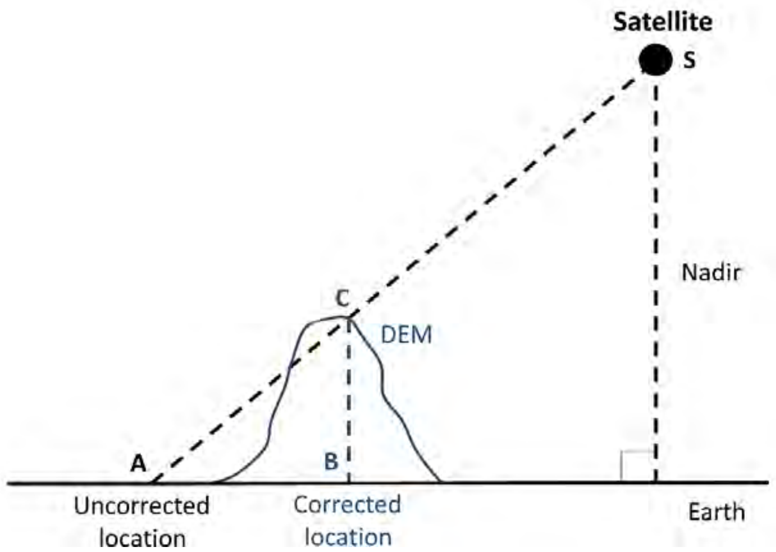



*Figure 2.1.2: Digital Elevation Model correction problem*

Satellite microwave radiometric images have a low spatial resolution, e.g. SSMIS F17 has about 13 km of spatial resolution. In order to fictitiously increase the resolution of a BT scenario, data are generally interpolated using a cubic interpolation, obtaining a finer spatial resolution than the nominal one. The sensitivity to the parameters of this arbitrary step is discussed in subsection 4.1 and subsection 4.2.

The next step of the target-contour matching algorithm is the extraction of a contour that can be carried out by applying (e.g. see Appendix A for details):

- Canny approach [9] to obtain a line;
- Sobel filter [13] to obtain an image gradient map.

The extracted contour can be cross-correlated with a reference to estimate the geolocation accuracy using the normalized cross-correlation function $\gamma(u,v)$. Picking the maximum of $(u,v)$ it is possible to obtain the lat-lon pixel displacements, that can be converted into shifts along x and y direction. In order to have an accuracy of about 0.1 pixel, the maximum is fitted with a polynomial of 4th order. From these pixel displacements it is possible to obtain the related latitude and longitude error and the corresponding distance error in km. An alternative way to obtain directly a displacement with sub-pixel accuracy is to use the Fast Normalized Cross-correlation (FNC) technique [16].

As mentioned, the reference contour can be different depending on the type of landmark target. The following Tab. 2.1.1 summarizes the different possible sources of contour references.

*Table 2.1.1: Summary of proposed reference*

| Reference source | Original source | Spatial resolution | Pre-processing needed |
|---|---|---|---|
| SAR | Level-1 GRD | 10-40 m | yes |
| GSHHG | GSHHG with full resolution | 40 m | no |
| DEM | GTOPO 30 | 30 arc seconds (~1 km) | no |

As shown in Tab. 2.1.1 we can have 3 different types of reference source, showing their original spatial resolution. In particular, for SAR images some preliminary pre-processing is necessary before using them as reference, as it is explained in Appendix C.

## 2.2 Criteria for landmark target selection

Considering homogeneous isothermal (constant temperature and interaction parameters) atmospheric layer of thickness $H$ with a small albedo (thus neglecting the multiple scattering contribution), it is possible to derive the analytical solution of the radiative transfer equation for the upwelling BT as follows:

$$T_B = e_s\, T_s\, e^{-(k_e L)} + (1-w)T_0[1-e^{-(k_e L)}] \tag{2.2.1}$$

where the symbols are
$e_s$ : *surface emissivity* (adim.)
$T_s$ : *surface temperature* (K)
$T_0 = T(z)$ : *constant atmospheric temperature* (K)
$k_e$ : *atmospheric extinction coefficient* ($km^{-1}$)
$w$ : *atmospheric albedo* (adim.)
$L$ : *atmospheric slant path* (*H = Lcos*θ *with* θ *the nadir angle*) (km)
$e^{-(k_e L)}$ =*t(L): atmospheric transmittance* (adim.)

Considering two different close pixels $p_1$ and $p_2$ and assuming a similar atmospheric layer with the same transmittance *t*(*L*), the BT contrast $\Delta T_B = T_B(p_1) - T_{B2}(p_2) = T_{B1} - T_{B2}$ can be written as follows:

$$\Delta T_B = e_{s_1}\, T_{s_1} e^{-(k_e L)} + (1-w)T_0[1-e^{-(k_e L)}] - e_{s_2}\, T_{s_2} e^{-(k_e L)} - (1-w)T_0[1-e^{-(k_e L)}] \tag{2.2.2}$$

thus

$$\Delta T_B = t(L)\,[e_{s_1}\, T_{s_1} - e_{s_2}\, T_{s_2}] \tag{2.2.3}$$

Therefore, in order to have a sufficiently high BT contrast, from eq. (2.2.3) we can essentially consider areas with different surface emissivity and/or surface temperature, such as sea/lake/ice coastlines or mountain chains. In the latter case we have a surface temperature variability due to the height difference between plain and mountain as well as a different atmospheric optical thickness (i.e., transmittance of the mountain pixel larger than the plain one) entailed by the different heights of the pixels themselves. A further feature to play with is the natural variability of surface emissivity.

Taking into account these concepts, for landmark target search we have basically considered the following two major types:

a) surface water bodies (liquid or ice) sufficiently large (wrt satellite FOV);
b) mountain areas with strong slopes (altitude gradients) in relatively dry regions.

## 2.3 Fuzzy-logic approach to target cloud-masking

The cloud-masking analysis was performed to improve the selection of useful satellite overpasses over the selected test sites, that is in this case those SSMIS overpasses where cloud coverage may cause a larger atmospheric opacity at 183 GHz. Along a period of one year there are several satellite passes, depending on the latitude and swath width, but these can be included in the geolocation error assessment analysis in the case of:

- clear sky or limited cloudy conditions in the selected areas;
- very dry conditions with little amount of water vapour;
- enough BT contrast of the target to extract a contour.

The approach under investigation for developing, as much as possible, an automated procedure for the selection of instrument overpasses of the target area consist in using a relatively simple fuzzy-logic approach to select only useful passes, as shown in in Fig. 2.3.1.

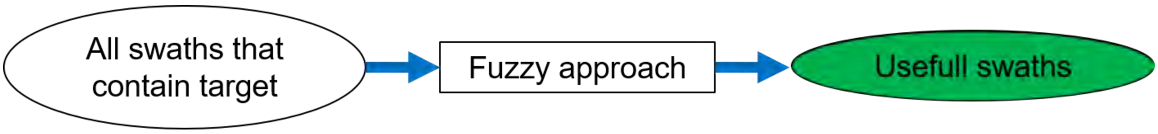
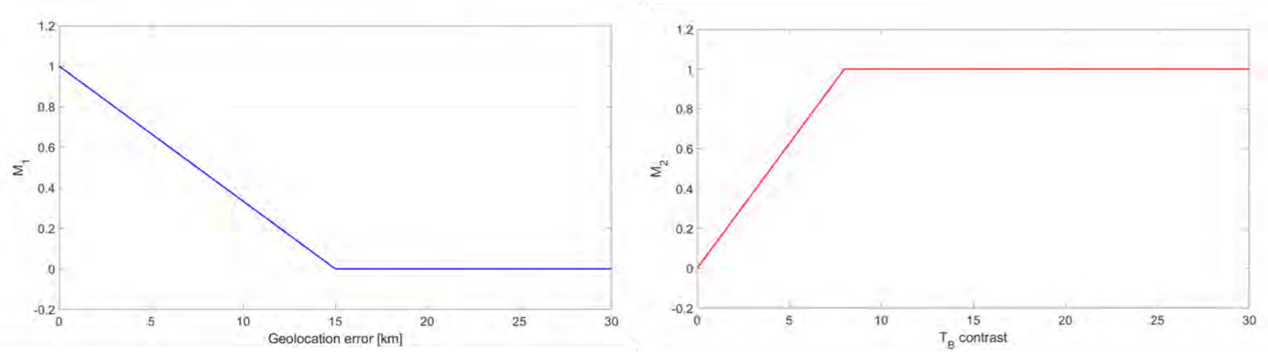



*Figure 2.3.1: Fuzzy-logic approach to cloud masking*

In the proposed fuzzy-logic approach, the idea is to use the estimated geolocation error and brightness temperature contrast of a specific target to understand if an overpass can be correctly used. For this purpose we use the membership functions $M_1$ and $M_2$, shown in Fig. 2.3.2. If the geolocation error is greater than a selected threshold or the BT contrast is lower than a specific threshold, the membership functions are linearly weighted. Thresholds are, to some extent, arbitrarily or empirically defined mainly depending on the channel spatial resolution at ground.




$$M_1 = \begin{cases} linear & Error \leq Error_{threshold} \\ 0 & Error > Error_{threshold} \end{cases} \qquad M_2 = \begin{cases} linear & \Delta T_B \leq \Delta T_{B\ threshold} \\ 1 & \Delta T_B > \Delta T_{B\ threshold} \end{cases}$$

*Figure 2.3.2: Proposed function for fuzzy approach*

After the definition of the membership functions $M_1$ and $M_2$, the inference function $I(x)$ is constructed by a multiplicative rule of the 2 membership functions:

$$I(x_1, x_2) = M_1(x_1)M_2(x_2) \tag{2.3.1}$$

where $x_1$ and $x_2$ are arbitrary variables. Finally, an image can be used to evaluate the geolocation error if it satisfies the following defuzzification step:

$$I(x_1, x_2) \geq I_{threshold} \tag{2.3.2}$$

where $I_{threshold}$ is typically set to 0.3 for all targets (see Appendix A, B and C). After a sensitivity analysis over the whole available dataset, the proposed inference function is:

$$I\left(\Delta T_{Bm}, \varepsilon\right) = M_1(\varepsilon) M_2(\Delta T_{Bm}) \tag{2.3.3}$$

where:

- $I(x)$ = inference function
- $\Delta T_{Bm}$ = mean BT contrast around target
- $\varepsilon$ = geolocation error
- $M_1(\varepsilon)$ = membership function depending on the geolocation error
- $M_2\left(\Delta T_{Bm}\right)$ = membership function depending on the BT contrast

The membership functions *M* and their parameters are provided in the Appendix A for high-altitude lake targets, in Appendix B mountain-chain targets and in Appendix for ice-shelf targets.

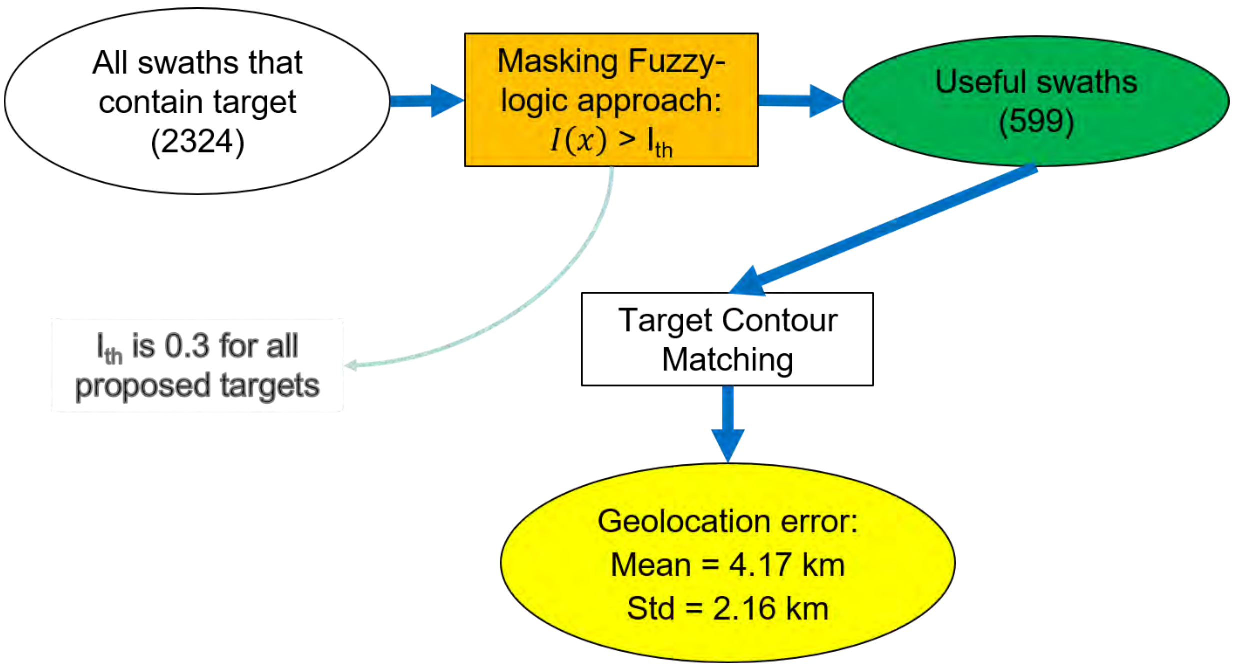



*Figure 2.3.3: Logical scheme to evaluate the geolocation error starting from all swaths along time period. Numbers refer to the example of the Ross ice shelf target (see text).*

As an example of the possible impact of this fuzzy-logic approach, we have considered the Ross ice shelf as a target during 2016. Fig. 2.3.3 shows the block diagram that allows evaluating the geolocation error starting from all available swaths during the selected one-year period.

This block diagram can, in principle, be extended to all satellite channels and all targets of interest, by properly setting the free parameters in the fuzzy-logic approach expressed by previous equations. Within 2016, there are 5118 SSMIS swaths, but only 2324 overpass the target. By applying the fuzzy-logic approach to select useful passes, the total number of passes has been reduced to 999. The overall mean geolocation error has been estimated to be about 4.5 km with a standard deviation of 2.1 km.

# 3. TARGET SELECTION AND GEOLOCATION ERROR ASSESSMENT

This section is devoted to landmark target selection and geolocation error assessment.

## 3.1 List and features of landmark targets

The following section contains the description of the results for all target agreed with EUMETSAT and listed in the following:

**Northern hemisphere**

- Qinghai lake
- Karakorum mountains
- Hudson Bay
- Nares Strait

**Southern hemisphere**

- Ross Antarctic ice shelf
- Filchner-Ronne Antarctic ice shelf
- Amery Antarctic ice shelf
- Titicaca lake
- Andean mountains

For the geolocation assessment test we use SSMIS F17 speceborne radiometer data. Its main specifications are reported in Fig. 3.2.1. In particular, considering the ICI application, we have selected the 183±6.6 GHz channel in horizontal polarization, downloaded from the following web site:

https://www.ncdc.noaa.gov/has/HAS.FileAppRouter?datasetname=CSU_SSMIS&subqueryby=STATION&applname=&outdest=FILE

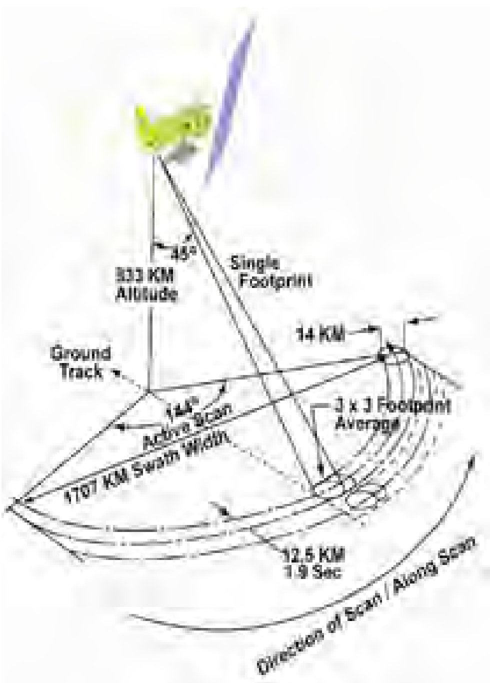



*Figure 3.1.1a: SSMIS characteristics*

Radiometric characteristics of the SSMIS (H: horizontal; V: vertical; RC: right circular).

| Frequency (GHz) | Polarization (V, H, and RC) | Along-track resolution (km) | Cross-track resolution (km) | Spatial sampling (km×km) | Instrument noise (K) |
|---|---|---|---|---|---|
| 19.35 | H, V | 73 | 47 | 45×74 | 0.35 |
| 22.235 | V | 73 | 47 | 45×74 | 0.45 |
| 37.0 | H, V | 41 | 31 | 28×45 | 0.22 |
| 50.3 | H | 17.6 | 27.3 | 37.5×37.5 | 0.34 |
| 52.8 | H | 17.6 | 27.3 | 37.5×37.5 | 0.32 |
| 53.596 | H | 17.6 | 27.3 | 37.5×37.5 | 0.33 |
| 54.4 | H | 17.6 | 27.3 | 37.5×37.5 | 0.33 |
| 55.5 | H | 17.6 | 27.3 | 37.5×37.5 | 0.34 |
| 57.29 | RC | 17.6 | 27.3 | 37.5×37.5 | 0.41 |
| 59.4 | RC | 17.6 | 27.3 | 37.5×37.5 | 0.40 |
| 63.283248±0.285271 | RC | 17.6 | 27.3 | 75×75 | 2.7 |
| 60.792668±0.357892 | RC | 17.6 | 27.3 | 75×75 | 2.7 |
| 60.792668±−0.357892±0.002 | RC | 17.6 | 27.3 | 75×75 | 1.9 |
| 60.792668±0.357892±0.005 | RC | 17.6 | 27.3 | 75×75 | 1.3 |
| 60.792668±0.357892±0.016 | RC | 17.6 | 27.3 | 75×75 | 0.8 |
| 60.792668±0.357892 ±0.050 | RC | 17.6 | 27.3 | 75×75 | 0.9 |
| 91.665 | H, V | 14 | 13 | 13×16 | 0.19 |
| 150 | H | 14 | 13 | 13×16 | 0.53 |
| 183.311±1 | H | 14 | 13 | 13×16 | 0.38 |
| 183.311±3 | H | 14 | 13 | 13×16 | 0.39 |
| 183.311±6.6 | H | 14 | 13 | 13×16 | 0.56 |

*Figure 3.1.1b: SSMIS channels description*

For each target we provide further information about its features in the following tables. The variability of the detection per day is mainly due to the latitude of the landmark targets (near-polar targets are observed with a higher repetitivity).

*Table 3.1.1: Summary of proposed targets, reference source and daily detectability*

| **Landmark target** | **Contour reference source** | **Detectability/day** |
|---|---|---|
| **Northern hemisphere** | | |
| Qinghai lake | GSHHG | 1 |
| Karakorum mountains | DEM | 1 |
| Hudson Bay | GSHHG | 1 |
| Nares Strait | SAR | 4-6 |
| **Southern hemisphere** | | |
| Ross Antarctic ice shelf | SAR | 4-6 |
| Filchner-Ronne Antarctic ice shelf | SAR | 4-6 |
| Amery Antarctic ice shelf | SAR | 3-5 |
| Titicaca lake | GSHHG | 1 |
| Andean mountains | DEM | 1 |

Along 2016, we have 5118 swaths and the Tab. 3.1.2 reports the all available samples for each target before and after the defuzzification step. In particular, the samples after this step represent the used dataset to validate the geolocation accuracy for each target.

*Table 3.1.2: Summary of proposed targets, reference source and daily detectability within the year 2016*

| Landmark target | SSMIS samples | Cloud-masked samples | Cloud-masked samples (percentage) |
|---|---|---|---|
| **Northern hemisphere** | | | |
| Qinghai lake | 629 | 84 | 13.4 % |
| Karakorum mountains | 707 | 51 | 7.2 % |
| Hudson Bay | 454 | 62 | 13.6 % |
| Nares Strait | 2140 | 560 | 26.2 % |
| **Southern hemisphere** | | | |
| Ross Antarctic ice shelf | 2324 | 599 | 25.7% |
| Filchner-Ronne Antarctic ice shelf | 2335 | 387 | 16.6% |
| Amery Antarctic ice shelf | 1244 | 153 | 12.3% |
| Titicaca lake | 532 | 41 | 7.7% |
| Andean mountains | 555 | 125 | 22.5% |

## 3.2 Qinghai lake in the northern hemisphere

Qinghai lake is a large lake in China with a surface of about 4500-6000 km$^2$ and an altitude of about 3200 m (see Fig 3.2.1). Its maximum length is about 105 km and the width is about 65 km so that it is larger with respect to SSMIS pixel at 183 GHz and it is possible to successfully detect it and extract its contour.

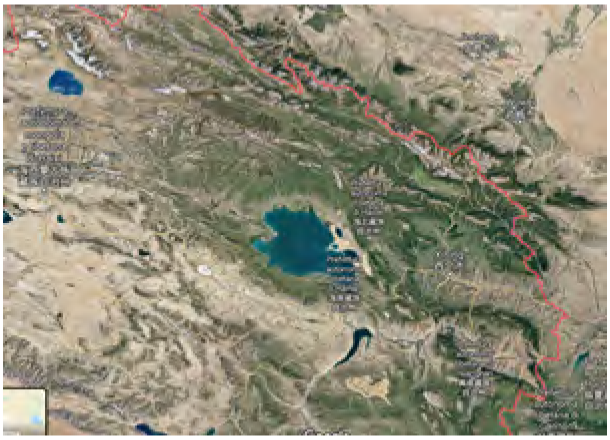


*Figure 3.2.1: Qinghai lake from Google Maps*

We have focused on the following box:

- Latitude = [36.2 ; 37.7];
- Longitude = [99.3 ; 101.0];

Tab. 3.2.1 shows the coordinates for the five points used to calculate the BT contrast following the eq. A.2.

The description of the proposed geolocation assessment methodology for this target class is provided in the Appendix A. Note that the mean BT contrast $\Delta T_B$ is derived from the following Eq. 3.2.1 using points in Tab. 3.2.1:

$$\Delta T_{Bm} = \frac{(T_B - T_A) + (T_C - T_A) + (T_D - T_A) + (T_E - T_A)}{4} \quad (3.2.1)$$

*Table 3.2.1.: Coordinates of five points used to calculate the BT contrast for Qinghai lake following the eq. A.2.*

| **Points** | **Latitude** [deg] | **Longitude** [deg] |
|---|---|---|
| A | 36.9500 | 100.1793 |
| B | 37.3719 | 100.1793 |
| C | 36.9500 | 100.7655 |
| D | 36.5750 | 100.1793 |
| E | 36.9500 | 99.5345 |

Fig. 3.2.2 shows two examples for Qinghai lake with SSMIS F17 at 183±6.6 GHz H.

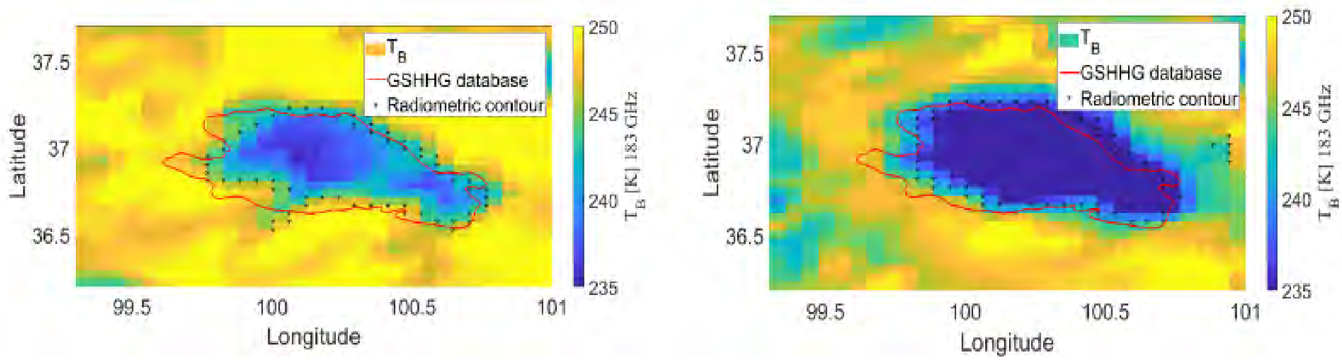



*Figure 3.2.2: Brightness temperature image at 183±6.6 GHz H over Qinghai lake with SSMIS F17 on 2016/12/01 (left) and 2016/12/02 (right). The red line represents the GSHHG shoreline database and black markers are provided by Canny edge detection from the radiometric image.*

The following Tab. 3.2.2 describes the number of samples in which the lake is visible from SSMIS F17 at 183 ± 6.6 GHz during 2016 applying the cloud-masking algorithm.

*Table 3.2.2.: Number of visible days during 2016 for Qinghai lake applying the cloud-masking algorithm*

| Jan | Feb | Mar | Apr | May | Jun | Jul | Aug | Sep | Oct | Nov | Dec | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 5 | 9 | 18 | 10 | 3 | 0 | 1 | 0 | 8 | 30 | 35 | 129 |

From Tab. 3.2.2 the total number of SSMIS available images in 2016 is 84 with geolocation assessment results summarized in Tab. 3.2.3.

*Table 3.2.3: Result for Qinghai lake in 2016*

| Geolocation assessment parameter | Value [km] |
|---|---|
| Geolocation accuracy average [km] | 5.10 |
| Geolocation accuracy standard deviation [km] | 2.03 |

## 3.3 Karakorum mountains in the northern hemisphere

It is a large mountain range spanning the borders between Pakistan, India and China with the northwest extremity of the range extending to Afghanistan and Tajikistan, as shown in Fig. 3.3.1.


*Figure 3.3.1: Karakorum mountains from Google Maps*

Its range is about 500 km, so we have focused on the following sub-box:

- Latitude = [35.5 ; 38.5];
- Longitude = [76.0 ; 80.0];

The BT contrast for Karakorum mountains is derived following the Eq. 3.3.1 with points in Tab. 3.3.1

$$\Delta T_{Bm} = \frac{(T_B - T_A) + (T_D - T_C) + (T_F - T_E) + (T_H - T_G)}{4} \qquad (3.3.1)$$

*Table 3.3.1.: Coordinates of eight points used to calculate the BT contrast for Karakorum mountains following the eq. B.2.*

| **Points** | **Latitude** [deg] | **Longitude** [deg] |
|---|---|---|
| A | 36.8385 | 76.3944 |
| B | 37.5308 | 76.3944 |
| C | 36.6538 | 77.4648 |
| D | 37.2077 | 77.4648 |
| E | 36.3769 | 78.5915 |
| F | 37.0692 | 78.5915 |
| G | 36.1462 | 79.6056 |
| H | 36.7462 | 79.6056 |

Fig. 3.3.2 shows an example for Karakorum mountains with SSMIS F17 at 183±6.6 GHz H.

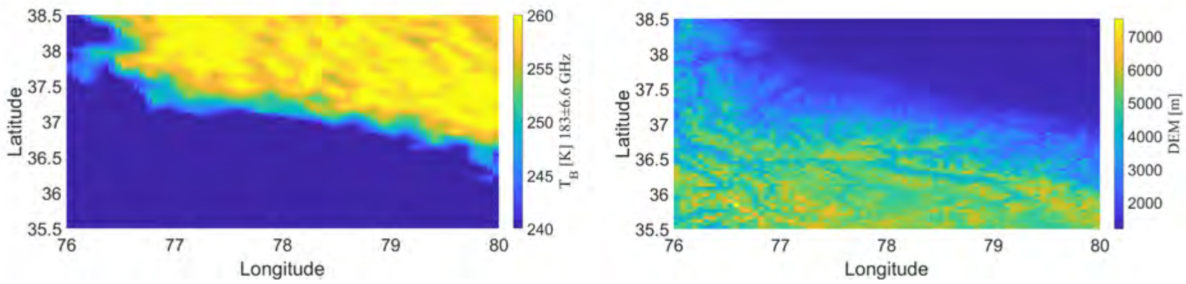



*Figure 3.3.2: In the left there is the brightness temperature image with SSMIS F17 at 183±6.6 GHz H over Karakorum mountains on 2016/10/19 In the right there is the reference digital elevation model.*

The description of the proposed geolocation assessment methodology for this target class is provided in the Appendix B.

*Table 3.3.2: Number of visible days during 2016 for Karakorum mountains applying the cloud-masking algorithm*

| **Jan** | **Feb** | **Mar** | **Apr** | **May** | **Jun** | **Jul** | **Aug** | **Sep** | **Oct** | **Nov** | **Dec** | **Total** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 36 | 34 | 36 | 26 | 24 | 6 | 8 | 1 | 20 | 36 | 37 | 38 | 302 |

From Tab. 3.3.2 the total number of SSMIS available images in 2016 is 51 with geolocation assessment results summarized in Tab. 3.3.3.

*Table 3.3.3: Result for Karakorum mountains in 2016*

| Geolocation assessment parameter | Value [km] |
|---|---|
| Geolocation accuracy average [km] | 4.47 |
| Geolocation accuracy standard deviation [km] | 1.86 |

## 3.4 Hudson Bay in the northern hemisphere

It is a large body of saltwater in northeastern Canada with a surface area of 1,230,000 km$^2$, as shown in Fig. 3.4.1.


*Figure 3.4.1: Hudson Bay from Google Maps*

We have focused on the following box:

- Latitude = [ 56.0 ; 62.0];
- Longitude = [-96.0 ; -87.0];

Tab. 3.4.1 shows the coordinates for the eight points used to calculate the BT contrast for Hudson Bay following the Eq. 3.4.1 and points in Tab. 3.4.1

$$\Delta T_{Bm} = \frac{(T_B - T_A) + (T_D - T_C) + (T_F - T_E) + (T_H - T_G)}{4} \qquad (3.4.1)$$

*Table 3.4.1.: Coordinates of eight points used to calculate the BT contrast for Hudson Bay following the eq. B.2.*

| Points | Latitude [deg] | Longitude [deg] |
|---|---|---|
| A | 61,4091 | -94,8545 |

| | | |
|---|---|---|
| B | 61.4091 | -92,8091 |
| C | 59.1364 | -95,4273 |
| D | 59.1364 | -93,7909 |
| E | 57.3182 | -92,9727 |
| F | 57.7727 | -91,9909 |
| G | 56.4091 | -89,1273 |
| H | 57.0909 | -88,7182 |

Fig. 3.4.2 shows two examples for Hudson Bay with SSMIS F17 at 183±6.6 GHz H.

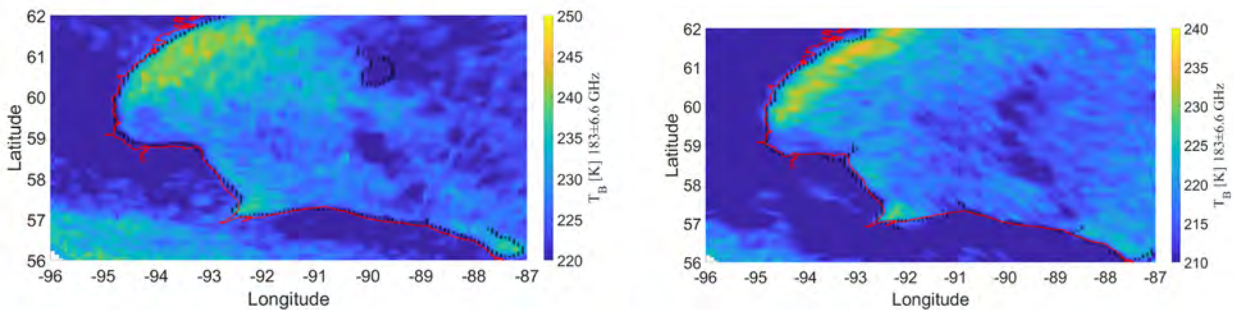



*Figure 3.4.2: Brightness temperature image at 183±6.6 GHz H over Hudson Bay with SSMIS F17 on 2016/01/27 (left) and 2016/02/11 (right). The red line represents the GSHHG shoreline database and black markers are provided by Canny edge detection from radiometric image.*

The description of the proposed geolocation assessment methodology for this target class is provided in the Appendix A.

The following Tab. 3.4.2 describes the number of samples in which the lake is visible from SSMIS F17 at 183 ± 6.6 GHz during 2016 applying the cloud-masking algorithm.

*Table 3.4.2: Number of visible days during 2016 for Hudson Bay applying the cloud-masking algorithm*

| **Jan** | **Feb** | **Mar** | **Apr** | **May** | **Jun** | **Jul** | **Aug** | **Sep** | **Oct** | **Nov** | **Dec** | **Total** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 28 | 38 | 14 | 14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 41 | 135 |

From Tab. 3.4.2 the total number of SSMIS available images in 2016 is 62 with geolocation assessment results summarized in Tab. 3.4.3.

*Table 3.4.3: Result for Hudson Bay in 2016*

| Geolocation assessment parameter | Value [km] |
|---|---|
| Geolocation accuracy average [km] | 5.28 |
| Geolocation accuracy standard deviation [km] | 2.56 |

## 3.5 Nares Strait in the northern hemisphere

Fig. 3.5.1 shows the Nares Strait, that is a waterway between Ellesmere Island and Greenland that connects the northern part of Baffin Bay with the Lincoln Sea (see Fig. 3.5.1).

*Figure 3.5.1: Nares Strait from Google Maps*

Fig. 3.5.2 reports an example of radiometric image in the left and SAR image in the right.

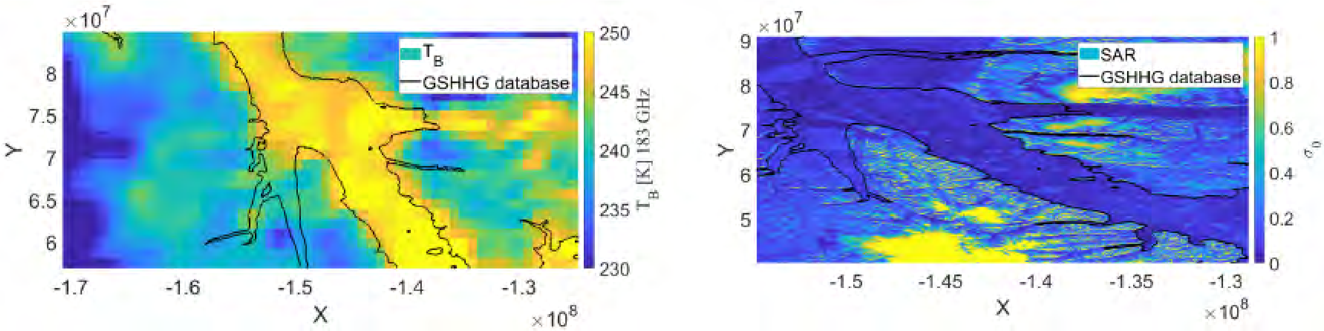



*Figure 3.5.2: Brightness temperature image at 183±6.6 GHz H over Nares Strait from SSMIS F17 on 2016/11/02 (left), the black line is provided by GSHHG shoreline database. SENTINEL 1 IW-GRD on 2017/01/12 (right)*

We have focused on the following box:

- Latitude = [ 80.0 ; 82.5];
- Longitude = [-66.0 ; -63.0];

The description of the proposed geolocation assessment methodology for this target class is provided in the Appendix B with two main differences:

- instead of the DEM gradient that is used as reference for mountain chain, in the case of Nares Strait we proposed to use the spatial horizontal gradient of the SAR image as reference. Therefore to calculate the geolocation accuracy we correlated the gradient of the BT with the gradient of SAR image.

- since we have a complex variable orography, in order to mitigate the "orographic noise" in the cross-correlation between BT and SAR gradient, the following inference function is adopted:

$$I(\varepsilon) = M_1(\varepsilon) M_2(\varepsilon) \tag{3.5.1}$$

where:

- $I(\varepsilon)$ = inference function
- $\varepsilon$ = geolocation error
- $M_1(\varepsilon)$ = term that considers the geolocation error with 5 km of spatial resolution
- $M_2(\varepsilon)$ = term that considers the geolocation error with 6 km of spatial resolution

The following Tab. 3.5.1 describes the number of samples in which the lake is visible from SSMIS F17 at 183 ± 6.6 GHz during 2016 applying the cloud-masking algorithm.

*Table 3.5.1: Number of visible days during 2016 for Nares Strait applying the cloud-masking algorithm*

| **Jan** | **Feb** | **Mar** | **Apr** | **May** | **Jun** | **Jul** | **Aug** | **Sep** | **Oct** | **Nov** | **Dec** | **Total** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 36 | 55 | 38 | 22 | 67 | 12 | 35 | 7 | 44 | 110 | 102 | 59 | 587 |

From Tab. 3.5.1 the total number of SSMIS available images in 2016 is 688 with geolocation assessment results summarized in Tab. 3.5.2.

*Table 3.5.2: Result for Nares Strait in 2016*

| **Geolocation assessment parameter** | **Value [km]** |
|---|---|
| Geolocation accuracy average [km] | 4.74 |
| Geolocation accuracy standard deviation [km] | 2.02 |

### 3.6 Antarctic ice shelves in the southern hemisphere

Antarctic ice shelves are thick suspended platforms of ice that forms where a glacier or ice sheet flows down to a coastline and onto the ocean surface. The ice covers the ground (grounded ice) and it extends into the ocean, with the lower part that detaches from the ground (grounding line), creating a suspended platform of ice (ice shelf), as shown in Fig. 3.6.2.


*Figure 3.6.1: Antarctic ice shelves*

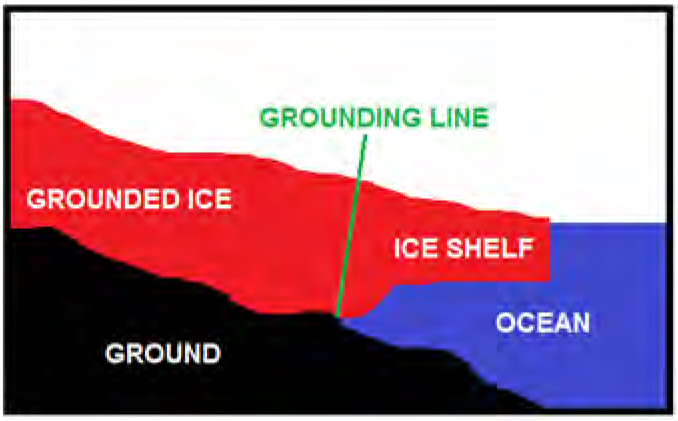



*Figure 3.6.2: Scheme of shelves coastline*

For this type of landmark target, we propose to use SAR images as reference, because they have very high spatial resolution (10-40 m). The adopted SAR data are the Level-1 Ground Range Detected (GRD) products, those consist of focused SAR data that has been detected, multi-looked and projected to ground range using an Earth ellipsoid model. The resulting product has approximately square spatial resolution pixels and square pixel spacing with reduced speckle and three possible spatial resolution: Full Resolution (FR), High Resolution (HR), Medium Resolution (MR). See Appendix C for more information on used SAR data.

#### 3.6.1 Ross ice shelf

The Ross ice shelf is the largest ice shelf of Antarctica (as of 2013 an area of roughly 500,809 km$^2$ and about 800 km across.

For this target we use a SAR image, as shown in Fig. 3.6.1.2. SAR data are preprocessed, as explained in Appendix C. As explained in Appendix C, at polar areas it is preferable to use a polar stereographic projection to reduce distortion in the image. Fig. 3.6.1.3 shows the same SAR image contained in the Fig. 3.6.1.2 projected in this reference system. The red markers show the contour obtaining with Canny algorithm on SAR image.

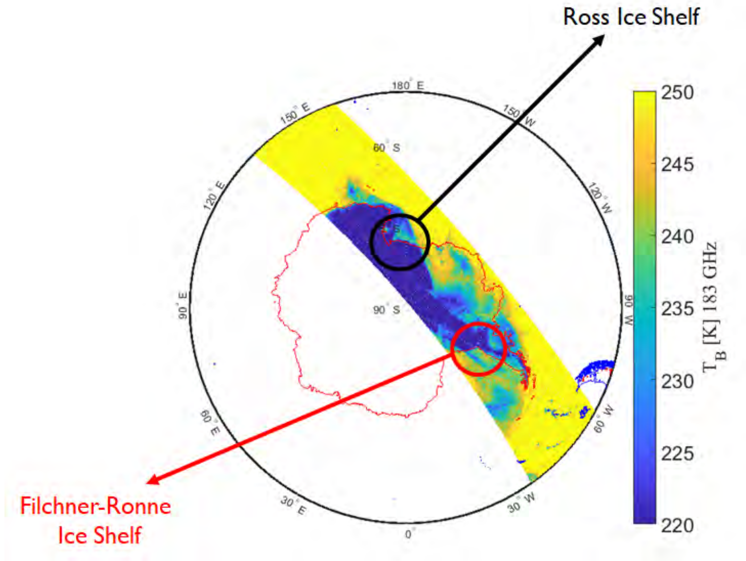



*Figure 3.6.1.1: BT at 183±6.6 GHz H from F17 SSMIS. The black circle indicates the Ross ice shelf and the red highlights the Filchner.Ronne ice shelf.*

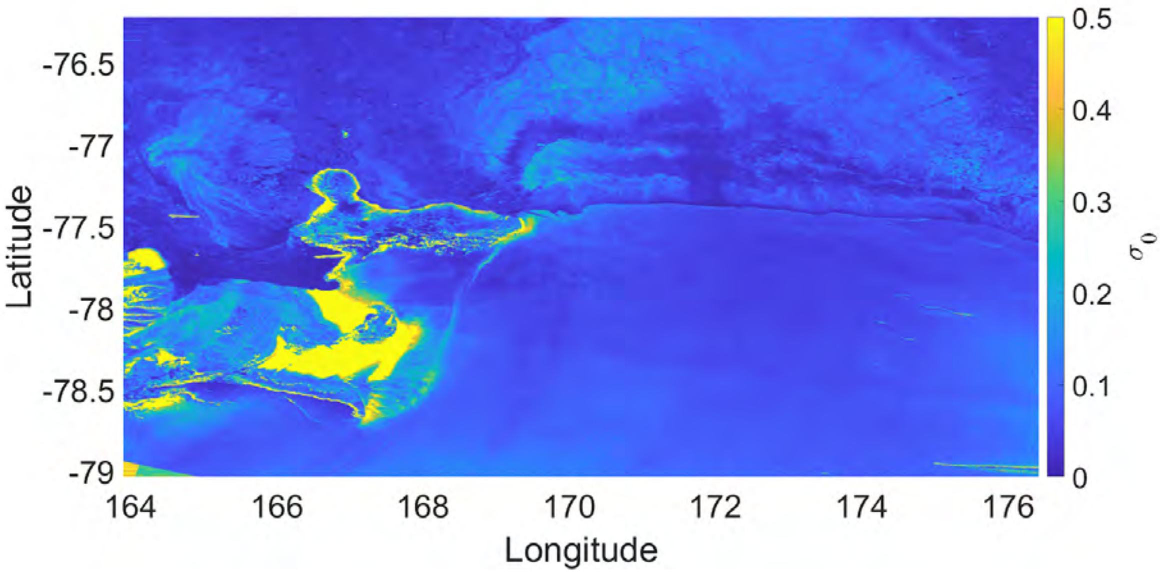



*Figure 3.6.1.2: Example of SAR images used as reference for Ross ice shelf*

The ice coastline can change its shape along season or it is possible that a big piece of coast can collapse creating an iceberg with a consequently big change in shape. As reports in [15] the ice velocity ranges from a few meters per year to several hundred meters per year in ice streams. Ice velocity increases as the ice moves seaward, reaching 1 km per year in the central portions of the ice front. We do not focus on the center of the shelf and its movement is much lower than geometrical resolution of radiometric imagery, indeed comparing SAR image in winter season and SAR image in summer, we obtain the same reference contour when we project the ice contour in the grid with 5 km of special resolution, as explained in the Appendix C. Therefore we must only monitor the formation of icebergs and to do this we compared SAR images on January 2016 with an image on 2017, observing the absence of significant variations in shape.

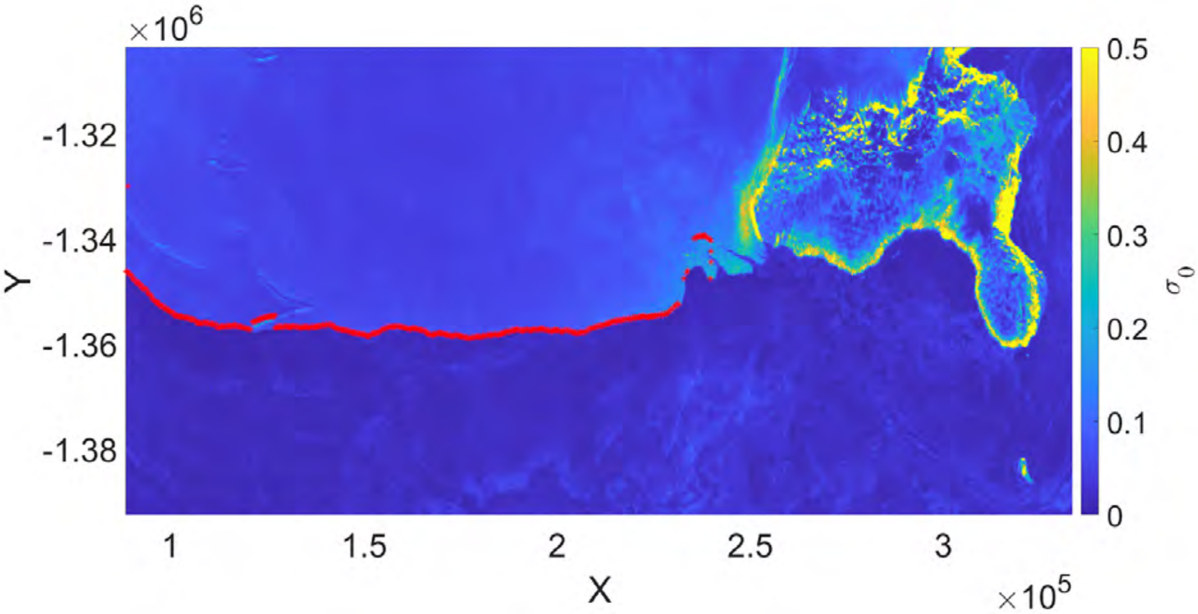



*Figure 3.6.1.3: SAR contour extracted with Canny algorithm*

We have focused on the following box:

- Latitude = [ -78.5 ; -76.5];
- Longitude = [ 170.6 ; 178.5];

BT contrast for Ross ice shelf has been calculated with the Eq. 3.6.1.1 wuth points in Tab. 3.6.1.1, expressed on polar stereographic map

$$\Delta T_{Bm} = \frac{(T_B - T_A) + (T_D - T_C) + (T_F - T_E) + (T_H - T_G)}{4} \quad (3.6.1.1)$$

*Table 3.6.1.1: Polar stereographic coordinates of eight points used to calculate the BT contrast for Ross ice shelf following the eq. C.2.*

| Points | X | Y |
|---|---|---|
| A | $6.3796\ 10^4$ | $-1.3149\ 10^6$ |
| B | $6.3796\ 10^4$ | $-1.3763\ 10^6$ |
| C | $1.0524\ 10^5$ | $-1.33\ 10^6$ |

| | | |
|---|---|---|
| D | 1.0524 $10^5$ | -1.4070 $10^6$ |
| E | 1.6222 $10^5$ | -1.3303 $10^6$ |
| F | 1.6222 $10^5$ | -1.3814 $10^6$ |
| G | 2.1920 $10^5$ | -1.3251 $10^6$ |
| H | 2.1920 $10^5$ | -1.3712 $10^6$ |

The description of the proposed geolocation assessment methodology for this target class is provided in the Appendix C. After the defuzzification step during 2016, we obtain the samples in the following Tab. 3.6.1.2:

*Table 3.6.1.2: Number of visible days during 2016 for Ross ice shelf*

| Jan | Feb | Mar | Apr | May | Jun | Jul | Aug | Sep | Oct | Nov | Dec | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 5 | 32 | 80 | 87 | 97 | 122 | 95 | 90 | 75 | 40 | 1 | 725 |

From Tab. 3.6.1.2 the total number of SSMIS available images in 2016 is 599 with geolocation assessment results summarized in Tab. 3.6.1.3. The mean BT contrast $\Delta T_B$ , derived from (C.3) and obtained from the 2016 dataset along the lake coastline, is about 16.4 K.

*Table 3.6.1.3: Result for Filchner-Ronne ice shelf on 2016*

| Geolocation assessment parameter | Value [km] |
|---|---|
| Geolocation accuracy average [km] | 5.30 |
| Geolocation accuracy standard deviation [km] | 2.18 |

### 3.6.2 Filchner-Ronne ice shelf

The whole Filchner-Ronne ice shelf covers some 430,000 km², making it the second largest ice shelf in Antarctica, after the Ross Ice Shelf, as shown in Fig. 3.6.1.1 by the black circle. The necessary steps to validate the geolocation accuracy for Filchner-Ronne ice shelf are the same as Ross ice shelf, as explained in Appendix C. The only difference is that the reference SAR image is different, as shown in Fig. 3.6.2.1.

Fig. 3.6.2.2 shows the same SAR image contained in the Fig. 3.6.2.1 projected on a polar stereographic map. The red marker shows the contour obtained with Canny algorithm on SAR image.

During 2016, there was no iceberg formation and the shape is almost the same throughout the entire year. we have focused on the following box:

- Latitude = [ -75.7 ; -74.4];
- Longitude = [ -64.5 ; -56.5];

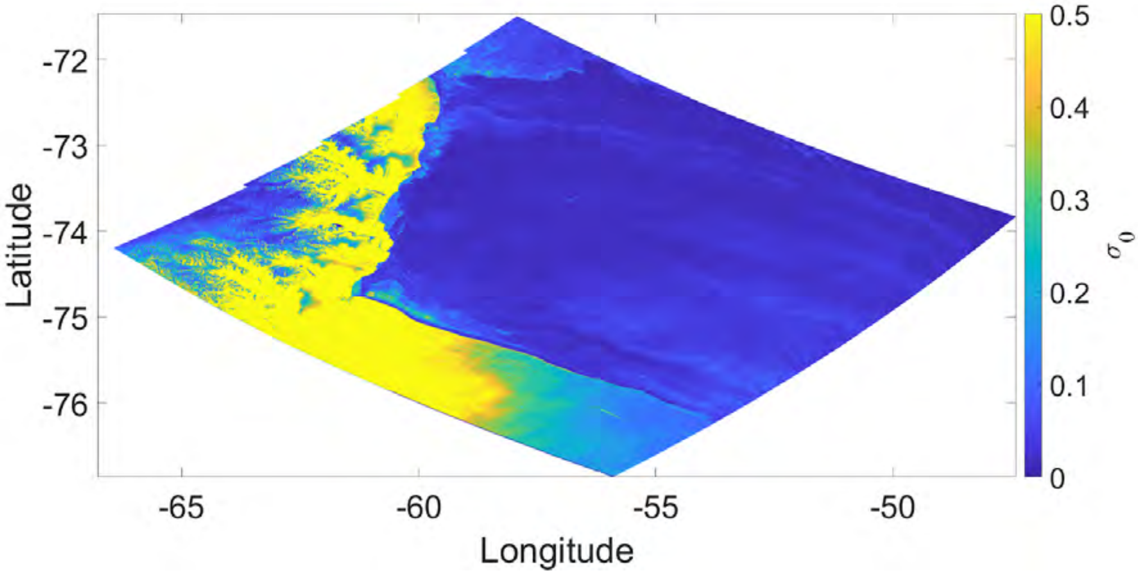



*Figure 3.6.2.1: Example of SAR images used as reference for Filchner-Ronne ice shelf*

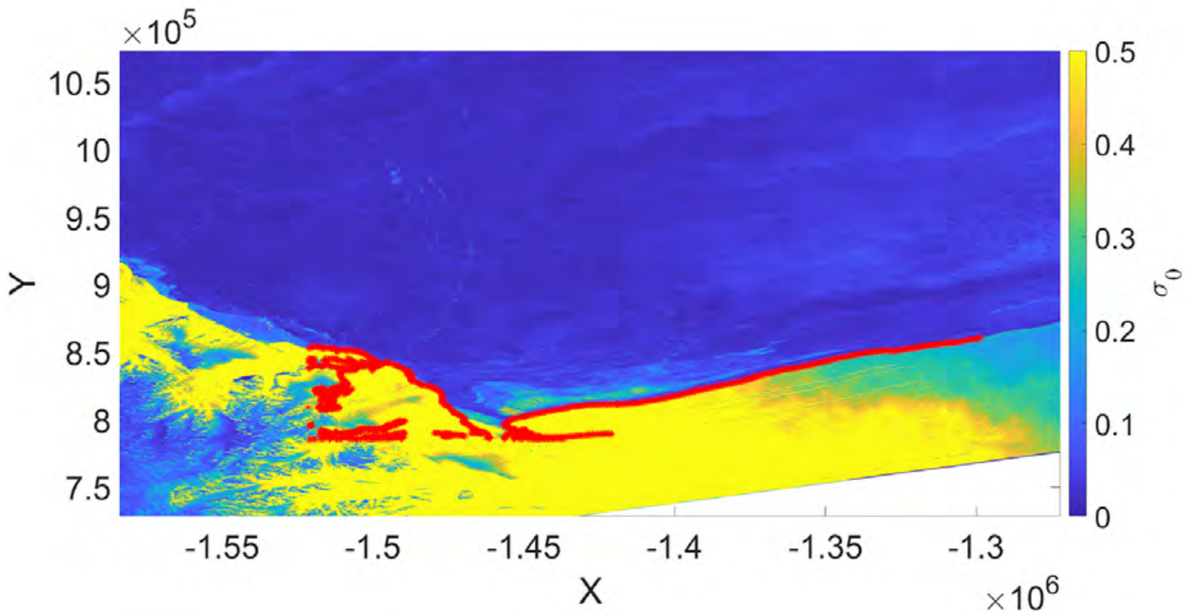



*Figure 3.6.2.2: SAR contour extracted with Canny algorithm*

BT contrast for Filchner-Ronne ice shelf has been calculated with the Eq. 3.6.1.1 with points in Tab. 3.6.2.1, expressed on a polar stereographic map.

$$\Delta T_{Bm} = \frac{(T_B - T_A)+(T_D - T_C)+(T_F - T_E)+(T_H - T_G)}{4} \tag{3.6.2.1}$$

*Table 3.6.2.1: Polar stereographic coordinates of eight points used to calculate the BT contrast for Filchner-Ronne ice shelf following the eq. C.2.*

| Points | X | Y |
|---|---|---|
| A | $-1.5011\ 10^6$ | $8.3170\ 10^5$ |
| B | $-1.5011\ 10^6$ | $8.8246\ 10^5$ |
| C | $-1.4453\ 10^6$ | $7.9110\ 10^5$ |
| D | $-1.4453\ 10^6$ | $8.2663\ 10^5$ |
| E | $-1.3794\ 10^6$ | $8.0633\ 10^5$ |
| F | $-1.3794\ 10^6$ | $8.4693\ 10^5$ |
| G | $-1.3185\ 10^6$ | $8.4186\ 10^5$ |
| H | $-1.3185\ 10^6$ | $8.7739\ 10^5$ |

The description of the proposed geolocation assessment methodology for this target class is provided in the Appendix C. After the defuzzification step during 2016, we obtain the samples in the following Tab. 3.6.2.2:

*Table 3.6.2.2: Number of visible days during 2016 for Filchner-Ronne ice shelf*

| Jan | Feb | Mar | Apr | May | Jun | Jul | Aug | Sep | Oct | Nov | Dec | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 25 | 84 | 82 | 58 | 90 | 99 | 73 | 30 | 0 | 0 | 541 |

From Tab. 3.6.2.2 the total number of SSMIS available images in 2016 is 387 with geolocation assessment results summarized in Tab. 3.6.2.3.

*Table 3.6.2.3: Result for Filchner-Ronne ice shelf on 2016*

| Geolocation assessment parameter | Value [km] |
|---|---|
| Geolocation accuracy average [km] | 4.31 |
| Geolocation accuracy standard deviation [km] | 1.89 |

### 3.6.3 Amery ice shelf

The necessary steps to validate the geolocation accuracy for Filchner-Ronne ice shelf are the same as Ross ice shelf, as explained in Appendix C. The only difference is that the reference SAR image is different, as shown in Fig. 3.6.2.1.

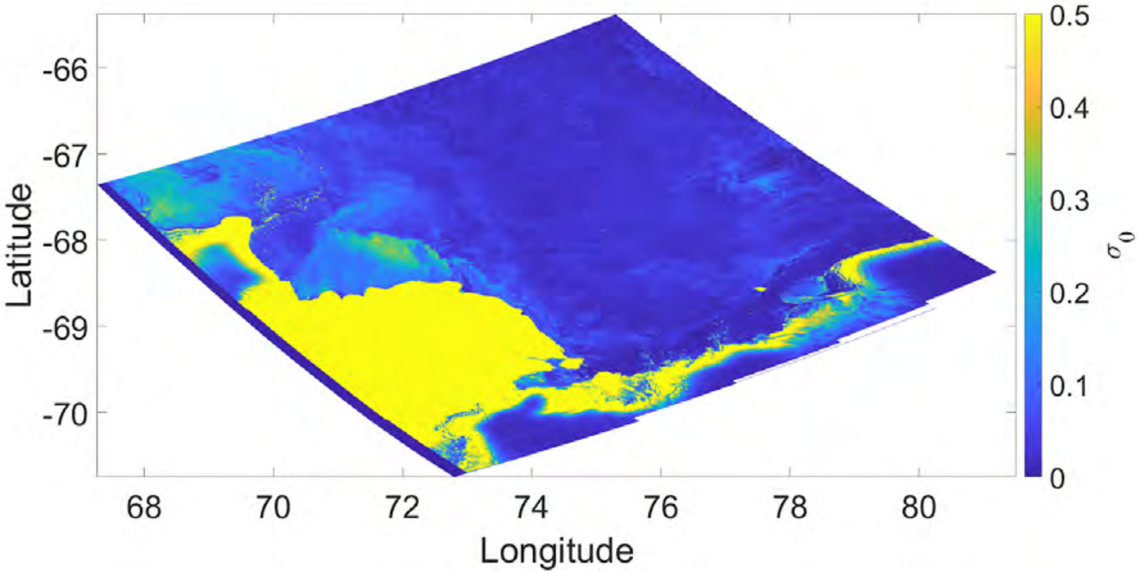



*Figure 3.6.3.1: Example of SAR images used as reference for Amery ice shelf*

Fig. 3.6.3.2 shows the same SAR image contained in the Fig. 3.6.2.1 projected on a polar stereographic map. The red marker shows the contour obtained with Canny algorithm on SAR image.

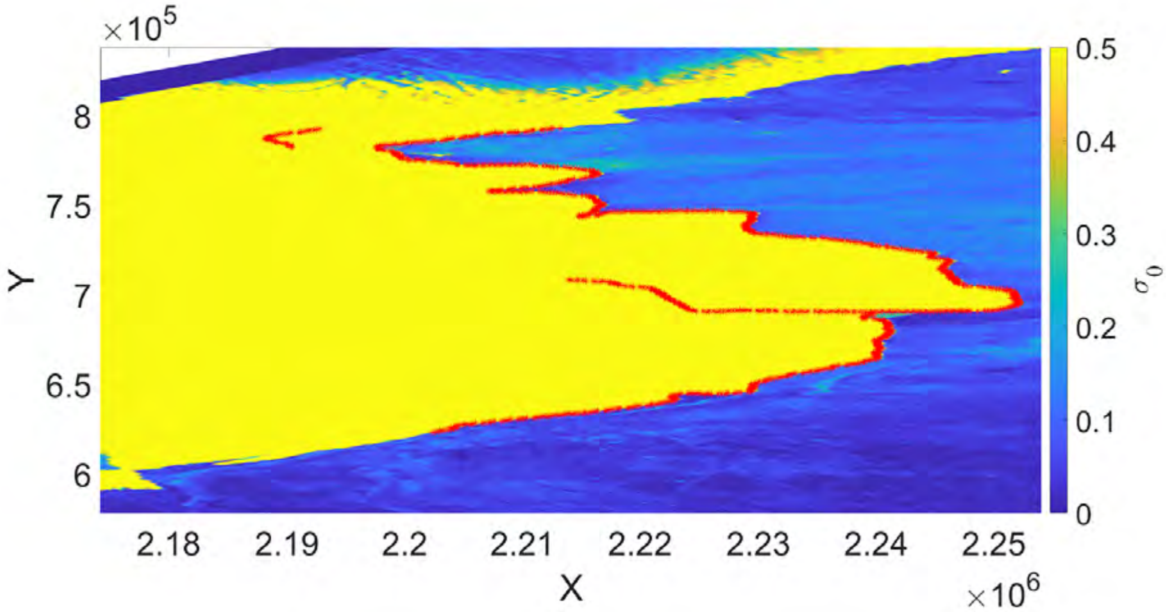



*Figure 3.6.3.2: SAR contour extracted with Canny algorithm*

During 2016, there was no iceberg formation and the shape is almost constant during the entire year. We have focused on the following box:

- Latitude = [ -69.1 ; -68.0];
- Longitude = [ 70.8 ; 74.2];

BT contrast for Amery ice shelf has been calculated with the Eq. 3.6.3.1 with points in Tab. 3.6.1.1, expressed on a polar stereographic map.

$$\Delta T_{Bm} = \frac{(T_B - T_A) + (T_D - T_C) + (T_F - T_E) + (T_H - T_G)}{4} \quad (3.6.3.1)$$

*Table 3.6.3.1: Polar stereographic coordinates of eight points used to calculate the BT contrast for Amery ice shelf following the eq. C.2.*

| Points | X | Y |
|---|---|---|
| A | $2.1984\ 10^6$ | $7.6657\ 10^5$ |
| B | $2.2356\ 10^6$ | $7.6657\ 10^5$ |
| C | $2.2303\ 10^6$ | $7.1870\ 10^5$ |
| D | $2.2675\ 10^6$ | $7.1875\ 10^5$ |
| E | $2.2515\ 10^6$ | $6.7093\ 10^5$ |
| F | $2.2143\ 10^6$ | $6.7093\ 10^5$ |
| G | $2.1984\ 10^6$ | $6.3373\ 10^5$ |
| H | $2.2356\ 10^6$ | $6.3373\ 10^5$ |

The description of the proposed geolocation assessment methodology for this target class is provided in the Appendix C. The number of obtained samples, after the defuzzification step during 2016, is given in Tab. 3.6.3.2:

*Table 3.6.3.2: Number of visible days during 2016 for Filchner-Ronne ice shelf*

| Jan | Feb | Mar | Apr | May | Jun | Jul | Aug | Sep | Oct | Nov | Dec | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 30 | 22 | 25 | 41 | 37 | 26 | 39 | 15 | 3 | 0 | 242 |

From Tab. 3.6.3.2 the total number of SSMIS available images in 2016 is 153 with geolocation assessment results summarized in Tab. 3.6.3.3.

*Table 3.6.3.3: Result for Filchner-Ronne ice shelf on 2016*

| Geolocation assessment parameter | Value [km] |
|---|---|

| | |
|---|---|
| Geolocation accuracy average [km] | 5.32 |
| Geolocation accuracy standard deviation [km] | 2.27 |

### 3.7 Titicaca lake in the southern hemisphere

Titicaca lake is a large, deep lake in the Andes on the border of Bolivia and Peru. It has a surface of about 8372 km$^2$ and an elevation of 3,812 m (see Fig. 3.7.1).

We have focused on the following box:

- Latitude = [ -17.5 ; -14.5]
- Longitude = [ -70.3 ; -68.0];


*Figure 3.7.1: Titicaca lake*

Note that the mean BT contrast $\Delta T_B$ is derived from the following Eq. 3.7.1 using points in Tab. 3.2.1:

$$\Delta T_{Bm} = \frac{(T_B - T_A) + (T_C - T_A) + (T_D - T_A) + (T_E - T_A)}{4} \qquad (3.7.1)$$

*Table 3.7.1.: Coordinates of five points used to calculate the BT contrast for Titicaca lake following the eq. A.2.*

| **Points** | **Latitude** [deg] | **Longitude** [deg] |
|---|---|---|
| A | -15.8766 | -69.3462 |
| B | -15.1745 | -69.3462 |
| C | -15.8766 | -68.6250 |

| | | |
|---|---|---|
| D | -16.5787 | -69.3462 |
| E | -15.8766 | -70.2115 |

Fig. 3.7.2 shows two examples for Titicaca lake with SSMIS F17 at 183±6.6 GHz H.

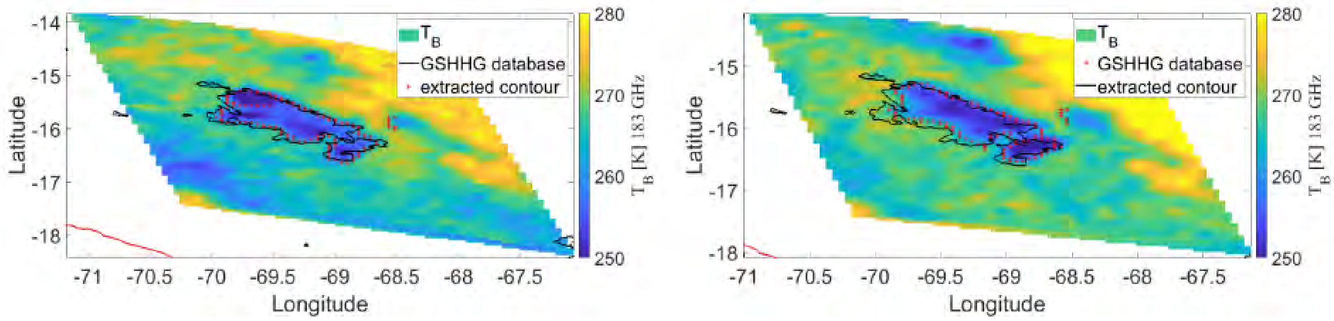



*Figure 3.7.2: Brightness temperature image at 183±6.6 GHz H over Titicaca lake with SSMIS F17 on 2016/05/31 (left) and 2016/07/31 (right). The red markers represent the GSHHG shoreline database and black markers are provided by Canny edge detection from radiometric images.*

The description of the proposed geolocation assessment methodology for this target class is provided in the Appendix A.

*Table 3.7.2: Number of visible days during 2016 for Titicaca lake applying the cloud-masking algorithm*

| Jan | Feb | Mar | Apr | May | Jun | Jul | Aug | Sep | Oct | Nov | Dec | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 3 | 8 | 11 | 8 | 10 | 5 | 3 | 2 | 2 | 52 |

From Tab. 3.7.2 the total number of SSMIS available images in 2016 is 41 with geolocation assessment results summarized in Tab. 3.7.3.

*Table 3.7.3: Result for Titicaca lake in 2016*

| Geolocation assessment parameter | Value [km] |
|---|---|
| Geolocation accuracy average [km] | 4.80 |
| Geolocation accuracy standard deviation [km] | 2.50 |

## 3.8 Andean mountains in the southern hemisphere

Andean mountains are the longest continental mountain range in the world, forming a continuous highland along the western edge of South America (see Fig. 3.8.1).

We have focused on the following box:

- Latitude = [ -15.8 ; -18.8]
- Longitude = [ -72.8 ; -69.2];

Tab. 3.8.1 shows the coordinates for the eight points used to calculate the BT contrast for Andean mountains following the eq. B.2.

Fig. 3.8.2 shows an example for Andean mountains with SSMIS F17 at 183±6.6 GHz H.


*Figure 3.8.1: Andean mountains from Google*

The BT contrast for Andean mountains is derived following the Eq. 3.8.1 with points in Tab. 3.8.1

$$\Delta T_{Bm} = \frac{(T_B - T_A) + (T_D - T_C) + (T_F - T_E) + (T_H - T_G)}{4} \quad (3.8.1)$$

*Table 3.8.1.: Coordinates of eight points used to calculate the BT contrast for Andean mountains following the eq. B.2.*

| **Points** | **Latitude** [deg] | **Longitude** [deg] |
|---|---|---|
| A | -16.0769 | -71.2432 |
| B | -16.5385 | -71.8757 |
| C | -16.2154 | -70.4162 |
| D | -17.0000 | -71.1459 |
| E | -16.9077 | -70.0270 |
| F | -17.4615 | -70.5622 |
| G | -17.6923 | -69.4432 |

| | | |
|---|---|---|
| H | -18.3846 | -69.9297 |

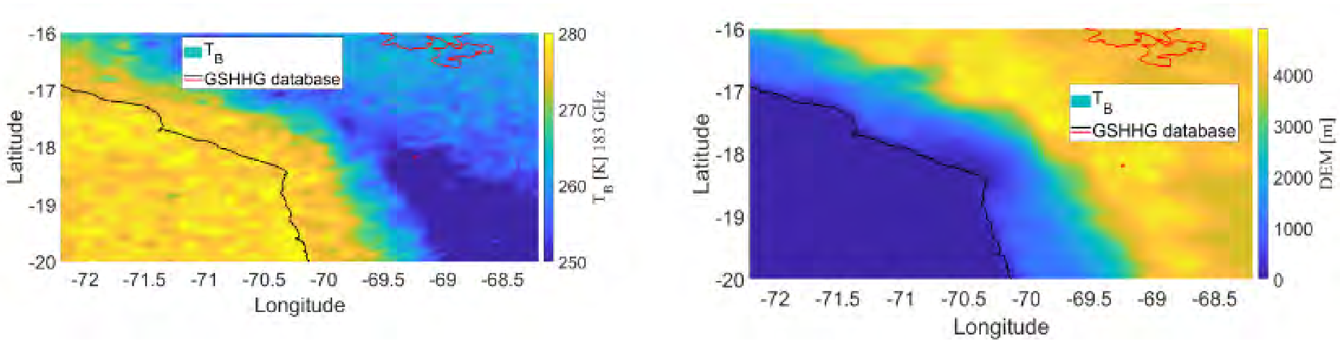



*Figure 3.8.2: In the left there is the brightness temperature image at 183±6.6 GHz H over Andean mountains SSMIS F17 on 2016/07/14 In the right there is the reference digital elevation model. The black line represents GSHHG shoreline database*

The description of the proposed geolocation assessment methodology for this target class is provided in the Appendix B.

*Table 3.8.2: Number of visible days during 2016 for Andean mountains applying the cloud-masking algorithm*

| Jan | Feb | Mar | Apr | May | Jun | Jul | Aug | Sep | Oct | Nov | Dec | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5 | 2 | 5 | 10 | 25 | 24 | 21 | 25 | 22 | 20 | 16 | 2 | 177 |

From Tab. 3.8.2 the total number of SSMIS available images in 2016 is 125 with geolocation assessment results summarized in Tab. 3.8.3.

*Table 3.8.3: Result for Andean mountains in 2016*

| Geolocation assessment parameter | Value [km] |
|---|---|
| Geolocation accuracy average [km] | 3.70 |
| Geolocation accuracy standard deviation [km] | 1.95 |

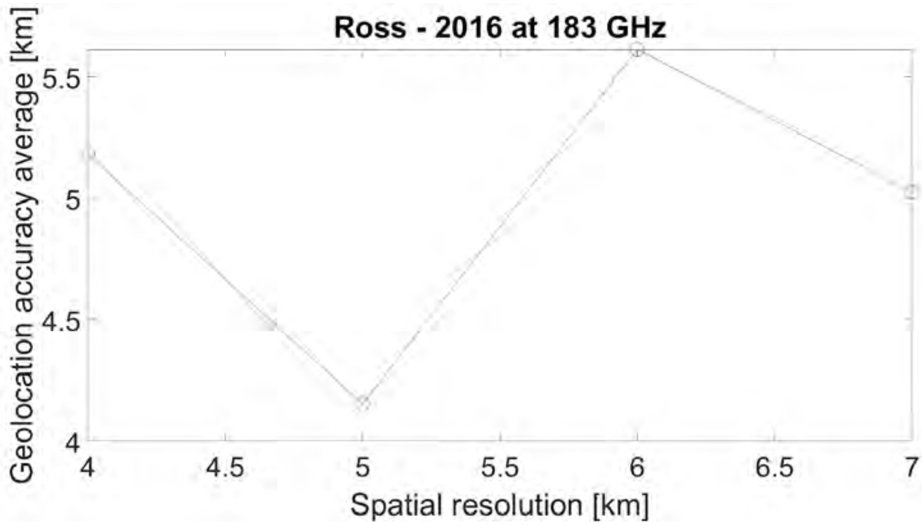
# 4. SENSITIVITY ANALYSIS OF ASSESSMENT METHODOLOGY

The error budget quantification of the geolocation assessment methodology is a difficult task. A feasible approach is to follow a sensitivity analysis of the various steps of the proposed methodology and to evaluate the optimal value of each parameter as well as the related accuracy. In particular we can select the following critical parameters:

1. Interpolation-grid spatial resolution
2. Spatial interpolation method
3. Cross-correlation method

## 4.1 Sensitivity to interpolation-grid spatial resolution

The first important step is the interpolation on a grid with higher resolution. The upsampling factor can be between 2 and 3.5 and here we have focused on the evenly spaced grid at 4, 5, 6 and 7 km. Considering the dataset described in Sec 3.6.1. For the Ross ice shelf and validating the geolocation accuracy changing the interpolating grid, we have obtained the mean value (average) and standard deviation, reported in Fig. 4.1.1.



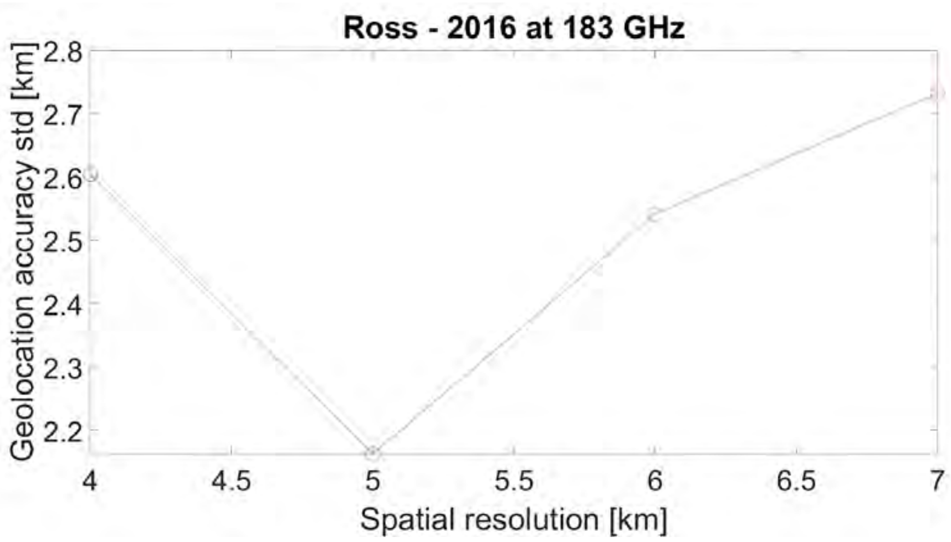




*Figure 4.1.1: Results for Ross ice shelf in relation to resampling grid spatial resolution. In the top there is the mean value of geolocation accuracy and in the bottom the related standard deviation.*

The Fig. 4.1.1 is also summarized in the following Tab. 4.1.1.

*Table 4.1.1: Results for Ross ice shelf in relation to resampling grid spatial resolution.*

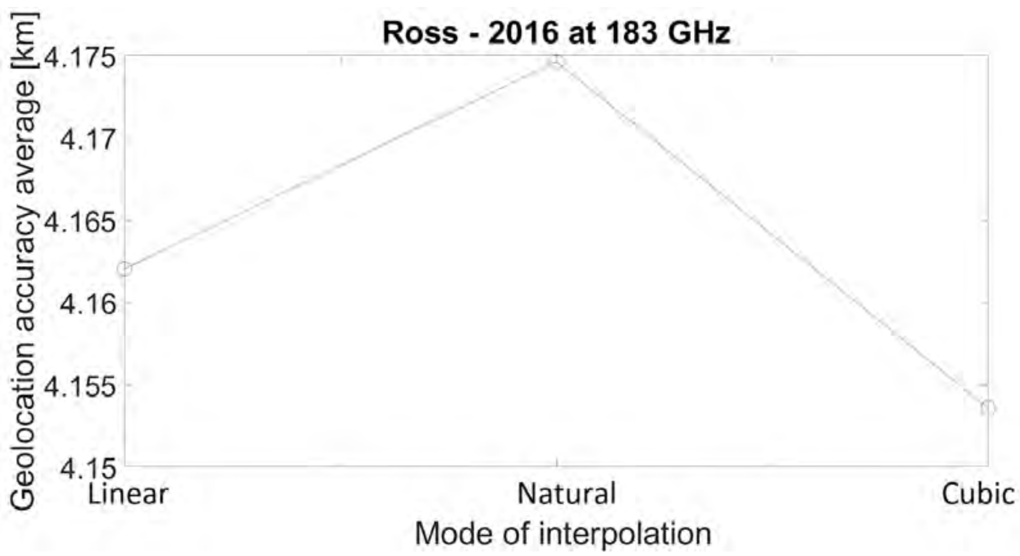
| **Spatial resolution** [km] | **4** | **5** | **6** | **7** |
|---|---|---|---|---|
| Geolocation accuracy average [km] | 5.18 | 4.17 | 5.6 | 5.03 |
| Geolocation accuracy standard deviation [km] | 2.60 | 2.16 | 2.54 | 2.73 |

The minimum value of mean geolocation accuracy and, even more important, the standard deviation is obtained for 5 km of spatial resolution. For this reason in all presented tests we have used this value in the interpolated grid. Calculating the standard deviation of the geolocation accuracy average for several spatial resolutions, it is possible to estimate the variance introduced by this step in the methodology. The obtained standard deviation is 0.62 km.

## 4.2 Sensitivity analysis to spatial interpolation method

In addition to the spatial resolution, the method of spatial interpolation is also a critical one. We have selected the following methods:

- **Linear**: Triangulation based on linear interpolation, such as using linear polynomials to construct new data points within the range of a discrete set of known data points
- **Neighbor**: Triangulation based on natural neighbor interpolation. The method is based on Voronoi tessellation of a discrete set of spatial points. This has advantages over simpler methods of interpolation, such as nearest-neighbor interpolation, in that it provides a smoother approximation to the underlying "true" function.
- **Cubic**: Triangulation based on cubic interpolation. Images resampled with cubic interpolation are smoother and have fewer interpolation artifacts with respect to linear interpolation, but it requires more computational costs.



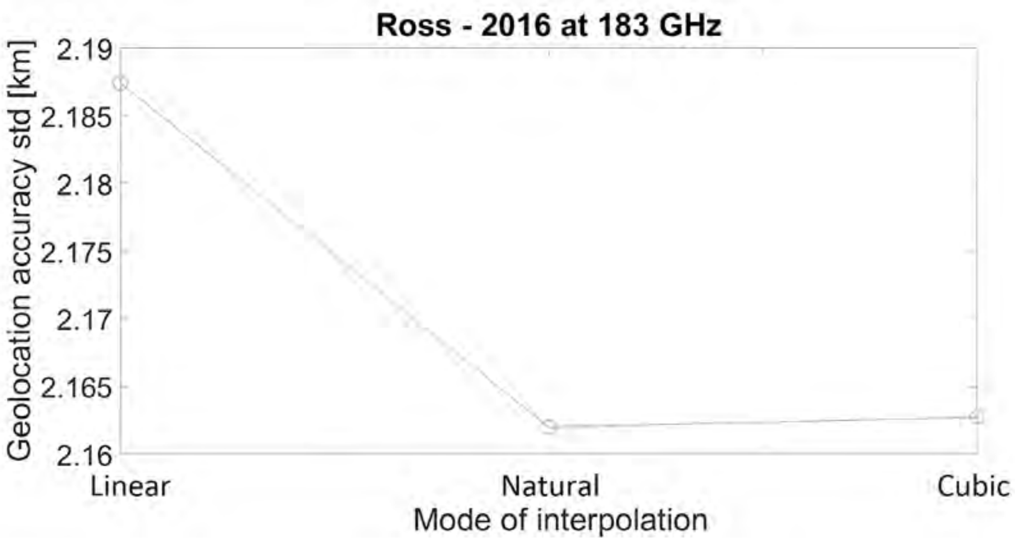



*Figure 4.2.1: Results for Ross ice shelf in relation to upsampling methods. Geolocation accuracy average in the top and geolocation accuracy standard deviation in the bottom.*

*Table 4.2.1: Results for Ross ice shelf in relation to upsampling methods.*

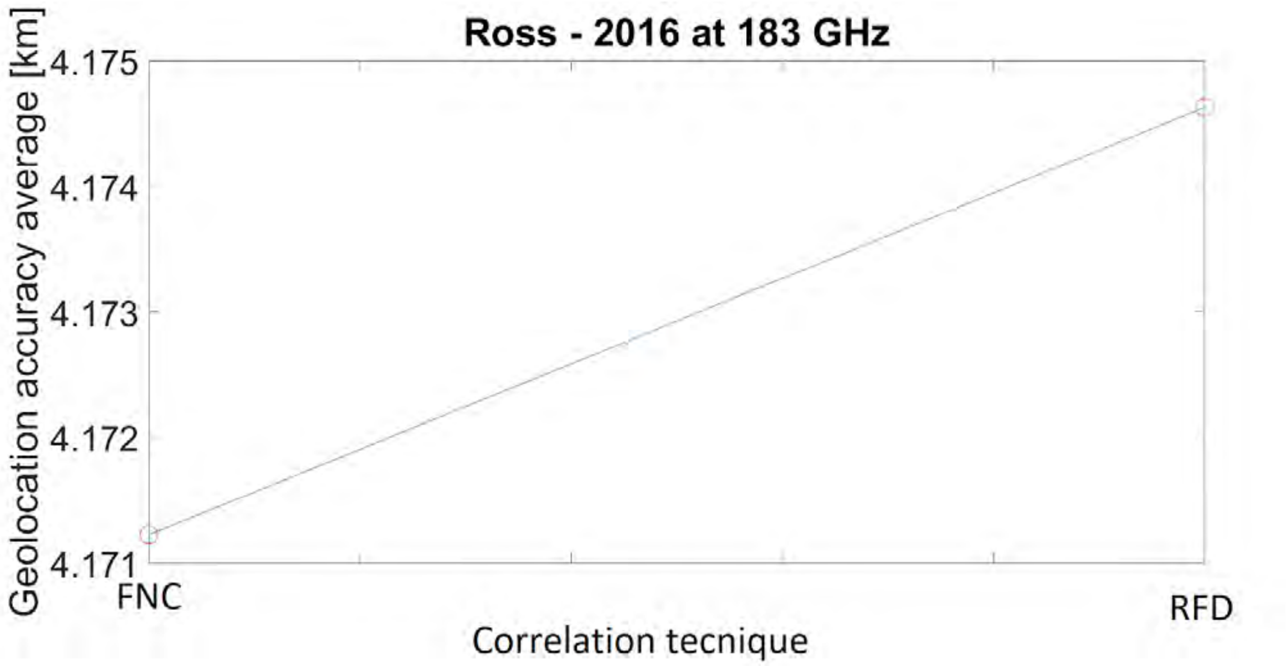
| Interpolation method | Linear | Natural | Cubic |
|---|---|---|---|
| Geolocation accuracy average [km] | 4.16 | 4.17 | 4.15 |
| Geolocation accuracy standard deviation [km] | 2.19 | 2.16 | 2.16 |

Calculating the standard deviation of the geolocation accuracy average for several interpolation methods it is possible to estimate the variance introduced by this step in the methodology. The obtained standard deviation is only 10 m. From observing Tab. 4.2.1, we can assume that the choice of interpolation method has only an insignificant impact on the results.

## 4.3 Sensitivity analysis to cross-correlation technique

The final step consists in evaluating the geolocation error using both a fast normalized cross-correlation (FNC) technique [16] and the registration in frequency domain (RFD) [17]. The cross correlation in space provides the normalized cross correlation matrix by directly correlating the two images (radiometric and SAR contours). The maximum of the correlation is then fitted by a 4th-order polynomial to reach sub-pixel accuracy. Conversely, the RFD method correlates the two-dimensional Fourier transform of the two images with sub-pixel image registration.

The two techniques are very similar so that we obtain very similar results, as shown in Fig. 4.3.1.



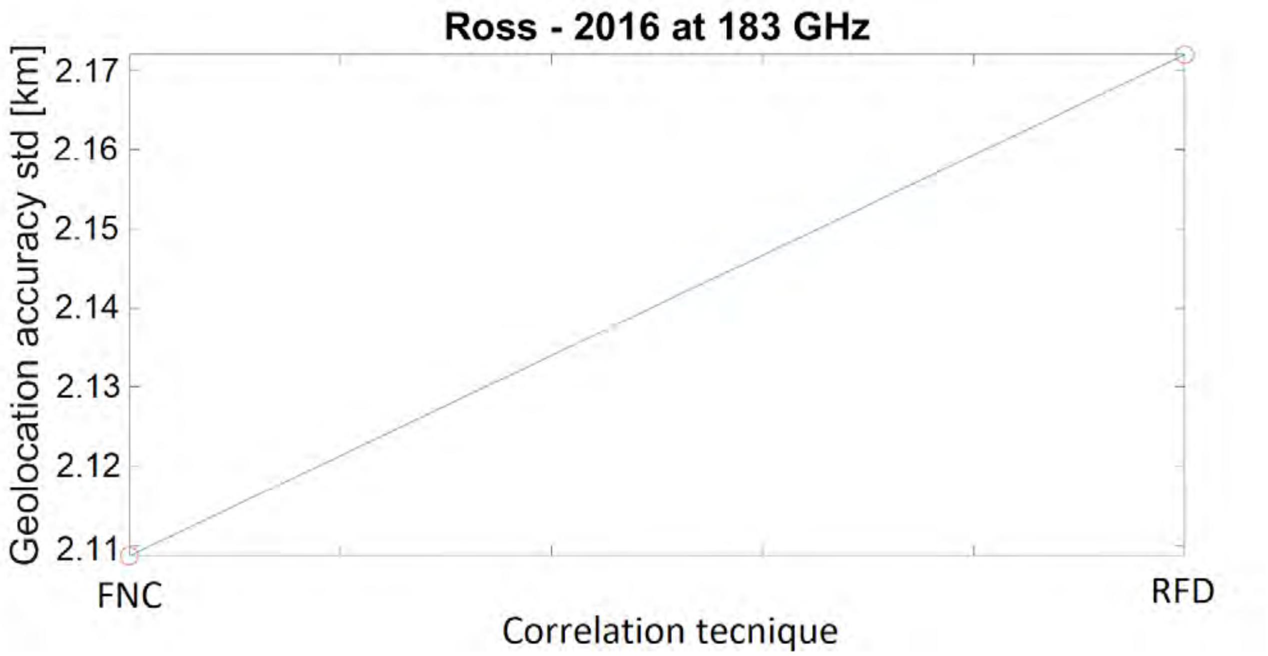




*Figure 4.3.1: Results for Ross ice shelf in relation to correlation technique. Geolocation accuracy average in the top and geolocation accuracy standard deviation in the bottom.*

*Table 4.3.1: Results for Ross ice shelf in relation to upsampling methods.*

| **Correlation technique** | **FNC** | **RFD** |
|---|---|---|
| Geolocation accuracy average [km] | 4.17 | 4.17 |
| Geolocation accuracy standard deviation [km] | 2.11 | 2.17 |

As a further parameter, we can calculate the standard deviation of the geolocation error average equal to 0.04 km. Between FNC and RFD we have almost the same results, because FNC is computed in the spatial domain whereas the RFD is computed in the frequency domain so that the small differences, highlighted in Tab. 4.3.1, are only due to numerics.

## 4.4 Overall sensitivity of the proposed methodology

To have an estimate of the methodology accuracy in presence of many error sources, we can recall the expression of the variance of a multivariate random function. For example, considering a random function $f(z) = x + y$ where $x$ and $y$ may represent additive zero-mean errors, its variance is given by:

$$< z^2 > \; = \sigma_z{}^2 = \; < (x+y)^2 > \; = \; < x^2 + y^2 + 2xy > \; = \; < x^2 > \; + \; < y^2 > \; + \; 2\, Cov\,(x,y) \qquad (4.4.1)$$

where the angle brackets stand for ensemble average. If it is possible to consider $x$ and $y$ statistically independent so that $Cov\,(x,y) = 0$, then it holds

$$< z^2 > \; = \; < x^2 > \; + \; < y^2 > \qquad (4.4.2)$$

and the corresponding standard deviation is:

$$\sigma_z = \sqrt{(\,\sigma_x{}^2 + \sigma_y{}^2)} \qquad (4.4.3)$$

The following Tab. 4.4.1 summarizes the standard deviation of the sensitivity analysis for each considered parameter, presented in Sec. 4.1, 4.2 and 4.3. In all cases the number of samples of each sensitivity numerical experiment is relatively small so that the standard deviation $\sigma$ should be better intended as a parametric variability $\Delta$.

*Table 4.4.1: Standard deviation or parametric variability derived from the sensitivity analysis to each parameter.*

| **Sensitivity parameter** | **Standard deviation or parametric variability [km]** |
|---|---|
| Interpolation-grid spatial resolution | 0.62 |
| Spatial interpolation method | 0.01 |
| Cross-correlation technique | 0.04 |

Extending eq. (4.4.3) to more than 2 variables and assuming independent error contributions, it is possible to evaluate the overall accuracy of the methodology in terms of its standard deviation or parametric variability:

$$\Delta \; = \sqrt{(0.62^2 + 0.01^2 + 0.04^2)} \sim 0.62\; km \qquad (4.4.4)$$

Previous results confirm that the most important parameter in the geolocation assessment methodology is the spatial resolution of the interpolation grid. It is worth noting that this choice also influences the spatial upscaling of reference contour sources such as coastline database or SAR imagery.

It can be also important to understand the importance of the dataset size to obtain a stable geolocation error result in terms of average and standard deviation. In particular, Fig. 4.4.1 shows the geolocation error accuracy and its standard deviation against the number of samples, considering the Ross ice shelf (having 599 cloud-masked samples).

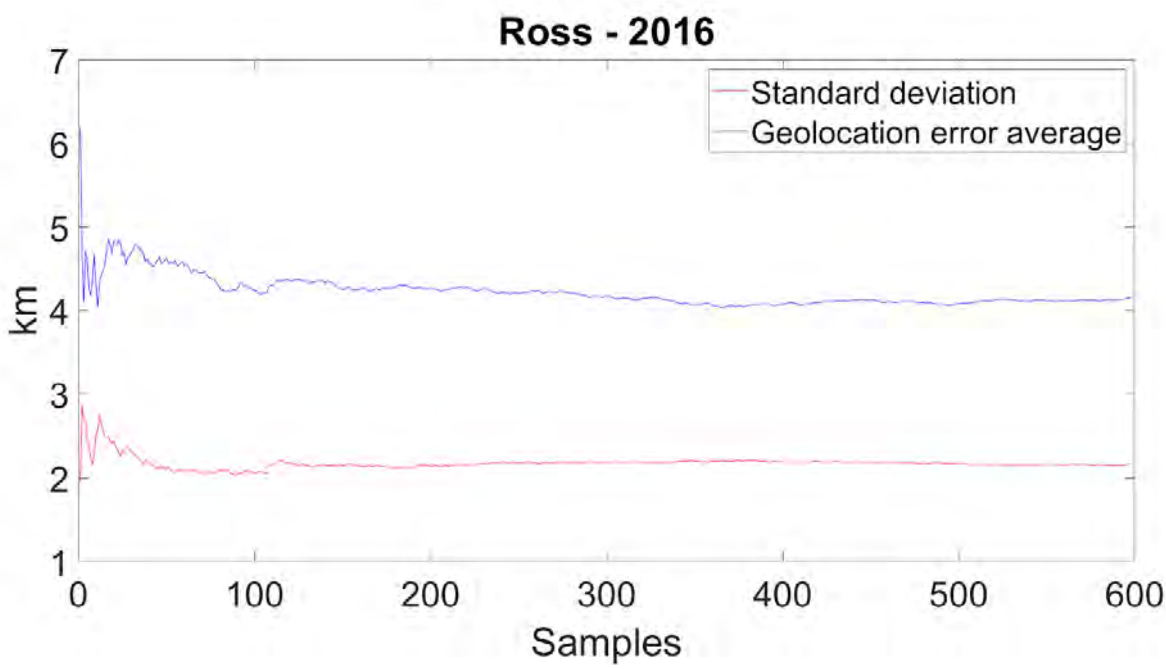



*Figure 4.4.1. Geolocation error average (blue line) and standard deviation (red line) for Ross ice shelf using SSMIS F17 at 183± 6.6 GHz at H polarization.*

Previous figure shows that the geolocation error average becomes stable with about 100 samples. To obtain a substantial stable value of the error standard deviation, about 70 samples seems to be enough.

## 4.5 Testing the nominal accuracy of geolocation error assessment

In order to evaluate the accuracy of the proposed TCM methodology, we can perform some numerical internal tests such as imposing an arbitrary and known geolocation error along latitude and/or longitude.

For example, let us consider an image over the Qinghai lake on 2016/11/11, as shown in Fig. 4.5.1

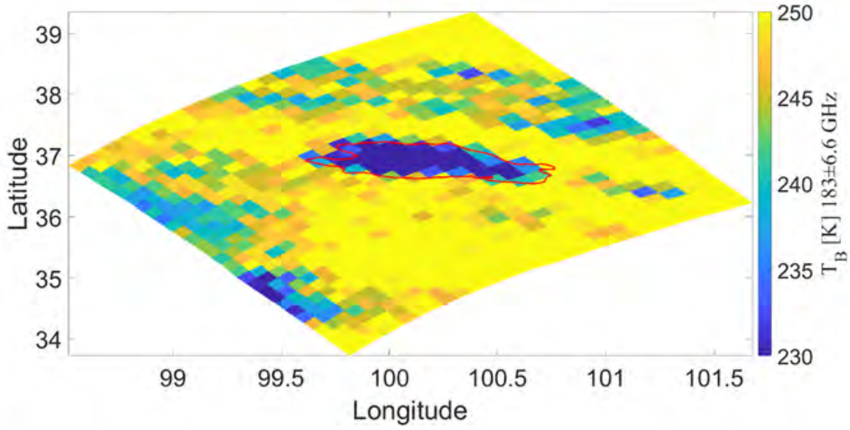



*Figure 4.5.1: Brightness temperature image at 183±6.6 GHz H over the Qinghai lake with SSMIS F17 on 2016/11/11. The red line represents the GSHHG shoreline database.*

Imposing a displacement only along latitude of +0.07°, which corresponds to about 7.78 km, we get what is shown in Figure 4.5.2.

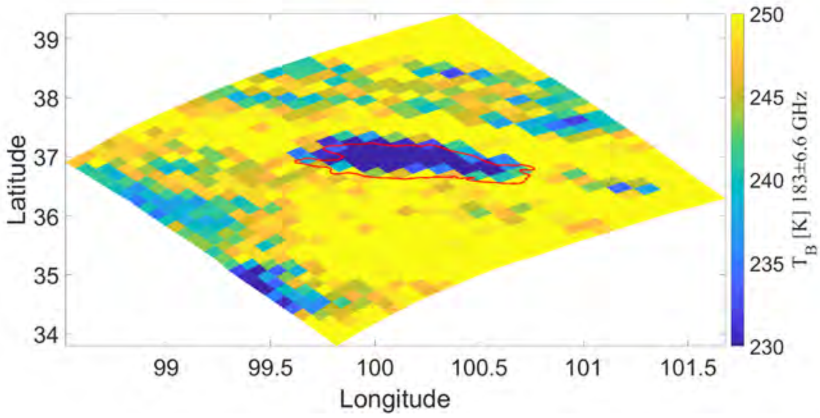



*Figure 4.5.2: Brightness temperature image at 183±6.6 GHz H over the Qinghai lake with SSMIS F17 on 2016/11/11 with a displacement along latitude of 0.07°. The red line represents the GSHHG shoreline database.*

Extracting the contours from both Fig. 4.5.1 and 4.5.2, we obtain the two curves in Fig. 4.5.3.

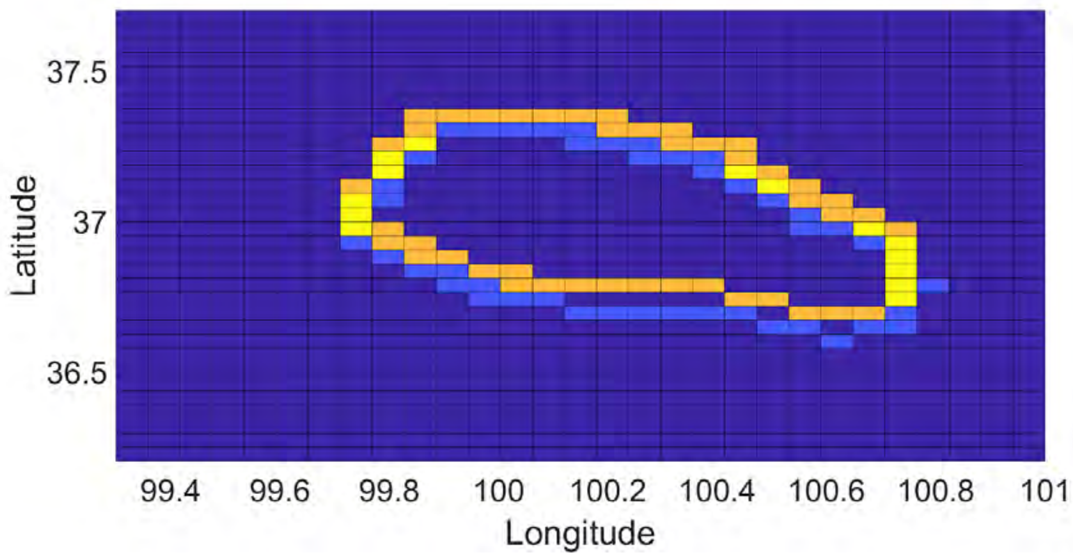



*Figure 4.5.3: Extracted contour using Canny algorithm from original image (blue) and shifted image (yellow).*

Cross-correlating the two contour curves, the proposed TCM methodology provides the following displacements:

- shift along latitude = 0.0636°
- shift along longitude = -0.0057°

The methodology returns a geolocation error of km 7.09, but with an error of 0.69 km respect to 7.78 km, that is the imposed geolocation error. Imposing also displacements both along latitude and longitude such as:

- shift along latitude = 0.06°
- shift along longitude = - 0.02 °

Using these values, we introduce an error of about 6.91 km, obtaining the two curves in Fig. 4.5.4.

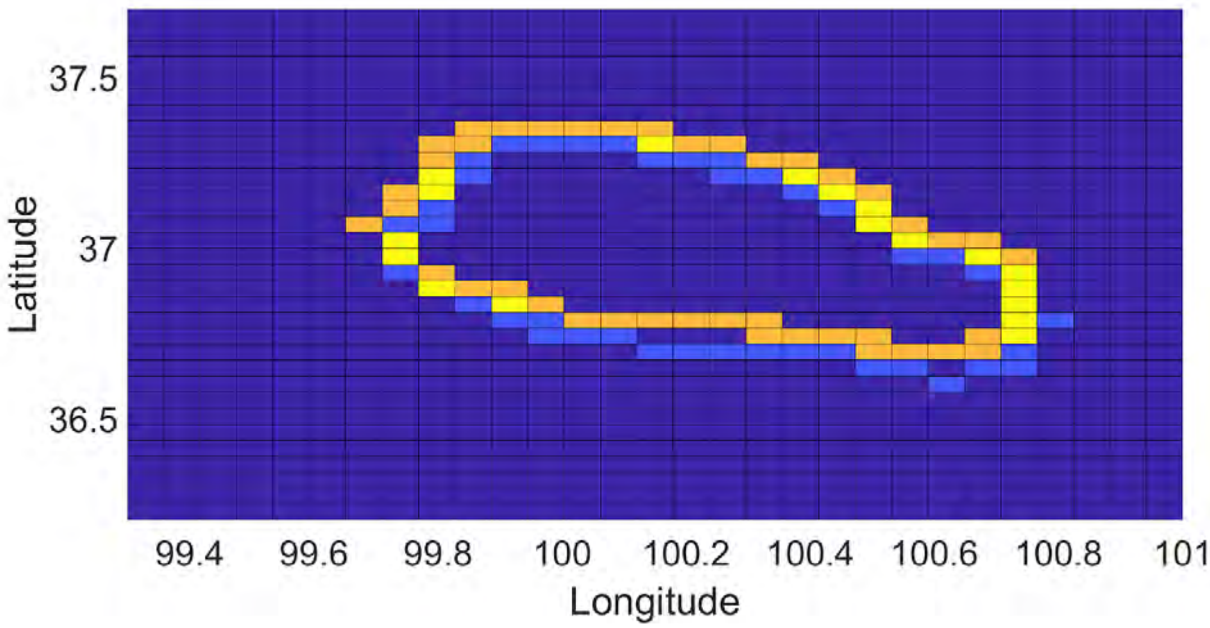



*Figure 4.5.4: Extracted contour using the Canny algorithm from original image (blue) and shifted image (yellow).*

Cross-correlating the two contour curves, the proposed methodology provides the following displacements:

- Shift along latitude = 0.0591°
- Shift along longitude = -0.0113°

The TCM methodology correctly returns the error along both directions with a displacement of 6.65 km, but with an error of 0.26 km. To better understand the accuracy of the results along longitude, we change the imposed displacements:

- shift along latitude = -0.04°
- shift along longitude = + 0.06 °

Using these values, we introduce an error of about 7.02 km, obtaining the two curves in Fig. 4.5.5.

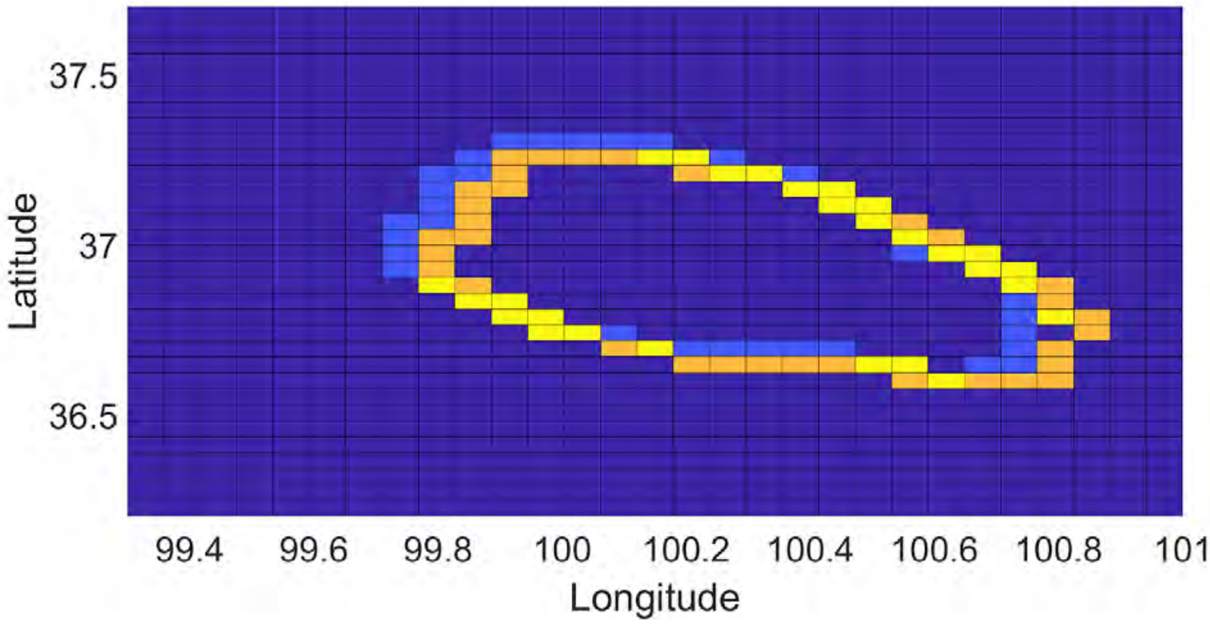



*Figure 4.5.5: Extracted contour using the Canny algorithm from original image (blue) and shifted image (yellow).*

Cross-correlating the two contour curves, the proposed methodology provides the following displacements:

- Shift along latitude = -0.0409°
- Shift along longitude = 0.0567°

The TCM methodology correctly returns the error along both directions with a displacement of 6.79 km, but with an error of 0.23 km. Imposing only a displacement along longitude:

- shift along latitude = 0.0°
- shift along longitude = - 0.08 °

Using these values, we introduce an error of about 7.25 km, obtaining the two curves in Fig. 4.5.6.

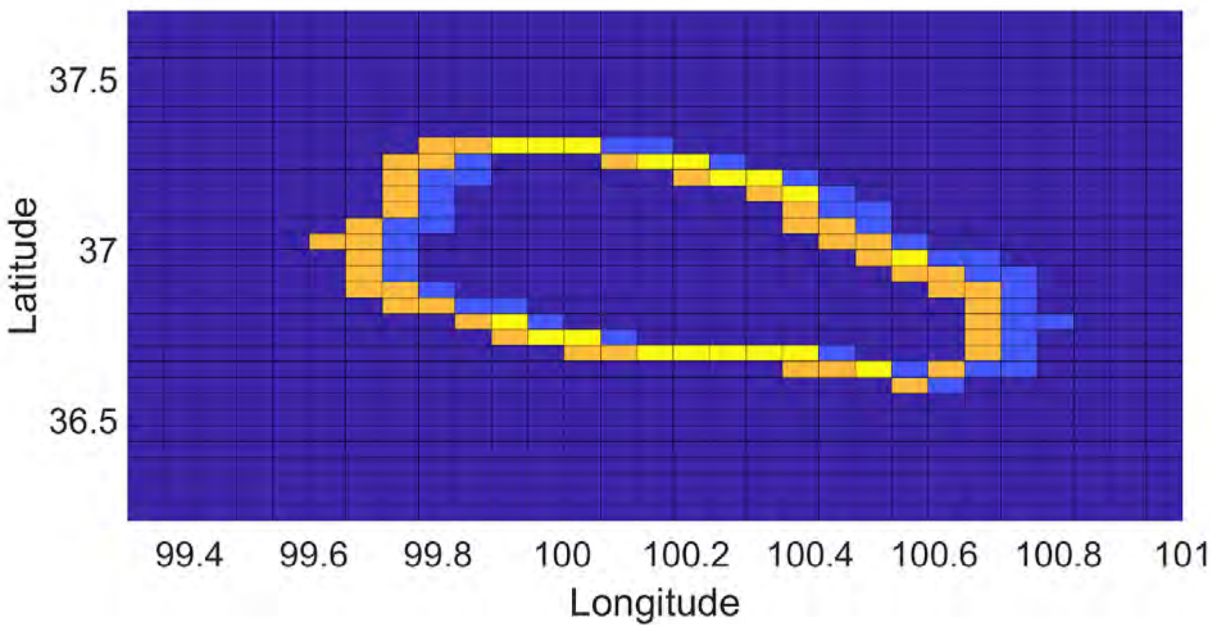



*Figure 4.5.6: Extracted contour using the Canny algorithm from original image (blue) and shifted image (yellow).*

Cross-correlating the two contour curves, the proposed TCM methodology provides the following displacements:

- Shift along latitude = 0,0°
- Shift along longitude = -0.0680°

The TCM methodology correctly returns the error along both directions with a displacement of 6.05 km, but with an error of 1.20 km. Considering an other day for Qinghai lake, e.g. 2016/11/03 as shown in Fig. 4.5.7

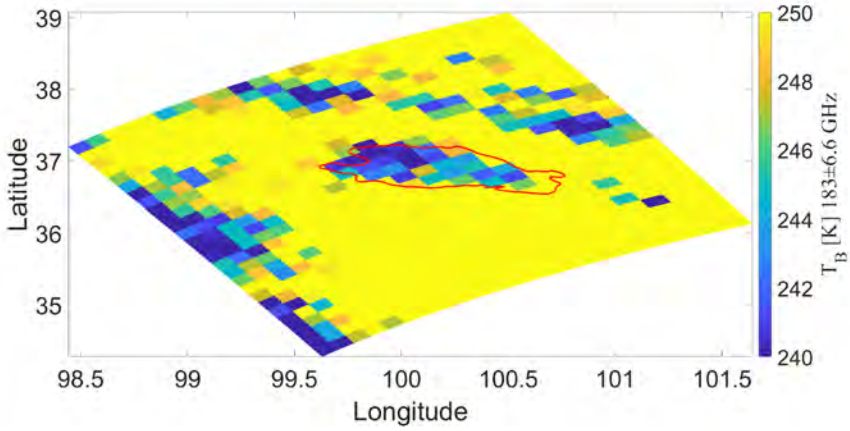



*Figure 4.5.7: Brightness temperature image at 183±6.6 GHz H over the Qinghai lake with SSMIS F17 on 2016/11/03. The red line represents the GSHHG shoreline database.*

Doing the same test again for this day, let imposed a displacement only along latitude of +0.07°, which corresponds to about 7.78 km, we get what shown in Figure 4.5.8.

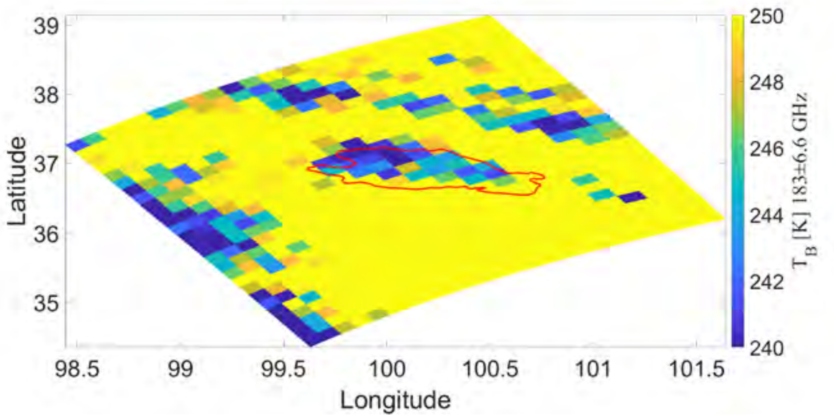



*Figure 4.5.8: Brightness temperature image at 183±6.6 GHz H over the Qinghai lake with SSMIS F17 on 2016/11/03 with a displacement along latitude of 0.07°. The red line represents the GSHHG shoreline database.*

Extracting the contours from both Fig. 4.5.7 and 4.5.8, we obtain the two curves in Fig. 4.5.9

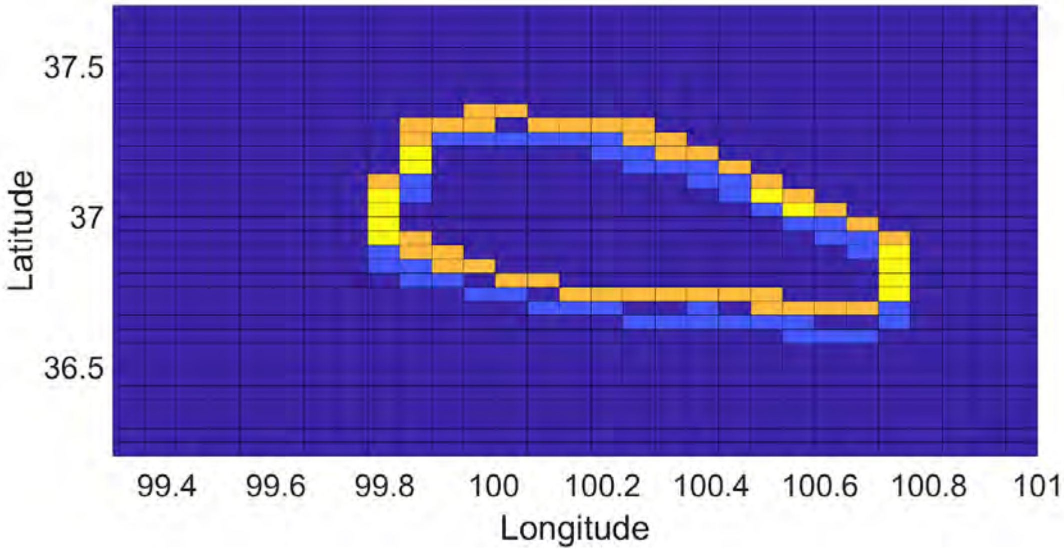



*Figure 4.5.9: Extracted contour using Canny algorithm from original image (blue) and shifted image (yellow).*

Cross-correlating the two contour curves, the proposed TCM methodology provides the following displacements:

- shift along latitude = 0.0727° (about 8.09 km)
- shift along longitude = 0.0°

The methodology correctly returns the error only along latitude, but with an error of 0.3 km. Let us impose the displacements both along latitude and longitude such as:

- shift along latitude = 0.06°
- shift along longitude = - 0.02 °

Using these values, we introduce an error of about 6.91 km, obtaining the two curves in Fig. 4.5.10.

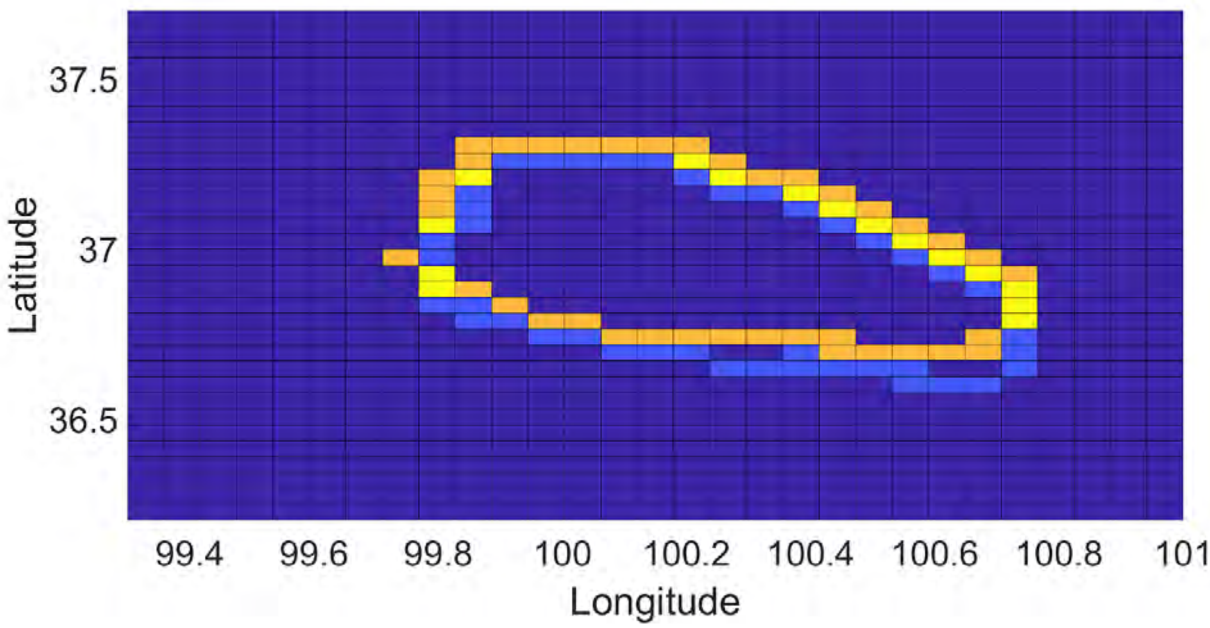



*Figure 4.5.10: Extracted contour using the Canny algorithm from original image (blue) and shifted image (yellow).*

Cross-correlating the two contour curves, the proposed methodology provides the following displacements:

- Shift along latitude = 0.0591°
- Shift along longitude = -0.0170°

The TCM methodology correctly returns the error along both directions with a displacement of 6.74 km, but with an error of 0.17 km. Let us impose the displacements both along latitude and longitude such as:

- shift along latitude = -0.04°
- shift along longitude = + 0.06 °

Using these values, we introduce an error of about 7.02 km, obtaining the two curves in Fig. 4.5.11.

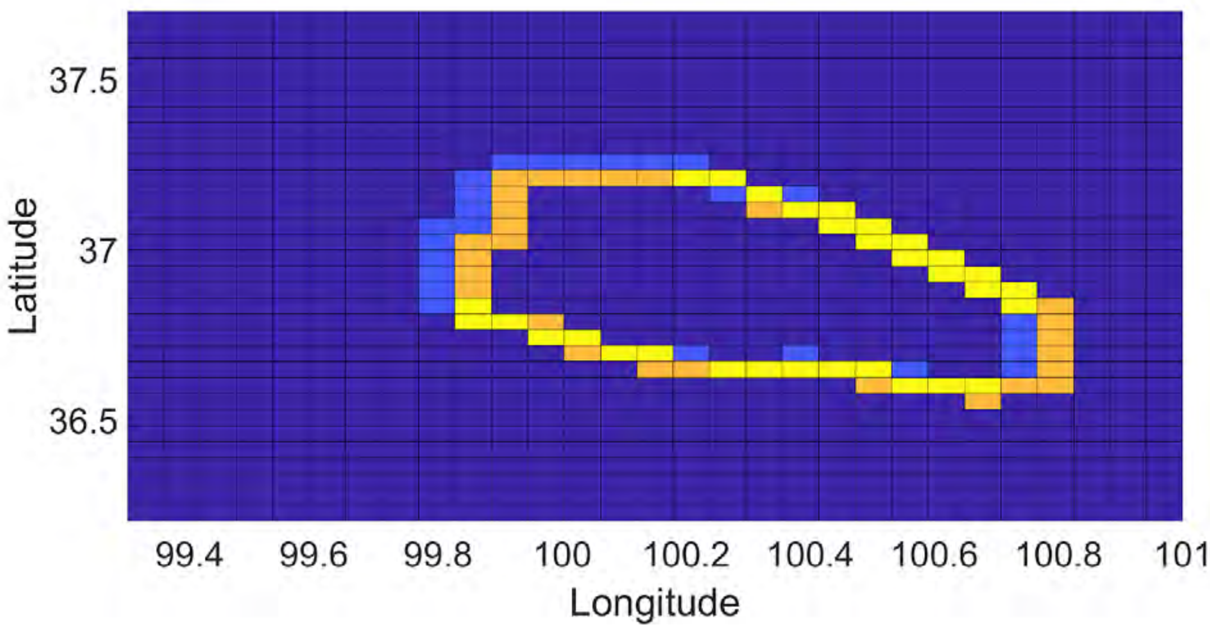



*Figure 4.5.11: Extracted contour using the Canny algorithm from original image (blue) and shifted image (yellow).*

Cross-correlating the two contour curves, the proposed methodology provides the following displacements:

- Shift along latitude = -0.0364 °
- Shift along longitude = 0.0567°

The TCM methodology correctly returns the error along both directions with a displacement of 6.46 km, but with an error of 0.56 km. Finally, imposing the displacements only along longitude:

- shift along latitude = 0.0°
- shift along longitude = - 0.08 °

Using these values, we introduce an error of about 7.25 km, obtaining the two curves in Fig. 4.5.12.

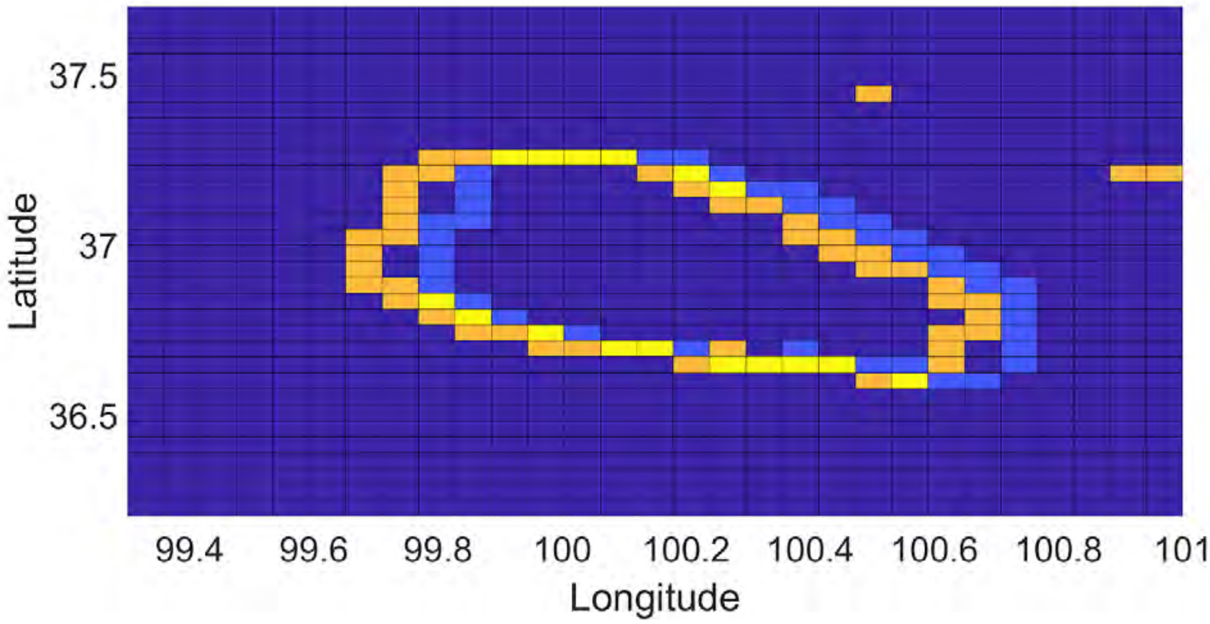



*Figure 4.5.12: Extracted contour using the Canny algorithm from original image (blue) and shifted image (yellow).*

Cross-correlating the two contour curves, the proposed methodology provides the following displacements:

- Shift along latitude = 0.0045°
- Shift along longitude = -0.0793°

The TCM methodology correctly returns the error along both directions with a displacement of 7.08 km, but with an error of 0.16 km.

## 4.6 Sensitivity analysis to the target sample number

This section shows the sensitivity of the geolocation error assessment to the number of samples for all proposed targets.

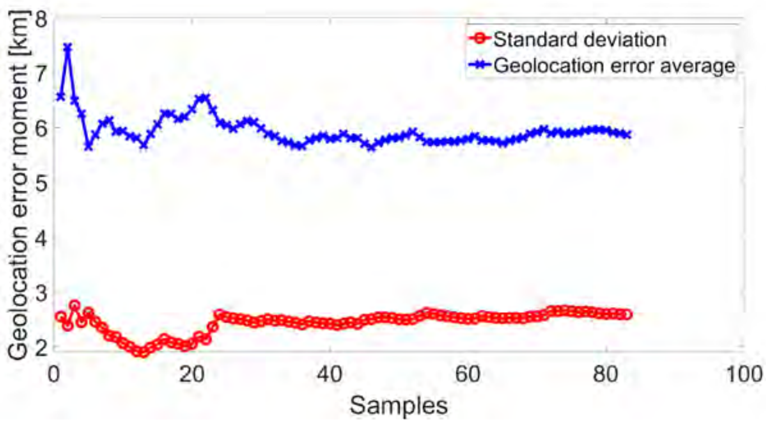



*Figure 4.6.1: Geolocation error average (blue line) and standard deviation (red line) for Qinghai lake using SSMIS F17 at 183±6.6 GHz (H).*

As an example, Fig. 4.6.1 reports the behaviour of the average geolocation accuracy and relative standard deviation with respect to the number of available images for Qinghai lake.

In order to estimate the minimum number of samples necessary to have the convergence of the results, we can introduce the relative difference of average mean error $\delta_\varepsilon(n)$ and the relative difference of average standard deviation $\delta_\sigma(n)$ as:

$$\delta_\varepsilon(n) = \frac{\varepsilon(n) - m_e}{m_e} 100 \qquad (4.6.1)$$

$$\delta_{\sigma_\varepsilon}(n) = \frac{\sigma_\varepsilon(n) - m_\sigma}{m_\sigma} 100 \qquad (4.6.2)$$

where:

- $\varepsilon(n)$ is the average of the geolocation displacement obtained considering the first $n$ samples
- $m_\varepsilon$ is the average of the geolocation displacement obtained considering all samples
- $\sigma_\varepsilon(n)$ is the average of the standard deviation of geolocation displacement obtained considering the first $n$ samples
- $m_\sigma$ is the average of the standard deviation of geolocation displacement obtained considering all samples

Note that both $\delta_\varepsilon(n)$ and $\delta_\sigma(n)$ are percentage values.

Setting arbitrary thresholds for both relative differences, we can impose the convergence of results when it happens simultaneously that:

$$\delta_\varepsilon(n) < \delta_{\varepsilon\, th} \qquad (4.6.3)$$

$$\delta_\sigma(n) < \delta_{\sigma\, th} \qquad (4.6.4)$$

where it holds:

- $\delta_{\varepsilon\, th}$ is the threshold for the average of geolocation displacement
- $\delta_{\sigma\, th}$ is the threshold for the average of standard deviation of geolocation displacement

Considering for simplicity the same percentage threshold $\delta_{th}$ for eq. (4.6.3) and (4.6.4), it is possible to write the following overall condition for the minimum optimal number $n_{opt}$:

$$n_{opt} = n \mid [\delta_\varepsilon(n) < \delta_{th} \;\&\; \delta_\sigma(n) < \delta_{th}] \qquad (4.6.5)$$

The number $n_{opt}$ of samples that satisfies the previous eq. (4.6.5) could be consider the minimum optimal number of target necessary samples to reach a convergence for both mean and standard deviation of the geolocation error. Moreover, considering the detectability/day number $n_{dd}$, summarized in Tab. 3.1.1, it is possible to transform the $n_{opt}$ into the minimum optimal number $n_{opt\,dd}$ of necessary days, needed to satisfy eq. (4.6.5), in the following way:

$$n_{opt\,dd} = \frac{n_{opt}}{n_{dd}} \qquad (4.6.6)$$

For example, in the case of lake and mountain chain targets, whose detectability/day is about 1, we obtain $n_{opt\,dd} = n_{opt}$.

Fig. 4.6.2 shows the relation between $n_{opt}$ and $\delta_{th}$ where $\delta_{th}$ assuming values from 0.5% to 5% with a step of 0.5%, as shown by the marker in Fig. 4.6.2. As expected, relaxing the threshold, $n_{opt}$ decreases.

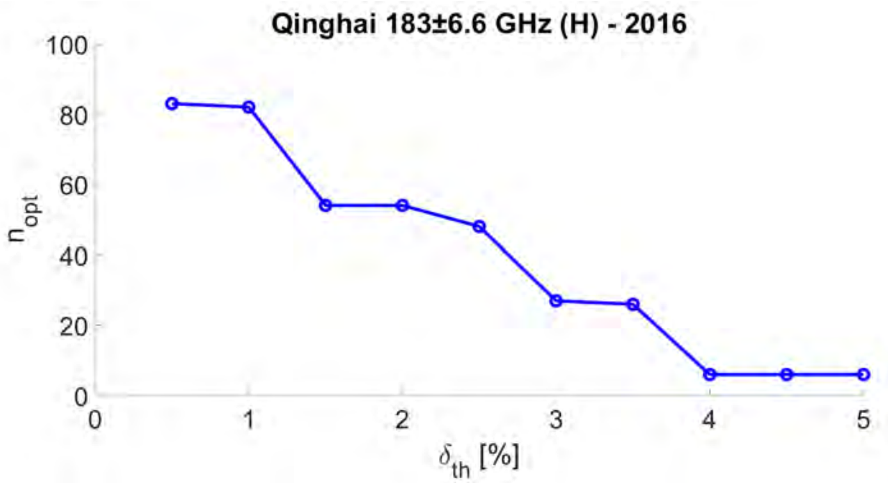



*Figure 4.6.2: Relation between the minimum number of necessary samples ($n_{opt}$) and $\delta_{th}$ threshold for Qinghai lake using SSMIS F17 at 183±6.6 GHz (H).*

Since $\delta_{th}$ is a percentage value, then the finite difference (in km) between $\varepsilon(n_{opt})$ and $m_\varepsilon$ and between $\sigma_\varepsilon(n)$ and $m_\sigma$ samples depends on the target, as shown in the following eq. (4.6.7) and (4.6.8):

$$\Delta\varepsilon = |\varepsilon(n_{opt}) - m_\varepsilon| = |\frac{\delta_{th}}{100} m_\varepsilon| \tag{4.6.7}$$

$$\Delta\sigma_\varepsilon = |\sigma_\varepsilon(n_{opt}) - m_\sigma| = |\frac{\delta_{th}}{100} m_\sigma| \tag{4.6.8}$$

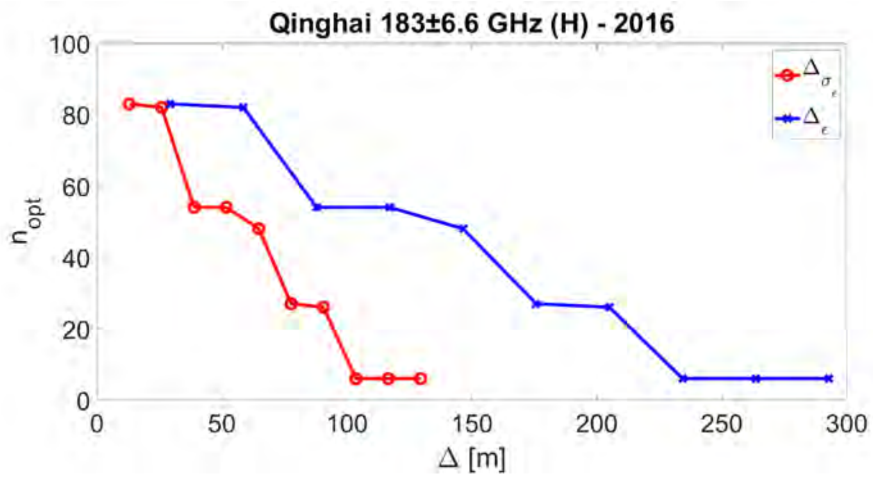



*Figure 4.6.3: Relation between the minimum number of necessary samples ($n_{opt}$) and $\Delta_\varepsilon$ in blue and between the minimum number of necessary samples ($n_{opt}$) and $\Delta_{\sigma_\varepsilon}$ for Qinghai lake using SSMIS F17 at 183±6.6 GHz (H).*

Below a similar approach is carried out for all other considered targets. In particular, grouping them depending on their different nature, e.g, lakes/bays, mountains, shelves and straits, it is possible to produce a set of three figures for each type of target, as shown in Fig. 4.6.4 and 4.6.5.

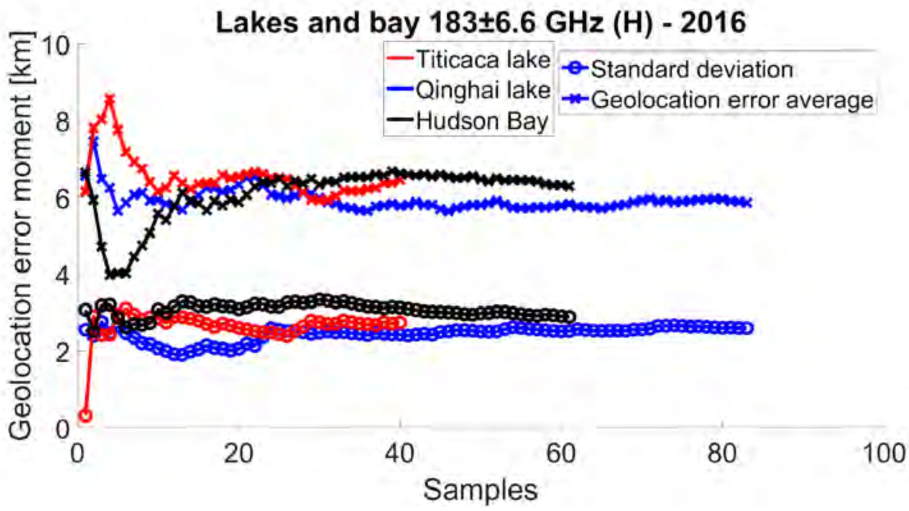



*Figure 4.6.4: Geolocation error average (x markers) and standard deviation (o markers) for Qinghai lake (red line), Titicaca lake (blue line) and Hudson bay (black line) using SSMIS F17 at 183±6.6 GHz (H).*

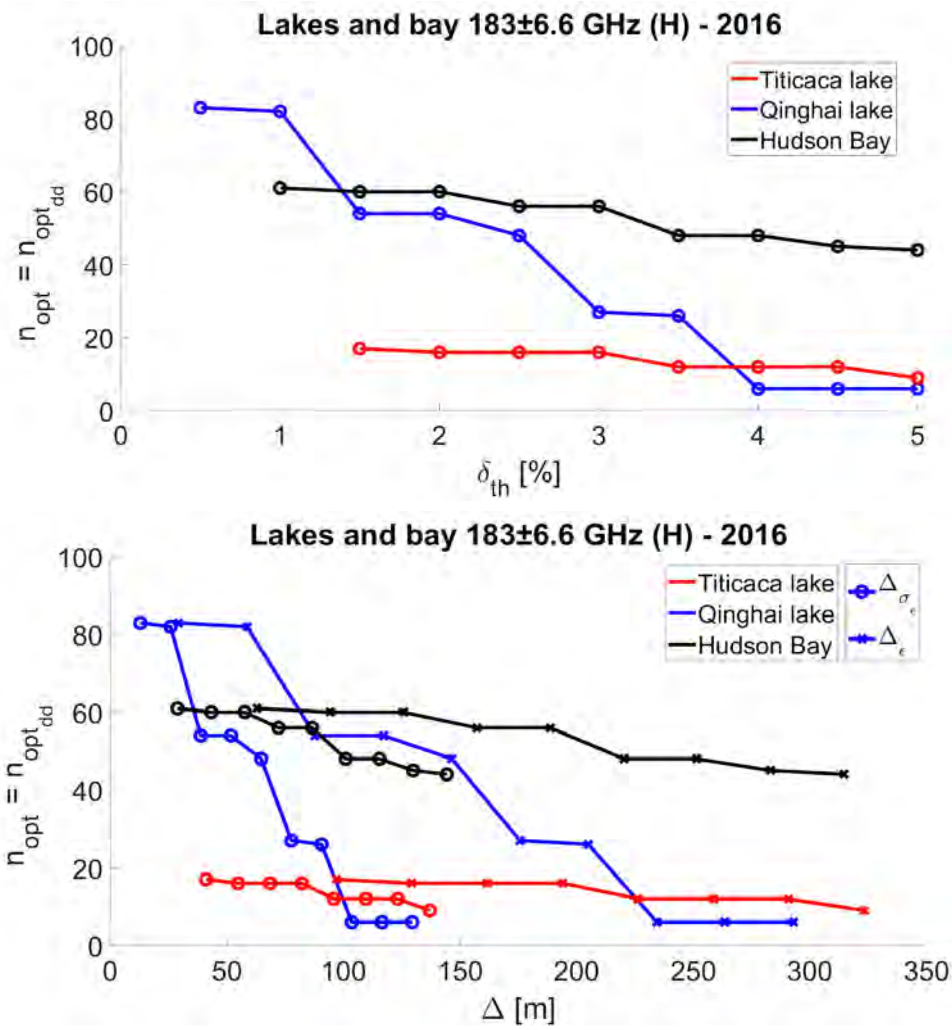



*Figure 4.6.5: (Top panel) Relation between the minimum number of necessary samples ($n_{opt}$) and $\delta_T$ threshold. (Bottom panel) Relation between the minimum number of necessary samples ($n_{opt}$) and $\Delta_\varepsilon$ with 'x' markers and between the minimum number of necessary samples ($n_{opt}$) and $\Delta_{\sigma_\varepsilon}$ with 'o' markers. The results are shown for Qinghai lake with red line, for Titicaca lake with blue line and for Hudson bay with black line.*

The following Fig. 4.6.6 and Fig. 4.6.7 report the results and the minimum number of necessary samples for Antarctic ice shelves.

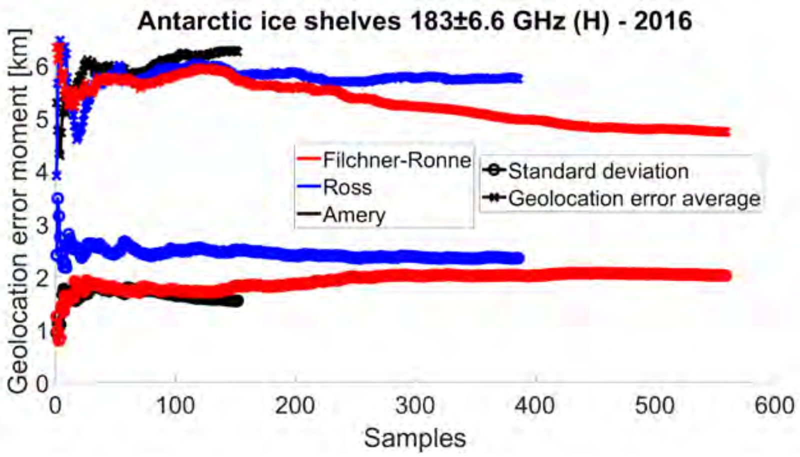



*Figure 4.6.6: Geolocation error average (x markers) and standard deviation (o markers) for Ross ice shelf (red line), Filchner-Ronne ice shelf (blue line) and Amery ice shelf (black line) using SSMIS F17 at 183±6.6 GHz (H).*

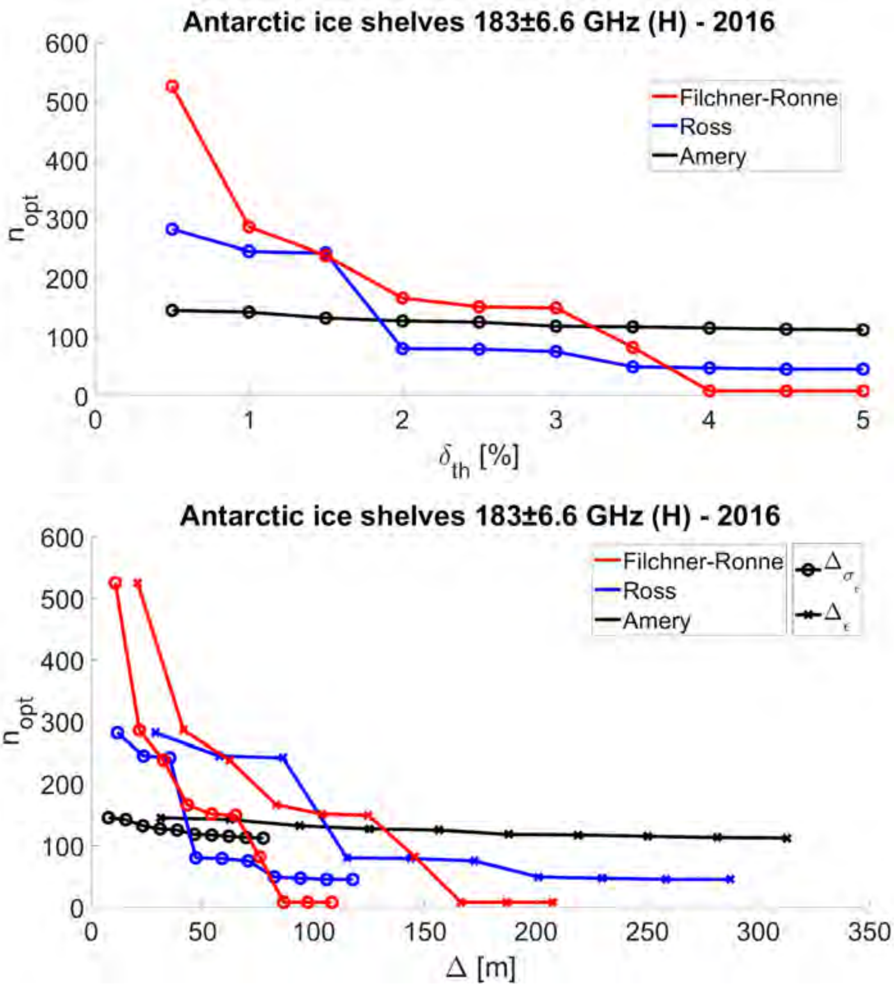
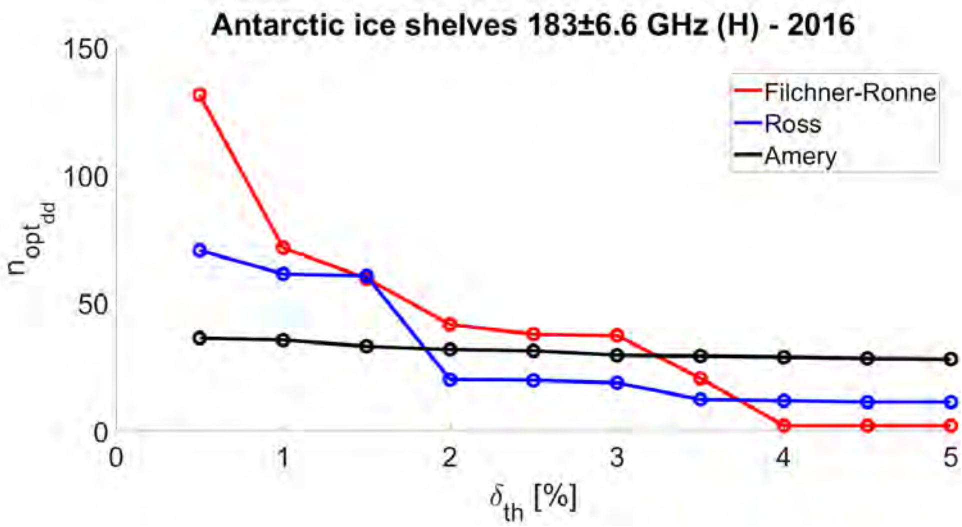



*Figure 4.6.7: (Top panel) Relation between the minimum number of necessary samples ($n_{opt}$) and $\delta_T$ threshold. (Bottom panel) Relation between the minimum number of necessary samples ($n_{opt}$) and $\Delta_\varepsilon$ with 'x' markers and between the minimum number of necessary samples ($n_{opt}$) and $\Delta_{\sigma_\varepsilon}$ with 'o' markers. The results are shown for Ross ice shelf with red line, for Filchner-Ronne ice shelf with blue line and Amery ice shelf with black line.*

Considering the eq. (4.6.6) with $n_{dd} = 4$ for ice shelves, it is possible to obtain the Fig. 4.6.8, that reports the minimum number of days necessary to obtain the convergence.



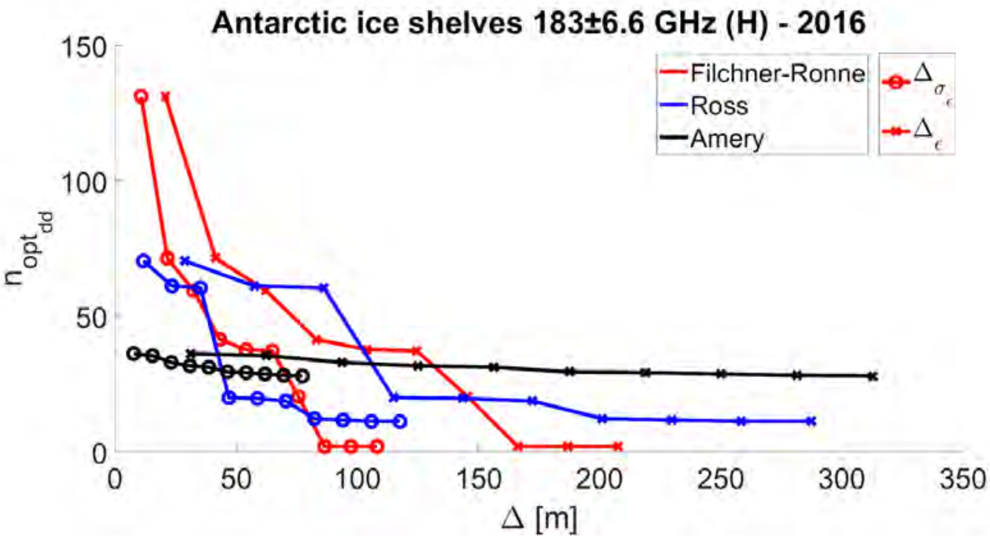




*Figure 4.6.8: (Top panel) Relation between the minimum optimal number $n_{opt\,dd}$ of necessary days and $\delta_T$ threshold. (Bottom panel) Relation between minimum optimal number $n_{opt\,dd}$ of necessary days and $\Delta_\varepsilon$ with 'x' markers and between minimum optimal number $n_{opt\,dd}$ of necessary days and $\Delta_{\sigma_\varepsilon}$ with 'o' markers. The results are shown for Ross ice shelf with red line, for Filchner-Ronne ice shelf with blue line and Amery ice shelf with black line.*

The following Fig. 4.6.9 and Fig. 4.6.10 report the results and the minimum number of necessary samples for Karakorum and Andean chains.

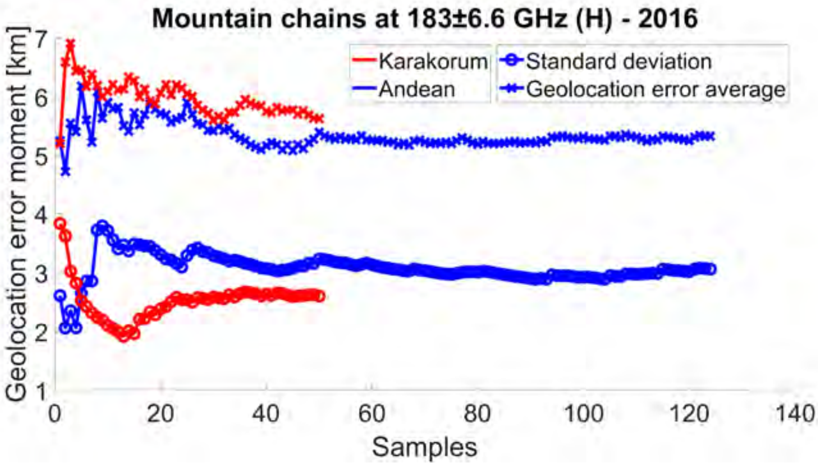



*Figure 4.6.9: Geolocation error average (x markers) and standard deviation (o markers) for Karakorum mountains (red line) and Andean chain (blue line) using SSMIS F17 at 183±6.6 GHz (H).*

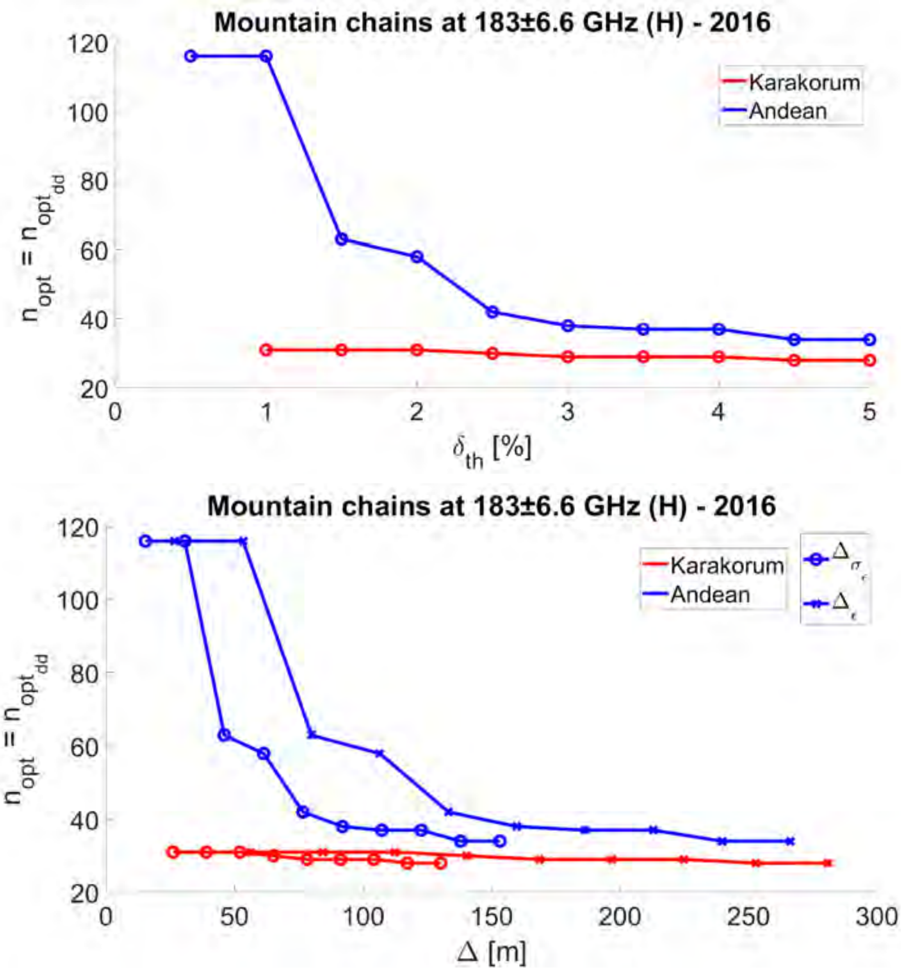



*Figure 4.6.10: (Top panel) Relation between the minimum number of necessary samples ($n_{opt}$) and $\delta_T$ threshold. (Bottom panel) Relation between the minimum number of necessary samples ($n_{opt}$) and $\Delta_\varepsilon$ with 'x' markers and between the minimum number of necessary samples ($n_{opt}$) and $\Delta_{\sigma_\varepsilon}$ with 'o' markers. The results are shown for Karakorum mountains with red line and Andean mountains with blue line.*

Fig. 4.6.11 and Fig. 4.6.12 report the results and the minimum number of necessary samples for Nares Strait.

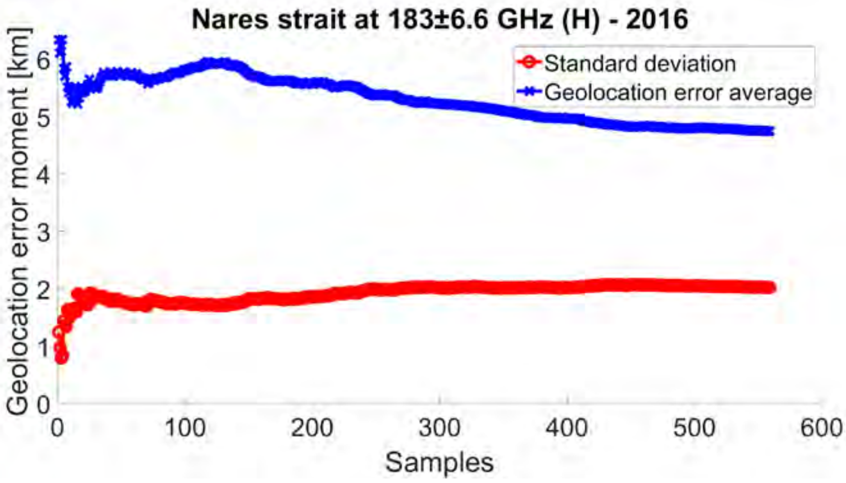



*Figure 4.6.11: Geolocation error average (x markers) and standard deviation (o markers) for Nares Strait using SSMIS F17 at 183±6.6 GHz (H).*

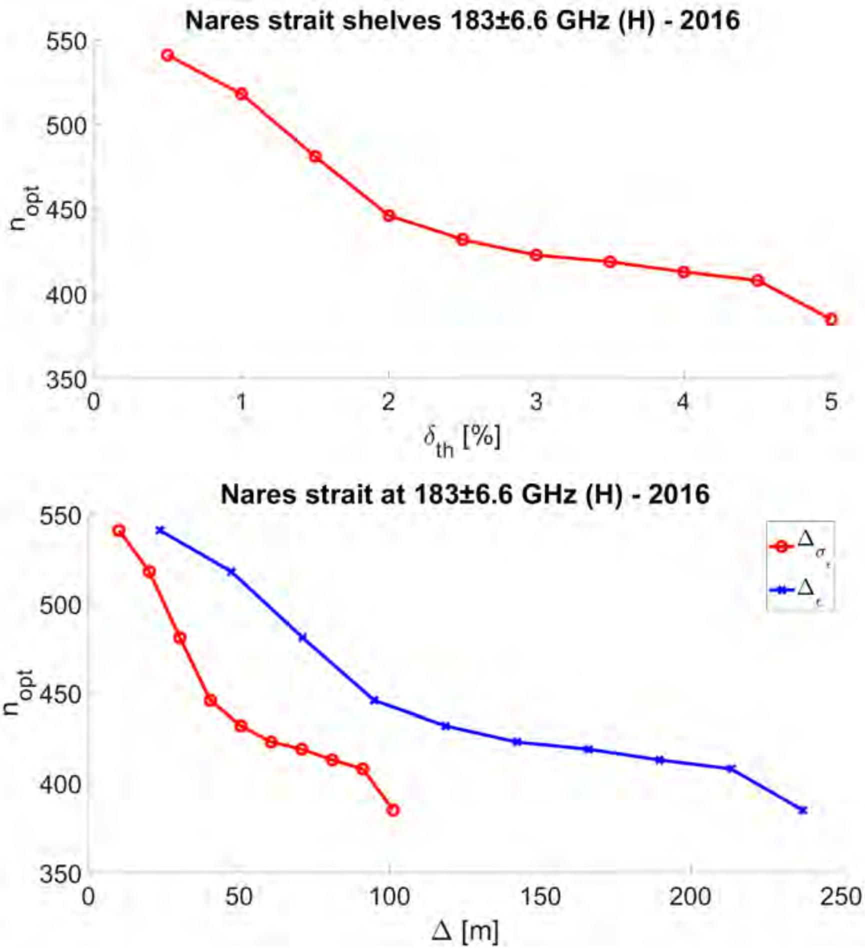



*Figure 4.6.12: (Top panel) Relation between the minimum number of necessary samples $(n_{opt})$ and $\delta_T$ threshold. (Bottom panel) Relation between the minimum number of necessary samples $(n_{opt})$ and $\Delta_\varepsilon$ with 'x' markers and between the minimum number of necessary samples $(n_{opt})$ and $\Delta_{\sigma_\varepsilon}$ with 'o' markers. The results are shown for for Nares Strait*

Considering the eq. (4.6.6) with $n_{dd} = 4$ for Nares Strait, it is possible to obtain the Fig. 4.6.13, that reports the minimum number of days necessary to obtain the convergence.

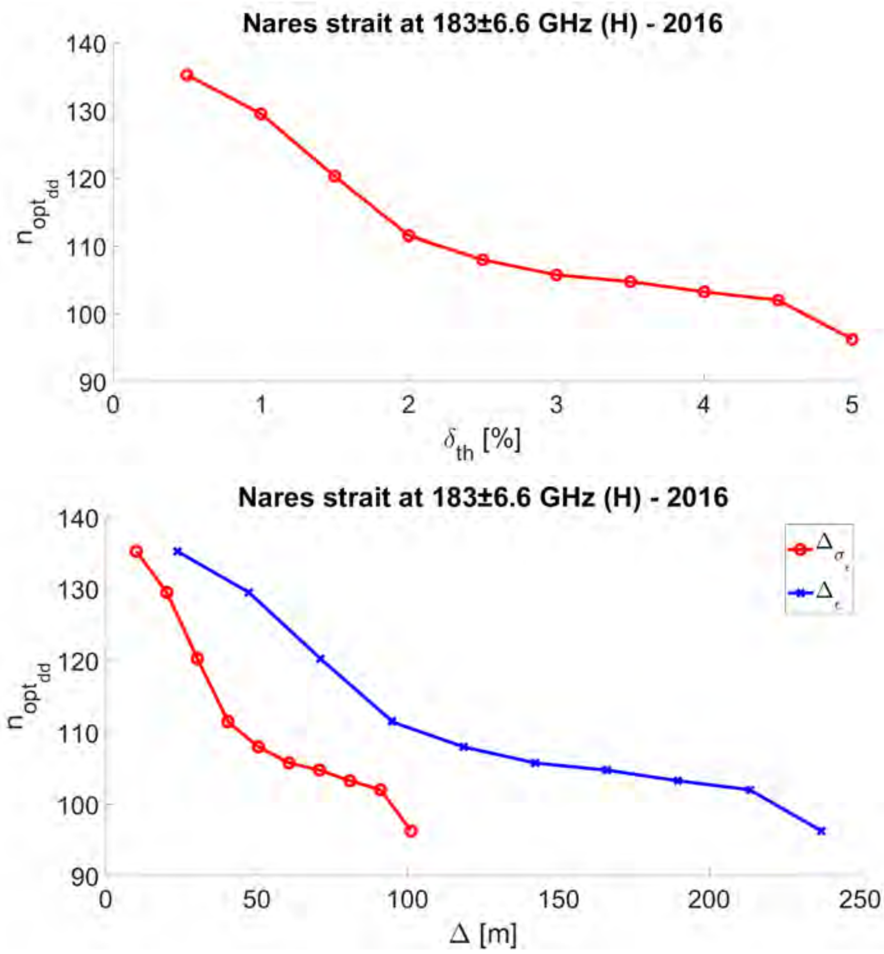



*Figure 4.6.13: (Top panel) Relation between the minimum optimal number $n_{opt\,dd}$ and $\delta_T$ threshold. (Bottom panel) Relation between the the minimum optimal number $n_{opt\,dd}$ and $\Delta_\varepsilon$ with 'x' markers and between the minimum optimal number $n_{opt\,dd}$ and $\Delta_{\sigma_\varepsilon}$ with 'o' markers. The results are shown for for Nares Strait*

In summary, the following Tab. 4.6.1 resumes $n_{opt}$ and $n_{optdd}$ obtained for all targets, setting the threshold $\delta_{th}$ to 1% and 2%.

| **Target** | $\delta_{th} = 1\%$ | | $\delta_{th} = 2\%$ | |
|---|---|---|---|---|
| | $n_{opt}$ | $n_{optdd}$ | $n_{opt}$ | $n_{optdd}$ |
| Qinghai lake | 82 | 82 | 54 | 54 |
| Karakorum mountains | 31 | 31 | 31 | 31 |
| Hudson Bay | 61 | 61 | 60 | 60 |

| | | | | |
|---|---|---|---|---|
| Nares strait | 518 | 130 | 446 | 112 |
| Ross ice shelf | 286 | 72 | 166 | 42 |
| Filchner-Ronne ice shelf | 245 | 62 | 80 | 20 |
| Amery ice shelf | 142 | 36 | 127 | 31 |
| Titicaca lake | - | - | 16 | 16 |

Note that in case of Titicaca lake for $\delta_{th} = 1\%$, we do not indicate $n_{opt}$ because we need all samples to respect this threshold.

# 5. ICI FREQUENCY-SCALING ASSESSMENT

In order to simulate the spaceborne ICI BT and slant-path attenuation, we have adopted a 1D radiative transfer model (with no scattering) using ERA-5 or radiosoundings (RAOB) as input atmospheric vertical profiles. In particular:

- ERA-5 data cover the Earth on a 30-km resolution grid and resolve the atmosphere using 137 levels from the surface up to a height of 80 km. The used atmospheric radiosoundings are distributed from University of Wyoming for many stations around the world. Tab. 5.1 summarizes the considered data sources.
- To obtain a more realistic simulated scenario, it is essentially to have a good representation of the surface emissivity. For this purpose we have adopted the surface-emissivity models TELSEM2 for water surface and TESSEM2 for land surface.

*Table 5.1: Summary of web sites to download atmospheric information*

| Data | Web Site |
|---|---|
| ERA-5 | https://cds.climate.copernicus.eu/#!/search?text=ERA5&type=dataset |
| Atmospheric soundings | http://www.weather.uwyo.edu/upperair/sounding.html |

## 5.1 Frequency scaling using SSMIS measured imagery

Considering the existing data provided from SSMIS F17 instrument it is possible to observe the differences between 150 GHz H and 183±6.6 GHz H. Fig. 5.1.1 shows that difference for Titicaca lake on 15/07/2016 and the BT contrast between land and water is, as expected, greater at 150 GHz. Also Fig. 5.1.2 and Fig. 5.1.3 confirm that observation.

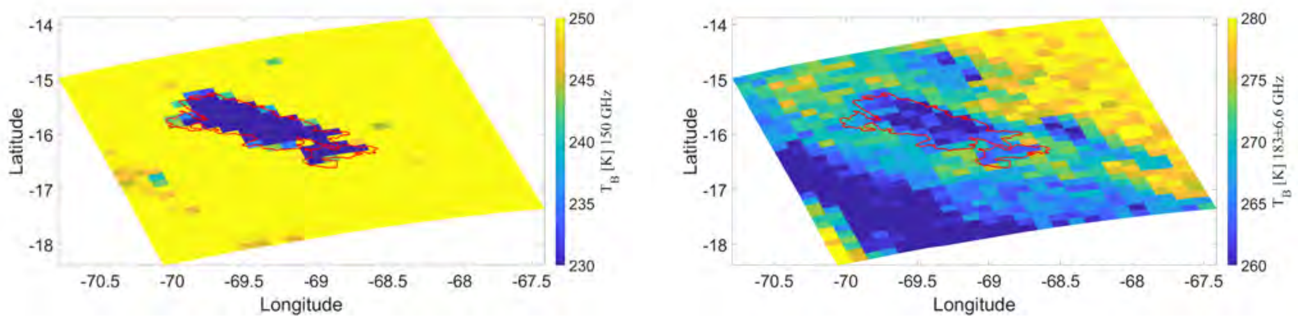



*Figure 5.1.1: BT over Titicaca lake on 15/07/2016 at 150 GHz (left) and 183 ±6.6 GHz (right).The red line represents the Titicaca lake from GSHHG database at full resolution.*

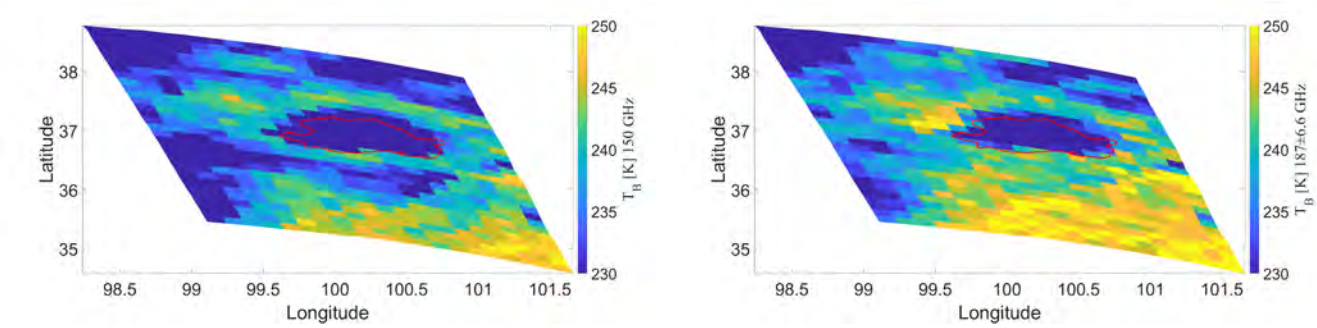



*Figure 5.1.2: BT over Qinghai lake on 11/11/2016 at 150 GHz (left) and 183 ±6.6 GHz (right).The red line represents the Qinghai lake from GSHHG database at full resolution.*

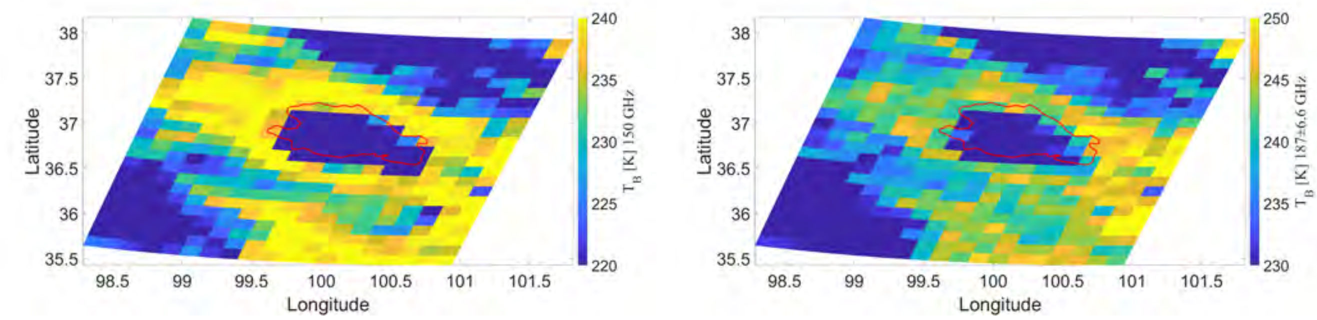



*Figure 5.1.3: BT over Qinghai lake on 30/11/2016 at 150 GHz (left) and 183 ±6.6 GHz (right).The red line represents the Qinghai lake from GSHHG database at full resolution.*

It is very useful to observe SSMIS-8 (150 GHz in horizontal polarization), because it can be used also to have an idea what we will see with ICI-4 (H) as will be explained in the next Sub-sec. 5.2 and 5.3.

## 5.2 Frequency scaling simulation using radiosounding profiles

Using atmospheric sounding it is possible to simulate BT in a single point, for example wanting to simulate the Antarctic area we can choose a station in Fig. 5.2.1.

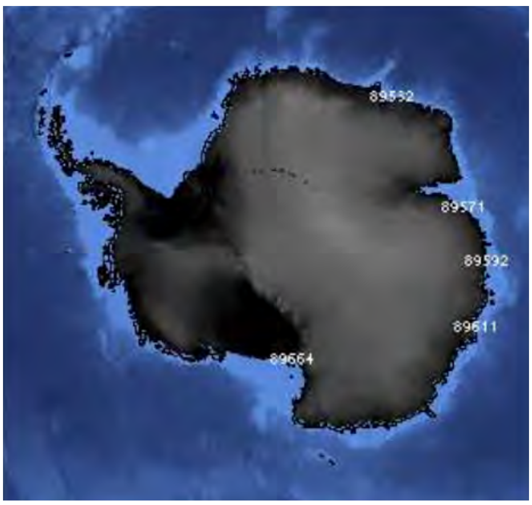



*Figure 5.2.1: Available weather stations on Antarctica region which atmospheric sounding*

Fig. 5.2.1 shows that a weather station is located within Ross ice shelf, i.e. the station 89664 of McMurdo. Using data available for this station, it is possible to simulate spaceborne BTs for different ICI channel frequencies. The numerical simulations in this case are provided setting a constant surface emissivity.

Fig. 5.2.2 reports the relation between 150 GHz and 243 GHz for simulations on McMurdo station from 01 June 2016 to 31 October 2016. Within this period we have almost two radiosoundings per day, obtaining a dataset containing 299 samples.

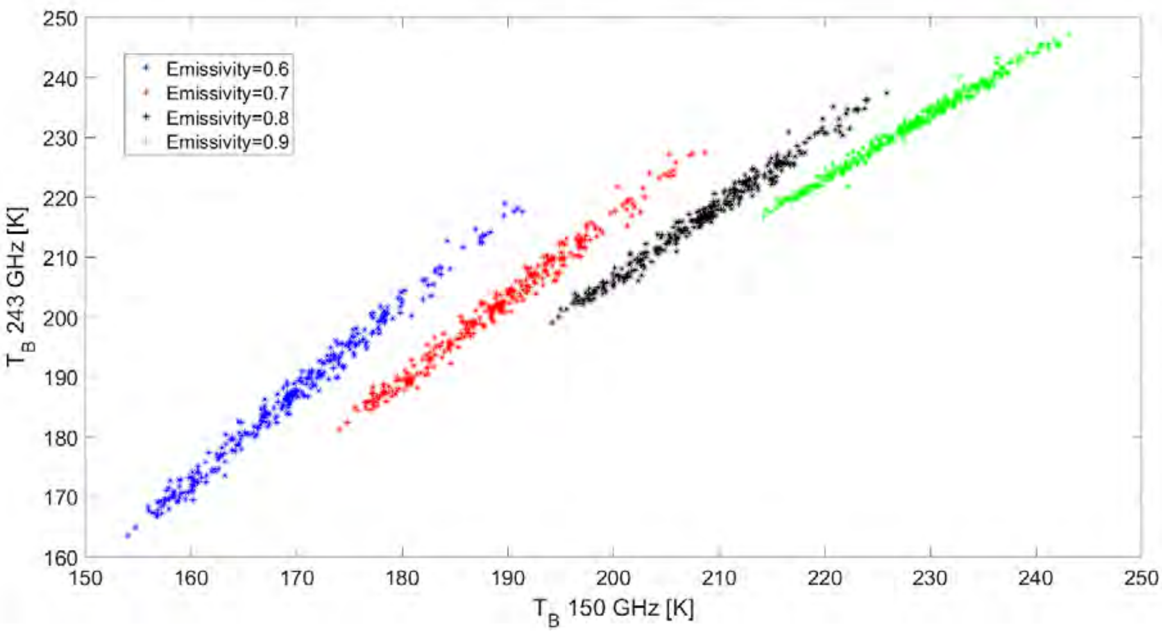



*Figure 5.2.2: Relation between 150 GHz and 243 GHz for simulations on McMurdo station from 01 June 2016 to 31 October 2016. Blue markers are the results for 0.6 of surface emissivity. Red points indicate surface emissivity of 0.7. Black markers are the results for surface emissivity of 0.8. Green markers are the results for 0.9 of surface emissivity at 53° view angle.*

Fig. 5.2.2 shows a linear relation between 150 GHz and 243 GHz, especially for lower values of surface emissivity. Tab. 5.2.1. Contains the equations of the linear regression that allow to approximate 243 GHz from 150 GHz.

*Table 5.2.1: BT frequency-scaling relation from 150 GHz and 243 GHz for several surface emissivity*

| **Emissivity** | **Equation (BT in K)** |
|---|---|
| 0.6 | $T_{B_{243}} = 1.5\ T_{B_{150}} - 65$ |
| 0.7 | $T_{B_{243}} = 1.4\ T_{B_{150}} - 56$ |
| 0.8 | $T_{B_{243}} = 1.2\ T_{B_{150}} - 38$ |
| 0.9 | $T_{B_{243}} = 1.1\ T_{B_{150}} - 11$ |

Fig. 5.2.3 and Fig. 5.2.4 show the relation between 183 GHz and 243 GHz and between 150 GHz and 183 GHz, respectively. No close to linear relationships were found between these pair of channels.

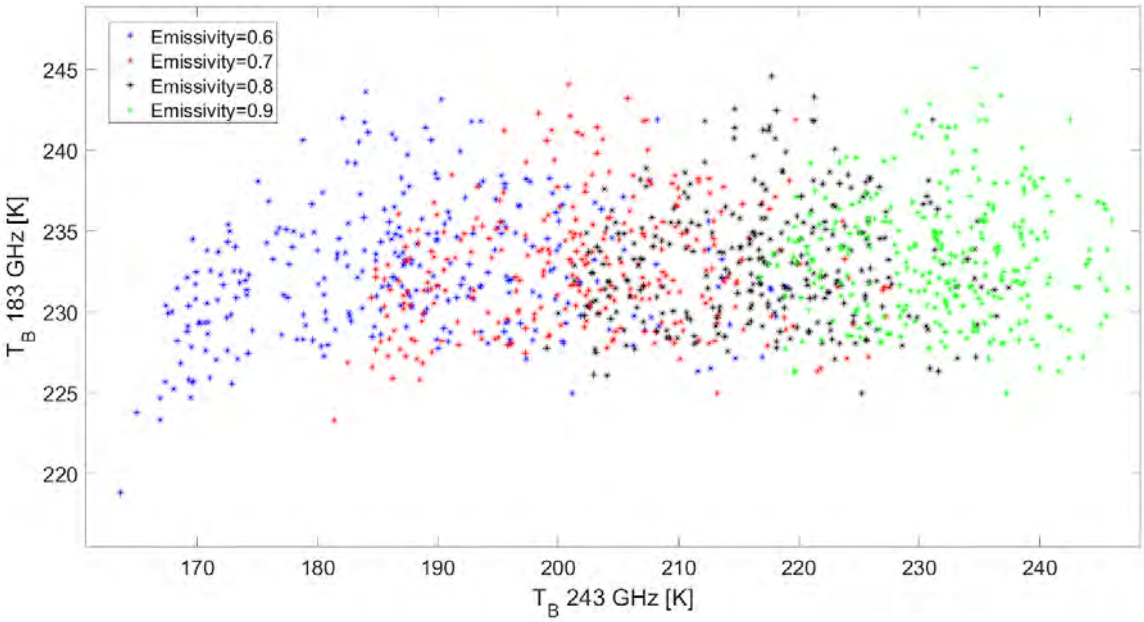



*Figure 5.2.3: Relation between 183 GHz and 243 GHz for simulations on MCMurdo station from 01 June 2016 to 31 October 2016. Blue markers are the results for 0.6 of surface emissivity. Red points indicate surface emissivity of 0.7. Black markers are the results for surface emissivity of 0.8. Green markers are the results for 0.9 of surface emissivity at 53° view angle.*

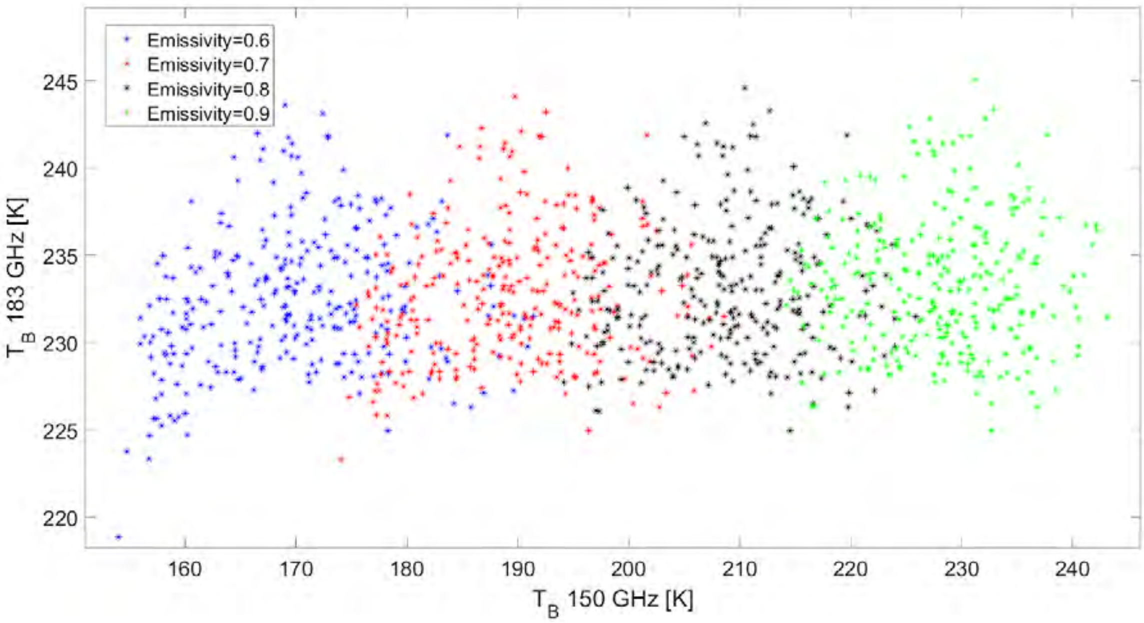



*Figure 5.2.4: Relation between 150 GHz and 183 GHz for simulations on MCMurdo station from 01 June 2016 to 31 October 2016. Blue markers are the results for 0.6 of surface emissivity. Red points indicate surface emissivity of 0.7. Black markers are the results for surface emissivity of 0.8. Green markers are the results for 0.9 of surface emissivity at 53° view angle.*

Fig. 5.2.5 shows the attenuation for 150 GHz, 183 GHz and 243 GHz.

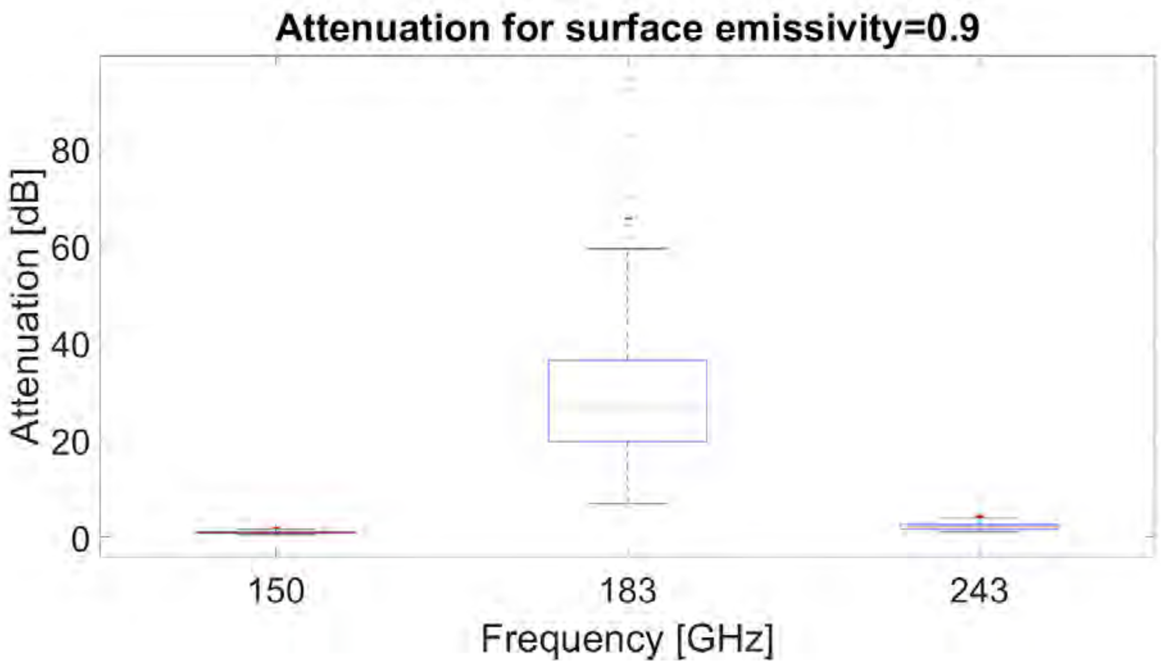



*Figure 5.2.5: Simulated attenuation obtained using radiosaudings over MCMurdo station for an emissivity of 0.9 at 53° view angle.*

The attenuation of 243 GHz is higher with respect to 150 GHz, as highlighted in Fig. 5.2.6. However, the attenuation at 243 GHz is lower than at 183 GHz, so it will allow to see more surface targets with higher BT contrast at 243 GHz.

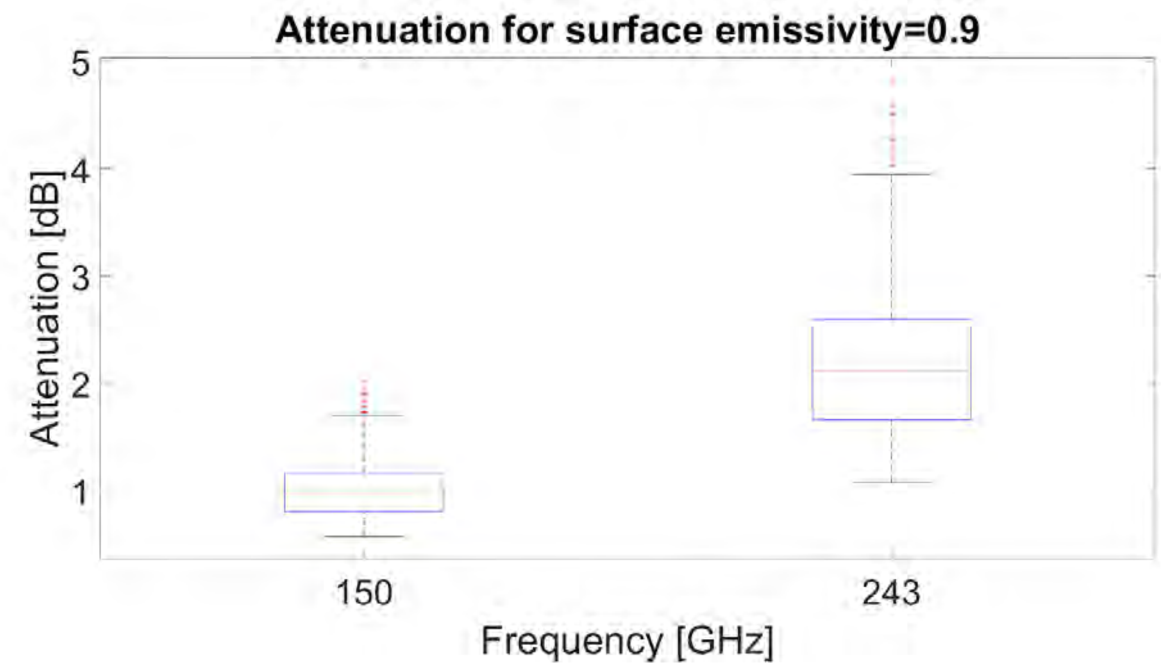



*Figure 5.2.6: Simulated attenuation obtained using radiosaudings over MCMurdo station for an emissivity of 0.9 at 53° view angle*

### 5.3 Frequency scaling simulation using ERA-5 clear-air scenarios

Using ERA-5 data as input for atmospheric vertical profiles, it is possible to simulate an entire scene and its BT map. In this respect, it is very import to properly characterize the surface emissivity knowing its nature as well as the observation geometry and central frequency. Input information to TELSEM2 and TESSEM2 recent and state-of-the-art models, described in [20], are listed in Tab. 5.3.1.

*Tab. 5.3.1: Information needed to compute the emissivity for land and water surface*

| **Type of surface** | **Input data** |
|---|---|
| Land (TELSEM2) | Viewing angle of view, frequency, geographical coordinates |
| Water (TESSEMS2) | Viewing angle, frequency, surface wind velocity, surface temperature, water salinity |

Providing the input data of Tab. 5.3.1 for the Qinghai lake at 183 GHz, we can simulate the surface emissivity as in Fig. 5.3.1 both at horizontal and vertical polarization using ERA5 on 1 December 2016 at 11:00 am.

Using surface emissivity maps of Fig, 5.3.1, it is then possible to simulate spaceborne BTs and the slant-path attenuation at 183±7 GHz, both at horizontal polarization (as the case of SSMIS) and at vertical polarization (as the case of ICI). For the ICI-1 at 183±7 GHz results for the BT and slant-path attenuation are shown in Fig. 5.3.2.

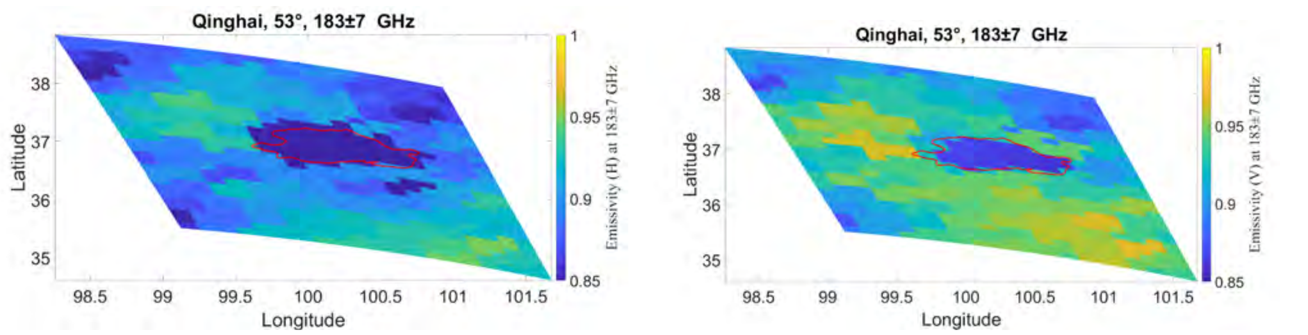
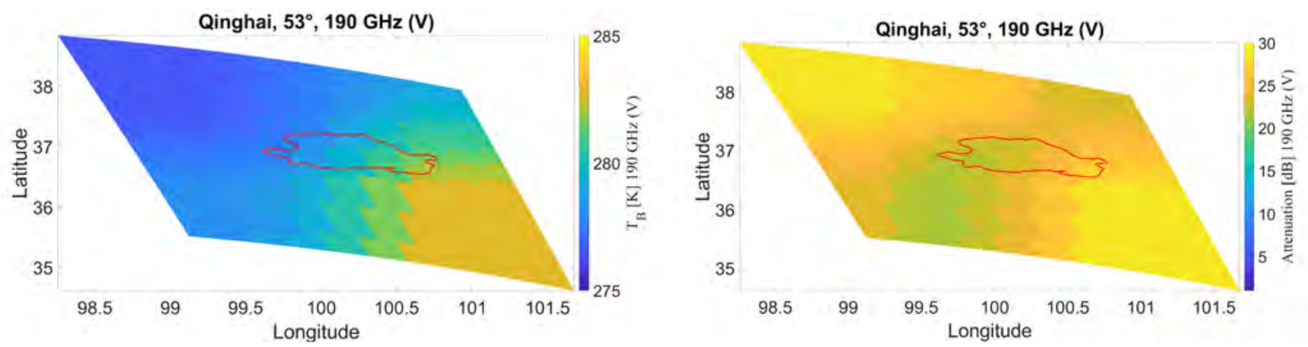



*Figure 5.3.1: Simulated surface emissivity at 183±7 GHz using TELSEM2 and TESSEM2. The red line represents the Qinghai lake from GSHHG database at full resolution.*



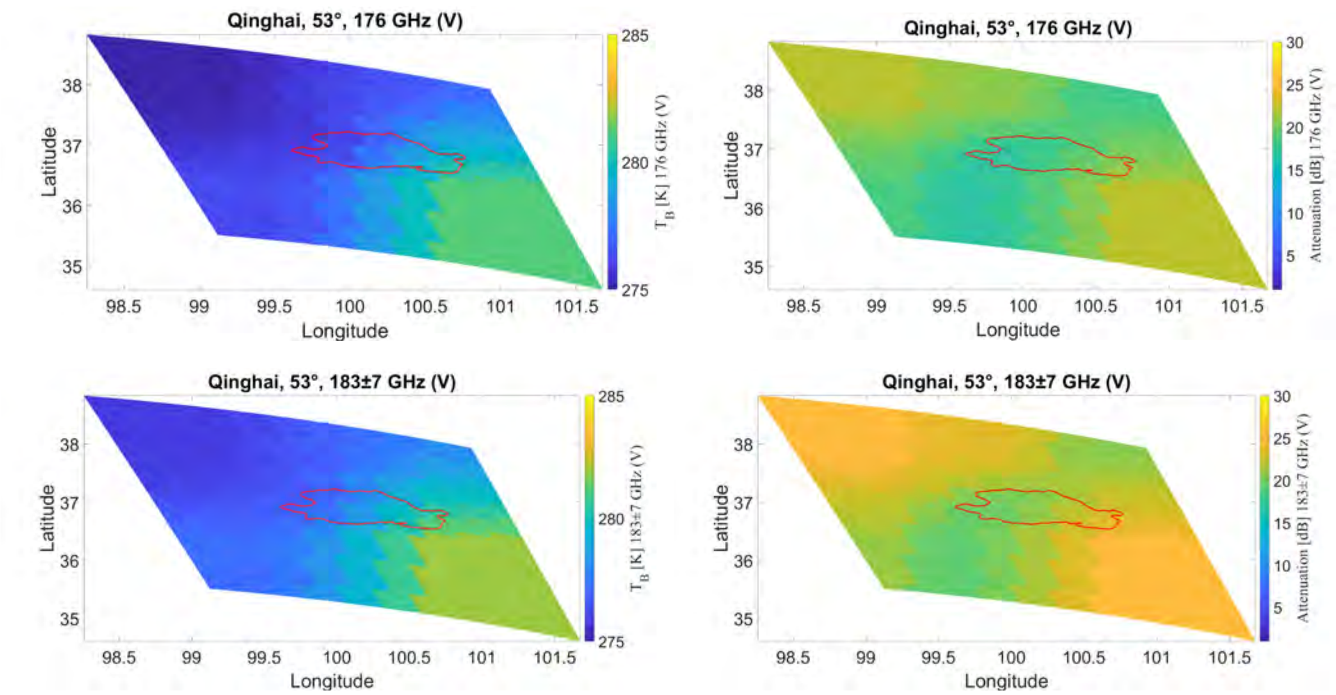



*Figure 5.3.2: Simulated brightness temperature and attenuation along slant direction at 190 GHz (top), at 176 GHz (center) and in 183±7 GHz (bottom) in vertical polarization. The red line represents the Qinghai lake from GSHHG database at full resolution.*

Fig 5.3.3 shows the simulation at 183±7 GHz (H) in the left and in the right there is a real image of SSMIS F17 at 183±6.6 GHz (H) (CSU_SSMIS_FCDR_V01R01_F17_D20161201_S1117_E1259_R51989).

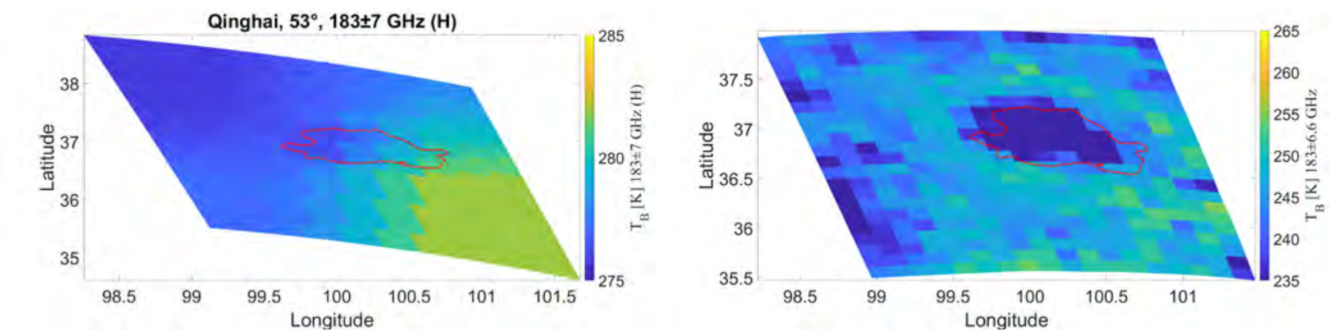



*Figure 5.3.3: Simulated brightness temperature along slant direction at 183±7 GHz (left) in horizontal polarization and real image of SSMIS F17 at 183±6.6 GHz (H). The red line represents the Qinghai lake from GSHHG database at full resolution.*

Note that both simulations at 183±7 GHz in vertical and horizontal polarization do not see the surface, differently from the real case in which SSMIS provided BT contrast for Qinghai lake.

Considering the ICI-4 channel at 243±2.5 GHz, results are reported in Fig. 5.3.4 both for horizontal and vertical polarizations. ICI-4 has both V than H polarization and according to Fig. 5.3.4 the H-polarization channel has more BT contrast over the lake as compared to the V-polarization one.

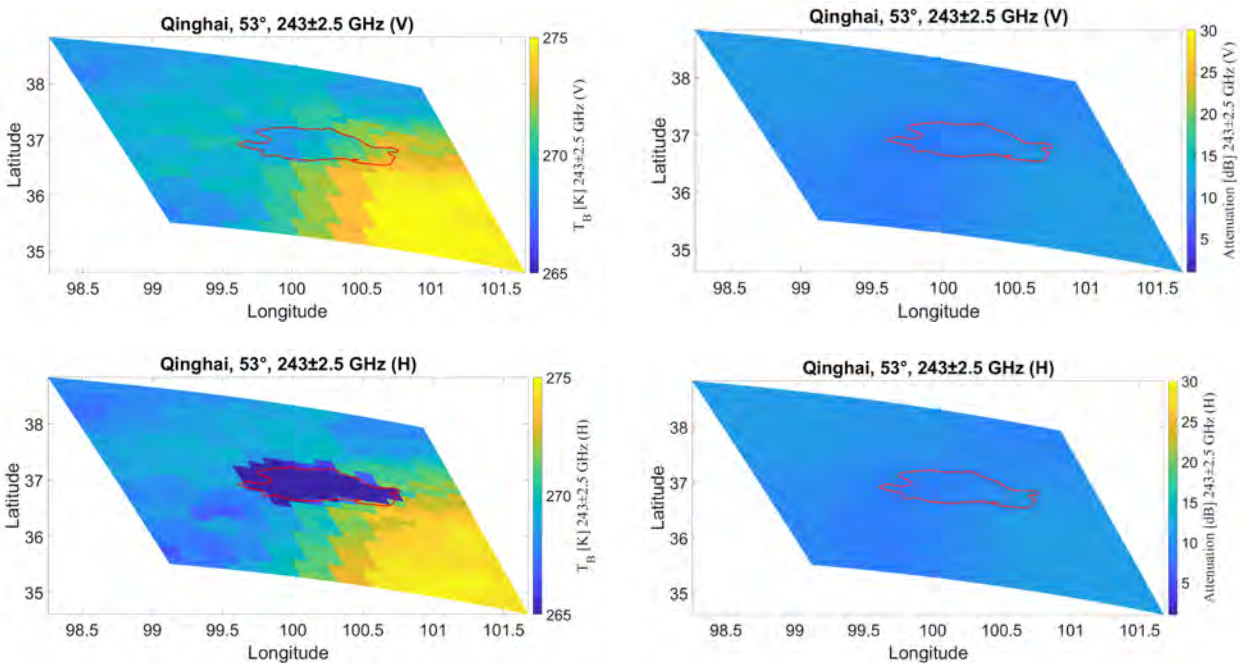



*Figure 5.3.4: Simulated brightness temperature and attenuation along slant direction at 243.5 GHz in vertical polarization (top) and in horizontal polarization (bottom). The red line represents the Qinghai lake from GSHHG database at full resolution.*

Fig. 5.3.4 shows that the landmark targets would be, as expected, more visible with horizontal polarization of ICI-4 channel.

## 5.4 Simulating brightness contrast and atmospheric water vapour effects

In this subsection we focus our attention on the atmospheric water vapour content and its relation with landmark targets visibility using the concept of the absolute BT contrast $\Delta T_B$, defined as:

$$\Delta T_B(f) = |T_B(e_{s1}, f) - T_B(e_{s2}, f)| \tag{5.4.1}$$

being $T_B$ is the BT simulated at the top-of-atmosphere (TOA) and $f$ the channel frequency, whereas $e_{s1}$ and $e_{s2}$ are the surface emissivities of the contiguous objects (e.g., land surface, sea water, ice shelf, lake water) characterizing each geolocation targets.

The following figures show the BT contrast considering different values of surface emissivity for 183±7 GHz and 243±2.5 GHz, using radiosounding over McMurdo station from 01 June 2016 to 31 October 2016. Fig. 5.4.1 reports the BT contrast using a constant arbitrary surface emissivity of 0.8 and 0.5, whereas Fig 5.4.2 shows the BT difference using a constant arbitrary surface emissivity of 0.9 and 0.8. If the surface emissivity difference decreases, consequently the BT contrast decreases, whereas, if the integrated water vapour increases, the BT contrast decreases more at 183 GHz than at 243 GHz. Setting a fixed value of surface emissivity at 243±2.5 GHz, we can obtain higher values of BT contrast with respect to the 183±7 GHz one so that we can expect a better landmark BT contrast for ICI-4.

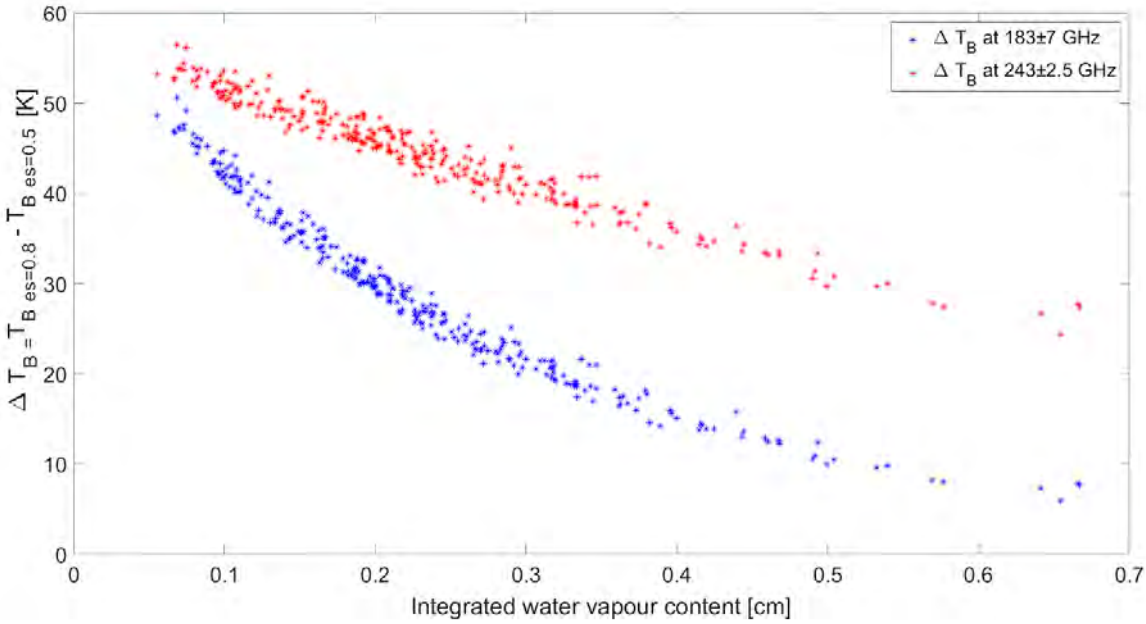



*Figure 5.4.1: Simulated brightness temperature contrast considering landmark target with surface emissivity of 0.8 and 0.5, considering radiosounding over McMurdo station from 01 June 2016 to 31 October 2016. Red markers represent the 245±2.5 GHz and blue points are the simulation at 183±7 GHz.*

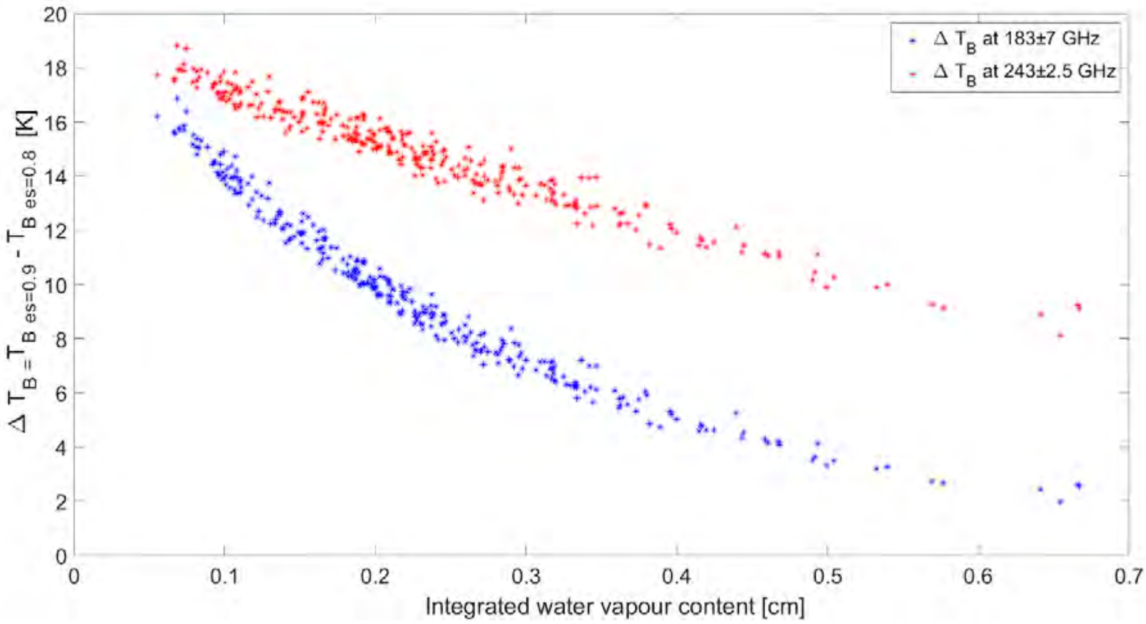



*Figure 5.4.2: Simulated brightness temperature contrast considering landmark target with surface emissivity of 0.9 and 0.8, considering radiosounding over McMurdo station from 01 June 2016 to 31 October 2016. Red markers represent the 245±2.5 GHz and blue points are the simulation at 183±7 GHz*

To simulate the behavior of the expected ICI BT contrast, we can similarly plot $\Delta T_B\,(f)$ using the other ICI frequencies (see Tab. 1.1), as shown in Fig. 5.4.3 and Fig. 5.4.4. Note that we have also added the results at 150 GHz for comparison as this central frequency is available for SSMIS.

As expected, the BT contrast decreases as the frequency increases up to values of less than 5 K at 664 GHz, whereas $\Delta T_B$ is larger at 150 GHz with respect to 183.3±7 GHz one which in turn is smaller than

243.2±2.5 GHz one. This means that ICI-4 exhibits an appealing potential for geolocation assessment. By reducing the surface emissivity contrast from 0.2 down to 0.05 (see Fig. 5.4.3 versus 5.4.4 and Fig. 5.4.5), the BT contrast is reduced if the atmosphere is unchanged.

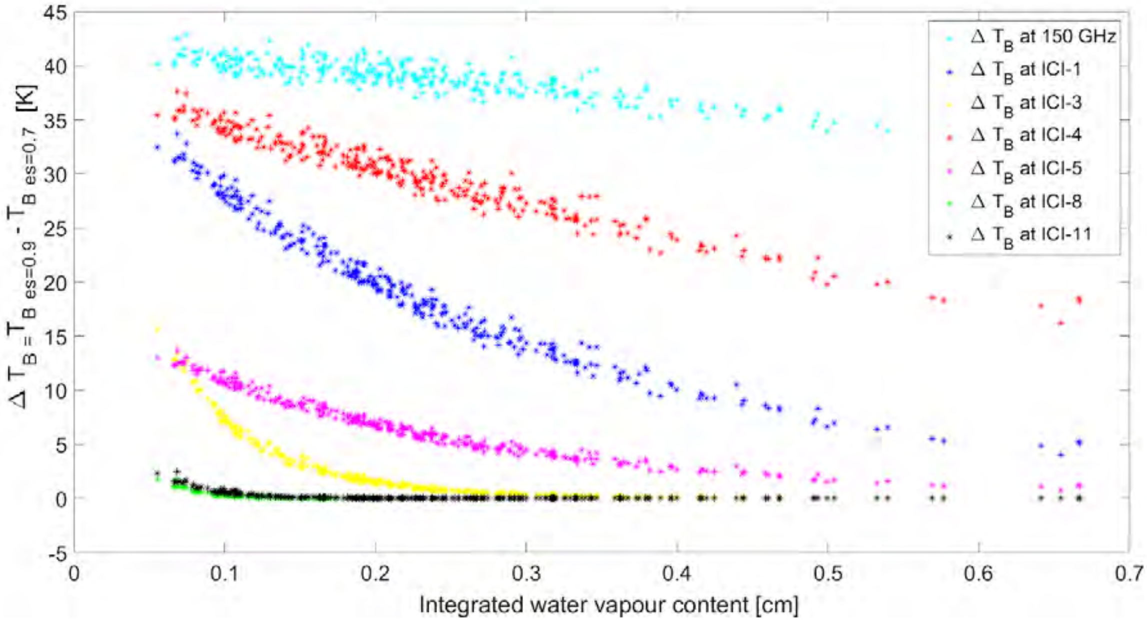



*Figure 5.4.3: Simulated brightness temperature contrast considering landmark target with surface emissivity of 0.9 and 0.7, considering radiosounding over McMurdo station from 01 June 2016 to 31 October 2016 at several ICI frequencies and 150 GHz.*

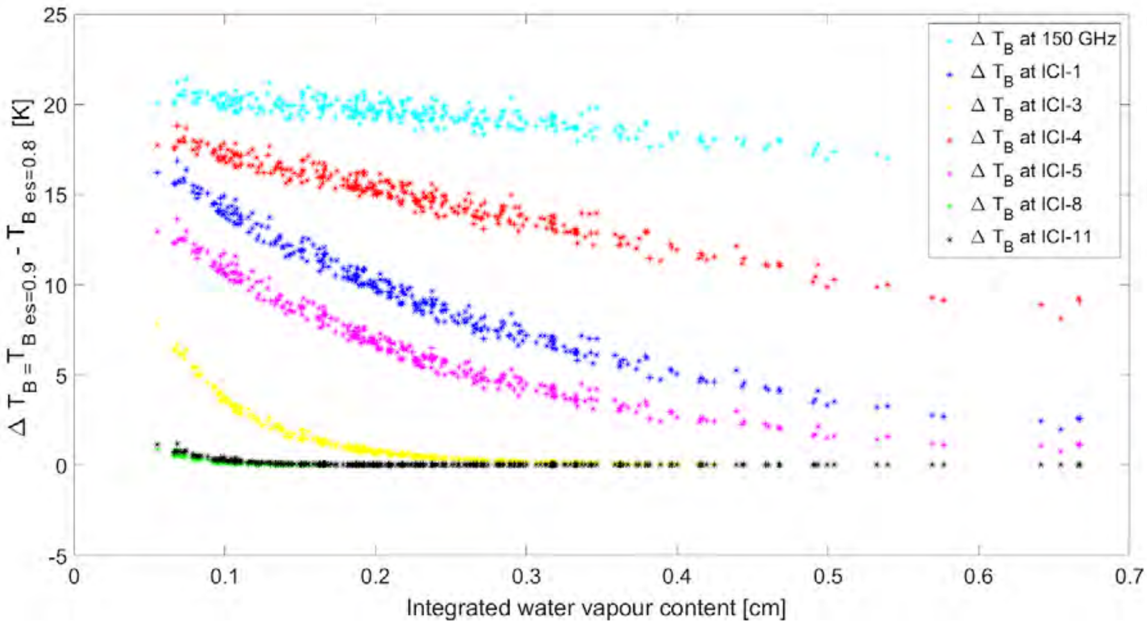



*Figure 5.4.4: Simulated brightness temperature contrast considering landmark target with surface emissivity of 0.9 and 0.8, considering radiosounding over McMurdo station from 01 June 2016 to 31 October 2016 at several ICI frequencies and 150 GHz.*

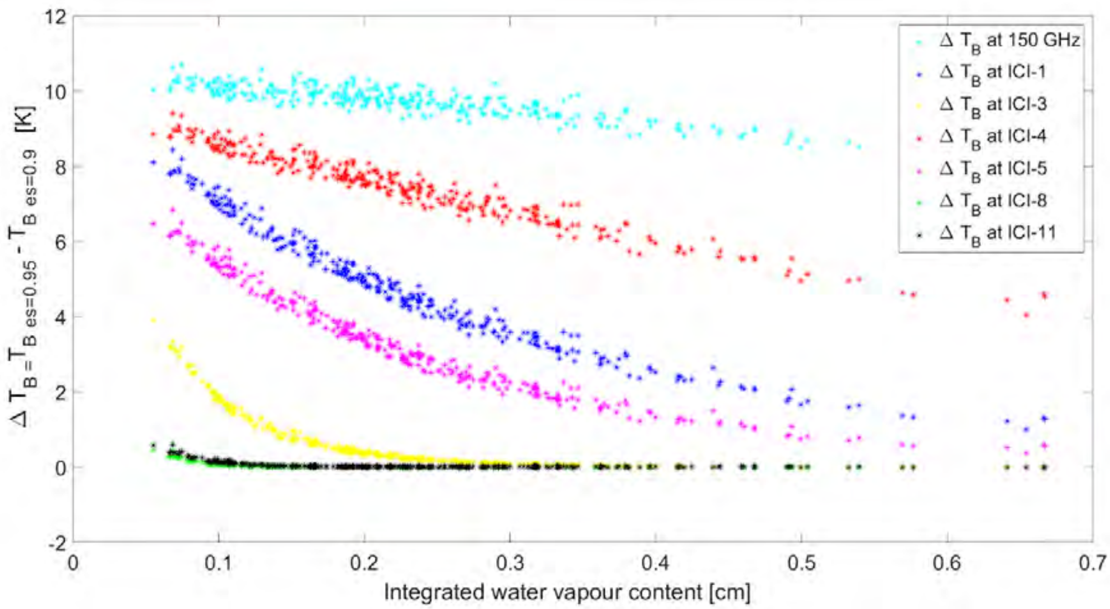



*Figure 5.4.5: Simulated brightness temperature contrast considering landmark target with surface emissivity of 0.95 and 0.9, considering radiosounding over McMurdo station from 01 June 2016 to 31 October 2016 at several ICI frequencies and 150 GHz.*

In order to simulate more realistic values of BT contrast between water and land, instead of assuming them arbitrarily constant, we can compute the surface emissivity with the TELSEM2 and TESSEM2 numerical models over the Qinghai lake, obtaining the mean values of surface emissivities, shown in Table 3. Note that TELSEM2 provides a constant value of land surface emissivity above 89 GHz, and ICI-1, ICI-5 and ICI-8 channels will have only the vertical polarization.

*Table 3: Surface emissivity for water and lake averaged over Qinghai lake provided by TELSEM2 and TESSEM2 for several frequencies*

| ICI channels | Frequency [GHz] | Land (H) | Sea water (H) | Land (V) | Sea water (V) |
|---|---|---|---|---|---|
| ICI-1 | 183±7 | *No channel available* | | 0.93 | 0.87 |
| ICI-4 | 243±2.5 | 0.89 | 0.57 | 0.93 | 0.90 |
| ICI-5 | 325±9.5 | *No channel available* | | 0,93 | 0.93 |
| ICI-8 | 448±7.2 | *No channel available* | | 0.93 | 0.95 |
| ICI-11 | 664±4.2 | 0.89 | 0.68 | 0.93 | 0.96 |

The atmospheric conditions in the Qinghai lake region have been extracted from the RAOB station n. 44373, highlighted in Fig. 5.4.6. This station is the closest to the Qinghai lake target area and the most similar in terms of annual climatology.


*Figure 5.4.6: RAOB station n. 44373 (red circle), available from Wyoming University over Southeast Asia. The blue spot is the region of Qinghai lake.*

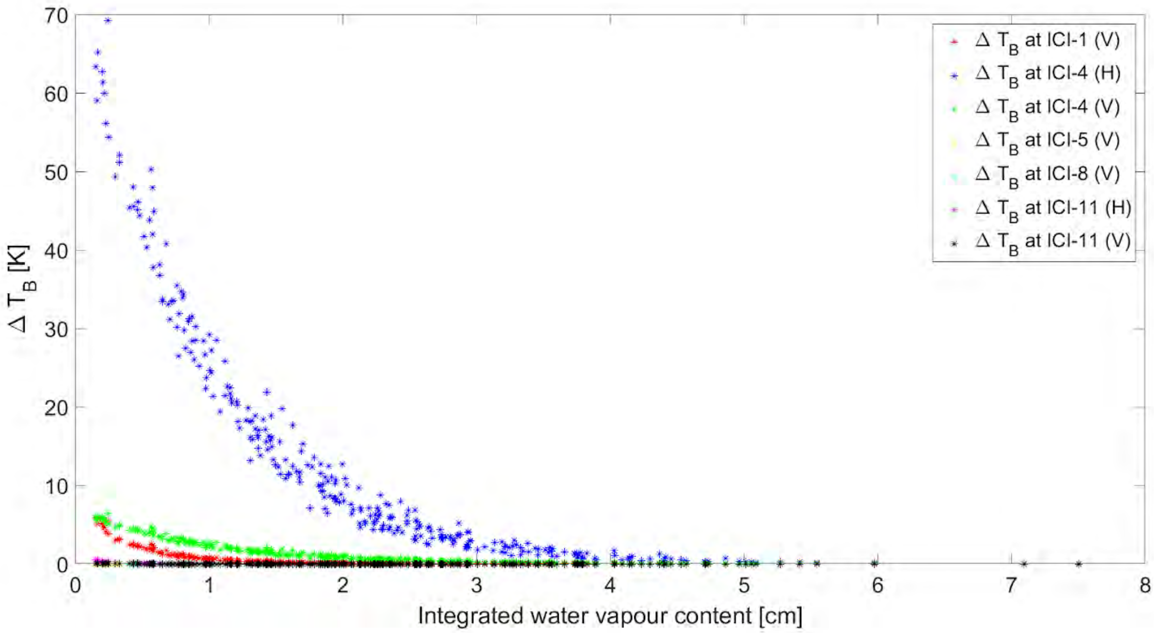



*Figure 5.4.7: Simulated brightness temperature contrast considering landmark target with surface emissivity from TELSEM2 and TESSEM2, considering radiosounding over 44373 (Asia) station from 01 June 2016 to 31 October 2016 at several ICI channels.*

Fig. 5.4.7 confirms what previously discussed and, in particular that, in case of targets with water/land coastline (e.g., Qinghai lake), we can expect a higher value of BT contrast using ICI-4 at horizontal polarization than using ICI-1.

Water vapour content depends on climate conditions and seasons. Following the Köppen geo-climatic classification, there are 5 main groups of climate regions, as shown in Fig. 5.4.8 [21]:

- A (tropical)
- B (dry)
- C (temperate)
- D (continental)
- E (polar)

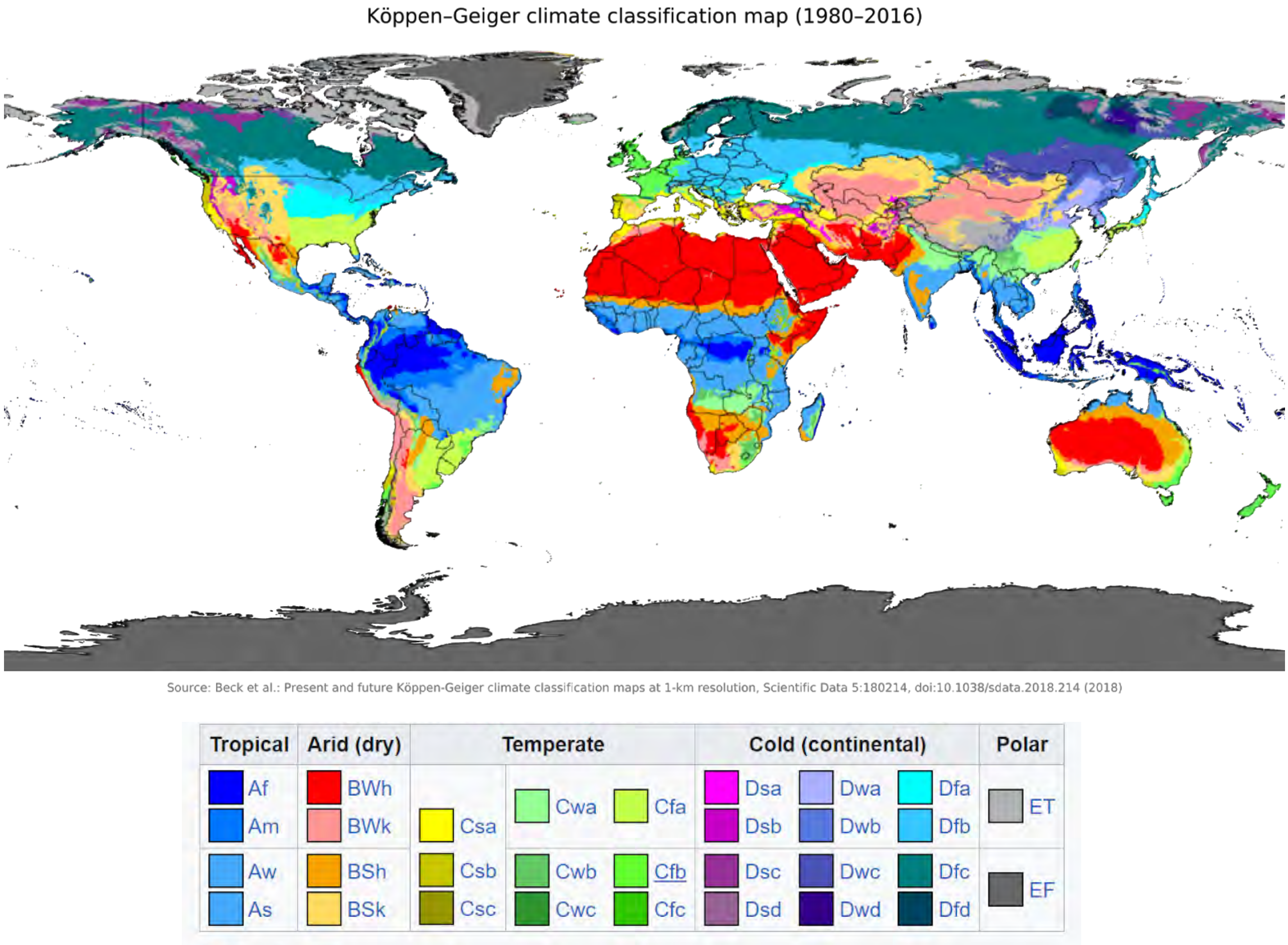



*Figure 5.4.8: Köppen-Geiger climate classification map [21]*

Selecting a RAOB station for each Köppen climate region, we can characterize the BT contrast against the integrated water vapour content using RAOB data of year 2016 (the same year of SSMIS satellite data). Tab. 5.4.2 shows the chosen RAOB stations from the University of Wyoming database with the corresponding number of available samples.

*Table 5.4.2: RAOB stations for the five climate regions following Köppen climate classification*

| **Climate region** (Köppen classification**)** | **RAOB station code** | **Region** (Wyoming website) | **RAOB station latitude** | **RAOB station longitude** | **Number of atmospheric radiosounding** |
|---|---|---|---|---|---|
| A (tropical) | 82917 | *South America* | -10.00 | -67.80 | 552 |

| | | | | | |
|---|---|---|---|---|---|
| B (Arid-dry) | 94461 | *South Pacific* | -25.03 | 128.28 | 356 |
| C (temperate) | 03953 | *Europe* | 51.93 | -10.25 | 728 |
| D (Cold-continental) | 30715 | *Southeast Asia* | 52.48 | 103.85 | 721 |
| E (polar) | 89664 | *Antarctica* | -77.85 | 166.66 | 721 |

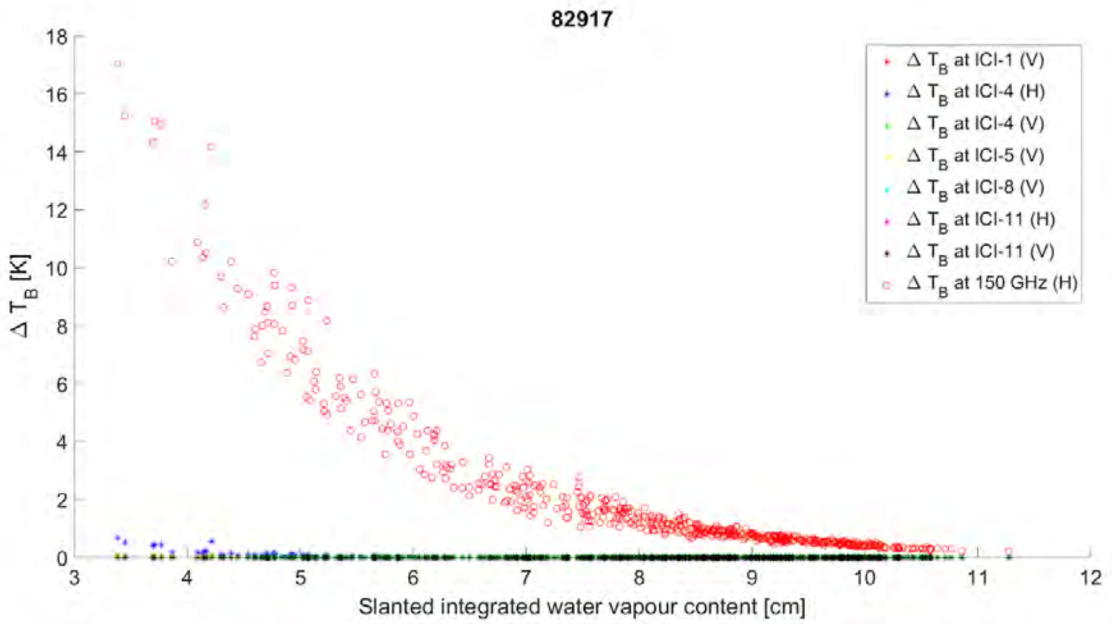



*Figure 5.4.9: Simulated brightness temperature contrast considering landmark target with surface emissivity from TELSEM2 and TESSEM2, considering radiosounding over 82917 station (tropical) on 2016 at several ICI channels and 150 GHz H.*

Fig. 5.4.9 shows BT contrast for a station in a tropical region, where integrated water vapour content is greater than 3 cm and we have very low BT contrast between land and water surfaces. This high value of water vapour content can explain why we have fewer visible days for the Titicaca lake (tropical climate) than for the Qinghai lake (arid-dry climate).
Fig. 5.4.10 shows BT contrast for a RAOB station in arid-dry region, Fig. 5.4.11 shows the BT contrast for a RAOB station in a temperate region and Fig. 5.4.12 shows the BT contrast for a RAOB station in cold-continental region.

Fig. 5.4.13 shows that lower values of integrated water vapour content involve an almost constant trend of $\Delta T_B\,(f)$ for window-frequency BTs (ICI-4 H), highlighted by blue markers and 150 GHz H (indicated by red points). Fig. 5.4.14 confirms this behavior, showing also $\Delta T_B$ at 89 GHz.

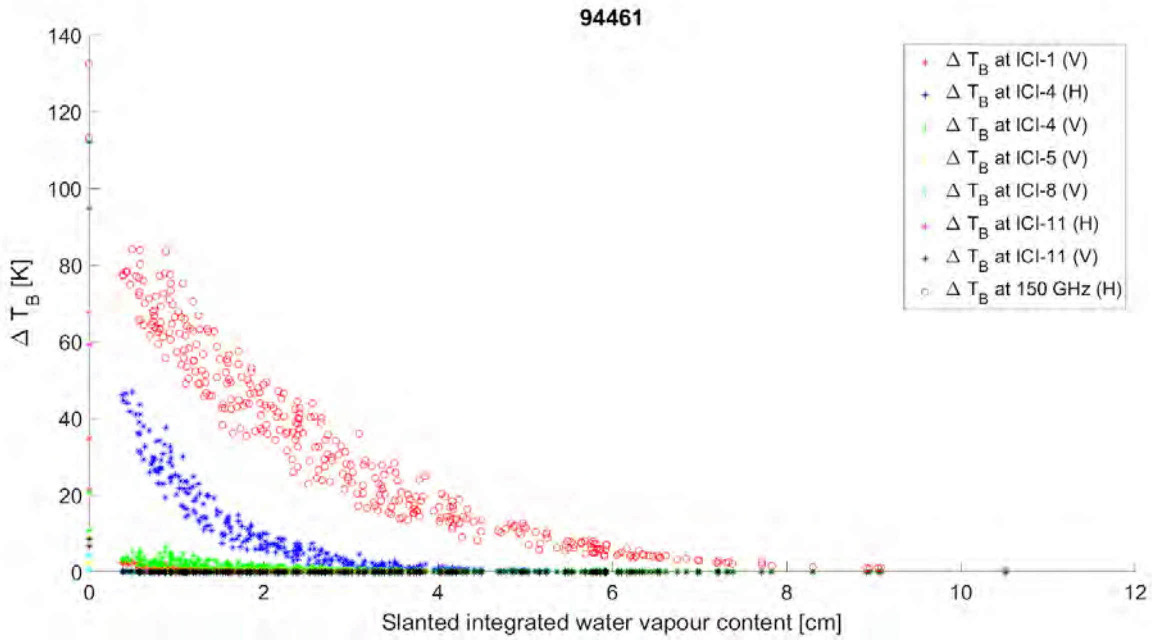



*Figure 5.4.10: Simulated brightness temperature contrast considering landmark target with surface emissivity from TELSEM2 and TESSEM2, considering radiosounding over 94461 station (Arid-dry) on 2016 at several ICI channels and 150 GHz H.*

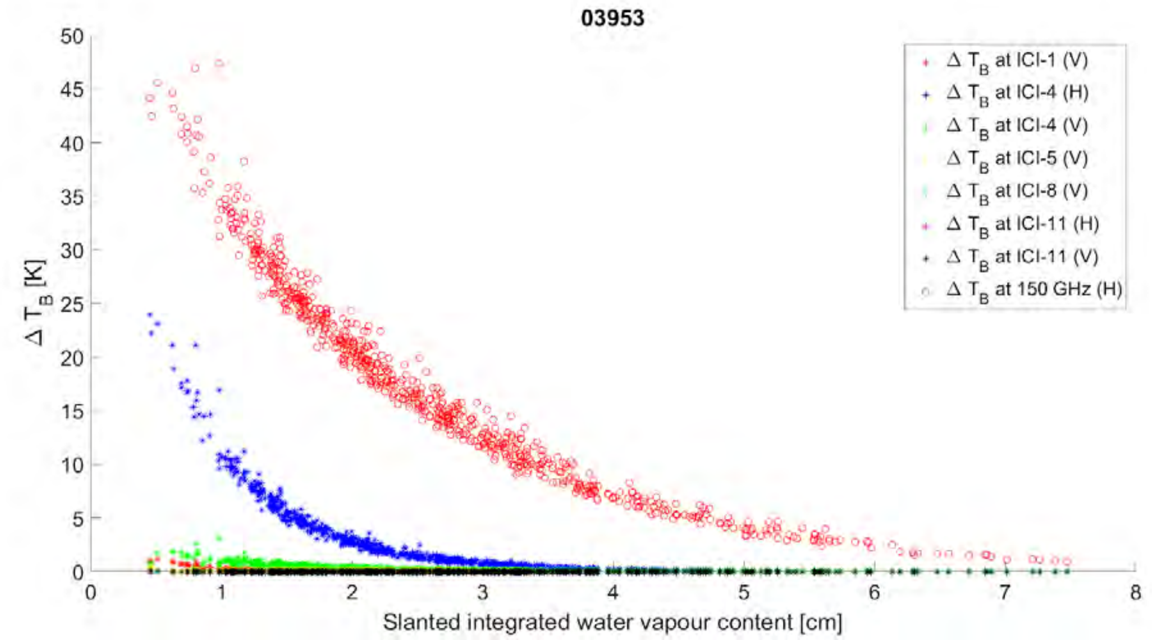



*Figure 5.4.11: Simulated brightness temperature contrast considering landmark target with surface emissivity from TELSEM2 and TESSEM2, considering radiosounding over 03953 station (temperate) on 2016 at several ICI channels and 150 GHz H.*

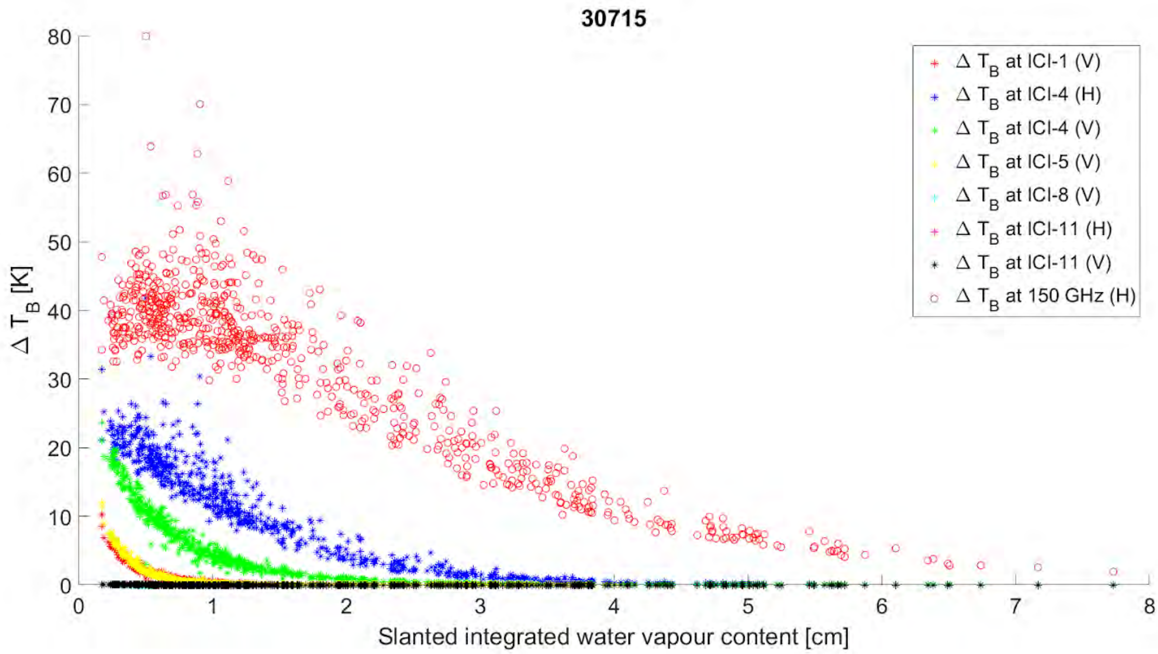



*Figure 5.4.12: Simulated brightness temperature contrast considering landmark target with surface emissivity from TELSEM2 and TESSEM2, considering radiosounding over 30715 station (Cold-continental) on 2016 at several ICI channels and 150 GHz H.*

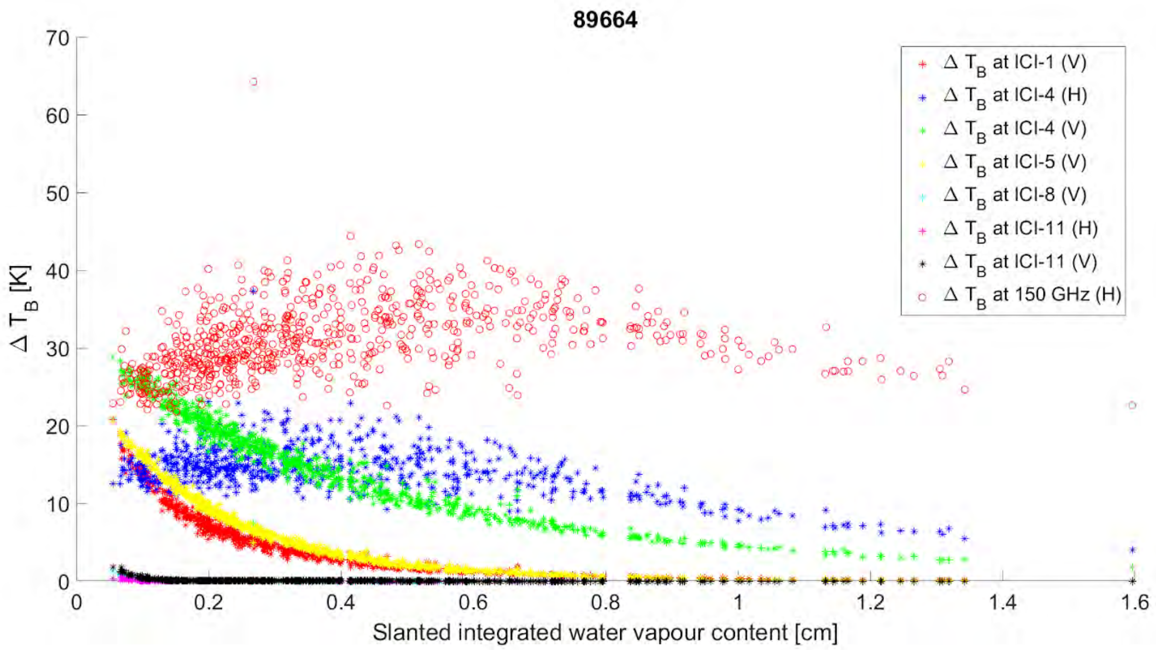



*Figure 5.4.13: Simulated brightness temperature contrast considering landmark target with surface emissivity from TELSEM2 and TESSEM2, considering radiosounding over 89664 station (polar) on 2016 at several ICI channels and 150 GHz H.*

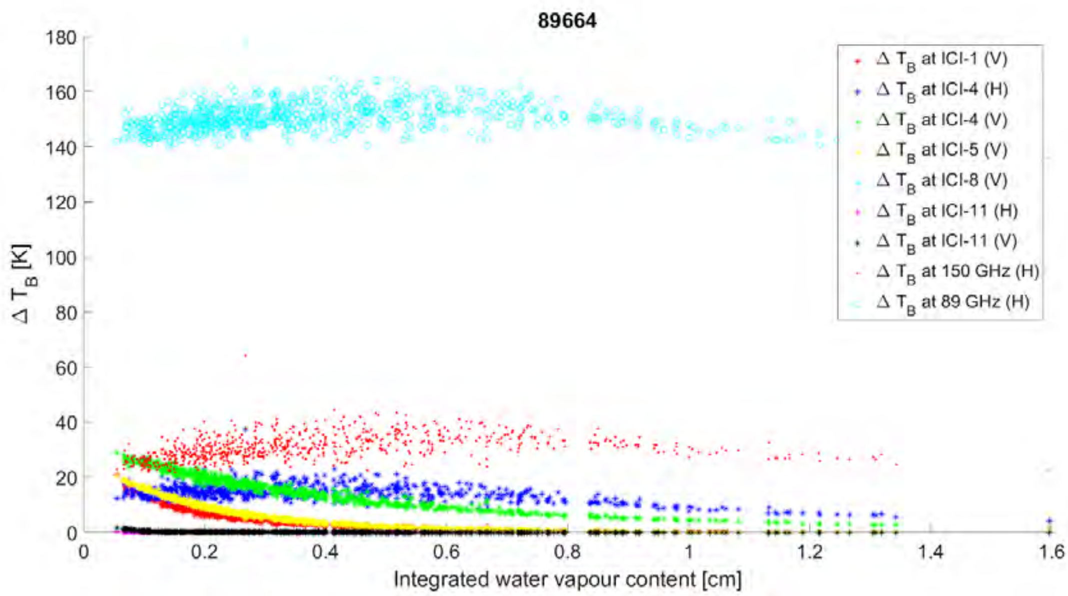



*Figure 5.4.14: Simulated brightness temperature contrast considering landmark target with surface emissivity from TELSEM2 and TESSEM2, considering radiosounding over 89664 station (polar) on 2016 at several ICI channels, 150 GHz H and 89 GHz.*

In all the previous simulations we substantially observe higher values of BT contrast for 150 GHz H and consequently for 243±2.5 GHz BT at horizontal polarization, confirming that we can better detect a surface target using ICI-4 H than 183-GHz channel. Note that, in general, we have obtained a sufficient BT contrast to extract a contour (i.e., more than 10 K) for an integrated water vapour content lower than about 1 cm.

Finally, Fig. 5.4.14 shows the BT simulation over a polar region, using McMurdo RAOB station, in 2016 where the water vapour content is much lower than that of the other considered regions. Looking at the BT contrast for several ICI channels, this figure confirms that in polar regions it is more probable to "see" the surface at millimeter waves and that Antartic ice shelves are very good surface targets.

## 5.5 Geolocation accuracy test using 150 GHz

From subsection 5.4 we have observed that the future ICI-4 (H) BT will probably be more similar to 150 GHz H than to 183 GHz H one. To further investigate this issue, in this section we will describe some tests over the Qinghai lake in 2016.

*Table 5.5.1: Results for geolocation evaluation accuracy using Qinghai lake as target at 150 GHz H and 183 ± 6.6 GHz H considering 84 images on 2016 from SSMIS F17*

| Frequency [GHz] | Geolocation accuracy average [km] | Geolocation accuracy standard deviation [km] |
|---|---|---|
| 150 (H) | 6.25 | 2.39 |
| 183±6.6 (H) | 5.10 | 2.03 |

Considering the same dataset described in subsection 3.2, we have 84 samples for SSMIS F17 150 GHz H. After applying the TCM technique, the geolocation accuracy average is 6.25 km with a standard deviation of 2.39 km. Tab. 5.5.1 summarizes the geolocation accuracy results for 150 GHz and for 183 ± 6.6 GHz.

Tab. 5.5.1 shows very similar results for both frequencies, but the advantage of 150 GHz is to increase the dataset, having more visible days. We have considered all 629 images that contains the lake and, applying the fuzzy-logic approach, we have obtained 265 samples with useful BT contrast around lake coastline to extract a contour., thus showing that 150 GHz has more surface visibility. Considering 265 images over the Qinghai lake, we obtain a geolocation accuracy average of 4.94 km with a standard deviation of 2.16 km. Using 150 GHz H we can get overall results similar to those obtained in Sect. 3.

Fig 5.5.1 shows the geolocation error accuracy and its standard deviation against the number of available SSMIS samples, considering 265 cloud-masked images at 150 GHz using the Qinghai lake target.

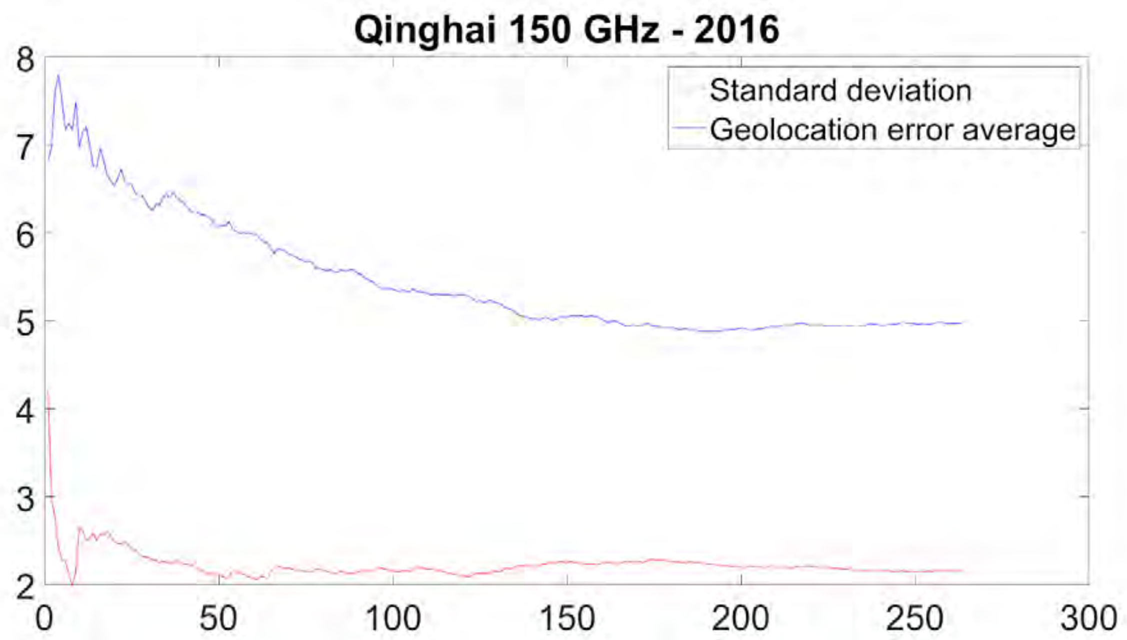



*Figure 5.5.1: Geolocation error average (blue line) and standard deviation (red line) for Qinghai lake using SSMIS F17 at 150 GHz (H).*

Fig. 5.5.1 shows that the geolocation error average reaches 5 km with about 140 samples. In this case, to obtain a stable value of its standard deviation, about 50 samples are sufficient. These numbers are smaller of about 30% than those shown in Fig. 4.5.1 for the 183-GHz channel, mainly due to different frequency vicinity to the absorption peak.

## 6. CONCLUSION

The goal of this Task-1 has been to propose a systematic methodology for geolocation error assessment, including the criteria regarding the search of landmark targets and the cloud-masking defuzzification step to filter the available dataset from cloud coverage contamination. The 6 appendixes discuss the details about the technical analysis of major target classes, such as high-latitude lakes, mountain ranges and ice shelves, as well as some details of the developed algorithms.

*Table 6.1: Summary of geolocation error accuracy results in term of average and standard deviation for all selected targets during 2016 using SSMIS F17 at 183±6.6 GHz at horizontal polarization. Sample yearly number and notes about target features are also reported.*

| **Target** | **Geolocation accuracy mean value [km]** | **Geolocation accuracy standard deviation [km]** | **Cloud-masked yearly sample number (percentage)** | **Notes** |
|---|---|---|---|---|
| **Northern hemisphere (NH)** | | | | |
| Qinghai lake | 5.10 | 2.03 | 129 (20.6%) | All shift directions are sampled due to the close contour. |
| Karakorum mountains | 4.47 | 1.86 | 302 (42.6%) | DEM resolution may impact the results. Useful oblique pattern. |
| Hudson Bay | 5.28 | 2.56 | 135 (49.0%) | All shift directions are sampled due to the U contour. |
| Nares strait | 4.55 | 1.65 | 587 (27.5%) | Slightly scattered contour with oblique pattern. |
| **NH average value** | *4.9 km* | *2.0 km* | | |
| **Southern hemisphere (SH)** | | | | |
| Ross ice shelf | 5.30 | 2.18 | 725 (31.1%) | Sharp high-resolution contour, but mainly horizontal pattern. |
| Filchner-Ronne ice shelf | 4.31 | 1.89 | 541 (22.9%) | Sharp high-resolution contour with a V contour |
| Amery ice shelf | 5.32 | 2.27 | 242 (19.5 %) | Sharp high-resolution contour with a nearly-vertical contour |
| Titicaca lake | 4.80 | 2.50 | 52 (9.8%) | All shift directions are sampled due to the close contour. |
| Andean mountains | 3.70 | 1.95 | 177 (19.3%) | DEM resolution may impact the results. Useful oblique pattern. |
| **SH average value** | *4.7 km* | *2.2 km* | | |

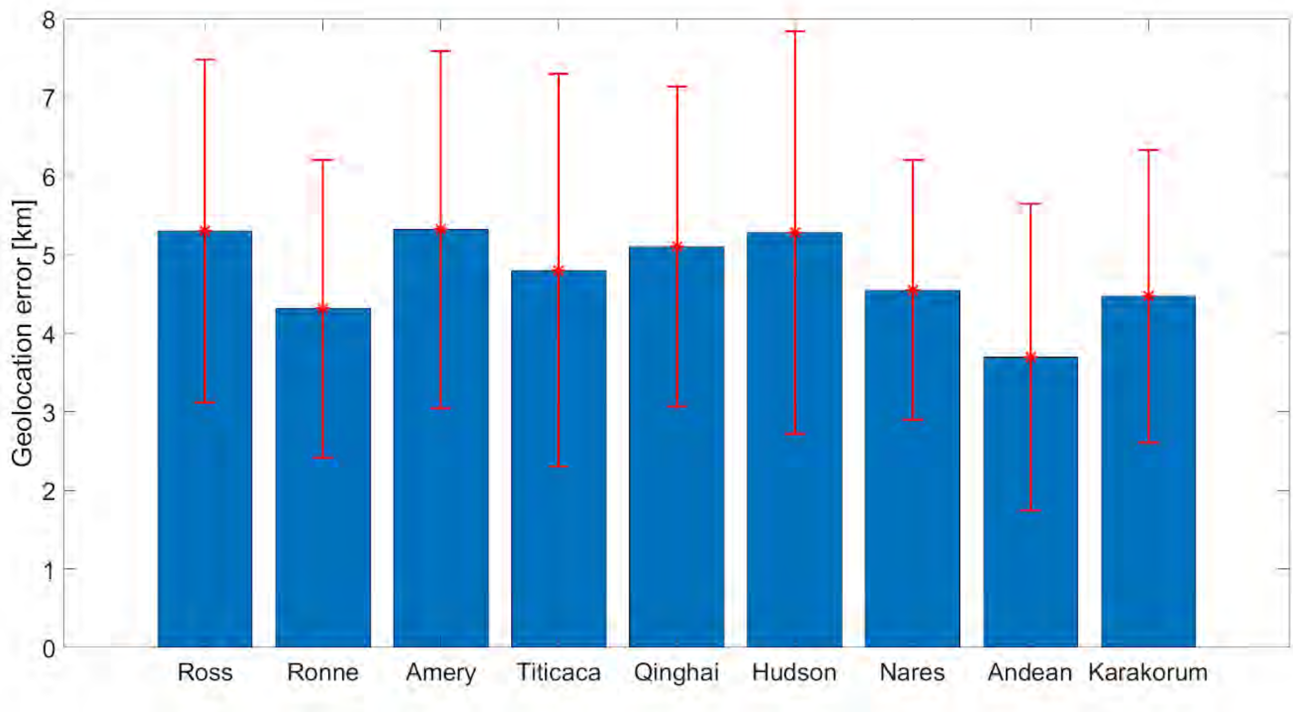



*Figure 6.1: Summary of geolocation error accuracy results in term of average and standard deviation for all selected targets during 2016 using SSMIS F17 at 183±6.6 GHz at horizontal polarization.*

A total number of **9 landmark targets** has been selected covering both the northern and southern hemisphere in order to guarantee good temporal coverage during the driest seasons. For each target results have been provided in terms of mean value and standard deviation of the geolocation error both in the northern hemisphere and southern hemisphere.

The problem of the **sensitivity analysis of the TCM geolocation error assessment methodology** to the most critical free parameters has been discussed as a proxy to the error budget estimate. The latter, as a matter of fact, is not easily defined for the lack an absolute reference (we are here estimating not the geolocation error, but its accuracy or the uncertainty of the geolocation error correction procedures). The conclusion is that the interpolation-grid spatial resolution provides a parametric variability of about 0.6 km. Moreover, from the sensitivity analysis to the cloud-masked sample size, we can conclude that about 50-75 images are sufficient to assess the geolocation error statistics for all landmark targets.

The Table 6.1 and Fig. 6.1 report a **summary of the geolocation accuracy validation** for all targets using SSMIS F17 at 183±6.6 GHz at horizontal polarization. The average value of about 4.8 km with a standard deviation of about 2.1 km can be interpreted as the mean geolocation error of SSMIS selected imagery. These numbers are comparable with the values given in Poe et al. [18] and Kunkee et al. [19] for SSMIS F16 (different from F17 we have used in this work) even though they mention the estimate of 4-5 km [13] and less than 6 km [19]. Note that Poe et al. [18] refer to their estimate as 1-sigma error value, whereas Kunkee et al. [19] stress the fairly good stability of their retrieved error.

The analysis, carried out using SSMIS channel at 183±7 GHz, has been extended to ICI channels at 183 GHz and 243 GHz through a **simulation-based frequency scaling**. The approach has involved the analysis of SSMIS imagery at 150 GHz and 183±7 GHz as well as the radiative transfer simulation of satellite brightness temperatures and slant-path attenuation from both available radiosounding profiles and

ERA5 reanalysis atmospheric profiles near the selected targets. Note that SSMIS 183±7 GHz channel is at H polarization, whereas the foreseen ICI 183±6.6 GHz one is at V polarization meaning that we expect a slightly reduced BT contrast for ICI with respect to SSMIS one (due to the larger V-polarized surface emissivity), as described in Section 5. Using the 243-GHz ICI-like channel we have shown that its BT is expected to be higher than SSMIS-like 150-GHz using both RAOB and ERA5 profiles.

It is finally worth mentioning that the **atmospheric transmittance at 243 GHz** is fairly comparable to the 150-GHz one for the selected targets. To some extent, we can presume that these results for the landmark target approach using SSMIS H-polarization 183±6.6 GHz, would be similar to the ones obtainable for ICI V-polarization 183±7 GHz and at least comparable or worse than those derivable from ICI H-polarization 243-GHz channel.

# APPENDIX A. TCM approach for high-altitude lake targets

The inputs to validate the geolocation using a lake are the images of SSMIS F17 at 183 ± 6.6 GHz (horizontal polarization) along the whole 2016 year. Initially it is necessary to extract only the spaceborne radiometric images containing the high-altitude lake.

Unfortunately, not all images can be useful for our purpose, as explained in the Sec. 4, due to possible atmospheric opacity in presence of clouds and precipitation. In order to apply the fuzzy-logic cloud masking to high-altitude lakes, referring to subsection 2.3, for $M_2(\Delta T_{Bm})$ we use the following equation:

$$M_2(\Delta T_{Bm}) = 1 \qquad if\ \Delta T_{Bm} \geq 8K \tag{A.1}$$
$$M_2(\Delta T_{Bm}) = \Delta T_{Bm}/8 \qquad if\ \Delta T_{Bm} < 8K$$

where $\Delta T_{Bm}$ is the mean BT contract around the target.

To evaluate the mean contrast around the lake, we can compute the BT difference along vertical and horizontal directions. For example, Fig. A.1 shows the pixels A, B, C, D and E, selected to compute the BT contrast $\Delta T_{Bm}$ for Qinghai lake, using the following equation:

$$\Delta T_{Bm} = \frac{(T_B - T_A) + (T_C - T_A) + (T_D - T_A) + (T_E - T_A)}{4} \tag{A.2}$$

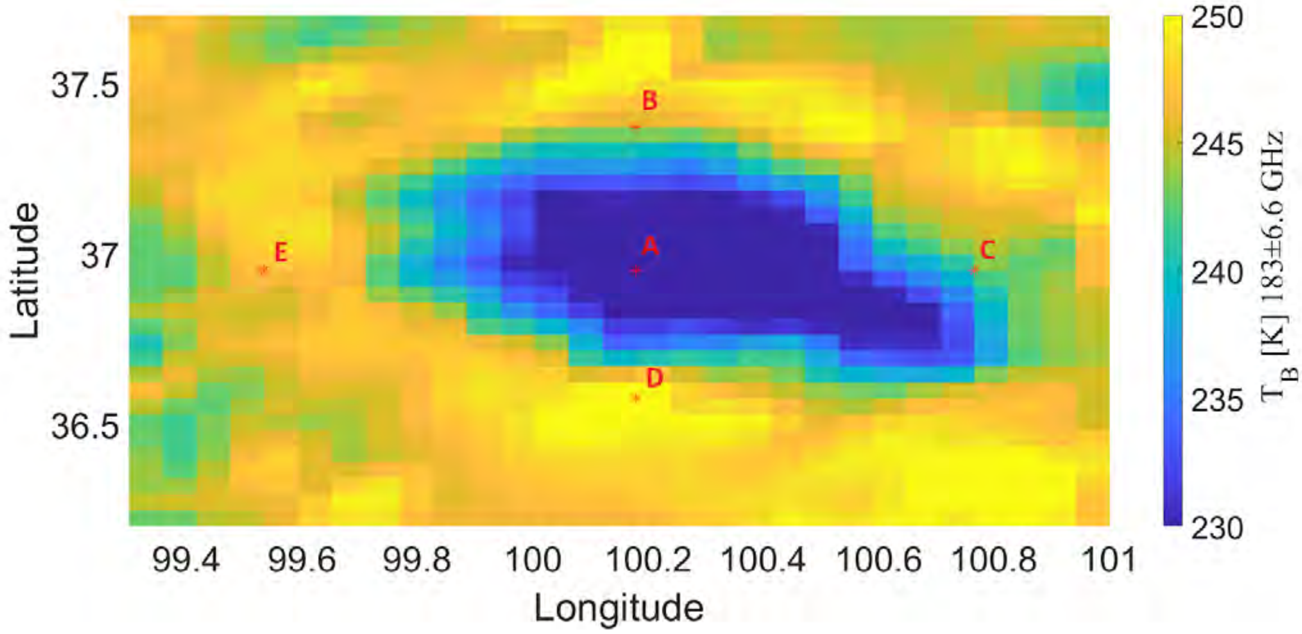



*Figure A.1: Brightness temperature (BT) image at 183±6.6 GHz H over Qinghai lake from SSMIS F17 on 2016/12/01. Five points are those used to calculate the BT contrast along vertical and horizontal directions*

The BT contrast is evaluated in the interpolated grid at 5 km of spatial resolution, in order to obtain the BT contrast between the same points for all images. The function $M_1(\varepsilon)$ can be calculated as:

$$M_1(\varepsilon) = 0 \qquad if\ \varepsilon \geq 15\ km$$
$$M_1(\varepsilon) = -\frac{\varepsilon}{15} + 1 \qquad if\ \varepsilon < 15\ km \tag{A.3}$$

Computing the inference function from (A.1), the considered image is selected only if $I\left(\Delta T_{Bm}, \varepsilon\right) > 0.3$. The remaining images can be used to validate the geolocation error using the lake as target.

As mentioned in Section 2, before the defuzzification step, it is necessary to correct the parallax error because the lake has a high altitude. To correct this error, we must find the intersection between the line of sight of the satellite and the orography, provided by GTOPO30, that is a digital elevation model (DEM) with a resolution of 30 arcsec (approximately 1 km) To better represent the footprint, we have calculated the average of the DEM with a spatial resolution of about 13 km, as pixels dimension, and the we found the intersection between the line of sight and this average DEM , as shown in Fig. A.2. After this correction the image is shifted, depending on ascending or descending orbit and the position of the target in the satellite swath.

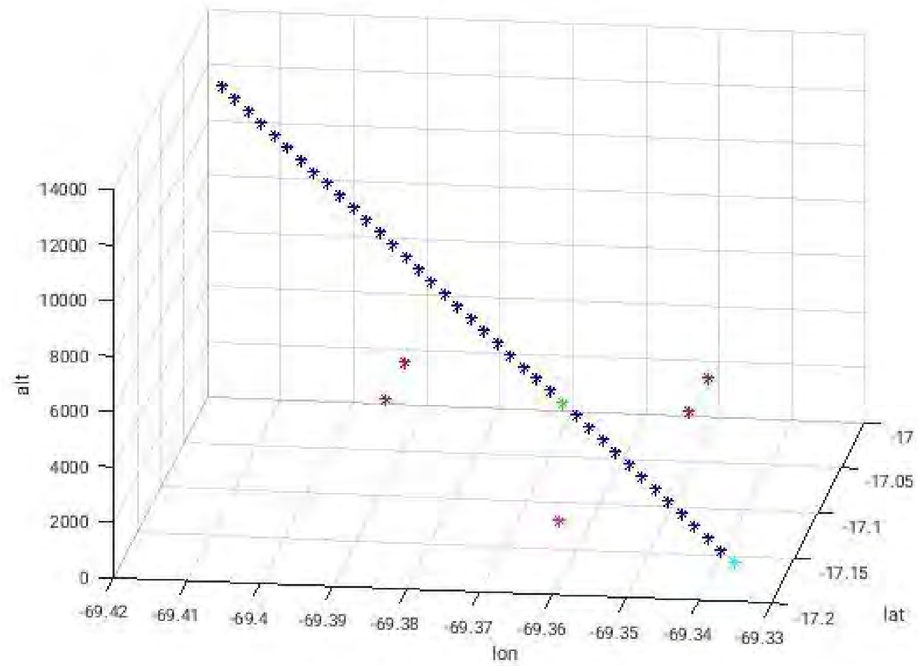



*Figure A.2: Example of parallax error correction: blue markers represent the discretized satellite line of sight. The four red points are the nearest points of DEM around the intersection between DEM and line of sight. The green marker is the first point of the line of sight that has an altitude lower than DEM. The cyan point is the intersection between line of sight and earth ellipsoid (WGS84). Finally, magenta marker represents the corrected coordinates on surface.*

To increase the BT image spatial resolution, the different samples are interpolated using cubic interpolation method in the same evenly spaced grid with 5km of spatial resolution, shown in Fig. A.3.

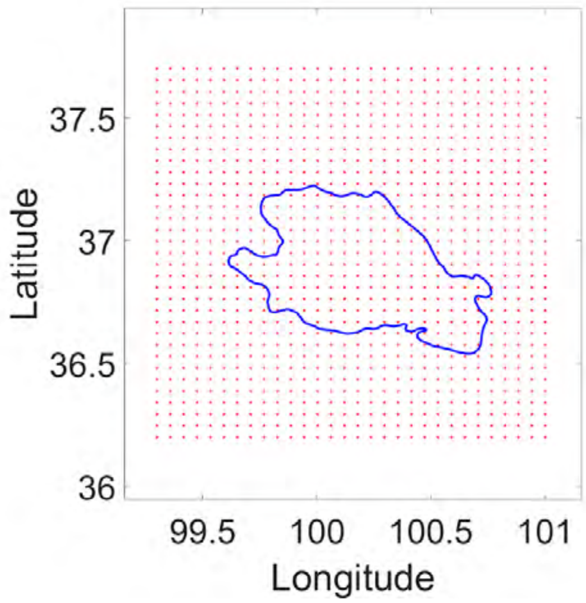



*Figure A.3: Grid used to interpolate the BT data for Qinghai lake. The blue line represents the contour of Qinghai lake provides by GSHHG database.*

After the interpolation step, it is possible to extract the contour and for Qinghai lake we adopt a Canny algorithm to obtain the contour line, as shown in Fig. A.4

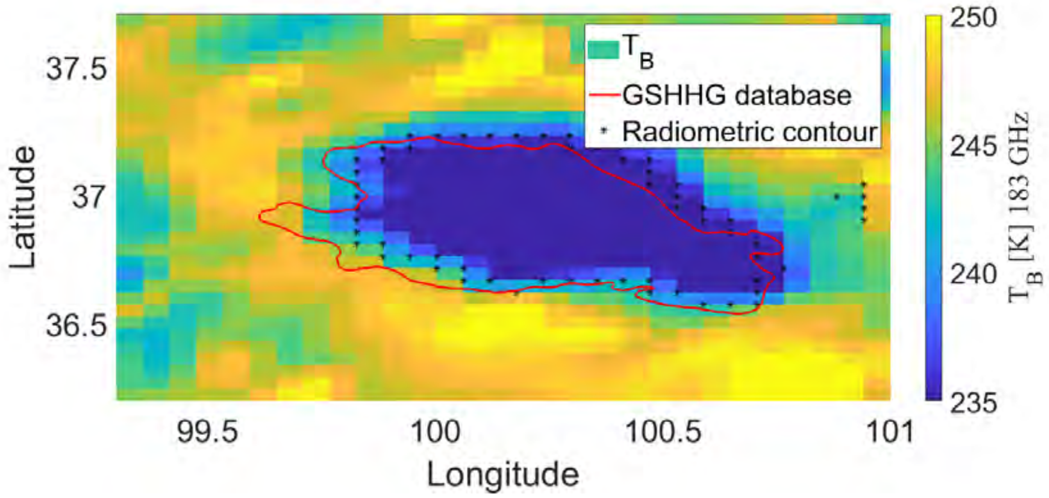



*Figure A.4: Brightness temperature (BT) image at 183±7 GHz H over Qinghai lake from SSMIS F17 on 2016/12/01. The red line represents the lake coastlines from GSHHG database, described in Wessel and Smith (1996). Black markers indicate the extracted contour by Canny method.*

To correlate the reference line with the satellite radiometric contour, we can project the GSSHG line on the same radiometric grid, using the nearest-neighbor technique, as shown in Fig. A.5:

The code is developed in Matlab environment and, summarizing all steps for this kind of target, we list:

1) Extract the box that contains target
2) Parallax error correction, as shown in Appendix E.
3) Interpolate data to fictitiously increase the spatial resolution, using *'griddata'* Matlab function
4) Apply Canny algorithm to extract radiometric contour, using '*edge*' Matlab function

5) Project GSHHG shoreline database on the same grid obtained at step 2, using nearest neighbour approach.
6) Calculate the normalized cross-correlation, using *'normxcorr2'* Matlab function
7) To reach sub-pixel accuracy, the maximum of the normalized cross-correlation is fitted by a 4th-order polynomial.
8) Take the coordinates of the maximum of fitted normalized cross-correlation.
9) Calculate the shift in pixels
10) Calculate the corresponding shift along latitude and longitude
11) Evaluate the displacement (in km) of the shift found in step 9.

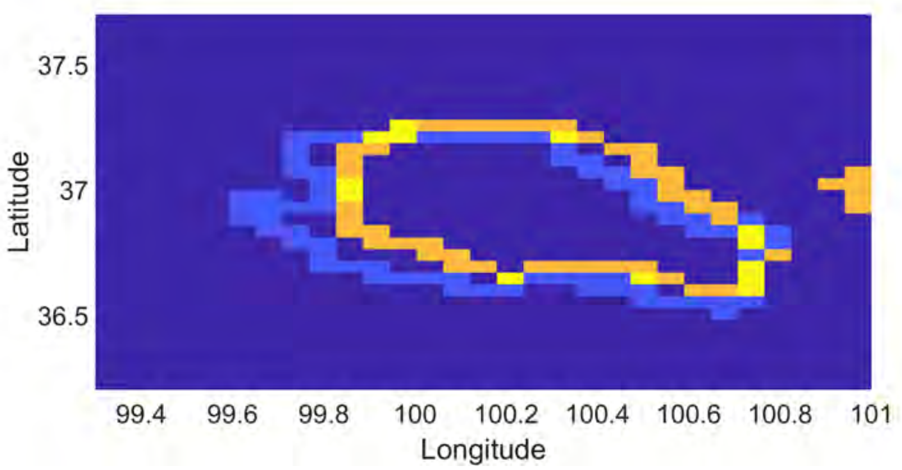



*Figure A.5: The orange line is the radiometric contour; the blue line is the reference line and the yellow is the overlap of the two lines.*

## APPENDIX B. TCM approach for mountain-chain targets

The inputs to validate the geolocation using mountain chains are the images of SSMIS F17 at 183 ± 6.6 GHz (horizontal polarization) along the entire 2016. Initially it is necessary to extract only the images that contain the target and then it is necessary the parallax error using DEM correction. The used inference function is:

$$I\left(\Delta T_{Bm}, \varepsilon\right) = M_1(\varepsilon)\, M_2(\Delta T_{Bm}) \tag{B.1}$$

where:

- $I(x)$ = inference function
- $\Delta T_{Bm}$ = mean BT contrast around lake
- $\varepsilon$ = geolocation error
- $M_1(\varepsilon)$ = membership function depending on the geolocation error
- $M_2\left(\Delta T_{Bm}\right)$ = membership function depending on the BT contrast

To evaluate the mean BT contrast along mountain, we can compute the BT difference along the horizontal and vertical axis. For example, Fig. B.1 shows the selected pixels to compute the BT contrast $\Delta T_{Bm}$ for Karakorum mountains, obtained by the following equation:

$$\Delta T_{Bm} = \frac{(T_B - T_A) + (T_D - T_C) + (T_F - T_E) + (T_H - T_G)}{4} \tag{B.2}$$

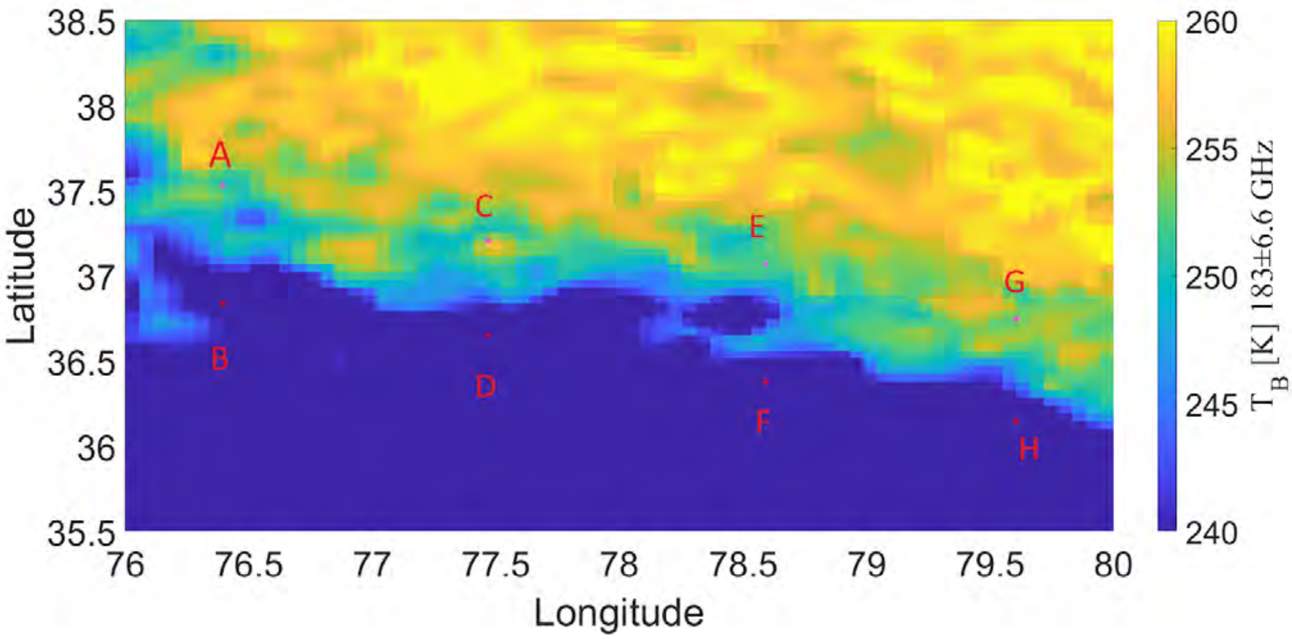



*Figure B.1: Brightness temperature (BT) image at 183±6.6 GHz H over Karakorum mountains from SSMIS F17 on 2016/01/02. Eight points are those used to calculate the BT contrast along mountain chain*

The BT contrast is evaluated in the interpolated grid at 5 km of spatial resolution, in order to obtain the BT contrast between the same points for all images. It is then possible to obtain the inference function and the single image is used only in case with $I(\Delta T_{Bm}, \varepsilon) > 0.3$ .

After the defuzzification step, we have the complete dataset to validate the geolocation accuracy. To increase the images spatial resolution, the different samples are interpolated using cubic interpolation in the same evenly spaced grid. After the interpolation step, we have calculated the gradient of the image using Sobel filter and use a DEM as a reference.

To correlate the gradient of BT temperature with the reference, we have reprojected DEM in the same grid, applying the Sobel filter to obtain the reference gradient. Finally, it is possible to correlate the two images, obtaining the relative displacement.

The code is developed in Matlab environment and, summarizing all steps for this kind of target, we list:

1) Extract the box that contains target
2) Parallax error correction, as shown in Appendix E.
3) Interpolate data to fictitiously increase the spatial resolution, using *'griddata'* Matlab function
4) Apply Sobel filter to DEM to calculate its gradient
5) Apply Sobel filter to radiometric image to calculate its gradient
6) Calculate the fast fourier transform to both gradients, using '*fft2'* Matlab function
7) Calculate the shift in pixels between to images, using *'dftregistration'* Matlab function
8) Calculate the corresponding shift along latitude and longitude
9) Evaluate the displacement (in km) of the shift found in step 8

## APPENDIX C. TCM approach for ice-shelf targets

The inputs to validate the geolocation using an ice shelf are the images of SSMIS F17 at 183 ± 6.6 GHz (horizontal polarization, H) along the entire 2016. Initially it is necessary to extract only images containing different shelves. Unfortunately, not all images can be useful for our purpose, as explained in Sec. 2.3. The proposed inference function is:

$$I\left(\Delta T_{Bm}, \varepsilon\right) = M_1(\varepsilon)\, M_2(\Delta T_{Bm}) \qquad \text{(C.1)}$$

where:

- $I(x)$ = inference function
- $\Delta T_{Bm}$ = mean BT contrast around lake
- $\varepsilon$ = geolocation error
- $M_1(\varepsilon)$ = membership function depending on the geolocation error
- $M_2\left(\Delta T_{Bm}\right)$ = membership function depending on the BT contrast

Considering that for ice shelves the BT contrast is higher with respect to other targets, like the Qinghai lake, we have increased this threshold. Therefore, for $M_1(c)$ around the lake we use the following equations:

$$\begin{aligned} M_2\left(\Delta T_{Bm}\right) &= 1 && \text{if } \Delta T_{Bm} \geq 15K \\ M_2\left(\Delta T_{Bm}\right) &= \Delta T_{Bm}/15 && \text{if } \Delta T_{Bm} < 15K \end{aligned} \qquad \text{(C.2)}$$

To evaluate the contrast $\Delta T_{Bm}$ around the lake, we propose to calculate the BT contrast differently along vertical and horizontal directions. Fig. C.1 shows the point selected for calculate the BT contrast, obtained by the following equation:

$$\Delta T_{Bm} = \frac{(T_B - T_A) + (T_D - T_C) + (T_F - T_E) + (T_H - T_G)}{4} \qquad \text{(C.3)}$$

$M_1(\varepsilon)$ can be calculated as:

$$\begin{aligned} M_1(\varepsilon) &= 0 && \text{if } \varepsilon \geq 15\ km \\ M_1(\varepsilon) &= -\tfrac{\varepsilon}{15} + 1 && \text{if } \varepsilon < 15\ km \end{aligned} \qquad \text{(C.4)}$$

Finally, it is possible to obtain the inference function and the single image is used only in case with $I\left(\Delta T_{Bm}, \varepsilon\right) > 0.3$.

Ice shelves are at sea level so that it is not necessary to perform a parallax error correction using DEM. Secondly, to fictitiously increase the spatial resolution of BT images for intercomparison purposes, data are upsampled on a regular grid through a triangulation method using a cubic interpolation. A polar stereographic map projection is used in this work. The new grid is regularly evenly spaced (about 5 km) in X-Y domain and the resulting BT, for Ross ice shelf, is shown in Figure D.2.

The limits of the box for Ross ice shelf are the following:

- Latitude = [-78.5 -76.5];

- Longitude = [170.6 178.5].

The BT contrast (Eq. C.3) is evaluated in the interpolated grid at 5 km of spatial resolution on a polar stereographic map, in order to obtain the BT contrast between the same points for all images.

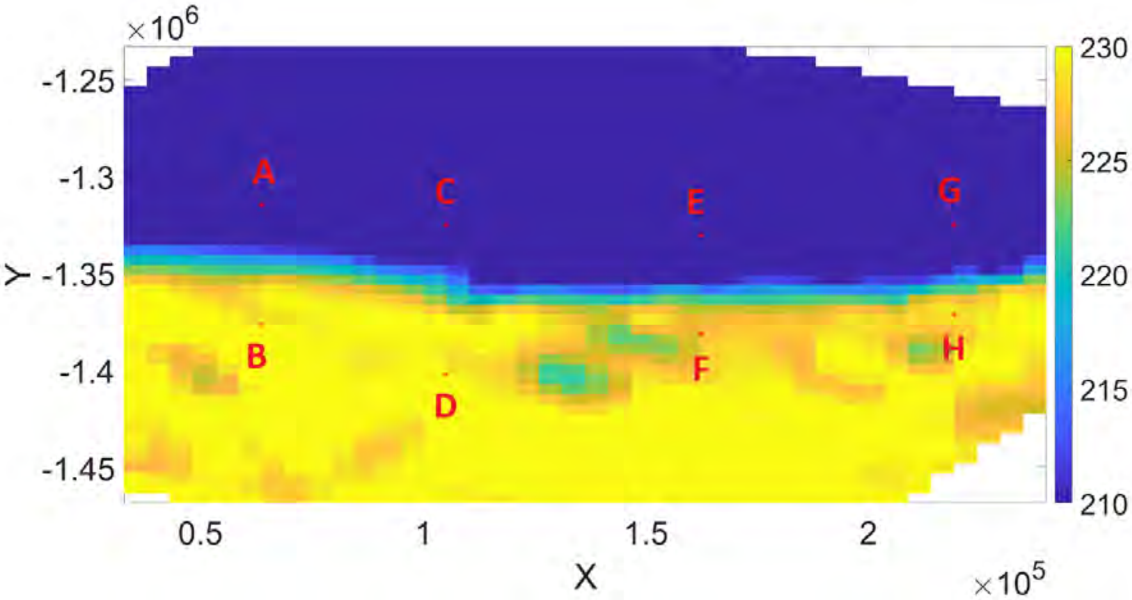



*Figure C.1: Brightness temperature (BT) image at 183±6.6 GHz H over Ross ice shelf from SSMIS F17. Eight five points are those used to calculate the BT contrast along coastline*

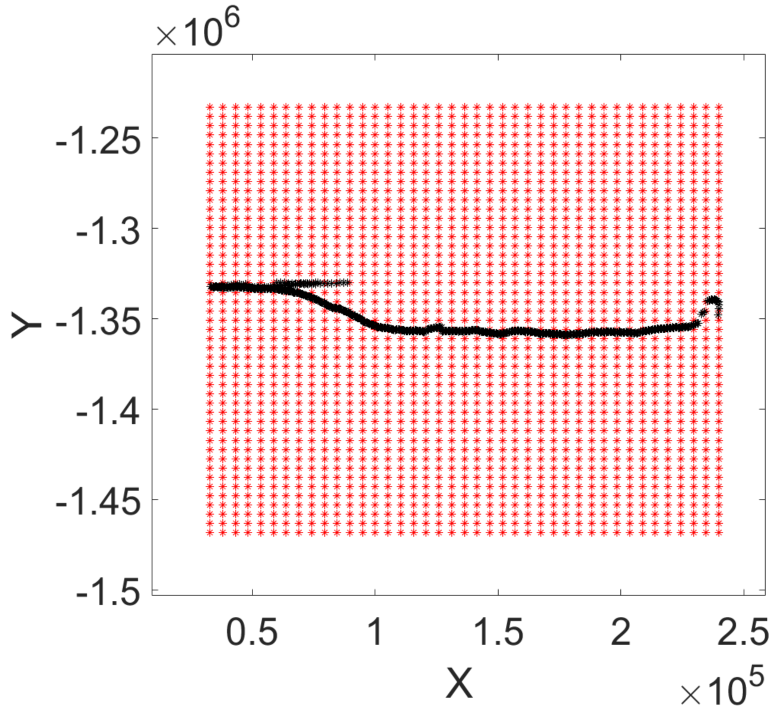



*Figure C.2: Grid used to interpolate the BT data for Ross ice shelf. Black markers represent the contour of Ross extracted from SAR image.*

In the Antarctic region SAR data are available in Extra-Wide Swath Mode with a 400-km swath at 20x40 $m^2$ spatial resolution and it is possible to download them from the following web site: https://scihub.copernicus.eu/dhus/#/home

For example, the image in Fig. 3.6.1.3 is obtained from the data: S1A_EW_GRDM_1SSH_20160724T111725_20160724T111829_012289_0131CA_1241

This dataset contains the following fields:

- amplitude_HH
- intensity_HH
- amplitude_HV
- intensity_HV

SAR data are provided as metadata, containing also the orbit state vectors, but it is generally not accurate and can be refined with the precise orbit files which are available days-to-weeks after the generation of the product. It is necessary to apply orbit file operator to obtain accurate satellite position and velocity information. To do this correction and for the other necessary steps, we have adopted the SNAP toolbox, downloadable from the web site http://step.esa.int/main/download/snap-download/. For Level-1 GRD it is also necessary a thermal noise removal.

To obtain imagery in which the pixel values can be directly related to the scene radar backscatter, the calibration step has to be carried out to have sigma-nought images. At this step the resolution is very high for our purpose and it is still present a speckle noise. To reduce it, it is possible to apply a multilook operator. After these steps, it is possible to extract a contour, as shown in Fig. A.3, but to use this as reference in the validation of geolocation accuracy it is necessary a further step. This reference line must be projected in the same regular evenly spaced (about 5 km) in X-Y domain, adopting the nearest neighbour approach, as shown in the following Fig. D.3.

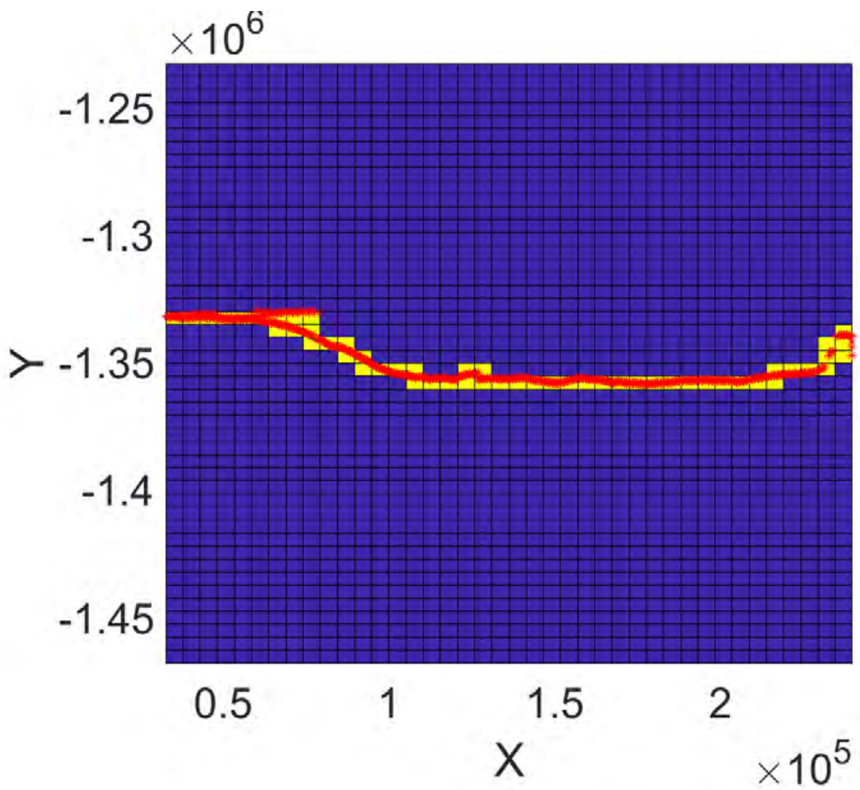



*Figure D.3: The red markers indicate the contour extracted from SAR data and yellow pixels represent the reference contour in the same radiometric grid with a spatial resolution of about 5 km.*

Correlating the obtained reference line with the extracted radiometric contour, it is then possible to validate the geolocation accuracy (in km).

The code is developed in Matlab environment and the steps for these targets are summarized as follows:

1) Extract the box that contains target
2) Project radiometric data on polar stereographic map, using '*polarstereo_fwd*' Matlab function
3) Interpolate data to fictitiously increase the spatial resolution, using *'griddata'* Matlab function
4) Apply Canny algorithm to extract radiometric contour, using '*edge'* Matlab function
5) Project SAR data on polar stereographic map, using '*polarstereo_fwd*' Matlab function
6) Apply Canny algorithm to extract SARcontour, using '*edge'* Matlab function
7) Project SAR contour in the same grid obtained at step 3, using nearest neighbour approach.
8) Calculate the normalized cross-correlation, using *'normxcorr2'* Matlab function
9) To reach sub-pixel accuracy, the maximum of the normalized cross-correlation is fitted by a 4th-order polynomial.
10) Take the coordinates of the maximum of normalized cross-correlation.
11) Calculate the shift in pixels
12) Calculate the corresponding shift along latitude and longitude. To reproject data on geographical coordinates (latitude-longitude) we have used the *'polarstereo_inv'* Matlab function
13) Evaluate the displacement (in km) of the shift found in step 11.

## APPENDIX D. Contour extraction and cross-correlation techniques for TCM

Using the target-contour matching algorithm, the extraction of a contour that can be carried out by applying two main methods:

- the Canny approach [9] to extract a line. This method consists of the following main steps:
  1. Convolution with Gaussian filter coefficient
  2. Convolution with Canny filter for horizontal and vertical orientation
  3. Calculating directions using atan2
  4. Thresholding

- the Sobel filter [13] to obtain a gradient map. This method consists of the following main steps:
  1. Convolution with two matrices to compute the derivative along x and y
  2. Computing the gradient magnitude

The extracted contour can be then cross-correlated with a reference to validate the geolocation error using the fast normalized cross-correlation (FNC) function $\gamma(u,v)$:

$$\gamma(u,v) = \frac{\sum_{x,y}\left[f(x,y)-\bar{f}_{u,v}\right]\left[t(x-u,y-v)-\bar{t}\right]}{\left\{\sum_{x,y}\left[f(x,y)-\bar{f}_{u,v}\right]^2 \sum_{x,y}\left[t(x-u,y-v)-\bar{t}\right]^2\right\}^{0.5}} \tag{D.1}$$

where $f$ is the BT image under consideration and the sum is over all pixels $(x,y)$ under the window containing the BT template $t$ positioned at $(u,v)$ displacements, $\bar{t}$ and $\bar{f}$ are the the means of the template and function, respectively, in the region under the template.

Picking the maximum of $(u,v)$ it is possible to obtain the lat-lon pixel displacements then converted into shifts along x and y direction. In order to have an accuracy of about 0.1 pixel, the maximum is fitted with a polynomial of 4th order. From these pixel displacements it is possible to obtain the related latitude and longitude error and the corresponding distance error in km.

An alternative way to obtain directly a displacement with sub-pixel accuracy is to use the registration in frequency domain (RFD) technique [17]. Between FNC and RFD we expect the same results, because the only difference is that FNC is computed in the spatial domain whereas the other one RFD is computed in the frequency domain, so that differences should only be numerical.

## APPENDIX E. Parallax error correction

In case of high-altitude targets, we have parallax error as shown in Fig. 2.1.2. In these cases the coordinates, provided by SSMIS F17 on WGS84 ellipsoid, must be corrected by finding the intersection between the line-of-sight of the satellite and the orography described by the DEM. The line of site is the black line in Fig. E.1, joining the satellite position (red marker in Fig. E.1) and the footprint on WGS84 (magenta marker in Fig. E.1).

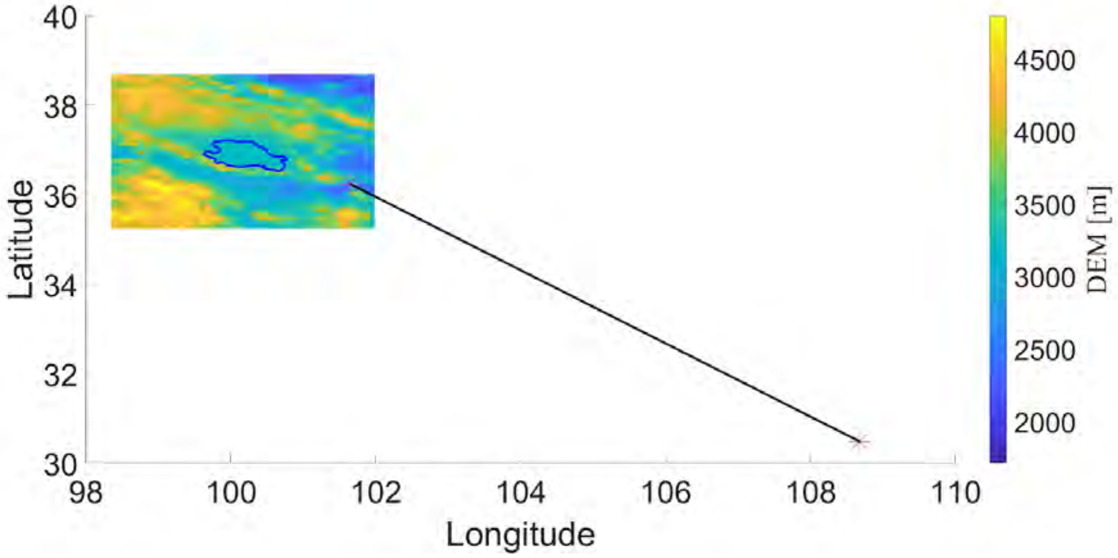



*Figure E.1: Digital elevation model with spatial resolution of about 10 km over Qinghai lake. Red marker represents the satellite position and magenta marker indicates the footprint. Blue line is Qinghai lake obtained from GSHHG shoreline database.*

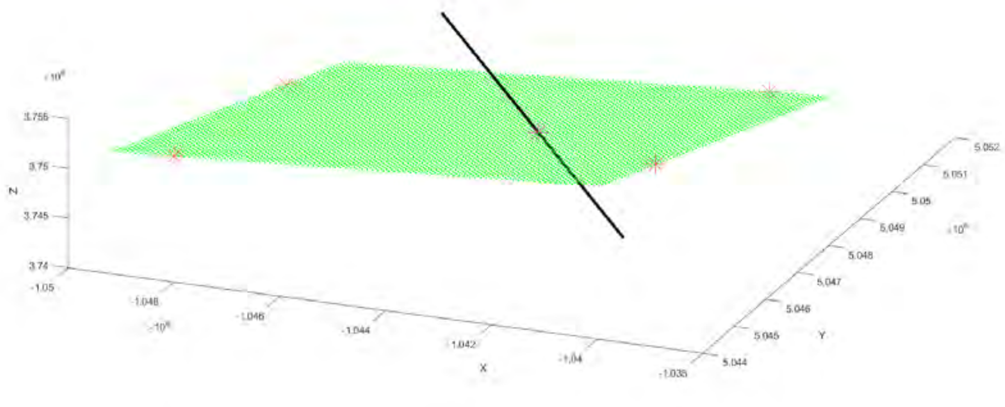


*Figure E.2: The four red markers are DEM points with resolution of about 10 km. Green markers are the surface generated from DEM points. Black line is the line of sight and magenta point is the intersection, that is the corrected coordinates.*

True coordinates is provided by its intersection with the surface generated by the nearest four points of DEM, as shown in Fig. E.2. To better represent the size of footprint, we had used a DEM averaged at about 10 km.

This correction is is made for each pixel of radiometric image, as shown in Fig. E.3.

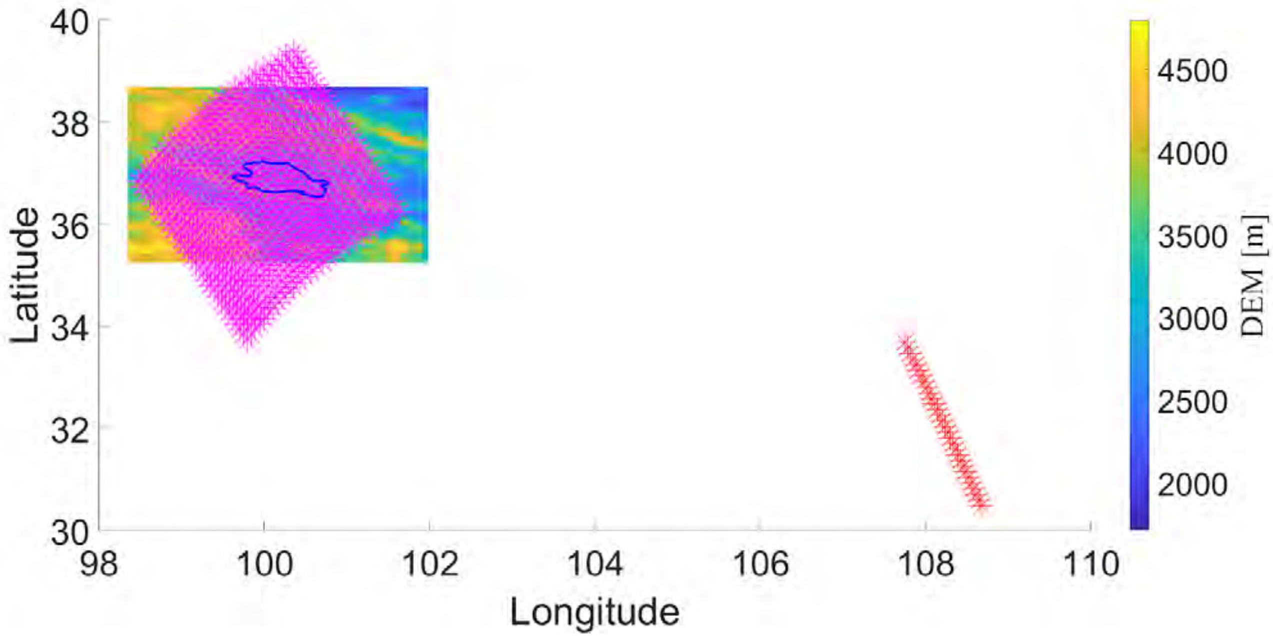



*Figure E.3: The red markers indicate the satellite positions. Magenta markers represent the radiometric footprint.*

After this correction the image is shifted, depending on ascending or descending orbit and the position of the target in the satellite swath.

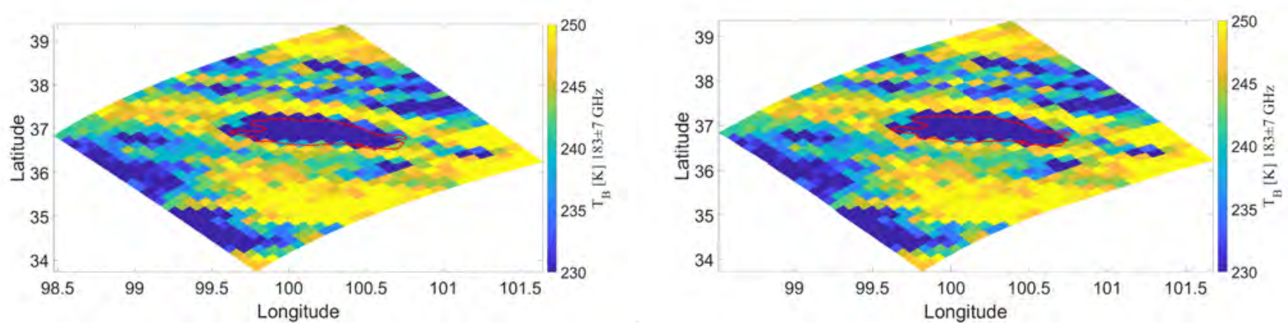



*Figure E.4: In the left there is the original image (it contains parallax error) and in the right there is the corrected image.*

Fig. E.4 contains an example of parallax error correction. As shown in Fig. E.3, in this example we have an ascending orbit and the target is in the left part of the swath. In this case, after the parallax error correction, the image is shifted in south-east direction (according to line of sight direction).

## APPENDIX F. Threshold selection for cloud-masking fuzzy-logic algorithm

To establish the threshold for $M_2(\varepsilon)$, we have considered the pixel dimension that is about 13 x 16 km$^2$ in case of SSMIS F17 1876.6 GHz. We can then expect a geolocation accuracy lower than this value and we had decided to put 15 km as a threshold for $M_2(\varepsilon)$ for all targets. To establish the threshold for $M_1(\Delta T_{Bm})$, we have observed the BT contrast for several targets. For example, considering the Ross ice shelf, that has 2324 samples along 2016, with BT contrast shown in Fig. F.1.

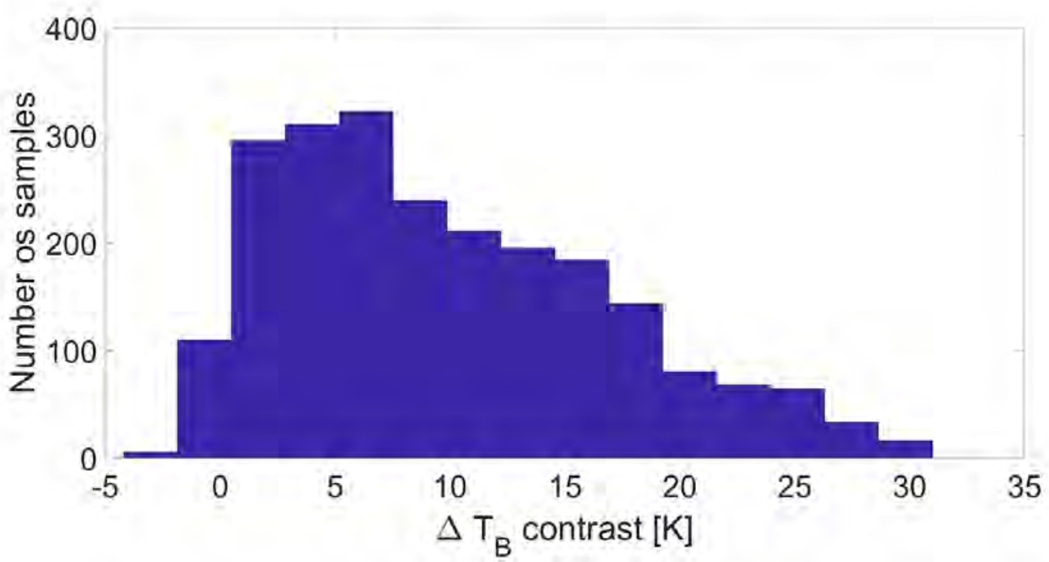



*Figure F.1: $\Delta T_{Bm}$ for Ross ice shelf considering all 2324 samples*

Then we have removed all images with negative value of $\Delta T_{Bm}$, reducing the dataset to 2206 samples, as shown in Fig. F.2.

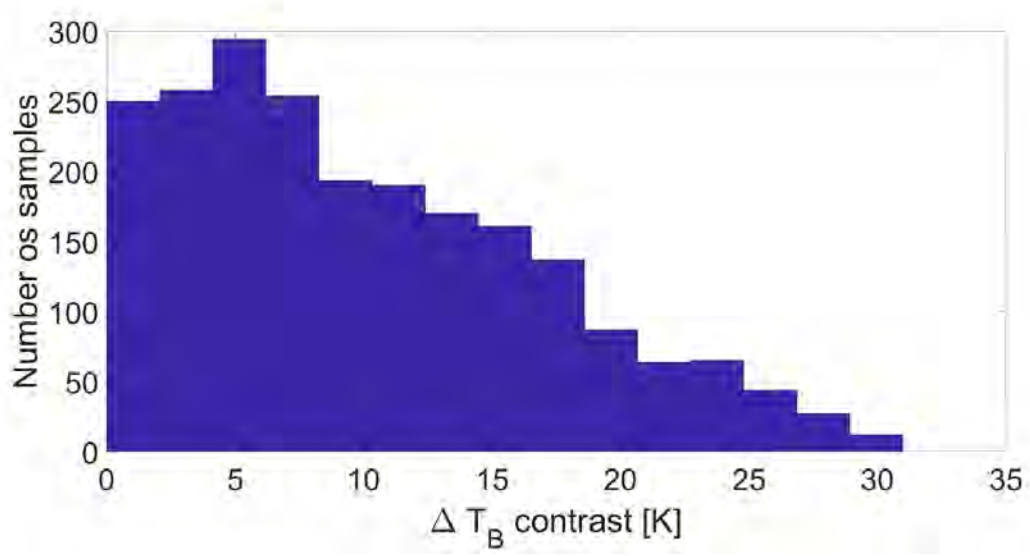



*Figure F.2: $\Delta T_{Bm}$ for Ross ice shelf considering all samples with positive value of $\Delta T_{Bm}$*

Then we have applied the fuzzy-logic approach only with $0$, reducing the dataset to 1349 samples with $\Delta T_{Bm}$ values shown in Fig. F.3

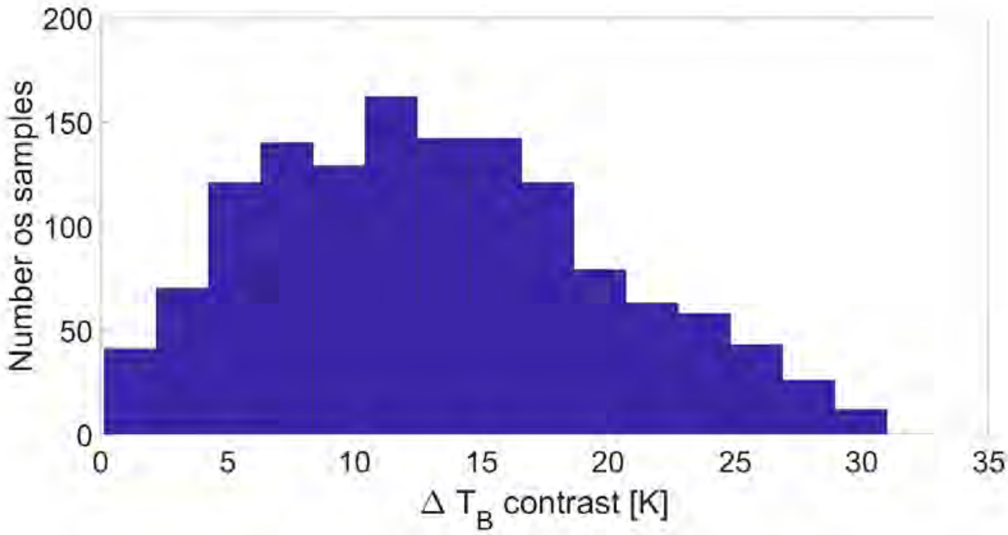



*Figure F.3: $\Delta T_{Bm}$ for Ross ice shelf after defuzzification considering only $M_2(\varepsilon)$*

To choose the threshold on $M_1(\Delta T_{Bm})$, we took the central value, obtaining 15 K. A similar approach it was made for other targets.

## BIBLIOGRAPHY

[1] P. Wessel and W. H. Smith, "A global, self-consistent, hierarchical, high-resolution shoreline database," Journal of Geophysical Research, vol. 101, no. B4, pp. 8741-8743, April 1996.

[2] W.E. Purdy et al., "Geolocation and pointing accuracy analysis for the WindSat sensor," IEEE Trans. Geosci. Remote Sens., vol. 44, no. 3, pp. 496-505, March 2006.

[3] 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 Trans. Geosci. Remote Sens., vol. 46, no. 4, pp. 913-922, April 2008.

[4] G. Heygster, H. Wiebe, G. Spreen , and L. Kaleschke, "AMSR-E geolocation and validation of sea ice concentrations based on 89 GHz Data," J. Remote Sens. Soc. Jpn., vol. 29, no. 1, pp. 226-235, January 2009.

[5] W. Berg, M. R. P. Sapiano, J. Horsman, and C. Kummerow, "Improved geolocation and Earth incidence angle information for a fundamental climate data record of the SSM/I sensors," IEEE Trans. Geosci. Remote Sens., vol. 51, no. 3, pp. 1501-1513, March 2013.

[6] H. Moradi, Meng, R. R. Ferraro, and S. Bilanow, "Correcting geolocation errors for microwave instruments aboard NOAA satellites," IEEE Trans. Geosci. Remote Sens., vol. 51, no. 6, pp. 3625-3637, June 2013.

[7] R. Bennartz, “On the use of SSM/I data in coastal regions. J. Atmos. Ocean Technol.”, 16, 417-431, 1999.

[8] Y. Han, F. Z. Weng, X. L. Zou, H. Yang, and D. Scott, ”Characterization of geolocation accuracy of Suomi NPP Advanced Technology Microwave Sounder measurements”, J Geophys Res-Atmos, 121(9), 4933-495010.1002/2015jd024278, 2016.

[9] Canny, J., “A computational approach to edge detection, ” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 6, 679-698, November 1986.

[10] Juneja, M., & Sandhu, P. S. Performance evaluation of edge detection techniques for images in spatial domain”. International journal of computer theory and Engineering, 2009, 1(5), 614.

[11] Vijayarani, S.; Vinupriya, M. “Performance analysis of canny and sobel edge detection algorithms in image mining”. International Journal of Innovative Research in Computer and Communication Engineering, 2013, 1.8: 1760-1767.

[12] Podder, P., Parvez, A. M. S., Yeasmin, M. N., & Khalil, M. I. “Relative Performance Analysis of Edge Detection Techniques in Iris Recognition System”. In 2018 International Conference on Current Trends towards Converging Technologies (ICCTCT), 2018, pp. 1-6. IEEE.

[13] Gupta, Samta, and Susmita Ghosh Mazumdar. "Sobel edge detection algorithm." International journal of computer science and management Research 2.2, 1578-1583, 2013

[14] Vincent, O. R., & Folorunso, O. “A descriptive algorithm for sobel image edge detection”. In Proceedings of Informing Science & IT Education Conference (InSITE), 2009, Vol. 40, pp. 97-107, California: Informing Science Institute.

[15] R.H. Thomas, D.R. MacAyeal, D.H. Eilers, and D.R. Gaylord, 1984. Glaciological Studies on the Ross Ice Shelf, Antarctica, 1972-1978. The Ross Ice Shelf: Glaciology and Geophysics Antarctic Research Series, Volume 42, Paper 2, 21-53.

[16] J. P. Lewis, “Fast Template Matching”, Vision Interface, p. 120-123, 1995.

[17] Manuel Guizar-Sicairos, Samuel T. Thurman, and James R. Fienup, "Efficient subpixel image registration algorithms," Opt. Lett. 33, pp. 156-158, 2008.

[18] 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 Trans. Geosci. Remote Sens., vol. 46, no. 4, pp. 913-922, April 2008.

[19] Kunkee, David B., et al. "Design and evaluation of the first special sensor microwave imager/sounder." IEEE Transactions on Geoscience and Remote Sensing 46.4, pp. 863-883, 2008.

[20] Prigent, C., Aires, F., Wang, D., Fox, S. and Harlow, C., 2016: Sea surface emissivity parameterization from microwaves to millimeter waves. Q.J.Roy. Meteorol. Soc. Accepted Author Manuscript. doi:10.1002/qj.2953

[21] Beck, H. E., Zimmermann, N. E., McVicar, T. R., Vergopolan, N., Berg, A., & Wood, E. F. “Present and future Köppen-Geiger climate classification maps at 1-km resolution. Scientific data”, 5, 180214, 2018

# GAMES

## "Geolocation Assessment/validation Methods for EPS-SG ICI and MWI"

In response to: EUMETSAT ITT 19/218140 "Development of Geolocation Validation Methods for EPS-SG ICI and MWI"

## Deliverable document 05 – D05
### "Final report – Task 2"

prepared by:**M. Montopoli and D. Casella (CNR ISAC)**
Contribution from **G. Panegrossi (CNR ISAC), B. Rydberg (Molflow), M. Papa and F.S. Marzano, (Sapienza University)**


| | |
|---|---|
| **Sapienza University of Rome**<br>**Center for Research in AeroSpace**<br>**(SUR-CRAS)** | |
| **Möller Data Workflow Systems AB (Molflow)** | |
| **National Research Council (CNR) of Italy**<br>**Institute of Atmospheric Sciences and Climate (ISAC)**<br>**(CNR-ISAC)** | |


Date: 22 January 2021

## LIST OF TRACKED CHANGES AND DOCUMENT VERSIONS

| Release | Personel | Comments |
|---|---|---|
| First release 26 June 2020 | Mario Montopoli: overall Editing and contributions on sections 1, 3, 4, 6 and 7. Daniele Casella: sections 2, 3, 4 5 and 7. Giulia Panegrossi provided some relevant guidance. | |
| Second release 16 July 2020 | as above | - Added material on section 4 Calculated Weighting functions for ICI for a standard mid latitude profile and explained which channels are more consistent with MSG water vapor channel Fig. 3.3.2 and reference main text,<br>- Calculated error scores for absolute geolocation using WVM and the 3 months dataset table 5.4.1.<br>- Added a simulated test using ICI4 as reference for relative geolocation (table 6.7.2). Modified conclusion on relative geolocation approach. |
| Third release 02 November 2020 | Casella, Panegrossi<br>Montopoli | - Overall corrections, rephrasing and restyling of the document<br>- Modification of results in table 6.7.1 and 6.7.2<br>- Substituted figure 6.7.2<br>- Rephrasing some parts of the conclusions for the relative geolocation |
| Fourth release 22 January 2021 | As above | - Correction of typos |

## LIST OF ACRONYMS

| | |
|---|---|
| AMSU | Advanced Microwave Sounding Unit |
| AR | Atmospheric river |
| ATMS | Advanced Technology Microwave Sounder |
| BT | Brightness Temperature |
| BG | Bakus Gilbert |
| DPR | Dual Precipitation Radar |
| DCC | Deep Convective Clouds |
| DEM | Digital Elevation Model |
| FCC | Filled Cross Correlation |
| FOV | Field Of View |
| GMI | GPM Microwave Imager |
| ICI | Ice Cloud Imager radiometer |
| MCCM | Masked Correlation Coefficient Matrix |
| MHS | Microwave Humidity Sounders |
| MW | Microwave |
| NOAA | National Oceanic and Atmospheric Administration |
| PMW | Passive MicroWave |
| RAOB | RAwinsonde OBservation |
| SEVIRI | Spinning Enhanced Visible and Infrared Imager |
| SSMIS | Special Sensor Microwave Imager Sounder |
| SUR | Sapienza University of Rome |
| $T_A$ | Antenna Temperature |
| TELSEM2 | Tool to Estimate Land.Surface Emissivities at Microwave version 2 |
| TESSEM2 | Tool to Estimate Sea Surface Emissivities at Microwave version 2 |
| WVM | Water Vapor Masses |

# 1 Introduction

## 1.1 Work package goal

The goal of this work package, within the GAMES (Geolocation Assessment/validation Methods for EPS-SG ICI and MWI) study, is to quantify the errors of the field of view geolocation of ICI, exploiting meteorological targets. Usually, ground targets are used to pursue the geolocation goal. Unfortunately, most of the ICI channels, namely those at frequency equal or greater than 325 GHz, have no chance to sample the surface features due to the strong gas absorption at those frequencies, thus preventing any ground-target based geolocation method from being utilised. Here is where the meteo-target based geolocation methods come mainly into play. Two distinct geolocation methods making use of meteo-targets are implemented: absolute and relative geolocation methods. The goal of the absolute geolocation is to estimate the geolocation error of a pivot ICI channel (eg. one of those around 183 GHz) with respect to external reference information. Contrarily, the relative geolocation is a self- referenced method since it uses ICI channel only, without relying on external auxiliary information, and it aims at finding the pointing error of the ICI channels (i.e those other than the pivot channel, for example at frequencies above 183 GHz) with respect to the pivot one.

The meteorological targets considered are the deep convective clouds (DCC) and water vapor features with strong gradients (e.g., atmospheric rivers, hereafter referenced as water vapor masses (WVM)), since these two target typologies are expected to be sufficiently detectable by the investigated ICI channels.

The effectiveness of the absolute geolocation method is assessed using actual observations from PMW sensors (e.g. SSMIS) and SEVIRI on board MSG. The GMI and spaceborne radar information (GPM DPR and Cloudsat CPR) are also used for verification. The rationale is to have the PMW 183.31 GHz channels of existing radiometers that mimic the 183 GHz ICI channels that need to be geolocated, whereas infrared SEVIRI channels act as reference. On the other hand, the relative geolocation is assessed using a simulated dataset of four ICI orbits. In this case, a reference ICI channel is assumed as already geolocated and its signature to specific atmospheric targets is compared with the rest ICI channels.

## 1.2 Summary of the main findings

**Absolute geolocation**

- Deep Convective Clouds (DCC) are difficult to use for geolocation because the signatures of such clouds between SEVIRI and PMW is not always consistent with each other, and because the parallax and distortion compensation caused by the different viewing geometry of SEVIRI and the PMW radiometer is not obvious to account for, and might introduce relevant errors that can strongly deteriorate the final geolocation result.
- Water Vapor Mass (WVM) features show a curvilinear profile due to a pronounced gradient in both MSG and PMW signatures. On top of this, WVM seems to be less affected by the parallax issue because WVMs have a smaller vertical extension than DCC. For these reasons WVMs are a better candidate for absolute geolocation than DCC.
- When considering the correlation between MSG and PMW detected WVM in terms of its spatial gradient, and for a period of three months, the estimated PMW geolocation error standard deviation is of the order of 3.6 km and the root mean square error (RMSE) is of the order of 5 km.

**Relative geolocation**

- Simulated signatures of WVM are not detectable by ICI simulated channels. The ice water path in clouds that often are found in correspondence with the WVM, dominates the ICI channel response.
- Simulations of ICI scenes for DCCs are more promising since those signatures are evident in all ICI channels.

- When considering the 183.31 GHz ICI-1 channel as reference, assuming ICI-1 being affected by a known absolute geolocation error, after remapping (using the Bakus Gilbert algorithm) all the other tested ICI channels on ICI-1,we can achieve estimates of the relative geolocation RMSE less than 3.0 km for channels up to ICI-5 and less than 5.0 km for the rest of the ICI channels. This results substantially improves when ICI-4V is considered as the reference channel, leading to relative geolocation RMSE of less than 3 km for ICI-1 and ICI-5 channels and more importantly an RMSE around 4 km for the other ICI channels with an error peak for ICI-8 that show RMSE around 4.15 km.

## 1.3 Document organization

The document starts with the discussion of the absolute geolocation method. This is accomplished describing the collected dataset of actual observations (Section 2), the achievable detectability of DCC and WVM (Section 3), the image correlation methods to be used (Section 4) and the absolute geolocation algorithms and relative results derived from their applications (Section 5). The description of the relative geolocation method and the associated discussion of the results are then provided (section 6). A Practical guidance is given in Section 7.

# 2 Dataset description

## 2.1 DCC Development Dataset Description

For the development and preliminary tests of absolute geolocation described in sections 3.2-3.3, a dataset of coincident observations from satellite-borne radar and MW radiometer and SEVIRI has been built. This dataset includes 11 case studies of DCCs localized in targeted areas observed within 15 minutes by all the instruments. The MW radiometer that has been used in this first dataset is the GPM Microwave Imager (GMI), equipped with two channels in the 183.31 GHz WV absorption band (i.e. 183.31±7 GHz and 183.31±3 GHz) at relatively high spatial resolution (see table 2.1.1 and table 2.1.2 for a comparison of some characteristics of the MW radiometers involved in this study).

***Table 2.1.1*** *183.31 GHz Channels of ICI, GMI and SSMIS.*

| **Sensor** | **Central frequency (GHz)** | **Bandwidth (MHz)** | **Pol.** | **IFOV (km)** |
|---|---|---|---|---|
| **ICI** | 183.31±7.0 | 2x2000 | V | 16 |
| | 183.31±3.4 | 2x2000 | V | 16 |
| | 183.31±2.0 | 2x2000 | V | 16 |
| **SSMIS F17** | 183.31 ± 6.6 | 1025 | H | 13.1x14.4 |
| | 183.31 ± 3.0 | 2038 | H | 13.1x14.4 |
| | 183.31 ± 1.0 | 3052 | H | 13.1x14.4 |
| **GMI** | 183.31 ± 7 | 2000 | V | 4.4x7.2 |
| | 183.31 ± 3 | 2000 | V | 4.4x7.2 |

***Table 2.1.2**: Orbital Characteristics of ICI, SSMIS and GMI*

| Sensor | Zenith angle | Swath | Altitude | ECT |
|---|---|---|---|---|
| **ICI** | 53,1° | 1700 km | 835 km | 09:30 desc |
| **SSMIS F17** | 53,1° | 1700 km | 848 km | 06:20 desc |
| **GMI** | 53° | 850 km | 407 km | Drifting 65° |

The first 10 DCC case studies have been chosen from a visual inspection of the 2B-CSATGPM dataset of CPR and GPM coincident observations (Turk 2015). The 11th case study has been chosen from Marra et al 2017. Table 2.1.3 summarizes the case studies in the development dataset.

***Table 2.1.3** Case studies of the DCC dataset.*

| Case Study Number | Case Study Date | Target Area Latitude Longitude | GEO-VIS/IR Radiometer | LEO-MW Radiometer | Other Coincident Observations |
|---|---|---|---|---|---|
| 1 | 23/082014 | 03N 30E | SEVIRI | GMI ATMS | A-Train |
| 2 | 24/03/2017 | 04S 32E | SEVIRI | GMI ATMS | A-Train |
| 3 | 17/11/2015 | 05N 52W | SEVIRI | GMI MHS | A-Train |
| 4 | 19/03/2016 | 06S 29E | SEVIRI | GMI | A-Train |
| 5 | 18/03/2016 | 11S 66E | SEVIRI | GMI ATMS | A-Train |
| 6 | 06/02/2017 | 12S 45W | SEVIRI | GMI ATMS | A-Train |
| 7 | 27/12/2015 | 13S 63E | SEVIRI | GMI | A-Train |
| 8 | 21/07/2016 | 18S 49W | SEVIRI | GMI ATMS | A-Train |
| 9 | 04/06/2015 | 22N 16E | SEVIRI | GMI ATMS | A-Train |
| 10 | 24/04/2014 | 34S 37E | SEVIRI | GMI | A-Train |
| 11 | 05/09/2013 | 41N 14E | SEVIRI | GMI MHS | - |

## 2.2 WVM Development Dataset Description

Similarly to the DCC development dataset in table 2.2.1, 6 case studies with clear WVM features have been selected. The list of case studies is in table 2.2.1. This dataset has been used to develop the WVM detection algorithm described in the following sections.

***Table 2.2.1*** *Case studies of the WVM dataset.*

| Case Study Number | Case Study Date | Target Area | GEO-VIS/IR Radiometer | LEO-MW Radiometer | WV pattern Typology |
|---|---|---|---|---|---|
| 1 | 25/10/2014 | Central Europe | SEVIRI | SSMIS-MHS | Atmospheric River |
| 2 | 19/02/2011 | Central Europe | SEVIRI | SSMIS-MHS | Atmospheric River |
| 3 | 12-13/09/2010 | Europe | SEVIRI | SSMIS-MHS | PV anomaly |
| 4 | 18/08/2012 | Atlantic Ocean | SEVIRI | SSMIS-MHS | Atmospheric River |
| 5 | 19/11/2009 | Africa | SEVIRI | SSMIS-MHS | Atmospheric River |
| 6 | 20/02/2010 | Atlantic Ocean | SEVIRI | SSMIS-MHS | Atmospheric River |

## 2.3 17-Days SSMIS-MSG Verification dataset

A third dataset of SSMIS and MSG measurements has been created t; it includes all DMSP-F17 orbits and MSG frames available for the 17 days listed in table 2.1.3 and 2.2.1. This dataset has been used to test the DCC and WVM algorithms described in section 5. The 17-Days SSMIS-MSG Dataset is composed of 180 SSMIS F17 orbits that have been divided into 203 15-minutes SSMIS orbit segments, each corresponding to a specific MSG snapshot.

## 2.4 3-Months SSMIS-MSG Dataset

A fourth larger dataset has been built for verification purposes. This dataset is composed of 90 days of all SSMIS DMSP-F17 orbits between 01/01/2017 and 31/03/2017. The SSMIS observations have been divided into 15-minutes 2871 orbit segments and synchronised with the corresponding MSG frame. SSMIS data have been downloaded from the SSMIS CSU Climate Data record (www.ncdc.noaa.gov/has) while the MSG data are the High Rate SEVIRI Level 1.5 Image Data - MSG - 0 degree and have been obtained from the Eumetsat archive (archive.eumetsat.int/usc/).

# 3. Detectability of atmospheric features for geolocation

## 3.1 Goal

The main goal of this section is to answer the question: can Deep Convective Clouds (DCC) and Water Vapour Mass (WVM) features be used as reference for MWI/ICI geolocation absolute error assessment?

The main aspects and issues relevant for this study to be addressed in the use of DCCs as geolocation targets (Method 1) are first summarized with reference to recent literature. A particular focus on the correction of the distortions due to observation geometry will be given. Then the WVM features are critically analysed (Method 2) as possible geolocation targets, focusing on the issues related to the differences in the clear sky Weighting Functions in the IR and in the MW spectrum

For both DCC and WVM, it has been assumed that the reference for assessing the geolocation errors comes from Meteosat geostationary satellites, and specifically from the MSG series of satellites in the 0-degree orbit. It has been also assumed that both the methods should rely on the 183.31 GHz water vapour absorption band channels of existing radiometers with conical scanning geometry (specifically GMI and SSMIS).

## 3.2 Characteristics and detectability of DCC

### 3.2.1 General characteristics of DCC

Deep Convective Clouds (DCC) have been studied by a number of authors in recent years especially because they play a central role in the transport of air and chemical species from the troposphere to the stratosphere. An overshooting convective cloud top is defined by the American Meteorological Society's Glossary of Meteorology (Glickman 2000) as ''a domelike protrusion above a cumulonimbus anvil, representing the intrusion of an updraft through its equilibrium level.'' Overshooting tops (OTs) indicate the presence of a deep convective storm with an updraft of sufficient strength to penetrate through the tropopause and into the lower stratosphere.

Alcala and Dessler (2002) and Liu and Zipser (2005) presented the properties of OTs and their distribution over the tropics with the precipitation radar on board the TRMM satellite. More recently, Liu and Liu (2015) characterized tropopause-reaching deep convection analysing 1 year of GPM Ku-band radar echoes in relation with several reference levels derived from the ERA-Interim reanalysis data set. In order to summarize their results for the objective of the present study, and focusing on the region observed by the MSG 0-Deg satellites, some interesting conclusions can be highlighted from Liu and Liu (2015) study:

1. Most of the Deep convection is found over land areas.

2. The regions where these phenomena are more common are central Africa and mid-latitude Europe (see Figure 3.2.1)

3. Deep convection shows a pronounced seasonal cycle with peaks in Summer months in mid-latitude regions and in Autumn-Spring months in the inter tropical region (see Figure 3.2.2)

4. A clear daily cycle is also present over land areas with a strong peak in the afternoon hours (figure 3.2.3).

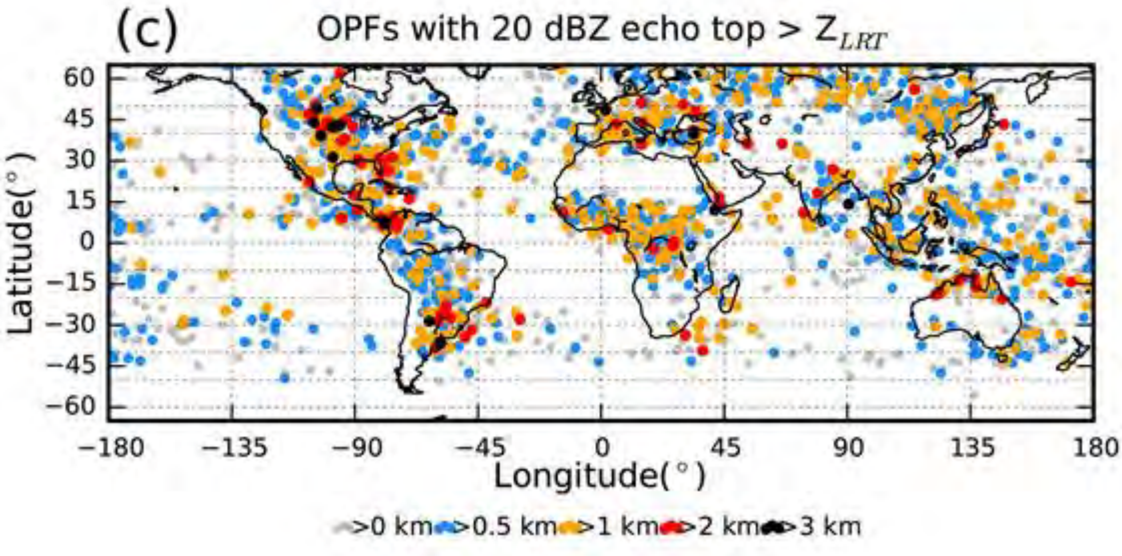



***Figure 3.2.1*** *Locations of Overshooting Tops identified with the tropopause definition given by WMO. The overshooting tops are categorized by the distance above the tropopause are shown in symbols of different colours. From Liu and Liu (2015)*

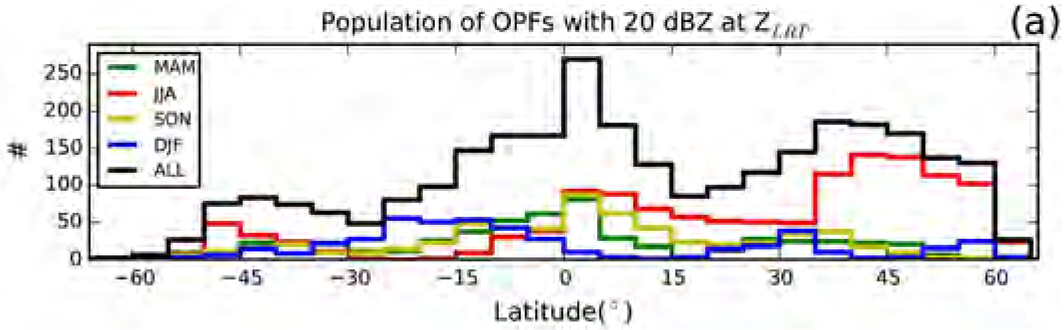



***Figure 3.2.2*** *(a) Zonal distribution of populations of Overshooting Tops. The occurrence in each 5° zone is calculated by dividing 20 dBZ pixels at the tropopause with total sampled pixels. From Liu and Liu (2015)*

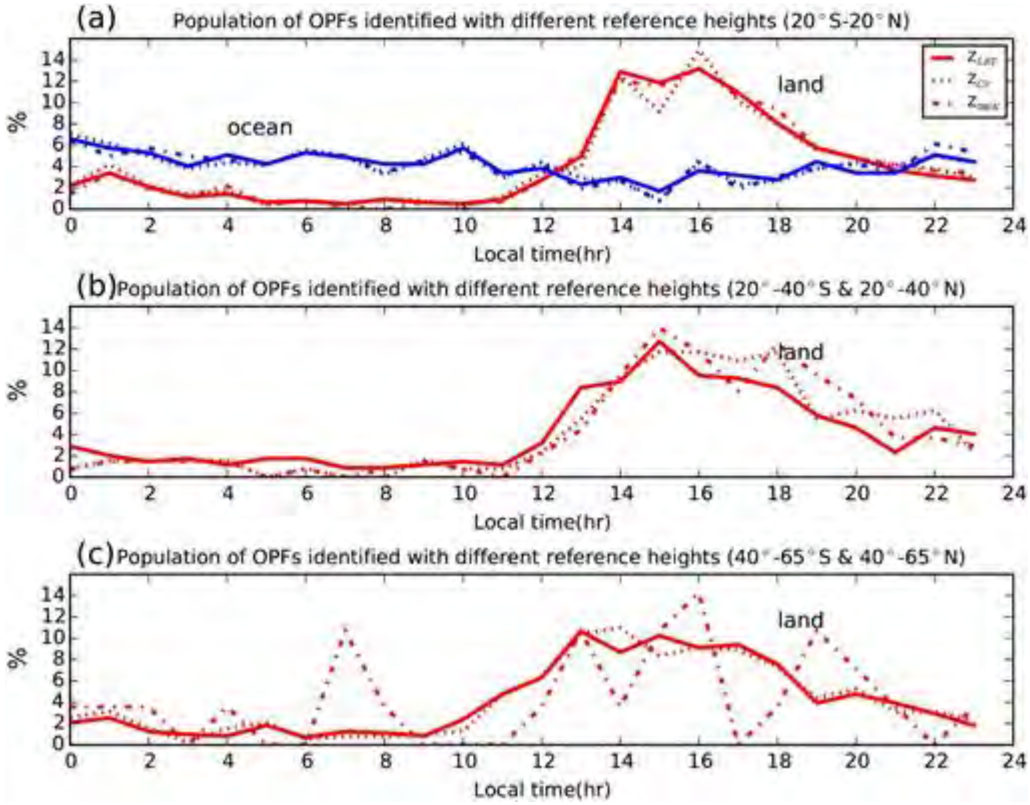



***Figure 3.2.3*** *Diurnal variation of population of overshooting precipitation features defined with different reference levels over land (red) and ocean (blue). (a) Over 20°S–20°N. (b) Over 20°S–40°S and 20°N–40°N. (c) Over 40°S–65°S and 40°N–65°N. From Liu and Liu (2015)*

Another very important information for the GAMES objectives comes from the Liu and Liu (2015) study in terms of the typical height of OTs that is estimated in 17-18 km near the equator, 8-18 km near the tropics and 6-12 km at mid-latitude. Moreover, the distortion due to the viewing geometry of the sensor (both from GEO and LEO satellites) for observation angles far from the nadir is of the same order of magnitude as the cloud top height. Considering that the geolocation accuracy requirement of the methods developed in the GAMES project is 2.5 km, it is clear that **a strong prerequisite for using DCC as a possible target for assessing the geolocation accuracy of ICI is a very accurate correction of the distortion due to the viewing geometry of the sensor**.

### 3.2.2 Detectability of Deep Convective Clouds from satellite

The detection of DCCs based on the 183.31 GHz band channels has been based on the work of Hong et al. (2005). This work has been used by several authors for studies related to deep convection and precipitation estimates from MW radiometers (e.g. Funatsu et al. 2009, Sanò et al. 2015, Ferraro et al 2015). The Hong et al. (2005) method for detecting DCC is based on a series of simple tests summarized by the formula:

$$_{17} > \ _{13} > \ _{37} > 0\ \mathrm{K} \qquad (1)$$

Where the $\Delta T_{ij}$ refers to the differences between two channels in the 183.31 GHz band, e.g.

$$_{17} = TB_{183.31\pm7} - TB_{183.31\pm1} \qquad (2)$$

This method is based on the fact that the channels in the 183.31 GHz absorption band in clear sky conditions have weighting functions (WFs) that peak at different heights within the troposphere (lowest for the 183±7 GHz and highest for the 183±1 GHz channel). This, together with the negative temperature lapse rate in the troposphere,makes the clear-sky TBs in the 183.31 GHz channels be ordered as:

$$TB_{183.31\pm7} > TB_{183.31\pm3} > TB_{183.31\pm1} \tag{3}$$

A deep convective cloud is optically thick and the TBs are sensitive to the upper cloud layers and to the presence of water vapor above the cloud top. Deep convection is associated with the transport of vapor into the stratosphere, where the temperature increases with height. Therefore, the order of the 183.31 GHz channels becomes:

$$TB_{183.31\pm1} > TB_{183.31\pm3} > TB_{183.31\pm7} \tag{4}$$

The series of tests in Hong et al. (2005) in (1) are carried out to verify the 183.31 GHz channel TB relations described in (4).

The detection of DCC and in particular of the OT region in IR and VIS from geostationary imagery has a long track record. Some methods are based on the difference between the IR thermal channels (e.g. 10.8 µm) and a WV channel (e.g. 6.4 µm). This approach has been developed by Setvak et al. (2007), and it relies on a mechanism similar to the Hong et al. (2005) method: the WV channel has a higher WF peak than the thermal IR channel, therefore an inverse order of the WV-IR TBs is found when there is water vapor in the stratosphere and when the cloud is optically opaque (i.e., in presence of DCC). Other methods (e.g. Bedka et al. 2010, Sun et al. 2019) rely on the detection of gradients of TB in the IR thermal channels in the proximity of the OTs due to the physical temperature evolution of the updraft: it is well known that a developing convective cloud extending upward has a top temperature that is lower than the environment temperature, and higher than the wet adiabatic, and follows a lapse rate of 7-9 K/km (Negri 1982, Adler et al 1983). Therefore, the OT detection method based on the gradient in the thermal IR channels is founded on the fact that the OT appears as a cooler region than the surrounding anvil cloud even if it develops above the tropopause. Finally, the last category of methods for OTs detection is based on the comparison between the cloud top height (CTH) and the tropopause height; OTs are defined as the cloud top region higher than the tropopause. In the present study several methodologies for OTs detection have been tested, from the WV-IR difference method (Setvak et al. 2007), to the texture/gradient thermal IR method (Bedka et al 2010), to CTH-tropopause height comparison. The final algorithm developed for this study takes into account two methods: a preliminary test on the thermal IR channel ($TB_{10.8mm} < 215$ K) used also by Bedka et al. (2010) as a preliminary test (the authors affirm that this simple test is able to identify 96% of the OTs with relatively frequent false alarms). The second test based on the CTH- tropopause height comparison,, (this method accuracy has a large dependence on both an accurate estimate of the CTH and of the tropopause height).

### 3.2.3 Conditions for using DCC geolocation targets: cloud top height and parallax correction computation in the Infrared.

As already said (see section 3.2.1) an important requisite for using DCC as a possible target for assessing the geolocation accuracy of ICI is a correction of the distortion due to the viewing geometry of the sensor (parallax correction). This correction, however, needs an accurate knowledge of the CTH. In the present study we have tested two CTH products for this purpose, the first is the CTH-MSG - 0-degree Eumetsat product. This product has shown to be unusable for the objectives of GAMES, the main reason being the spatial resolution that is 4x4 MSG IR pixels and that is insufficient to resolve the relatively small OTs, that usually are smaller that 25 km in diameter. The second product tested is the Optimal Cloud Analysis (OCA) product that estimates the cloud top pressure, optical thickness and mean radius from SEVIRI using an optimal estimation method. A third product has also been developed during GAMES for CTH estimate. This product makes use of a temperature profile taken from NWP model (ECMWF reanalysis ERA-5) and adopts the assumption that the thermal IR SEVIRI channel at 10.8 μm is a good estimate of the cloud top temperature. The cloud top temperature is transformed in CTH through the use of the model T profile. In order to take into account the characteristic behaviour of updraft in terms of cloud top temperature, the temperature profile is modified above the tropopause level and substituted with a constant lapse rate of 8 K/km, that is a mean value for the lapse rate within the updraft region of the cloud (see Negri 1982 and Adler et al. 1983). Summarizing the CTH algorithm follows a series of steps:

1. ERA 5 ECMWF reanalysis has been used
2. Temperature and pressure vertical profiles in proximity of the region of interest in space and time (within 1 hour) are selected;
3. Pressure levels between 400-50 hPa are selected;
4. Profiles are averaged over the region of interest;
5. The tropopause is identified using WMO definition – lower (in height) lapse rate < 2 K/km;
6. All T values at levels above the tropopause are modified with a lapse rate of 8 K/km.
7. The CTH is estimated by comparing the T profile obtained with the IR 10.8 μm channel TB value;

This CTH estimate method has been compared (together with the OCA CTH product) with cloud top height from CPR, the results are shown in Figure 3.2.4 and Figure 3.2.5 for the 10 case studies where the CPR was available, considering only the areas where deep convective clouds have been detected. The resulting CTH shows a relatively high correlation with CPR CTH and a relatively small mean error (0.4 km versus 0.6 km of OCA-CTH). The RMSE is not very small (about 1 km), but smaller than OCA CTH (almost 2 km), and the correlation coefficient is close to 0.8 (vs. 0.26 for the OCA CTH product) .

| | OCA | Est |
|---|---|---|
| RMSE [km] | 1.9177 | 1.0255 |
| Mean Err [km] | 0.6189 | 0.4265 |
| Corr. | 0.2573 | 0.7977 |

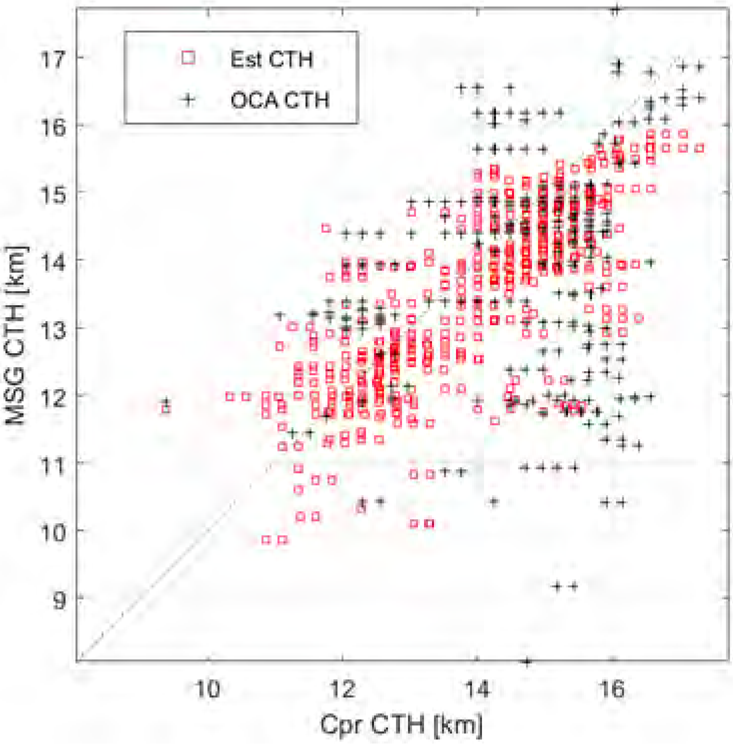



***Figure 3.2.4*** *Comparison of MSG estimated CTH and OCA CTH with CPR CTH, some statistics are also shown.*

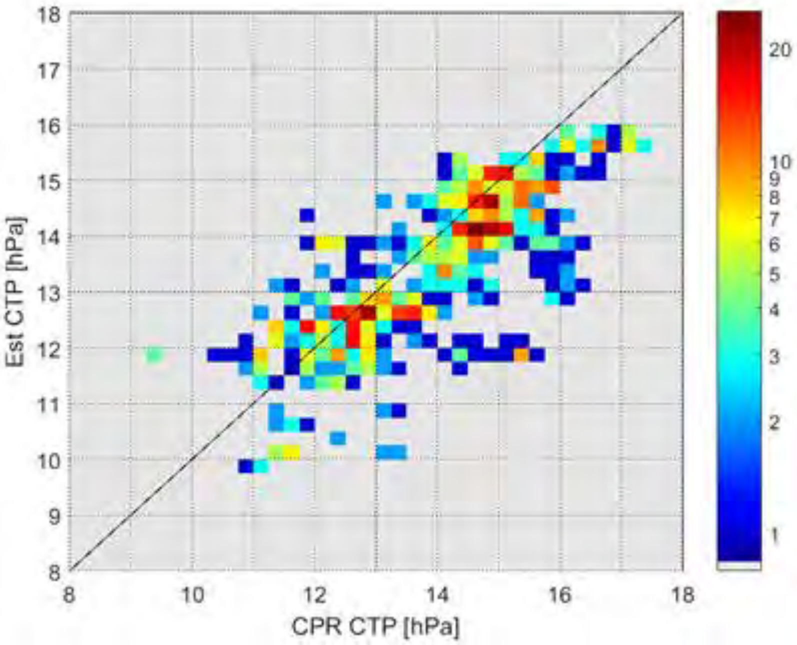



***Figure 3.2.5*** *Density scatterplot of MSG estimated Cloud Top Pressure (CTP) and CloudSat CPR CTP.*

The CTH obtained using the T profile is not perfect and other CTH products could be tested in order to obtain better estimates. The fact that the CTH algorithm described here gives better results than the actual operational products could be due to the fact that it is optimized for deep convective clouds that are on a global scale a small fraction of the clouds observed by SEVIRI.

The CTH is then used in order to correct the parallax effect. The parallax correction algorithm is simple, the position of the satellite is taken from the ancillary data of the sensor, either MW-LEO or IR-GEO. Then, the azimuth and elevation angles of the satellite w.r.t each pixel center observed by the satellite are calculated, through a change of reference system from geodetic to local spherical coordinates (using the MATLAB function geodetic2aer ). Then the parallax shift is calculated using the formula:

$$= \quad ( \; ) = \quad ( \; ) \tag{5}$$

where $\varepsilon$ is the satellite elevation angle and $\beta$ is the observation angle. This simple formula does not take into account the curvature of the Earth, however for the observation angles of this study (i.e. the SEVIRI observation angle have been limited to 27.4°) the errors due to considering the Earth as flat in proximity of the cloud are smaller than 3 cm for an extreme cloud top height value of 20 km.

### 3.2.4 Cloud Top Height and Parallax Correction in the Microwave

The parallax compensation for the IR data has been described in the previous section, however the MW radiometers with conical observation geometry, including MWI and ICI, have an observation angle around 53° that is almost constant along the scan. A parallax correction is therefore needed and the CTH information needs to be calculated also for the MW grid. Figure 3.2.6 shows the scheme that has been followed to translate the CTH information from the IR grid to the MW and the effect on a real case study of each module:

1. First the CTH is calculated from the IR TBs as described in section 3.2.3;
2. Than the CTH is corrected for IR parallax using Eq. 5;
3. Than the IR-Parallax-Corrected (PC) CTH are projected to the line of sight of the MW radiometer (this procedure is equivalent to parallax correction in Eq.5 with a "-" sign before );
4. Than the CTH are convolved with a Gaussian filter (approximating the MW antenna pattern)
5. Finally the CTH are used to compensate for the parallax of the MW observation geometry.

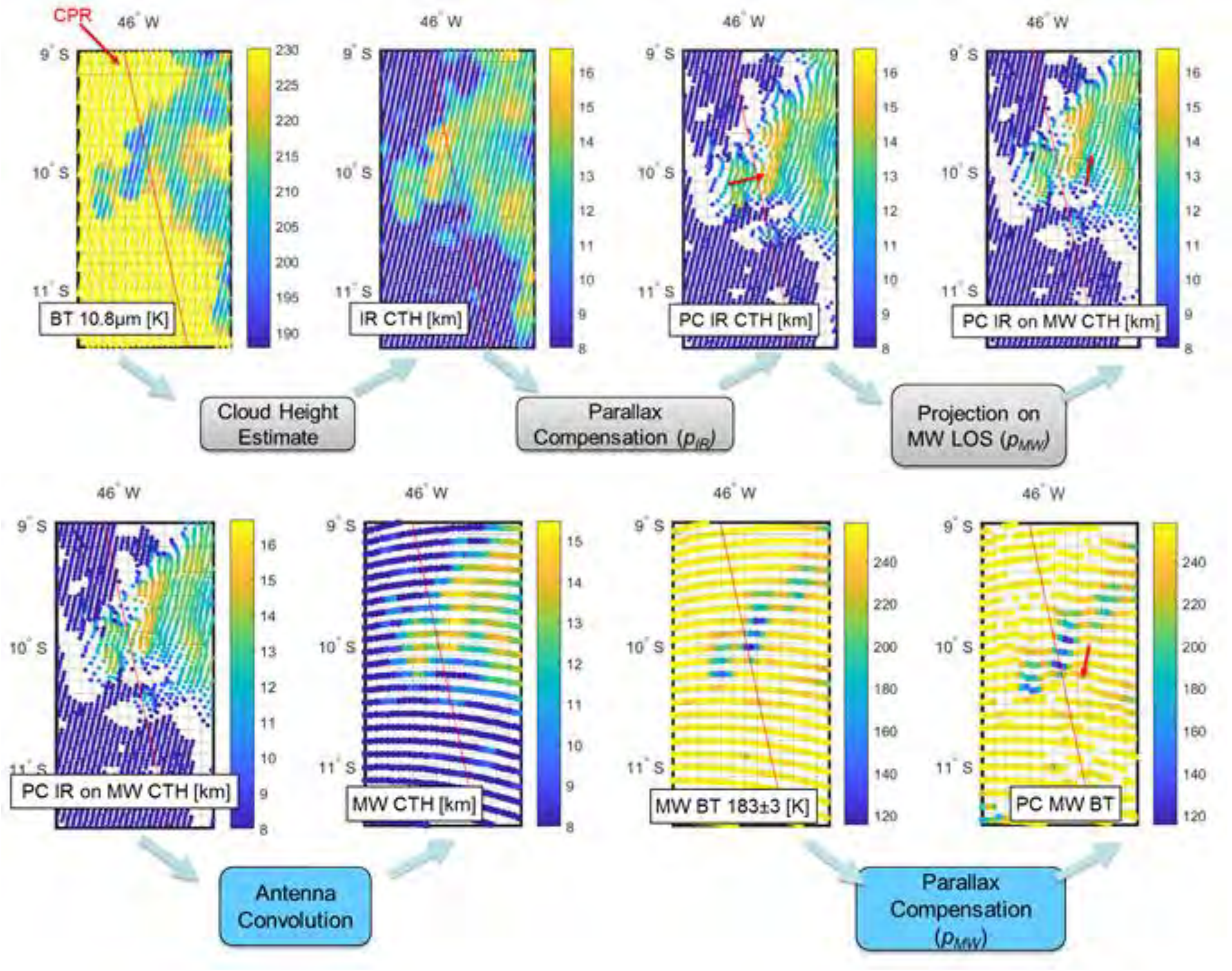



***Figure 3.2.6** Scheme of Parallax Compensation for MW with an example of application to a DCC target.*

Figure 3.2.7 shows an example of application of the parallax correction in one of the 11 case studies. The uncorrected TBs are shown together with the CPR reflectivity in the top panel. While the IR TB seems to match fairly well with the DCC clouds, due probably also to the relatively small angle of observation from SEVIRI in this case (~20°), the MW TBs are clearly mismatched. In the bottom panel conversely, the TBs corrected for parallax match very well with the radar both in the MW and in IR.

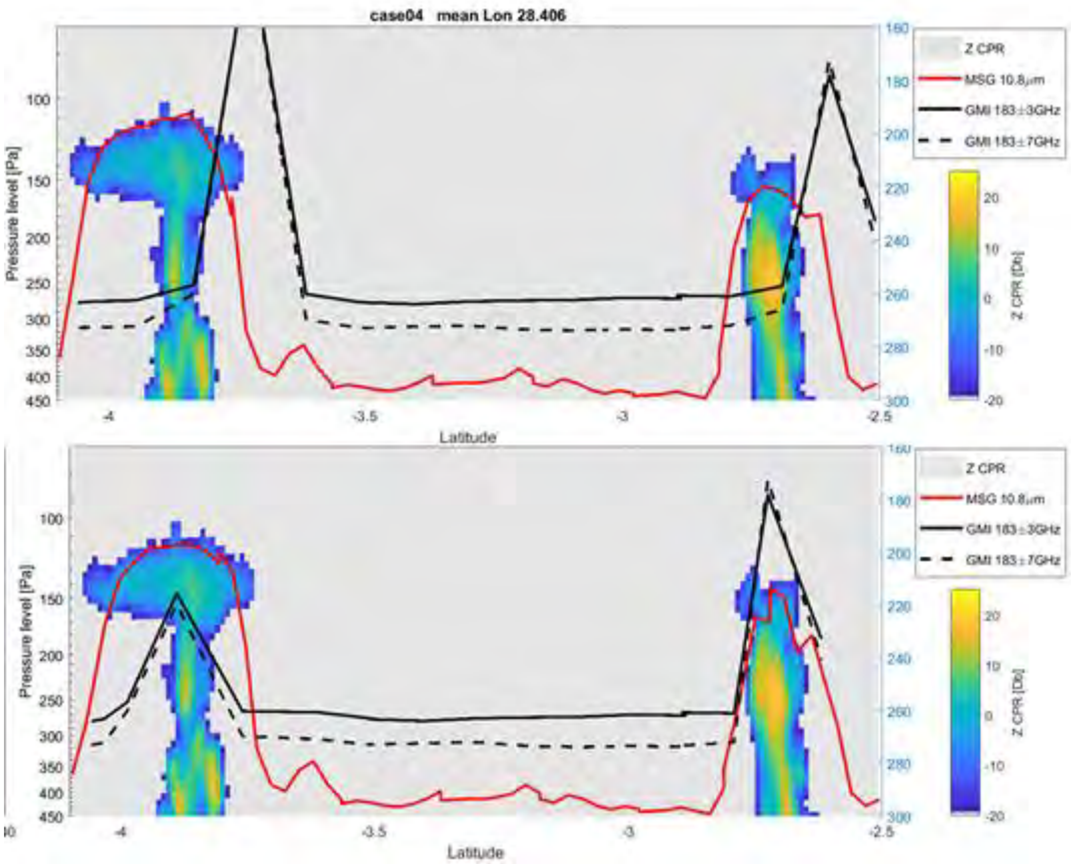



***Figure 3.2.7*** *Example of application of the parallax correction for both MW and IR for the case study n. 4 in Table 2.1.3. Top panel shows the CPR reflectivity vertical cross-section, and TBs (right hand y-axis) for SEVIRI 10.8 μm and GMI 183±1 GHz and 183±3 GHz channels nearest to the CPR track without parallax correction. Bottom panel shows the same variables with IR and MW parallax correction.*

## 3.3 Characteristics and detectability of WVM

One of the main goals of GAMES in Task 2 is to understand if water vapor mass features (WVM) can be used as geolocation targets. Comparing any scene from SEVIRI in one of the water vapor absorption band channels (e.g. 6.2 μm or 7.3 μm) with the same scene observed in the 183.31 GHz absorption band many similarities appear. Cloud covered areas, and particularly tick and high clouds - including deep convective clouds, show colder TBs than the clear sky areas. However, the correlations between MW and IR in the WV absorption bands in cloud covered areas are prone to the same complexities of the relations between the thermal channels in IR and MW. In clear sky conditions both MW and IR in the WV absorption bands show similar features, depending both on the distribution of water vapor over the scene. A condition to use the WVM in clear sky as a geolocation target is that the observed feature shows a strong horizontal gradient and a sharp shape. We assume that the main conditions to use WVM as geolocation targets are:

1. Clear sky conditions;
2. High contrast: presence of strong horizontal gradient of TBs in both MW and IR;
3. Sharpness: the feature should have a sharp (well identifiable) shape;

4. Similar weighting functions: IR and MW WV channels to be compared should be associated with similar clear-sky weighting functions;

5. Negligible parallax correction.

A good candidate to fulfil these conditions are the stratospheric intrusions, i.e., intrusions of dry air into the tropopause, and appear in both IR and WV as sharp filaments of relatively warmer TBs with a relatively strong horizontal gradient. The last condition in particular may be satisfied due to the fact that the weighting functions (both in IR and MW) in dry condition show peaks at lower levels, reducing the impact of the different viewing geometry.

### 3.3.1 Conditions for using WVM as geolocation targets: Weighting Functions

The conditions summarized in the previous section are partially related to the selection of the scene and of the WVM feature to be used as a geolocation target. Conditions 1-3 in particular could be met by a proper selection of the regions of interest in order to select only the features that satisfy them. Conversely, this section will discuss how condition 4 can be met by using a linear combination of channels. Condition 5 is very difficult to be fulfilled, due to the fact that parallax correction of a water vapor feature depends on the weighting function (WF), and primarily to the height of the peak of the WF. Figure 3.3.1 shows the WFs of SEVIRI and SSMIS for a standard mid-latitude summer temperature and water vapor profile. SEVIRI WFs have been calculated at nadir while for SSMIS at 53.1°. From figure 3.3.1 it is clear how the peak of the SEVIRI channel at 6.2 μm is located around 350 hPa between the peaks of the SSMIS 183±1 GHz and 183±3 GHz channels. Therefore, it is possible to combine these two SSMIS channels in order to obtain a WF more similar to the SEVIRI one at 6.2 μm. The purple line in the right panel of figure 3.3.1 shows the WF resulting from the linear combination of SSMIS 183.31±1 GHz and 183.31±3 GHz channels called $TB_x$, and given by:

$$= ( \quad _{183\pm1} + \quad _{183\pm3})/2 \tag{6a}$$

The linear combination of TBs is equivalent to the linear combination of the Weighting Functions if the channel is opaque (i.e. when the contributions to the measured TB from the surface – reflection and emission terms- are negligible), in this case:

$$_1 + \quad _2 = \int_0^\infty \quad ( \, , \, )\left( \frac{\ \ _1}{} + \frac{\ \ _2}{} \right) \quad = \int_0^\infty \quad ( \, , \, )( \quad _1 + \quad _2 ) \tag{6b}$$

Where, 1 and 2 indicate the two SSMIS channels (183.31±1 GHz and 183.31±3 GHz), τ is the optical thickness, B(T,ν) is the Planck function (assumed to be the same for the two channels at frequencies ν1 and ν2).

In this study we have compared TBx derived from SSMIS defined in Eq. 6 with the SEVIRI 6.2 μm channel, for WVM features in clear sky conditions. Results are summarized in section 5.3 and 5.4.

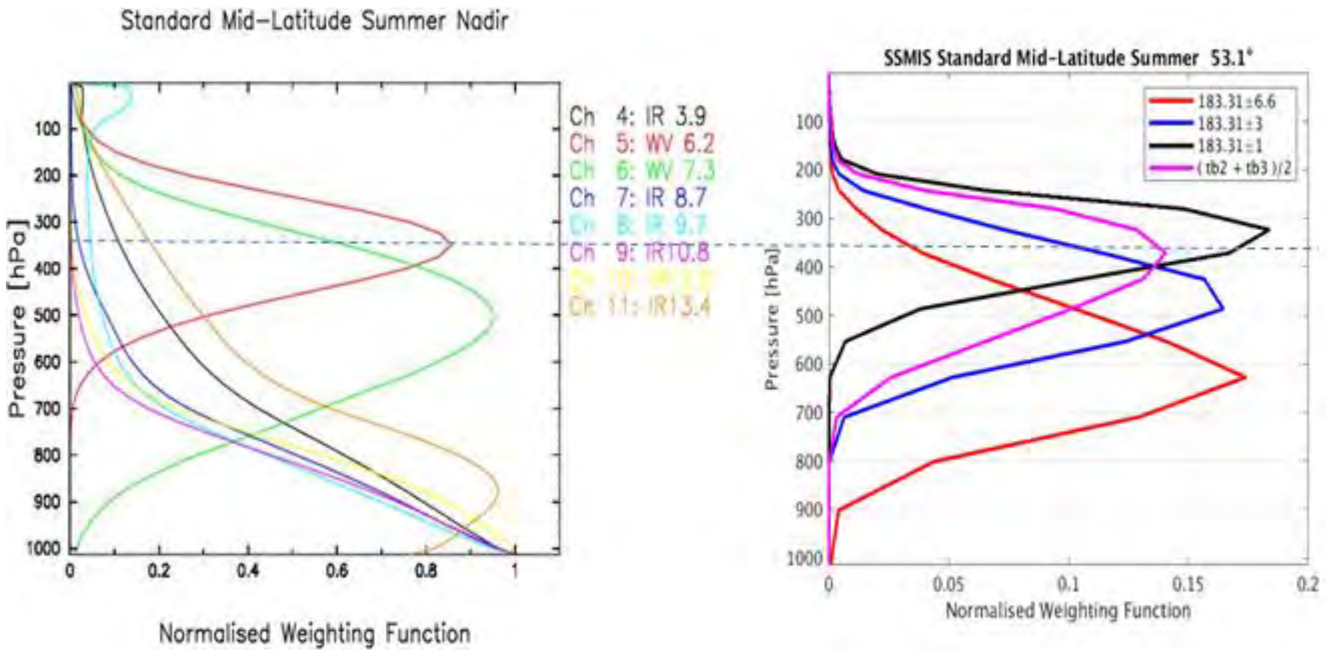



***Figure 3.3.1** Weighting Functions of SEVIRI (left) and of 183.31 GHz band SSMIS for a standard Mid-Latitude Summer Profile.*

Figure 3.3.2 shows the weighting functions for the ICI channels at frequencies higher than 183.31 GHz for the same atmospheric profile of figure 3.3.1. It is clear that at least three ICI channels: ICI-7 (325.15±1.5 GHz), ICI-8 (448±7.2 GHz) and ICI-11 (664±4.2 GHz) show similar weighting functions peaking nearly around 350 hPa. Thus, ICI-7, ICI-8 and ICI-11could be used with ICI channel at 183.31 GHz to detect WVM features in a consistent way with SEVIRI 6.2 um water vapor channel. In summary, the analysis of the ICI weighting functions suggests that the use of WVM for ICI absolute geolocation could be effectively accomplished exploring various ICI channels (not only those at 183.31 GHz) albeit this hypothesis needs to be tested with actual data that are not available at the time of writing of this report.

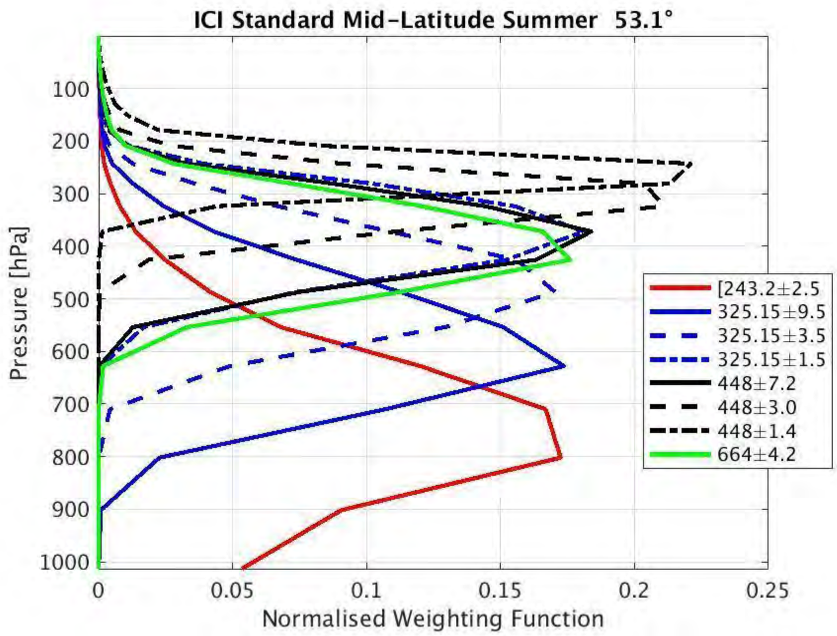



*Figure 3.3.2 Weighting Functions of ICI for channels at frequency higher than 183.31 GHz for a standard Mid-Latitude Summer Profile.*

# 4. Correlation methods

## 4.1 Goal

The goal of this section is to define the correlation metric necessary to measure the degree of similarity and the spatial horizontal shift between a reference and a tested scene in terms of deep convective clouds (DCC) or water vapour masses (WVM) features. Correlation metric is a common tool that can be used in the absolute geolocation approaches as well as in the relative ones. In the former case, the reference observation is derived from SEVIRI IR channels which is correlated with the tested scene that comes from the PMW radiometer in the 183 GHz band. Contrarily, in the relative geolocation case, the correlation is tested among the ICI-MW channels when one of them is assumed to be the reference.

Independently from the geolocation approach considered, the rationale usually followed is to maximise the correlation between reference and tested TB channels of DCC or WVM scenes (i.e. between IR and PMW TBs channels in case of absolute geolocation or between ICI channels in case of relative geolocation). The estimated horizontal shift between the reference and tested scene is the one that maximise the correlation between the two scenes.

To implement any correlation approach, some requirements need to be satisfied by the input scenes. Firstly, the reference and tested scenes need to be described in the same reference grid (and in some cases a grid remapping is necessary before computing the correlation). Secondly, some correlation approaches may require to have the input data in matrix form to be able to implement the 2D discrete Fourier Transform that allows speeding up the computation time and/or implement some refined correlation methods. This implies that void values that lay in the analysed domain (eg. those values that we do not want to consider in the correlation analysis but are present in the analysed domain) needs to be carefully replaced in some way.

In the following subsections the strategies used to correlate the reference and tested scenes are discussed. Their use is described in more detail in sections 5 and 6 where absolute and relative geolocation algorithms are investigated.

## 4.2 Correlation strategy for absolute geolocation

The input for the correlation is the TB within selected spatial domains called Region of Interests (ROIs). ROIs can include DCC or WVM depending on the target typology under investigation. Methods for filtering out some unwanted features, that are represented by those pixels inside the ROI that do not belong to DCC or WVM target typology, need to be implemented. For example, for DCC targets we want to eliminate (i.e. label them as void values) those pixels within the analysed ROI that do not show the typical signatures of DCC. Contrarily for the WVM target, it is necessary to screen out all the clouds that can potentially mask the scene within the ROI. As a consequence, the way used to manage the void values has led to two distinct approaches: Masked Correlation Coefficient Matrix (MCCM) and Filled Cross Correlation (FCC).

### 4.2.1 Masked Correlation Coefficient Matrix (MCCM)

The Masked Correlation Coefficient Matrix (MCCM) is based on the diagram in Figure 4.2.1. The Parallax corrected TBs in the IR 10.8 m reference channel and the tested SSMIS channel at 183.31±1 GHz are the input of the procedure. The latter is pre-processed in order to identify the target of interest. For example, in Figure 4.2.1 only DCC targets in the SSMIS scene are considered within the identified ROI, whereas the rest is set to void pixels (shown in white). Then the procedure begins by performing a progressive horizontal shift of the reference IR TBs in the latitude and longitude domain with a fixed step ($\delta\theta,\delta\phi$). IR TBs are subsequently convolved with a gaussian filter approximating the MW antenna pattern in order to map the IR scene into the MW grid. In a further step, the correlation coefficient between non-void pixels (i.e. DCC pixels in the MW scene), is calculated. The procedure is repeated for various multiples of (δθ,δϕ) in order to fill a correlation coefficient matrix. In the actual algorithm the correlation matrix has been built with 20 by 20 steps of (δθ, δϕ) both equal to 0.02°. The (δθ,δϕ) spacing corresponds to the minimum shift that is reachable by the algorithm or to the algorithm sensitivity. In order to increase the sensitivity, after a maximum has been found, the whole procedure is repeated around the maximum with finer 20 by 20 shifts of (δθ,δϕ) equal to 0.002°.

### 4.2.2 Filled Cross Correlation (FCC)

The Filled Cross Correlation (FCC) overcomes the limitations due to the void values produced by filtering out the unwanted features within the ROI. In this case the IR and MW inputs are first screened substituting the unwanted pixels with some constant filling values, then interpolated to a regularly spaced grid (a 1 km spaced grid has been used) and finally compared calculating a cross correlation function. Similarly to the landmark approach in Task 1, The normxcorr2 of MATLAB is used to calculate the correlation function. This method has been applied to WVM targets, considering as input the horizontal gradient of the TBs (both in the IR and in the MW) and using as constant filling values for screened pixels 0 K/km. FCC is largely more efficient in terms of computation time than MCCM (~10 times faster), however some degree of uncertainty is introduced when both TBs are interpolated to a common grid that has a higher grid spacing than both IR and MW resolution. In fact, the interpolation method that is used has a fairly strong impact on the results.  In this work we tested the impact of using a bilinear interpolation and a cubic interpolation.

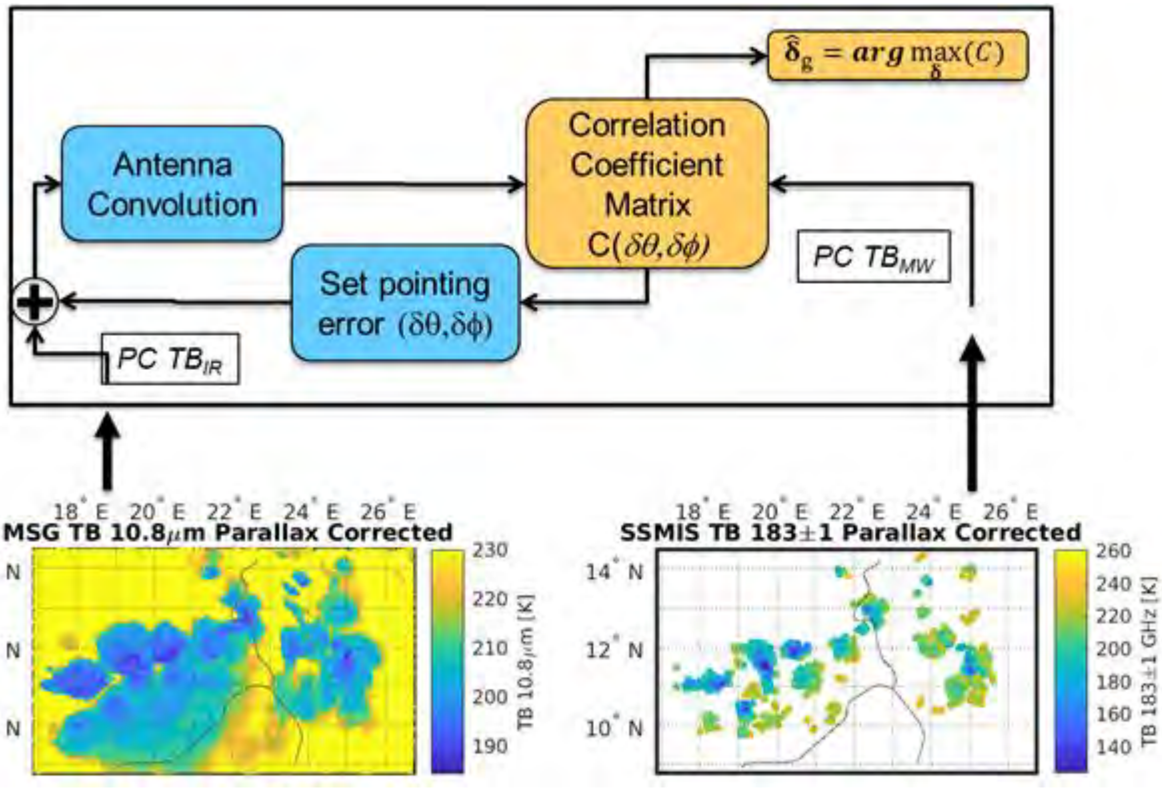



***Figure 4.2.1*** *Diagram of the Masked Correlation Coefficient Matrix with input examples of DCC targets*

## 4.2 Correlation strategy for relative geolocation

As it will be clarified in Section 6, the relative geolocation method makes use of a correlation approach similar to the FCC method previously described. The reference and the tested ICI channel are correlated with each other and the outcome is a correlation function whose maximum peak position gives an estimate of the horizontal displacement between the considered channels. Since the relative geolocation mainly focuses on DCC targets only, and since only ICI MW channels are considered, void values are not present in the analysed ROIs. Thus, in this respect the procedure is a little bit easier to apply.

## 4.3 Correlation metric tools

In the previously described correlation methods, a large use is made of the correlation function. To compute it we identified two tools: normxcorr2 function and the routine proposed by Guizar-Sicairos et al. (2008) (GS2008). Both of them are implemented in MATLAB and are based on fast Fourier Transform that allows speeding up the computation time. However, the difference between the two tools is not only in the computation time since the GS2008 foresees two steps search to find the best mismatch between the input images. In this section we show some synthetic experiments comparing the performance of the two routines.

Figure 4.3.1 shows the setup of the first experiment where we artificially created two DCC like images, the reference and the shifted one (left and middle panels respectively) and displayed the norxcorr2 correlation function between the two images (right panels) overimposing the estimated displacements (red squares for normxcorr2 and cyan diamonds for and GS2008). The true displacement is represented by a black square. In the shifted image we did not perform a shift only (case a in the upper panels) but we also varied the footprint size and introduced some noise (cases b and c, respectively). The results are different as a function of the Noise and the footprint size variations introduced. Only solid shifts without noise and footprint size variations produce a perfect estimate (top right panel). It is worth noting that the footprint size variations produce the multiple peak issue in the normxcorr2 (red squares in case b right panel) and a considerable displacement error in the GS2008 approach. To have an idea of the overall result we varied the original size in the reference image by a factor ±50% and introduced a normally distributed noise with standard deviation varied within 10% to 100% of the input signal amplitude. For each noise variation step, 50 realizations are generated. Doing so, we obtained the results in Figure 4.3.2 where the GS2008 outperforms normxcorr2 in all considered cases. We also verified (not shown) the use of gradients for the cases shown in Figure 4.3.1 but we did not find any improvement (not shown).

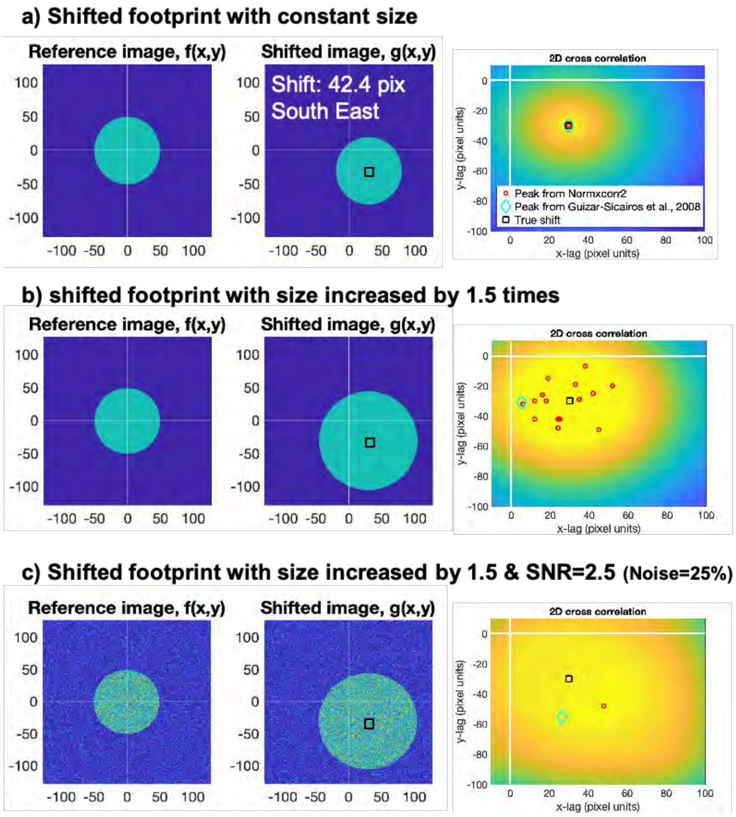



***Figure 4.3.1*** *DCC Synthetic correlation functions experiment. Left column: reference image in three cases a), b) and c) described in the panel titles. Middle column: Modified image (shift+ size variation + noise). Right column: correlation function from normxcorr2 and estimated shift from normxcorr2 (red square) and* GS2008 *(cyan diamond). The True shift is shown as a black square.*

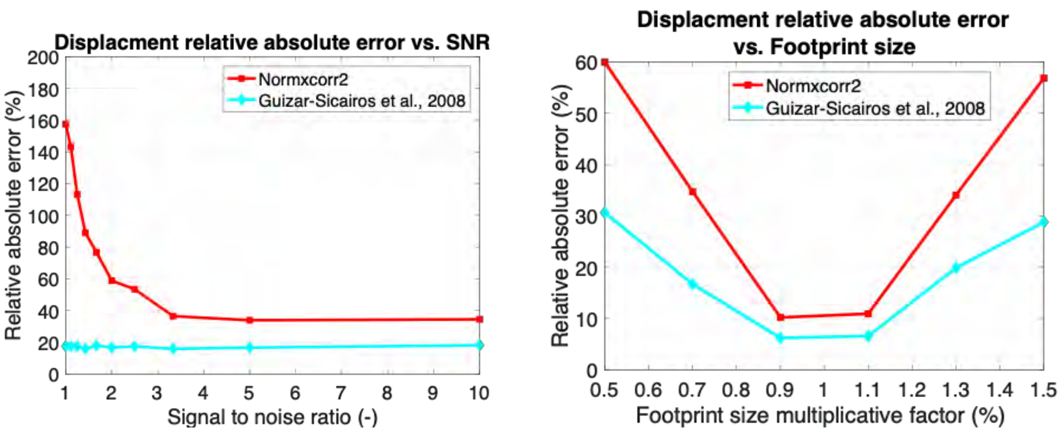



***Figure 4.3.2*** *Relative displacement error estimation as a function of signal to noise ratio and size factor.*

Similarly to what was done in Figure 4.3.1, in Figure 4.3.3 we performed a second test where a WVM-like feature is reproduced. In this case we did not introduce any size variation because the WVM-like feature is intended to reproduce a water vapor front that is moving eastward and is not completely included in the ROI. The perfect displacement estimation is achieved only in the ideal case without noise (cases a and c). Note that in the case c) we considered the gradients of the simulated input images. The quantitative results introducing normally distributed noise variations with standard deviation varying within 10% to 100% of the input signal amplitude are displayed in Figure 4.3.4. In this case, if we consider the gradients of the WVM-like input feature and signal to noise ratio larger than 2, we did not obtain any difference between using normxcorr2 and the GS2008 routine.

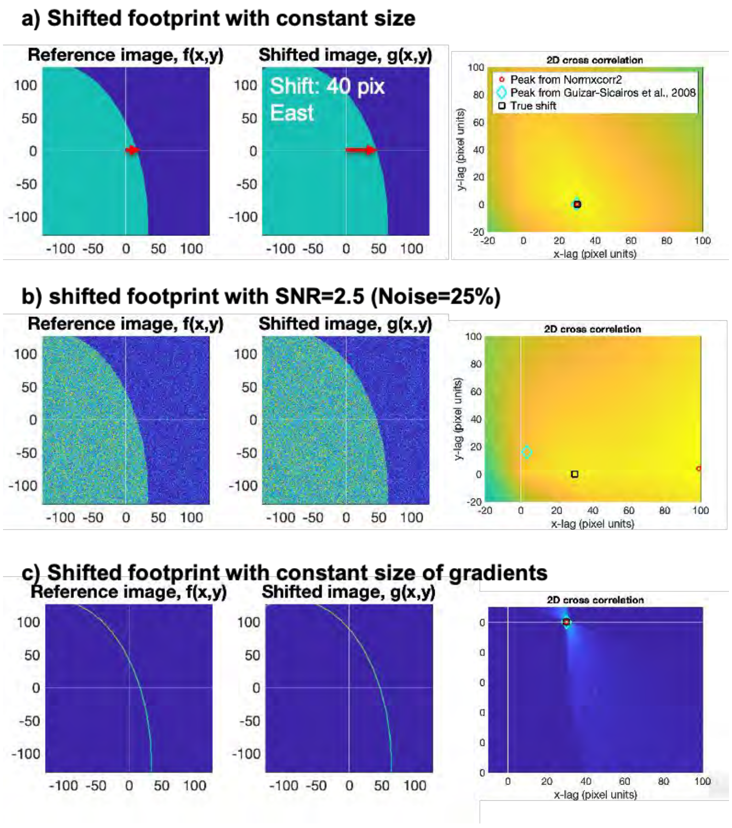



***Figure 4.3.3*** *as in Figure 4.3.1 but for a WVM-like feature.*

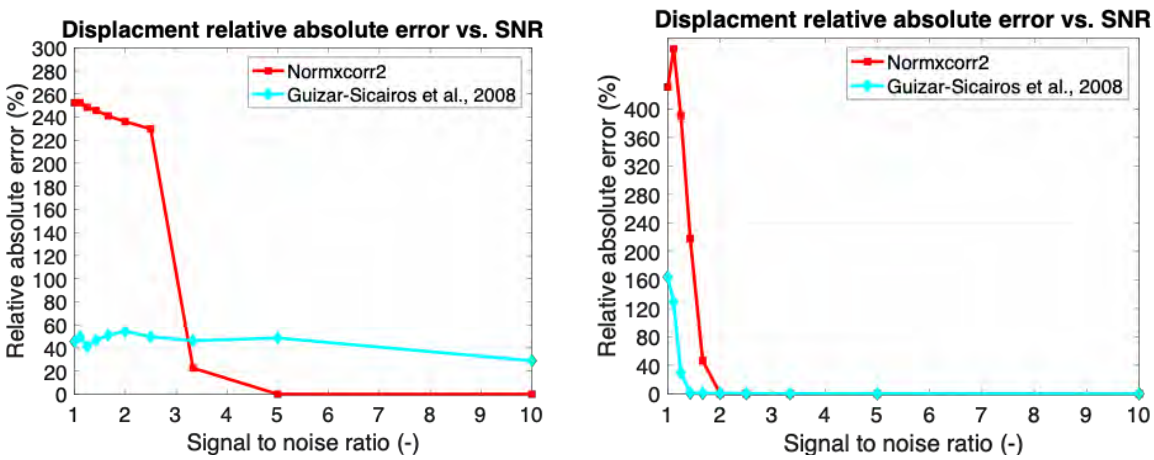



***Figure 4.3.4.*** *Left panel: error as a function of signal to noise ratio for WVM-like features as in Figure 4.3.3. Right panel: case where gradients of WVM-like features are considered.*

# 5. Absolute geolocation methods based on atmospheric targets

## 5.1 Goal

The main goal of this section is to verify the absolute geolocation accuracy that can be obtained using DCC and WVM targets. In order to achieve this goal three algorithms have been developed one for DCC, (described in section 5.2) and two for WVM (in section 5.3-5.4).

This verification has been based on a dataset of SEVIRI and SSMIS coincident observations that spans over 17 days, i.e., the same days of the case studies listed in table 2.1.3 including all the SSMIS F17 orbits and all the SEVIRI time frames for a given day (see section 2.3).

## 5.2 DCC Algorithm

The DCC algorithm is composed of two blocks: a ROI Definition block and a Geolocation Error Estimate Block. The external inputs of the algorithm are:

1) The SSMIS TBs, the 3 channels in the 183.31 GHz band, with SSMIS TBs coordinates and referencing times, and satellite position (latitude, longitude and altitude);

2) SEVIRI TBs from channel 10.8 μm, together with pixel latitude, longitude and time;

3) Temperature and geopotential height profiles (together with latitude, longitude time and pressure levels) from NWP model.

**The ROI Definition Block** takes as inputs the SSMIS and SEVIRI TBs. The first step of the algorithm is finalized to synchronize in space and time the SSMIS and SEVIRI observations. For every SSMIS orbit the portion of the orbit within MSG Full Disk region (defined here as the region with observation angle smaller than 27.4°) is selected. Then, the SSMIS is separated in 15-min long observation frames and corresponding to each MSG time frame. Finally, the MSG image is cut selecting the area with SSMIS observations (TBs). In a second step he ROI definition block applies some preliminary tests in order to verify if the SSMIS-MSG data may include some DCC target: the first test is applied to MW TBs and corresponds to the Hong et al. (2005) test (described in section 2.4 – Eq.1), the second test is applied to thermal IR and is:

$$ {}_{10.8} < 215 \tag{7} $$

This test has been already used by other authors (Bedka et al. 2010) as a preliminary test for the identification of the OTs and should be able to identify 96% of the OTs with relatively large false alarms. In the framework of GAMES, the presence of false alarms is mitigated by the contemporary use of the MW data. The regions where both tests are verified is defined as a new ROI. In order to take into account the geographic shifts due to observation geometry between IR and MW at this stage of the algorithm it is sufficient that the DCC in IR and MW (the pixels that passed both tests) are found within 1 pixel (SSMIS or SEVIRI) from each other. A ROI is a Latitude and Longitude box and all IR and MW data (TBs, coordinates, etc.) inside a ROI are the input of the Geolocation Error Estimate Block. We want to highlight that a ROI can include several storm cells and overshooting tops.

**The Geolocation Error Estimate Block** first calculates the Cloud Top Height (CTH) using the MSG TBs (channel at 10.8 μm) and the temperature profile from NWP model as described in section 3.2.3. Then the IR-CTH are translated in the SSMIS grid and observation geometry with the methodology described in detail in section 3.2.4 (IR-CTH are parallax corrected, projected into the SSMIS line of sight, and convolved with a gaussian antenna pattern). The CTH in the MW and IR grids are used to compensate for parallax effect the IR and MW TBs. The MW TBs are also tested with the Hong et al. (2005) test (see section 3.2.2 – Eq.1) to create the screening mask that selects only the pixels with DCCs. Finally, the parallax corrected TBs (10.8 µm for MSG and 183.31±1 GHz for SSMIS) are the input for the cross-correlation calculation step. This algorithm uses as a metric for geolocation error the Masked Correlation Coefficient Matrix (MCCM) method described in section 4.2. The geolocation error is the spatial shift (δθ,δϕ) corresponding with the maximum correlation coefficient of the correlation coefficient matrix, and expressed in km. We want to stress that for each ROI only one geolocation error is estimated.

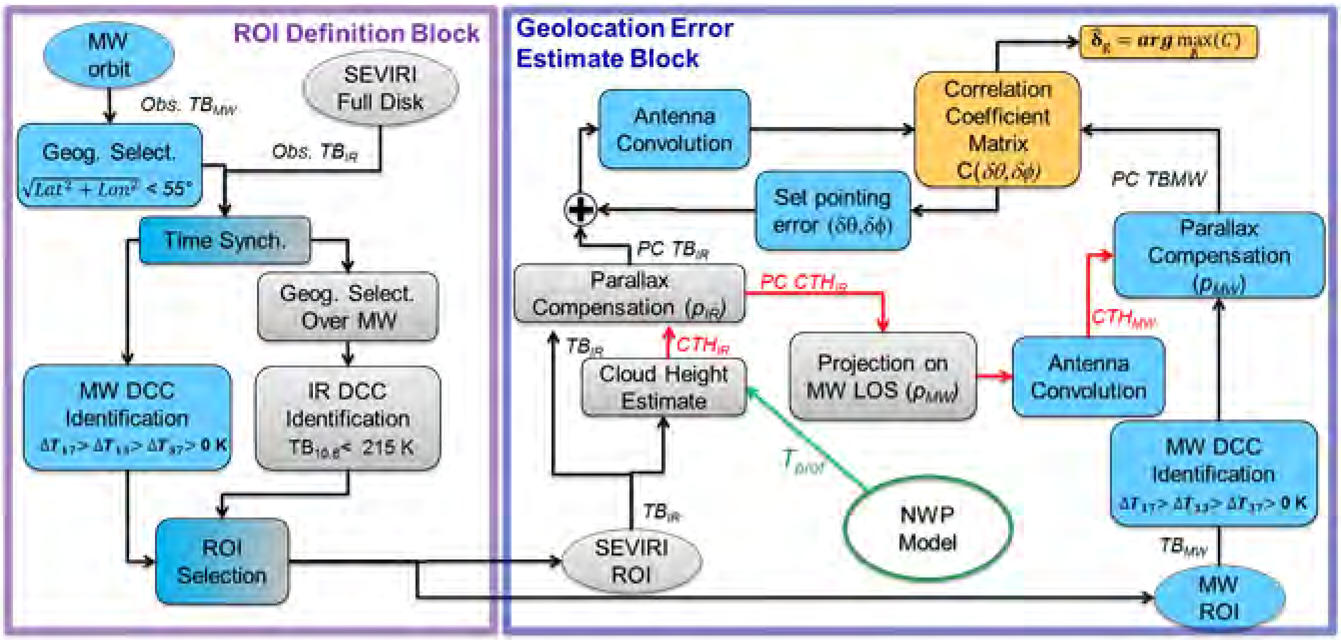



***Figure 5.2.1*** *Scheme of the DCC Algorithm*

### 5.2.1 Results and Discussion

The estimated geolocation errors from the DCC algorithm for the full 17-days dataset have been calculated and some descriptive statistics are reported in this section. However, it is clear that the DCC algorithm needs some additional conditions in order to avoid the scenes where for some reason it does not work properly. A critical analysis of the conditions in which the DCC algorithm fails is given at the end of this section. Therefore, some further conditions have been imposed for selecting the ROI that are usable as geolocation targets. The first set of conditions are consistency checks:

1. Number of MW pixel recognized as DCC in ROI >20
2. Maximum correlation coefficient in the matrix > 0.5
3. Maximum Cloud top height in ROI > 8 km

The first condition is needed to assure that the correlation coefficient is calculated over a sufficient number of pixels. The second assures that some correlation exists between IR and MW TBs. Finally, the third condition verifies that the DCC features in the ROI include at least one OT.
With these conditions the histogram of the estimated geolocation errors is the one shown in Figure 5.2.2, in the top panel (red line). This histogram clearly shows some outliers that can be eliminated by adding a further condition:

4. Geolocation Error < 25 km

With this further constraint the errors histogram becomes the one plotted with a black line in Figure 5.2.2 (top and bottom panel) and the error statistics are reported in Table 5.2.1.

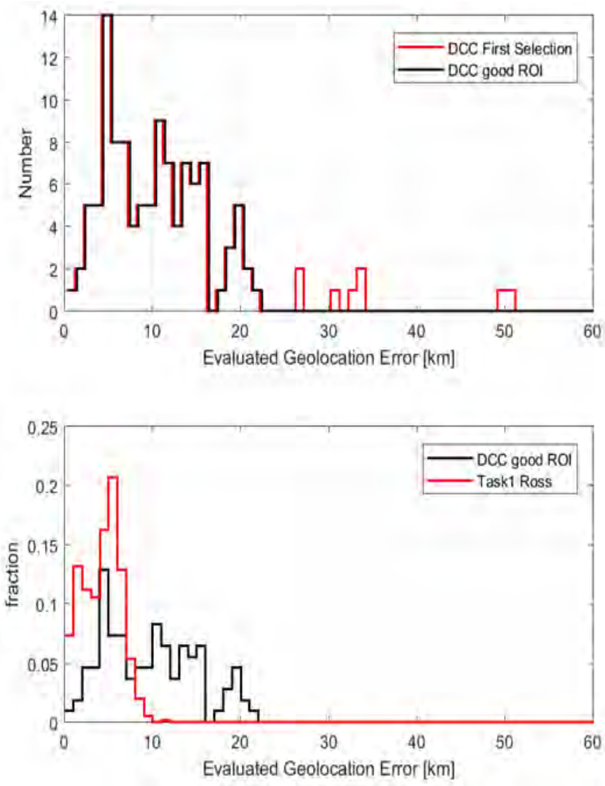



***Figure 5.2.2*** *Histogram of the Geolocation Error from the DCC Algorithm: the top panel shows the error histogram with conditions 1-3 (red line) and including condition 4 (black line). The bottom panel shows the comparison of DCC results (test 1-4, black line) compared with the results from one of the geolocation targets of Task 1 the Ross Ice Shelf (red line).*

***Table 5.2.1****: Error statistics of the DCC Algorithm*

| Days | Orbits in MSG FD area | Num. of SSMIS segments | Num. of ROI | Num. of Good ROI | Mean Distance km | Std Distance km | Mean correlation |
|---|---|---|---|---|---|---|---|
| 17 | 180 | 203 | 170 | 109 | 9,7 | 5,33 | 0.71 |

The statistics of the geolocation error from the DCC algorithm show that the use of DCC as target could increase the number of targets per day that have been identified by Task 1 to 6.4 targets per day. In principle this number can be further increased if the algorithm is applied also to other GEO satellites that are equipped with a thermal IR channel. Unfortunately, the error mean and standard deviation are high and the standard deviation is near to double the GAMES requirements of 2.5 km accuracy. It is worth noting, however, that the mode value of the geolocation errors from the DCC algorithm is very near to the one from Task 1 (Ross Ice shelf). The correlations between IR and MW TBs is not so high, even if some very low correlation cases have been removed from the statistics (consistency check 2). This relatively low correlation is very common for DCCs and is probably related to the different and complex relationship between IR or MW TBs and cloud properties. IR TBs are sensitive to the cloud top temperature and (secondarily) to the mean effective radius in the higher cloud layers. MW TBs at 183.31 GHz , instead, are sensitive to ice scattering that is strongly dependent on ice hydrometeor (mostly snow flakes and aggregates) density, shape and concentration in the cloud at higher levels. In particular, there could be a strong difference in these variables between the cloud OT region and the anvil. If the correlation between IR and MW is usually weak in DCC it becomes nearly negligible in particular conditions: when the updrafts are particularly strong and long lasting and if a plume cirrus cloud is formed over the deep convective cloud. Both these conditions have been met in case 11 where an extreme hail storm hit the Naples city in Italy, producing tennis-ball sized hail. Figure 5.2.3 shows a comparison of the IR and MW TBs for a common DCC case study (case study 01) and for an extreme case study (case study 11). The relation between IR and MW TBs is very different in the two cases due to the presence of a plume cirrus cloud over the deep convective cloud and to the strength and duration of the updraft in the second case (see Panegrossi et al. 2017). Another critical issue of the DCC algorithm is that it strongly relies on the calculation of the CTH from IR TBs, where several assumptions have been made and are not always satisfied: first, we have neglected any effect on the IR thermal channel of the water vapor above the cloud top; moreover outside of the OT region the algorithm assumes a simple relation of equivalence between CTT and environmental temperature that is satisfied only if that portion of the cloud is in thermal equilibrium with the environment; finally the 8 K/km lapse rate assumed for the OTs is a mean value, but it can vary from cloud to cloud. In order to conclude this critical discussion, DCCs cannot actually be used as geolocation targets without a more precise calculation of the CTH.

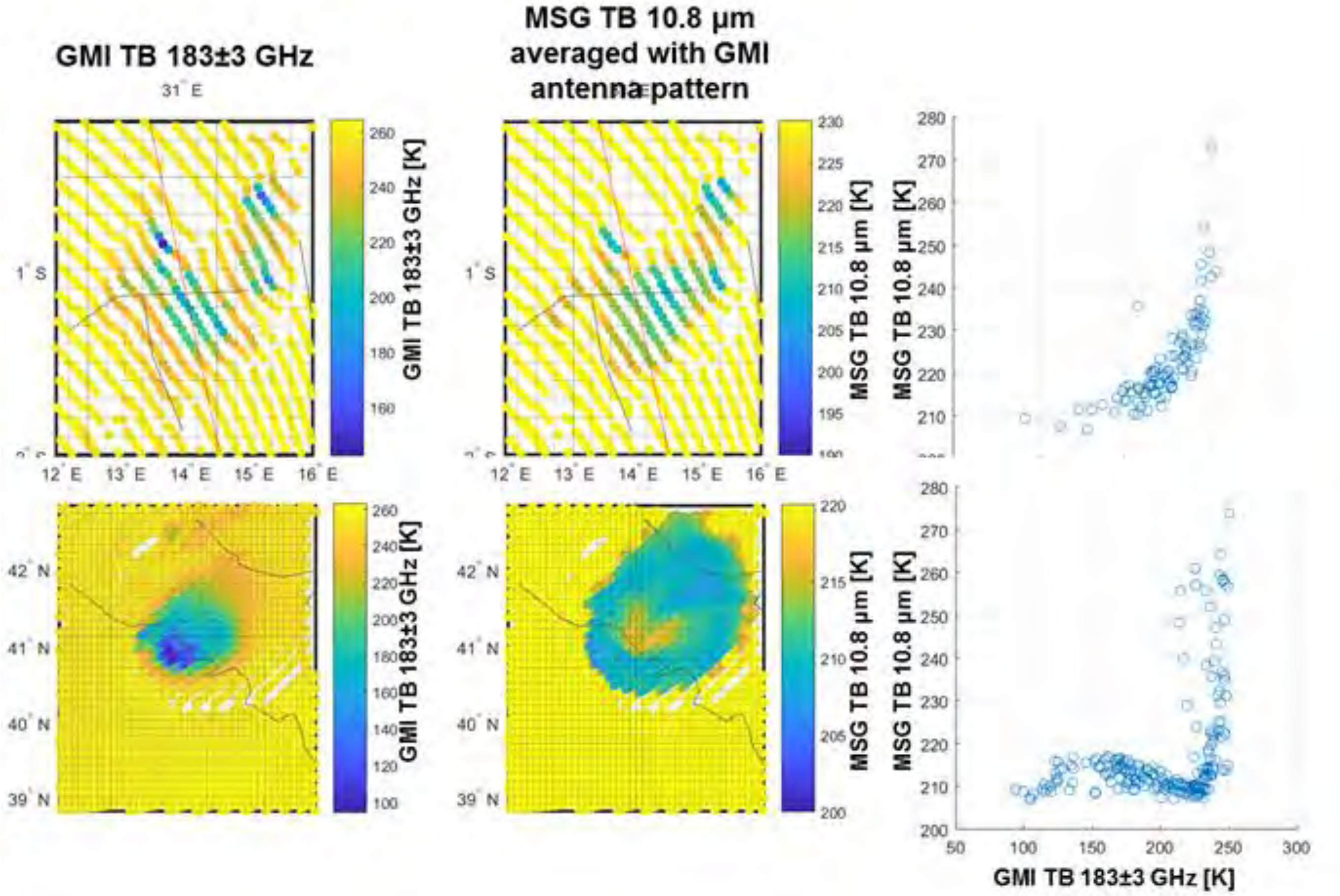



***Figure 5.2.3*** *Comparison of GMI TBs (parallax corrected) and MSG 10.8 µm TBs (parallax corrected and antenna pattern averaged) for case study 01 (23/08/2014) in the top panels and for case study 11 (05/09/2013) bottom panels.*

## 5.3 WVM-MCCM Algorithm

The structure of the WVM-MCCM algorithm is similar to the DCC one. The algorithm is composed of two main blocks, the first is finalized to identify the ROIs, and the second to estimate the geolocation error.

**The ROI Definition Block** is very similar to the same block in the DCC algorithm. The first modules are finalized to synchronize the MSG and SSMIS TBs in time and space. One main difference is related to the input data: the SSMIS TBs are all the channels in the 183.31 GHz band, however the main calculations are performed for the $TB_x$ as defined in section 3.3.1 Eq.6. The MSG inputs are both the thermal channel at 10.8 µm (used for cloud screening) and the 6.2 µm WV channel (used to identify the WVM features). The second step of the ROI Definition Block consists of the cloud screening: a given pixel is identified as "cloud" if it satisfies the conditions:

$$TB_{183.31\pm3} - TB_{183.31\pm1} < 10\ \mathrm{K} \tag{8}$$

$$TB_{10.8} < 260\ \mathrm{K} \tag{9}$$

Where Eq.8 is valid for SSMIS and Eq.9 for MSG.

The final step of the block is the identification of the WVM features, performed by searching for contiguous areas where the absolute value of the horizontal gradient of the MW $TB_x$ is greater than 10 K/km. ROIs are identified in the last step of the block as the regions where there is a contiguous area of WVM feature with a sufficient extension in the MW scene of at least 100 pixels.

**The Geolocation Error Estimate Block** is simpler than the one of the DCC algorithm since all the modules related to the parallax correction are not present. The algorithm takes as input the MSG 6.2 µm channel and the SSMIS derived $TB_x$ in the ROI. The $TB_x$ are tested for cloud masking applying the test defined in Eq.8. Then, the MW and IR TBs are used as input in the cross correlation calculation step that is performed following the MCCM scheme described in section 4.2, the same scheme that has been applied to DCCs. In particular, as for the DCC case, the geolocation error ($\delta_g$) is the shift ($\delta\theta,\delta\phi$) corresponding with the maximum correlation coefficient of the correlation coefficient matrix, and expressed in km. We want to stress that for each ROI only one geolocation error is estimated.

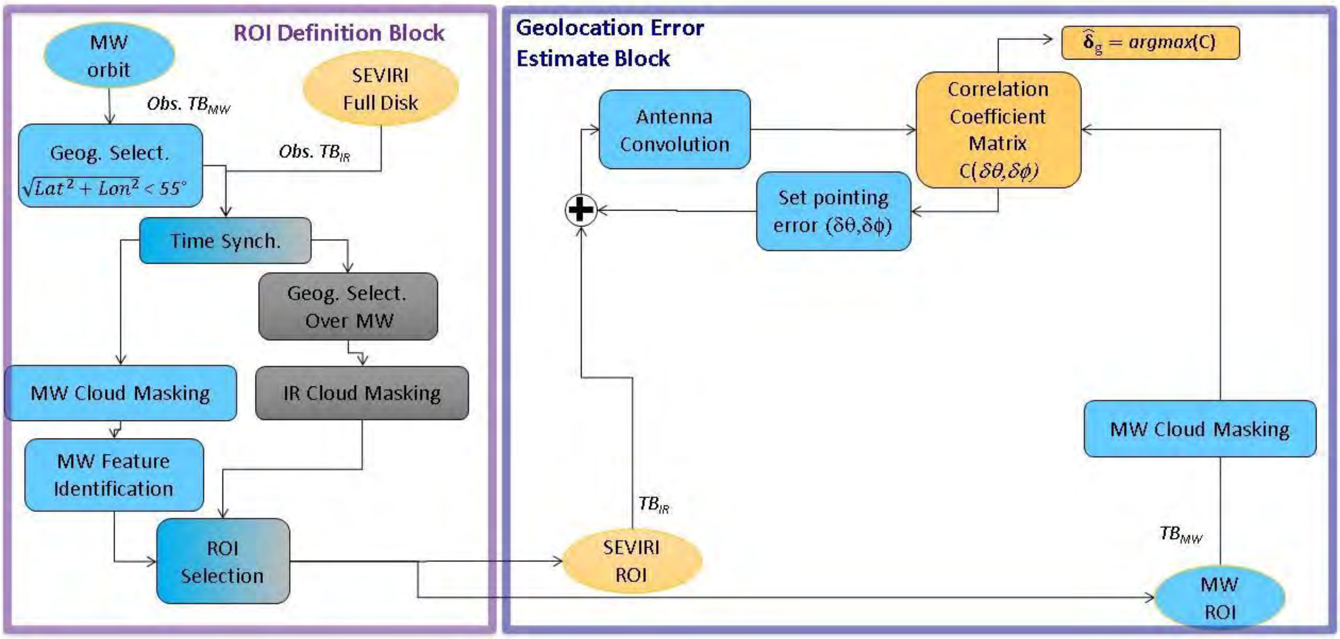



***Figure 5.3.1*** *WVM-MCCM scheme*

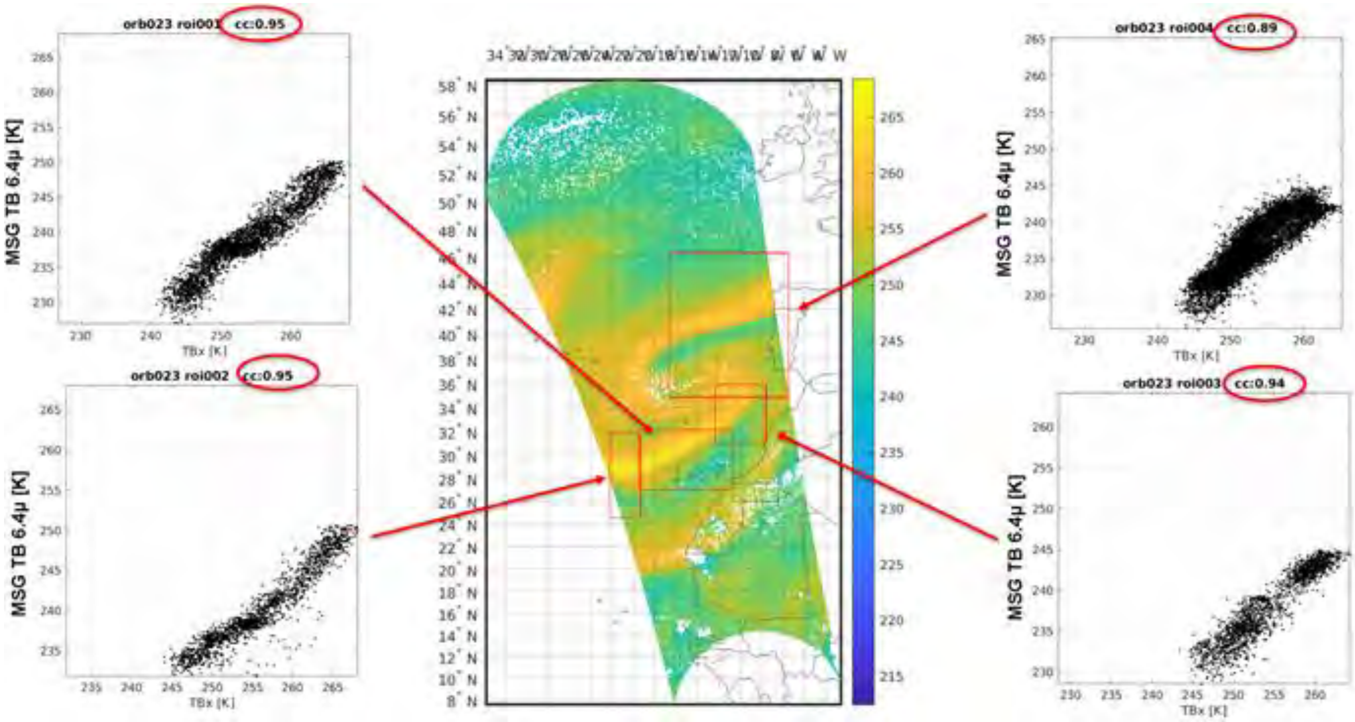



***Figure 5.3.1*** *Example of application of the ROI Detection Block for WVM-MCCM (central panel) with scatter plots for each ROI showing the correlation between IR TB and MW $TB_x$.*

### 5.3.1 Results and Discussion

The application of the WVM-MCCM algorithm to the 17-days SSMIS-MSG dataset revealed that SSMIS TBx and SEVIRI 6.2 μm channel TBs aare strongly correlated in proximity of the WVM features. Figure 5.3.1 shows an example of the application of the ROI Detection Block for one SSMIS orbit segment. After the antenna pattern convolution, the IR-TBs and the MW-TBx show very strong correlation coefficients (often higher than 0.9). A comparison of the correlation coefficients (Figure 5.3.2) obtained from DCC and from WVM-MCCM algorithms show that the second has higher correlation coefficients.

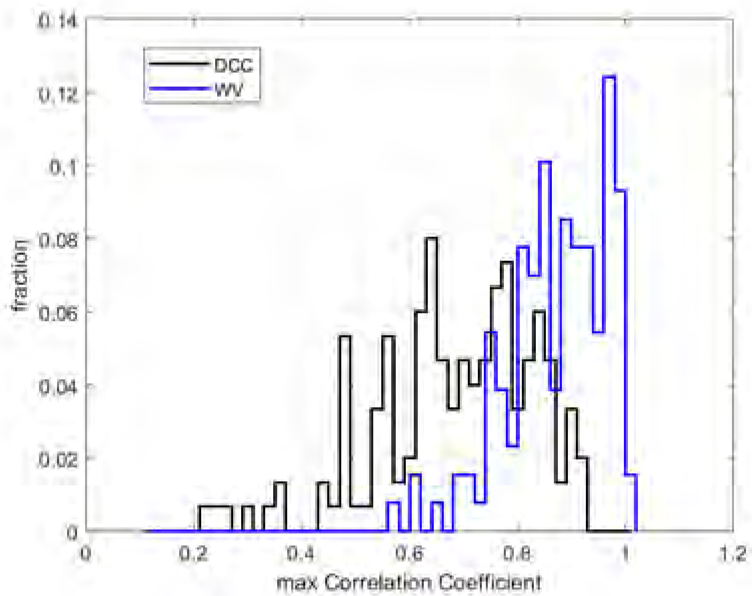



***Figure 5.3.2*** *Histogram of the correlation coefficients from the WVM-MCCM and DCC algorithms.*

However, the WVM-MCC algorithms suffer from the issue of multiple solutions. This issue can be well explained looking at the example in Figure 5.3.3. In this example the shape of the WVM feature is clearly visible in both MW and IR as a sharp variation from relatively cold TBs (around 255K for TBx and 235 K for IR) to relatively warm TBs (around 270 K and 245 K respectively). The shape of the feature, however, is very simple almost following a line, which makes the correlation coefficient matrix almost insensitive to translations along the feature main direction. This can be observed in the same figure looking at the correlation coefficient matrix, that shows very high values of the correlation coefficient along a diagonal direction. In this condition the position of the maximum value of the Correlation Coefficient Matrix is not a good metric for the geolocation error estimate: many positions correspond to the maximum value of the correlation coefficient. Therefore the MCCM-WVM algorithm has been modified in the computation of the geolocation error: instead of using the position of the maximum correlation, the 90$^{th}$ percentile region of the correlation coefficient is defined (black line contour in the upper-right panel of Figure 5.3.3), and the position in this area nearest to the origin of the axis is selected, corresponding to geolocation error equal to 0 km. The applied metric does not represent a solution to the issue of multiple solutions, and it should be considered as an attempt (substantially failed) to control this issue.

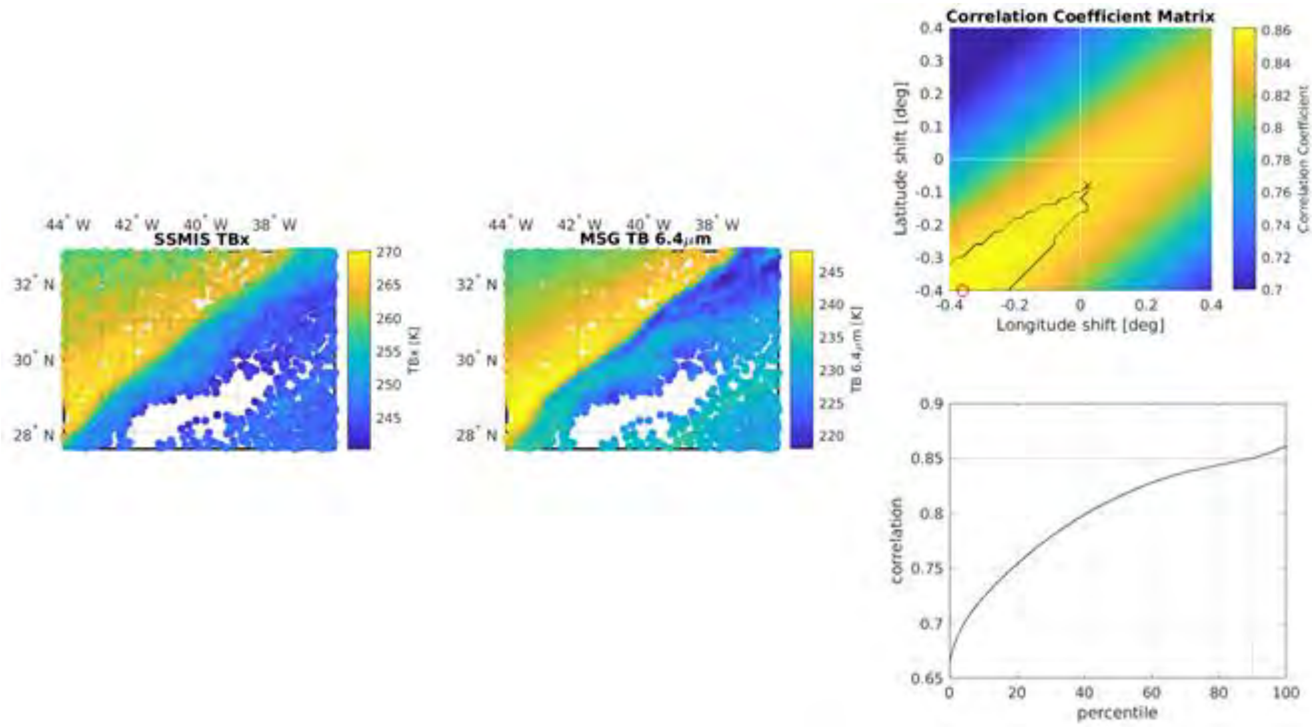
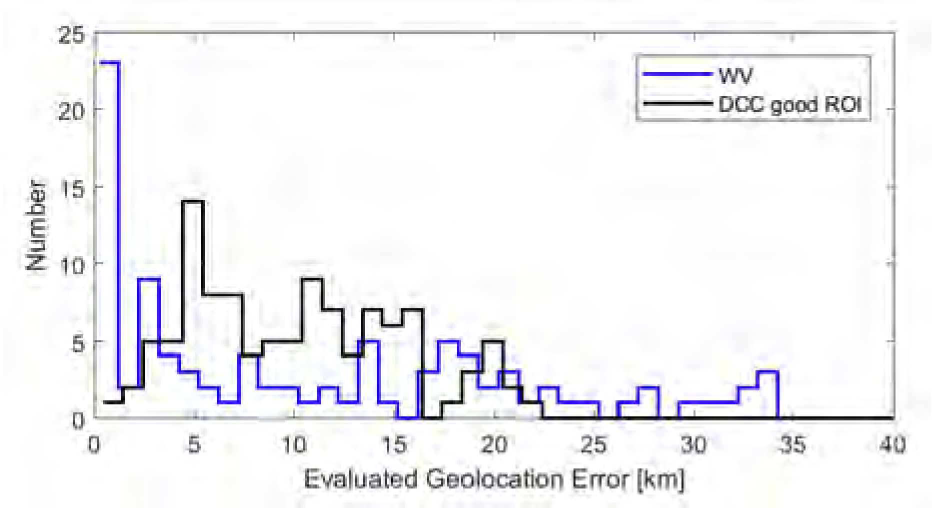



***Figure 5.3.3*** *Example of application of the WVM-MCCM algorithm to a ROI, left and central panels show the MW-TB$_x$ and the IR-TB at 6.2 µm. Upper right panel shows the resulting correlation coefficient matrix, while the bottom-right panel shows the correlation in the correlation coefficient matrix as a function of the percentile.*

Figure 5.3.4 shows the histogram of the geolocation errors obtained by applying the WVM-MCCM to the 17-days SSMIS-MSG dataset; the figure shows also the corresponding histogram from the DCC algorithm for comparison. The WVM-MCCM algorithm shows a geolocation error distribution that has its maximum around 0 km with a large spread between 1 and 35 km without a clear distribution. In this case it is therefore worthless to set a threshold value for determining outliers. Some descriptive statistical quantities of the error from WVM-MCCM are reported in Table 5.3.1. It appears clear that the main issue of the WVM-MCCM algorithm is related to the multiple solutions (i.e., the insensitivity to horizontal shifts along the feature main direction). Summarizing, the WVM-MCCM algorithm results show a very high mean error and standard deviation (~4 times the GAMES accuracy requirements) and very high maximum correlation coefficients.



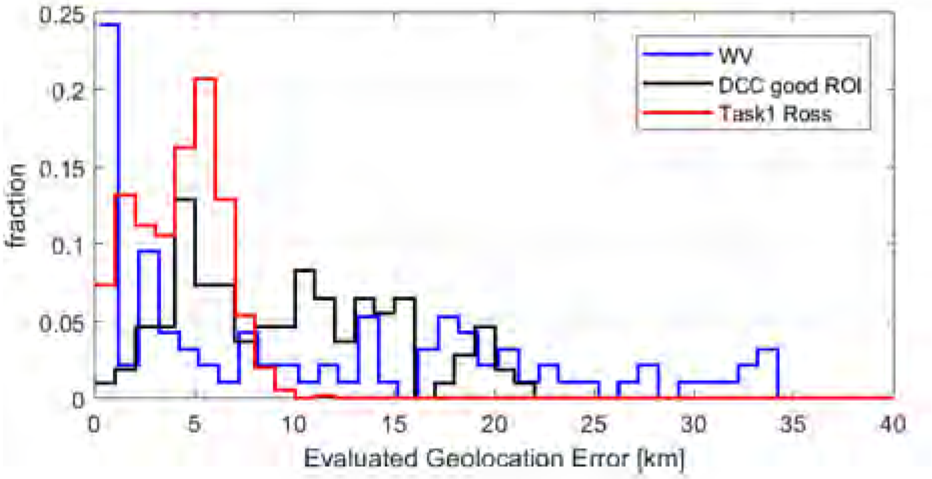




***Figure 5.3.4*** *Histogram of geolocation errors from the WVM-MCCM algorithm, compared to the ones from DCC algorithm and from Task1 Ross Ice Shelf.*

**Table 5.3.1**: Error statistics of the WVM-MCCM Algorithm

| Days | Orbits in MSG FD area | N SSMIS segments | N ROI | N Good ROI | Mean Distance km | Std Distance km | Mean correlation |
|---|---|---|---|---|---|---|---|
| 17 | 180 | 203 | 129 | 95 | 10.77 | 10.39 | 0.90 |

## 5.4 WVM-FCC Algorithm

The WVM-FCC algorithm scheme is shown in figure 5.4.1. The ROI Detection Block has not been modified from the WVM-MCCM scheme. The Geolocation Error Estimate Block, however, is very different. Both IR TB (6.2 µm) and MW $TB_x$ are processed by a series of modules: first the horizontal TB gradient module is calculated, then two different tests are applied to IR and MW TBs in order to identify the clouds and compute the TB gradients, pixel identified as cloudy are set to the default value of 0 K/km. Then, both IR and MW TBs are regridded to a common equally spaced grid (1 km spacing), using cubic interpolation. Finally, the cross correlation between the IR and MW gradients is calculated. In order to test the impact on the estimated geolocation errors, the cross correlation has been calculated using two different functions:

1. normxcorr2 MATLAB function.
2. TheGuizar-Sicairos et al. (2008) (GS2008) approach already described in section (4.3)

The estimated error is the distance (in km) between the cross correlation matrix origin (0,0) and the position of the maximum of the cross correlation matrix.

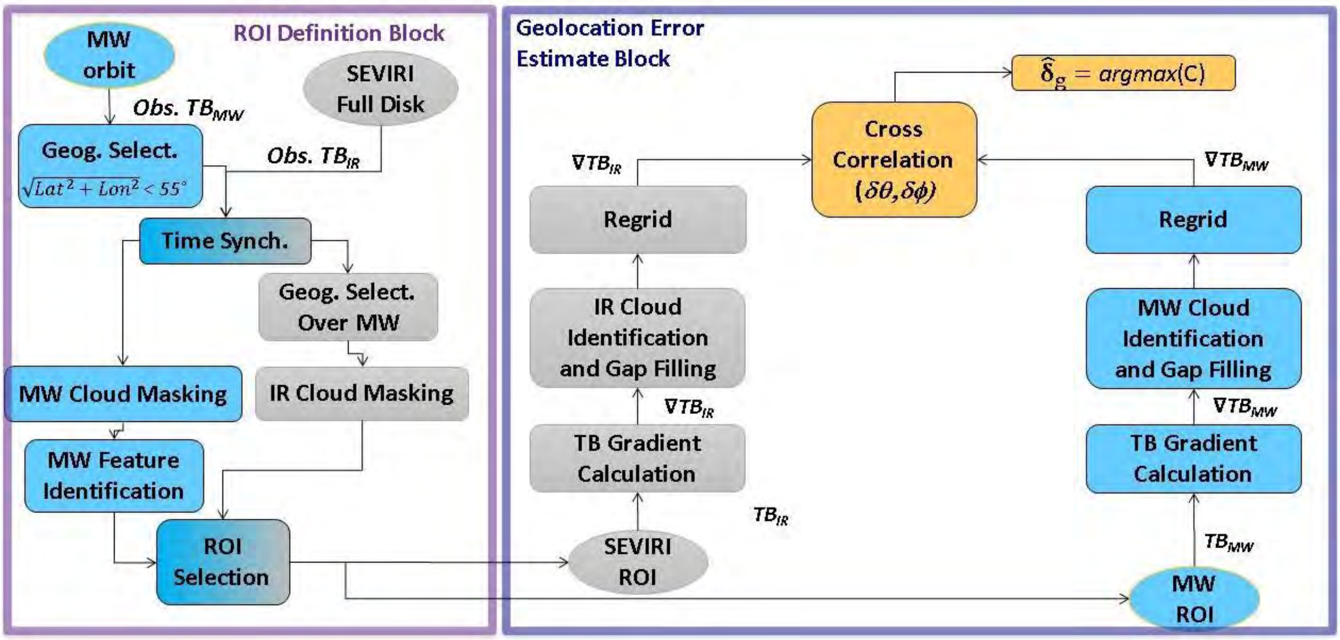



***Figure 5.4.1*** *scheme of the WVM-FCC algorithm*

### 5.4.1 Results and Discussion

Applying the WVM-FCC to the 17-days SSMIS-MSG dataset the obtained maximum correlation coefficient between IR and MW are significantly lower than for the WVM-MCCM (see Figure 5.4.2). This is due to two main reasons: 1) the WVM-FCC algorithm compares gradients of TBs instead of TBs in the WMV-MCCM, and 2) the correlation coefficients are calculated in a high spatial resolution and regularly spaced grid in the WVM-FCC algorithm, while the MW spacing grid used in the WVM-MCCM algorithm has a lower spatial resolution (~ 16 km).

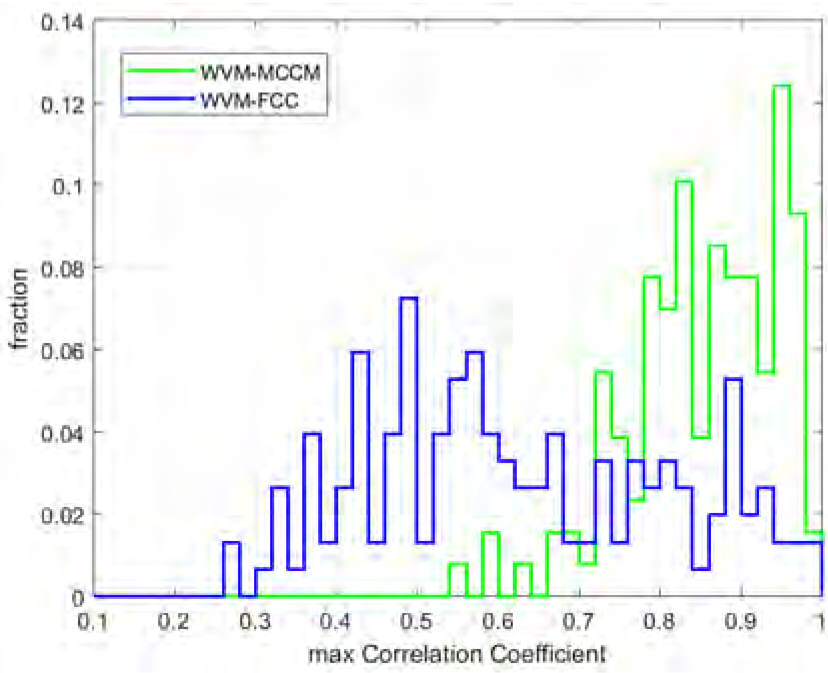



***Figure 5.4.2*** *Comparison of the maximum correlation coefficients obtained in the MCCM and in the FCC algorithms*

The geolocation errors resulting from the WVM-FCC method shown in Figure 5.4.3 and in Table 5.4.1, are smaller than for WVM-MCCM method (Fig. 5.3.4, and Table 5.3.1). Figure 5.4.3 shows the histogram of the geolocation error obtained from the WVM-FCC algorithm applying two different functions for the calculation of the cross correlation (normxcorr2 – blue line and dftregisration by GS2008 – black line). These are also compared with Task1 Ross Ice Shelf target statistics. These results have been obtained by imposing some further consistency checks:

1. Cloud Cover in the ROI less than 50% of the domain

2. Maximum correlation coefficient higher than 0.5

Moreover, in order to exclude from the statistics the remaining outliers clearly visible in Figure 5.4.3 a further condition has been imposed:

3. Estimated geolocation error lower than 15 km

Imposing these conditions, we obtain the error statistics shown in Table 5.4.1. The number of usable targets per day is estimated to about 2, while the mean distance and standard deviations are very small compared to WVM-MCCM and DCC algorithm results. The standard deviation in particular (3.6 km) is close (but still higher) than the required accuracy of GAMES (2.5 km). The correlation coefficients, as already discussed in the previous paragraph, are lower than in the WVM-MCCM case. Last row of table 5.4.1 shows the results that have been obtained applying the WVM-FCC algorithm to the 3-months SSMIS-MSG dataset described in section 2.4. This test has been performed in order to improve the robustness of the results with a more extended dataset. The main results are confirmed: both the standard deviation and mean error are very small (compared to DCC and WVM-MCCM algorithms) but still higher than the GAMES requirements. However, the number of targets needs to be highlighted: the use of a dataset based on selected case studies results in an overestimate of the number of usable targets per day, that in the 3-months dataset it is reduced to around 1 per day.

From the results of the WVM-FCC algorithm, some concluding remarks can be drawn. The use of TB gradients instead of TBs in the WVM features identification has a strong impact on the magnitude of the accuracy of the method, substantially overcoming the multiple solution issue, due to the fact that the WVM features identified in the gradient space show sharper and better-defined shapes. However, the use of gradients of TBs implies a more conservative selection of the ROIs, resulting in ~ 1 usable targets per day in the WVM-FCC method versus ~5.6 targets per day of WVM-MCCM and ~6.4 targets per day of DCC. Moreover, the use of TB gradients implies a generally lower correlation coefficient between IR and MW. Finally, the choice of the function used to calculate the cross correlation has a minor impact on the error statistics.

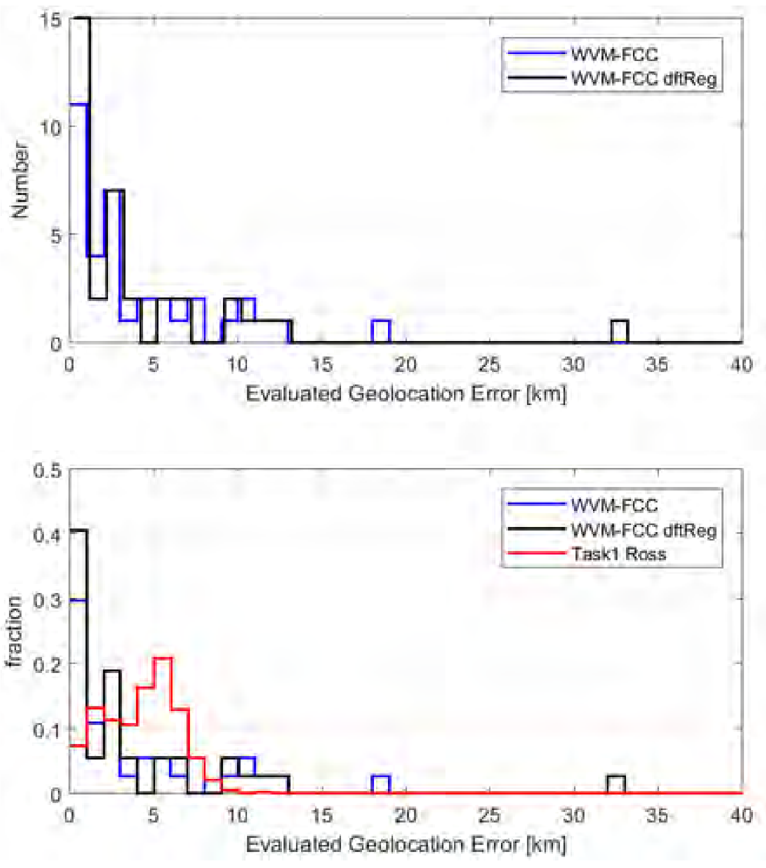



***Figure 5.4.3*** *Error distribution with the WVM-FCC algorithm*

***Table 5.4.1**: Error statistics of the WVM-FCC Algorithm*

| | Days | Orbits in MSG FD area | N SSMIS segments | N ROI | N Good ROI | Mean Distance km | Std Distance km | Mean correlation |
|---|---|---|---|---|---|---|---|---|
| Normxcorr2 | 17 | 180 | 203 | 152 | 35 | 3.30 | 3.72 | 0.63 |
| DFTregistration by Guizar-Sicairos 2008 | 17 | 180 | 203 | 152 | 35 | 3.13 | 3.69 | 0.63 |
| Normxcorr2 | 90 | 969 | 2871 | 591 | 95 | 3.64 | 3.60 | 0.66 |

# 6. Relative geolocation based on atmospheric targets

## 6.1 Goal

This subtask aims at exploring the relative geolocation issue of ICI channels. Higher frequency ICI channels (i.e. from ICI-5 to ICI-11 having frequency equal or greater than 325.15 GHz) will be likely not able to observe any ground reference due to the strong gas absorption at these frequencies and consequently any classical geolocation strategy making use of ground reference cannot be applied on these channels. To verify the geolocation of test channels from ICI-5 to ICI-11 a possible way out strategy could be to assume ICI-1 or ICI-4 as reference channel assuming one of them, already geolocated using some ground reference technique previously applied as in AD1. Thus, the relative geolocation of the test channels can be verified comparing each single test channel with the selected geolocated reference channel. Of course, any geolocation error in the reference channel will propagate through the test channels analysed, quantitatively affecting the final result. The comparison we want to verify in this sub task is in terms of antenna temperatures, $T_A$, of some atmospheric features previously detected.

However, the sensitivity of $T_A$ to the selected atmospheric feature can vary as a function of the channel considered as well as of the sensor's viewing geometry. Consequently, the comparison between the test and the reference channels could be affected by variations of $T_A$ caused by a different sensitivity to atmospheric vertical layers that could mask those $T_A$ variations that are specifically attributed to the geolocation pointing errors that we want to estimate.

The ultimate goal of this sub task is to provide quantitative guidance to determine to what extent (and for which ICI test channels and atmospheric targets) a relative pointing error retrieval can be efficiently implemented.

## 6.2 Input simulations

To fulfill the subtask goal we considered a simulated dataset of $T_A$ for some ICI channels already generated in a previous study (AD1 AD2). ICI simulations are provided by MolFlow and they take in input ECMWF, ERA5 atmospheric scenarios with a horizontal resolution (0.25°, i.e. about 30 km), vertical resolution variable up to an altitude of 80 km, and time sampling of 1 hour. The date considered includes four reference Metop-A orbits:

- orbit 4655 and 4656: from 08:00 to 13:00 UTC from ERA5 forecast@2007-09-12T06:00:00
- orbit 6985: from 08:00 to 11:00 UTC from ERA5 forecast @2008-02-23T06:00:00
- orbit 9744: from 13:00 to 16:00 UTC from ERA5 forecast @2008-09-04T06:00:00

Core calculations of $T_A$ are performed by ARTS v2.3.x (Buehler et al., 2018) taking into account the emissivities of open water according to TESSEM2 (Tool to Estimate Sea-Surface Emissivity from Microwaves to sub-Millimeter waves, (Prigent et al., 2017)), the emissivity of land according to a modified TELSEM2, (a Tool to Estimate Land Surface Emissivities at Microwave frequencies, (Aires et al, 2011)), absorption and emission of water vapour are considered according to (Rosenkranz et al., 1999) whereas particle single scattering data are taken from the database presented by (Eriksson et al., 2018). A **Backus Gilbert** (BG) **interpolation method** (Stogryn et al.,1976) has also been applied to some ICI channels to homogenise their Field of Views (FOVs). The BG is mainly needed because the exact projection of the FOVs differs among the various ICI channels even in absence of a geolocation error. This is caused by some pointing offsets, ($\Delta\theta_i$, $\Delta\varphi_i$) for *i*-th channel in the elevation and azimuth direction, respectively, that are added to the main tilt direction, ($\theta$, $\varphi$). Because of the MWI/ICI conical scan geometry, the azimuth angle, $\varphi$, is variable in time along the cross-track scan, whereas the elevation angle ($\theta$), defined with respect to nadir, is constant ($\theta$=44.767°). For completeness, Table 6.2.1 summarises these offsets ($\Delta\theta_i$ ,$\Delta\varphi_i$) for ICI whereas Figure 6.2.1 shows the FOV nominal positions (i.e. positions in case of error-free perfect geolocation) for some ICI channels. Table 6.2.1, in the last two columns also lists the true unknown pointing errors ($\delta\theta_0$ ,$\delta\varphi_0$)) that we artificially introduced in our simulations to check the pointing accuracy of some geolocation retrieval schemes, as it will be discussed later on. For a selected channel, the FOVs position separation is 2.7 km across track and 9 km (center scan) or 5 km (outer part of the scan) along track (Figure 6.2.1c).

Since the ICI radiometer integration time for each individual sample is about 0.661045 ms and it is shorter than the time period necessary to sweep out a single projected FOV across the scan, we have several footprints overlapping for a given position and for each channel. This allows for a footprint matching procedure using the Backus-Gilbert methodology by which we can produce a remapping of the original data for a test channel as it was observed in the viewing geometry of another reference channel. In short, the Backus-Gilbert methodology linearly interpolates $T_A$ from one test channel into the geometrical grid of another reference channel, thus compensating for the FOV spatial mismatch displayed in Figure 6.2.1b. Some pre-calculated static weights allow the above-mentioned interpolation. The weights are found, after a trade-off analysis which include a minimisation of a penalty function that considers both the effective noise of the remapped data and the fit to the target footprint. It should be noted that the optimal set of weights are channel-dependent as well as they depend on the scan position. Note that, in the outer part of the scan, a proper remapping is impossible to perform due to differences in swath width between channels.

***Table 6.2.1****: Summary of ICI channels, their characteristic and relative instantaneous pointing offset.*

| Ch. id | Frequency (GHz) | Bandwidth (Mhz) | Nedt (K) | Bias (K) | Footprint size -3dB (km) | Elevation Offset $\Delta\theta_i$ (°) | Azimuth Offset $\Delta\varphi_i$ (°) | Elevation Pointing error $\delta\theta_0$(°) | Azimuth Ponting error $\delta\varphi_0$ (°) |
|---|---|---|---|---|---|---|---|---|---|
| ICI-1 | 183.31±7.0 | 2x2000 MHz | 0.8 | 1.0 | 16 km | -0.7801282 | 0.000000 | 0 | 0 |
| ICI-2 | 183.31±3.4 | 2x1500 MHz | 0.8 | 1.0 | 16 km | -0.7801282 | 0.000000 | 0 | 0 |
| ICI-3 | 183.31±2.0 | 2x1500 MHz | 0.8 | 1.0 | 16 km | -0.7801282 | 0.000000 | 0 | 0 |
| ICI-4V | 243.20±2.5 | 2x3000 MHz | 0.7 | 1.5 | 16 km | 0.71056695 | -3.397678152 | 0.07 | 0.13 |
| ICI-4H | 243.20±2.5 | 2x3000 MHz | 0.7 | 1.5 | 16 km | 0.7308017 | 3.384629254 | 0.07 | 0.13 |
| ICI-5 | 325.15±9.5 | 2x3000 MHz | 1.2 | 1.5 | 16 km | -0.82190055 | -2.226341035 | 0.07 | 0.13 |
| ICI-6 | 325.15±3.5 | 2x2400 MHz | 1.2 | 1.5 | 16 km | -0.82190055 | -2.226341035 | 0.07 | 0.13 |
| ICI-7 | 325.15±1.5 | 2x1600 MHz | 1.5 | 1.5 | 16 km | -0.82190055 | -2.226341035 | 0.07 | 0.13 |
| ICI-8 | 448.00±7.2 | 2x3000 MHz | 1.4 | 1.5 | 16 km | -0.8221742 | 2.240223316 | 0.07 | 0.13 |
| ICI-9 | 448.00±3.0 | 2x2000 MHz | 1.6 | 1.5 | 16 km | -0.8221742 | 2.240223316 | 0.07 | 0.13 |
| ICI-10 | 448.00±1.4 | 2x1200 MHz | 2.0 | 1.5 | 16 km | -0.8221742 | 2.240223316 | 0.07 | 0.13 |
| ICI-11V | 664.00±4.2 | 2x5000 MHz | 1.6 | 1.5 | 16 km | 0.7522477 | -1.367038422 | 0.07 | 0.13 |
| ICI-11H | 664.00±4.2 | 2x5000 MHz | 1.6 | 1.5 | 16 km | 0.8755013 | 0.94089788 | 0.07 | 0.13 |

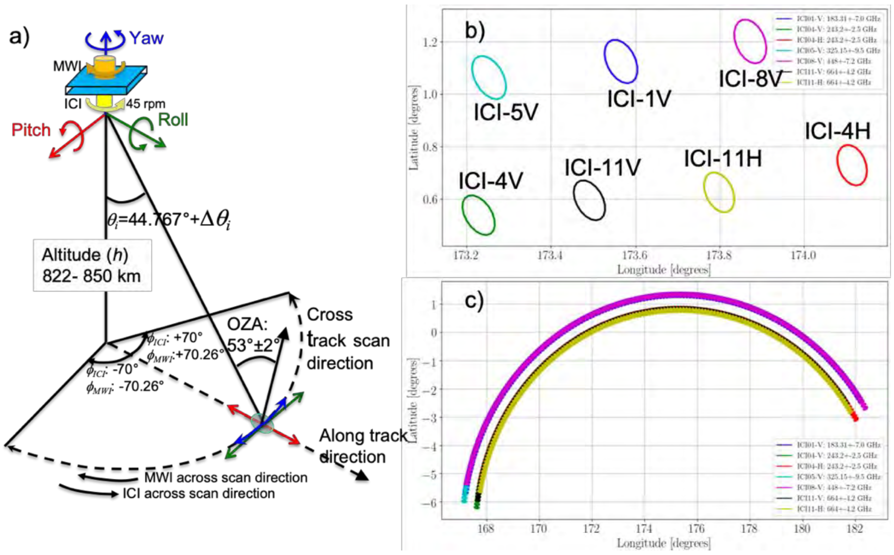



*Figure 6.2.1: Panel a): Viewing geometry for ICI and MWI. Panel b): Instantaneous, relative positions of -3 dB footprints on the geoid for some ICI channels. Panel c): relative positions of -3 dB footprints on geoid for some ICI channels and for every 5th cross track sample of a complete scan.*

## 6.3 Selected atmospheric targets: Atmospheric rivers

Some case studies which include Atmospheric Rivers (ARs) have been selected from ERA5 forecast as indicated by the black boxes in **Figure 6.3.1**. Atmospheric rivers (ARs) are defined as narrow, long and transient corridors of strong horizontal water vapor transport that is typically associated with a low-level jet (Ralph et al., 2016).

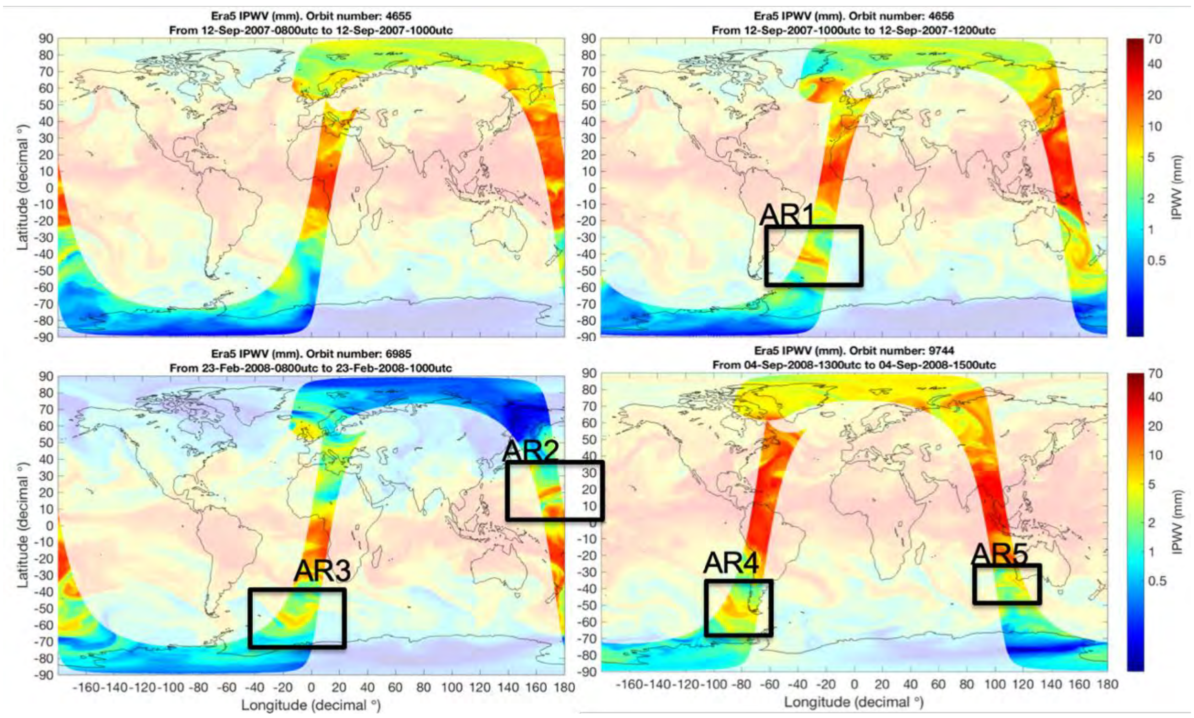



*Figure 6.3.1: Integrated Precipitable Water Vapor (mm) from ERA5 forecasts as in the title of each panel for some selected Region of Interest (ROI) for five identified atmospheric rivers (ARs).*

Four examples of ARs are highlighted in more detail in Figure 6.3.2 in terms of $T_A$ for several MWI and ICI channels as specified in each figure panel. For each AR case, the vertical Integrated Precipitable Water Vapor (IPWV) in (mm), the Cloud Liquid and Ice equivalent Water Paths (mkg/kg), labeled as LWP and IWP, respectively, are also shown in the first column of panels. As expected, the water vapor signature of

AR is well visible in the MWI-2V at 23.8 GHz channel that shows a good visual correlation with IPWV pattern. Contrarily, the IPWV pattern is partially detected in the other represented channels namely ICI-1 (183±7 GHz), ICI-4V (243±2.5 GHz), ICI-5 (325.15 ±1.5 GHz), ICI-8 (448.00V±7.2 GHz), and ICI-11V (664.00V±4.2). In the ERA5 dataset, as evidenced in Figure 6.3.1, we found five signatures that are likely associated with ARs. Unfortunately, when looking at these signatures in terms of simulated MWI/ICI $T_A$, we did not find any relevant signatures associated with ARs for channels above or equal to 183.31 GHz, as it can be deduced by analysing Figure 6.3.2. In practice, in the analysed cases, the IPWV patterns are not recognisable in the ICI channels. This can be due to the masking effects caused by coexisting higher level clouds and by a poor sensitivity of channels having weighting functions picking above the AR top that usually extend no more than 4 km a.sl.. In our AR case studies, we verified that ICI channels $T_A$, match pretty well with coexisting IWP patterns, when they are present. This suggests that ARs could be hardly exploitable for higher channel ICI/MWI geolocation purposes, unless they are associated with the presence of coexisting ice aloft that causes scattering effects in the ICI channel $T_A$. However, the latter condition is similar to what expected for convective cloud cases which are analyzed in more detail in the next section.

## 6.4 Selected atmospheric targets: Deep convective Clouds

Eight case studies of Deep Convective Clouds (DCC), have been selected from the ERA5 forecast as indicated by the black boxes in Figure 6.4.1. The selection was done by identifying regions with higher Ice Water Path. Figure 6.4.2 shows each selected region in terms of simulated $T_A$ for several ICI channels as indicated in each figure's panel. As expected, it is evident as the DCC $T_A$ depression signature is quite detectable by all selected ICI channels, although the pattern of the $T_A$ can considerably vary when moving from ICI-1 to ICI-11.

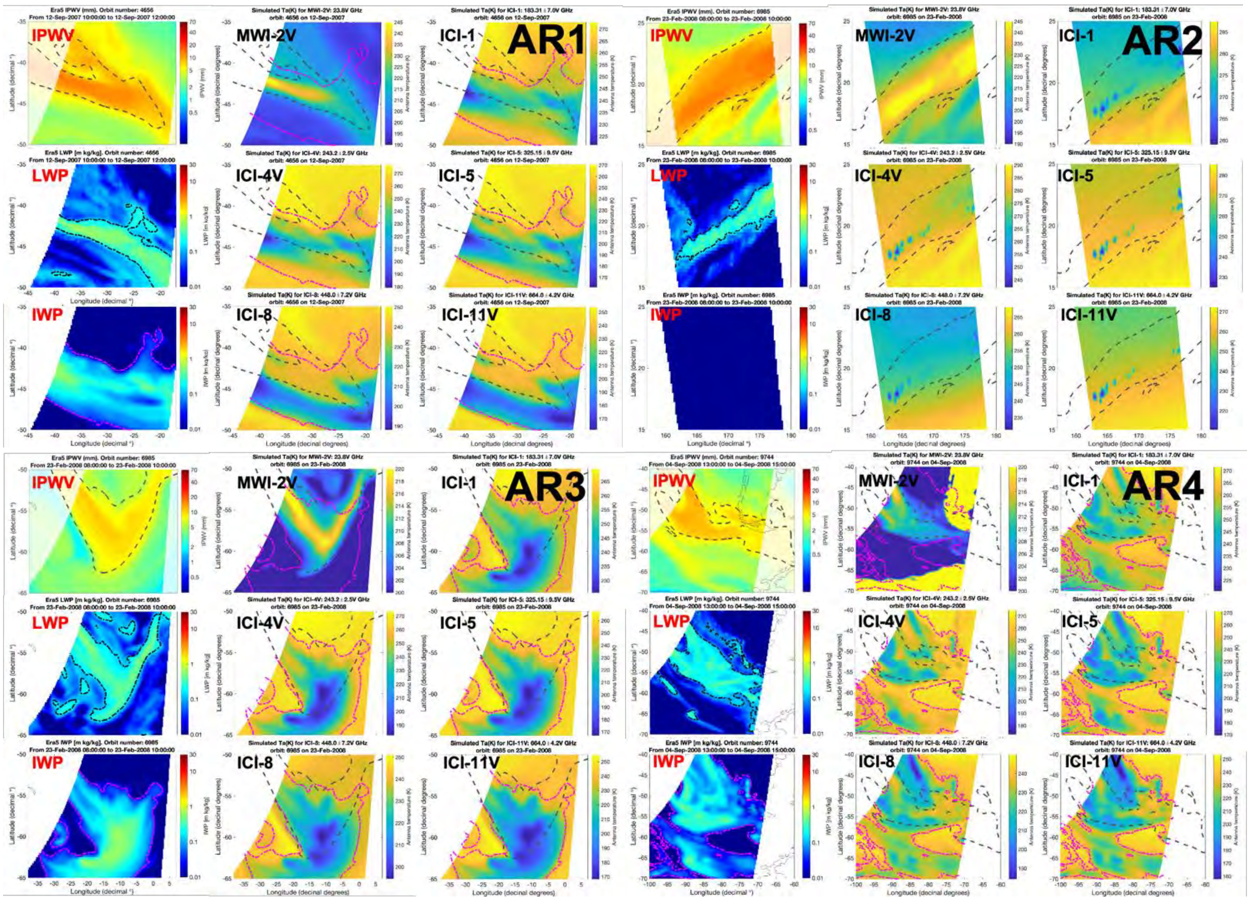



***Figure 6.3.2**: Simulated $T_A$ (K) Atmospheric River (AR) signatures for some MWI / ICI channels as in the title of each panel. The first column panels from the top show the vertical Integrated Water Vapor (mm), the cloud Liquid Water Path and the Ice equivalent Water Path in (mkg/kg), respectively.*

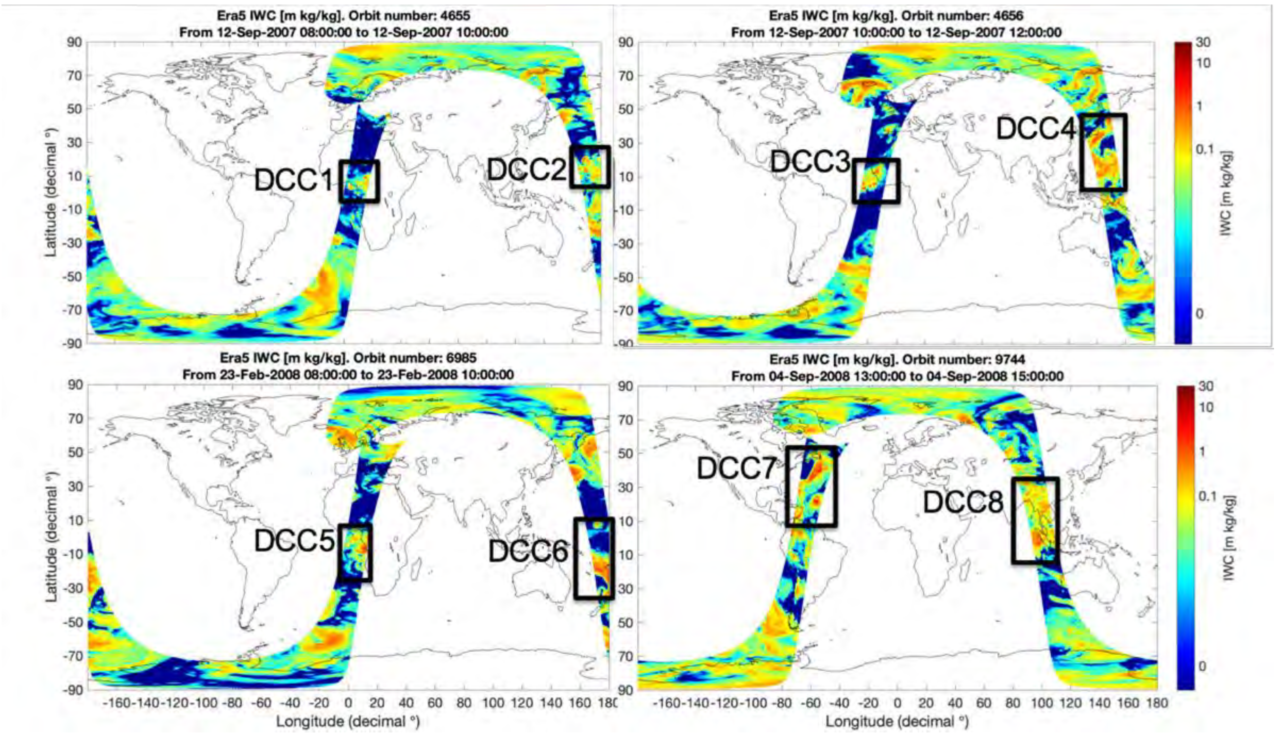



***Figure 6.4.1****: Integrated Ice Water Path (m kg/kg) from ERA5 forecasts as in the title of each panel for some selected Region of Interest (ROI) for eight identified Deep Convective Clouds (DCCs).*

## 6.5 Proof of concept test dataset

To assess the relative geolocation accuracy of ICI channels, a test dataset has been generated. In the test dataset we added a true unknown pointing error to all ICI channels to simulate a malfunctioning in the pointing system of ICI. The pointing error configuration that we assumed is listed in the last two columns in Table 6.2.1, where we indicated with the symbols $\delta\theta_0$ and $\delta\varphi_0$ the true unknown pointing errors in the elevation and azimuth directions, respectively. Note that ICI-1, ICI-2 and ICI-3 channels are supposed to have no pointing errors in our tests because they could benefit by a geolocation correction procedure previously applied using some specific ground reference target. For this reason, in our initial tests we assumed ICI-1 as the perfectly geolocated reference channel. An example of the actual (error-affected) pointing position along the direction $^{(0,0)}=(\theta_i+\delta_{\theta 0},\ \varphi_i+\delta_{\varphi 0})$ of an instantaneous FOV for channel *i*-th, is shown in Figure 6.5.1 (red star position A). The angles, $\theta_i=\theta+\Delta\theta_i$ and $\varphi_i=\varphi+\Delta\varphi_i$ identify the direction $=(\theta_i,\ \varphi_i)$ of the nominal satellite-Earth line of sight pointing (i.e. error-free case) for channel *i*-th (black circle position B in Figure 6.5.1). Point B is horizontally separated by approximately 3.2 km from the actual pointing position in A. Note that in Figure 6.5.1, Point A just represents the horizontal displacement of a single FOV which should have been positioned in B, whereas an exact illustration of the pointing error should represent the displacement to the North West by 3.2 km of all FOVs in each scan line (blue circles). However, since the pointing position in B is in principle unknow, one way to obtain an estimate of position B is to test several trial pointing positions around the nominal one (known) in B. To this aim, we simulated, for each FOV of several ICI channels and for each considered orbit, several trial pointing positions, $^{(\ ,\ )}=(\theta_i+\delta_{\theta k},\ \varphi_i+\delta_{\varphi l})$ (indicated by orange stars in Figure 6.5.1) around the position B of the nominal visited FOVs. The generated trial pointing errors ($\delta\theta_k$, $\delta\varphi_l$) are listed in Table 6.5.1. The indexes *k* and *l* vary both from 0 to 8 and they identify the (*k, l*)-th trial pointing position error pair among the 9×9 combinations that can be obtained picking up differently the values in Table 6.5.1. In order to have a test dataset in the same reference grid, for each modification of the pointing positions, we used the Backus-Gilbert approach to remap all FOVs of channels ICI-2, ICI-4, ICI-5, ICI-8 and ICI-11 into the ICI-1 reference channel.

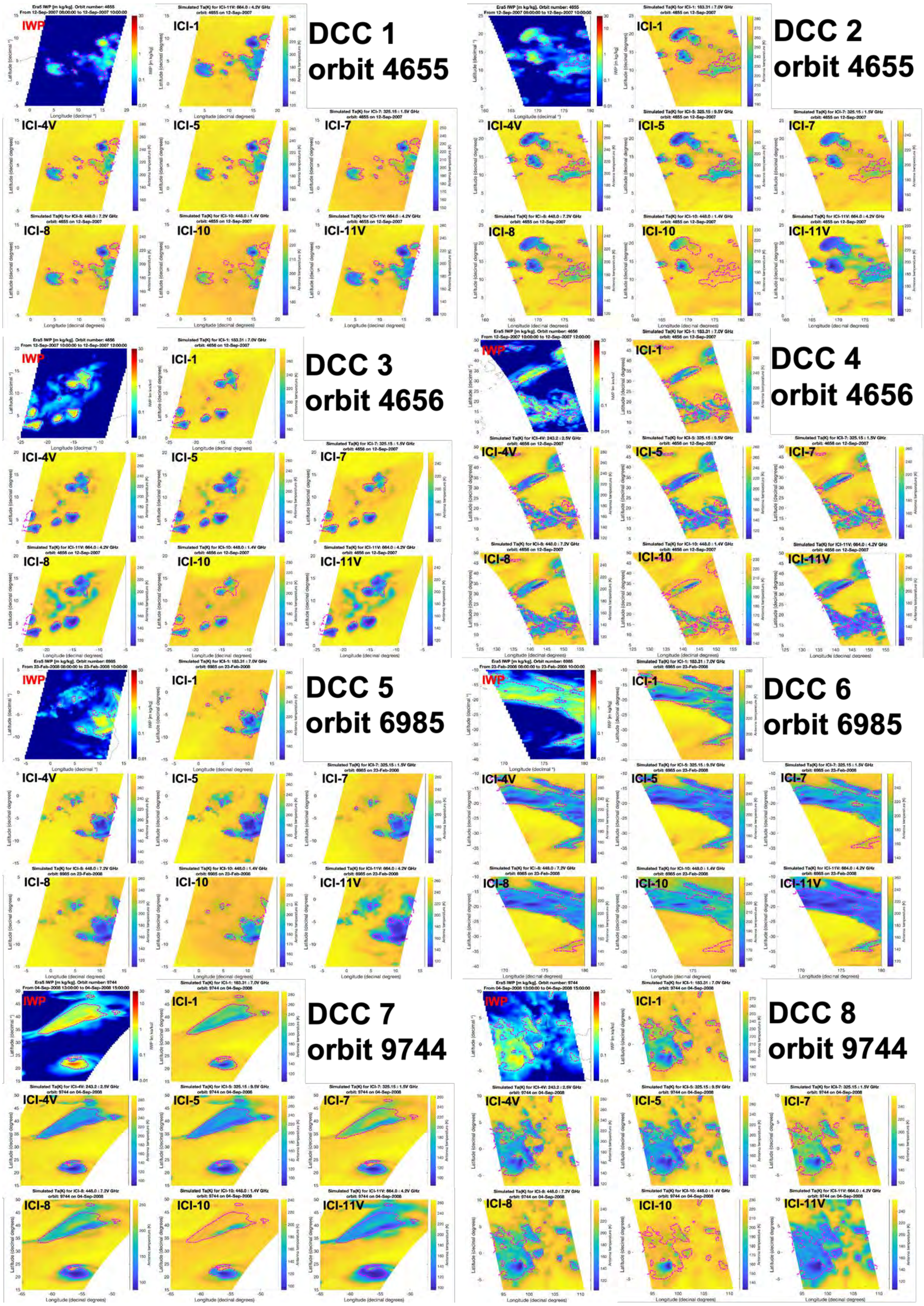



***Figure 6.4.2**: Simulated $T_A$ (K) Deep Convective Cloud (DCC) signatures for some ICI channels as specified in the title of each panel. The upper left panel in each box shows the Ice Water Path (m kg/kg) of the scene.*

***Table 6.5.1**: Assumed trial pointing positions with respect to the nominal pointing position*

| Azimuth trial pointing positions $\delta\varphi_l$ (°) | -0.36 | -0.27 | -0.18 | -0.09 | 0 | 0.09 | 0.18 | 0.27 | 0.36 |
|---|---|---|---|---|---|---|---|---|---|
| Elevation trial pointing positions $\delta\theta_k$ (°) | -0.18 | -0.135 | -0.09 | -0.045 | 0 | 0.045 | 0.09 | 0.135 | 0.18 |

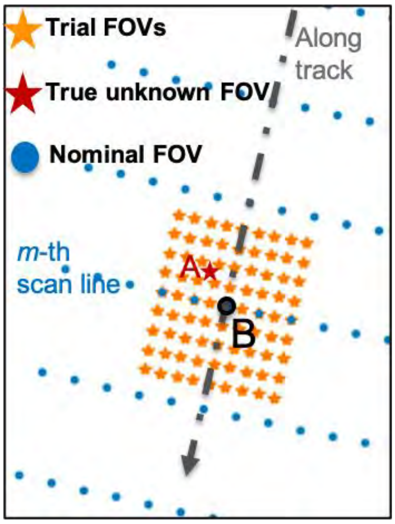



***Figure 6.5.1**: Example of the distribution of nominal FOV positions for ICI-5 (blue dots) The orange stars indicate the trial positions obtained according to the trial azimuth and elevations in Table 6.5.1. The red star (A) indicates the position for the assumed azimuth and elevation error of 0.13 and 0.07 degrees (about 3.2 km in distance) with respect to the nominal FOV position in B.*

## 6.6 Relative geolocation strategies

ICI relative geolocation aims at producing an estimate ( ˆ , ˆ ) of the pointing error ($\delta_{\theta 0}$, $\delta_{\varphi 0}$), by using some correlation approach between $T_A($ $^{(0,0)})$ and $T_A($ $_1)$ which are the antenna temperatures of the tested (ICI-*i*-th) and reference channel (ICI-1) observed along the actual direction $^{(0,0)}$=($\theta_i$+$\delta_{\theta 0}$, $\varphi_i$+$\delta_{\varphi 0}$) and the nominal direction $^{(0,0)}_1$=($\theta_1$, $\varphi_1$), respectively. A perfect estimate would imply ˆ = $_0$ and ˆ = $_0$. However, since $^{(0,0)}$and $_1$ vary on two different reference grids, a Backus-Gilbert remapping operation is applied to remap $T_A($ $^{(0,0)})$ on $T_A($ $_1)$. We indicated the remapped version of $T_A($ $^{(0,0)})$ on $_1$ with $T_A($ $^{(0,0)}$; $_1)$. The two independent approaches followed to obtain ( ˆ , ˆ ), are described in Figure 6.6.1 and 6.6.2 and explained in the next sections.

### 6.6.1 Closed Loop Correlation (CLC)

In Figure 6.6.1 we implemented the approach suggested in (Bennartz et al., 2005), hereafter referred to as **Closed Loop Correlation (CLC)** approach. In this approach some explicit iterations are applied to test several possible pointing errors ( , ). Since the indexes *k* and *l* vary both from 0 to 8, we have 9×9 iterations. At each iteration the correlation coefficient ( , ) between $T_A($ $^{( , )}$; $_1)$ and $T_A($ $_1)$ is calculated and the result is stored. The indexes *k* and *l* are then updated before the next iteration starts again. The maximum number of iterations is set to 81 according to Table 6.5.1. The expected final result is a 9×9 correlation matrix showing ( , ) and where its *argmax* coincides with our final estimation ( ˆ , ˆ ).

It is worth noting that since the resolution of the trial pointing positions in Table 6.5.1 is low it does not allow an accurate position estimation. Thus, before calculating the *argmax*, we applied a cubic interpolation to ( , ) to a finer grid of resolution of 0.0036° and 0.0073° in elevation and in azimuth, respectively. The procedure is applied to each of the eight regions of interest previously identified in Figure 6.4.2 to have 8 × 9 × 9=648 values of .

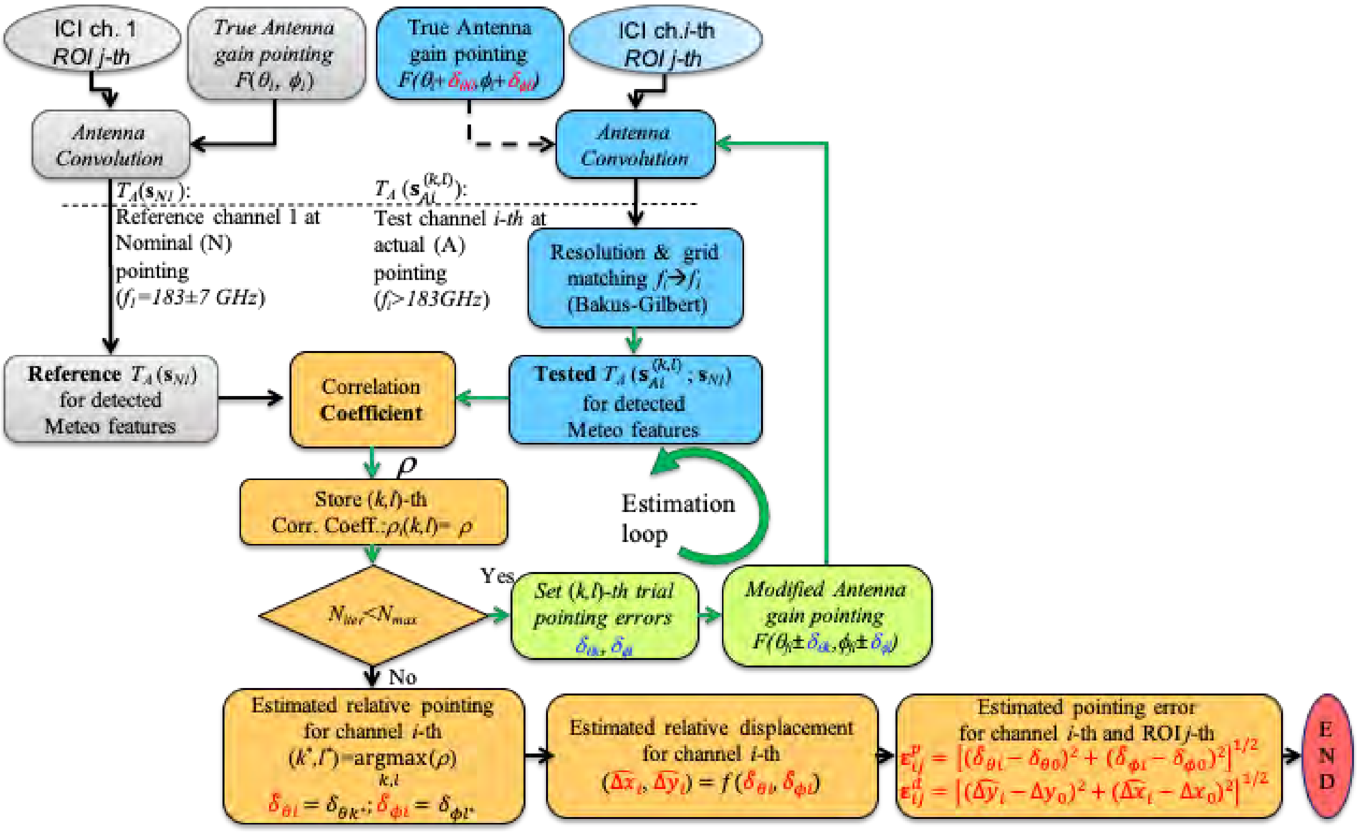



***Figure 6.6.1**: Block diagram of Closed Loop Correlation approach to test the relative pointing error of ICI channel i-th with respect to a ICI reference channel for DCC targets.*

### 6.6.2 Open Loop Correlation (OLC)

The second approach implemented (figure 6.6.2) is slightly different from the CLC. It does not include any explicit closed loop and for this reason it is named **Open Loop Correlation (OLC).** Contrarily to CLC, in OLC a bi-dimensional correlation function, $(x_i,y_i)$, instead of a correlation matrix ( , ) between $T_A($ $^{(0,0)}$; $_1)$ and $T_A($ $_1)$, is calculated using the algorithm described inGuizar-Sicairos et al. (2008). Hence, the across $(x_i)$ and along track $(y_i)$ displacement between the two inputs, ( , ), is directly estimated looking at the argmax of $C_i$ without the need of any loop. However, in order to make the results of the OLC and CLC approaches comparable each other, the following conversion formulas for a platform at altitude $h$, were used to convert ( , ) back into ( , ) or vice versa.

$$\Delta \quad = h \cdot \quad \cdot [ \quad ( \quad )]^2 \tag{10}$$

where Δ can be or and is or , respectively. In order to avoid including inputs that are poorly correlated with each other, and to avoid the risk of the final result deterioration, a correlation check can be added in order to discard input ROIs for which the correlation coefficient between the registered version (i.e. displacement compensated) of $T_A(\ ^{(0,0)};\ _1)$ and reference $T_A(\ _1)$ is below a fixed threshold $h$.

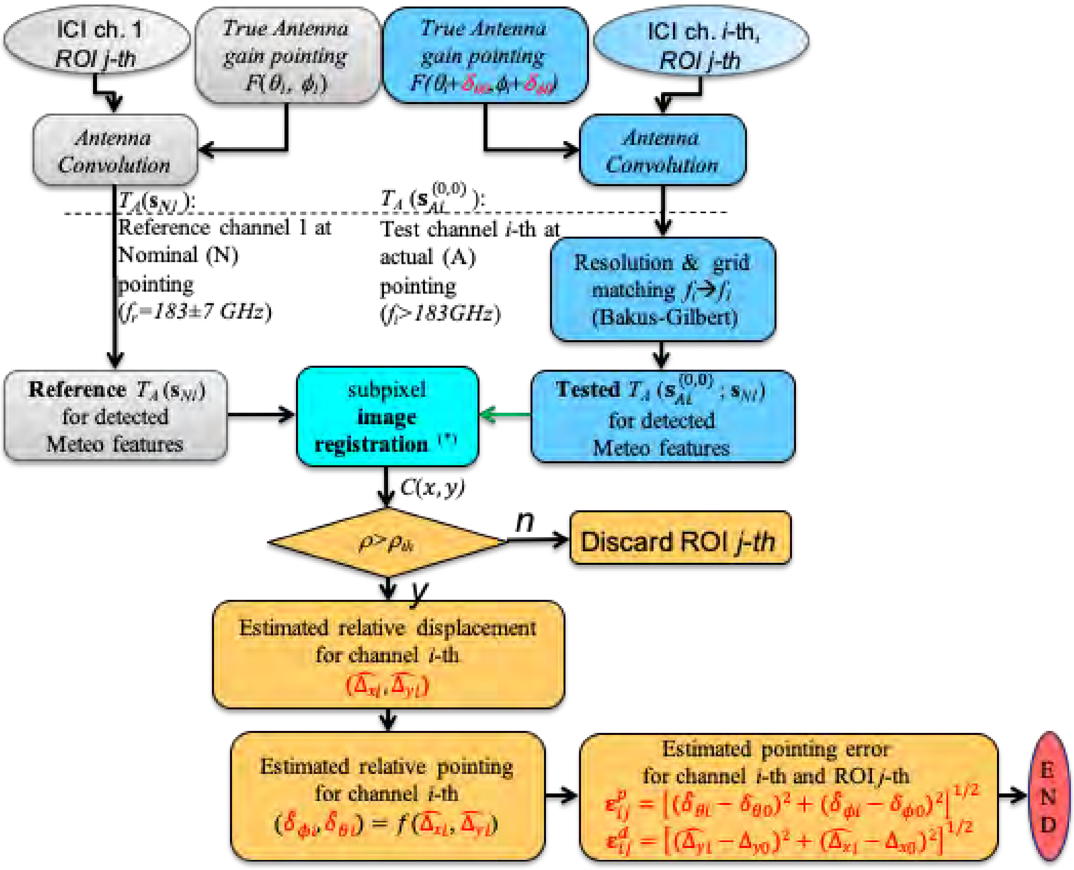



***Figure 6.6.2**: Block diagram of Open Loop Correlation approach to test the relative pointing error of ICI channel i-th with respect to a ICI reference channel for DCC targets. (*) subpixel image registration in (Guizar-Sicairos et al., 2008) is used to find the best correlation between reference and tested $T_A$.*

### 6.6.3 Automatic detection of Region Of Interests (ROIs)

Both schemes shown in Figures 6.6.1 and 6.6.2 for the relative geolocation of ICI channels start from a predefined region of interest (ROI). For a practical implementation, the ROI definition needs to be automated. The approach followed to define the ROI for DCC targets is inspired by the method suggested by Hong et al. (2005). Hence, the criteria used to identify a DCC pattern from ICI, foresees the following check:

$$\Delta\ _{31} > \Delta\ _{32} > \Delta\ _{21} > 0 \tag{11}$$

where $\Delta$ is the difference in (K) between $T_A$ at $183.31 \pm i$ GHz and $T_A$ at $183.31 \pm j$ GHz. Figure 6.6.3 (left) shows an example of the implementation of Eq. (11) when it is applied to the simulated scenario of DCC 3, orbit 4656, shown in figure 6.4.2. Subsequent steps are needed in order to identify the final ROIs. ROIs are defined as bounding boxes containing DCCs. In the first step, after the DCC identification, some bounding boxes are defined around each DCC in the ICI-1 $T_A$ field (Figure 6.6.3 middle panel). Such bounding boxes are enlarged by 20% with respect to their natural minimum size in order to have the chance to include the same DCC feature in each tested ICI channel that in principle can be affected by some

geolocation error with respect to the reference channel (e.g. ICI-1). Note that bounding boxes expansion here implemented, considers the “box-expansion” in the number of pixels without any georeferencing. This can lead to differences when detecting DCC in the first or last part of the orbit.
In the second step (right panel), the overlapping bounding boxes are merged together in the attempt of including those DCCs belonging to the same convective system.

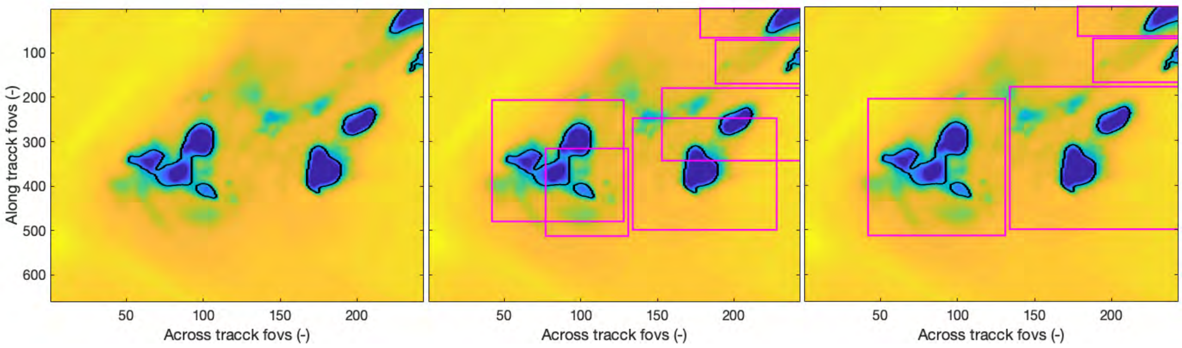



**Figure 6.6.3**: *Deep convective clouds (DCCs) detection are highlighted by black contours (left) of $T_A$ at 183.31±7 GHz (ICI-1). Some preliminarily regions of interest (ROIs) are defined as enlarged bounding boxes (magenta) around each DCC (middle). The overlapping bounding boxes are merged together thus defining the final ROIs (magenta lines on the right panel) used in input to CLC and OLC methods.*

## 6.7 Results

Results of the CLC and OLC approaches are here presented. Figure 6.7.1, in each panel, shows the 9×9 interpolated correlation matrix ( , ) for the ICI-*i*-th channel averaged over the eight ROIs of DCC (grey colour) manually identified in Figure 6.4.2. To obtain the spatial displacement we assumed a constant sensor altitude and distance between scan lines and samples. In this figure, ICI-1 is set as reference and the lower bound threshold for the correlation coefficient $_h$ used to discard poorly correlated ROIs, is set to zero, hence all the possible correlation degrees are considered. Geolocation error scores obtained applying the automatic detection of DCC features are discussed later. Overimposed to the average there are the circles and triangles markers that show the positions of estimated (ˆ ,ˆ ) for CLC (blue circles) and OLC (orange triangles) approaches.
The empty cyan circles and yellow triangles are the average estimated pointing positions considering all the eight manually selected DCC cases and they represent the first final result of the two tested procedures. The unknown true pointing position is indicated by a red filled square. As can be noted, in three cases out of seven, the requirement of 2.5 km RMSE is not met (see for example orange RMSE values for OLC in each panel of figure 6.7.1 for ICI-8 and ICI-11). Contrarily, the STD error is in all cases well below 2.5 km for OLC approach. Hereafter, to have a more conservative error analysis, we compared the different channel performances in terms of RMSE than error STD. For ICI-8 and ICI-11 channels the relative pointing approach seems to be less accurate with RMSE above 2.5 km. Note that we included ICI-2 in the pointing verification although we do not expect that ICI-2 as well as ICI-3 can have a different pointing with respect to ICI-1 since these three channels share the same front end. Some differences between the CLC and OLC are also noted. OLC (orange markers) is in general more precise (lower error standard deviation) than CLC (blue markers) and shows a lower RMSE, especially when considering ICI-8 and ICI-11 where both OCL and CLC gives RMSE larger than the 2.5 km error requirement. Contrarily CLC outperforms OLC when considering ICI-4H, ICI-4V. As a general guidance for future implementation, if we accept an upper limit geolocation error higher than 2.5 km, let say below 5km, OLC could be good solution because it reasonably fulfil the geolocation requirement, it guarantees higher geolocation precision (i.e. lower dispersion) and it is expected to be fast to implement since it does not require estimation loops.

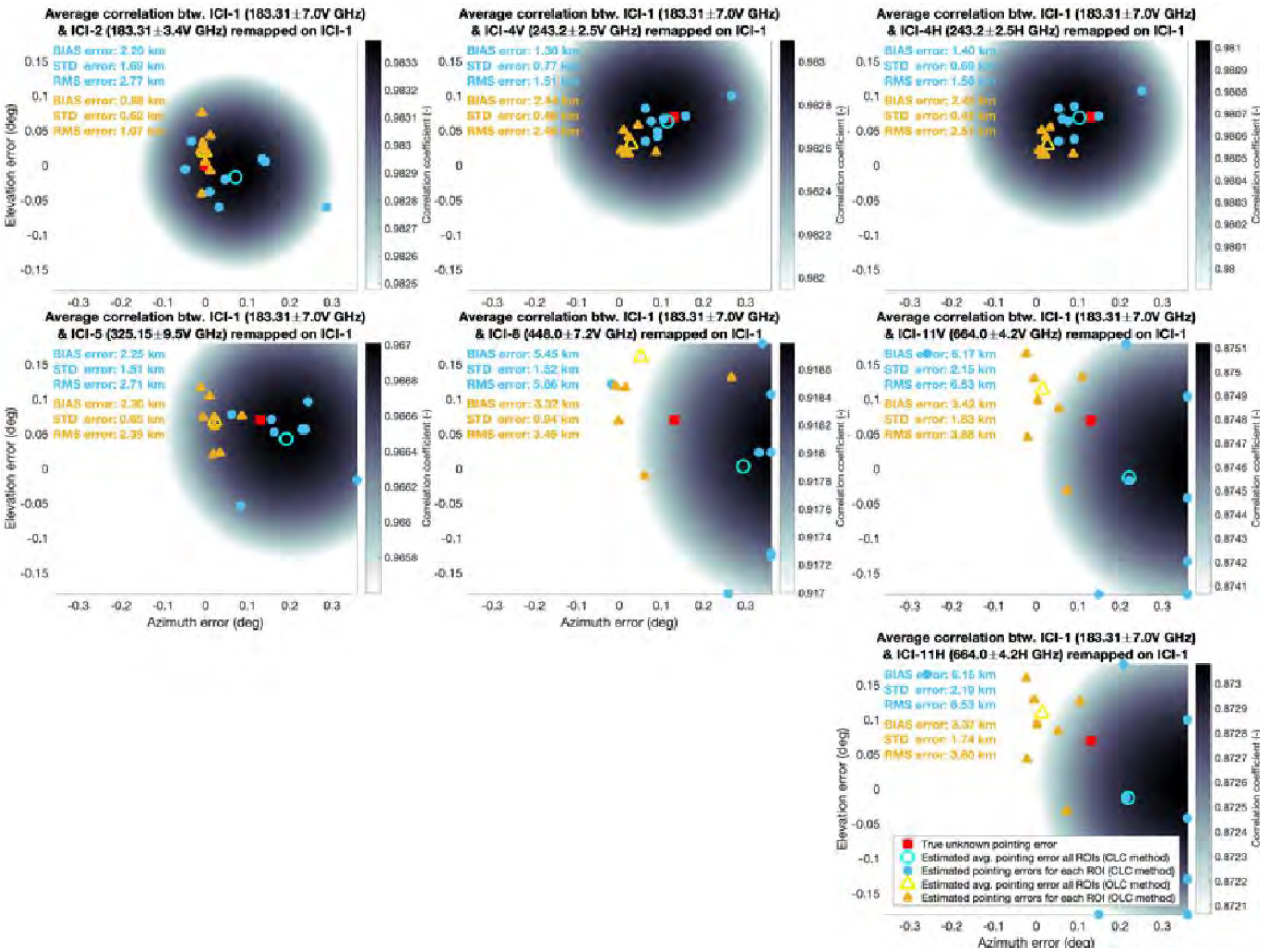



***Figure 6.7.1****: Average correlation diagrams and quantitative pointing retrieval errors for ICI channels as indicated in each panel's title.*

Table 6.7.1, lists the geolocation errors for the open loop correlation (OLC) approach as a function of the method used for the identification of the DCC regions of interest. In this table, the geolocation errors shown in Figure 6.7.1 for the OLC method are indicated as "manually detected ROI" and labelled as "test case A". Contrarily, when ROIs are automatically detected following the methodology previously exposed in Section 5.6.3, we obtain slightly different results in terms of RMSE (test case B) with respect to the "manually detected ROI" (test case A), albeit the overall conclusions remain substantially unchanged. The main difference between the two ROI detection methods are found in terms of STD that in some cases (e.g.: ICI-1 vs. ICI-11) can be more than double. In general, the scores in Table 6.7.1, point out the importance of the ROI selection. On top of this, it is evident that the geolocation error increases by approximately 70% for ICI-8 and ICI-11 with respect to ICI-4 and ICI-5, when ICI-1 is set as reference.

The slightly worse performance obtained for ICI-8 and ICI-11 with respect to ICI-4 and ICI-5 shown in Table 6.7.1 test case B, could be related to the differences in the information content in the $T_A$ observations of these two sets of channels. Qualitatively, this is quite evident from Figure 6.4.2 where the $T_A$ depression due to the scattering by the ice in the convective cloud has a different pattern, i.e. clouds are more smeared, for ICI-11 than ICI-4, for example.

To further test the performance of OLC method, we set ICI-4V as the reference channel. As done for ICI-1, we assumed ICI-4 to be already (perfectly) geolocated using some external method previously applied.

For easier computation, we still continue remapping all the considered channels into ICI-1 grid and we consider the same channel's error structure shown in Table 6.2.1.

***Table 6.7.1***: *Geolocation errors (km) for the open loop correlation (OLC) approach as a function of the method used for the identification of the DCC Regions Of Interest and for various channel comparisons. ICI-1 is set as a reference channel. In test case B, those displacements larger than 12 km are discarded by the analysis.*

| | **TEST CASE A** | | | | **TEST CASE B** | | | |
|---|---|---|---|---|---|---|---|---|
| | Manually identified ROIs as in figure 5.4.2 (8 ROIs are considered) | | | | Automatically identified ROIs as in section 5.6.3 (all detected ROIs are considered) | | | |
| REF vs. TEST | BIAS | STD | RMSE | N ROIs | BIAS | STD | RMSE | N ROIs |
| ICI-1 vs. ICI-2 | 0.88 | 0.62 | 1.07 | 8 | 1.07 | 0.70 | 1.28 | 18 |
| ICI-1 vs. ICI-4V | 2.44 | 0.46 | 2.49 | 8 | 2.41 | 0.57 | 2.48 | 18 |
| ICI-1 vs. ICI-4H | 2.48 | 0.42 | 2.51 | 8 | 2.44 | 0.55 | 2.50 | 18 |
| ICI-1 vs. ICI-5 | 2.30 | 0.65 | 2.39 | 8 | 2.48 | 0.70 | 2.58 | 18 |
| ICI-1 vs. ICI-8 | 3.32 | 0.94 | 3.45 | 8 | 4.92 | 3.15 | 4.40 | 16 |
| ICI-1 vs. ICI-11V | 3.42 | 1.83 | 3.88 | 8 | 4.53 | 2.72 | 4.58 | 17 |
| ICI-1 vs. ICI-11H | 3.37 | 1.74 | 3.80 | 8 | 4.46 | 2.63 | 4.54 | 17 |

***Table 6.7.2***: *As for Table 6.7.1 but when channel ICI-4V (243.2 ±2.5 GHz) is considered as reference. All the channels still continue to be remapped into ICI-1 grid for convenience. In test case B, those displacements larger than 12 km are discarded by the analysis.*

| | **TEST CASE A** | | | | **TEST CASE B** | | | |
|---|---|---|---|---|---|---|---|---|
| | Manually identified ROIs as in figure 5.4.2 (8 ROIs are considered) | | | | Automatically identified ROIs as in section 5.6.3 (all detected ROIs are considered) | | | |
| REF vs. TEST | BIAS | STD | RMSE | N ROIs | BIAS | STD | RMSE | N ROIs |
| ICI-4V vs. ICI-2 | 2.71 | 0.46 | 2.75 | 8 | 2.91 | 0.61 | 2.97 | 18 |
| ICI-4V vs. ICI-4V | 0.00 | 0.00 | 0.00 | 8 | 0.00 | 0.00 | 0.00 | 18 |
| ICI-4V vs. ICI-4H | 0.06 | 0.03 | 0.07 | 8 | 0.10 | 0.13 | 0.16 | 18 |
| ICI-4V vs. ICI-5 | 1.01 | 0.44 | 1.10 | 8 | 0.95 | 0.83 | 1.27 | 18 |
| ICI-4V vs. ICI-8 | 3.46 | 1.81 | 3.90 | 8 | 3.29 | 2.54 | 4.15 | 17 |
| ICI-4V vs. ICI-11V | 2.65 | 1.02 | 2.84 | 8 | 2.92 | 2.44 | 3.81 | 18 |
| ICI-4V vs. ICI-11H | 2.54 | 0.98 | 2.72 | 8 | 2.84 | 2.37 | 3.70 | 18 |

The results obtained are listed in Table 6.7.2, which needs to be compared with Table 6.7.1. Comparing the RMSE values in these two tables, for test case B, it is noted that the use of ICI-4V as reference channel instead of ICI-1, substantially improves the geolocation error by a factor 2, for ICI-5, and 1.2 for ICI-11, respectively, but, on the other hand, the geolocation of ICI-2 deteriorates by a factor 2.3 whereas ICI-8 is unaffected.
However, in general the ICI-4V seems to be a better choice than ICI-1 to be the pivot channel, since it produces, except for ICI-2, lower errors whereas the contrary is true when ICI-1 is selected as reference. Likely, similar conclusions are expected if we consider ICI-4H as reference. A special care needs to be put

for the geolocation of ICI-2 (and likely to ICI-1 and ICI-3) that could suffer from larger errors when ICI-4 is the reference.
The better performance obtained making use of the reference ICI-4V, could be fostered by a larger correlation of ICI-4V with the other tested channels. This can be deduced from Figure 6.7.2 where it is noted that the correlation among ICI-4V and all the other tested channels (panel b) is in general higher than in the cases where ICI-1 is set as reference (panel a). Particularly instructive is the case of ICI-2 that shows a lower correlation (larger error) in panel b) when ICI-4V is set as reference as opposed to panel a) where ICI-1 is the reference. This suggests that the lower correlation of ICI-4V vs. ICI-2 might be responsible for the larger ICI-2 geolocation error that we found.

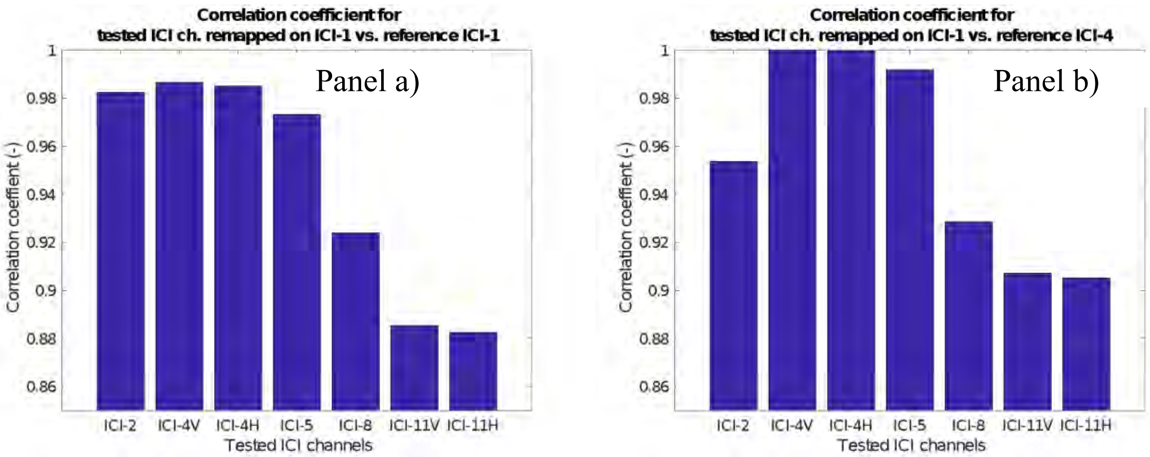



*Figure 6.7.2: correlation coefficient between tested and ICI-1 (left) and ICI-4V (right) reference channel.*

A final consideration about the performance of ICI-4V is in terms of FOV geometrical aspects. Since ICI-4 and ICI-11 align along the same across track direction (figure 6.2.1 b), it potentially has a viewing geometry closer to ICI-11 than ICI-1. This could contribute to explain the better results in geolocating ICI-11 when using ICI-4 as reference channel instead of ICI-1. However, from our results, the benefits of using ICI-4 as reference, are not limited to ICI-11 but extend to ICI-5 as well (Table 6.7.2). Thus, once all the channels are remapped on the same reference system, the geometric proximity principle does not seem to be the key factor that drives the final results.

## 6.8 Conclusions

For what discussed in the previous sections the conclusion is that the relative pointing accuracy estimation methodology is mainly useful if the information content in the measured $T_A$ is similar (i.e. higher correlation level) between the two channels of concern: reference and test channel. This implies that if the geolocation error requirement is as low as 2.5 km, none of the methods tested strictly satisfy such requirement for all ICI channels in terms of RMSE albeit it does in terms of error STD. The Open Loop Correlation method, assuming ICI-4 as the perfectly geolocated reference channel, is promising since from our simulated tests it produces errors below 4.1 km. Contrarily, if ICI-1 can be accurately geolocated making use of some external methods previously applied (e.g. using ground reference targets as done in Task 1), we can verify the geolocation of ICI-2 (and likely ICI-3) as well as ICI-4 and ICI-5 with a satisfactory RMS error closer to 2.5 km, albeit, we do not expect pointing differences in ICI-1, ICI-2 and ICI-3 since they all share the same pointing frame. The drawback when selecting ICI-1 as a reference channel is that it produces larger RMS geolocation errors than those obtained with ICI-4, which result of the order of 4.5 km for ICI-8 and ICI-11.
Obviously, it is worth noting that residual pointing errors in the assumed reference channel add to the relative geolocation errors that we found. In this respect, ICI-4 could be geolocated more accurately than ICI-1 by using the landmark approach, since ICI-4 it is less sensitive to surface emissivity than ICI-1.

# 7. Practical guidance and future recommendations

- As a general guidance for future implementation, Open loop Correlation approach for relative geolocation (Figure 6.6.2), could be a good solution because it is expected to fulfil the geolocation requirement (in terms of relative error STD), it guarantees higher geolocation precision (i.e. lower dispersion), and it is expected to be fast to implement since it does not require estimation loops.
- Assuming ICI-4 as a reference channel in the OLC, we expect to achieve relative geolocation errors of less than 4.1 km for all the ICI channels with larger errors for ICI-8. The relative geolocation of ICI-1, ICI-2 and ICI-3 could be an issue when ICI-4 is set as a reference, and an absolute geolocation approach (e.g. using landmarks or water vapor masses features) could be a safer option in this case.
- Water Vapor Masses (WVM) features analysed in this document as seen by ICI-2 and ICI-3-like SSMIS channels and MSG SEVIRI water vapor 6.3 µm channel, have demonstrated to be a good candidate target to be used for an absolute geolocation of ICI-2 and/or ICI-3. In this respect, the achieved geolocation errors are fully comparable, in terms of RMSE, with those obtained from the landmark approach. Then, for a future implementation, we suggest to accurately take the WVM approach into consideration at least as an optional off-line tool even operated by third parties.

# 8. References

AD1 GAMES, Geolocation Assessment/validation Methods for EPS-SG ICI and MWI", In response to: EUMETSAT ITT 19/218140 "Development of Geolocation Validation Methods for EPS-SG ICI and MWI", Final report – Task 1.

AD2: Technical report Backus-Gilbert footprint matching methodology applied on MWI and ICI observations TR/BG/MWI-ICI Issue 1.0, Rev.

Adler, R.F, M. J. Markus, D. D. Fen, G. Szejwach, and W. E. Shenk, 1983: Thunderstorm top structure observed by aircraft overflights with an infrared radiometer. J. Climate Appl. Meteor., 22, 579–593.

Aires F, Catherine Prigent, Fr´ed´eric Bernardo, Carlos Jim´enez, Roger Saunders, and Pascal Brunel. A tool to estimate land-surface emissivities at microwave frequencies (telsem) for use in numerical weather prediction. Q. J. R. Meteorol. Soc., 137(656):690–699, 2011.

Alcala, C. M., and Dessler, A. E. (2002), Observations of deep convection in the tropics using the Tropical Rainfall Measuring Mission (TRMM) precipitation radar, J. Geophys. Res., 107(24), 4792, doi:10.1029/2002JD00245

Bedka, K., Brunner, J., Dworak, R., Feltz, W., Otkin, J., & Greenwald, T. (2010). Objective satellite-based detection of overshooting tops using infrared window channel brightness temperature gradients. Journal of applied meteorology and climatology, 49(2), 181-202.

Bennartz, 2005, Technical report "NOOA-18 absolute geo-rerefence of AMSU-A and MHS and relative pointing accuracy of MHS and AMSU-A".

Buehler, S. A. , J. Mendrok, P. Eriksson, A. Perrin, R. Larsson, and O. Lemke. ARTS, the Atmospheric Radiative Transfer Simulator – version 2.2, the planetary toolbox edition. Geosci. Model Dev., 11(4):1537–1556, 2018. doi: 10.5194/gmd-11-1537-2018.

Eriksson P, R. Ekelund, J. Mendrok, M. Brath, O. Lemke, and S. A. Buehler. A general database of hydrometeor single scattering properties at microwave and sub-millimetre wave- lengths. Earth Syst. Sci. Data, 10(3):1301–1326, 2018. doi: 10.5194/essd-10-1301-2018

Ferraro, R., Beauchamp, J., Cecil, D., & Heymsfield, G. (2015). A prototype hail detection algorithm and hail climatology developed with the advanced microwave sounding unit (AMSU). Atmospheric Research, 163, 24-35.

Funatsu, B. M., Claud, C., & Chaboureau, J. P. (2009). Comparison between the large-scale environments of moderate and intense precipitating systems in the Mediterranean region. Monthly Weather Review, 137(11), 3933-3959.

Glickman, T., Ed., 2000: Glossary of Meteorology. 2nd ed. Amer. Meteor. Soc., 855 pp.

Guizar-Sicairos M., Samuel T. Thurman, and James R. Fienup, "Efficient subpixel image registration algorithms," Opt. Lett.33, 156-158 (2008)

Hong, G., Heygster, G., Miao, J., and Kunzi, K. ( 2005), Detection of tropical deep convective clouds from AMSU-B water vapor channels measurements, J. Geophys. Res., 110, D05205, doi:10.1029/2004JD004949.

Liu, C., and E. J. Zipser (2005), Global distribution of convection penetrating the tropical tropopause, J. Geophys. Res., 110, D23104, doi:10.1029/2005JD006063.

Marra, A. C., Porcù, F., Baldini, L., Petracca, M., Casella, D., Dietrich, S., ... & Panegrossi, G. (2017). Observational analysis of an exceptionally intense hailstorm over the Mediterranean area: Role of the GPM Core Observatory. Atmospheric research, 192, 72-90.

Negri, A. J., 1982: Cloud-top structure of tornado storms on 10 April 1979 from rapid scan and stereo satellite observa- tions. Bull. Amer. Meteor. Soc., 63, 1851–1859.

Prigent C, F Aires, D Wang, S Fox, and C Harlow. Sea-surface emissivity parametrization from microwaves to millimetre waves. Quarterly Journal of the Royal Meteorological Society, 143 (702):596–605, 2017.

Ralph, F.M., K.A. Prather, D. Cayan, J.R. Spackman, P. DeMott, M. Dettinger, C. Fairall, R. Leung, D. Rosenfeld, S. Rutledge, D. Waliser, A.B. White, J. Cordeira, A. Martin, J. Helly, and J. Intrieri, 2016: CalWater Field Studies Designed to Quantify the Roles of Atmospheric Rivers and Aerosols in Modulating U.S. West Coast Precipitation in a Changing Climate. Bull. Amer. Meteor. Soc., 97, 1209–1228, https://doi.org/10.1175/BAMS-D-14-00043.1

Rosenkranz P. W.. Water vapor microwave continuum absorption: A comparison of measurements and models. *Radio Sci.*, 33(4):919–928, 1998. (correction in 34, 1025, 1999), ftp://mesa.mit.edu/phil/lbl_rt.

Sanò, P., Panegrossi, G., Casella, D., Di Paola, F., Milani, L., Mugnai, A., ... & Dietrich, S. (2015). The Passive microwave Neural network Precipitation Retrieval (PNPR) algorithm for AMSU/MHS observations: description and application to European case studies. Atmospheric Measurement Techniques, 8(2).

Setvak, M., R. M. Rabin, and P. K. Wang, 2007: Contribution of the MODIS instrument to observations of deep convective storms and stratospheric moisture detection in GOES and MSG imagery. Atmos. Res., 83, 505–518

Stogryn, A. 1976: Estimates of brightness temperatures from scanning radiometer data, *IEEE Trans. Antennas Propagat.*, AP-26, 720–726.

Sun, L. X., Zhuge, X. Y., & Wang, Y. (2018). A Contour-Based Algorithm for Automated Detection of Overshooting Tops Using Satellite Infrared Imagery. IEEE Transactions on Geoscience and Remote Sensing, 57(1), 497-508

Turk, J. CloudSat-GPM coincidence dataset (version 1C). In NASA Technical Report; California Institute of Technology: Pasadena, CA, USA, 2016

# GAMES

## "Geolocation Assessment/validation Methods for EPS-SG ICI and MWI"

In response to: EUMETSAT ITT 19/218140 "Development of Geolocation Validation Methods for EPS-SG ICI and MWI"

## Deliverable document 08 – D08
## **Algorithm Theoretical Baseline Document**

prepared by

**M. Papa, F.S. Marzano**
**M. Montopoli, D. Casella, B. Rydberg**

| | |
|---|---|
| **Sapienza University of Rome**<br>**Center for Research in AeroSpace**<br>**(SUR-CRAS)** | |
| **Möller Data Workflow Systems AB**<br>**(Molflow)** | |
| **National Research Council of Italy**<br>**Institute of Atmospheric Sciences and Climate**<br>**(CNR-ISAC)** | |

Date: 18 November 2020

# OUTLINE

# LIST OF ACRONYMS

| | |
|---|---|
| AMSU | Advanced Microwave Sounding Unit |
| ATMS | Advanced Technology Microwave Sounder |
| BT | Brightness Temperature |
| DCC | Deep Convective Clouds geolocation-error assessment method |
| DEM | Digital Elevation Model |
| FNC | Fast Normalized Cross-correlation |
| FR | Full Resolution |
| GRD | Ground Range Detected |
| GSHHG | Global Self-consistent, Hierarchical, High-resolution Geography database |
| HR | High Resolution |
| ICI | Ice Cloud Imager radiometer |
| MHS | Microwave Humidity Sounders |
| MR | Medium Resolution |
| NOAA | National Oceanic and Atmospheric Administration |
| RAOB | RAwinsonde OBservation |
| RFD | Registration in Frequency Domain |
| SAR | Synthetic Aperture Radar |
| SNAP | Sentinel Application Platform |
| SNPP | Suomi National Polar-orbiting Partnership |
| SSMIS | Special Sensor Microwave Imager Sounder |
| SUR | Sapienza University of Rome |
| TCM | Target Contour Matching |
| TELSEM2 | Tool to Estimate Land.Surface Emissivities at Microwave version 2 |
| TESSEM2 | Tool to Estimate Sea Surface Emissivities at Microwave version 2 |
| TOA | Top Of Atmosphere |
| WVS II | Vector Shoreline Data Bank II |

# 1. INTRODUCTION

## 1.1 Purpose of this document

The EPS-SG Ice Cloud Imager (ICI) is a sub-millimetre wave conical imager on board of the EUMETSAT Polar System – Second Generation (EPS-SG) and it will have 11 channels with frequencies around 183, 243, 325, 448 and 664 GHz, as shown in Tab. 1.1.

*Table 1.1: Summary of ICI channels*

| CHANNEL | FREQUENCY (GHz) | BANDWIDTH (MHz) | NEΔT (K) | BIAS (K) | POLARISATION | FOOTPRINT SIZE AT 3 dB |
|---|---|---|---|---|---|---|
| ICI-1 | 183.31±7.0 | 2x2000 MHz | 0.8 | 1.0 | V | 16 km |
| ICI-2 | 183.31±3.4 | 2x1500 MHz | 0.8 | 1.0 | V | 16 km |
| ICI-3 | 183.31±2.0 | 2x1500 MHz | 0.8 | 1.0 | V | 16 km |
| ICI-4 | 243.2±2.5 | 2x3000 MHz | 0.7 | 1.5 | V, H | 16 km |
| ICI-5 | 325.15±9.5 | 2x3000 MHz | 1.2 | 1.5 | V | 16 km |
| ICI-6 | 325.15±3.5 | 2x2400 MHz | 1.3 | 1.5 | V | 16 km |
| ICI-7 | 325.15±1.5 | 2x1600 MHz | 1.5 | 1.5 | V | 16 km |
| ICI-8 | 448±7.2 | 2x3000 MHz | 14 | 1.5 | V | 16 km |
| ICI-9 | 448±3.0 | 2x2000 MHz | 1.6 | 1.5 | V | 16 km |
| ICI-10 | 448±1.4 | 2x1200 MHz | 2.0 | 1.5 | V | 16 km |
| ICI-11 | 664±4.2 | 2x5000 MHz | 1.6 | 1.5 | V, H | 16 km |

These wavelengths allow to detect ice clouds, whereas the emission signal from the surface is predominantly masked by high water vapour opacity. The latter is a problem for the geolocation assessment because current methods, comparing coastlines in imagery data with the known geographic locations, are not readily applicable to all ICI channels. However, in very dry atmospheric conditions, geolocation validation techniques using landmark targets could still be applied on observations at 183.3±7 GHz and 243.2 GHz.

For the channels where high water vapour opacity masks out the emission signal from the surface an alternative approach must be used, and it is suggested that the pointing of these channels are validated in a relative sense, using atmospheric targets and correlate data to 183.3±7 GHz or 243.2 GHz data to obtain a relative pointing error estimate compared to the 183.3±7 GHz or 243.2 GHz channel.

The purpose of this document is to describe the basis and details of the algorithms applied for geolocation assessment of ICI data, both using land mark and atmospheric targets.

## 1.2 Literature review

A wide experience has been accumulated so far on the geolocation error validation for satellite-based microwave radiometers at lower microwave frequencies (10-50 GHz, outside the absorption bands) by exploiting their strong difference in terms of surface emissivity between land and ocean. Global-scale coastlines can be used as surface landmarks with a significant contrast in terms of measured brightness temperature (BT). Comparing the latter with a reference coastline database [1], it is possible to assess the spaceborne sensor geolocation error.

In [2] Purdy et al. the shoreline obtained from WindSat satellite imagery and the World Vector Shoreline Data Bank II (WVS II) is compared. The position of the coastline is obtained taking the peak of the first derivative of radiometric data along scan and cross scan direction, after a cubic spline interpolation to obtain a more smoothed curve. Poe et al. [3] apply a similar method on Special Sensor Microwave Imager/Sounder (SSMIS) using data provided by spacecraft F-16.

In **[4]** Heygster et al. exploit the fact that, when geolocation errors are present, the projected footprints have different shifts considering ascending or descending swaths. Since the brightness temperature (BT) differences between ascending or descending swaths are higher along coastlines, they evaluate the geolocation error using data from AMSR-E at 89 GHz. Berg et al. [5] use the BT difference between ascending and descending swaths to obtain the attitude error for SSM/I spacecraft. Finally, Moradi et al. **[6]** correct the pitch, yaw and roll angles for Advanced Microwave Sounding Unit (AMSU) and Microwave Humidity Sounders (MHS) minimizing the difference in brightness temperature between ascending and descending swathes.

Along the coastline, the measured signal consists of radiation received from both land and water surfaces and Bennartz [7] proposed to use a high-resolution land–sea mask to infer the fraction of water surface for each measurement. He has developed a method to validate the geolocation accuracy using the convolution of land-sea masks that is suitable to apply for channels that are sensitive to land/sea contrast. Han et al. [8] adapted this so-called "Land/sea Fraction Method" for the NOAA 16-18 satellites and also for ATMS on SNPP.

Several algorithms can be applied to extract contour from images, starting from the simplest and faster to the most sophisticated, but with higher computational costs. In the following work we have focused on the Canny edge detector [9], because it is a fast algorithm that is able to detect both strong and weak edges [10], whereas its accuracy is slightly better than other algorithms [11], [12]. In addition we have also used the Sobel filter [13] as it is a fast approximation of image gradient [14].

## 1.3 Organization of the report

This ATBD is organized as follows.

Section 2 contains a description of the algorithms involved in the proposed methodology for geolocation assessment of MWI/ICI data using land mark targets, including a description of input and output data of the algorithm.

Section 3 describes the proposed algorithm for validation of the geolocation of data from ICI/MWI channels where high water vapour opacity masks out the emission signal from the surface, and where atmospheric targets can be used to assess the relative pointing error between two channels.

# 2. ASSESSMENT METHODOLOGY USING LANDMARK TARGETS

## 2.1 Overview

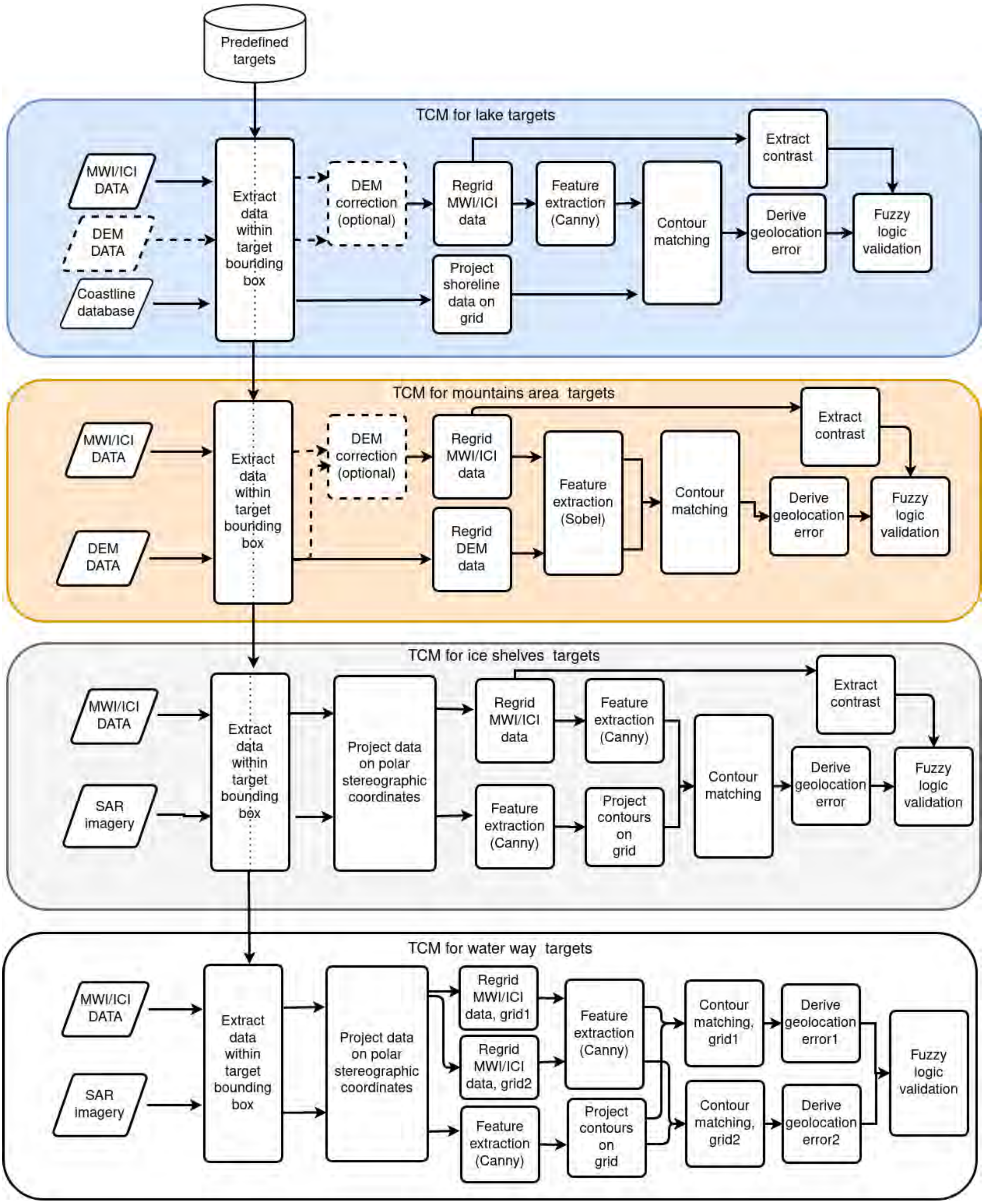



*Figure 2.1.1: Logical scheme of proposed methodology to validate the geolocation using landmark targets (target-contour matching algorithm). The methodology varies somewhat with type of targets and four different flows are shown.*

Figure 2.1.1 gives an overview of a target contour matching algorithm for geolocation error assessment of MWI or ICI Level-1B data using landmark targets as reference. The algorithm varies with type of target and four different but similar algorithms are shown.The algorithm operates on data covering one of a number of predefined landmark targets. A landmark target is primarily defined by a bounding box (min and max latitude/longitude) covering the target (e.g. the Titicaca Lake) with some margins. Four types of landmark targets (lakes - coastline database, ice shelves - SAR imagery, water way - SAR imagery, and mountain area targets - DEM database) are handled by the algorithm and the type of reference data and details of the algorithm varies with type of target.

MWI or ICI data with a geolocation within a predefined bounding box associated to a given target is gridded (upsampled) on a regular latitude and longitude grid (or using polar stereographic coordinates for targets located at high latitudes), having a finer resolution than that of the original data. This is done using a linear interpolation method, and this allows to extract more smooth features from the images. Features/contours from the gridded data are extracted using image processing filters, i.e. the Canny [9 edge [9] detection or  Sobel filter [13] depending on the type of target.

Target specific reference data within the predefined bounding box associated to a given target is then used to construct a reference contour at the same grid onto which MWI or ICI data was upsampled.

Images containing the extracted features/contours are then fed into a contour matching algorithm, where shifts in the image based on MWI or ICI data relative to the reference image is detected and estimated. Once the shift in the MWI/ICI image has been estimated it is a quite straightforward task to calculate the geolocation error.

A target (except for a waterway target) is also defined by a number of predefined coordinates within the bounding box in order to allow for a cloud filtering of data. A brightness temperature (BT) contrast of the current data/image is derived considering these coordinates, and in order to validate that the derived shift is acceptable a fuzzy-logic check is performed, considering both the value of the derived shift and that the BT contrast present in the image is sufficient.

Two geolocation errors are derived for a water way target, using two different grid resolutions, and the cloud filtering of data is done using a fuzzy logic check of the obtained geolocation errors.

### 2.1.1 Landmark targets

Landmark targets that can be used for validation of the geolocation of ICI/MWI data are listed in Table 2.1.1. These targets can be classified into three categories: lakes located at high altitude, ice shelves and mountain areas (see Games Report Task 1 for a more complete description of actual proposed landmark targets).

*Table 2.1.1: Summary of proposed targets, reference source and daily detectability*

| **Landmark target** | **Contour reference source** | **Detectability/day** |
|---|---|---|
| | **Northern hemisphere** | |
| Qinghai lake | GSHHG | 1 |
| Karakorum mountains | DEM | 1 |
| Hudson Bay | GSHHG | 1 |
| Nares Strait | SAR | 4-6 |
| | **Southern hemisphere** | |
| Ross Antarctic ice shelf | SAR | 4-6 |
| Filchner-Ronne Antarctic ice shelf | SAR | 4-6 |
| Amery Antarctic ice shelf | SAR | 3-5 |
| Titicaca lake | GSHHG | 1 |
| Andean mountains | DEM | 1 |

### 2.1.2 Criteria for landmark target selection

The physical basis for why the targets described in the preceding selection can be used for a validation purpose is here described. Considering a homogeneous isothermal (constant temperature and interaction parameters) atmospheric layer of thickness *H* with a small albedo (thus neglecting the multiple scattering contribution), it is possible to derive the analytical solution of the radiative transfer equation for the upwelling BT as follows:

$$= \quad ^{-(\quad)} + (1 - \quad)\,_0[1 - \quad ^{-(\quad)}] \tag{2.1.1}$$

where the symbols are

: (adim.)

: (K)

$_0 = (\ )$: (K)

: (km$^{-1}$)

: (adim.)

: ( = θ θ )(km)

$^{-(\quad)}$ =*t(L)*: *atmospheric transmittance* (adim.)

Considering two different close pixels $p_1$ and $p_2$ and assuming a similar atmospheric layer with the same transmittance $t(L)$, the BT contrast $\Delta T_B = T_B(p_1) - T_{B2}(p_2) = T_{B1} - T_{B2}$ can be written as follows:

$$\Delta \quad = \quad {}_{1} \; {}_{1} \; {}^{-(\quad)} + (1 - \quad) \; {}_{0}[1 - {}^{-(\quad)}] - {}_{2} \; {}_{2} \; {}^{-(\quad)} - (1 - \quad) \; {}_{0}[1 - {}^{-(\quad)}] \qquad (2.1.2)$$

thus

$$\Delta \quad = (\;)[ \; {}_{1} \; {}_{1} - {}_{2} \; {}_{2} \; ] \qquad (2.1.3)$$

Therefore, in order to have a sufficiently high BT contrast, from eq. (2.1.3) we can essentially consider areas with different surface emissivity and/or surface temperature, such as sea/lake/ice coastlines or mountain chains. In the latter case we have a surface temperature variability due to the height difference between plain and mountain as well as a different atmospheric optical thickness (i.e., transmittance of the mountain pixel larger than the plain one) entailed by the different heights of the pixels themselves. A further feature to play with is the natural variability of surface emissivity.

Taking into account these concepts, for landmark target search we have basically considered the following two major types:

a) surface water bodies (liquid or ice) sufficiently large (wrt satellite FOV);
b) mountain areas with strong slopes (altitude gradients) in relatively dry regions.

## 2.2 Target-contour matching block diagram and data flow

### 2.2.1 Input data

#### 2.2.1.1 Data

Within the GAMES project, the proposed target-contour matching (TCM) algorithm is shown in Figure 2.1.1. Input data to the TCM algorithm are the satellite radiometric imagery containing the landmark target, and reference data described in Table 2.2.1. The reference data to use depends on the type of target; SAR data is used for ice shelf and water way targets, GSHHG data is used for high altitude lake targets, and DEM data is used for mountain area targets.

*Table 2.2.1: Summary of proposed reference data. Note that a pre-processing of SAR data is needed and this is further described in Appendix A.*

| Reference source | Original source | Spatial resolution | Pre-processing needed |
|---|---|---|---|
| SAR | Level-1 GRD | 10-40 m | yes |
| GSHHG | GSHHG with full resolution | 40 m | no |
| DEM | GTOPO 30 | 30 arc seconds (~1 km) | no |

#### 2.2.1.2 Algorithm parameters

The parameters that defines a landmark target are the following:

- Minimum and maximum latitude/longitude covering the target
- A gridpoint spacing parameter used to create a regular grid covering the bounding box
- A number of coordinates within the bounding box to be used to extract the contrast of the image or a second gridpoint spacing parameter

The GAMES TCM algorithm varies somewhat with type of target, and hence algorithm parameters for extracting features/contours from MWI/ICI data and reference data varies with type of target, and these are:

- Canny edge detection parameters (lower and upper bound for hysteresis thresholding and standard deviation of the Gaussian filter) for extracting contours from MWI/ICI data for lake and ice shelf targets
- Canny edge detection parameters for extracting contours from SAR data for ice shelf targets

The Sobel filter is applied for mountains area targets and this filter takes no adjustable input parameters.

Input parameter to the algorithm used for the actual contour matching, between the MWI/ICI and the reference data, with sub-pixel precision is an upsampling factor. If this upsampling factor is set to e.g. 10 it will allow to detect shifts with a 0.1 pixel precision.

Three model parameter are used to define the fuzzy logic control of data and these are:

- a reference shift
- a reference contrast or a second reference shift
- a reference threshold for acceptance

### 2.2.2 Target contour matching algorithm

The target contour matching algorithm (TCM) is shown in Fig. 2.1.1. The TCM algorithm varies somewhat with type of target as described in the following subsections.

#### 2.2.2.1 Lake at high altitude target

The TCM algorithm using lake at high altitude target can be described by:

1) Extract MWI/ICI and GSHHG shoreline database data inside a predefined bounding box that contains the target
2) Apply a DEM correction of MWI/ICI data if desirable

3) Regrid MWI/ICI data onto a regular latitude and longitude grid covering the target bounding box
4) Apply Canny algorithm to extract a contour from the gridded MWI/ICI data
5) Project GSHHG shoreline database on the same grid obtained at step 3, using nearest neighbour approach.
6) Apply a contour matching algorithm on the two images containing the contours in order to derive (sub) pixel shifts
7) Convert these shifts to distance shift along latitude and longitude directions
8) Extract the BT contrast found in the MWI/ICI data
9) Fuzzy-logic control of data

#### 2.2.2.2 Mountains area target

The TCM algorithm using mountains area target can be described by:

1) Extract MWI/ICI and DEM data inside a predefined bounding box that contains the target
2) Apply a DEM correction of MWI/ICI data if desirable
3) Regrid MWI/ICI and DEM data onto a regular latitude and longitude grid covering the target bounding box
4) Apply Sobel filter on the DEM data to obtain the gradient magnitude
5) Apply Sobel filter to MWI/ICI to obtain the gradient magnitude
6) Follow step 6 to 9 of the of the lake at high altitude case, but applying the contour matching algorithm on the gradient magnitude images

#### 2.2.2.3 Ice shelf

The TCM algorithm using ice shelf target can be described by:

1) Extract MWI/ICI and SAR data inside a predefined bounding box that contains the target
2) Project MWI/ICI and SAR data on polar stereographic coordinates (x, y)
3) Regrid MWI/ICI onto a regular x and y grid covering the target bounding box
4) Apply Canny algorithm to extract a contour from the gridded MWI/ICI data
5) Apply Canny algorithm to extract SAR contour
6) Project SAR contour in the same grid obtained at step 3, using nearest neighbour approach
7) Follow step 6 to 9 of the of the lake at high altitude case

#### 2.2.2.4 Water way target

The GAMES TCM algorithm for a water way target (i.e. Nares Strait) is very similar to the one of the ice shelf targets. The difference is that the target of Nares Strait is defined by two grid spacing parameters but no coordinates for deriving a BT contrast. That is, the algorithm does all its calculations (step 3 to 7 of the Ice Shelf Target, excluding step 8 of the Lake at high altitude target) for two different grids and two different results are consequently obtained. The obtained results are only considered valid if both results are reasonably determined by fuzzy logic.

### 2.2.3 Output data

The output data of the TCM algorithm is mainly an estimated displacement error of the ICI/MWI image and a quality estimate describing if the result is useful or not. Other output variables are described below (note that the output depends to some extent on the type of target, e.g. a second_shift is only obtained for a water way target):

- **filename:** the name of the MWI/ICI Level1B file used
- **sensor:** the name of the sensor
- **channel**: the channel ID used
- **date**: a representative datetime of the measurement
- **lat_center**: latitude center [degrees] of the bounding box used
- **lon_center**: longitude center [degrees] of the bounding box used
- **shift_x**: the derived shift in validated data [km] along longitude direction
- **shift_y**: the derived shift in validated data [km] along latitude direction
- **shift**: the derived shift in validated data [km]
- **contrast**: contrast in image [K]
- **second_shift**: the second derived shift in validated data [km]
- **valid:** a validity flag of derived shift

## 2.3 Target contour matching algorithm details

### 2.3.1 Digital elevation model correction

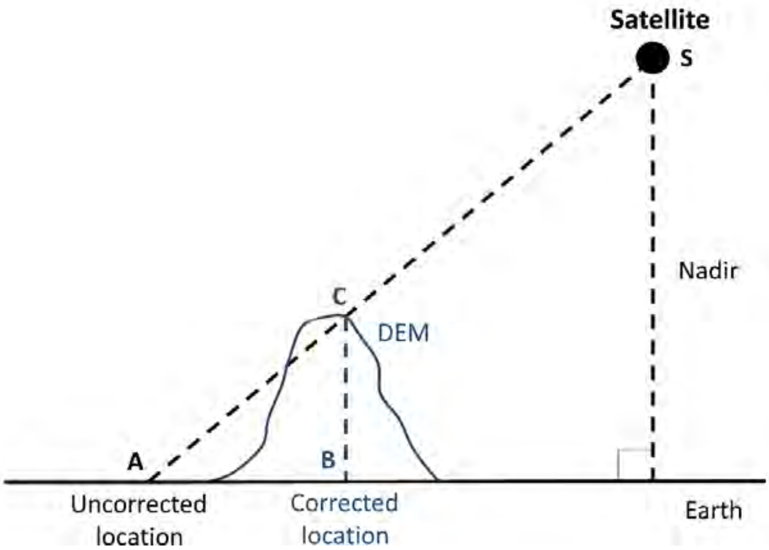



*Figure 2.3.1: Digital Elevation Model correction*

In case of high-altitude targets, we can have an initial error in the data as shown in Fig. 2.3.1, if the pixel positions of the original data has been estimated neglecting the topography of Earth. A DEM correction of data applied in the TCM algorithm, to correct the position of a given pixel, can be described by:

1) Reduce DEM data match the resolution of MWI/ICI data by a simple box blur filter (each pixel in the resulting DEM image has a value equal to the average value of its neighboring pixels)
2) Make a Delaunay triangulation of a set of points (latitude and longitude coordinates on a regular grid) covering the geoid surface of interest or the bounding box associated to the target (triangulation of points on a regular grid is in principal trivial)
3) Generate a set of triangles in ECEF (*earth-centered, earth-fixed*) Cartesian coordinates (x, y, z) from the triangulation and the DEM data
4) Apply the Möller-Trumbore [22] ray-triangle intersection algorithm to find the closest point to the sensor where the line of sight from the sensor towards Earth intersects with any of the triangles. To speed up this part of the algorithm, only triangles that have a centroid within a distance 30 km from the uncorrected position are considered. This simplification is safe to use for MWI/ICI observations as incidence angles are smaller than 55°, so it is impossible that the corrected positions should be more than 30 km away from the uncorrected position.

### 2.3.2 Polar stereographic coordinates

Extraction of contours from images associated with targets located at high latitudes is done using polar stereographic coordinates. At high latitude, this coordinate system is better suited than geodetic coordinates as directions can become complicated, with all geodetic north–south lines converging at the poles. The conversion back and forth between geodetic and polar Stereographic coordinates is done following the description given on page 161-162 in [23].

### 2.3.4 Feature extraction

Two different image processing filters are used to extract contour/features from the data, and these are the Canny edge detector and the Sobel filter. Prior to the feature extraction, MWI/ICI data is gridded onto a regular grid using a linear interpolation of the original data. It was found that using a grid with a gridpoint spacing of about 5 km is an adequate resolution to use. This is a coarser and finer resolution than that of the spacing of original MWI/ICI data in the across and along track, respectively, but finer than that of the footprints, and generally a coarser resolution than that of the reference data. Hence, it should be clear that the gridpoint spacing parameter is a compromise taking many different types of resolutions into account. Anyhow, it was found that a spacing of 5 km allows for extracting smooth contours from the data of concerns, at the same time as average results does not critically depend on the resolution used.

#### 2.3.4.1 Canny edge detector

The Canny edge detector [9] is an edge detection operator that uses a multi-stage algorithm to extract edges in images. It uses a filter based on the derivative of a Gaussian in order to compute the intensity of the gradients. The Gaussian reduces the effect of noise present in the image. Then, potential edges are thinned down to 1-pixel curves by removing non-maximum pixels of the gradient magnitude. Finally, edge pixels are kept or removed using hysteresis thresholding on the gradient magnitude. The Canny filter has three adjustable parameters:

- the width of the Gaussian (the noisier the image, the greater the width)
- low and high threshold for the hysteresis thresholding

The Canny edge detector is used for lakes at high and ice shelf targets within the TCM algorithm. An example is shown in Figure 2.3.4.1, that shows an image over the Qinghai lake on 2016/12/01 from SSMIS. Displayed in the figure is also the extracted contour using the Canny filter (width=$\sqrt{2}$, low=0.2, high=0.5) and the boundary between the lake and land from the GSHHG database. To obtain a reference contour the Canny filter is also applied on SAR data for ice shelf targets. Contours describing the boundary between the lake and land are directly available from the GSHHG database, and how these are processed, in order to be useful as reference data, is described in Sect. 2.3.4.3.

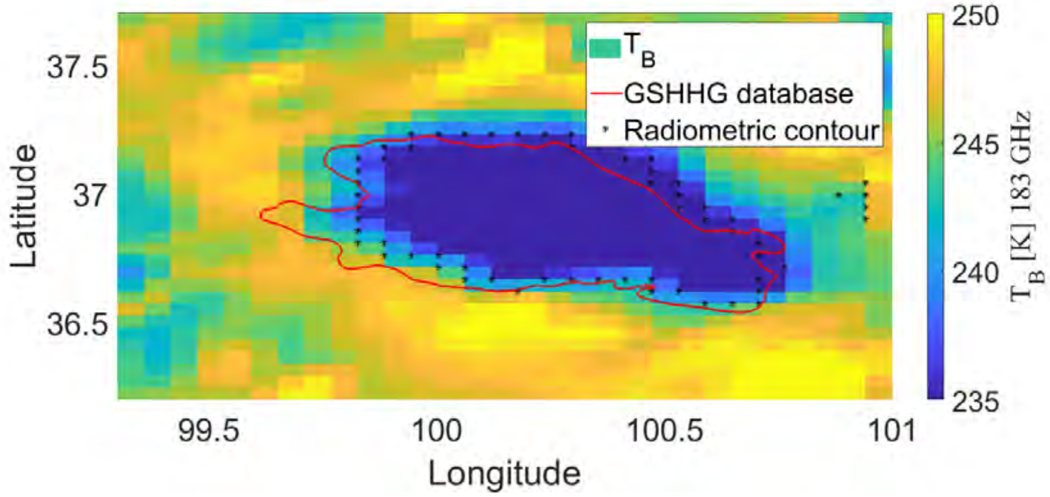



*Figure 2.3.4.1: Brightness temperature (BT) image at 183±7 GHz H over Qinghai lake from SSMIS F17 on 2016/12/01. The red line represents the lake coastlines from GSHHG database, described in Wessel and Smith (1996). Black markers indicate the extracted contour by Canny method.*

### 2.3.4.2 Sobel filter

The Sobel operator or Sobel filter [13] is a filter that computes an approximation of the gradient of the image intensity. The operator uses two 3×3 kernels which are convolved with the original image to calculate approximations of the derivatives, one for horizontal changes, and one for vertical changes. The Sobel filter has no adjustable parameter and the magnitude of the gradient is used as an image feature for mountains area targets (Figure 2.3.4.2) within the TCM algorithm. The Sobel filter is also applied on DEM data to obtain a reference image.

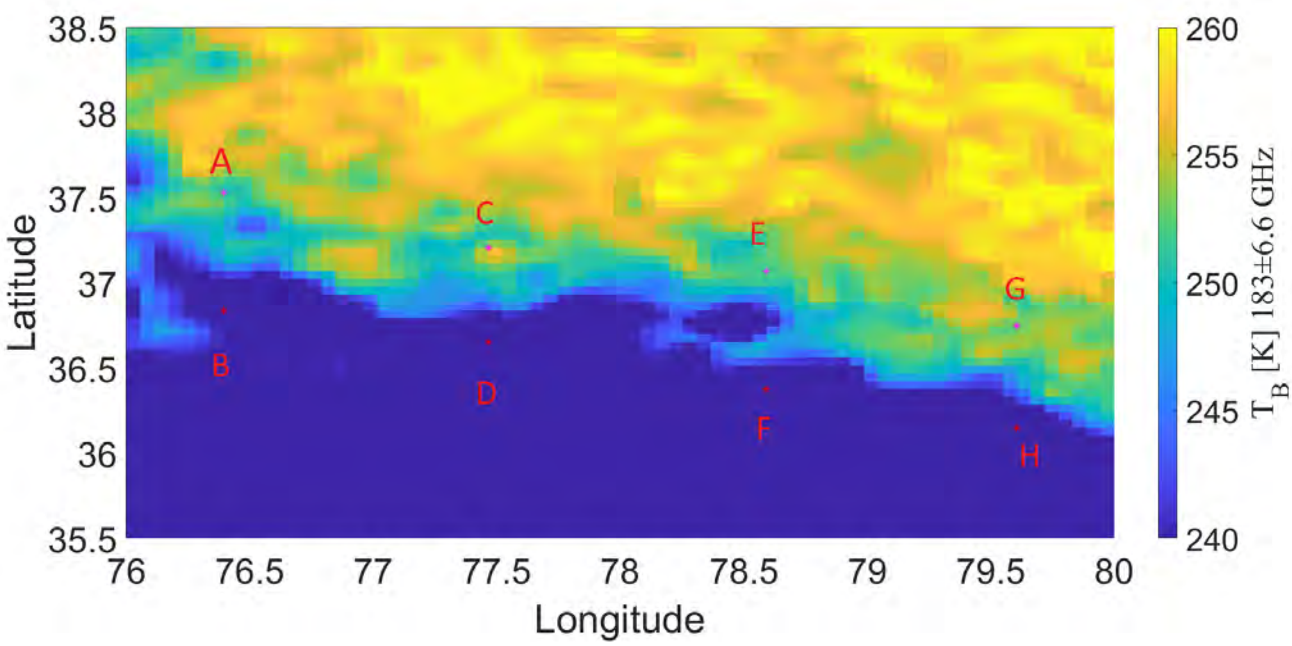



*Figure 2.3.4.2: Brightness temperature (BT) image at 183±6.6 GHz H over Karakorum mountains from SSMIS F17 on 2016/01/02. Eight points are those used to calculate the BT contrast along mountain chain*

#### 2.3.4.3 Gridding of contours

The Canny filter was described to be used to extract contour for lake at high altitude targets. The boundary between lake and land, that can be obtained from the GSHHG database, is used as the reference for contour matching. However, the contour extracted from MWI/ICI data is defined on grid points on a regular latitude and longitude grid, whereas the reference land and lake boundary is not. To allow for a contour matching the land and lake boundary must be defined on the same regular grid as is used for MWI/ICI data. This transformation is done using a simple approach, where an image with pixel values of zero or one is constructed. The closest image pixel of each point of the land and lake contour is given the value one.

A similar approach is used to construct reference data from SAR images for ice shelf targets. The Canny filter is used to extract contours using the original resolution of the SAR data. The obtained contour is then translated to the regular grid used for MWI/ICI data using the same approach as in the lake and land boundary case.

The approach used for gridding of contours introduces some error/noise in the reference contour, as the gridded reference contour can be misplaced by a distance up to half the resolution of the grid compared to the actual location. However, this applies to a single point of the grid, and it is unlikely that all points of the reference contour are biased with the same offset. In other words, the error should be random rather than systematic, and ultimately lead to a small random error in the contour matching.

### 2.3.5 Contour matching

The contours/features extracted from MWI/ICI and reference data are cross-correlated against each other to detect and estimate shifts between the data. This is done using an efficient subpixel image registration algorithm described in [17], or more exactly the algorithm referred to as the single-step discrete Fourier transform (DFT) approach in [17]. This algorithm provides an estimate of the column and row shifts of the features within the MWI/ICI image as compared to reference image. The algorithm uses selective upsampling by a matrix-multiply DFT to dramatically reduce computation time and memory without sacrificing accuracy. With this procedure all the image points are used to compute the upsampled cross-correlation in a very small neighborhood around its peak. In GAMES Report for Task1 results obtained using this algorithm is compared to an alternative image registration algorithm where the cross-correlation is calculated in the spatial domain. Differences were found to be negligible small, and it was concluded that the choice of contour matching algorithm is not a critical issue for TCM algorithm, and [17] was selected as the preferable algorithm to use, as it is widely used and well described in the literature. An implementation of the contour matching algorithm applied within the TCM is provided by the author of [17] on the Matlab file exchange site, or more exactly from

https://se.mathworks.com/matlabcentral/fileexchange/18401-efficient-subpixel-image-registration-by-cross-correlation

#### 2.3.5.1 Image index shift to distance

Once the pixel displacement in x and y (column and row) direction of the image is estimated, the corresponding latitudinal, longitudinal, and distance error can be determined as we know the position of the grid points / pixels of the image, and the resolution of the grid. The distance error is calculated from how much the center position of the grid must be moved to match the reference image, using a Haversine / great circle distance calculation, i.e. the geolocation error ϵ is the distance between the two coordinates

$$( \quad \_ \quad , \quad \_ \quad )$$

and

$$( \quad \_ \quad + \Delta \quad \cdot \Delta \; , \quad \_ \quad + \Delta \quad \cdot \Delta \; ),$$

where $\Delta$ and $\Delta$ is the derived sub-pixel column and row shift of the data, and $\Delta$ and $\Delta$ is the latitudinal and longitudinal grid spacing, respectively. The Haversine calculation applied uses the WGS84 radius corresponding to the mean of the two latitudes of concern.

### 2.3.6 Image contrast

For all targets, except Nares Strait, a brightness temperature (BT) contrast is calculated for each MWI/ICI image covering the target, in order to allow for a cloud filtering of data. The way the contrast is calculated is most easily understood by looking at Figure 2.3.4.2, where eight predefined target specific coordinates (A, B, …, H) are displayed, and the image contrast or $\Delta$ is calculated as:

$$\Delta \quad = \frac{(\;) - \;(\;) + \;(\;) - \;(\;) + \;(\;) - \;(\;) + \;(\;) - \;(\;)}{4}, \qquad (2.3.6)$$

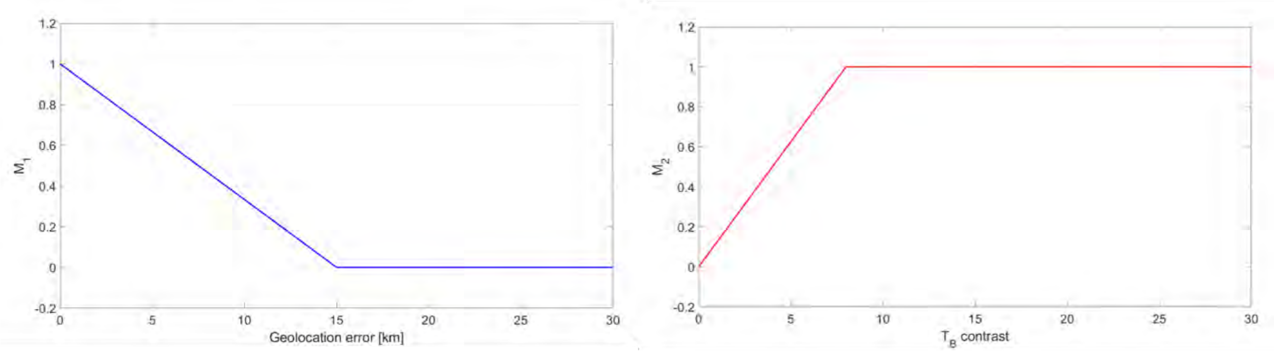
or as the mean of the difference between four sets of points. For some other targets, like the Titicaca Lake the contrast is calculated using five predefined coordinates, and the contrast is then calculates as:

$$\Delta \quad = \frac{(\;) - \;(\;) + \;(\;) - \;(\;) + \;(\;) - \;(\;) + \;(\;) - \;(\;)}{4}$$

### 2.3.7 Fuzzy-logic approach to target cloud-masking

A fuzzy-logic check is applied to verify if the derived data of the TCM algorithm is useful. The scene around the target may be cloud covered, and this can make the result useless, as the cloud coverage may mask the contrast due to the landmark target.

In the proposed fuzzy-logic approach, the idea is to use the estimated geolocation error and brightness temperature contrast of a specific target to decide if an overpass can be correctly used. For this purpose we use the membership functions $M_1$ and $M_2$, shown in Fig. 2.3.7. If the geolocation error is greater than a predefined maximum shift or the BT contrast is lower than a predefined minimum acceptable contrast, the membership functions are linearly weighted. Thresholds are, to some extent, arbitrarily or empirically defined mainly depending on the channel spatial resolution at ground.




$$M_1 = \begin{cases} linear & Error \leq Error_{threshold} \\ 0 & Error > Error_{threshold} \end{cases} \qquad M_2 = \begin{cases} linear & \Delta T_B \leq \Delta T_{B\ threshold} \\ 1 & \Delta T_B > \Delta T_{B\ threshold} \end{cases}$$

*Figure 2.3.7: Proposed function for fuzzy approach*

After the definition of the membership functions $M_1$ and $M_2$, the inference function *I*(*x*) is constructed by a multiplicative rule of the 2 membership functions:

$$(\ _1,\ _2) = \ _1(\ _1)\ \ _2(\ _2) \qquad (2.3.7.1)$$

where $_1$ and $_2$are arbitrary variables. Finally, an image can be used to evaluate the geolocation error if it satisfies the following defuzzification step:

$$(\ _1,\ _2) \geq \ _h\ \ _h \qquad (2.3.7.2)$$

where $I_{threshold}$ is typically set to 0.3 for all targets (see Appendix A, B and C in Games Report Task 1). After a sensitivity analysis using a dataset from SSMIS, the proposed inference function is:

$$\left(\Delta\qquad, \varepsilon\right) = \ _1(\varepsilon)\ \ _2(\Delta\qquad) \qquad (2.3.7.3)$$

where:

- ( ) = inference function
- Δ = mean BT contrast around target
- ε = geolocation error
- $_1(\varepsilon)$ = membership function depending on the geolocation error
- $_2\left(\Delta\qquad\right)$ = membership function depending on the BT contrast

The membership functions *M* and their parameters are provided in the Appendix A for high-altitude lake targets, in Appendix B mountain-chain targets and in Appendix for ice-shelf targets of Games Report Task 1. Note that for the water way target no BT contrast is calculated, but two different geolocation errors are derived and the proposed interference function is

$$\left(\epsilon_1, \epsilon_2\right) = \quad {}_1\left(\varepsilon_1\right) \quad {}_1\left(\varepsilon_2\right) \qquad (2.3.5.7)$$

## 3. RELATIVE GEOLOCATION BASED ON ATMOSPHERIC TARGETS

Higher frequency ICI channels, i.e. from ICI-5 to ICI-11 operating at a frequency close to or greater than 325.15 GHz, will likely not be able to observe any ground reference due to the strong gas absorption at those frequencies. Consequently, any classical geolocation assessment strategy making use of ground reference cannot be applied on those channels. To verify the geolocation of channels from ICI-5 to ICI-11 a possible strategy could be to use ICI-1 or ICI-4 as a reference channel, assuming that one of them is already geolocated, using some ground reference technique previously applied, as described in Sect. 2. Thus, the relative geolocation error of the test channels can be estimated comparing each single test channel with the selected geolocated reference channel. Of course, any geolocation error in the reference channel will propagate through test channels analysed, quantitatively affecting the final result.

Figure 3 describes the ICI observation geometry (conical scanner), and the figure also shows that at a given time the position of footprints on ground from the various channels will differ. The integration time for each individual sample is about 0.661045 ms, and this is shorter than the time period necessary to sweep out a single projected field of view across the scan. Consequently, we have several footprints overlapping each other for a given position and for each channel. This allows for a footprint matching procedure using the Bakus-Gilbert methodology by which we can produce a remapping of the original data for a given channel as it was observed in the view geometry of another reference channel. This is a processing step that Eumetsat will perform for ICI and MWI data (a Level2 product). A pointing error in one of the ICI channels will clearly show up in the Level1B data, and it is assumed that the error will also show up with a corresponding error in the remapped Level2 data. The remapped Level2 data is therefore used for the relative geolocation error assessment, as the data from two channels should be more directly comparable as they should "see" the same scene for a given pixel, although it is noted that the remapping itself can potentially introduce some errors.

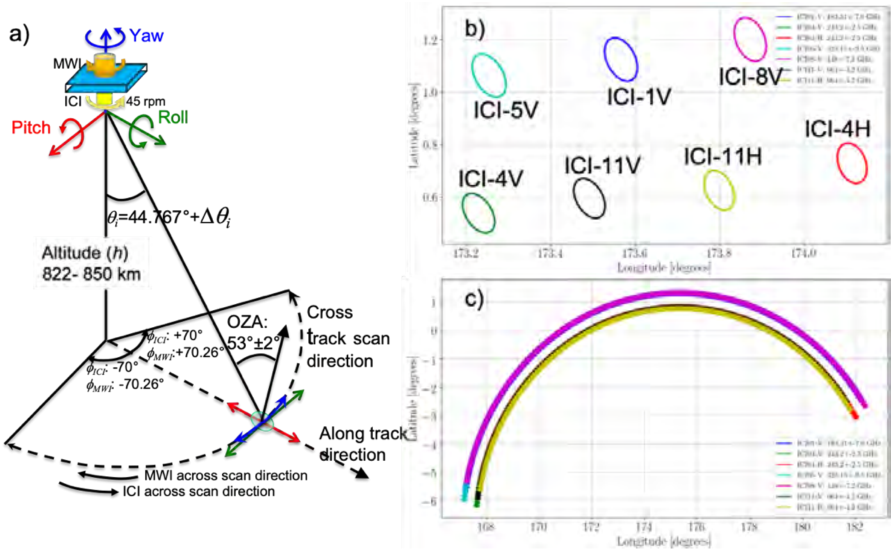



***Figure 3**: Panel a): Geometry of view for ICI and MWI. Panel b): Instantaneous, relative positions of -3 dB footprints on the geoid for some ICI channels. Panel c): relative positions of -3 dB footprints on geoid for some ICI channels and for every 5th cross track sample of a complete scan.*

## 3.1 Atmospheric targets definition

### 3.1.1 Criteria for atmospheric target selection

An obvious requirement is that the atmospheric target must generate a signal in the observation for both the reference and test channel. It was found (Games Report Task 2) that deep convective cloud (DCC) systems are a useful type of atmospheric target in this respect, as such systems generate a signal in all ICI channels, and the occurrence frequency is high enough to be useful for a validation purpose. A drawback with using the DCC system as an atmospheric target is that ICI is designed to observe clouds, and the various channels are sensitive to different features of such systems. Therefore, we can not expect that the correlation between data from two channels is perfect for observations around the DCC system, that would have been the ideal situation in terms of a relative geolocation error assessment.

## 3.2 Relative geolocation block diagram and data flow

### 3.2.1 Input data

#### 3.2.1.1 Data

Input data to the algorithm is remapped ICI / MWI data (a Level2 product as explained previously)

#### 3.2.1.2 Algorithm parameters

The algorithm takes three input parameters related to how bounding boxes covering DCC systems are constructed, and these are:

- BBOX_MIN_SIZE: (integer) # minimum number of pixel in each bounding box
- BBOX_EXPANSION_FACTOR: (float) # expansion factor to be applied to each bounding box
- MERGE_BBOXES: (boolean) # merges overlapping bounding boxes

The algorithm also uses MWI and ICI instrument parameters:

- Main tilt angle of the antenna
- Channel specific elevation offset angles

in order to allow for a retrieval of a relative pointing error.

### 3.2.2 Data flow

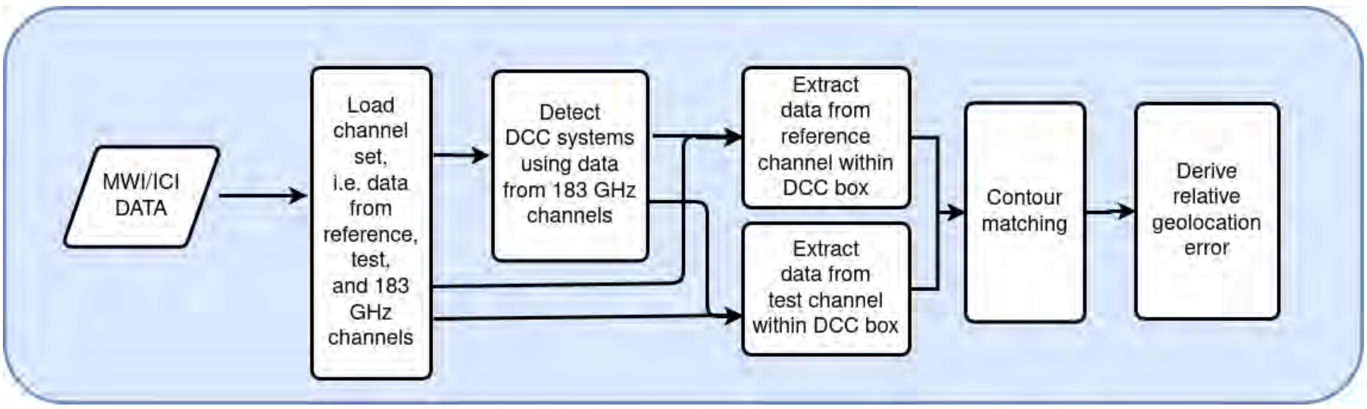



***Figure 3.3.2****: Block diagram of Open Loop Correlation approach to test the relative pointing error of ICI channel i-th with respect to a ICI reference channel for DCC targets. The contour matching is done using*

*subpixel image registration (Guizar-Sicairos et al., 2008) to find the best correlation between reference and tested* $T_A$.

Figure 3.3.2 gives an overview of the dataflow of the relative geolocation error retrieval algorithm. Remapped MWI or ICI data is the input data to the algorithm. Data from channels around 183 GHz is used to detect regions, or bounding boxes (min and max scan and sample numbers), covering DCC systems. Data for the reference and test channels within each derived bounding box is extracted and fed into a contour matching or *subpixel image registration* algorithm [17], where row and column (scan and sample) shifts are detected. The derivation of the relative geolocation error is then a straightforward geometrical task.

### 3.2.3 Output data

The main output of the algorithm is the estimated distance shift between data from two channels, and the corresponding error in terms of the elevation and azimuth view angle of the test channel, and a more complete description is found below:

- **level2_file**: the name of the Level2 file used
- **sensor**: the name of the sensor
- **reference_channel**: the reference channel ID used
- **test_channel**: the test channel ID used
- **scan_number_min**: minimum scan number of the bounding box
- **scan_number_max**: maximum scan number of the bounding box
- **sample_number_min**: minimum sample number of the bounding box
- **sample_number_max**: maximum sample number of the bounding box
- **delta_x_est_km**: estimated shift in the across-track direction [km]
- **delta_y_est_km**: estimated shift in the along-track direction [km]
- **azm_est_deg**: estimated error in azimuth viewing angle [degrees]
- **elv_est_deg**: estimated error in elevation viewing angle [degrees]
- **corrcoef**: correlation coefficient between the corrected data from the test channel and data from the reference channel

## 3.3 Relative geolocation algorithm details

### 3.3.1 Automatic detection of Region Of Interests (ROIs)

ROIs need to be defined for the relative geolocation assessment of ICI channels. For a practical implementation, the ROI definition needs to be automated. The approach followed to define the ROI for DCC targets is inspired by the method suggested by [25]. Hence, the criteria used to identify a DCC pattern from ICI, foresees the following check:

$$\Delta_{31} > \Delta_{32} > \Delta_{21} > 0 \quad (12)$$

where Δ is the difference in (K) between $T_A$ at $183\pm i$ GHz and $T_A$ at $183\pm j$ GHz. **Figure 3.3.1** (left) shows an example of the implementation of eq. (12) when it is applied on a simulated scenario (see Further in GAMES report Task2). The processing steps applied to derive ROIs or bounding boxes are the following:

- Create a DCC mask (i.e. 2-dimensional array (scan and sample number) where values are True if Eq. 12 is True)
- Apply a contour finding algorithm on the mask and convert obtained contours to bounding boxes, by taking the min and max scan and sample numbers of each contour
- Filter the bounding boxes, remove small bounding boxes (suggestion is to require at least 20 pixels in both the along and across-track direction)
- Expand the remaining bounding bounding boxes (suggestion expansion is 20 %)
- Merge overlapping bounding boxes

The process described above is visualized by Figure 3.3.1. The bounding boxes are enlarged by 20% with respect to their natural minimum size in order to have the chance to include the same DCC feature in each tested ICI channel that in principle can be affected by some geolocation error with respect to the reference channel (e.g. ICI-1). In the last step (right panel), the overlapping bounding boxes are merged together in an attempt to include those DCCs belonging to the same convective system. That is, if two bounding boxes overlap, a new larger bounding box that completely covers the two smaller bounding boxes is created and this one replaces the two smaller bounding boxes.

The derivation of bounding boxes is in principle a non-complex algorithm except for the contour finding part. For this purpose the contour finding algorithm from openCV (Open Computer Vision Library) is used, that is based on the algorithm presented in [25].

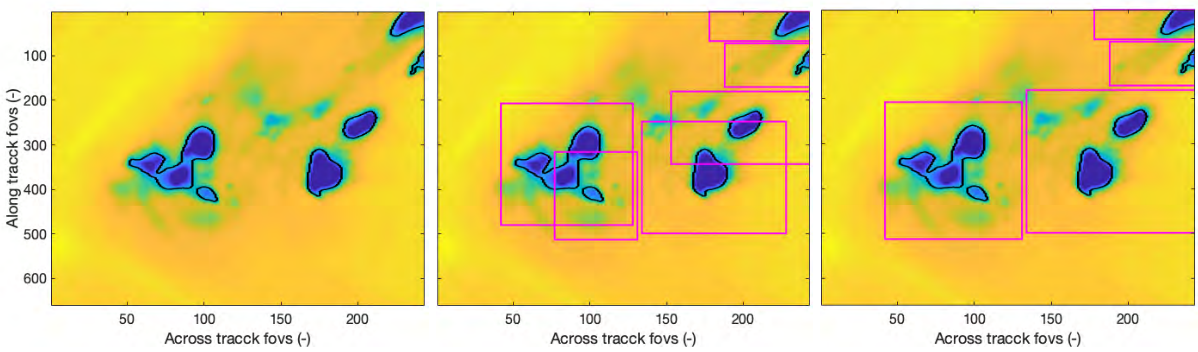



**Figure 3.3.1**: *Deep convective clouds (DCCs) detection are highlighted by black contours (left) of $T_A$ at 183±7 GHz (ICI-1). Some preliminarily regions of interest (ROIs) are defined as enlarged bounding boxes (magenta) around each DCC (middle). The overlapping bounding boxes are merged together thus defining the final ROIs (magenta lines on the right panel) used in input to CLC and OLC methods.*

### 3.3.2 Extraction of data within bounding box covering a DCC system

Bounding boxes covering DCC systems are derived following the description in the preceding section, and a bounding box is defined in terms of minimum and maximum scan and sample numbers.

Consequently, the extraction of data from the reference and test channel within a given bounding box is a trivial task.

### 3.3.3 Contour matching / Image registration and displacement retrieval

The registration of images, covering a DCC bounding box, from two ICI channels is done using an efficient subpixel image registration algorithm [17], that also is deployed for the landmark targets. But here no preprocessing image filter is applied, except from that the Fourier transform of the two images holding TB data are calculated prior to the actual image registration.

The algorithm provides image sub-pixel shifts (rows and columns or scan and sample number shifts) or $\Delta$ $\Delta$ of data from the test channel compared to the reference channel. That is, the estimated shift is the shift that results in the greatest correlation between the test and reference data. It is noted that for a given case, the estimated shift is not necessarily due to a pointing error of the test channel as it could also be related to the actual scene and difference in sensitivity of the two channels. However, biases that remain after averaging obtained results from many scenes are very likely to be due to pointing errors, since it is difficult to imagine that the mean shift is not close to zero if there is no relative pointing error between the channels.

#### 3.3.3.1 Quality control

The image registration algorithm applied provides a corrected/shifted version of the test image. The correlation coefficient of the corrected and the reference image is calculated and this measure is considered to give the opportunity to be used to discard situations with very low unphysical correlation coefficients.

#### 3.3.3.2 Displacement retrieval

The displacement of data from the test channel as compared to the reference channel is estimated as

$$\Delta \qquad = \Delta \cdot \Delta \qquad [\mathrm{km}]$$

$$\Delta \qquad = \Delta \cdot \Delta \qquad [\mathrm{km}]$$

where $\Delta$ and $\Delta$ is the sub-pixel sample and scan number shifts, respectively, as obtained from the contour matching algorithm, and$\Delta$ and $\Delta$ is the distance between samples in the across and along-track direction, respectively, of the bounding box center position and its neighbors, and this data is available from the input data as geodetic latitudes and longitudes are available in the Level2 data. The distance is estimated using the Haversine / great circle calculation using the WGS84 radius corresponding to the mean of the two latitudes of concern.

### 3.3.4 Displacement to pointing error conversion

An estimate of the pointing error, in terms of an azimuth and elevation offset angle, is given by the algorithm, and calculated as:

$$\Delta \quad = \Delta \qquad \cdot \quad (\theta) \, /$$

and

$$\Delta \quad = \Delta \qquad / \, ( \; / \quad ^{2}(\theta))$$

where is the altitude and θ is the test channel view angle.

## 4. CONCLUSION

Details of algorithms developed to allow for a geolocation error assessment of MWI and ICI data are presented in this document. Four similar algorithms using data for landmark targets as a reference that can be applied for data from channels where water vapor absorption does not mask out the sensitivity to the surface, and one algorithm for a relative geolocation error estimation method, primarily developed for the higher frequency channels of ICI, are described.

The algorithms developed for the landmark targets were previously tested on data from SSMIS, primarily for the channel operating around 183±6.6 GHz, and found to provide error estimates fitting with earlier independent studies. Hence, it is therefore likely that the algorithms will be useful for geolocation assessments of ICI/MWI data.

The algorithm developed for the relative geolocation error assessment has only been tested on a limited simulated dataset, giving that it can not be guaranteed that the method will provide useful error estimates for actual data.

## Appendix A. Preprocessing of SAR data

Sentinel-1 level-1 GRD products consist of focused SAR data that have been detected, multi-looked and projected to ground range using an Earth ellipsoid model. The Sentinel-1 GRD scene is composed of square pixels. To preprocess the data we use SNAP toolbox (https://step.esa.int/main/toolboxes/snap/) and this appendix describes a standard generic workflow to preprocess Sentinel-1 GRD data. In particular the necessary steps are listed in the following:

### A.1 Apply Orbit File

The orbit state vectors provided in the metadata of a SAR product are generally not accurate and can be refined with the precise orbit files which are available days-to-weeks after the generation of the product. The orbit file provides accurate satellite position and velocity information. The operation of applying a precise orbit available in SNAP allows the automatic download and update of the orbit state vectors for each SAR scene in its product metadata, providing an accurate satellite position and velocity information.

### A.2 Thermal Noise Removal

Sentinel-1 image intensity is disturbed by additive thermal noise and thermal noise removal reduces noise effects in the inter-sub-swath texture. In particular, normalizing the backscatter signal within the entire Sentinel-1 scene and resulting in reduced discontinuities between sub-swaths for scenes in multi-swath acquisition modes. The thermal noise removal operator available in SNAP for Sentinel-1 data can also re-introduce the noise signal that could have been removed during level-1 product generation, and update product annotations to allow for re-application of the correction [26]. Sentinel-1 level-1 products provide a noise look-up table (LUT), provided in linear power, for each measurement data set and used to derive calibrated noise profiles matching the calibrated GRD data.

### A.3 Border Noise Removal

While generating level-1 products, it is necessary to correct the sampling start time in order to compensate for the change of the Earth's curvature. At the same time, azimuth and range compression leads to radiometric artefacts at the image borders. The border noise removal algorithm [27], available as an operator in SNAP, was designed in order to remove low intensity noise and invalid data on scene edges.

### A.4 Calibration

Calibration is the procedure that converts digital pixel values to radiometrically calibrated SAR backscatter. The information required to apply the calibration equation is included within the Sentinel-1 GRD product; specifically, a calibration vector included as an annotation in the product allows simple conversion of image intensity values into sigma nought values. The calibration reverses the scaling factor applied during level-1 product generation, and applies a constant offset and a range-dependent gain, including the absolute calibration constant. Sigma nought specifies the strength of reflection in terms of the geometric cross section of a conducting sphere, and represents the radar cross section of a distributed target over that expected from an area of one square meter.

### A.5 Multilook Operator

Generally, a SAR original image appears speckled with inherent speckle noise. To reduce this inherent speckled appearance, several images are incoherently combined as if they corresponded to different looks of the same scene. This processing is generally known as multilook processing and we adopted 10 x 10 number of looks as processing parameters, obtaining a final spatial resolution of about 400 m x 400 m.

## BIBLIOGRAPHY

[1] P. Wessel and W. H. Smith, "A global, self-consistent, hierarchical, high-resolution shoreline database," Journal of Geophysical Research, vol. 101, no. B4, pp. 8741-8743, April 1996.

[2] W.E. Purdy et al., "Geolocation and pointing accuracy analysis for the WindSat sensor," IEEE Trans. Geosci. Remote Sens., vol. 44, no. 3, pp. 496-505, March 2006.

[3] 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 Trans. Geosci. Remote Sens., vol. 46, no. 4, pp. 913-922, April 2008.

[4] G. Heygster, H. Wiebe, G. Spreen , and L. Kaleschke, "AMSR-E geolocation and validation of sea ice concentrations based on 89 GHz Data," J. Remote Sens. Soc. Jpn., vol. 29, no. 1, pp. 226-235, January 2009.

[5] W. Berg, M. R. P. Sapiano, J. Horsman, and C. Kummerow, "Improved geolocation and Earth incidence angle information for a fundamental climate data record of the SSM/I sensors," IEEE Trans. Geosci. Remote Sens., vol. 51, no. 3, pp. 1501-1513, March 2013.

[6] H. Moradi, Meng, R. R. Ferraro, and S. Bilanow, "Correcting geolocation errors for microwave instruments aboard NOAA satellites," IEEE Trans. Geosci. Remote Sens., vol. 51, no. 6, pp. 3625-3637, June 2013.

[7] R. Bennartz, "On the use of SSM/I data in coastal regions. J. Atmos. Ocean Technol.", 16, 417-431, 1999.

[8] Y. Han, F. Z. Weng, X. L. Zou, H. Yang, and D. Scott, "Characterization of geolocation accuracy of Suomi NPP Advanced Technology Microwave Sounder measurements", J Geophys Res-Atmos, 121(9), 4933-495010.1002/2015jd024278, 2016.

[9] Canny, J., "A computational approach to edge detection, " IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 6, 679-698, November 1986.

[10] Juneja, M., & Sandhu, P. S. Performance evaluation of edge detection techniques for images in spatial domain". International journal of computer theory and Engineering, 2009, 1(5), 614.

[11] Vijayarani, S.; Vinupriya, M. "Performance analysis of canny and sobel edge detection algorithms in image mining". International Journal of Innovative Research in Computer and Communication Engineering, 2013, 1.8: 1760-1767.

[12] Podder, P., Parvez, A. M. S., Yeasmin, M. N., & Khalil, M. I. "Relative Performance Analysis of Edge Detection Techniques in Iris Recognition System". In 2018 International Conference on Current Trends towards Converging Technologies (ICCTCT), 2018, pp. 1-6. IEEE.

[13] Gupta, Samta, and Susmita Ghosh Mazumdar. "Sobel edge detection algorithm." International journal of computer science and management Research 2.2, 1578-1583, 2013

[14] Vincent, O. R., & Folorunso, O. "A descriptive algorithm for sobel image edge detection". In Proceedings of Informing Science & IT Education Conference (InSITE), 2009, Vol. 40, pp. 97-107, California: Informing Science Institute.

[15] R.H. Thomas, D.R. MacAyeal, D.H. Eilers, and D.R. Gaylord, 1984. Glaciological Studies on the Ross Ice Shelf, Antarctica, 1972-1978. The Ross Ice Shelf: Glaciology and Geophysics Antarctic Research Series, Volume 42, Paper 2, 21-53.

[16] J. P. Lewis, "Fast Template Matching", Vision Interface, p. 120-123, 1995.

[17] Manuel Guizar-Sicairos, Samuel T. Thurman, and James R. Fienup, "Efficient subpixel image registration algorithms," Opt. Lett. 33, pp. 156-158, 2008.

[18] 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 Trans. Geosci. Remote Sens., vol. 46, no. 4, pp. 913-922, April 2008.

[19] Kunkee, David B., et al. "Design and evaluation of the first special sensor microwave imager/sounder." IEEE Transactions on Geoscience and Remote Sensing 46.4, pp. 863-883, 2008.

[20] Prigent, C., Aires, F., Wang, D., Fox, S. and Harlow, C., 2016: Sea surface emissivity parameterization from microwaves to millimeter waves. Q.J.Roy. Meteorol. Soc. Accepted Author Manuscript. doi:10.1002/qj.2953

[21] Beck, H. E., Zimmermann, N. E., McVicar, T. R., Vergopolan, N., Berg, A., & Wood, E. F. "Present and future Köppen-Geiger climate classification maps at 1-km resolution. Scientific data", 5, 180214, 2018

[22] Möller, T., Trumbore, B, 1997: "Fast, Minimum Storage Ray-Triangle Intersection". *Journal of Graphics Tools*. **2**: 21–28. doi:10.1080/10867651.1997.10487468

[23] Snyder, J. P. 1987: "Map Projections - A Working manual", U.S. Geological Survey professional paper 1395 https://doi.org/10.3133

[24] Hong, G., Heygster, G., Miao, J., and Kunzi, K. 2005: "Detection of tropical deep convective clouds from AMSU-B water vapor channels measurements", *J. Geophys. Res.*, 110, D05205, doi:10.1029/2004JD004949.

[25] Satoshi Suzuki and others. "Topological structural analysis of digitized binary images by border following." *Computer Vision, Graphics, and Image Processing*, 30(1):32–46, 1985.

[26] SNAP Software, Help Document 2019. Available online: https://step.esa.int/main/toolboxes/snap (accessed on 1 May 2020).

[27] Guillaume, H. Masking "No-value" Pixels on GRD Products generated by the Sentinel-1 ESA IPF. ESA Tech. Rep, 2015. Reference MPC-0243, Issue 1.0.

# games

*Release 1.1*

The Games Developer Team

Jan 19, 2021

# CONTENTS:

CHAPTER

# ONE

# GAMES

## 1.1 Overview

This is GAMES Python package toolbox. GAMES is an acronym for Geolocation Assessment/validation Methods for EPS-SG ICI and MWI. The Ice Cloud Imager (ICI) and Microwave Imager (MWI) are two instruments that will perform meteorological observations from the polar orbit, within the EUMETSAT Polar System - Second Generation (EPS-SG), that will be in operation in the 2022-2043 timeframe. The GAMES Python package contains tools that will allow for making a geolocation assessment/validation of data from such instruments, and the toolbox has been developed inside an EUMETSAT funded activity. The GAMES toolbox has been developed using data from the Special Sensor Microwave Imager/Sounder (SSMIS) to test various algorithms, and the toolbox therefore also handles SSMIS data.

Geolocation validation methods using both various types of landmark targets as a reference source and correlative measurements of atmopsheric targets are covered by GAMES. The input data and algorithms applied to derive geolocation error in ICI and MWI data varies with the type of target used. A geolocation validation method using a specific type of target is denoted as a pipeline in this document. The GAMES package can be built in such a way that you can run these pipelines from a command line interface, given that required input data is provided, so even a non experienced Python user should be able to use GAMES.

### 1.1.1 Purpose of this document

The purpose of this doucment is to describe how to install and run the pipelines of GAMES and how to use the GAMES toolbox. A description of the basis of the algorithms used for the geolocation validation is found in a GAMES-ATBD, and is not covered in this document.

CHAPTER

TWO

# INSTALLATION

## 2.1 Building a GAMES Docker image

The GAMES package contains a build script that can be invoked in order to build a GAMES Docker image. This GAMES Docker image is portable, and GAMES pipelines (see Chapter 3) can be executed inside a Docker container through a command line interface (CLI) on a host machine that has the Docker Engine installed:

- a docker image is a non-changeable file containing libraries, source code, tools and other files needed to run applications.
- a docker Container is the run time instance of the image, and data files can be mounted into this container

Docker Engine is available for a variety of platforms and instructions how to install it is available here . The GAMES docker image can simply be built by:

```
$ ./build.sh
```

## 2.2 Installing requirements in a virtualenv

It is also possible to use the GAMES package outside the GAMES Docker image, but then you have to follow the installation guide below. Python packages should almost never be installed on the host Python environment, in order to avoid problems that can arise due to dependencies on different versions of packages. The requirements of the GAMES package are preferably installed in a virtualenv. A suitable virtualenv for the GAMES package can be created by first installing the package virtualenvwrapper on the host (so check that you are not in a virutalenv before installing):

```
$ sudo pip install virtualenvwrapper
```

Also add this to your shell startup file:

```
export WORKON_HOME=$HOME/.virtualenvs  # The virtualenvs are stored here.
export PROJECT_HOME=$HOME/Devel  # Location of your development project directories
source /usr/local/bin/virtualenvwrapper.sh
```

Then you can create a virtualenv named *games* by:

```
$ mkvirtualenv --python=/usr/bin/python3.8 games
```

and you can change to this envorinment by:

```
$ workon games
```

and if yoy want to change back:

```
$ deactivate
```

The dependecies of GAMES can then be collected from PyPI and installed by:

```
$ workon games
$ pip install -r requirements.txt
```

### 2.2.1 Running tests

The GAMES package contains tests for each of its modules, in order to facilitate further development. These tests can be executed using tox, which can be installed by:

```
$ pip install tox
```

and the tests can then be runned by:

```
$ tox
```

CHAPTER

THREE

# QUICKSTART

## 3.1 Running a pipeline using the GAMES Docker image

You can query the GAMES Docker image, in order to find out what pipelines that are available, by:

```
$ docker run --rm molflow/games --help

usage: games.sh [-h] {gshhg,gtopo30,sar,rpe}

Games Admin

positional arguments:
  {gshhg,gtopo30,sar,rpe,rpe}
                      The service to run. For more help on a particular
                      service do `SERVICE --help`

optional arguments:
  -h, --help            show this help message and exit
```

The help message above basically tells you that four different pipelines are available, and how you can proceed if you want to run one of those. For instance, if you now want to run the *gshhg* pipeline, you can ask for more help by:

```
$ docker run --rm molflow/games gshhg --help

  usage: games.sh gshhg [-h] [-g GSHHG_DIR] [-l LEVEL1B_DIR] [-t GTOPO_DIR]
                        [-o OUT_DIR] [-d] [-r] [-w] [-f UPSAMPLING_FACTOR]
                        [-a HIGH_THRESHOLD] [-b LOW_THRESHOLD] [-c SIGMA] [-v]
                        {ici,mwi,ssmis} channel level1b_files [level1b_files ...]
                        {qinghai,hudson,titicaca}

    Gelocation validation of microwave imager data using
    boundary between land and lake or ocean from the GSHHG
    database as reference data

positional arguments:
  {ici,mwi,ssmis}       Name of the sensor
  channel               ChannelID to be used for validation
  level1b_files         Level1B file(s)
  {qinghai,hudson,titicaca}
                        Target to be used for validation

optional arguments:
  -h, --help            show this help message and exit
  -g GSHHG_DIR, --gshhg-dir GSHHG_DIR
```

(continues on next page)

(continued from previous page)

```
                        Datadir where the GSHHG dataset is found, i.e. the
                        gshhg-shp-2.3.6/GSHHS_shp/ directory,
                        default value is /gshhg
  -l LEVEL1B_DIR, --level1b-dir LEVEL1B_DIR
                        Datadir where Level1B files for the sensor are found,
                        default value is /level1b
  -t GTOPO_DIR, --gtopo-dir GTOPO_DIR
                        Datadir where GTOPO30 dataset is found,
                        default value is /gtopo
  -o OUT_DIR, --out-dir OUT_DIR
                        Datadir for saving output, default value is /outdir
  -d, --demcorrection   Flag used to deterimine if a DEM correction, using the GTOPO30
                        data, shall be applied on imager data
  -r, --orthorectified  Flag used to determine if a reconstruction of the
                        orthorectified latitudes and longitudes shall be applied based
                        on correction data from the Level1B file. This option can only
                        be used for ICI and MWI data and not together with the
                        demcorrection flag
  -w, --storeresult     Flag for writing output to file
  -f UPSAMPLING_FACTOR, --upsampling-factor UPSAMPLING_FACTOR
                        Upsampling factor for image registration, default value is 10
  -a HIGH_THRESHOLD, --high-threshold HIGH_THRESHOLD
                        Upper bound for hysteresis thresholding in Canny edge filter,
                        default value is 0.5
  -b LOW_THRESHOLD, --low-threshold LOW_THRESHOLD
                        Lower bound for hysteresis thresholding in Canny edge filter,
                        default value is 0.2
  -c SIGMA, --sigma SIGMA
                        Standard deviation of the Gaussian filter applied in Canny
                        edge filter, default value is 1.4142135623730951
  -v, --verbose         Be verbose
```

From the description above you can see that this pipeline can run for three different sensors and three different targets. As an example, the pipeline can be run for SSMIS channel 9 data of a given Level1B file and using the Qinghai Lake (qinghai) as target reference by:

```
$ gshhg="/your/local/path/to/gshhg-shp-2.3.6/GSHHS_shp/"
$ gtopo30="/your/local/path/to/gtopodata/"
$ level1b="/your/local/path/to/smissdata/"
$ outdir="/your/local/path/to/where/you/want/results/"
$ level1bfile="CSU_SSMIS_FCDR_V01R01_F17_D20161221_S2348_E0130_R52279.nc"
$
$ docker run --rm \
    -v $gshhg:/gshhg -v $gtopp:/gtopo -v $level1b:/level1b -v $outdir:/outdir \
    molflow/games gshhg ssmis 9 qinghai $level1bfile \
    -g /gshhg -t /gtopo -l /level1b -r /outdir \
    --demcorrection  --storeresult --verbose
```

Note that a number of data sources are needed as input for the pipeline, and you need to mount data directories into the GAMES Docker container. This is what happens with the -v flag above. For example, the directory "/your/local/path/to/gshhg-shp-2.3.6/GSHHS_shp/" is mapped to a /gshhg directory of the Docker container, and you then use "-g /gshhg" to give this information to the *gshhg* pipeline. In practise, giving this information to the *gshhg* pipeline is not needed as the /gshhg is the default location.

The output of the pipeline is written to a file that can be found in the directory "/your/local/path/to/where/you/want/results/".

Note also that you can process many level1b files in single run, and this is recommended to do in order to save computation time, as this will avoid loading of DEM data (if you use the *demcorrection* flag) over and over again.

## 3.2 Running a pipeline in a virtualenv

The GAMES pipelines can also be runned in a virtualenv, but you need to run it from the src directory. The description of available pipelines can be obtained from:

```
games/src$ python3 -mgames --help
```

This will give you the same help as described in the preceeding section. You can then proceed and run the pipeline in a similar manner as described in the previous section, but there is no need to mount any directories.

## 3.3 Import modules from GAMES

You need to add the the path to the GAMES package to your PYTHONPATH by:

```
$ export PYTHONPATH=$PYTHONPATH:/your/local/path/to/games/src
```

in order to be able to import GAMES modules in the Python interpreter. Then you can import the qinghai module (main module for the "gshhg" pipeline) as:

```
>>> from games.utils import qinghai
```

More examples are given in the following chapters, including an API description (Chapter *GAMES API*) of available GAMES functions and methods.

### 3.3.1 Source code modification

If the GAMES package source code is modified:

- the modification will be applied directly if you run a pipeline in a virtualenv
- you should restart your Python interpreter, if you prior to the change have started a Python interpreter, in order to apply the modification
- you must rebuild the GAMES Docker image in order to apply the modification within the Docker image

CHAPTER

FOUR

# TUTORIALS

## 4.1 Available validation methods and targets

The GAMES toolbox contains geolocation validation methods using both landmark and atmospheric targets. These targets are:

- lake at high altitude or high latitude
- ice shelf, i.e. a large floating platform of ice that forms where a glacier or ice sheet flows down to a coastline and onto the ocean surface
- mountain area target
- deep convective clouds (DCC), a DCC system is composed of cumulonimbus type of clouds that can be many kilometers thick and with cloud tops in the upper part of the troposphere.
- waterway (i.e. Nares Strait, a waterway between Ellesmere Island and Greenland)

Three different type of methods, denoted as pipelines, are using landmark targets as a reference source, and one pipeline is using an atmospheric type of target. These pipelines, targets, and prefedined members, are listed below:

**Table 1**. GAMES pipelines/validations methods, type of target, and target members.

| Name of pipeline | Type of target | Target members |
|---|---|---|
| gshhg | lake at high altitude or coast-line at high latitude | Qinghai Lake, Titicaca Lake, and Hudson Bay |
| gtopo30 | mountains area | Andean Mountains and Karakorum mountains |
| sar | ice shelf and waterway | Ross Antarctic ice shelf, Filchner-Ronne Antarctic ice shelf, Amery Antarctic ice shelf, and Nares Strait |
| rpe | deep convective clouds | N/A |

The name of each pipeline for landmark targets is based on the main data source used for validating the imager data and input data is described in more details in a later section.

The GAMES landmark targets are prefedined inside a GAMES module called *targets* (see Section *targets*). The definition of a target, in GAMES, depends on which pipeline it belongs to. All targets are defined by a bounding box (min and max latitude and longitude) covering the target (see Section *utils* for details about bounding box objects), and a grid spacing parameter that can be used to create a regular 2-dimensional grid covering the bounding box. This grid will be used by the calculation within the pipeline. Furthermore, all targets, except “Nares Strait”, are also defined by a number of coordinates (five or eight) located within the bounding box. These five or eight coordinates are used for a cloud screening purpose, or more specifically to calculate the contrast of the imager data across the target, and the obtained result from running the pipeline can be considered invalid if the contrast is not high enough. The target of Nares Strait is defined by a second grid spacing parameter. That is, the *sar* pipeline will do all its calculations for two different grids for *Nares Strait*, and two different results are consequently obtained. The obtained results are only considered valid if both results are reasonable determined by fuzzy logic. An additional difference between the pipelines/targets is the coordinates system used. The *gshhg* and *gtop30* piplines are using geodetic latitude and longitude coordinates. The *sar* pipeline, with associated targets located at high latitudes, are using polar stereographic coordinates. In GAMES, a target object has an attribute named *grid* that is a *Grid* object, and this *Grid* object "knows" useful things about the grid (see Section *grid* for a full reference).

An example of how to load a target using the Python interpreter is given below:

```
>>> from games.utils import targets
>>> qinghai = targets.get_lake_target(targets.Lake.QINGHAI_LAKE)
>>> qinghai.bbox
BoundingBox(lat_min=36.2, lat_max=37.7, lon_min=99.3, lon_max=101)
>>> qinghai.grid
Grid(bbox=BoundingBox(lat_min=36.2, lat_max=37.7, ...), spacing=5.0)
```

The pipeline named *rpe* (relative pointing error) is using data influenced by deep convective clouds (DCC) for a validation purpose. The occurence of DCC is not stationary in space, and hence, there is no predefined region of interest or bounding box related to this type of target. Instead data that is to be validated, either from ICI or MWI, is used to identify bounding boxes covering DCC (more specifically data from three channels around 183 GHz are used for this purpose). This means that it is a bit more complicated to obtain these bounding boxes, as you need to load some data, but this can be done as described below:

```
>>> from games.utils.dcc_mask import get_dcc_bboxes
>>> from games.utils.sensor import Sensor, SensorType, ChannelICI
>>> from games.utils.ici_and_mwi_reader import load_channel_set
>>> # example showing how to get bounding boxes covering DCC systems
>>> BBOX_MIN_SIZE = 20  # minimum number of pixel in each bbox
>>> BBOX_EXPANSION_FACTOR = 1.2  # expansion factor to be applied to each bbox
>>> MERGE_BBOXES = True  # merges the overlapping boxes
>>> level1_file = "/full/path/to/the/file"
>>> sensor = Sensor.from_type(SensorType.ICI)
>>> ref_channel = sensor.get_channel(ChannelICI.ICI1)
>>> test_channel = sensor.get_channel(ChannelICI.ICI11V)
>>> # load data from a set of channels including the ones around 183 GHz
>>> channel_set = load_channel_set(
    level1_file, sensor, ref_channel, test_channel)
>>> # channel_set.dcc_channel_set contains data from the three 183 GHz channels
>>> bounding_boxes = get_dcc_bboxes(
        channel_set.dcc_channel_set,
        BBOX_MIN_SIZE,
        BBOX_EXPANSION_FACTOR,
        MERGE_BBOXES
>>> )
```

The return type of *get_dcc_bboxes*, in the example above, is a list containing *BoundingBoxDCC* objects, as it is possible that several DCC systems are detected. Note that a *BoundingBoxDCC* is a different type of bounding box object than used for the landmark targets. The *BoundingBoxDCC* object is using scan and sample number coordinates (see details in Section *dcc mask*).

## 4.2 Sensors and channels

The GAMES pipelines have been developed to handle data from the three sensors ICI, MWI, and SSMIS (the *rpe* pipeline only handles data from ICI and MWI), and this section gives an overview how these sensors and their associated channels are defined inside the GAMES package.

A sensor and channel is implemented as a *Sensor* and *Channel* object, respectively, in the GAMES toolbox (see Section *sensor* for a more detailed description). An attribute of a *Sensor* object is *channels*, that is a list of *Channel* objects. Both *Sensor* objects, representing ICI, MWI, and SSMIS, and associated *Channel* objects are predefined in the *sensor* module of GAMES. An example of how to load instances of the ICI *Sensor* and the ICI-1 *Channel* object in the Python interpreter is given below:

```
>>> from games.utils.sensor import Sensor, SensorType, ChannelICI
>>> sensor_ici = Sensor.from_type(SensorType.ICI)
>>> channel = sensor_ici.get_channel(ChannelICI.ICI1)
```

The attributes of a *Channel* object can be used to identify where associated data is located in a Level1B file, and example how to load data using the Python interpreter is given in the following section. The *Sensor* object has an attribute *channels_for_dcc_boxes*,

```
>>> sensor_ici.channels_for_dcc_boxes
(<ChannelICI.ICI1: 1>, <ChannelICI.ICI2: 2>, <ChannelICI.ICI3: 3>)
```

and these channels will be used by the *rpe* pipeline to detect DCC, as described in the previous section. The channel viewing angle is also described by the *Sensor* and *Channel* objects, and can be obtained as:

```
>>> sensor_ici.get_theta_deg(ChannelICI.ICI1)
43.9868718
```

A relevant usecase for the viewing angle is that it can be used for mapping a horizontal displacement of a point on ground, as seen by a sensor at a given altitude, into azimuth and elevation offset angles, e.g:

```
>>> from games.utils.validation import displacement_to_pointing_error
>>> sensor_altitude = 825.0 # km
>>> displacement_across = 1.0 # km
>>> displacement_along = 1.0 # km
>>> view_angle = sensor_ici.get_theta_deg(ChannelICI.ICI1)
>>> azimuth_offset, elevation_offset = displacement_to_pointing_error(
      displacement_across, displacement_along, view_angle, sensor_altitude)
>>> azimuth_offset, elevation_offset
(0.04996879166935218, 0.0359524930761808)
```

In the example above a function was imported from the GAMES *validation* module, and this module contains methods / functions that are responsible for the core calculation of the GAMES pipelines (see Section *validation* for interfaces),

## 4.3 Input data

Common input data to all of the pipelines is imager data, i.e. data from one of the three sensors ICI, MWI, and SMMIS (except for the *rpe* pipeline that only handles ICI and MWI data). It should be noted that the pipelines that are using landmark targets operate on Level1B data, whereas the *rpe* pipeline is using Level1C data. The meaning of Level1C data here is that data from the various channels of ICI or MWI have been remapped onto the “footprints” from one reference channel. This is a Level1C product that EUMETSAT intends to produce.

Other input data types varies between pipelines (see Table 2) and data sources handled by GAMES are:

- DEM (digital elevation model) data: GAMES only supports the use of GTOPO30 data, having a horizontal grid spacing of 30 arc seconds (approximately 1 kilometer)
- boundary between lake and land data: GAMES only supports the use of GSHHG data (Global Self-consistent, Hierarchical, High-resolution Geography Database), that is high-resolution (40 m) geography data set
- Level-1 Ground Range Detected (GRD) products consist of focused SAR data that has been detected, multi-looked and projected to ground range using an Earth ellipsoid model.

DEM data is used for two purposes, both for correcting the center position of samples (Level1B data) that originally was estimated neglecting the topography of Earth and for creating a reference image for mountain chain area targets.

SAR data, for monitoring of ice shelf targets, is used by the *sar* pipeline and it should be noted that the usage of the *sar* pipeline and SAR data is a bit demanding for the user:

- ice shelf edges are not stationary in time, and consequently the *sar* pipeline is using time varying reference SAR data, while the other pipelines are using static reference datasets.
- *sar* pipeline is using data based on Level-1 GRD SAR data as reference data, the SAR data must be preprocessed, outside the GAMES toolbox, prior to be used by the *sar* pipeline. The required preprocessing procedure is described in GAMES-ATBD.

This means that it is more demanding for the user to use the "sar" pipeline compared to the other pipelines. The user must provide the *sar* pipeline a preprocessed SAR file that covers the target of interest. A preprocessed SAR file can potentially be used for a validation purpose for a long time period (e.g a year or so), as the edges of the predefined ice shelf targets normally only vary slowly with time. However, iceberg calving can occur, and it is up to the user to keep track of ice shelf changes, and updating of the reference SAR file to use, accordingly. Additionally, the setting of Canny edge detection parameters, used by the *sar* pipeline is of particular importance for SAR data, as an optimal setting is image specific and is difficult to derive without actually inspecting obtained contours. This means that if the reference SAR file is updated for a target, you might need to tune the Canny edge detection parameters in order to obtain adequate edges.

**Table 2**. Input data to GAMES pipelines/validations methods.

| Name of pipeline | Input data |
|---|---|
| gshhg | ICI, MWI, or SSMIS Level1B data, DEM data, boundary between lake and land data |
| gtopo30 | ICI, MWI, or SSMIS Level1B data and DEM data |
| sar | ICI, MWI, or SSMIS Level1B data, and SAR data |
| rpe | ICI or MWI Level1C data |

Example of how to load input data relevant for running the *gshhg* pipeline for SSMIS channel 9 data is given below:

```
>>> from games.utils import targets
>>> from games.utils.sensor import Sensor, SensorType, ChannelSSMIS
>>> from games.utils import gshhs_reader
>>> from games.utils import gtopo30_reader
>>> from games.utils import ssmis_reader
>>> # example showing how to load some data
>>> gtopo_dir = "/path/to/gtopo30/data"
>>> gshhg_dir = "/path/to/gshhg/data"
>>> level1b_file = "/path/to/ssmis/level1b/file"
>>> target = targets.get_lake_target(targets.Lake.QINGHAI_LAKE)
>>> sensor = Sensor.from_type(SensorType.SSMIS)
>>> channel = sensor.get_channel(ChannelSSMIS.SSMIS9)
>>> dem = gtopo30_reader.get_dem(target.bbox, gtopo_dir)
>>> lakes_data = gshhs_reader.get_features(
        target.bbox,
        gshhs_reader.GshhsResolution.Full,
```

(continues on next page)

(continued from previous page)

```
        gshhs_reader.GshhsLayer.Lake,
        gshhg_dir
>>> )
>>> ssmis_data = ssmis_reader.get_data(level1b_file, channel, target.bbox)
```

Example of how to load SAR data relevant for running the *sar* pipeline and to qualitatively check your Canny edge detection setting is given bellow:

```
>>> import os
>>> import numpy as np
>>> import matplotlib.pyplot as plt
>>> from games.utils import targets
>>> from games.utils import sar_reader
>>> from games.utils.polar_stereo import polar_lonlat_to_xy
>>> from games.utils.validation import GeolocationValidator
>>> HIGH_THRESHOLD = 0.1
>>> LOW_THRESHOLD = HIGH_THRESHOLD * 0.4
>>> SIGMA = np.sqrt(2)
>>> sarfile = os.path.join(
>>>     "your/local/path/to",
>>>     "S1A_EW_GRDM_1SSH_20160724T111725_20160724T111829_012289_0131CA_1241_Orb_
↪Noise-Cor_Cal_ML.nc"
>>> )
>>> target = targets.get_ice_shelf_target(targets.IceShelf.ROSS_ANTARCTIC_ICE_SHELF)
>>> sar = sar_reader.get_data(sarfile, target.bbox)
>>> # project sar data on polar stereographic map
>>> bbox_xy = target.bbox.to_xy()
>>> x, y = polar_lonlat_to_xy(
>>>     sar.lon.flatten(), sar.lat.flatten(), lat_ts=bbox_xy.lat_ts
>>> )
>>> # extract sar contour
>>> geo = GeolocationValidator(target.grid)
>>> edges = geo.get_edges(
>>>     sar.sigma0_hh_filled,
>>>     LOW_THRESHOLD,
>>>     HIGH_THRESHOLD,
>>>     SIGMA,
>>>     True,
>>>     validate=False
>>> )
>>> # plot the data and check that the edges appear where you expect
>>> edge_filter = edges.flatten() == 1
>>> plt.scatter(x, y, s=5, c=np.log10(sar.sigma0_hh_filled))
>>> plt.plot(x.flatten()[edge_filter], y.flatten()[edge_filter], 'r.')
>>> plt.xlim([bbox_xy.lower_left.x, bbox_xy.lower_right.x])
>>> plt.ylim([bbox_xy.lower_left.y, bbox_xy.upper_left.y])
>>> plt.show()
```

The interfaces of available data import functions are further described in Sect. *Data import functions and objects*.

## 4.4 Pipelines

The four available pipelines of GAMES are defined in four different modules/scripts (each of them contains a CLI as described in the previous chapter):

- gtopo30: andean.py
- gshhg: qinghai.py
- rpe: relative_pointing_error.py
- sar: ross.py

Input data to the pipelines varies as described in the previous section, but the main task of each pipeline is to apply image processing filters on the imager data to be validated and on the pipeline specific reference data, in order to extract features from the data (such as a contour representing the boundary between land and lake). The extracted features from the imager and reference data are then cross-correlated in order to detect and estimate possible shifts in the imager data.

Even though the actual calculation performed for a given pipeline is rather complex, the pipeline scripts have been written to be readable (relatively easy to understand). Methods / functions that are responsible for the actual calculation are imported from a *validation* module of GAMES (see Section *validation* for interfaces).

How to import requried input for the *gshhg* pipeline was described in the preceeding section, and an example of how to run the last steps of the pipeline in the Python interpreter using this data is given below:

```
>>> from games.utils import qinghai
>>> from games.utils import dem_correction
>>> # preceeding section describes how to load data!
>>> DEM_REDUCTION_FACTOR = 11  # factor for downsampling DEM data
>>> out_dir = "/path/to/where/you/want/to/store/data"
>>> dem.reduce_dem(DEM_REDUCTION_FACTOR)
>>> corrected_pos = dem_correction.dem_correction(
        ssmis_data.samples_to_latlon(),
        ssmis_data.sat_to_latlonalt(),
        dem
>>> )
>>> ssmis_data.update_positions(corrected_pos)
>>> result = qinghai.validate(ssmis_data, lakes_data, target)
```

The *validate* function of the *qinghai* module is the core function of the pipeline, and the function returns an instance of a *ValidationResult* object described in the next section. Above, it can also be seen that a DEM correction of SSMIS data is performed prior to the validation. The code example below shows basically what happens inside the function *validate*:

```
>>> from games.utils.validation import GeolocationValidator, isvalid
>>> # preceeding section describes how to load data used below!
>>> # Uppsampling factor for image registration
>>> UPSAMPLING_FACTOR = 10
>>> # Canny edge detection parameters
>>> HIGH_THRESHOLD = 0.5
>>> LOW_THRESHOLD = HIGH_THRESHOLD * 0.4
>>> SIGMA = np.sqrt(2)
>>> # Fuzzy logic parameters for function isvalid
>>> CONTRAST0 = 8.  # [K]
>>> SHIFT0 = 15.  # [km]
>>> THRESHOLD = 0.3
>>> geo = GeolocationValidator(target.grid)
>>> # up-sample radiometric data
```

(continues on next page)

(continued from previous page)

```
>>> highres_tb = geo.griddata(
        ssmis.lon.flatten(), ssmis.lat.flatten(), ssmis.tb.flatten()
>>> )
>>> # Edge filter the data using a Canny algorithm
>>> edges = geo.get_edges(
        highres_tb, LOW_THRESHOLD, HIGH_THRESHOLD, SIGMA, True
>>> )
>>> # get lake contours on the high resolution grid
>>> lakes_highres = geo.features_to_meshgrid(lakes_data)
>>> # cross correlate and get shifts
>>> shift_idx, shift_idy, _ = geo.register_images(
        lakes_highres, edges, UPSAMPLING_FACTOR
>>> )
>>> # transform these shifts to distance errors
>>> shift_x, shift_y, shift = geo.grid.index_shift_to_distance(shift_idx, shift_idy)
>>> # get contrast
>>> contrast = geo.get_contrast(
        highres_tb, target.points, target.first_point_reference
>>> )
>>> # fuzzy logic check
>>> valid = isvalid(contrast, shift, CONTRAST0, SHIFT0, THRESHOLD)
```

The other GAMES pipelines in the list above can be run in a similar manner (see Section *Pipeline core functions* for an interface description), but examples are not included here. The reader is recommended to have a look in the pipeline scripts, if this is of interest.

## 4.5 Output data

The obtained result from running a pipeline is a *ValidationResult* object, that is written to a netcdf formatted file, if the pipeline is run through the CLI.

The content of this file, for the landmark target pipelines, is:

**Output file format for landmark target pipelines**

netcdf CSU_SSMIS_FCDR_V01R01_F17_D20160114_S1006_E1148_R47438_qinghai_ssmis_9 {

**dimensions**:

*date* = 1 ;

**variables**:

string filename(*date*) ;

description = *the name of the Level1B file* ;

string sensor(*date*) ;

description = *the name of the sensor* ;

int channel(*date*) ;

description = “the channel ID used” ;

double date(*date*) ;

description = *a representative datetime of the measurement* ;

units = *seconds since 2020-01-01 00:00:00.000* ;

string target(date) ;

description = *target ID* ;

float lat_center(*date*) ;

description = *latitude center of the bounding box used* ;

units = *degrees* ;

float lon_center(*date*) ;

description = *longitude center of the bounding box used* ;

units = *degrees* ;

float shift_x(*date*) ;

description = *the derived shift in validated data along longitude direction* ;

units = *km* ;

float shift_y(*date*) ;

description = *the derived shift in validated data along latitude direction* ;

units = *km* ;

float shift(*date*) ;

description = *the derived shift in validated data* ;

units = *km* ;

float second_shift(*date*) ;

description = *the second derived shift in validated data* ;

units = *km* ;

float contrast(*date*) ;

description = *contrast in image* ;

units = *K* ;

int valid(*date*) ;

description = *a validity flag of derived shift based on a fuzzy logic check: 0 = unvalid, 1 = valid* ;

int coverage_problem(*date*) ;

description = *data coverage problem: 0 = no problem, 1 = no imager data found within the target bounding box, 2 = no target data found within the target bounding box* ;

string sarfile(*date*) ;

description = *the name of the SAR file used* ;

int dem_correction(*date*) ;

description = *dem correction using GTOPO30 data flag: 0 = not applied, 1 = applied* ;

int dem_reduction_factor(*date*) ;

description = *factor for downsampling DEM data* ;

float dem_spacing(*date*) ;

description = *spacing of DEM data for triangulation* ;

units = *m* ;

int orthorectified(*date*) ;

description = *orthorectified flag: 0 = not applied, 1 = applied* ;

int upsampling_factor(*date*) ;

description = *upsampling factor for image registration* ;

float high_threshold_tb(*date*) ;

description = *upper bound for hysteresis thresholding in Canny edge filter used for imager data* ;

float low_threshold_tb(*date*) ;

description = *lower bound for hysteresis thresholding in Canny edge filter applied on imager data* ;

float sigma_tb(*date*) ;

description = *Standard deviation of the Gaussian filter inside the Canny edge filter applied on imager data* ;

float high_threshold_sar(*date*) ;

description = *upper bound for hysteresis thresholding in Canny edge filter applied on SAR data* ;

float low_threshold_sar(*date*) ;

description = *lower bound for hysteresis thresholding in Canny edge filter applied on SAR data* ;

float sigma_sar(*date*) ;

description = *Standard deviation of the Gaussian filter inside the Canny edge filter applied on SAR data* ;

}

The name of the output file has the format:

- {filename_no_extension}_{target}_{instrument}_{channel}.nc

In the Python interpreter, you can load the content of the file into a ValidationResult instance as:

```
>>> from games.utils.validation import ValidationResult
>>> outputfile = "/full/path/to/the/file.nc"
>>> data = ValidationResult.from_file(outputfile)
```

The content of the the output file for the *rpe* pipeline is:

**Output file format for *rpe* pipeline**

netcdf test_orbit4655_scene1_ICI_relative_error_sensor_ici_channel12 {

**dimensions**:

*bbox* = unlimited ;

*setting* = 1 ;

**variables**:

string level1_file(*bbox*) ;

description = *the name of the Level1 file used* ;

string sensor(*bbox*) ;

description = *the name of the sensor* ;

int reference_channel(*bbox*) ;

description = *the reference channel ID used* ;

int test_channel(*bbox*) ;

description = *the test channel ID used* ;

int scan_number_min(*bbox*) ;

description = *minimum scan number of the bounding box* ;

int scan_number_max(*bbox*) ;

description = *maximum scan number of the bounding box* ;

int sample_number_min(*bbox*) ;

description = *minimum sample number of the bounding box* ;

int sample_number_max(*bbox*) ;

description = *maximum sample number of the bounding box* ;

float delta_x_est_km(*bbox*) ;

description = *estimated shift in the across-track direction* ;

units = *km* ;

float delta_y_est_km(*bbox*) ;

description = *estimated shift in the along-track direction* ;

units = *km* ;

float azm_est_deg(*bbox*) ;

description = *estimated error in azimuth viewing angle* ;

units = *degrees* ;

float elv_est_deg(*bbox*) ;

description = *estimated error in elevation viewing angle* ;

units = *degrees* ;

float corrcoef(*bbox*) ;

description = *correlation coefficient between the corrected data from the test channel and data from the reference channel* ;

int upsampling_factor(*setting*) ;

description = *upsampling factor for image registration* ;

int bbox_min_size(*setting*) ;

description = *minimum size of bounding boxes to keep, both width and height must be greater or equal to min_size* ;

float bbox_expansion_factor(*setting*) ;

description = *expansion factor applied to each bounding box* ;

int merge_bboxes(*setting*) ;

description = *merge overlapping bounding boxes flag: 0 = not applied, 1 = applied* ;

}

and can be loaded as

```
>>> from games.utils.relative_pointing_error import load_result
>>> outputfile = "/full/path/to/the/file.nc"
>>> data = load_result(outputfile)
```

CHAPTER

FIVE

# GAMES API

This section contains automatically generated reference documentation.

## 5.1 Utils

### 5.1.1 coordinates

coordinates.py: Coordinate definitions

**class** games.utils.coordinates.**ECEF**(*x*, *y*, *z*)

Bases: object

Object for ECEF coordinates.

**Parameters**

- **x** (Union[float, ndarray]) – x-coordinate(s) [m]
- **y** (Union[float, ndarray]) – y-coordinate(s) [m]
- **z** (Union[float, ndarray]) – z-coordinate(s) [m]

**as_array**()

**Return type** array

**class** games.utils.coordinates.**LatLon**(*lat*, *lon*)

Bases: object

Object for geodetic coordinates.

**Parameters**

- **lat** (Union[float, ndarray]) – latitude(s) [degrees]
- **lon** (Union[float, ndarray]) – longitude(s) [degrees]

**property coordinates**

**Return type** Tuple[ndarray, ndarray]

**class** games.utils.coordinates.**LatLonAlt**(*lat*, *lon*, *alt*)

Bases: *games.utils.coordinates.LatLon*

Object for geodetic coordinates.

**Parameters** **alt** (Union[float, ndarray]) – altitude(s) [m]

**property coordinates**

**Return type** `Tuple[ndarray, ndarray]`

**class** `games.utils.coordinates.`**`XY`**(*x*, *y*)

Bases: `object`

Object for polar stereographic coordinates.

**Parameters**

- **x** (`Union[float, ndarray]`) – x-coordinate(s) [m]
- **y** (`Union[float, ndarray]`) – y-coordinate(s) [m]

**property coordinates**

**Return type** `Tuple[ndarray, ndarray]`

### 5.1.2 ecef

ecef.py Convert between lat/lon and earth-centered/earth-fixed coordinates.

**Notes: This function assumes the WGS84 model.** Latitude is customary geodetic (not geocentric).

**Source: "Department of Defense World Geodetic System 1984"** Page 4-4 National Imagery and Mapping Agency Last updated June, 2004 NIMA TR8350.2

Implemented based on original by: Michael Kleder

`games.utils.ecef.`**`ecef_to_latlonalt`**(*coords*)

Convert earth-centered earth-fixed (ECEF) cartesian coordinates to latitude, longitude, and altitude.

Inputs may be scalars, vectors, or matrices of the same size and shape. Outputs will have that same size and shape.

**Parameters** **coords** (*ECEF*) – ECEF coordinates object having attributes x: ECEF X-coordinate [m] y: ECEF Y-coordinate [m] z: ECEF Z-coordinate [m]

**Returns:**

**LatLonAlt coordinates object with** lat: geodetic latitude [degrees] lon: longitude [degrees] alt: altitude above WGS84 ellipsoid [m]

**Return type** *LatLonAlt*

`games.utils.ecef.`**`latlonalt_to_ecef`**(*coords*)

Convert latitude, longitude, and altitude to earth-centered, earth-fixed (ECEF) cartesian

Inputs may be scalars, vectors, or matrices of the same size and shape. Outputs will have that same size and shape.

**Parameters** **coords** (*LatLonAlt*) – LatLonAlt coordinates object with attributes lat: geodetic latitude [degrees] lon: longitude [degrees] alt: altitude above WGS84 ellipsoid [m]

**returns:**

**ECEF coordinates object with attributes** x: ECEF X-coordinate [m] y: ECEF Y-coordinate [m] z: ECEF Z-coordinate [m]

**Return type** *ECEF*

### 5.1.3 polar stereo

polar_stereo.py Convert back and forth between Polar Stereographic (x, y) coordinates and geodetic longitude and latitude following Map Projections - A Working manual - by J.P. Snyder. 1987

`games.utils.polar_stereo.`**`polar_lonlat_to_xy`**(*lon*, *lat*, *lat_ts*, *lon_0=0.0*, *re=6378137*, *e=0.0818191908426215*)

Convert from geodetic longitude and latitude to Polar Stereographic (X, Y) coordinates in m following Map Projections - A Working manual - by J.P. Snyder. 1987

**Parameters**

- **lon** (`array`) – longitude array in degrees
- **lat** (`array`) – latitude array [degrees]
- **lat_ts** (`float`) – true-scale latitude [degrees]
- **lon_0** (`float`) – meridian along positive Y axis
- **re** (`float`) – Earth radius [m]
- **e** (`float`) – Earth eccentricity

**Returns:** two-element tuple of numpy arrays containing (X, Y) in m

**Return type** `Tuple[array, array]`

`games.utils.polar_stereo.`**`polar_xy_to_lonlat`**(*x*, *y*, *lat_ts*, *lon_0=0.0*, *re=6378137*, *e=0.0818191908426215*)

Convert from Polar Stereographic (x, y) coordinates to geodetic longitude and latitude following Map Projections - A Working manual - by J.P. Snyder. 1987

**Parameters**

- **x** (`array`) – X coordinate(s) in m
- **y** (`array`) – Y coordinate(s) in m
- **lat_ts** (`float`) – true-scale latitude in degrees
- **lon_0** (`float`) – meridian along positive Y axis
- **re** (`float`) – Earth radius in m
- **e** (`float`) – Earth eccentricity

  **Returns:** two-element tuple of numpy arrays containing (longitude, latitude).

**Return type** `Tuple[array, array]`

### 5.1.4 utils

utils.py Bounding box objects for latitude/longitude and polar stereographic coordinates.

**class** `games.utils.utils.`**`BoundingBox`**(*lat_min*, *lat_max*, *lon_min*, *lon_max*)

Bases: `object`

Bounding box using latitude/longitude coordinates.

**Parameters**

- **lat_min** (`float`) – min latitude coordinate [degrees]
- **lat_max** (`float`) – max latitude coordinate [degrees]

- **lon_min** (float) – min longitude coordinate [degrees]
- **lon_max** (float) – max longitude coordinate [degrees]

**property as_tuple**

**Return type** Tuple[float, float, float, float]

**property center**

**Return type** *LatLon*

**in_bbox**(*point*)

**Return type** bool

**intersect**(*other*)

**Return type** bool

**property lat_extension**

**Return type** float

**property lat_extension_km**

**property lon_extension**

**Return type** float

**property lon_extension_km**

**property lower_left**

**Return type** *LatLon*

**lower_left_xy**(*lat_ts*)

**Return type** *XY*

**property lower_right**

**Return type** *LatLon*

**lower_right_xy**(*lat_ts*)

**Return type** *XY*

**to_xy**()

Returns a bounding box specifying x_min, x_max, y_min, ymax in Polar Stereographic (x, y) coordinates. Note that this bounding box covers a larger area than the lat/lon bounding box due to the difference between the two coordinates systems.

**Return type** *BoundingBoxXY*

**property upper_left**

**Return type** *LatLon*

**upper_left_xy**(*lat_ts*)

**Return type** *XY*

**property upper_right**

**Return type** *LatLon*

**upper_right_xy**(*lat_ts*)

**Return type** *XY*

**validate_points**(*points*)
raises if any of the point is oustide the bounding box

**Return type** None

**class** games.utils.utils.**BoundingBoxXY**(*x_min*, *x_max*, *y_min*, *y_max*, *lat_ts*)
Bases: object

BoundingBox using Polar Stereographic (x, y) coordinates. Can be initialised by the bounding box using latitude/longitude coordinates (see to_xy method of BoundingBox).

**Parameters**

- **x_min** (float) – min x coordinate [m]
- **x_max** (float) – max x coordinate [m]
- **y_min** (float) – min y coordinate [m]
- **y_max** (float) – max y coordinate [m]
- **lat_ts** (float) – true-scale latitude [degrees]

**property center**

**Return type** *XY*

**in_bbox**(*point*)

**Return type** bool

**property lower_left**

**Return type** *XY*

**property lower_right**

**Return type** *XY*

**property upper_left**

**Return type** *XY*

**property upper_right**

**Return type** *XY*

**validate_points**(*points*)
raises if any of the point is oustide the bounding box

**Return type** None

**property x_extension_km**

**Return type** float

**property y_extension_km**

**Return type** float

### 5.1.5 distance

distance.py Calculate distance between lat/lon pairs.

`games.utils.distance.get_distance(orig, dest)`

Get Haversine distance in km between lat/lon pairs.

**Parameters**

- **orig** (*LatLon*) – coordinates of origin(s) [degrees]
- **dest** (*LatLon*) – coordinates of destination(s) [degrees]

**Returns:** distance(s) between pairs [km]

**Return type** `Union[float, ndarray]`

`games.utils.distance.get_distance_xy(p1, p2, lat_ts)`

Returns the distance in km between two points in Polar Stereographic (x, y) coordinates.

**Parameters**

- **p1** (*XY*) – Polar Stereographic coordinate
- **p2** (*XY*) – Polar Stereographic coordinate
- **lat_ts** (`float`) – true-scale latitude in degrees

**Returns:** distance (float) in km between the two points

**Return type** `float`

### 5.1.6 grid

grid.py Grid objects with handy methods for bounding box objects using geodetic or polar stereographic coordinates.

`class games.utils.grid.Grid(bbox, spacing)`

Bases: `object`

Grid object for a BoundingBox using latitude/longitude coordinates.

**Parameters**

- **bbox** (*BoundingBox*) – a bounding box using latitude/longitude coordinates
- **spacing** (`float`) – grid spacing [km]

`closest_ids(lat, lon)`

**Return type** `Tuple[int, int]`

`closest_lat_id(lat)`

**Return type** `int`

`closest_lon_id(lon)`

**Return type** `int`

`property delta_lat`

**Return type** `float`

`property delta_lon`

**Return type** `float`

**index_shift_to_distance**(*shift_idx*, *shift_idy*)

Get index shifts as an error in distance.

**Parameters**

- **shift_idx** (`float`) – index shift in longitude direction
- **shift_idy** (`float`) – index shift in latitude direction

Returns:

a tuple of three shifts [km] or signed error in longitude direction, latitude direction, and total shift

**Return type** `Tuple[float, float, float]`

**property lats**

**Return type** `array`

**property lons**

**Return type** `array`

**property meshgrid**

**Return type** `Tuple[ndarray, ndarray]`

**property n_lats**

**Return type** `int`

**property n_lons**

**Return type** `int`

**validate_x**(*x*)

raises if x has inconsistent shape

**Return type** `None`

**property zeros**

**Return type** `ndarray`

**class** `games.utils.grid.`**GridXY**(*bbox*, *spacing*)

Bases: `object`

Grid object for a BoundingBox using Polar Stereographic coordinates.

**Parameters**

- **bbox** (*BoundingBoxXY*) – a bounding box using Polar Stereographic coordinates
- **spacing** (`float`) – grid spacing [km]

**closest_ids**(*y*, *x*)

**Return type** `Tuple[int, int]`

**closest_x_id**(*x*)

**Return type** `int`

**closest_y_id**(*y*)

**Return type** `int`

**property delta_x**

**Return type** `float`

**property delta_y**

**Return type** `float`

**index_shift_to_distance**(*shift_idx*, *shift_idy*)

Get index shifts as an error in distance.

**Parameters**

- **shift_idx** (`float`) – index shift in longitude direction
- **shift_idy** (`float`) – index shift in latitude direction

Returns:

a tuple of three shifts [km] or signed error in longitude direction, latitude direction, and total shift

**Return type** `Tuple[float, float, float]`

**property meshgrid**

**Return type** `Tuple[ndarray, ndarray]`

**property n_xs**

**Return type** `int`

**property n_ys**

**Return type** `int`

**validate_x**(*x*)

raises if x has inconsistent shape

**Return type** `None`

**property xs**

**Return type** `array`

**property ys**

**Return type** `array`

**property zeros**

**Return type** `ndarray`

**exception** games.utils.grid.**InputError**

Bases: `Exception`

**args**

**with_traceback**()

Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

### 5.1.7 targets

targets.py: Definitions of gelocation validation targets for GAMES pipelines

**class** games.utils.targets.**IceShelf**(*value*)

Bases: enum.Enum

An enumeration.

**AMERY_ANTARCTIC_ICE_SHELF = 'amery'**

**FILCHNER_RONNE_ANTARCTIC_ICE_SHELF = 'filchner'**

**ROSS_ANTARCTIC_ICE_SHELF = 'ross'**

**class** games.utils.targets.**IceShelftTarget**(*members*)

Bases: object

**Parameters members** (Dict[*IceShelf*, *TargetXY*]) – dict holding GAMES ice shelf targets

**class** games.utils.targets.**Lake**(*value*)

Bases: enum.Enum

An enumeration.

**HUDSON_BAY = 'hudson'**

**QINGHAI_LAKE = 'qinghai'**

**TITICACA_LAKE = 'titicaca'**

**class** games.utils.targets.**LakeTarget**(*members*)

Bases: object

**Parameters members** (Dict[*Lake*, *TargetLayer*]) – dict holding GAMES lake at high altitude targets

**class** games.utils.targets.**Mountain**(*value*)

Bases: enum.Enum

An enumeration.

**ANDEAN_MOUNTAINS = 'andean'**

**KARAKORUM_MOUNTAINS = 'karakorum'**

**class** games.utils.targets.**MountainTarget**(*members*)

Bases: object

**Parameters members** (Dict[*Mountain*, *Target*]) – dict holding GAMES mountains area targets

**class** games.utils.targets.**Target**(*id*, *bbox*, *points*, *spacing*)

Bases: *games.utils.targets.TargetBase*

Target class for mountains area targets.

**Parameters**

- **id** (Union[*Lake*, *Mountain*]) – id of target
- **bbox** (*BoundingBox*) – a bounding box object
- **points** (List[*LatLon*]) – a list of latitide/longitude coordinates within the bounding box to be used to determine the contrast in an image
- **spacing** (float) – spacing [km] for grid covering the bounding box

**property first_point_reference**
True if five points are used

**Return type** `bool`

**property grid**

**Return type** *Grid*

**class** games.utils.targets.**TargetBase**(*points*)
Bases: `object`

Target base object.

**Parameters** **points** (Union[List[*LatLon*], List[*XY*]]) – a list of coordinates (five or eight) to be used to determine the contrast in an image

**property first_point_reference**
True if five points are used

**Return type** `bool`

**class** games.utils.targets.**TargetLayer**(*id*, *bbox*, *points*, *spacing*, *layer*)
Bases: *games.utils.targets.Target*

Target class for lake targets.

**Parameters** **layer** (*GshhsLayer*) – layer of GSHHS data to use

**property first_point_reference**
True if five points are used

**Return type** `bool`

**property grid**

**Return type** *Grid*

**class** games.utils.targets.**TargetSpacing**(*id*, *bbox*, *spacing1=5.0*, *spacing2=6.0*)
Bases: `object`

Target class for water way targets.

**Parameters**

- **id** (*WaterWay*) – id of target
- **bbox** (*BoundingBox*) – a bounding box object
- **spacing1** (`float`) – spacing [km] for grid covering the bounding box
- **spacing2** (`float`) – a second spacing [km] for grid covering the bounding box

**property grid1**

**Return type** *GridXY*

**property grid2**

**Return type** *GridXY*

**class** games.utils.targets.**TargetXY**(*id*, *bbox*, *points*, *spacing=5.0*)
Bases: *games.utils.targets.TargetBase*

Target class for ice shelf targets.

**Parameters**

- **id** (*IceShelf*) – id of target

- **bbox** (*BoundingBox*) – a bounding box object
- **points** (List[*XY*]) – a list of polar stereographic coordinates within the bounding box to be used to determine the contrast in an image
- **spacing** (float) – spacing [km] for grid covering the bounding box

**property first_point_reference**
True if five points are used

**Return type** bool

**property grid**

**Return type** *GridXY*

**class** games.utils.targets.**WaterWay**(*value*)
Bases: enum.Enum

An enumeration.

**NARES_STRAIT = 'nares'**

**class** games.utils.targets.**WaterWayTarget**(*members*)
Bases: object

**Parameters members** (Dict[*WaterWay*, *TargetSpacing*]) – dict holding GAMES water way targets

games.utils.targets.**get_ice_shelf_target**(*ice_shelf*)
Get predefined target setting for a specific ice shelf.

**Parameters target** – the name of the target

**Returns:** a target object

**Return type** *TargetXY*

games.utils.targets.**get_lake_target**(*lake*)
Get predefined target setting for a specific lake.

**Parameters target** – the name of the target

**Returns:** a target object

**Return type** *TargetLayer*

games.utils.targets.**get_mountain_target**(*mountain*)
Get predefined target setting for a specific mountain.

**Parameters target** – the name of the target

**Returns:** a target object

**Return type** *Target*

games.utils.targets.**get_water_way_target**(*water_way*)
Get predefined target setting for a specific water way.

**Parameters target** – the name of the target

**Returns:** a target object

**Return type** *TargetSpacing*

### 5.1.8 sensor

sensor.py Defintion of ICI, MWI, and SSMIS channels.

**class** games.utils.sensor.**Channel**(*sensor*, *id*, *receiver*, *id_receiver=0*, *id_horn=0*, *elevation_offset=0*)

Bases: object

Object derived for ICI, MWI, and SSMIS channels.

Note that the purpose of three of the attributes of a channel object is to be used to identify where associated data is located in a Level1B file holding ICI, MWI, and SSMIS data, and predefined instances of Channel objects for these instruments are available from a "Sensor" class of this module.

**Parameters**

- **sensor** (*SensorType*) – sensor type
- **id** (Union[*ChannelICI*, *ChannelMWI*, *ChannelSSMIS*]) – channel type
- **receiver** (str) – name of the receiver (name of the variable where data for this channel is found in the Level1b file)
- **id_receiver** (int) – id of the receiver (as defined in the EPS-SG ICI/MWI Level 1B Product Format Specification)
- **id_horn** (int) – id of the horn (as defined in the EPS-SG ICI/MWI Level 1B Product Format Specification)
- **elevation_offset** (float) – elevation offset angle [degrees] relative to the main tilt angle of the antenna

**property index**

**Return type** int

**property index_horn**

**Return type** int

**property index_receiver**

**Return type** int

**property ncvar_radiance_name**

**Return type** str

**class** games.utils.sensor.**ChannelICI**(*value*)

Bases: enum.Enum

An enumeration.

**ICI1 = 1**

**ICI10 = 11**

**ICI11H = 13**

**ICI11V = 12**

**ICI2 = 2**

**ICI3 = 3**

**ICI4H = 5**

**ICI4V = 4**

**ICI5 = 6**

**ICI6 = 7**

**ICI7 = 8**

**ICI8 = 9**

**ICI9 = 10**

**class** games.utils.sensor.**ChannelMWI**(*value*)

Bases: enum.Enum

An enumeration.

**MWI10 = 18**

**MWI11 = 19**

**MWI12 = 20**

**MWI13 = 21**

**MWI14 = 22**

**MWI15 = 23**

**MWI16 = 24**

**MWI17 = 25**

**MWI18 = 26**

**MWI1H = 2**

**MWI1V = 1**

**MWI2H = 4**

**MWI2V = 3**

**MWI3H = 6**

**MWI3V = 5**

**MWI4H = 8**

**MWI4V = 7**

**MWI5H = 10**

**MWI5V = 9**

**MWI6H = 12**

**MWI6V = 11**

**MWI7H = 14**

**MWI7V = 13**

**MWI8H = 16**

**MWI8V = 15**

**MWI9 = 17**

**class** games.utils.sensor.**ChannelSSMIS**(*value*)
Bases: enum.Enum

An enumeration.

**SSMIS10 = 10**

**SSMIS11 = 11**

**SSMIS8 = 8**

**SSMIS9 = 9**

**class** games.utils.sensor.**Sensor**(*type*, *channels*, *channels_for_dcc_boxes*, *theta=45.0*)
Bases: object

Sensor object for ICI, MWI, and SSMIS.

Note that an instance of a predefined sensor object can be obtained from the from_type method, e.g sensor_ici = Sensor.from_type(SensorType.ICI)

**Parameters**

- **type** (*SensorType*) – type of sensor
- **channels** (List[*Channel*]) – channels of the sensor
- **channels_for_dcc_boxes** (Tuple[Union[*ChannelICI*, *ChannelMWI*, *ChannelSSMIS*], Union[*ChannelICI*, *ChannelMWI*, *ChannelSSMIS*], Union[*ChannelICI*, *ChannelMWI*, *ChannelSSMIS*]]) – channels to use for DCC detection
- **theta** (float) – main tilt angle of the antenna [degrees]

**static from_type**(*sensor_type*)

**Return type** *Sensor*

**get_channel**(*channel_id*)

**Return type** *Channel*

**get_theta_deg**(*channel_id*)

**Return type** float

**class** games.utils.sensor.**SensorType**(*value*)
Bases: enum.Enum

An enumeration.

**ICI = 'ici'**

**MWI = 'mwi'**

**SSMIS = 'ssmis'**

## 5.2 Data import functions and objects

### 5.2.1 gtopo30

gtopo30_reader.py Reader of the GTOPO30 digital elevation model dataset, that is divided into 33 tiles. GTOPO30 is a global data set covering the full extent of latitude from 90 degrees south to 90 degrees north, and the full extent of longitude from 180 degrees west to 180 degrees east. This code does not support the reading of the special GTOPO30 antarctic tile.

`games.utils.gtopo30_reader.`**`get_dem`**`(`*bbox*, *demdatadir*`)`

Get digital elevation model data within a bounding box.

**Parameters**

- **`bbox`** (*`BoundingBox`*) – a bounding box object
- **`demdatadir`** (`str`) – datadir where the GTOPO30 dataset is found

Returns:

GTOPO30 data within a bounding box

**Return type** *`DEM`*

### 5.2.2 sar

**`class`** `games.utils.sar_reader.`**`SarData`**`(`*filename*, *lat*, *lon*, *sigma0_hh*`)`

Bases: `object`

**`property sigma0_hh_filled`**

**Return type** `ndarray`

`games.utils.sar_reader.`**`get_data`**`(`*sar_file*, *bbox*`)`

Get SAR data within a bounding box.

**Parameters**

- **`sar_file`** (`str`) – full path of the SAR file
- **`bbox`** (*`BoundingBox`*) – a bounding box object

Returns:

SAR data within a bounding box

**Return type** `Optional[`*`SarData`*`]`

### 5.2.3 gshhs

gshhs_reader.py Get features from the GSHHS dataset, e.g. the boundary between lake and land.

**`class`** `games.utils.gshhs_reader.`**`GshhsLayer`**`(`*value*`)`

Bases: `enum.Enum`

An enumeration.

**`AntarcticGround = 'L6'`**

**`AntarcticIce = 'L5'`**

**IslandInLake = 'L3'**

**Lake = 'L2'**

**LandOcean = 'L1'**

**PondInIsland = 'L4'**

**class** games.utils.gshhs_reader.**GshhsResolution**(*value*)

Bases: enum.Enum

An enumeration.

**Crude = 'c'**

**Full = 'f'**

**High = 'h'**

**Intermediate = 'i'**

**Low = 'l'**

games.utils.gshhs_reader.**get_features**(*bbox*, *resolution*, *layer*, *datadir*)

Get features from the GSHHS dataset.

**Parameters**

- **bbox** (Union[*BoundingBox*, Iterable[*BoundingBox*]]) – a bounding box object
- **resolution** (*GshhsResolution*) – data resolution
- **layer** (*GshhsLayer*) – the layer to extract
- **datadir** (str) – datadir where the GSHHS dataset can be found

Returns:

features form the GSHHS dataset within a bounding box

### 5.2.4 ssmis

ssmis_reader.py Reader of SSMIS LevelB data files.

**class** games.utils.ssmis_reader.**ImagerData**(*filename*, *channel*, *time*, *lat*, *lon*, *tb*, *lat_sat*, *lon_sat*, *alt_sat*)

Bases: object

**property nsamples**

**Return type** int

**property nscans**

**Return type** int

**samples_to_latlon**()

**Return type** List[*LatLon*]

**sat_to_latlonalt**()

**Return type** List[*LatLonAlt*]

**update_positions**(*corrected*)

**Return type** None

`games.utils.ssmis_reader.`**`get_data`**(*ssmis_file*, *channel*, *bbox*, *atleast_four=True*)
Get SSMIS data.

**Parameters**

- **ssmis_file** (`str`) – full path to the SSMIS level1b file
- **channel** (*Channel*) – a channel of SSMIS
- **bbox** (*BoundingBox*) – a boudning box object
- **atleast_four** (`bool`) – four points flag, if set to True the function will return None if less than two data points are found in any of the two dimension of the data

Returns:

SSMIS data within a bounding box for a given channel

**Return type** `Optional[`*ImagerData*`]`

`games.utils.ssmis_reader.`**`get_start_time`**(*ssmis_file*)

**Return type** `Optional[datetime]`

### 5.2.5 ici and mwi

ici_and_mwi_reader.py Reader of ICI and MWI Level1B data files.

**class** `games.utils.ici_and_mwi_reader.`**`ChannelSet`**(*lat_sensor*, *lon_sensor*, *alt_sensor*, *lat_fov*, *lon_fov*, *ref_data*, *test_data*, *dcc_ch1_data*, *dcc_ch2_data*, *dcc_ch3_data*)

Bases: `object`

**property dcc_channel_set**

**Return type** *DCCChannelSet*

**delta_acrosstrack_km_fov**(*id0*, *id1*)

**Return type** `float`

**delta_alongtrack_km_fov**(*id0*, *id1*)

**Return type** `float`

**scancut**()
Returns filtered data, scan and sample numbers that only contain non-finite data are removed

**Return type** *ChannelSet*

`games.utils.ici_and_mwi_reader.`**`get_data`**(*imager_file*, *channel*, *bbox*, *atleast_four=True*, *orthorectified=True*)
Get imager data.

**Parameters**

- **imager_file** (`str`) – full path to the imager level1b file
- **channel** (*Channel*) – a channel of ICI or MWI
- **bbox** (*BoundingBox*) – a boudning box object
- **atleast_four** (`bool`) – four points flag, if set to True the function will return None if less than two data points are found in any of the two dimension of the data

- **orthorectified** (bool) – flag used to determine if a reconstruction of the orthorectified latitudes and longitudes shall be applied

Returns:

imager data within a bounding box for a given channel

**Return type** Optional[*ImagerData*]

games.utils.ici_and_mwi_reader.**get_start_time**(*level1_file*)

**Return type** datetime

games.utils.ici_and_mwi_reader.**load_channel_set**(*level2_file*, *sensor*, *ref_channel*, *test_channel*)

Get data from a set of channels.

**Parameters**

- **level2file** – full path to a level2 file
- **sensor** (*Sensor*) – a sensor
- **ref_channel** (*Channel*) – the channel to consider as reference channel
- **test_channel** (*Channel*) – the channel to consider as test channel

Returns:

data from a set of channels

**Return type** *ChannelSet*

games.utils.ici_and_mwi_reader.**orthorectify**(*lats*, *lons*, *delta_lats*, *delta_lons*)

Reconstruction of the orthorectified latitudes and longitudes.

**Parameters**

- **lats** (ndarray) – geodetic latitudes [degrees]
- **lons** (ndarray) – geodetic longitudes [degrees]
- **delta_lats** (ndarray) – distance between latitude obtained using DEM and latitude on ellipsoid [m]
- **delta_lons** (ndarray) – distance between longitude obtained using DEM and longitude on ellipsoid [m]

**Return type** Tuple[ndarray, ndarray]

### 5.2.6 dcc mask

dcc_mask.py This module is responsible for identifying regions or bounding boxes containing deep convective clouds (DCC).

The approach to identify DCC is inspired by Hong et al. 2005 "Detection of tropical deep convective clouds from AMSU-B water vapor channels measurements", and uses antenna temperature data from three channels for this purpose.

**class** games.utils.dcc_mask.**BoundingBoxDCC**(*x_min*, *x_max*, *y_min*, *y_max*)

Bases: object

BoundingBox for coverage of a deep convective cloud system using scan and sample number coordinates.

**Parameters**

- **x_min** (int) – min sample number
- **x_max** (int) – max sample number
- **y_min** (int) – min scan number
- **y_max** (int) – max scan number

**property center**

**Return type** Tuple[int, int]

**static from_contour**(*contour*)

Transforms a contour to a bounding box.

**Parameters** **contour** (List[List[Tuple[int, int]]]) – contour surrounding typically a DCC system

Returns:

a bounding box

**Return type** *BoundingBoxDCC*

**property height**

**Return type** int

**resize**(*width*, *height*, *x_max*, *y_max*)

**Return type** *BoundingBoxDCC*

**property width**

**Return type** int

**class** games.utils.dcc_mask.**DCCChannelSet**(*ch1_data*, *ch2_data*, *ch3_data*)

Bases: object

Class for holding antenna temperature data from three channels.

**Parameters**

- **ch1_data** (*TaData*) – antenna temperature data from a first channel
- **ch2_data** (*TaData*) – antenna temperature data from a second channel
- **ch3_data** (*TaData*) – antenna temperature data from a third channel

**class** games.utils.dcc_mask.**TaData**(*scan_number*, *sample_number*, *ta*)

Bases: object

Class for holding antenna temperature data.

**Parameters**

- **scan_number** (ndarray) – scan number
- **sample_number** (ndarray) – sample number
- **ta** (ndarray) – antenna temperature data

**slice_ta**(*bbox*)

**Return type** ndarray

`games.utils.dcc_mask.`**`get_dcc_bboxes`**(*channel_set*, *min_size*, *expansion_factor*, *merge*)

Returns a list of bounding boxes where each bounding box is supposed to cover a region containing deep convective clouds (DCC).

**Parameters**

- **channel_set** (*DCCChannelSet*) – data from three channels
- **min_size** (`int`) – minimum size of bounding boxes to keep, both width and height must be greater than or equal to min_size
- **expansion_factor** (`float`) – expansion factor
- **merge** (`bool`) – flag for merging overlapping bounding boxes

Returns:

a list of bounding boxes

**Return type** `List[`*BoundingBoxDCC*`]`

`games.utils.dcc_mask.`**`get_dcc_mask`**(*ref1*, *ref2*, *ref3*)

Get a deep convective clouds (DCC) mask, inspired by Hong et al. 2005 "Detection of tropical deep convective clouds from AMSU-B water vapor channels measurements"

**Parameters**

- **ref1** (*TaData*) – antenna temperature data ideally at 183.31 +- 7.0 GHz
- **ref2** (*TaData*) – antenna temperature data ideally at 183.31 +- 3.4 GHz
- **ref3** (*TaData*) – antenna temperature data ideally at 183.31 +- 2.0 GHz

Returns:

a DCC mask (2-d array)

**Return type** `ndarray`

## 5.3 Data processing modules

### 5.3.1 dem correction

dem_correction.py Correcting positions (lat, lon) of imager data where the position originally has been estimated neglecting the topography of Earth.

**`class`** `games.utils.dem_correction.`**`DEM`**(*lat*, *lon*, *alt*)

Bases: `object`

Class derived to be able to generate a triangle mesh covering a digitial elevation model (DEM) surface

**Parameters**

- **lat** (`array`) – latitudes [degrees]
- **lon** (`array`) – longitudes [degrees]
- **alt** (`ndarray`) – altitudes [m]

**get_triangle_mesh**()

Returns a triangle mesh covering the DEM, triangles are derived from triangulation of a set of points, the (lat, lon) coordinates of the DEM.

Returns:

a list of Triangle objects

**Return type** List[*Triangle*]

**property latmax**

**Return type** float

**property latmin**

**Return type** float

**property lonmax**

**Return type** float

**property lonmin**

**Return type** float

**reduce_dem**(*n*)

Reduce resolution of DEM data by n times, by calculating averages in blocks of n x n of the DEM

**Parameters n** (int) – DEM reduction factor, must be an odd number

Returns:

None, updates alt attribute, the alt array will have the same size but data will be smoothed

**Return type** None

**regrid_dem**(*spacing*)

Regrid DEM data to a resolution of spacing.

**Parameters spacing** (float) – desired grid resolution [m]

Returns:

None, updates the lat, lon, and alt attributes

**Return type** None

**class** games.utils.dem_correction.**Triangle**(*vertex0*, *vertex1*, *vertex2*)

Bases: object

Triangle class.

**Parameters**

- **vertex0** (*ECEF*) – position of first corner of a triangle
- **vertex1** (*ECEF*) – position of second corner of a triangle
- **vertex2** (*ECEF*) – position of third corner of a triangle

**centroid**()

**Return type** *ECEF*

**class** games.utils.dem_correction.**TriangleMesh**(*triangles*)

Bases: object

Class derived to be able to efficiently find the intersection of a line and a triangle mesh.

**Parameters** **triangles** (List[*Triangle*]) – a list of triangles objects

**filter**(*dest*, *maxdist*)

Filters the triangles in the triangle mesh.

**Parameters**

- **dest** (*ECEF*) – a position
- **maxdist** (float) – distance [m]

Returns:

an array of booleans, the value is true if the triangle centroid is found within a distance maxdist from dest

**Return type** array

**intersection_line_triangle**(*orig*, *dest*, *maxdist*)

Finds the intersection between a line and a triangle mesh based on the Möller-Trumbore ray-triangle intersection algorithm

**Parameters**

- **orig** (*ECEF*) – origin of the line or ray
- **dest** (*ECEF*) – destination of ray
- **maxdist** (float) – distance [m], used to filter which triangles to consider in the calculation, only triangles with a centroid located within this distance are considered

Returns:

the ecef position of the closest point to orig where the line of sight from orig to dest intersects with any of the triangle. returns the position of dest if no intersection is found

**Return type** *ECEF*

games.utils.dem_correction.**dem_correction**(*positions_samples*, *positions_sat*, *dem*, *spacing=5000.0*, *maxdist=30000.0*)

Returns DEM corrected positions of the observed samples.

Corrects positions (lat, lon) of e.g SSMIS data where the position has been estimated neglecting the topography of Earth. The corrected position is the closest point to the satellite where the line of sight from the satellite to the old position intersects with a triangulated Earth surface (using DEM data).

**Parameters**

- **positions_samples** (List[*LatLon*]) – sample positions that has been estimated by neglecting the topography of Earth
- **positions_sat** (List[*LatLonAlt*]) – satellite positions from where the samples were measured
- **dem** (*DEM*) – DEM data
- **spacing** (float) – spacing of dem [m], DEM data is regridded to have this spacing prior to the triangulation. Hence, this parameter will have an impact on the size of the triangles.
- **maxdist** (float) – distance [m], used to filter triangles to us in calculation

Returns:

DEM corrected positions

**Return type** List[*LatLonAlt*]

### 5.3.2 validation

validation.py Module for methods and functions that are responsible for the actual calculation for gelocation validation of data.

**class** games.utils.validation.**GeolocationValidator**(*grid*)

Bases: object

Class derived for gelocation validation of imager data using land mark target as reference.

**Parameters grid** (Union[*Grid*, *GridXY*]) – grid object that defines a regular grid onto which imager and reference data can be gridded, and further used by the methods of this class

**difference**(*image*, *p1*, *p2*)

Returns the difference between the image pixel value at point p2 and p1.

**Parameters**

- **image** (ndarray) – image
- **p1** (Union[*LatLon*, *XY*]) – a position
- **p2** (Union[*LatLon*, *XY*]) – a position

Returns:

difference (float)

**Return type** float

**features_to_meshgrid**(*features*)

Projects a list of positions onto a regular grid following a nearest neighbour approach.

**Parameters features** (Union[List[*LatLon*], List[*XY*]]) – a list of points describing e.g. the boundary between lake and land

Returns:

a 2-d array (a gridded representation of the input features)

**Return type** ndarray

**get_contrast**(*image*, *points*, *first_point_reference*)

Get the contrast of the image, following the GAMES Report Task 1 definition.

**Parameters**

- **image** (ndarray) – gridded observations [K]
- **points** (Union[List[*LatLon*], List[*XY*]]) – list of positions to use for deriving contrast
- **first_point_reference** (bool) – if set to True the first point is used as the reference point, and if set to False the contrast is calculated as the mean of the difference between the image pixel value for every consecutive pair of points, i.e. the average of image(p1) - image(p0) and image(p3) - image(p2) if we consider four points.

**Returns:** the contrast (float) [K]

**Return type** `float`

**get_edges** (*image*, *low_threshold*, *high_threshold*, *sigma*, *scale*, *validate=True*)
Edge filter an image using the Canny algorithm.

Canny, J., A Computational Approach To Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6):679–698, 1986.

**Parameters**

- **image** (`ndarray`) – image
- **low_threshold** (`float`) – lower bound for hysteresis thresholding
- **high_threshold** (`float`) – upper bound for hysteresis thresholding
- **sigma** (`float`) – standard deviation of the Gaussian filter
- **scale** (`bool`) – if this flag is set to True, data will be scaled to be in the range 0 to 1 before Canny filter is applied
- **validate** (`bool`) – if this flag is set to True, the shape of image is validated against the grid attribute

Returns:

edge filtered image (2-d array of floats)

**Return type** `ndarray`

**get_gradient_magnitude** (*image*)
Calculates the gradient of an image using a Sobel filter and returns the gradient magnitude.

Sobel, I., Feldman, G., "A 3x3 Isotropic Gradient Operator for Image Processing", presented at the Stanford Artificial Intelligence Project (SAIL) in 1968.

**Parameters** **image** (`ndarray`) – image

Returns:

gradient of image (2-d array of floats)

**Return type** `ndarray`

**griddata** (*lons*, *lats*, *values*)
Get data gridded on a regular grid.

**Parameters**

- **lons** (`array`) – logitudes [degrees]
- **lats** (`array`) – latitudes [degrees]
- **values** (`array`) – values [unit of values]

Returns:

gridded data (2-d array)

**Return type** `ndarray`

**static register_images** (*image1*, *image2*, *upsampling_factor*)

Returns shifts detected between the two input images using an efficient subpixel image registration by crosscorrelation following the algorithm descibed in

Manuel Guizar-Sicairos, Samuel T. Thurman, and James R. Fienup, "Efficient subpixel image registration algorithms" Opt. Lett. 33, 156-158 (2008).

**Parameters**

- **image1** (`ndarray`) – reference image
- **image2** (`ndarray`) – image to register
- **upsampling_factor** (`int`) – upsampling factor, must be greater than 0

Returns:

index shifts in x, y direction (tuple of floats) of image2 w.r.t. image1, (i.e. shift image2 by these #'s to match image1) and the correlation between the registered and reference image

**Return type** `Tuple[float, float, float]`

**class** `games.utils.validation.`**ValidationResult** (*filename*, *sensor*, *channel*, *date=None*, *lat_center=nan*, *lon_center=nan*, *shift_x=nan*, *shift_y=nan*, *shift=nan*, *contrast=nan*, *second_shift=nan*, *valid=False*, *coverage_problem=0*, *sarfile=None*)

Bases: `object`

Class for holding validation result for a landmark target.

**Parameters**

- **filename** (`str`) – the name of the Level1B used
- **sensor** (`str`) – the name of the sensor
- **channel** (`int`) – the channel ID used
- **date** (`Optional[datetime]`) – a representative datetime of the measurement
- **lat_center** (`float`) – latitude center [degrees] of the bounding box used
- **lon_center** (`float`) – longitude center [degrees] of the bounding box used
- **shift_x** (`float`) – the derived shift in validated data [km] along longitude direction
- **shift_y** (`float`) – the derived shift in validated data [km] along latitude direction
- **shift** (`float`) – the derived shift in validated data [km]
- **contrast** (`float`) – contrast in image [K]
- **second_shift** (`float`) – the second derived shift in validated data [km]
- **valid** (`bool`) – a validity flag of derived shift
- **sarfile** (`Optional[str]`) – the name of the SAR file used

**Para coverage_problem** data coverage problem within target bounding box, 0 = no problem, 1 = no imager data found within the target bounding box, 2 = no target data found within the target bounding box

**asdict** (*recurse=True*, *filter=None*, *dict_factory=<class 'dict'>*, *retain_collection_types=False*)

Return the `attrs` attribute values of *inst* as a dict.

Optionally recurse into other `attrs`-decorated classes.

**Parameters**

- **inst** – Instance of an `attrs`-decorated class.
- **recurse** (*bool*) – Recurse into classes that are also `attrs`-decorated.
- **filter** (*callable*) – A callable whose return code determines whether an attribute or element is included (`True`) or dropped (`False`). Is called with the *attr.Attribute* as the first argument and the value as the second argument.
- **dict_factory** (*callable*) – A callable to produce dictionaries from. For example, to produce ordered dictionaries instead of normal Python dictionaries, pass in `collections.OrderedDict`.
- **retain_collection_types** (*bool*) – Do not convert to `list` when encountering an attribute whose type is `tuple` or `set`. Only meaningful if `recurse` is `True`.

**Return type** return type of *dict_factory*

**Raises** **attr.exceptions.NotAnAttrsClassError** – If *cls* is not an `attrs` class.

New in version 16.0.0: *dict_factory*

New in version 16.1.0: *retain_collection_types*

**static from_file** (*filename*)

**Return type** *ValidationResult*

**outfile** (*outdir*, *target*)

**Return type** `str`

**write** (*outdir*, *target*, *settings*)

Write results to a file in netcdf format.

**Parameters**

- **outdir** (`str`) – name of directory where to write results
- **target** (`str`) – name of the target

**Return type** `None`

games.utils.validation.**displacement_to_pointing_error** (*displacement_acrosstrack*, *displacement_alongtrack*, *view_angle*, *altitude*)

Converts horizontal displacement of a point on ground, as seen by a sensor at a given altitude and view angle, into an azimuth and elevation offset angle.

**Parameters**

- **displacement_acrosstrack** (`float`) – displacement in acrosstrack direction [km]
- **displacement_alongtrack** (`float`) – displacement in alongtrack direction [km]
- **view_angle** (`float`) – sensor view angle [degrees] (0 means nadir)
- **altitude** (`float`) – platform altitude [km]

Returns:

tuple of azimuth and elevation offset angles [degrees]

**Return type** `Tuple[float, float]`

`games.utils.validation.`**`isvalid`**(*contrast*, *shift*, *contrast0*, *shift0*, *threshold*)

Validate derived shift and contrast.

**Parameters**

- **contrast** (`float`) – derived contrast [K]
- **shift** (`float`) – derived shift [km]
- **contrast0** (`float`) – reference contrast [K]
- **shift0** (`float`) – reference shift [km]
- **threshold** (`float`) – reference threshold [-]

Returns:

a boolean, True if contrast and shift is high and low enough, respectively, following a "fuzzy logic" approach described in GAMES Report Task 1

**Return type** `bool`

`games.utils.validation.`**`isvalid_by_shifts`**(*shift1*, *shift2*, *shift0*, *threshold*)

Validate derived shifts.

**Parameters**

- **shift1** (`float`) – derived shift [km]
- **shift2** (`float`) – derived shift [km]
- **shift0** (`float`) – reference shift [km]
- **threshold** (`float`) – reference threshold [-]

Returns:

a boolean, True if shifts are low enough, following a "fuzzy logic" approach described in GAMES Report Task 1

**Return type** `bool`

## 5.4 Pipeline core functions

### 5.4.1 andean

`games.utils.andean.`**`validate`**(*ssmis*, *dem*, *target*, *upsampling_factor=10*)

Returns derived shift and contrast in imager data using DEM data and mountains area target, following method described in GAMES Report Task 1. A Sobel filter is applied on both DEM and imager data to obtain the gradient magnitude of data, and the data are then cross-correlated against each other, in order to derive a possible shift between the data.

**Parameters**

- **ssmis** (*`ImagerData`*) – imager data
- **dem** (*`DEM`*) – digital elevation model data

- **target** (*Target*) – target instance defining a bounding box and grid to use in calculation, and coordinates to use to derive contrast in image
- **upsampling_factor** (int) – upsampling factor for image registration

Returns:

validation result (a ValidationResult object)

**Return type** *ValidationResult*

### 5.4.2 ross

`games.utils.ross.`**`validate`**(*ssmis*, *sar*, *target*, *upsampling_factor=10*, *high_threshold_tb=0.45*, *low_threshold_tb=0.18000000000000002*, *sigma_tb=1.4142135623730951*, *high_threshold_sar=0.1*, *low_threshold_sar=0.04000000000000001*, *sigma_sar=1.4142135623730951*)

Returns derived shift and contrast in imager data using reference SAR data for target, following methods described in GAMES Report Task 1. A Canny edge filter is applied on both imager and SAR data, in order to derive a possible shift between the data.

**Parameters**

- **ssmis** (*ImagerData*) – imager data
- **sar** (*SarData*) – SAR data
- **target** (Union[*TargetXY*, *TargetSpacing*]) – target instance defining a bounding box and two differents grid to use in calculation
- **upsampling_factor** (int) – upsampling factor for image registration
- **high_threshold_tb** (float) – upper bound for hysteresis thresholding in Canny edge filter for imager data
- **low_threshold_tb** (float) – lower bound for hysteresis thresholding in Canny edge filter for imager data
- **sigma_tb** (float) – standard deviation of the Gaussian filter applied in Canny edge filter for imager data
- **high_threshold_sar** (float) – upper bound for hysteresis thresholding in Canny edge filter for SAR data
- **low_threshold_sar** (float) – lower bound for hysteresis thresholding in Canny edge filter for SAR data
- **sigma_sar** (float) – standard deviation of the Gaussian filter applied in Canny edge filter for SAR data

Returns:

validation result (a ValidationResult object)

**Return type** *ValidationResult*

### 5.4.3 qinghai

games.utils.qinghai.**validate**(*ssmis*, *lakes*, *target*, *upsampling_factor=10*, *high_threshold=0.5*, *low_threshold=0.2*, *sigma=1.4142135623730951*)

Returns derived shift and contrast in imager data using boundary between land and lake or ocean as reference data, following a method described in GAMES Report Task 1. A Canny edge filter is applied on the imager data, and resulting data is cross-correlated with the reference data, in order to derive a possible shift between the data.

**Parameters**

- **ssmis** (*ImagerData*) – imager data
- **lakes** (List[*LatLon*]) – a list of coordinates describing the boundary between lake and land
- **target** (*TargetLayer*) – target instance defining a bounding box and grid to use in calculation, and coordinates to use to derive contrast in image
- **upsampling_factor** (int) – upsampling factor for image registration
- **high_threshold** (float) – upper bound for hysteresis thresholding in Canny edge filter
- **low_threshold** (float) – lower bound for hysteresis thresholding in Canny edge filter
- **sigma** (float) – standard deviation of the Gaussian filter applied in Canny edge filter

Returns:

validation result (a ValidationResult object)

**Return type** *ValidationResult*

### 5.4.4 relative pointing error

**class** games.utils.relative_pointing_error.**PointingError**(*level1_file*, *sensor*, *reference_channel*, *test_channel*, *scan_number_min*, *scan_number_max*, *sample_number_min*, *sample_number_max*, *delta_x_est_km=nan*, *delta_y_est_km=nan*, *azm_est_deg=nan*, *elv_est_deg=nan*, *corrcoef=nan*)

Bases: object

Class for holding result from the relative pointing error estimation.

**Parameters**

- **level1_file** (str) – the name of the Level1C file used
- **sensor** (str) – the name of the sensor
- **reference_channel** (int) – the reference channel ID used
- **test_channel** (int) – the test channel ID used

- **scan_number_min** (`int`) – minimum scan number of the bounding box
- **scan_number_max** (`int`) – maximum scan number of the bounding box
- **sample_number_min** (`int`) – minimum sample number of the bounding box
- **sample_number_max** (`int`) – maximum sample number of the bounding box
- **delta_x_est_km** (`float`) – estimated shift in the across-track direction [km]
- **delta_y_est_km** (`float`) – estimated shift in the along-track direction [km]
- **azm_est_deg** (`float`) – estimated error in azimuth viewing angle [degrees]
- **elv_est_deg** (`float`) – estimated error in elevation viewing angle [degrees]
- **corrcoef** (`float`) – correlation coefficient between the corrected data from the test channel and data from the reference channel

**asdict** (*recurse=True*, *filter=None*, *dict_factory=<class 'dict'>*, *retain_collection_types=False*)

Return the `attrs` attribute values of *inst* as a dict.

Optionally recurse into other `attrs`-decorated classes.

**Parameters**

- **inst** – Instance of an `attrs`-decorated class.
- **recurse** (*bool*) – Recurse into classes that are also `attrs`-decorated.
- **filter** (*callable*) – A callable whose return code determines whether an attribute or element is included (`True`) or dropped (`False`). Is called with the *attr.Attribute* as the first argument and the value as the second argument.
- **dict_factory** (*callable*) – A callable to produce dictionaries from. For example, to produce ordered dictionaries instead of normal Python dictionaries, pass in `collections.OrderedDict`.
- **retain_collection_types** (*bool*) – Do not convert to `list` when encountering an attribute whose type is `tuple` or `set`. Only meaningful if `recurse` is `True`.

**Return type** return type of *dict_factory*

**Raises** **attr.exceptions.NotAnAttrsClassError** – If *cls* is not an `attrs` class.

New in version 16.0.0: *dict_factory*

New in version 16.1.0: *retain_collection_types*

`games.utils.relative_pointing_error.`**get_relative_pointing_error**(*level1_file*, *sensor*, *ref_channel*, *test_channel*, *min_size=20*, *expansion_factor=1.2*, *merge_bboxes=False*, *upsampling_factor=100*)

Returns derived relative pointing error between a reference and a second channel of MWI or ICI, following method described in GAMES Report Task 2.

Three MWI or ICI channels around the 183.31 GHz water vapor line are used to identify regions (bounding boxes) with deep convective clouds and pointing errors are derived for data within bounding boxes covering these regions.

**Parameters**

- **level1_file** (str) – full path to a level1c file
- **sensor** (*Sensor*) – sensor
- **ref_channel** (*Channel*) – channel to be considered as reference channel
- **test_channel** (*Channel*) – channel to be considered as test channel
- **min_size** (int) – minimum size of bounding boxes to keep, both width and height must be greater than or equal to min_size
- **expansion_factor** (float) – bounding box expansion factor
- **merge_bboxes** (bool) – flag for merging overlapping bounding boxes
- **upsampling_factor** (int) – upsampling factor for image registration

Returns:

derived relative pointing error

**Return type** List[*PointingError*]

games.utils.relative_pointing_error.**load_result**(*filename*)

Load result.

**Parameters** **filename** (str) – full path to the file to load

**Return type** List[*PointingError*]

games.utils.relative_pointing_error.**write_result**(*outdir*, *pointing_errors*, *settings*)

Writes derived pointing errors to a netcdf formated file.

**Parameters**

- **outdir** (str) – name of directory where to write results
- **pointing_errors** (List[*PointingError*]) – list of derived errors

**Return type** None

# PYTHON MODULE INDEX

## g

# INDEX

## H



## I



## K



## L

## M



## N



## O



## P

## Q



## R



## S



## T



## U



## V



## W



## X

## Y



## Z