Surface Water and Ocean Topography Project

Algorithm Theoretical Basis Document
Long Name: Level 2 KaRIn high rate pixel cloud science algorithm software
Short Name: L2_HR_PIXC

Initial Release

Document Custodians:

Brent Williams JPL Algorithm Engineer	Date	Roger Fjørtoft CNES Algorithm Engineer	Date

Approved by:

Curtis Chen JPL Algorithm System Engineer	Date	Roger Fjørtoft CNES Algorithm System Engineer	Date

Concurred by:

Tamlin Pavelsky JPL Hydrology Lead	Date	Jean Francois Cretaux CNES Hydrology Lead	Date

Paper copies of this document may not be current and should not be relied on for official purposes. The current version is in the JPL Product Data Management System (EPDM: https://epdm.jpl.nasa.gov) and the CNES Product Data Management System

See Sect 1.4 for how to cite this document

July 13, 2023
JPL D-105504

National Aeronautics and Space Administration
Jet Propulsion Laboratory
California Institute of Technology

cnes

Contributing Authors

Name	Affiliation
Xiaoqing Wu	Jet Propulsion Laboratory
Dustin Lagoy	Jet Propulsion Laboratory
Alex Corben	Jet Propulsion Laboratory

Science Team Reviewers

Name	Affiliation
Michael Durand	Ohio State University
Herve Yésou	Service Régional de Traitement d'Image et de Télédétection (SERTIT), Université de Strasbourg
Florence Tupin	Telecom Paris, Institut Polytechnique de Paris

Change Log

VERSION	DATE	SECTIONS CHANGED	REASON FOR CHANGE
Initial Release	2023-07-13	ALL	Initial Release Approved for public release (URS217417/CL#23-3279)

Table of Contents

Contributing Authors 2

Science Team Reviewers 2

Change Log 3

Table of Contents 4

Table of Figures 7

Table of Tables 9

List of TBC Items 10

List of TBD Items 10

1 Introduction 11
  1.1 Purpose 11
  1.2 Scope 11
  1.3 Document Organization 11
  1.4 Citing This Document 11

2 Overview 12
  2.1 Background and Context 12
  2.2 Pixel Cloud Reference Location and Sampling Relationships 12
  2.3 Functional Flow 15
    2.3.1 Illustrative Example 18

3 Algorithm Descriptions 34
  3.1 MakeUnflatSLC 34
    3.1.1 Purpose 34
    3.1.2 Input Data 34
    3.1.3 Output Data 34
    3.1.4 Mathematical Statement 34
    3.1.5 Accuracy 36
  3.2 MakeRareIntf 37
    3.2.1 Purpose 37
    3.2.2 Input Data 37
    3.2.3 Output Data 37
    3.2.4 Mathematical Statement 37
      3.2.4.1 Calibration and Geophysical Corrections 39
    3.2.5 Accuracy 41
  3.3 MakeClassMaps 43
    3.3.1 MakeCohPow 44
      3.3.1.1 Input Data 44
      3.3.1.2 Output Data 44
      3.3.1.3 Mathematical Statement 44
      3.3.1.4 Accuracy 47
    3.3.2 WaterDetection 48
      3.3.2.1 Input Data 48
      3.3.2.2 Output Data 49

3.3.2.3 Mathematical Statement ... 49
3.3.2.4 Accuracy ... 51
3.3.3 WaterFracEst ... 55
3.3.3.1 Input Data ... 55
3.3.3.2 Output Data ... 55
3.3.3.3 Mathematical Statement ... 56
3.3.3.4 Accuracy ... 56
3.3.4 DarkWaterFlagging ... 59
3.3.4.1 Input Data ... 60
3.3.4.2 Output Data ... 60
3.3.4.3 Mathematical Statement ... 60
3.3.4.4 Accuracy ... 62
3.3.5 LayoverFlagging ... 64
3.3.5.1 Input Data ... 64
3.3.5.2 Output Data ... 64
3.3.5.3 Mathematical Statement ... 64
3.3.5.4 Accuracy ... 65
3.3.6 BrightLandFlagging ... 66
3.3.6.1 Input Data ... 66
3.3.6.2 Output Data ... 66
3.3.6.3 Mathematical Statement ... 66
3.3.6.4 Accuracy ... 66
3.3.7 LowCohFlagging ... 67
3.3.7.1 Input Data ... 67
3.3.7.2 Output Data ... 67
3.3.7.3 Mathematical Statement ... 67
3.3.7.4 Accuracy ... 68
3.3.8 BuildClassMaps ... 70
3.3.8.1 Input Data ... 70
3.3.8.2 Output Data ... 70
3.3.8.3 Mathematical Statement ... 70
3.3.8.4 Accuracy ... 71
3.3.9 Accuracy of MakeClassMaps ... 71
3.4 MakeMedIntf ... 73
3.4.1 Purpose ... 73
3.4.2 Input Data ... 73
3.4.3 Output Data ... 73
3.4.4 Mathematical Statement ... 73
3.4.4.1 Calibration and Geophysical Corrections ... 75
3.4.5 Accuracy ... 75
3.5 UnwGeoloc ... 76
3.5.1 SpatialUnwrap ... 77
3.5.1.1 Input Data ... 78
3.5.1.2 Output Data ... 78
3.5.1.3 Mathematical Statement ... 78
3.5.1.4 Accuracy ... 79
3.5.2 BreakRegions ... 79
3.5.2.1 Input Data ... 79
3.5.2.2 Output Data ... 79
3.5.2.3 Mathematical Statement ... 79
3.5.2.4 Accuracy ... 83

3.5.3 AmbigSolver ... 85
3.5.3.1 Input Data ... 85
3.5.3.2 Output Data ... 85
3.5.3.3 Mathematical Statement ... 85
3.5.3.4 Accuracy ... 87
3.5.4 Geolocate ... 89
3.5.4.1 Input Data ... 89
3.5.4.2 Output Data ... 89
3.5.4.3 Mathematical Statement ... 89
3.5.4.4 Accuracy ... 92
3.5.5 Accuracy of UnwGeoloc ... 92
3.6 CompGeophysCorr ... 96
3.6.1 Purpose ... 96
3.6.2 Input Data ... 96
3.6.3 Output Data ... 96
3.6.4 Mathematical Statement ... 96
3.6.5 Accuracy ... 98
3.7 MakePixC ... 99
3.7.1 Purpose ... 99
3.7.2 Input Data ... 99
3.7.3 Output Data ... 99
3.7.4 Mathematical Statement ... 99
3.7.4.1 Quality Flagging ... 100
3.7.5 Accuracy ... 103
4 Accuracy of L2_HR_PIXC Processor ... 104
4.1 Height ... 104
4.2 Slope ... 107
4.3 Area ... 109
5 References ... 112
Appendix A. **Acronyms ... 113**
Appendix B. **Simulations ... 114**

Table of Figures

Figure 1. Notional illustration of the relationships between the posting and information content of SLC ("raw"), rare, and medium samples (well-done aggregation not shown). Note that the rare and medium posting grids are identical, but the interferometric quantities themselves are shape-adaptively-multilooked for the medium layer. The detected water (blue) is not itself aggregated to the medium level, but is used to define which rare interferometic pixels are multilooked to produce each medium interferometric pixel. The green and purple shaded boxes represent a simplified illustration of the adaptive averaging windows for two different rare/medium water pixels (with the corresponding colored "x"). The shaded pixels are those used to multilook the interferometric quantities into the pixel with the corresponding "x" for each color. This is a simplified, notional illustration of the adaptive multilooking (see Section 3.4 for more specific details). 14

Figure 2. Flow diagram of the Level 2 Pixel cloud processing steps (functions) used to generate the L2_HR_PIXC product (TODO: update this figure, Dynamic Geophys and XOverCal aux data goes into Make RareIntf too). 18

Figure 3. Simulation input dem elevation wrapped at 20m (upper), and land type mask (lower) for the Missouri River example. 21

Figure 4. Slant-plane single look complex (SLC) images of the reference channel (upper), and the secondary channel (lower) for the Missouri River example. Note that the slant-plane image is oriented differently than the ground projection images (approximately flipped vertically). These quantities are produced by the SLC unflattening algorithm described in Section 3.1. 22

Figure 5. Slant-plane images of the rare flat interferogram magnitude and phase (upper), and the rare unflattened interferogram magnitude and phase (lower) for the Missouri River example. These quantities are produced by the rare interferogram algorithm described in Section 3.2. 23

Figure 6. Slant-plane images of the rare interferogram coherent power (upper), and the coherent gain (lower) for the Missouri River example. These quantities are produced by the coherent power computation algorithm described in Section 3.3.1. 24

Figure 7. Slant-plane images of the detected water (upper), and the estimated water fraction (lower) for the Missouri River example. These quantities are produced by the water detection algorithm described in Section 3.3.2 and the water fraction estimation algorithm described in Section 3.3.3. 25

Figure 8. Slant-plane images of the prior water probability (upper), and the corresponding prior water mask (lower) for the Missouri River example. Note that the prior probability and mask differ from the truth water mask as well as the noisy SWOT data, especially in areas with narrow water features. This is likely due to the resolution limitations of the prior data used to make the mask. These quantities are produced by the dark water flagging algorithm described in Section 3.3.4. 26

Figure 9. Slant-plane image of the layover impact flag for the Missouri River example. This quantity is produced by the layover flagging algorithm described in Section 3.3.5. 27

Figure 10. Slant-plane image of the low coherence detected water flag for the Missouri River example. This quantity is produced by the low-coherence flagging algorithm described in Section 3.3.7. 27

Figure 11. Slant-plane images of the rare interferogram multivalued classification (upper), and zoomed estimated multivalued classification (lower left) along with the corresponding truth classification (lower right) for the Missouri River example. The multivalued classification quantity is produced by the algorithm described in Section 3.3.8. 28

Figure 12. Slant-plane images of the medium interferogram magnitude and flattened phase (upper), and the medium interferogram coherence (lower) for the Missouri River example. These quantities are produced by the medium interferogram algorithm described in Section 3.4. 29

Figure 13. Slant-plane images of the phase unwrapping region map (upper), and the spatially unwrapped phase over each region (lower) for the Missouri River example. These quantities are produced by the phase unwrapping algorithm described in Section 3.5. 30

Figure 14. Slant-plane images of estimated latitude (top left), longitude (top right), height (bottom left and bottom right) for the Missouri River example. These quantities are produced by the geolocation algorithm described in Section 3.5.4. 31

Figure 15. Slant-plane images of the phase noise standard deviation, which is derived from the interferometric coherence (upper), and the resulting estimate of the height error standard deviation (lower) for the Missouri River example. These quantities are produced by the geolocation algorithm described in Section 3.5.4. 32

Figure 16. Three dimensional scatterplots of the water pixels for a zoomed in portion of the Missouri River example. Gray lines indicate the estimated geolocation error vector. These quantities are produced by the geolocation algorithm described in Section 3.5.4. 33
Figure 17. Illustration of reference locations. A related figure with modified or “conditioned” reference locations is plotted in Figure 19. 35
Figure 18. Flow diagram of lower-level algorithms within the MakeClassMaps function (TODO: update figure intRefDEM-> RefDEM). 43
Figure 19. Illustration of reference location conditioning. Recall that a similar figure with the reference locations without conditioning is plotted in Figure 17. 46
Figure 20. Ka-band normalized radar cross-section (σ0) versus incidence (zenith) angle from the Global Precipitation Mission (GPM) (left), and σ0 (with noise subtraction) versus incidence angle over inland water from AirSWOT. 48
Figure 21. Iterative water detection and background land and water power parameter estimation [12] [13]. 51
Figure 22. Estimated (P) and measured (p) false-detection (fdbw) and missed-detection (mdbw) rates of bright water for simulation with representative dataset. 53
Figure 23. Estimated (P) and measured (p) false-detection (fdbw) and missed-detection (mdbw) rates of bright water excluding true land/water edges for simulation with representative dataset (truth land/water edges is only known in simulated cases). 53
Figure 24. Estimated (P) and measured (p) false-detection (fdbw) and missed-detection (mdbw) rates of bright water excluding detected land/water edges for simulation with representative dataset. 54
Figure 25. Coherence time estimates from AirSWOT using two estimation methods (left) and azimuth smearing width for AirSWOT and SWOT (right). 55
Figure 26. Simulated water fraction error for detected edge pixels (water and land near the shore). 58
Figure 27. AirSWOT power images over Mono Lake for two different surface wind speed conditions (~3-4 m/s for left, ~5-7 m/s for middle), and Ka-band geophysical model function showing σ0 versus incidence angle for several wind speeds (right). 59
Figure 28. Detection rates for land, bright, and dark water given various truth classes. The probability of missed detection of dark water (p_mddw) is plotted for all data and all water features greter than 100m wide, while the probability of false detection of dark water is plotted only for all data. 63
Figure 29. Example of detected low-coherence water that is caused by water-water layover. 69
Figure 30. Flow diagram of lower-level algorithms within the UnwGeoloc function. (TODO: update this flow diagram with all inputs [except ClassMaps?] also going to BreakRegions, and I think ClassMaps going to SpatialUnwrap).77
Figure 31 Ambiguity height versus cross-track (top) and cross-track shift for one ambiguity error versus cross-track (bottom). 81
Figure 32. Pixel-wise phase ambiguity error rate as a function of cross-track distance (TODO: ignore after 60km…resolve issue). 88
Figure 33 Range sphere, Doppler one, and phase cone used in geolocating each slant-plane pixel (TODO: annotate this figure with T, P, v, b etc.). 90
Figure 34. Height error for detected bright water as a function of cross-track distance. 93
Figure 35. Height error for near shore pixels (both land and water near shorelines) as a function of cross-track distance. 94
Figure 36. Height error for detected dark water as a function of cross-track distance. 95
Figure 37. Height error for detected land as a function of cross-track distance. 95
Figure 38. Normalized radar cross-section (σ0) error as a function of incidence angle. 103
Figure 39. Projected water feature height bias as a function of cross-track distance. 106
Figure 40. Projected water feature 1- σ height error as a function of cross-track distance. 107
Figure 41. Projected water feature 1- σ slope error as a function of cross-track distance. 108
Figure 42. Projected water feature area % error bias as a function of cross-track distance. 110
Figure 43. Projected water feature 1- σ area % error as a function of cross-track distance. 110
Figure 44. Dark water occurrence rate vs distance to shore and incidence angle for the simulation representative dataset. 115

Table of Tables

Table 1. High-level description of the functions used to generate the L2_HR_PIXC product. 16
Table 2. High-level description of the functions within the MakeClassMaps function. 43
Table 3. High level description of the functions within the UnwGeoloc function. 76

List of TBC Items

Page	Section

List of TBD Items

Page	Section

1 Introduction

1.1 Purpose

The purpose of this Algorithm Theoretical Basis Document (ATBD) is to describe the physical and mathematical basis for the science data processing algorithms that are used to generate the SWOT Level 2 KaRIn high-rate water mask pixel cloud (L2_HR_PIXC) science data product [1]. These algorithms are applied in the L2_HR_PIXC science algorithm software (SAS). The L2_HR_PIXC SAS performs processing on the single-look complex (SLC) images (the L1B_HR_SLC product [2]) to produce the L2_HR_PIXC product. The L2_HR_PIXC product gives the heights and geolocations of the image pixels of water regions observed by SWOT, along with many other quantities that enhance the utility or that are needed to generate derived products.

1.2 Scope

The scope of this document is to:

1. Identify the list of primary functions that compose the processing steps within the L2_HR_PIXC SAS and their flow. These functions are broken down by the primary functional steps involved in the processing.
2. Describe the purpose of each of the functions.
3. Describe the input data to each function.
4. Describe the output data from each function.
5. Describe the mathematical basis of the algorithm in each function.
6. Describe the expected accuracy and/or limitations of the algorithm in each function.
7. Provide the relevant references for the algorithms described in this document.

1.3 Document Organization

Section 1 provides the purpose and scope of this document.

Section 2 provides the background and context of the algorithms described in this document, the functional flow of the primary functions (e.g., block diagram).

Section 3 provides the algorithm description for each of the functions shown in the block diagram, including input data, output data, mathematical basis, and expected accuracy.

Section 4 provides a summary of the accuracy of the L2_HR_PIXC processor.

Section 5 provides references for the algorithms described in this document.

Appendix A provides a listing of the acronyms used in this document.

Appendix B provides a description of the simulation methodology used to evaluate performance of the L2_HR_PIXC algorithms.

1.4 Citing This Document

Please cite this document as follows:

> JPL D-105504, “SWOT Algorithm Theoretical Basis Document: Level 2 KaRIn High Rate Pixel Cloud (L2_HR_PIXC) Science Algorithm Software,” Jet Propulsion Laboratory Internal Document, 2023.

2 Overview

2.1 Background and Context

The Surface Water and Ocean Topography (SWOT) mission is a partnership between two communities, physical oceanography and hydrology, to share high vertical accuracy topography data produced by the payload, whose principal instrument is the Ka-band Radar Interferometer (KaRIn). The details of SWOT mission objectives and requirements can be found in the SWOT Science Requirements Document [3].

To accomplish the science objectives for Oceanography and Hydrology, the KaRIn instrument produces and downlinks low-rate, onboard-processed SAR data everywhere, and high-rate, minimally processed data mainly over terrestrial surfaces. The high-rate data is of primary interest for hydrology studies.

The lowest level high-rate science data product is the L1B_HR_SLC product [2], which represents single-look complex (SLC) synthetic aperture radar (SAR) images. The L1B_HR_SLC product, however, is likely to be of use to only a few specialized hydrology investigations because it represents mainly radar-specific quantities and is not geolocated. The L2_HR_PIXC processor takes the L1B_HR_SLC product as input and applies several algorithms including water detection, phase unwrapping, and geolocation to produce the L2_HR_PIXC pixel-cloud product [1]. The L2_HR_PIXC product has several quantities that are directly useful for hydrology (primarily geolocated water heights). The L2_HR_PIXC product is then used to generate the more aggregated river and lake single pass (SP) and cycle-averaged (Avg) “vector” products (L2_HR_RiverSP, L2_HR_RiverAvg [4], L2_HR_LakeSP, L2_HR_LakeAvg [5]) as well as the raster product (L2_HR_Raster [6]), which are likely better suited for most hydrology applications.

This document describes the algorithms that are used to generate the L2_HR_PIXC product. First, some high-level background is provided including: relevant details about the SLC grid and interferometric phase from which pixel-cloud processing begins, the sampling and spatial resolution involved throughout the pixel-cloud processing chain, and a description of the functional flow of the algorithms. Section 3 describes the algorithms that are used to generate the L2_HR_PIXC product from the L2_HR_SLC product and various auxiliary data inputs. Section 4 presents the overall accuracy assessment of the L2_HR_PIXC algorithms. Note that more general information on wide swath altimetry can be found in [7], while specific information about the SWOT instrument design and error budget can be found in [8].

2.2 Pixel Cloud Reference Location and Sampling Relationships

The KaRIn instrument employs cross-track interferometry to obtain a 3-D reconstruction of the surface topography. In simplified terms, each of the two co-registered radar SLC images (corresponding to the two receive antennas or channels) resolve a 2-D, complex-valued image of the same surface as projected into the radar imaging coordinates (also called the slant-plane). Note that for the KaRIn instrument, two spatially independent swaths are observed on either side of the satellite using the same pair of antennas, but separated by polarization. For a given swath, multiplying one of the two complex images by the complex conjugate of the other image provides a 2-D interferogram whose phase can be used to measure target locations in 3-D space, thereby providing a map of Earth-surface topography. For more details on SAR interferometry see [9].

This section provides a brief introduction to the relationships between sample locations and the Earth surface at various stages of the KaRIn HR processing flow. Additional detail is provided in subsequent sections, but the high-level description of reference locations and sampling in this section facilitate the discussions that follow in later sections.

The slant-plane image coordinates have an along-track or azimuth component and a range component; for a side-looking SAR, the range is approximately aligned with the cross-track direction. Resolution in the along-track dimension is based on Doppler principles, while resolution in the range dimension is based on the bandwidth of the transmit waveform.

The slant-plane SLC sampling grid is constructed by computing “reference locations” that represent an assumed Earth surface from a digital elevation model (DEM). The SWOT SLC sampling grid is “deskewed,” so lines of cross-track samples are chosen to lie perpendicular to the nadir track (i.e., at broadside). SAR focusing to these reference locations results in SLC image pairs whose relative phase is “flattened” such that the phase of the interferogram formed from the SLCs is related to the difference between the true imaged surface and the reference surface defined by the DEM.

While reference locations and their associated phase flattening are defined by the L1B_HR_SLC product that is the input to the L2_HR_PIXC processing flow, the reference locations, and hence the phase flattening as well, are redefined multiple times during pixel-cloud processing. This is done to apply different assumptions on the shape of the reference surface that are chosen to enhance algorithm performance. This is because spatial averaging of complex values is most advantageous when expected variations in the phase are compensated prior to averaging, while the estimates of the expected phase improve as the steps in the processing progress. (Note that pixel-cloud processing does not necessarily use the same DEM as SLC processing.) In some cases, the reference locations are “conditioned” in a manner that the assumed reference surface is aphysical in some respects but is nonetheless most appropriate for the specific algorithm in which the reference locations are used. Details of reference locations and conditioning are explained in the corresponding descriptions of the functions in Section 3.

Pixel-cloud processing also involves multiple stages of spatial averaging, with different portions of the data processing done at different stages. Figure 1 illustrates the different grid postings and resolutions involved in the processing. The SLC images are posted on a grid whose along-track spacing matches the presummed echo spacing. For the purpose of the pixel cloud processor, this is considered the “raw” grid (note, however, that the term “raw data” can refer to different quantities in other contexts). Within the pixel cloud processor, the data undergo an initial stage of along-track spatial averaging to obtain what is known as the “rare” layer. Later, the data undergo a second stage of (shape-adaptive) spatial averaging to obtain what is known as the “medium” layer. Information from both the rare and medium layers is included in the pixel-cloud product. Additional averaging typically occurs in processing downstream of the L2_HR_PIXC product and is output for each pixel cloud pixel in a companion product called the L2_HR_PIXCVec product [10]. This can be considered a “well-done” layer. These different layers or flavors of spatial smoothing are provided in the level 2 products so that science users can select a product that is “cooked to order” for their particular preference and application.

The posting in the pixel cloud corresponds to the rare interferogram posting for both the “rare” and the “medium” layers. That is, each pixel in the pixel cloud corresponds one-to-one to a particular pixel in the radar interferogram after azimuth multilooking to a downsampled grid. This downsampled grid is referred to as the “rare grid”, whether it is describing “rare” or “medium” (or “well-done”) quantities. In other words, the data are multilooked, but *not* downsampled in the "medium" averaging step (i.e., low-pass filtered in place without decimation).

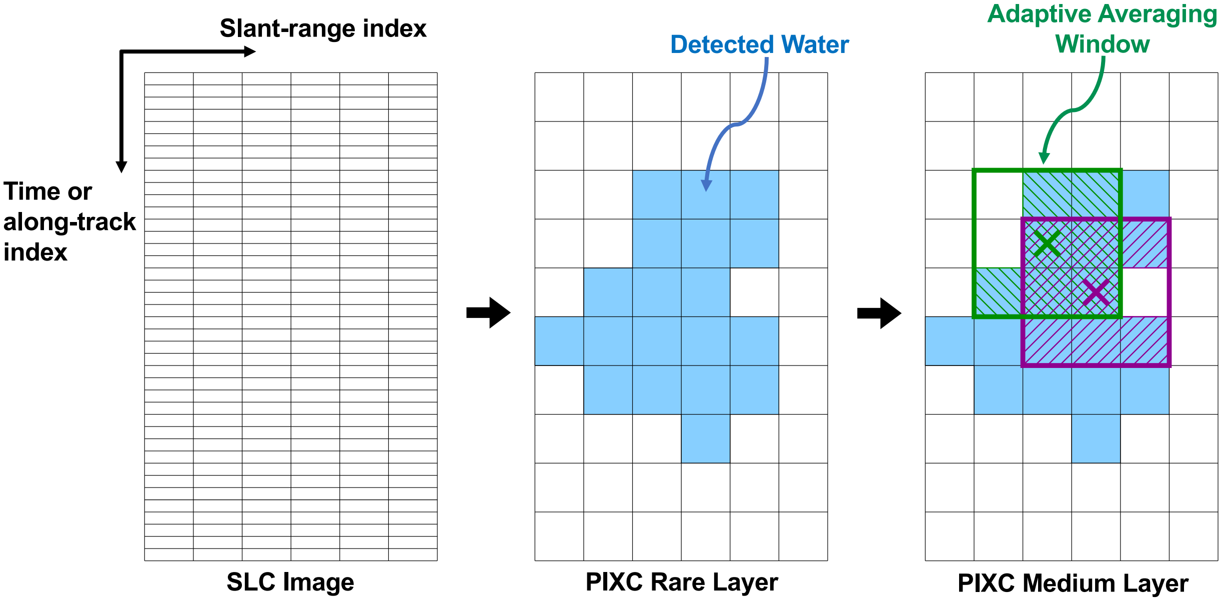


Figure 1. Notional illustration of the relationships between the posting and information content of SLC ("raw"), rare, and medium samples (well-done aggregation not shown). Note that the rare and medium posting grids are identical, but the interferometric quantities themselves are shape-adaptively-multilooked for the medium layer. The detected water (blue) is not itself aggregated to the medium level, but is used to define which rare interferometic pixels are multilooked to produce each medium interferometric pixel. The green and purple shaded boxes represent a simplified illustration of the adaptive averaging windows for two different rare/medium water pixels (with the corresponding colored "x"). The shaded pixels are those used to multilook the interferometric quantities into the pixel with the corresponding "x" for each color. This is a simplified, notional illustration of the adaptive multilooking (see Section 3.4 for more specific details).

2.3 Functional Flow

This section gives an overview of the high-level flow of pixel-cloud processing and summarizes the various functions and intermediate data elements produced throughout the processing chain.

Table 1 provides a brief description of each of the processing functions that are used to generate the L2_HR_PIXC product from the L2_HR_SLC product. Figure 2 illustrates the flow of these processing steps.

MakeUnflatSLC

The L2_HR_PIXC processing begins with the L1B_HR_SLC product, which includes two complex-valued slant-plane images whose relative phase is “flattened,” or referenced to the surface represented by a Digital Elevation Model (DEM), which is provided in the SLC product (the SLC GrDEM). Unflattened SLC images (UnflatSLCs) are computed by unflattening the relative phase of the Level 1 SLC images using the SLC GrDEM. This enables the pixel cloud processing to continue independently of the assumptions and trades applied when choosing the SLC GrDEM, enabling the use of a potentially different reference DEM more tailored to the pixel cloud processing. Note that the algorithms described in this document are intended to be compatible with different choices of reference DEMs (potentially different from the DEMs used in other SWOT processors, e.g., the SLC processor). The current implementation uses the same reference DEM for both SLC and PIXC processing, but this may not always be the case (e.g, reprocessing of PIXCs from existing SLCs with an updated DEM). The current DEM used by the PIXC processor is a combination of NASADEM, MERIT, and the mean sea surface (MSS). Note that the algorithm is the same regardless of the specifics related to the DEM.

MakeRareIntf

The unflattened SLC images (UnflatSLCs) are interfered to produce the “raw” interferogram. The crossover calibration and some geophysical corrections are applied. The interferogram phase at this stage is very noisy. Although the “raw” or “single-look” interferogram phase could be used to geolocate the pixels, the measurements are more accurate if the interferogram is first spatially multilooked to regularize the phase (initially, averaging complex interferogram values gives greater benefit than averaging geolocated heights, as well as reducing elevation biases).

After the “raw” interferogram is formed, it is flattened again using a flat-Earth reference surface that does not contain topographic variations. This is done to avoid artifacts over water surfaces, which are expected to be relatively flat, that might arise from errors in a reference DEM with topography. Such artifacts would cause errors in the azimuth multilooking of the interferogram and reference locations used to form the “rare” interferogram, which occurs next. After the “rare” multilooking, the interferogram phase is unflattened again, resulting in the rare unflattened interferogram (RareUnFlatIntf).

Additional looks are needed to get to the point that complex averaging no longer provides significant benefit over averaging of geolocated height values. Because of the narrow spatial extent of many water bodies under observation, however, the multilooking should be done adaptively using knowledge of the surface classification (e.g., water or land). This implies that classification should be performed next, and geolocation should not be done until after “medium” averaging is completed.

MakeClassMaps

The MakeClassMaps module is applied to the rare interferogram (RareUnflatIntf) and includes many algorithms: computing the so-called coherent power, detecting water, estimating water fractions, flagging dark water, and computing layover related uncertainties. This results in a multivalued classification mask as well as a continuous water fraction estimate for each rare interferogram pixel (ClassMaps).

MakeMedIntf

The rare interferogram (RareUnFlatIntf) is then flattened again and adaptively multilooked using the multivalued classification mask to perform medium-level smoothing of the measurements, which is needed to improve the precision of the height reconstruction and geolocation. The shape-adaptive multilooking smooths pixels of the same detected class over a fixed-sized window, resulting in a medium interferogram (MedFlatIntf). Note that many canonical multilooking approaches were investigated, including Lee Filters and non-local approaches [probably should cite these here…?]; however, the simple adaptive multilooking provides a good trade-off between complexity and performance, while enabling consistency with the detected water mask as well as direct control over the spatial extent of the smoothing.

UnwGeoloc

Heights and locations for each medium pixel (MedPixFull) are then generated by phase unwrapping and geolocating the data to obtain the medium interferogram (MedFlatIntf).

CompGeophysCorr

Next, geophysical references and height corrections (e.g., geoid, Earth tides, media delays, etc.) are computed for each pixel in the geolocated medium interferogram (GeophysCorr).

MakePixC

The geolocated medium interferogram is then pruned to exclude pixels that are not near water (based on both the detected water and a prior inclusion mask or pruning mask), as well as trimmed to exclude the overlap of the SLC tiles. Many additional intermediate quantities are also collected, and the pixel cloud product is produced.

Table 1. High-level description of the functions used to generate the L2_HR_PIXC product.

Function Name	Description
MakeUnflatSLC	Computes unflattened SLC images (UnflatSLC) by unflattening the interferometric phase of the SLC images (L1_HR_SLC).
MakeRareIntf	Computes rare unflattened interferogram (RareUnflatIntf) by reflattening the unflattened SLC images, multilooking and downsampling the interferogram and reference locations in azimuth, and then unflattening again.
MakeClassMaps	Computes classification maps (ClassMaps) by applying water detection, estimating pixel-wise water fraction, flagging dark water, and flagging layover.
MakeMedIntf	Computes the medium interferogram (MedFlatInt) by flattening the phase and performing multilook-smoothing of the rare interferogram adaptively using the multivalued classification mask.
UnwGeoloc	Computes heights and locations (MedPixFull) for each medium pixel by unwrapping the phase, estimating absolute phase ambiguity, and geolocating the measurements.
CompGeophysCorr	Computes geophysical references and height corrections (GeophysCorr) (e.g., geoid, tides, media delays, etc.) for each pixel.

MakePixC	Computes the final L2_HR_PIXC product by pruning the medium geolocated interferogram to only keep the pixels near water.

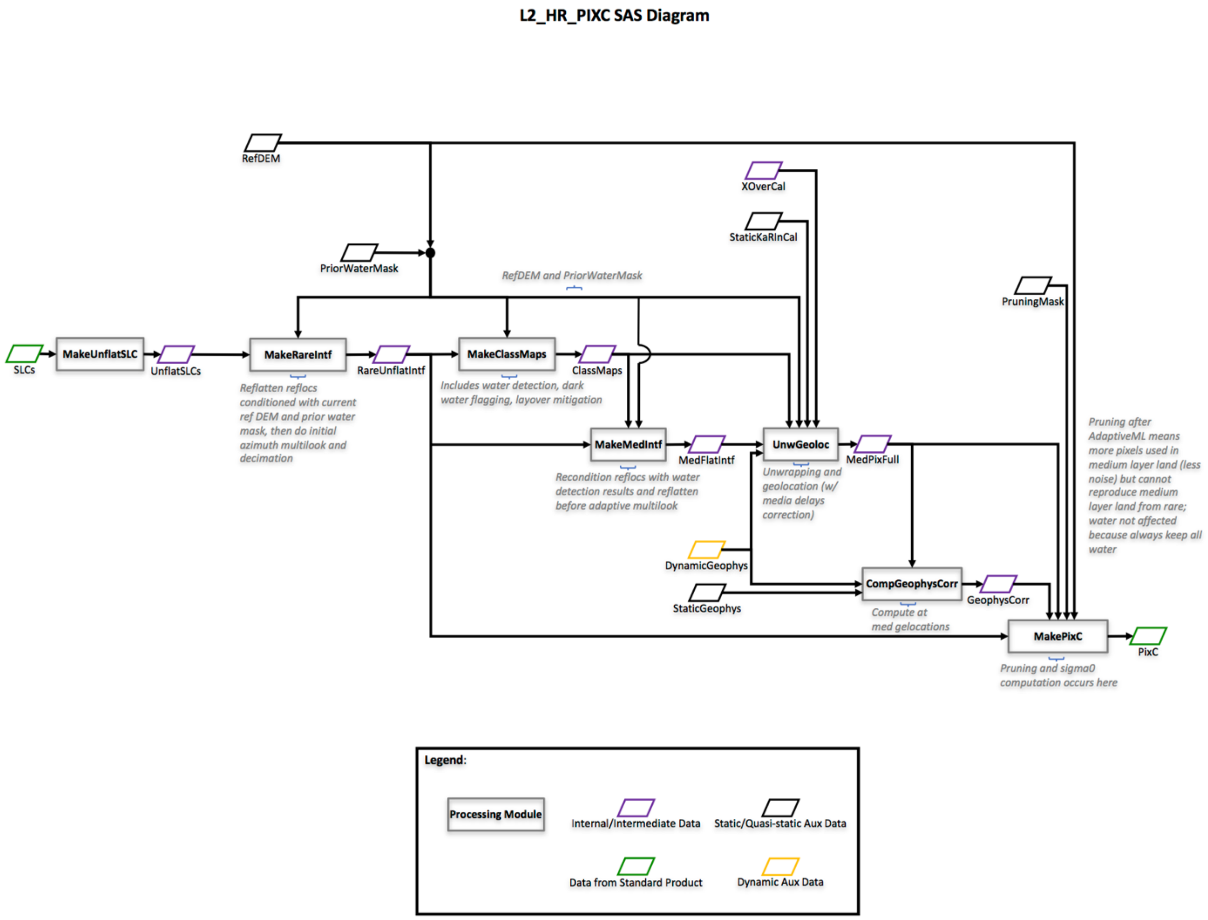


Figure 2. Flow diagram of the Level 2 Pixel cloud processing steps (functions) used to generate the L2_HR_PIXC product (TODO: update this figure, Dynamic Geophys and XOverCal aux data goes into Make RareIntf too).

2.3.1 Illustrative Example

An example using the SWOT HR simulator is presented here to illustrate the data as it propagates through the pixel-cloud processor. This realistic simulation is run over a unique scene over the Missouri River. The simulation methodology for generating this scene is the same as that used in the assessments of the algorithms and is presented in Appendix B. Figure 3 shows a zoomed-in region of the DEM elevation and land type mask used in this example. Throughout this example, a zoomed-in portion of the swath is shown in the various plots for the various stages of processing. The figures are presented in the order the data is generated in the functional flow, grouped under the sub-processor headings similar to the functional flow description above. Note that this is for illustrative purposes so only a subset of data variables are presented here.

MakeUnflatSLC (See Section 3.1)

Figure 4 shows the single look complex images for the reference and secondary channels. The magnitude is in grayscale, while the phase is plotted in color. The phase of each channel individually appears random, but the relative phase between the two (the interferometric phase) encodes information about the surface height.

MakeRareIntf (See Section 3.2)

Figure 5 shows the rare interferogram magnitude and phase before and after the rare unflattening. The grayscale is the interferogram magnitude, while the colors represent the interferometric phase. Note that the azimuth axis indices differ from the SLC images due to the rare multilooking and downsampling. Note also that the phase of the flattened interferogram is not exactly zero since the reference locations are not the truth locations. Furthermore, the phase fringes in the unflattened interferogram are oriented to vary primarily in the range (or cross track) direction and are not a function of the reference surface.

MakeClassMaps (See Section 3.3)

Figure 6 shows the coherent power and the corresponding coherent gain of the rare interferogram (These unique quantities are defined more precisely in Section 3.3.1). The coherent power is used for water detection and the coherent gain shows that there is a typical gain of about 3dB for water regions compared to a non-coherent combination of the reference and secondary channel power measurements. The gain is not always as good for non-water areas (which can actually help with water detection).

Figure 7 shows the detected water and the noisy water fraction estimate. The detected water roughly corresponds to bright pixels in the coherent power image. Note that in the mountainous region in the upper left portion of the images many small water bodies are detected. Some of these are correctly detected small mountainous lakes, but many of these bright features correspond to false detection issues where land-land layover increases the relative brightness of certain patches of pixels. Note also that the water fraction estimate at this resolution is quite noisy (and can take on values less than zero and greater than one). Nevertheless, this estimate is valuable to handle issues related to coherence time smearing (an artifact explained below).

Figure 8 shows the slant-plane projected prior water probability and the final binary prior water mask (The prior water probability or occurrence map is defined in Section 3.3.4). These are produced during the dark water flagging algorithm.

Figure 9 shows the layover impact flag. Note that most of the water areas are expected to not be significantly impacted by layover though there are some exceptions.

Figure 10 shows the low-coherence, detected water flag. Note that the areas in this image where there is detected water that has low coherence primarily occur in the mountainous regions in the upper left of the image. This is because the effect of layover is to increase the brightness (making it more likely to be detected), but often will reduce the coherence (since multiple energy from multiple places, typically with different phases, are integrated into the same range bin).

Figure 11 shows the final multivalued classification and an extra-zoomed portion of the estimated and truth multivalued classification images. Note that the detected water generally corresponds well to the truth water. Further, the regions of true dark water also generally correspond to the same regions of detected dark water.

MakeMedIntf (See Section 3.4)

Figure 12 shows the medium interferogram (magnitude and phase), as well as the interferometric coherence. Note that the speckle variability is significantly reduced in the medium interferogram compared to the rare interferogram, while preserving sharp contrast between land/water boundaries. The coherence image is also regularized similarly. The coherence image is primarily used to estimate the phase noise standard deviation, but can also be used as an independent measurement for science user purposes (e.g., distinguishing volume scattering from vegetation vs surface scattering).

UnwGeoloc (See Section 3.5)

Figure 13 shows the phase unwrapping regions and spatially unwrapped phase. Each region gets an absolute phase ambiguity assigned to it and the unwrapped phase is converted to an absolute phase that can be used to geolocate to latitude, longitude, and height above the ellipsoid

Figure 14 shows the geolocation images of latitude, longitude, and height above the ellipsoid for each pixel in the slant-plane. The bottom two images are the same just with a different color scaling to emphasize the slope over the river.

Figure 15 shows the slant-plane images of the phase noise standard deviation and the estimated height error standard deviation. The height error standard deviation is not reported in the product directly, but is computed from the phase noise standard deviation and the sensitivity of the height with respect to phase. Note that the estimated height (and geolocation) errors are generally low (a few cm) for water areas, but can be several meters for non-water regions.

MakePixC (See Section 3.7)

Figure 16 shows a three-dimensional representation of the geolocation and classification for a few azimuth lines for zoomed in portion of the river, with error bars also reported. The top image shows a view from above, while the bottom image shows a view from a moderate angle. The colors correspond to the multivalued classification, while the gray lines are the 3D-geolocation estimated error vector (computed from the phase noise standard deviation scaled by the sensitivity to the phase of the latitude, longitude, and height). Note that the error vectors are oriented orthogonal to the range vector and the azimuth direction. Therefore, from above they look like they are oriented in the cross-track direction (mostly longitude), but also have a height component.

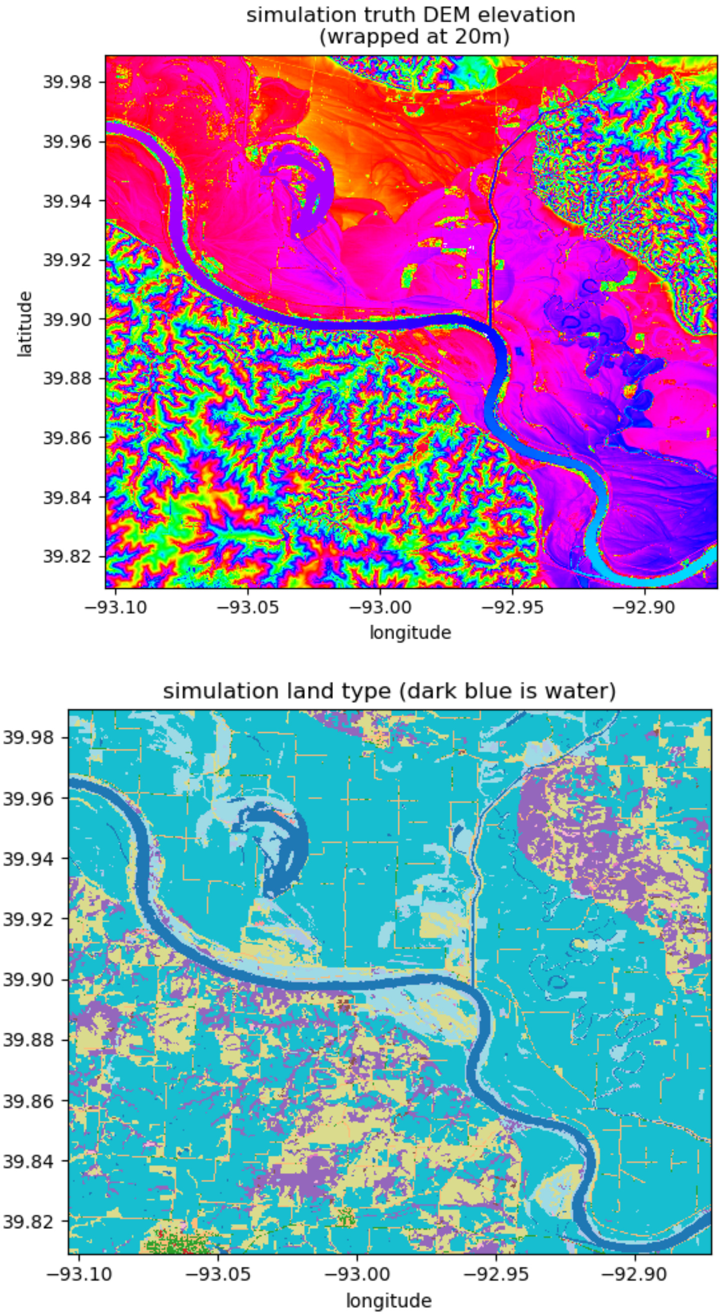


Figure 3. Simulation input dem elevation wrapped at 20m (upper), and land type mask (lower) for the Missouri River example.

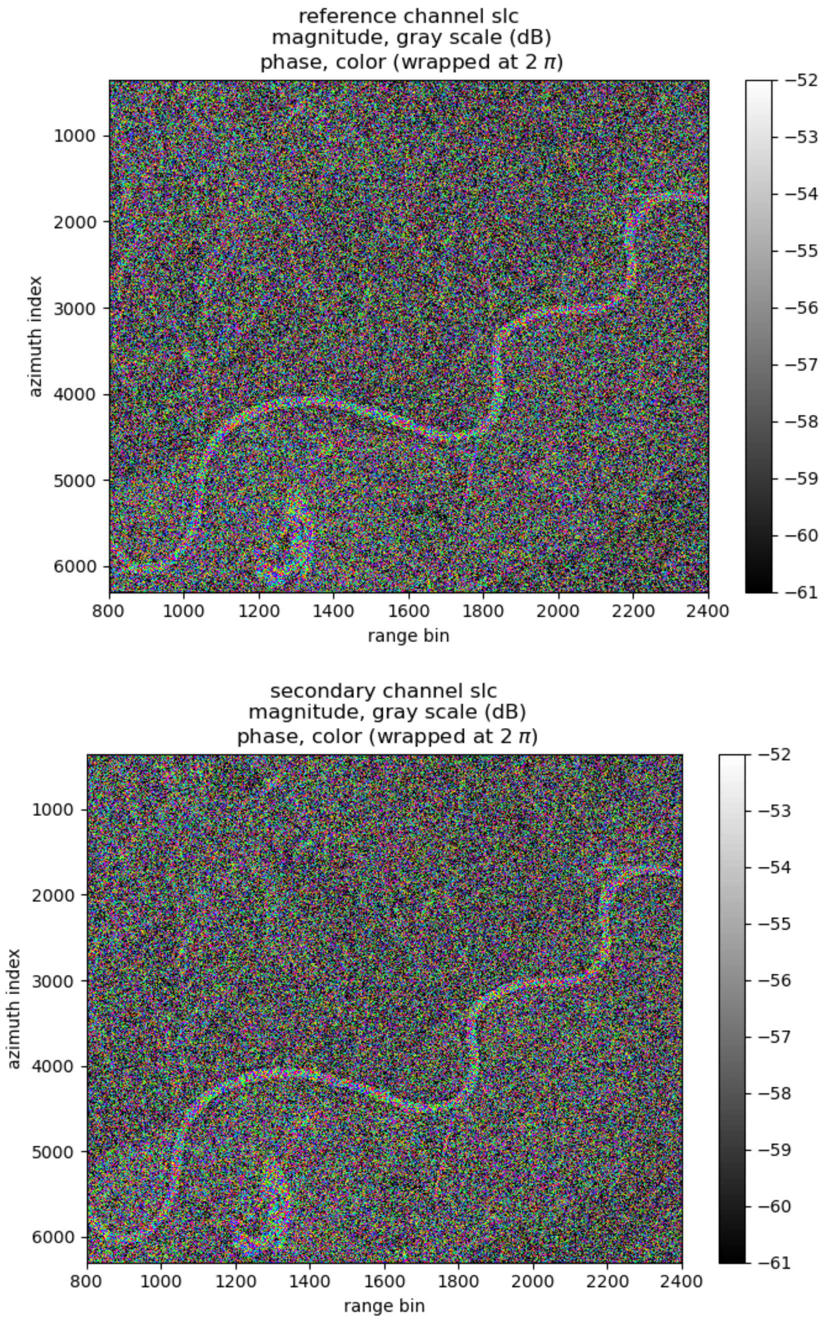
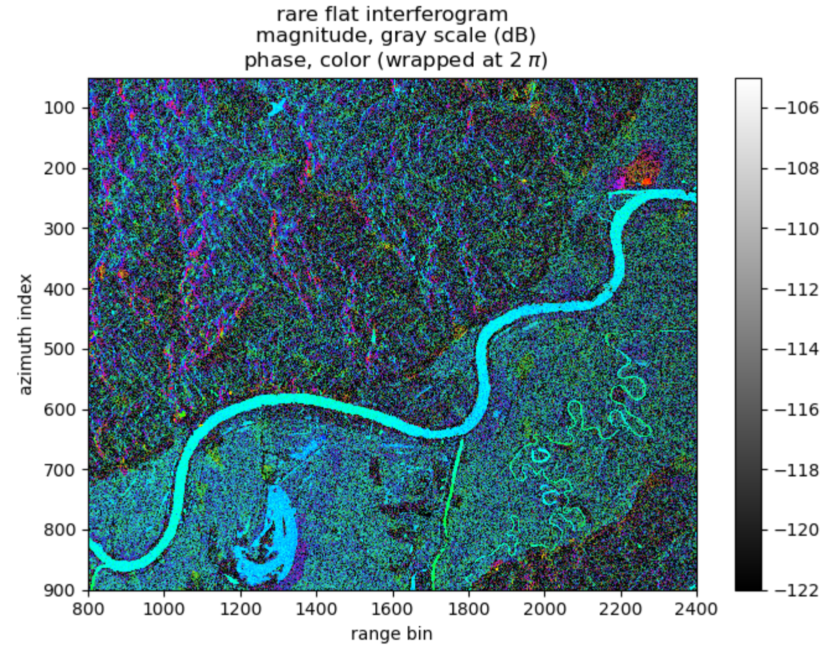


Figure 4. Slant-plane single look complex (SLC) images of the reference channel (upper), and the secondary channel (lower) for the Missouri River example. Note that the slant-plane image is oriented differently than the ground projection images (approximately flipped vertically). These quantities are produced by the SLC unflattening algorithm described in Section 3.1.



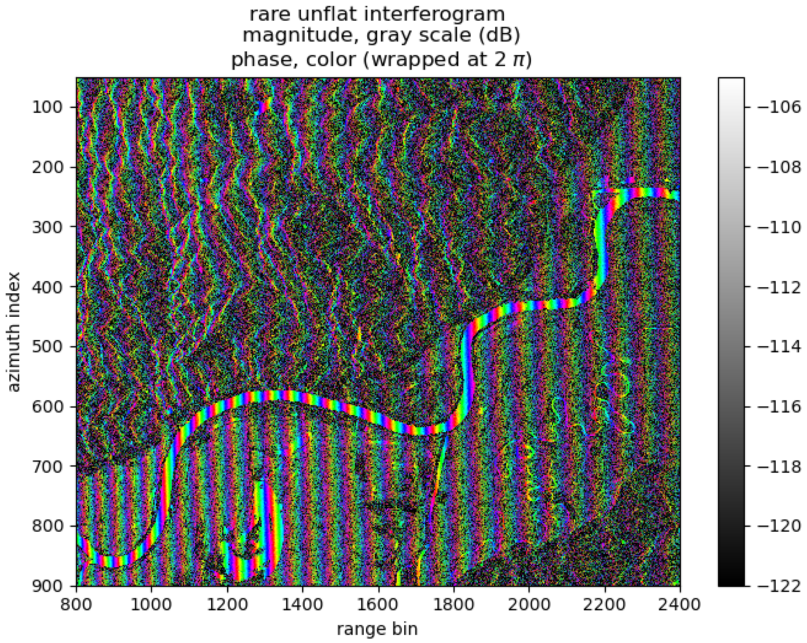


Figure 5. Slant-plane images of the rare flat interferogram magnitude and phase (upper), and the rare unflattened interferogram magnitude and phase (lower) for the Missouri River example. These quantities are produced by the rare interferogram algorithm described in Section 3.2.

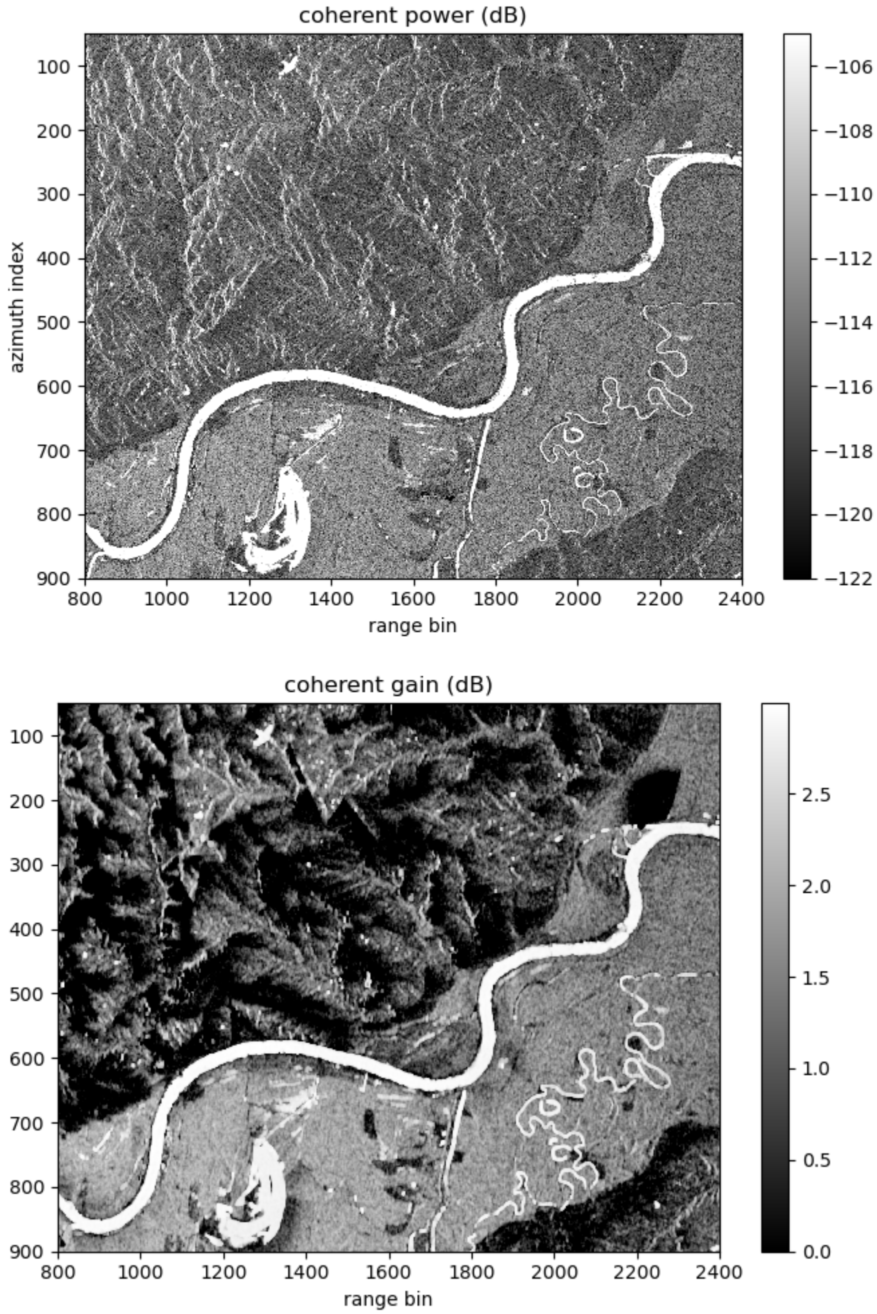


Figure 6. Slant-plane images of the rare interferogram coherent power (upper), and the coherent gain (lower) for the Missouri River example. These quantities are produced by the coherent power computation algorithm described in Section 3.3.1.

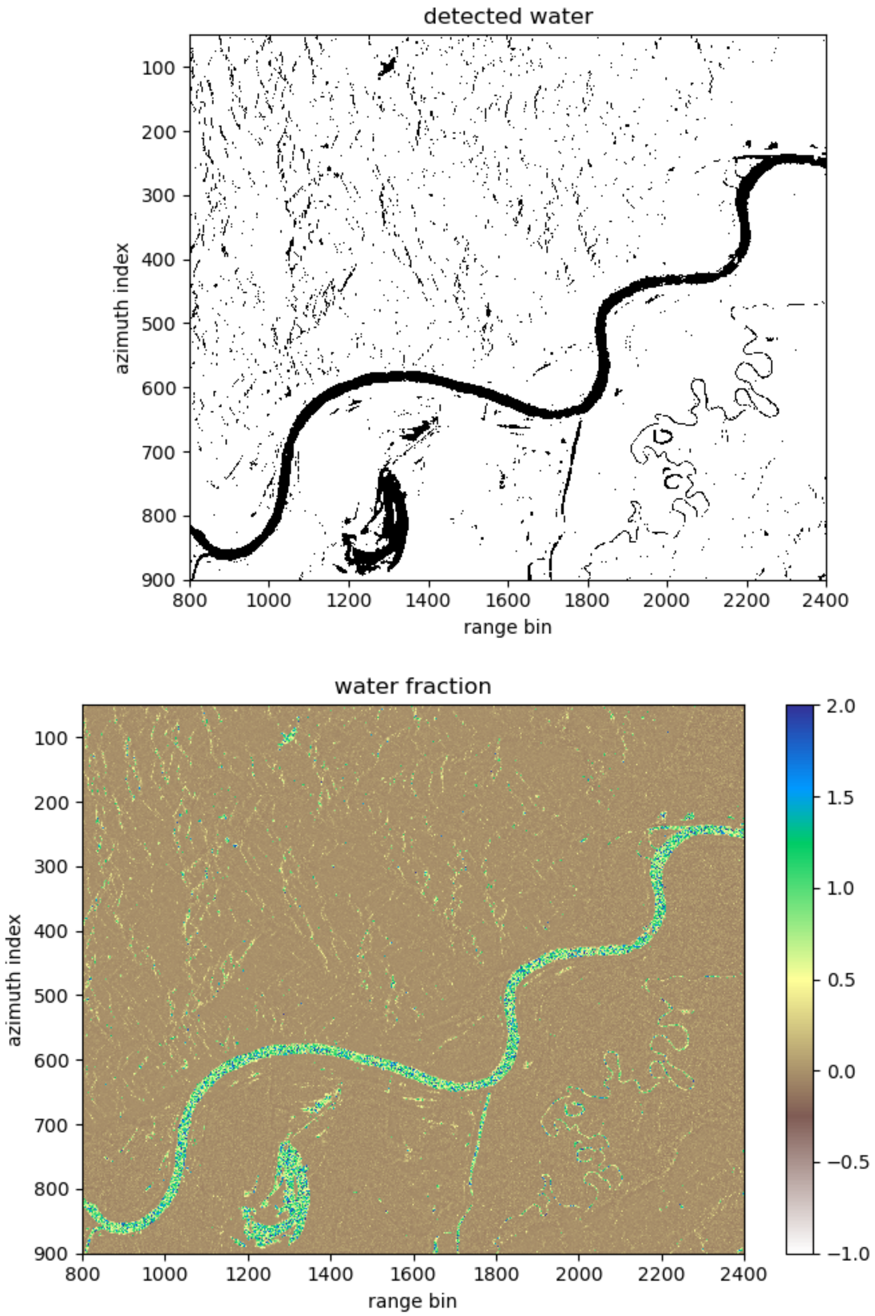


Figure 7. Slant-plane images of the detected water (upper), and the estimated water fraction (lower) for the Missouri River example. These quantities are produced by the water detection algorithm described in Section 3.3.2 and the water fraction estimation algorithm described in Section 3.3.3.

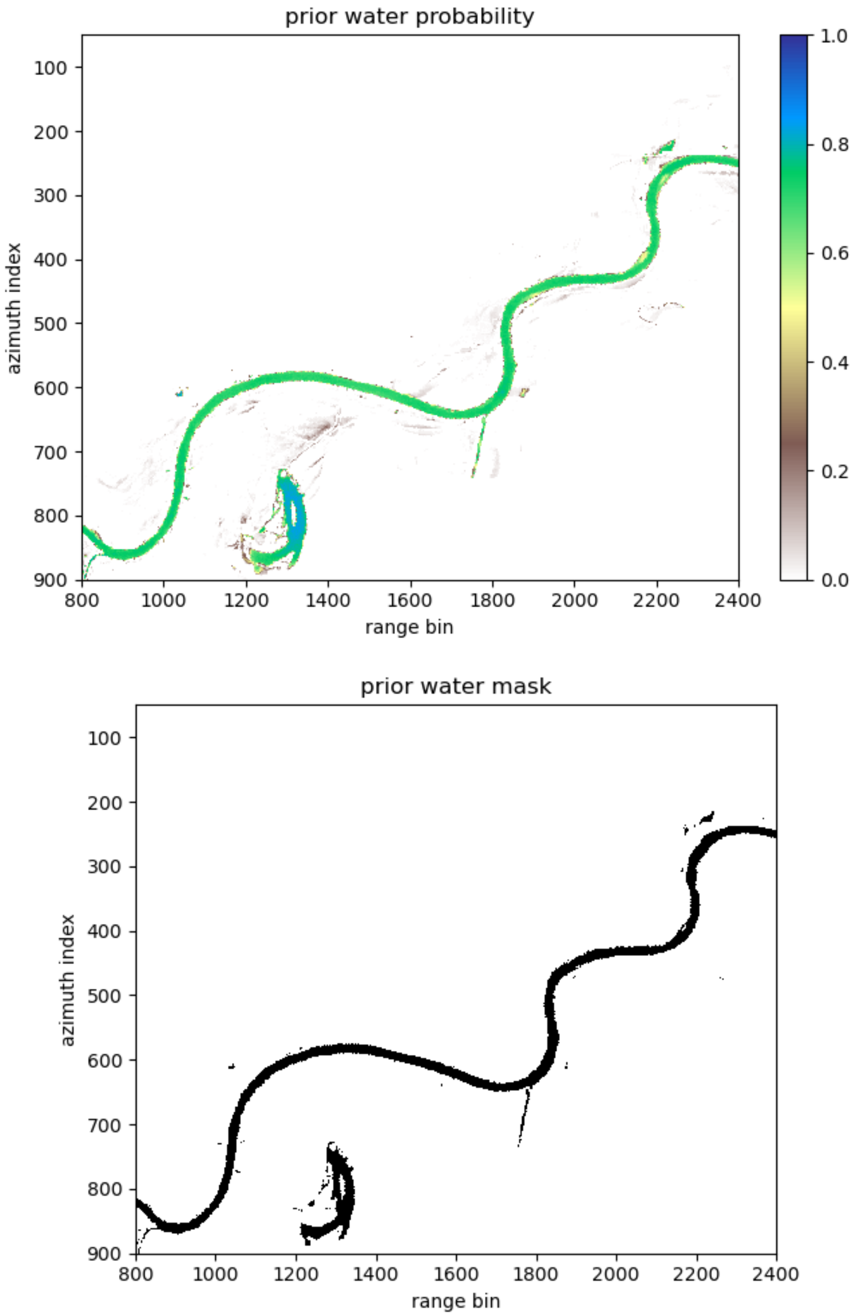


Figure 8. Slant-plane images of the prior water probability (upper), and the corresponding prior water mask (lower) for the Missouri River example. Note that the prior probability and mask differ from the truth water mask as well as the noisy SWOT data, especially in areas with narrow water features. This is likely due to the resolution limitations of the prior data used to make the mask. These quantities are produced by the dark water flagging algorithm described in Section 3.3.4.

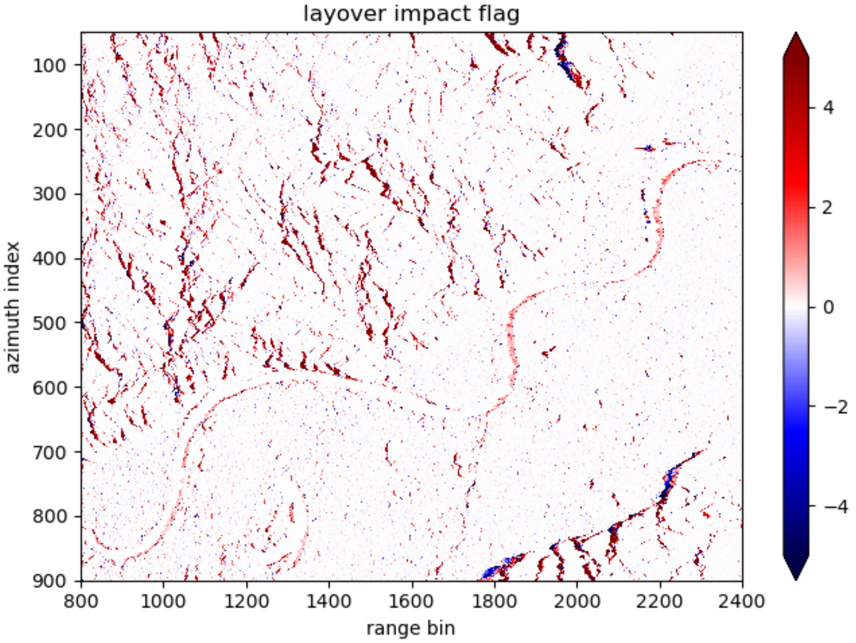


Figure 9. Slant-plane image of the layover impact flag for the Missouri River example. This quantity is produced by the layover flagging algorithm described in Section 3.3.5.

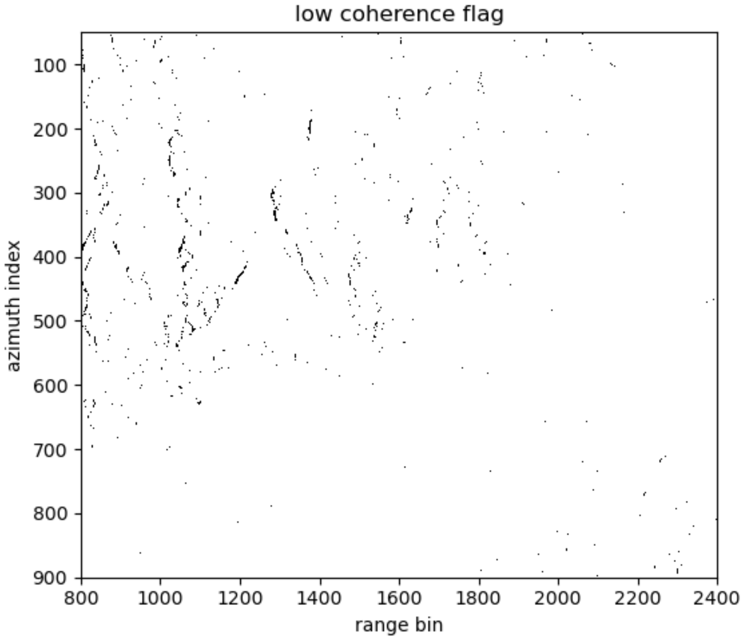


Figure 10. Slant-plane image of the low coherence detected water flag for the Missouri River example. This quantity is produced by the low-coherence flagging algorithm described in Section 3.3.7.

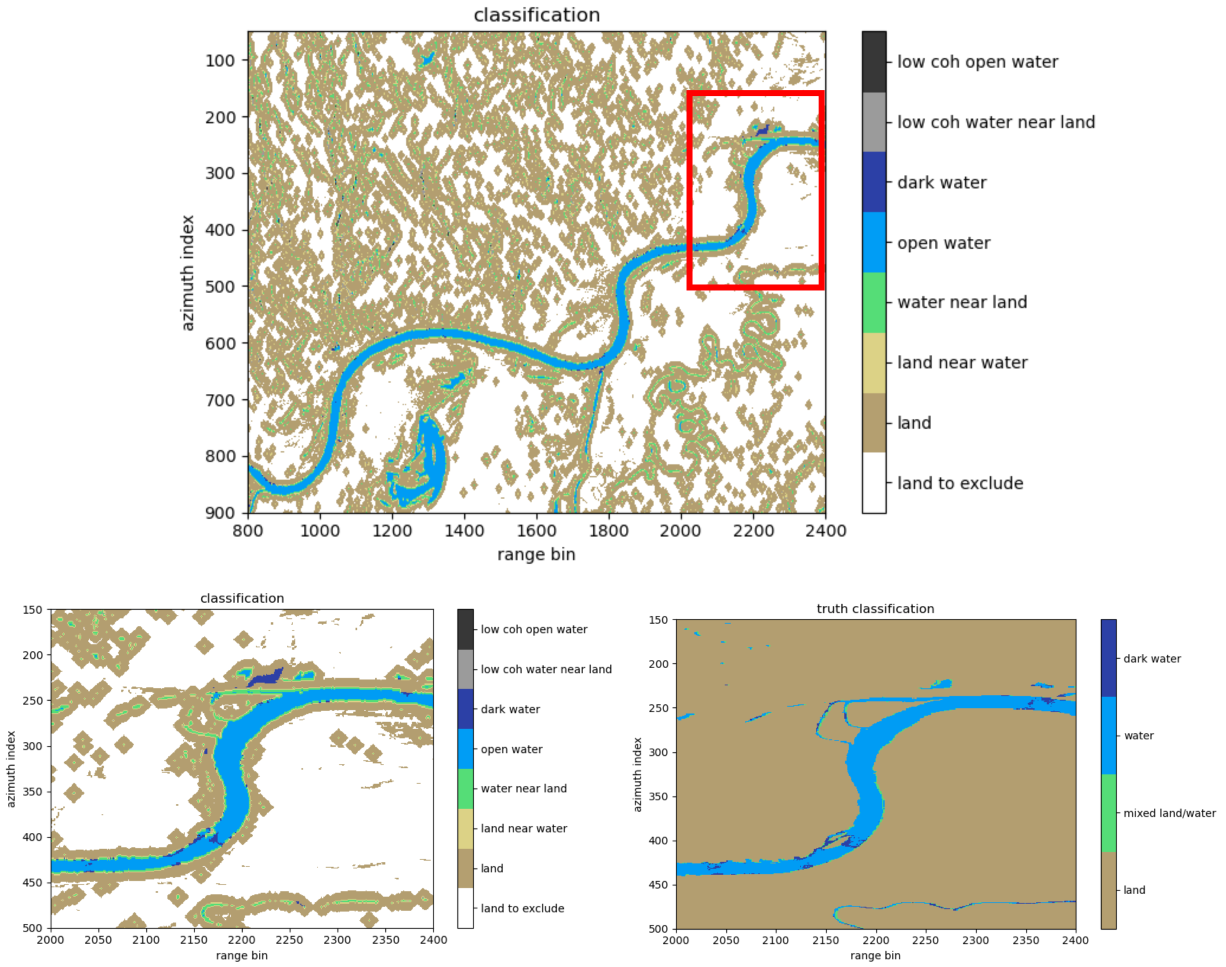


Figure 11. Slant-plane images of the rare interferogram multivalued classification (upper), and zoomed estimated multivalued classification (lower left) along with the corresponding truth classification (lower right) for the Missouri River example. The multivalued classification quantity is produced by the algorithm described in Section 3.3.8.

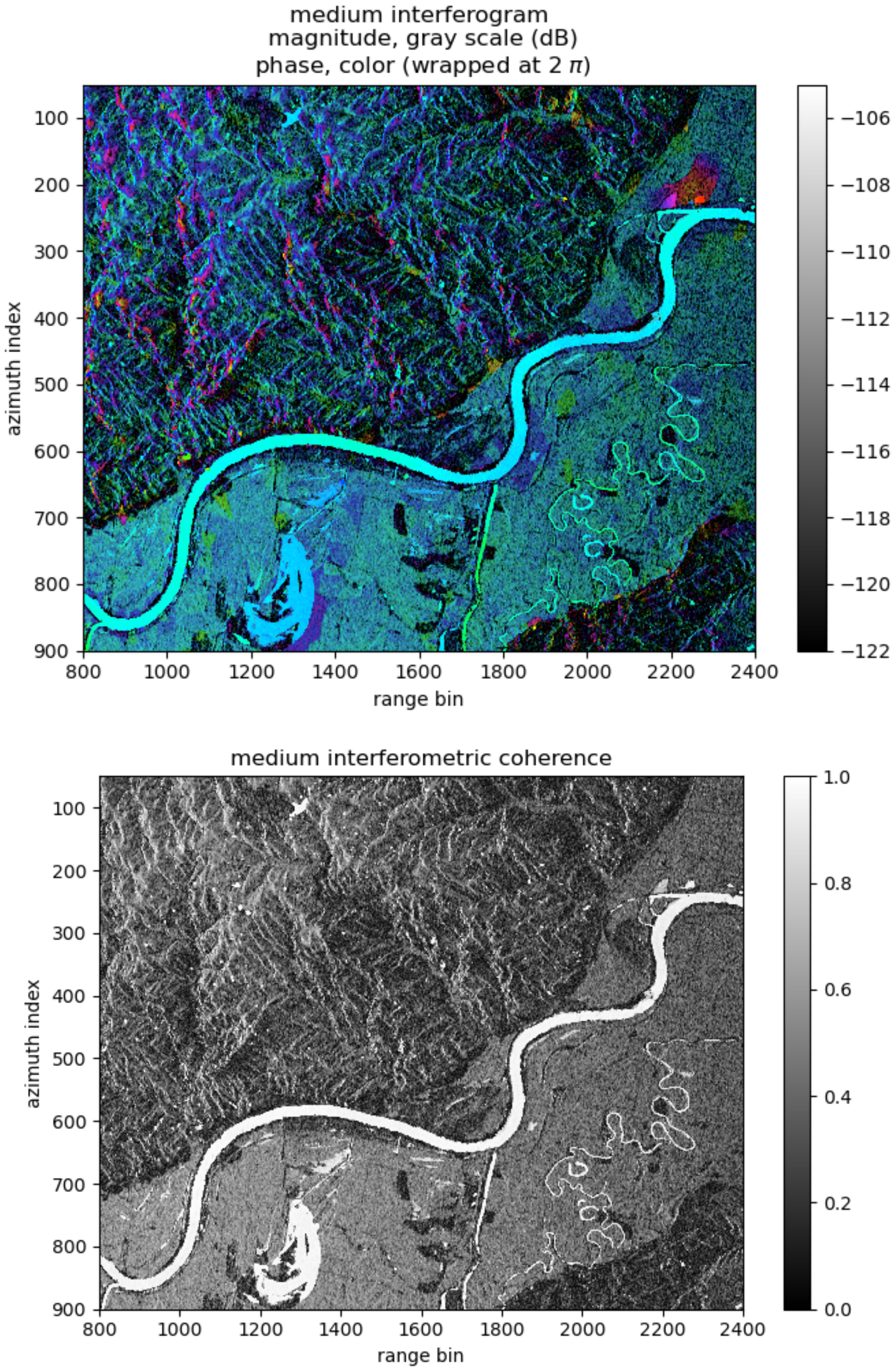


Figure 12. Slant-plane images of the medium interferogram magnitude and flattened phase (upper), and the medium interferogram coherence (lower) for the Missouri River example. These quantities are produced by the medium interferogram algorithm described in Section 3.4.

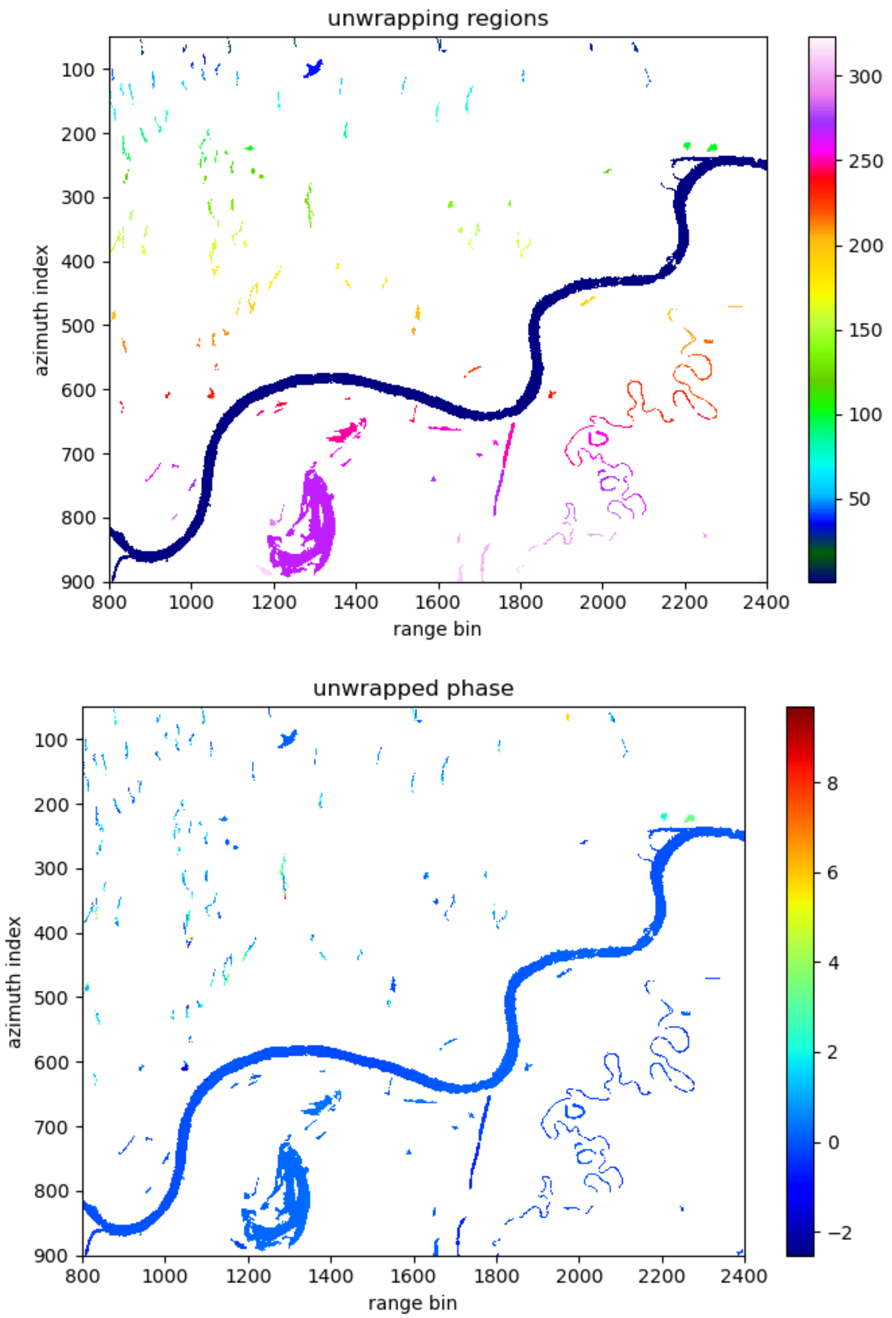


Figure 13. Slant-plane images of the phase unwrapping region map (upper), and the spatially unwrapped phase over each region (lower) for the Missouri River example. These quantities are produced by the phase unwrapping algorithm described in Section 3.5.

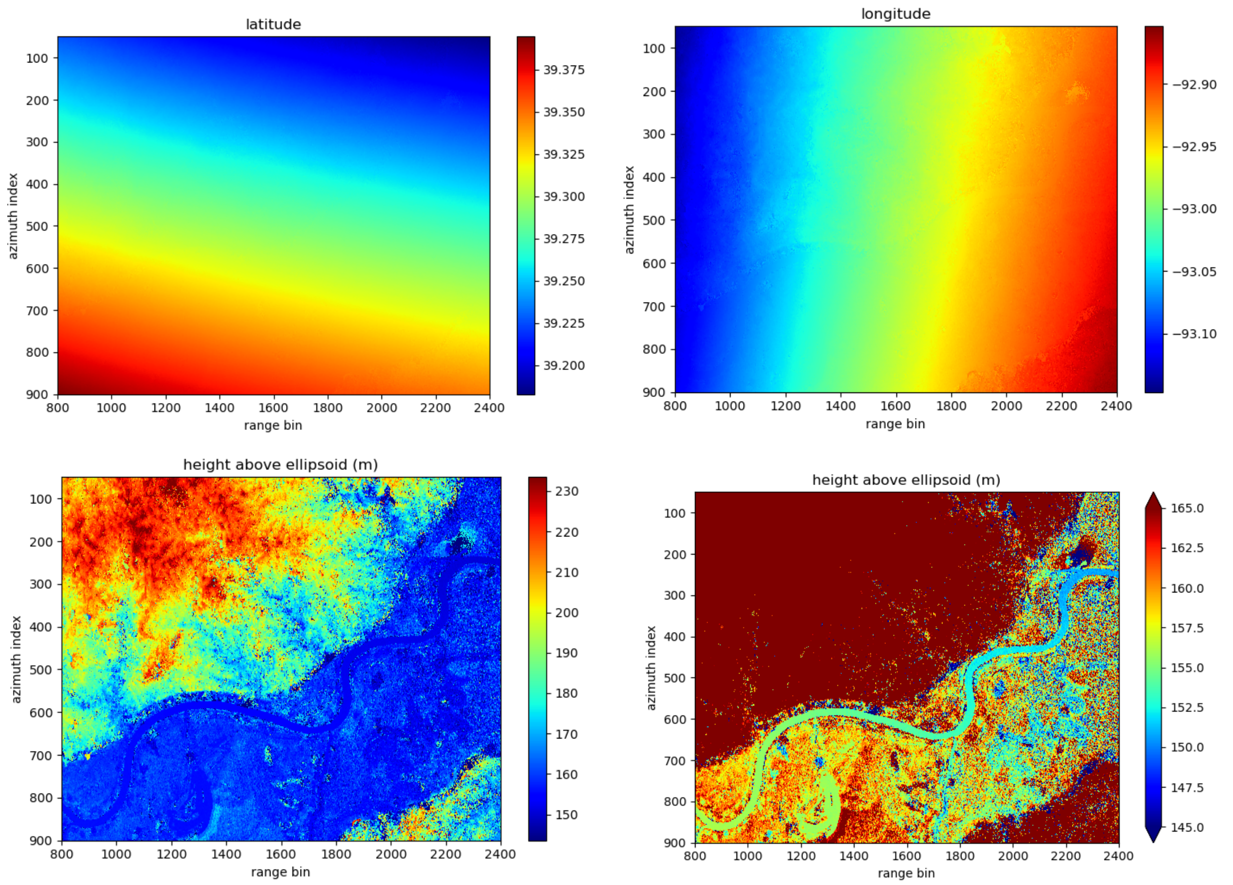


Figure 14. Slant-plane images of estimated latitude (top left), longitude (top right), height (bottom left and bottom right) for the Missouri River example. These quantities are produced by the geolocation algorithm described in Section 3.5.4.

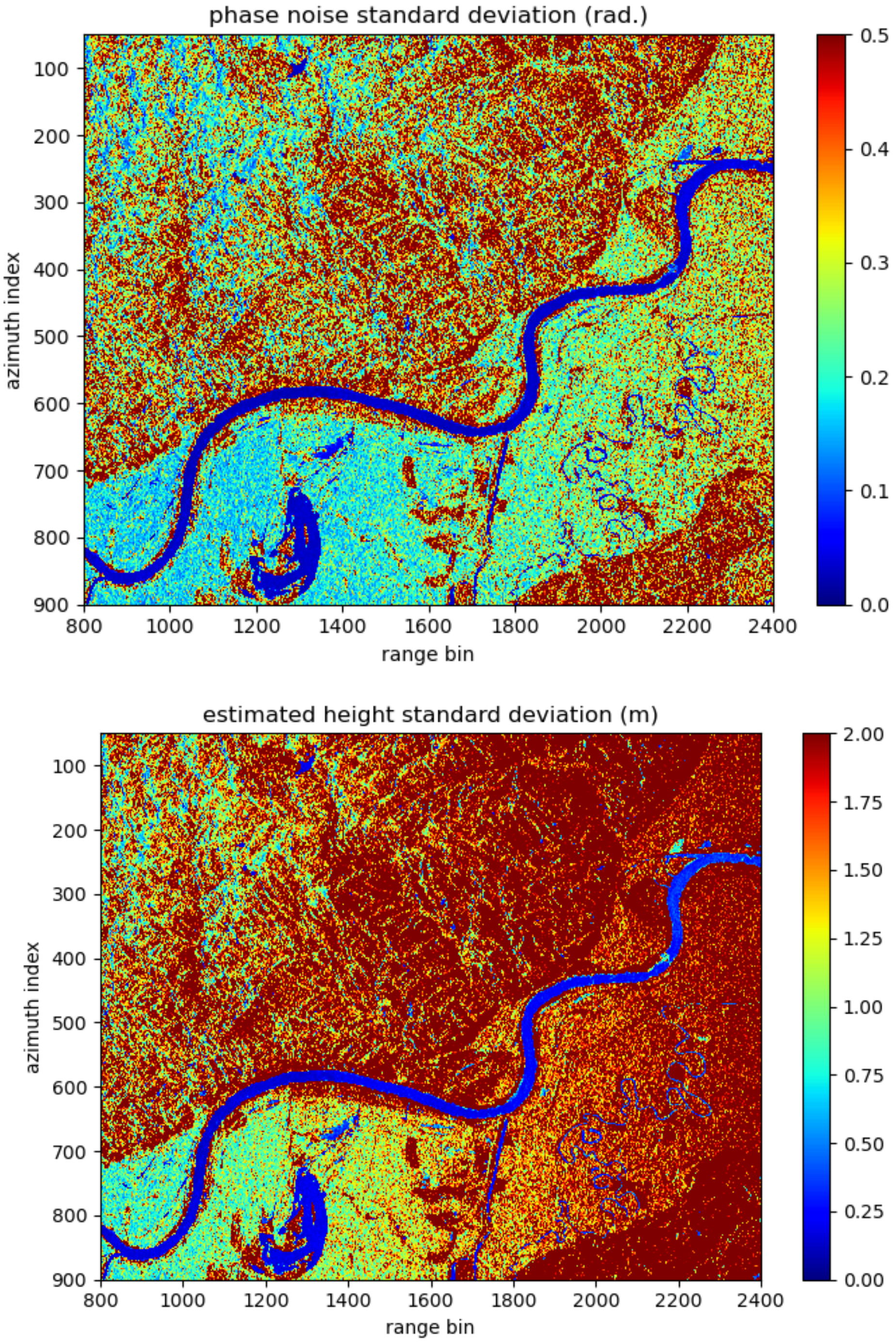


Figure 15. Slant-plane images of the phase noise standard deviation, which is derived from the interferometric coherence (upper), and the resulting estimate of the height error standard deviation (lower) for the Missouri River example. These quantities are produced by the geolocation algorithm described in Section 3.5.4.

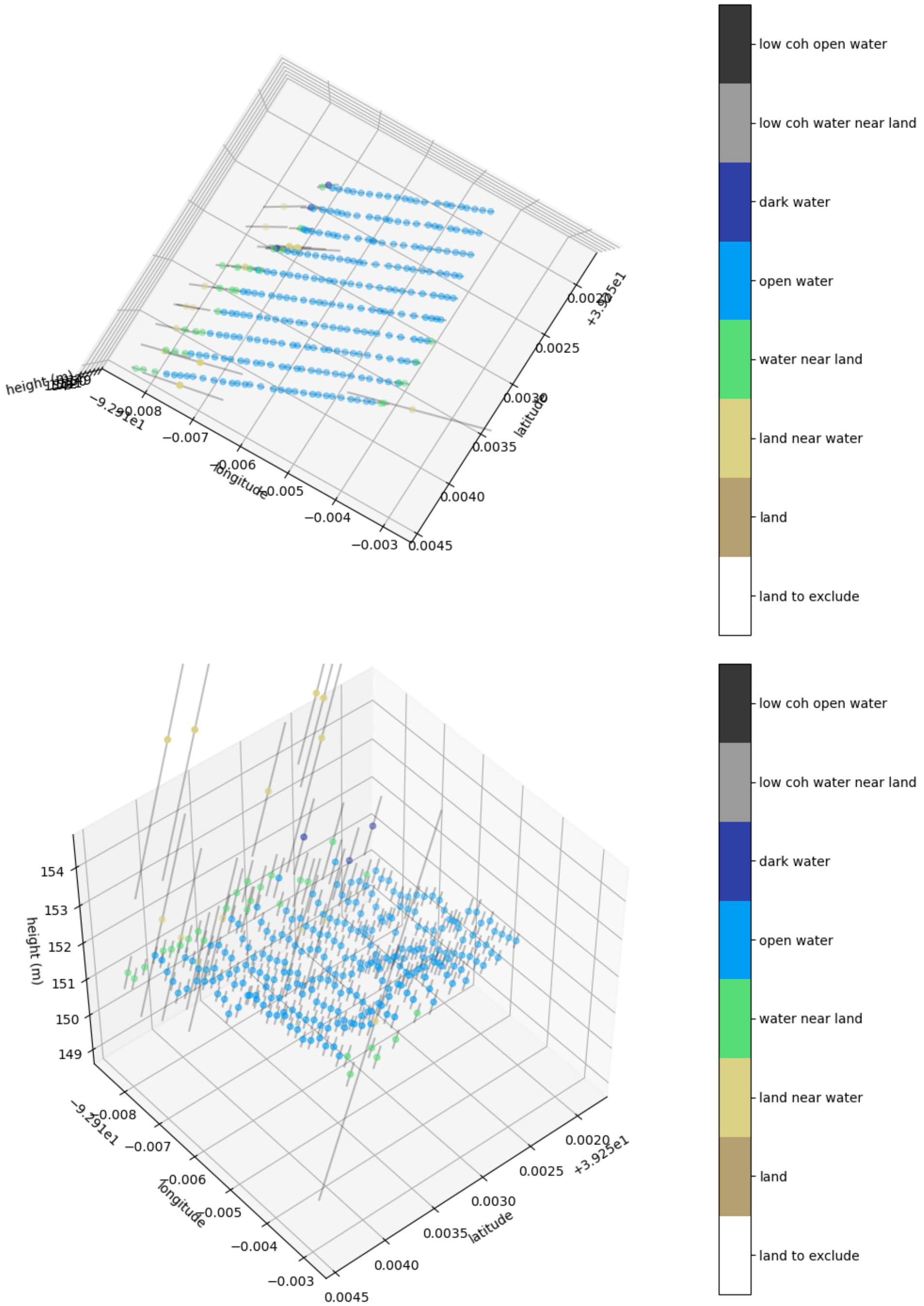


Figure 16. Three dimensional scatterplots of the water pixels for a zoomed in portion of the Missouri River example. Gray lines indicate the estimated geolocation error vector. These quantities are produced by the geolocation algorithm described in Section 3.5.4.

3 Algorithm Descriptions

This section describes the algorithms in the pixel-cloud processing flow illustrated in Figure 2. Each of the following subsections describes a module in the flow diagram.

3.1 MakeUnflatSLC

3.1.1 Purpose

The purpose of this function is to remove the effective phase flattening done in the SLC processor. The SLC processor does not explicitly apply a flattening process, rather it focuses the SLC images to a grid defined by the reference surface, which has the effect of producing a phase that is flat relative to that surface. In various stages of the pixel cloud processing, different reference surfaces (or reference locations) are applied (even when using the same Auxiliary Reference DEM input as the SLC processor). Therefore, this unflattening step is applied so that the data may be re-flattened in subsequent processing steps using different reference surfaces that are better suited to the pixel-cloud algorithms (at various stages of the procsses). This unflattening of the phase of the SLC data is done using the same reference GrDEM used during SLC processing, as provided in the L1B_HR_SLC data product. Recall that Figure 4 in Section 2.3.1 shows an example of the SLC images for a simulated river scene.

3.1.2 Input Data

Description	Source
SLC slant-plane complex images, satellite orbit time varying parameter (TVP) data, SLC reference DEM data (GrDEM), radiometric calibration images, and the noise power measurements	L1B_HR_SLC

3.1.3 Output Data

Description: UnflatSLC
Unflattened SLC slant-plane images for the two interferometric channels
Satellite orbit TVP data (simply passed through)
Radiometric calibration images (simply passed through)
Noise power measurements (simply passed through)

3.1.4 Mathematical Statement

The relative phase between the two SLC images is rotated in the complex plane so as to remove the relative phase flattening resulting from the SLC processing. The reference phase is computed by generating reference locations using the orbit and GrDEM provided with the SLC product. The same method (and code implementation) that is used in the SLC processor to generate the SLC grid and reference locations is used in the PIXC processor to re-compute the reference locations for this initial unflattening step [11].

An illustration of reference locations for a set of range contours is depicted in Figure 17. Note that some range bins have multiple intersections with the terrain; in such cases, the reference location is chosen to be the one closest in cross-track to the nadir track. The object in the upper left of the plot represents the SWOT spacecraft. The reference phase is proportional to the range difference between the two channels to a given target on the reference surface. After removing the phase flattening from the SLC processor (i.e., re-inserting the GrDEM reference phase) in the MakeUnflatSLC module, the resulting interferometric phase is no-longer a function of the reference DEM, but is directly a function of the location of the true surface relative to the KaRIn antennas, such that even a perfectly smooth, level surface would have cross-track variations in interferometric phase.

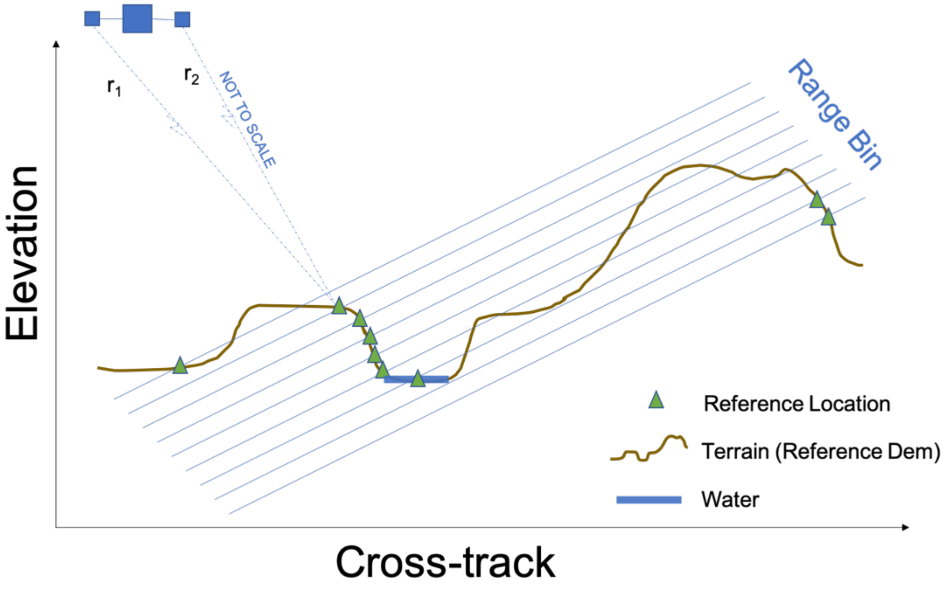


Figure 17. Illustration of reference locations. A related figure with modified or “conditioned” reference locations is plotted in Figure 19.

The GrDEM reference locations are computed under the assumption that the scattering targets on the ground are on the surface defined by the GrDEM. As the choice of which SLC image (or a combination of both) to rotate in phase is arbitrary, the SLC-GrDEM-derived reference phase (ϕ_{slc}) for each “raw” slant-plane pixel (x) is applied to the secondary SLC image $Z_2(x)$ to rotate its phase relative to the other image, the reference SLC image $Z_1(x)$. Note that the term reference SLC here is not related to the use of the word reference in other contexts later on (e.g., reference DEM, reference surface, reference locations etc). The meaning of the term reference is to be understood from the context in which it is used. Note that we use a single variable x to represent an index in a 2-D raw slant-plane image grid. The +*y* channel (e.g., the channel that receives on the +*y* antenna) is always used as the reference channel regardless of whether the +*y* or -*y* antenna transmits or whether the satellite is in a yaw-flipped state. +*y* and -*y* refer to the spacecraft coordinates where the +*y* direction is to the right of the spacecraft when the yaw is zero [TODO: cite user guide once written]. The reference channel is monostatic (transmits and receives on the same antenna) and secondary channel is bistatic (receives the same transmit signal, but on the other antenna). Ignoring common phase terms between the two channels, the reference and secondary SLC images are complex-valued images that can be represented as

$$Z_1(x) = |Z_1(x)| \cdot e^{-j\frac{2\cdot\pi}{\lambda}\left(r_{1,t}(x) + r_{1,t}(x)\right)} \tag{3.1.4.1}$$

$$Z_2(x) = |Z_2(x)| \cdot e^{-j\frac{2\cdot\pi}{\lambda}\left(r_{1,t}(x) + r_{2,t}(x) + \Delta r(x)\right)} \tag{3.1.4.2}$$

where $j = \sqrt{-1}$ represents the unit imaginary number, λ is the radar wavelength, and $r_{1,t}(x)$ and $r_{2,t}(x)$ are the range to the *true* target from the reference channel and secondary channel effective phase centers, and $\Delta r(x) = r_1(x) - r_2(x)$ is the difference between the range to the *reference* target from the reference channel and secondary channel effective phase centers. The equation to unflatten the secondary SLC can be expressed as

$$\tilde{Z}_2(x) = Z_2(x) \cdot e^{-j\phi_{slc}(x)} \tag{3.1.4.3}$$

where $\tilde{Z}_2(x)$ is the unflattened secondary SLC, and the reference phase is computed as

$$\phi_{slc}(x) = -\frac{2\cdot\pi}{\lambda} \cdot \Delta r(x) \tag{3.1.4.4}$$

This is consistent with the interferogram convention described below (Equation (3.2.4.1)) where the interferogram is formed by multiplying the reference channel by the conjugate of the secondary channel. The phase of the resulting “unflattened” interferogram $\phi_u(x)$ can be expressed as

$$\begin{aligned}
\phi_u(x) &= \measuredangle\left(Z_1(x) \cdot \tilde{Z}_2^*(x)\right) && (3.1.4.5)\\
&= \measuredangle\left(Z_1(x) \cdot Z_2^*(x)\right) + \phi_{slc}(x) && (3.1.4.6)\\
&= -\frac{2\cdot\pi}{\lambda}\left(r_{1,t}(x) - r_{2,t}(x) - \Delta r(x) + \Delta r(x)\right) && (3.1.4.7)\\
&= -\frac{2\cdot\pi}{\lambda}\,\Delta r_t(x) && (3.1.4.5)
\end{aligned}$$

where $\Delta r_t(x) = r_{1,t}(x) - r_{2,t}(x)$. Therefore, the unflattened interfereogram phase is no longer a function of the reference surface, but only a function of the true surface topography.

This procedure undoes the SLC flattening, and puts back in the interferometric phase fringes that were taken out during SLC processing. This approach enables reflattening in the next step with a different DEM.

3.1.5 Accuracy

This operation does not introduce additional errors. The operation is precise to within floating point precision.

3.2 MakeRareIntf

3.2.1 Purpose

This function transforms the two SLC images into an azimuth-multilooked interferogram. There are several steps involved in this algorithm including: 1) interfering the two SLC images (Unflattened), 2) flattening the phase with a flat-Earth reference surface (which minimizes spatial phase artifacts over water surfaces in the subsequent multilooking due to reference DEM spatial variations in elevation), 3) multilooking and downsampling in the azimuth (along-track) dimension, 4) unflattening the phase once again with the multilooked reference locations, 5) applying crossover and some geophysical corrections. Note that this minimal level of multilooking is tailored to be suitable for subsequent water detection, but the additional, adaptive-multilooking step (in MakeMedIntf) is needed to optimize the phase noise for geolocation. Thus, the rare interferogram (phase) is not geolocated, although it is on the same grid as the medium interferogram geolocations. Recall that Figure 5 in Section 2.3.1 shows the rare interferogram for a simulated river scene.

3.2.2 Input Data

Description	Source
Unflattened SLC slant-plane images, satellite orbit TVP data, radiometric calibration images, and noise power measurements (UnflatSLC)	MakeUnflatSLC.
Geophysical inputs: wet troposphere, dry troposphere, ionosphere	DynamicGeophys
Crossover calibration inputs	XOverCal

3.2.3 Output Data

Description: RareUnFlatIntf
Rare Interferogram image on the rare-grid posting, including the unflattened complex interferogram, the two individual channel powers, and the number of looks
Satellite orbit TVP data (simply passed through)
Radiometric calibration images on the rare grid
Noise power measurements (simply passed through)

3.2.4 Mathematical Statement

The reference SLC $Z_1(x)$ and unflattened secondary SLC $\tilde{Z}_2(x)$ are interfered by multiplying each pixel of the reference SLC image by the conjugate of the secondary SLC image, producing the unflattened "raw" complex interferogram

$$I(x) = Z_1(x) \cdot \tilde{Z}_2^*(x) \tag{3.2.4.1}$$

Note that strictly speaking, $I(x)$ represents the complex interferogram at the raw posting; however, the term "interferogram" is in this document also used as shorthand to refer to the many parameters existing at the stage of processing of the interferogram formation. For example, the term "raw interferogram" can refer to the quantity $I(x)$ as well as other, related quantities such as the two channel powers $P_1$and P_2 which are computed as

$$P_i(x) = |Z_i(x)|^2 = Z_i(x) \cdot Z_i^*(x), for\ i \in \{1,2\} \tag{3.2.4.2}$$

Furthermore, the term "interferogram" may be used more generally to refer to the intermediate products (e.g., RareUnflatIntf and MedFlatIntf), which include additional data variables (e.g., the thermal noise, radiometric calibration, and TVP data). The use of the term interferogram in multiple contexts is unavoidable. It will be clear from the context which meaning of the term "interferogram" is being used.

The phase is reflattened using newly created reference locations on the raw grid computed from a flat-Earth reference surface (e.g., ellipsoid) with a constant average height over the observed scene. This approach of using flat-Earth reference locations minimizes reference location phase artifacts introduced due to imprecise knowledge of the topography, and also enables the reference locations for water, which is relatively flat, to be multilooked with minimal artifacts. These reference locations are used to compute a flat-Earth reference interferometric phase $\phi_f(x)$ that is used to flatten the interferogram by complex multiplication with the conjugate of a unit-magnitude reference interferogram with the reference phase.

$$I_f(x) = I(x) \cdot e^{-j\phi_f(x)} \tag{3.2.4.3}$$

Once the "raw" interferogram is flattened, simple azimuth multilooking is applied. This can be thought of as simply averaging a given number of consecutive interferogram pixels in azimuth (7 in the nominal configuration), and downsampling equivalently. The rare multilooking is applied to the complex interferogram $I_f(x)$, as well as the reference and secondary channel powers $P_1(x)$ and $P_2(x)$. Furthermore, the flat Earth reference location Earth Centered Earth Fixed (ECEF) (x, y, z) locations are also multilooked and downsampled using the same method.

There are some subtleties related to how to define the reference locations after multilooking. For example, one could average the reference location ECEF (x, y, z) locations or the range, Doppler, and phase values, or one could just create new reference locations at the rare grid using the same reference DEM. However, since we are using flat-Earth reference locations and only multilooking over a small extent the differences between these approaches are negligible. The baseline algorithm applies reference location averaging of the ECEF (x, y, z) locations.

Note that although azimuth multilooking is done by a factor of 7, the effective number of looks (i.e. by which factor the variance is reduced) is actually around 4, because the SLC images are oversampled in azimuth and hence partially correlated from sample to sample. That is, the statistical distribution of the rare pixels can be represented as though 4 independent looks are taken, even though 7 actual pixels are multilooked.

This amount of smoothing for the rare interferogram is chosen to be approximately the minimum amount of smoothing needed to give robust statistics for water detection (e.g., a gamma distribution with a high enough shape parameter as opposed to one more extremely left-skewed like the exponential distribution of a single look). It also corresponds to approximately the maximum amount of simple azimuth smoothing that can be done to maintain resolution fine enough to resolve water features on the order of the science goals (i.e., to resolve ~50 m features with < ~25 m posting).

Once the rare, flattened interferogram is produced along with the rare flat-Earth reference locations, the phase is again unflattened. Finally, crossover calibration and some geophysical corrections are applied to the interferogram (as detailed below in subsection 3.2.4.1). This results in the rare, unflattened interferogram, which is passed through the rest of the processor and is reported in the L2_HR_PIXC product.

3.2.4.1 Calibration and Geophysical Corrections

The rare, unflattened interferogram is corrected for the crossover calibration and a few geophysical corrections. This subsection overviews the mathematical description of these corrections at a high level. Readers should see the GECO software documents for more details on the interpolation approaches to the geophysical and crossover corrections [TODO: revise and cite GECO docs]. Note that except for the media attenuation correction, this stage of corrections is intended to apply preliminary "bulk" approximations to minimize artifacts and errors in the remaining processes. Later in the processor there are "delta" corrections applied on top of this "bulk" correction using more precise geolocations. There are four classes of corrections that are applied at this stage of processing: 1) a bulk range correction (from media delays and crossover calibration), 2) an interferometric phase correction (from media delays and crossover calibration), 3) a radiometric attenuation correction (from media attenuation), and 4) an interferometric baseline dilation correction (from the crossover calibration). Note that these corrections are based on geophysical or calibration auxiliary data that are all slowly varying spatially.

First, consider the atmospheric/media attenuation correction. This is obtained from the interpolated vertical atmospheric attenuation $a_{vert}(x)$ looked up at the flat Earth reference location lat/lon positions and the azimuth row time. This quantity is then modified to take into account the slant path variation as a function of incidence angle $\theta_{inc}(x)$. This can be expressed mathematically as

$$a_{atten}(x) = \frac{a_{vert}(x)}{\cos\left(\theta_{inc}(x)\right)} \qquad (3.2.4.1.1)$$

where (x) refers to the pixel location on the rare slant-plane image grid (as opposed to the raw grid (x)). This media attenuation is used to scale the radiometric calibration (Xfactor) for both the reference and secondary channels.

Now consider the range correction related to the media delays. The following fields are interpolated to the rare flat Earth reference locations: the wet troposphere range delay $r_{wet}(x)$, the dry troposphere range delay $r_{dry}(x)$, and the ionosphere total electron content $c_{te}(x)$. These are used to compute a pixel-wise range correction for the reference ($r_{media,1}(x)$) and secondary ($r_{media,2}(x)$) channels as follows

$$r_{media,i}(x) = \frac{r_{wet}(x) + r_{dry}(x) + c_{iono} \cdot c_{te}(x)}{\cos\left(\theta_{el,i}(x)\right)} \qquad (3.2.4.1.2)$$

where $\theta_{el,i}(x)$ is the elevation angle of the reference ($i = 1$) or secondary ($i = 2$) channel to the reference target and

$$c_{iono} = -40.3e^{16} \cdot \left(\frac{\lambda}{c}\right)^2 \cdot c_{alt} \qquad (3.2.4.1.3)$$

where λ is the radar wavelength and c is the speed of light. The scaling factor of c_{alt} (nominally set to 0.8) is applied to account for the contribution of the topside ionosphere between the SWOT altitude and the GNSS altitude at which the Global Ionospheric Map (GIM) was generated. The scale factor of $40.3e^{16}\ \frac{m}{s^2}$ is a conversion factor from total electron content (TEC) and frequency to range delay. The bulk constant range delay is selected as the median of $r_{media,1}(x)$.

Now consider the phase correction that is related to the differential range delay. This is computed from the reference and secondary media delays as

$$\phi_{media}(x) = \frac{2\pi}{\lambda}\left(r_{media,1}(x) - r_{media,2}(x)\right) \qquad (3.2.4.1.4)$$

The crossover calibration also contributes to the range and phase corrections as well as provides an interferometric baseline correction. The various crossover correction model parameters are interpolated to the azimuth row time. These are then expanded to the full slant-plane image (by replication across the range indices for each azimuth line). This results in the following slant-plane crossover correction parameter images: 1) the slope 1 term $s_{1,xover}(x)$, 2) the slope 2 term $s_{2,xover}(x)$, 3) the baseline height error term $b_{xover}(x)$, 4) the left timing height error term $t_{l,xover}(x)$, and 5) the right timing height error term $t_{r,xover}(x)$. For details on the crossover calibration model and the role of these parameters see the crossover calibration documentation [cite crossover docs here].

The crossover correction to range is obtained directly from the timing height error term (either the left or right depending on which swath the tile is over)

$$r_{xover}(x) = \begin{cases} t_{r,xover}(x) & if\ swath\ is\ right \\ t_{l,xover}(x) & if\ swath\ is\ left \end{cases} \qquad (3.2.4.1.5)$$

where the difference between height error, timing error, and range error are neglected (which is a difference of a few millimeters).

The crossover correction of the phase drift is computed from the slope terms as

$$\phi_{drift}(x) = \begin{cases} \dfrac{(s_{1,xover}(x)+s_{2,xover}(x))\cdot 10^{-6}\cdot D_{ct}(x)}{\frac{dh}{d\phi}(x)} & if\ swath\ is\ right \\ \dfrac{(s_{1,xover}(x)- s_{2,xover}(x))\cdot 10^{-6}\cdot D_{ct}(x)}{\frac{dh}{d\phi}(x)} & if\ swath\ is\ left \end{cases} \qquad (3.2.4.1.6)$$

where D_{ct} is the cross-track distance in meters and $\frac{dh}{d\phi}(x)$ is the sensitivity of height to phase assuming the flat Earth reference locations (whose computation is detailed below in Section (reference it)).

The interferometric baseline dilation correction is computed from the coefficient for quadratic height error due to baseline dilation as

$$b_d(x) = \frac{b_{xover}(x)\cdot D_{ct}^2(x)}{\frac{dh}{db}(x)} \qquad (3.2.4.1.7)$$

where D_{ct} is the cross-track distance in meters and $\frac{dh}{db}(x)$ is the sensitivity of height to baseline assuming the flat Earth reference locations (whose computation is detailed below in the Section (reference it)). This baseline dilation correction is used to adjust the 3D locations of the reference and secondary channel phase centers (e.g., the lever arms relative to the platform position), by scaling the length of the baseline (but not the orientation).

The final range correction is applied here as a constant, "bulk" offset, although a more refined "delta" correction is applied later on in the processor using more precise geolocations because the corrections are spatially varying and depend on the geolocations (see Section 3.4.4.1 and Section 3.5.4.3). The motivation for this is because both the media and crossover range corrections are mostly a constant offset with a small magnitude variation. The bulk range correction is obtained by taking the sum of the median of the media and crossover range correction terms above

$$r_{corr} = median_x\{r_{media,1}(x)\} + median_x\{ r_{xover}(x)\} = r_{media} + r_{xover} \quad (3.2.4.1.8)$$

where the median is taken over the whole slant-plane image and where the reference media delay is used (neglecting the small difference between the reference and secondary channels). This range correction is added to the reference starting range of the slant-plane images (e.g., the range corresponding to each range bin in the slant-plane images is offset).

The final phase correction is simply the sum of the media and crossover phase corrections

$$\phi_{corr}(x) = \phi_{media}(x) + \phi_{drift}(x) \quad (3.2.4.9)$$

for every pixel (e.g., not a bulk correction). This phase correction is applied to the flat interferogram

$$I_{corr}(x) = I_f(x) \cdot e^{j\phi_{corr}(x)} \quad (3.2.4.1.10)$$

before the final unflattening stage, thus preserving the correction in the rare unflattened interferogram that is used elsewhere in the processor.

3.2.5 Accuracy

The multilooking process in this function degrades the resolution while reducing noise. The use of a flat-Earth reference surface, as opposed to a DEM with more realistic topography, for reference locations reduces artifacts related to inaccuracies in reference DEM topography variations, but preserves phase variations from the true topography. That is, the flat-Earth reference surface does not introduce height variations of its own, because it is flat. Technically, the height error between truth and the reference height causes the flattened phase to have a ramp (or wrapped phase fringes) in the cross-track direction; however, this can be neglected when multilooking in azimuth only. Thus, in the azimuth direction, the flattened phase will have variations related primarily to the actual surface topography, and not the flat-Earth reference surface. These variations are averaged over in the multilooking process, resulting in a phase that may not correspond to a point that is actually on the true surface. Since water is generally flat, the artifacts for water areas are typically negligible, although this is not necessarily the case for the land pixels. Another effect introduced by azimuth multilooking occurs at water/land boundaries where land pixels are averaged with water pixels (e.g., mixed pixels). If the land is at a different height this can corrupt the water phase, similar to the effect of layover; however, the average power of mixed pixels is in between the water and land power, rather than the accumulation of both as occurs in layover. Nevertheless, the corrupting impact of the land surface is generally minor, as the water is generally much brighter than the land. One other case to consider is when there is an actual height discontinuity in a water surface, for example over a spillway, waterfall, or loch gate. If these cases are oriented so that the discontinuity occurs exactly in the azimuth direction, then the azimuth multilooking affectively averages the two phases, giving something in between for the one azimuth pixel. On the other hand, if the discontinuity is exactly in the cross-track direction, then azimuth multilooking is not impacted, although in this case there will generally be water-water layover corrupting the measurement.

The geophysical and crossover calibration corrections applied here employ a few approximations (e.g., applying a constant bulk range correction, using flat Earth reference locations for the interpolations, and neglecting the difference between height offset, timing offset, and range offset in the crossover range correction). However, these approximations are more than sufficiently accurate for the purposes for which they are applied at this stage in the processing. Furthermore, the each of these approximations (except for using approximate reference locations for the radiometric attenuation correction) have a more precise delta correction applied later in the processing chain that accounts for any deficiency here. For the radiometric calibration, there is some sensitivity near the swath edges when using the flat Earth reference locations; however, the errors are only expected to be within a few dB, which is generally within the typical range of accuracy expected for the radiometric calibration of radar imagery. This approximation impacts primarily the water detection algorithm; however, there is robustness built into that algorithm to account for uncertainty in the radiometric calibration. The estimate of the normalized radar cross-section will also be impacted, but is expected to be within typical uncertainty bars.

3.3 MakeClassMaps

Due to its complexity, this algorithm is split into the different pieces that are shown in Table 2 and Figure 18. The primary objectives of this algorithm are to detect water and flag issues such as dark water, low-coherence water, bright land, and layover.

Table 2. High-level description of the functions within the MakeClassMaps function.

Function Name	Description
MakeCohPow	Computes the rare coherent power (CohPower) from the rare interferogram.
WaterDetection	Computes a water map (WaterMap) by detecting water and computing the background land and water powers.
WaterFracEst	Computes the water fraction (WaterFrac) from the coherent power and the background land and water power estimates.
LayoverFlagging	Computes a layover map (LayoverMap) containing a layover impact quantity that is used for uncertainty estimation.
DarkWaterFlagging	Computes a dark water map (DarkWaterMap) by identifying water pixels that were not detected as water by the water detection algorithm.
BrightLandFlagging	Computes a bright land map (BrightLandMap) by identifying land pixels that are likely to be falsely detected as water by the water detection algorithm.
LowCohFlagging	Computes a flag for regions of low coherence, but bright, water (LowCohMap).
BuildClassMaps	Computes the water classification map (ClassMaps) by collecting and combining the various water and land classes and building the multivalued classification fields, including identifying land/water edges.

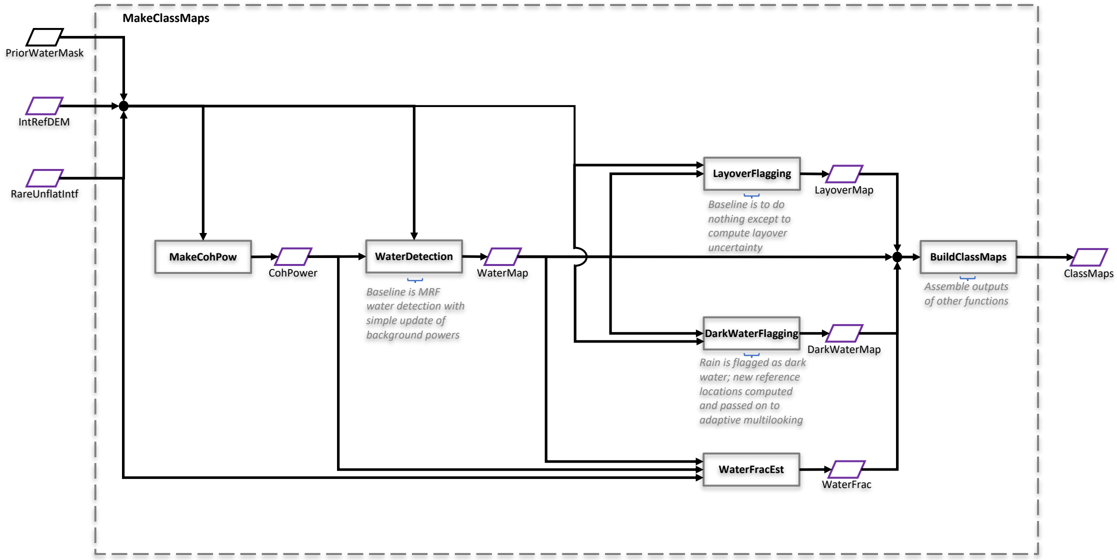


Figure 18. Flow diagram of lower-level algorithms within the MakeClassMaps function (TODO: update figure intRefDEM-> RefDEM).

3.3.1 MakeCohPow

This function computes the so-called coherent power slant-plane image. This quantity is used later in water detection and water fraction estimation. Recall that Figure 6 in Section 2.3.1 shows the coherent power image for a simulated river scene.

3.3.1.1 Input Data

Description	Source
High resolution reference DEM (RefDEM)	Static Auxiliary Data
Rare complex interferogram and two channel powers, satellite orbit TVP data (RareUnflatIntf)	MakeRareIntf
Prior water occurrence map (PriorWaterMask)	Static Auxiliary Data

3.3.1.2 Output Data

Description: CohPower
Rare coherent power slant-plane image
Rare prior-water-conditioned reference locations

3.3.1.3 Mathematical Statement

The coherent power (P_c) is computed for the rare interferogram in preparation for water detection and water fraction estimation. This power estimate can be thought of as a coherent combination of the two SLC channel powers where the phase of the secondary SLC image has been well flattened to align it with the phase of the reference SLC image. In principle, the coherent power can be computed directly from the SLC images and a coherent power reference phase $\phi_c(x)$ (different from the reference phases used above) by

$$P_c(x) = \left|\frac{Z_1(x)+ \tilde{Z}_2(x)\cdot e^{-i\phi_c(x)}}{\sqrt{2}}\right|^2 \qquad (3.3.1.3.1)$$

and then multilooking to the rare grid. However, at this stage we only have access to the rare interferogram, not the SLC images. Nevertheless, multiplying the term inside the absolute value by its complex conjugate, the coherent power computation reduces to an expression in terms of the unflattened complex interferogram, the two channel powers, and the reference phase as

$$P_c(x) = \frac{P_1(x)+P_2(x)}{2} + |I(x)| \cdot \cos\big(\angle I(x) - \phi_c(x)\big) \qquad (3.3.1.3.2)$$

where (x) refers to the pixel location on the rare slant-plane image grid (as opposed to the raw grid (x)), and $\phi_c(x)$ is a rare coherent power reference location phase. Note that $\phi_c(x)$ is derived from a "conditioned" version of reference locations that are computed from the reference DEM and is described below.

The use of coherent power, as opposed to the incoherent average or individual channel powers (P_1 and P_2), can provide a signal power boost relative to the thermal noise floor P_N. That is, consider the thermal signal to noise ratio (SNR) defined as

$$SNR_c(x) = \frac{P_c(x)}{P_N} = \frac{P_1(x)+P_2(x)}{2P_N} + \frac{|I(x)|}{P_N} \cdot \cos\left(\angle I(x) - \phi_c(x)\right) \qquad (3.3.1.3.3)$$

$$= \frac{P_1(x)+P_2(x)}{2P_N} + \frac{\sqrt{P_1(x)P_2(x)}}{P_N} \cdot \cos\left(\angle I(x) - \phi_c(x)\right) \qquad (3.3.1.3.4)$$

Now consider the case where $P_1(x) \approx P_2(x)$, which is a rough approximation for SWOT. For this case, we have

$$SNR_c(x) \approx \frac{P_1(x)}{P_N} + \frac{P_1(x)}{P_N} \cdot \cos\left(\angle I(x) - \phi_c(x)\right) \qquad (3.3.1.3.5)$$

$$= \frac{P_1(x)}{P_N}\left(1 + \cos\left(\angle I(x) - \phi_c(x)\right)\right) \qquad (3.3.1.3.6)$$

$$= SNR(x)\left(1 + \cos\left(\angle I(x) - \phi_c(x)\right)\right) \qquad (3.3.1.3.7)$$

Therefore, the "coherent-power" approach enables an effective thermal SNR gain of up to about a factor of 2 (or 3 dB) if the true surface topography is known well (e.g., the term in the parentheses goes to 2). However, if the reference phase estimate is poor, the SNR may actually decrease (e.g., the term in the parentheses can approach 0).

To handle the case where coherent power attenuates the signal, a filtering operation is performed to guarantee that there is no attenuation relative to the individual channel powers (i.e., that the gain is at least a factor of 1). In order to deal with speckle, this operation first spatially filters both the coherent power and the incoherent average of the two channel powers. Then the nonfiltered coherent power is selected if the filtered version is greater than the filtered incoherent power, otherwise the nonfiltered incoherent power is selected. This operation can be expressed as

$$\mathcal{P}_c(x) = \begin{cases} P_c(x) & if\ \hat{P}_c(x) \geq \sqrt{\widehat{P_1(x) \cdot P_2(x)}} \\ \sqrt{P_1(x) \cdot P_2(x)} & otherwise \end{cases} \qquad (3.3.1.3.8)$$

where $\hat{P}_c(x)$ is a spatially filtered version of the noisy coherent power $P_c(x)$ and $\sqrt{\widehat{P_1(x) \cdot P_2(x)}}$ is the spatially filtered version of the incoherent power $\sqrt{P_1(x) \cdot P_2(x)}$. The spatial filtering is employed to locally estimate the average power so as to mitigate the effects of speckle.

The computation of the coherent power relies on the absolute accuracy of the flattening phase, not its spatially-relative accuracy, so the flat-Earth reference location phase, previously computed in MakeRareIntf, is not generally accurate enough to produce a significant gain in the coherent power. Thus, another computation of reference locations with a more accurate DEM is needed here to produce the reference phase $\phi_c(x)$. In order to handle artifacts that can be introduced by DEM errors near land/water boundaries, the reference location approach used here also applies a "conditioning" step to smooth the water and land heights near expected water regions, based on the prior water mask. These prior-water-conditioned reference locations are also passed to the WaterDetection routine where they can be reused for projecting prior information about the surface into the slant-plane.

The "conditioning" refers to a height smoothing that makes the water and nearby land locally flat at about the same elevation as the water. This is done to try to minimize artifacts due to DEM uncertainty and water elevation fluctuations. Figure 19 illustrates the concept of conditioning the reference locations. Note that since the terrain elevation around water has been made flat and smoothed to have a similar height as the water height (dashed line), the reference locations (green triangles) tend to become more regular around the water features (compared to Figure 17). That is, there are fewer large jumps in height and cross-track location near the later edges that are often caused by layover or inconsistencies between the DEM topography and prior knowledge of the water locations. This is especially obvious when looking at the difference between Figure 17 and Figure 19 for the two reference locations farthest in cross-track (e.g., the right two triangles in Figure 17 are next to the right most water reference locations in slant-plane, but far in height/cross-track, while in Figure 19 the right most two conditioned reference locations are next to each other in both the slant-plane and in cross-track and have a similar height as the water). As a general rule, when the reference DEM is a little off, either in height or in location (or has a relatively coarse spatial resolution), large artifacts in the phase of water areas can result when the DEM is not conditioned (Figure 17). This is ameliorated when the reference locations are conditioned (Figure 19). Note that although the height of the land topography around water after conditioning may differ significantly from the truth, resulting in artifacts over land, the objective is to minimize distortions over water.

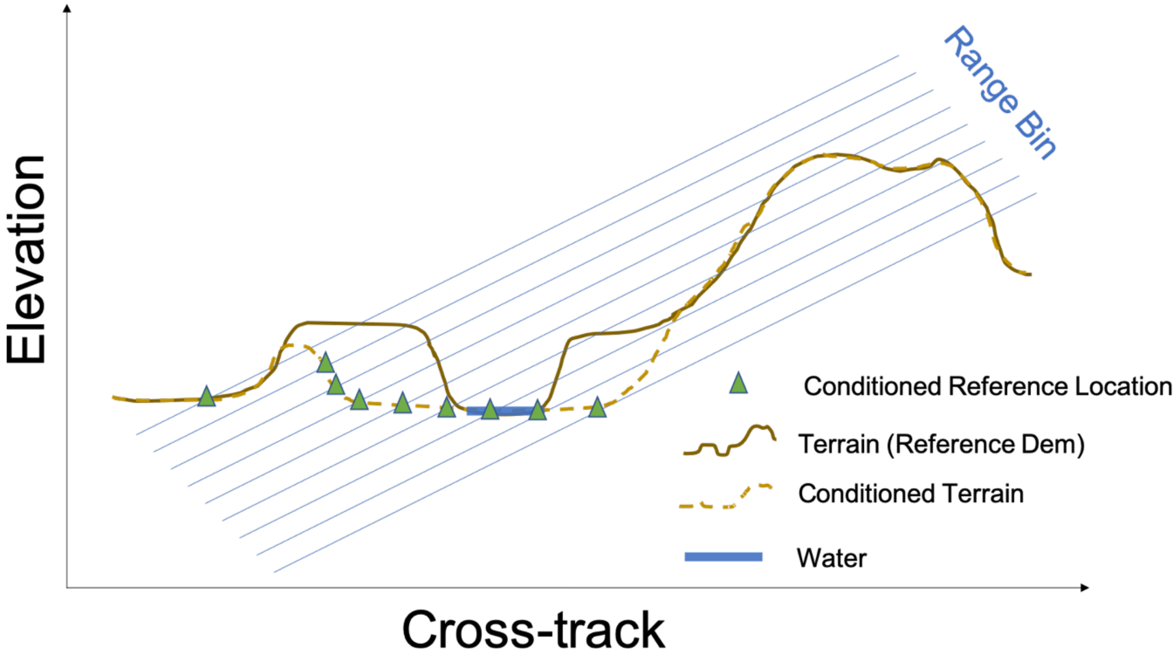


Figure 19. Illustration of reference location conditioning. Recall that a similar figure with the reference locations without conditioning is plotted in Figure 17.

There are two general approaches to apply the height smoothing for the conditioning step: 1) adjusting the elevations in the RefDEM directly before computing reference locations (as depicted in the figure above), or 2) adjusting the heights of the reference locations in the slant-plane and then recomputing the locations under applicable constraints on the measurement geometry. These approaches both produce the desired conditioned reference locations, but the second approach is computationally more efficient. Thus, the baseline algorithm applies approach 2 at this point in the processing chain. The reference location conditioning algorithm can be described as follows:

Reference Location Conditioning Algorithm:

1. Generate the standard RefDEM reference locations using the reference DEM and reference water mask.
2. Smooth land heights around water bodies in the slant-plane to estimate land height as the water height of the nearest water body.
   - (2-1) Apply image segmentation on slant-plane water mask.
   - (2-2) Apply weighted smoothing filter (51x51 pixels). The weighting is chosen to weight more heavily the heights from the water pixels with a similar value to the pixel in the center of the window, while effectively deweighting nearby water pixels from a different height distribution (e.g., from a spatially connected water feature at a different height such as the water on either side of a dam).
     - (2-2-1) Replace non-water pixel heights with the smoothed heights from nearest segment.
     - (2-2-2) Repeat (2-2) until 4 iterations.
3. Regenerate the reference locations for each pixel to be at the new height, but the same range-Doppler contour as the unflattened rare interferogram.
4. Apply the weighted smoothing height filter one more time (51x51 pixels), smoothing land and water together to minimize land/water edge artifacts.

3.3.1.4 Accuracy

The distribution of the coherent power before the filtering operation can be shown to be the same form as the distribution of the individual power measurements (e.g., gamma distributed), although with different parameters (e.g., mean, and standard-deviation) [12]. The filtering operation technically distorts the distribution making it no longer gamma distributed. Nevertheless, the difference between the distributions is very minor and the coherent power distribution can be approximated by a gamma distribution.

The phase computed from the conditioned reference locations can be very different from the truth for land pixels near water. Nevertheless, this can actually improve water detection by potentially reducing the coherent power (or gain) for land, but not for water, resulting in an increased land/water contrast.

3.3.2 WaterDetection

This function detects water in the rare interferogram. As part of this process the background land and water powers are also estimated. The water detection algorithm is primarily based on the image power but involves spatial regularization. Recall that Figure 7 in Section 2.3.1 shows the detected water for a simulated river scene.

The power returned to the radar from the surface is related to the normalized radar cross section (NRCS) or σ_0. At Ka band and near nadir, water is typically about an order of magnitude brighter than land, although there are some exceptions (e.g., dark or specular water). Figure 20 shows the σ_0 vs incidence angle curves for inland water, ocean, and land as derived from GPM data as well as a plot of σ_0 vs incidence angle for inland water obtained from AirSWOT. Note that the SWOT system noise is relatively large compared to the expected land return, so water detection is applied on the signal-plus-noise power, rather than on calibrated σ_0 images after noise subtraction (which would be numerically unstable in areas of low σ_0).

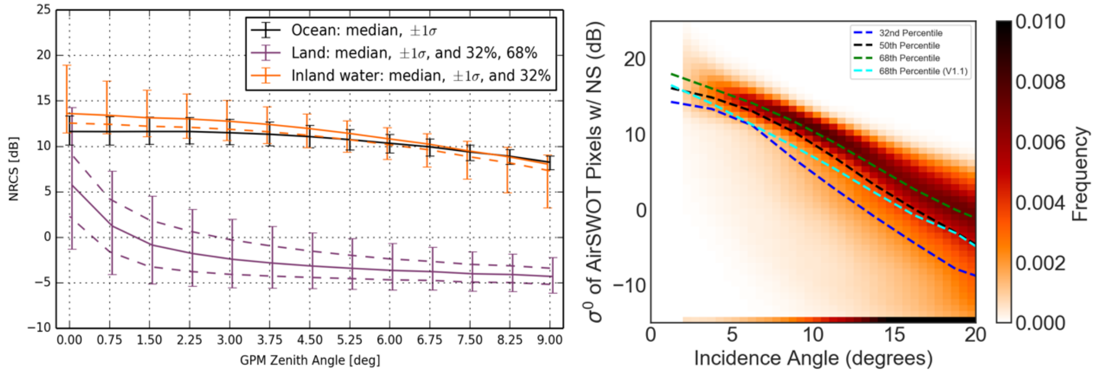


Figure 20. Ka-band normalized radar cross-section (σ_0) versus incidence (zenith) angle from the Global Precipitation Mission (GPM) (left), and σ_0 (with noise subtraction) versus incidence angle over inland water from AirSWOT.

The water in a given scene is detected using the brightness or power of the rare measurements (specifically the rare coherent power). Thus, a simple way to view water detection is to imagine it as a power threshold (which may vary from pixel to pixel). If the measured power is greater than this threshold, the pixel is classified as water; otherwise, it is classified as land or "not water." In reality, the water detection process is more complicated than this because spatial regularization is employed to enforce spatial consistency.

Note that the other measured quantities, the interferometric coherence and phase (or height), are not used at this stage of processing because 1) the SNR is generally so low that the coherence is dominated by the scene brightness anyway (i.e. adds little information), and 2) imposing assumptions about the phase related to water height flatness for the purpose of detecting water in the slant plane can be problematic (e.g., phase noise making it difficult to assess slope locally, artifacts in the reference phase making phase not look flat, or water-water layover introducing phase anomalies).

3.3.2.1 Input Data

Description	Source

Satellite orbit TVP data, radiometric calibration, and noise power measurements (RareUnflatIntf)	MakeRareIntf
Rare coherent power, rare reference locations (CohPower)	MakeCohPow
Prior water occurrence maps (PriorWaterMask)	Static Auxiliary Data

3.3.2.2 Output Data

Description: WaterMap
Water mask representing the binary water detection for every rare-grid pixel
Detection rate images (e.g., estimates of false detection and missed detection rates)
Background land and water powers for every rare-grid pixel

3.3.2.3 Mathematical Statement

The water is detected using the so-called coherent power, which is a coherent combination of the two interferogram channel powers. As described in Section 3.3.1 the coherent power generally results in a higher SNR than the incoherent combination of the two interferogram channel powers. The higher SNR tends to result in a greater effective land/water contrast (since land brightness is often lower than the thermal noise floor), improving the ability to detect water [12].

Knowledge of the distribution of the power measurement is needed to design a robust detection routine and to estimate resulting error rates. Although the filtering of the coherent power technically modifies the distribution, the resulting power for each pixel can be approximated as an independent gamma distributed random variable, which can be expressed as

$$f(\boldsymbol{\mathcal{P}}_c |\, \boldsymbol{\mu_d}) = \prod_x f(\mathcal{P}_c(x) |\, \mu_d(x)) \qquad (3.3.2.3.1)$$

where

$$f(\mathcal{P}_c(x) |\, \mu_d(x)) = \frac{1}{\mu_d(x) \cdot \Gamma(L(x))} \left(\frac{L(x) \cdot \mathcal{P}_c(x)}{\mu_d(x)} \right)^{L(x)-1} exp \left\{ \frac{-L(x) \cdot \mathcal{P}_c(x)}{\mu_d(x)} \right\} \quad \forall\, x \qquad (3.3.2.3.2)$$

and

$$\mu_d(x) = \begin{cases} \mu_1(x) & if\ pixel\ x\ is\ water\ (d(x) = 1) \\ \mu_0(x) & if\ pixel\ x\ is\ not\ water\ (d(x) = 0) \end{cases} \qquad (3.3.2.3.3)$$

is the background land power, $\mu_0(x)$, or water power, $\mu_1(x)$ (depending on whether the class of pixel x is land or water [d(x) = 1, or 0]), $L(x)$ is the effective number of looks, and $\Gamma(\cdot)$ is the gamma function that generalizes the factorial.

The water detection algorithm is based on maximum a posteriori probability (MAP) estimation, where the prior is chosen to enforce spatial regularity employing Markov Random Fields (MRF). The prior can also incorporate information from the prior water probability map projected into the slant-plane (using the coherent power reference locations for the projection). Many details and trades are presented in [13] and [12]. Here, we summarize the approach implemented as the current baseline. The MAP objective function that is optimized with respect to the classification image $\boldsymbol{d}$ can be expressed as

$$J(\boldsymbol{d}) = f(\boldsymbol{\mu_d}|\boldsymbol{\mathcal{P}}_c) = \frac{f(\boldsymbol{\mathcal{P}}_c|\,\boldsymbol{\mu_d})\,f(\,\boldsymbol{\mu_d})}{f(\boldsymbol{\mathcal{P}}_c)} \qquad (3.3.2.3.4)$$

Note the use of the bold notation to represent the vectorized version of all the pixels of the slant-plane images (e.g., $\boldsymbol{d}$ corresponds $d(x)$ for all x). Optimizing J with respect to $\boldsymbol{d}$ gives the estimated water detection image

$$\widehat{\boldsymbol{d}} = \underset{\boldsymbol{d}}{\operatorname{argmax}}\{f(\boldsymbol{\mathcal{P}}_c|\,\boldsymbol{\mu_d})\,f(\,\boldsymbol{\mu_d})\} = \underset{\boldsymbol{d}}{\operatorname{argmin}}\{-\log\{f(\boldsymbol{\mathcal{P}}_c|\,\boldsymbol{\mu_d})\} - \log\{f(\,\boldsymbol{\mu_d})\}\} \qquad (3.3.2.3.5)$$

where $f(\boldsymbol{\mathcal{P}}_c|\,\boldsymbol{\mu_d})$ is the maximum-likelihood term that represents the spatially independent gamma distributed data, and $f(\,\boldsymbol{\mu_d})$ is the prior that enforces spatial regularization. For the prior, a MRF based on an Ising model that favors homogeneous outputs is used

$$-\log\{f(\,\boldsymbol{\mu_d})\} = \beta \sum\nolimits_{x\sim y} \psi\big(d(x), d(y)\big) + k \qquad (3.3.2.3.6)$$

where

$$\psi\big(d(x), d(y)\big) = \begin{cases} 0 & if\ d(x) = d(y) \\ 1 & if\ d(x) \neq d(y) \end{cases} \qquad (3.3.2.3.7)$$

is the Ising model that penalizes neighbor pixels whose classes differ, $x{\sim}y$ indicates that pixels x and y are neighbors in the considered neighborhood (in this case using 4-connectivity), and β is a weighting term. The term k is a scaling that is constant with respect to $\boldsymbol{d}$ and can be neglected when optimizing $J(\boldsymbol{d})$.

Neglecting terms that are not a function of the class, the total solution to the cost function can be written as

$$\widehat{\boldsymbol{d}} = \underset{\boldsymbol{d}}{\operatorname{argmin}} \left\{\sum\nolimits_x L(x)\left(\log\{\mu_d(x)\} + \frac{\mathcal{P}_c(x)}{\mu_d(x)}\right) + \beta \sum\nolimits_{x\sim y} \psi\big(d(x), d(y)\big)\right\} \qquad (3.3.2.3.8)$$

This cost function is solved efficiently by graph-cut methods. For more details see [12].

Note that the cost function depends on prior knowledge of the background land and water power. These are initialized with a global prior on σ_0 for land and water and inputs of the system calibration parameters (noise power and the radiometric calibration X-factor). However, there can be significant spatial variations on the scene brightness. Thus, the method iteratively estimates the background powers from the data and reclassifies the image in order to improve the robustness of the algorithm to uncertainties in σ_0. Figure 21 illustrates the iterative nature of the water detection approach.

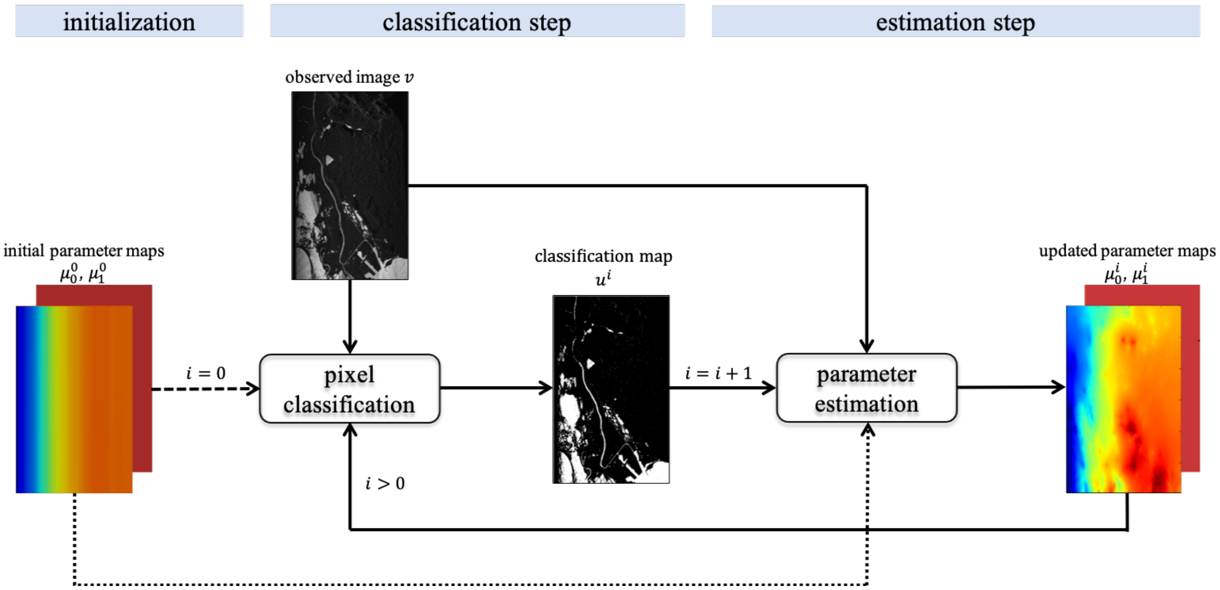


Figure 21. Iterative water detection and background land and water power parameter estimation [12] [13].

Several approaches to estimate the background powers have been explored including simple estimation with smoothing, MRF approaches [12] [13], as well as using prior water masks. All three options are implemented, but the current baseline is to apply a simple estimation approach on each iteration i, which, for the background water power (μ_1), can be expressed as

$$\mu_1^{i+1}(x) = \tilde{\mu}_1^i(x) * K_1(x) \tag{3.3.2.3.9}$$

where

$$\tilde{\mu}_1^i(x) = \begin{cases} \mu_1^i(x) & if\ \hat{d}^i(x) = 1 \\ \mu_1^f(x) & if\ \hat{d}^i(x) = 0 \end{cases} \tag{3.3.2.3.10}$$

and

$$\mu_1^f(x) = \mu_1^i(x) \cdot w_1(x) * K_1(x) \tag{3.3.2.3.11}$$

where $K_1(x)$ is a smoothing kernel, the superscripts i and $i + 1$ refer to the current and next iteration, $\hat{d}^i(x)$ is the detected water/land classification, for the current iteration, and $w_1(x)$ is a weighing function that favors measured data of the same class. The land background power (μ_0) can be expressed similarly, but replacing the subscripts and class values from 1 to 0. Typically, only one or two iterations are needed to reap the majority of the benefits from the iterative approach to background water/land power estimation.

3.3.2.4 Accuracy

Water detection errors vary throughout the swath and occur more frequently in regions of low SNR. An estimate of the detection error rates per pixel is also computed and provided in the L2_HR_PIXC product itself. An estimate of the detection rates for the regularized prior is implemented as an option in the processor; however, a worst-case estimate which ignores the spatial regularization is computed and reported in the product. The false-detection rate and missed-detection rate are computed as follows

$$P_{FD}(x) = 1 - \gamma\left(L(x), \frac{L(x)\cdot\mathcal{P}_t(x)}{\mu_0(x)}\right) \tag{3.3.2.3.12}$$

$$P_{MD}(x) = \gamma\left(L(x), \frac{L(x)\cdot\mathcal{P}_t(x)}{\mu_1(x)}\right) \tag{3.3.2.3.13}$$

where $\gamma(\cdot)$ is the lower incomplete gamma function, and $\mathcal{P}_t(x)$ is the power threshold image corresponding to a non-regularized MAP estimate (i.e. which is equivalent to setting the MRF regularization parameter to zero), which can be expressed as

$$\mathcal{P}_t(x) = \frac{\log\{\mu_1(x)\} - \log\{\mu_0(x)\}}{\left(\frac{1}{\mu_0(x)} - \frac{1}{\mu_1(x)}\right)} \tag{3.3.2.3.14}$$

Note that although water is typically much brighter than land at SWOT incidence angles, it can sometimes be very dark due to specular scattering when the water is very calm. Since this water detection algorithm detects bright returns as water, it does not generally do well in the presence of dark water. This is the rationale for the dark-water flagging algorithm described below.

On the other hand, there are some non-water features that are expected to be bright and thus will be incorrectly detected as water (e.g., urban developed areas, as well as land-land layover in regions with significant terrain). The bright land flagging algorithm described below produces an estimate of non-water regions that are likely to be detected as water.

To assess the water detection performance, SLC simulations have been run for a representative dataset. The detection rates with respect to the slant-plane-projected truth water mask are computed as well as the estimated detection rates. Figure 22 shows the estimated (as per Equations (3.3.2.3.12) and (3.3.2.3.13)) and measured false-detection and missed-detection rates as a function of cross-track distance from nadir. Although detection rates are all relatively small, the measured false-detection rate tends to be a little larger than the estimated, and the missed detection rate has a somewhat unexpected sharp increase in the near range. Both of these differences are primarily due to the issues related to azimuth point-target response (PTR) smearing, which can cause land edges near water to be falsely detected as water, as well as small or narrow water bodies (especially in the near range) to entirely miss-detected. Figure 23 shows the false-detection rates and missed detection rates excluding the edge pixels. When excluding true land/water edges, the estimated and measured detection rates become more consistent (and generally lower). Figure 24 shows the detection error rates excluding the detected water edges. Note that the effect of the over-detection of water on the land-water edges seems to be mitigated, but the missed detection issue in the near range is not completely compensated. This is because narrow features can often be missed-detected, especially in the near range where the pixel size is larger, and so detected edges do not capture these missed detections. The water fraction approach described in the next subsection is used to mitigate the azimuth PTR related over-detection effect.

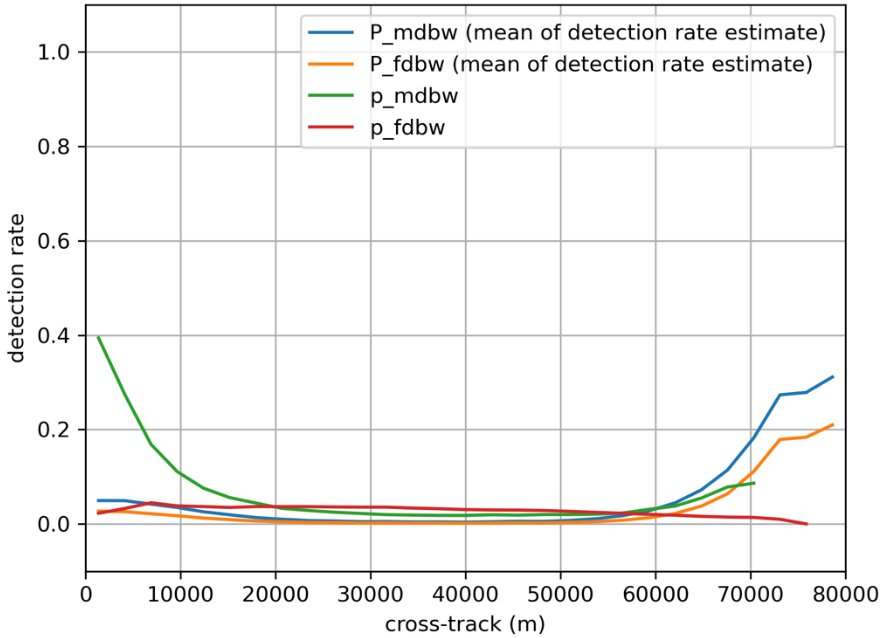


Figure 22. Estimated (P) and measured (p) false-detection (fdbw) and missed-detection (mdbw) rates of bright water for simulation with representative dataset.

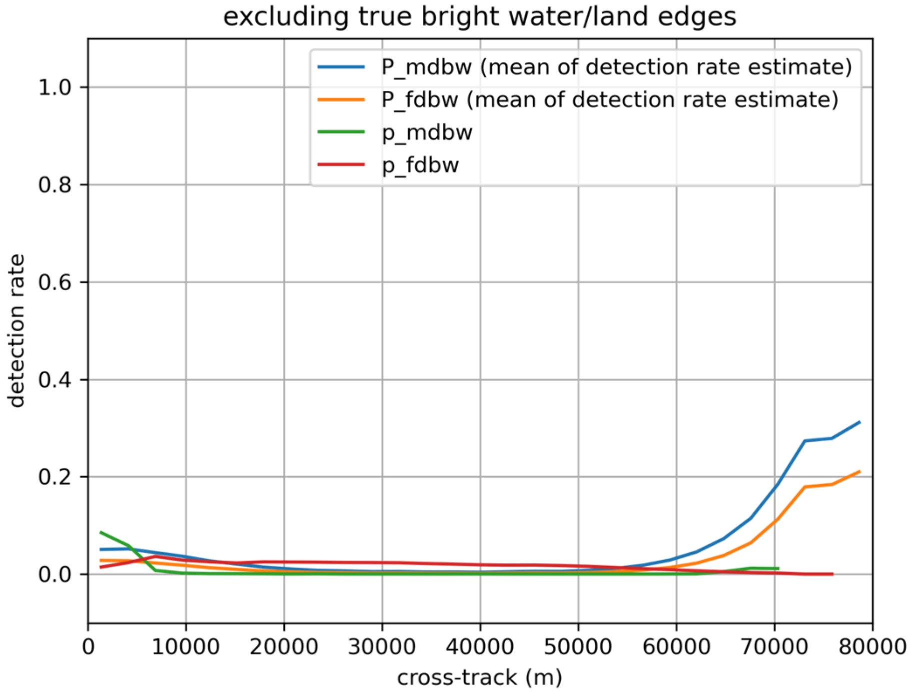


Figure 23. Estimated (P) and measured (p) false-detection (fdbw) and missed-detection (mdbw) rates of bright water excluding true land/water edges for simulation with representative dataset (truth land/water edges is only known in simulated cases).

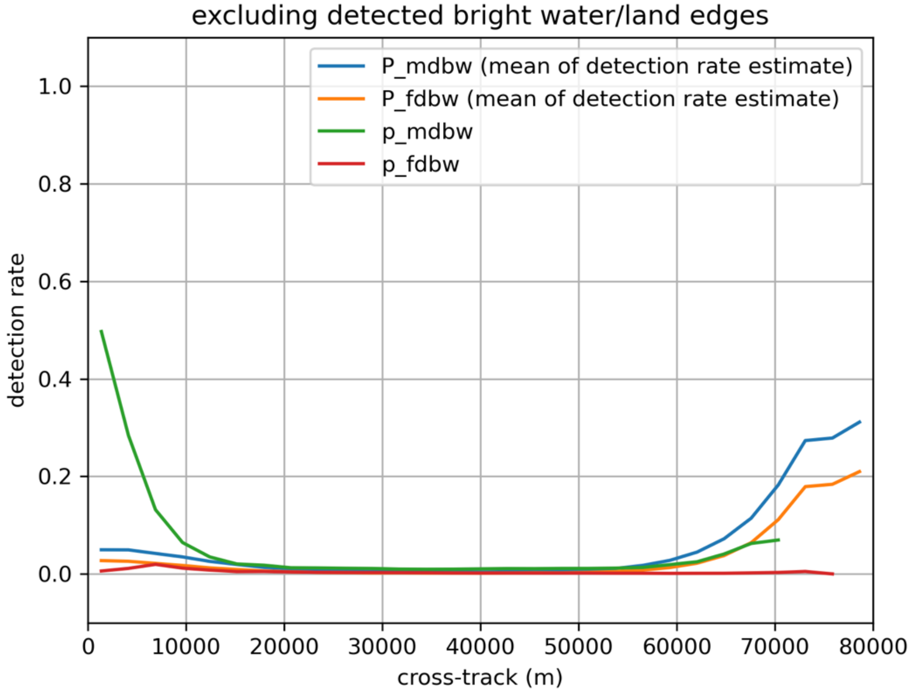


Figure 24. Estimated (P) and measured (p) false-detection (fdbw) and missed-detection (mdbw) rates of bright water excluding detected land/water edges for simulation with representative dataset.

3.3.3 WaterFracEst

This function estimates the water fraction for each pixel of the rare slant-plane grid. Over the finite resolution of a slant-range pixel, some surfaces are a mixture of land and water (not just one or the other, as assumed in the binary detection approach). In these cases, a binary water detection method can produce biased results. This is especially problematic near land/water boundaries, since the water can have a different azimuth resolution than land due to decorrelation of the moving water over the SAR focusing integration time (i.e., water has a short coherence time). Using an estimate of the water fraction around the water body boundaries, for example, can improve the accuracy of water body area estimates. Recall that Figure 7 in Section 2.3.1 shows the water fraction estimate for a simulated river scene.

Figure 25 shows coherence time estimates from AirSWOT using two different methods (e.g., using along track interferometry (ATI), and measuring the widths of the transitions in the power images) as well as the azimuth smearing width for AirSWOT and what is expected for the SWOT observation geometry. For inland water bodies the coherence time ranges from about 10 to 30 ms, which corresponds to limiting the azimuth resolution of water to about 7 to 25 m for SWOT. This suggests that the backscattered energy from water generally smears out over a few pixels in azimuth, but it can vary in extent spatially and from pass-to-pass. Near land/water boundaries this means that pixels that are actually land get extra water power, which makes them more likely to be detected as water (over-detection bias). The water fraction estimate provides robustness to this effect.

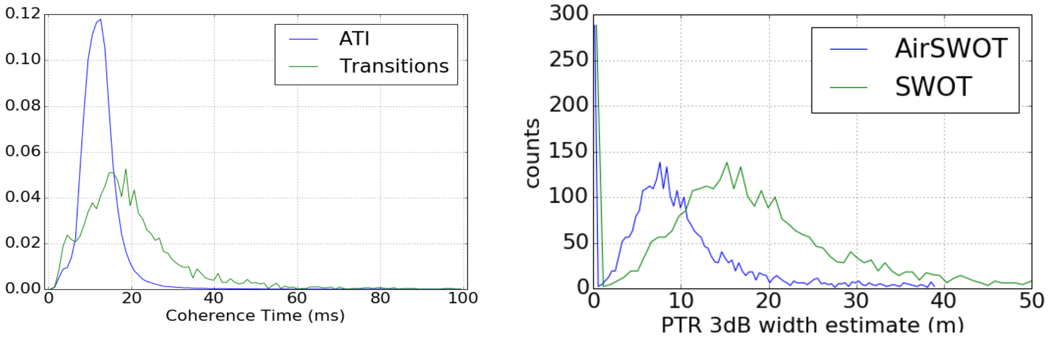


Figure 25. Coherence time estimates from AirSWOT using two estimation methods (left) and azimuth smearing width for AirSWOT and SWOT (right).

3.3.3.1 *Input Data*

Description	Source
Rare coherent power slant-plane image (CohPower)	MakeCohPow
Background land and water power estimates (WaterMap)	WaterDetection
Rare number of looks (RareUnflatIntf)	MakeRareIntf

3.3.3.2 *Output Data*

Description: WaterFrac
Water fraction estimate for every rare-grid pixel
Water fraction uncertainty estimate for every rare-grid pixel

3.3.3.3 Mathematical Statement

The estimate of the water fraction is computed using the estimates of the background land (μ_0) and water (μ_1) powers output from the WaterDetection function. The water fraction α can be expressed as

$$\alpha(x) = \frac{\mathcal{P}_c(x) - \mu_0(x)}{\mu_1(x) - \mu_0(x)} \tag{3.3.3.3.1}$$

which is a noisy but unbiased estimate of the water fraction (under the condition that $\mu_1(x)$ and $\mu_0(x)$ are known precisely). Note that although the true water fraction can only take on values between 0 and 1 (excluding the case of water-water layover), the noisy estimate $\alpha(x)$ can be either negative or greater than one. The water fraction is generally very noisy on a pixel-wise basis, but after aggregating over entire water features it is able to provide an unbiased, low-dispersion estimate of the water body area (also with much lower likelihood of nonsensical values less than 0 or greater than 1). This water fraction estimate is a powerful tool to account for the azimuth smearing of water pixels that is caused by the temporal decorrelation of water over the SAR focusing integration time.

3.3.3.4 Accuracy

An estimate of uncertainty of the water fraction estimate is computed and reported in the L2_HR_PIXC product. The water fraction estimate defined above is a function of the measured power and the background land and water powers. If there is no uncertainty in the background power estimates, the distribution of the water fraction is just a shifted and scaled version of the measured power distribution. Assuming a gamma distribution for the power, the variance of the distribution of $\alpha(x)$can be expressed as

$$Var\{\alpha(x)\} = \frac{\bar{\mathcal{P}}_c{}^2(x)}{L(x)[\mu_1(x) - \mu_0(x)]^2} = \frac{\left[\bar{\alpha}(x)\,\mu_1(x) + \left(1 - \bar{\alpha}(x)\right)\mu_0(x)\right]^2}{L(x)\,[\mu_1(x) - \mu_0(x)]^2} \tag{3.3.3.4.1}$$

where the overbar notation represents the expected value (i.e., the mean) of the corresponding quantities. This expression can be used to estimate the uncertainty of the water fraction estimate. However, this approach to estimate the water fraction uncertainty can be inaccurate when we only have access to noisy estimates of the background powers. We can handle errors in the background powers and produce a better estimate of the variance of the water fraction estimate by the following approach. Note that the measured power is gamma-distributed with rate and shape parameters that are functions of the water fraction, the background powers, and the effective number of looks. We can apply an improper non-informative conjugate prior on the rate parameter resulting in an inverse-gamma distributed posterior distribution. The variance of the posterior distribution is known. This results in the variance of the water fraction estimate becoming

$$Var\{\alpha(x)\} = \frac{L^2(x)\bar{\mathcal{P}}_c{}^2(x)}{(L(x)-1)^2\,(L(x)-2)\,[\bar{\mu}_1(x) - \bar{\mu}_0(x)]^2} \tag{3.3.3.4.2}$$

$$= \frac{L(x)\left[\bar{\alpha}(x)\,\bar{\mu}_1(x) + \left(1 - \bar{\alpha}(x)\right)\bar{\mu}_0(x)\right]^2}{(L(x)-1)^2(L(x)-2)\,[\bar{\mu}_1(x) - \bar{\mu}_0(x)]^2} \tag{3.3.3.4.3}$$

where we use the actual estimates of water fraction and the background land and water power for $\bar{\alpha}(x)$, $\bar{\mu}_0(x)$ and $\bar{\mu}_1(x)$, respectively. The standard deviation (the square-root of $Var\{\alpha(x)\}$) is reported in the L2_HR_PIXC product.

The water fraction estimation approach is assessed using simulation. Note that when estimating the water area over entire water features in the river and lake algorithms the water fraction is used to scale the pixel area for pixels near shorelines (detected as edges as described in Section 3.3.8), while the total pixel area is used for the water "interior" water (pixels not near the shore). Therefore, the assessment here focuses on the water fraction estimates of the estimated shoreline pixels. Figure 26 shows the water fraction error distribution with the 50 percentile and 1-sigma (68 percentile of the absolute value) as a function of cross-track distance from nadir. Fr this plot as well as the other similar plots later in this document the percentiles are computed directly from the 2D histograms, which are accumulated over all pixels and simulated tiles in the representative dataset by counting all pixels that fall within the y-axis bins for each cross-track bin. We only use the water fraction estimate on the detected edge pixels in the down-stream algorithms, so the corresponding pixel-wise area percent errors for the edge pixels are expected to be on the order of 2% bias with less than 30% 1-sigma variability (the cross-track averages from 10 to 60km are in the corresponding legend entries). The variability of these results are expected to aggregate down when aggregating over entire water features (rivers, lakes, and raster bins) suggesting that the result will be a relatively low bias with an acceptable variability, especially for wider features with relatively few edge pixels. Although narrower water features (e.g., rivers) that are oriented in the cross-track direction may exhibit a bias of up to 2% making the resulting water area and river width estimates less reliable, this is still expected to be well within the river width requirements for 100m rivers.

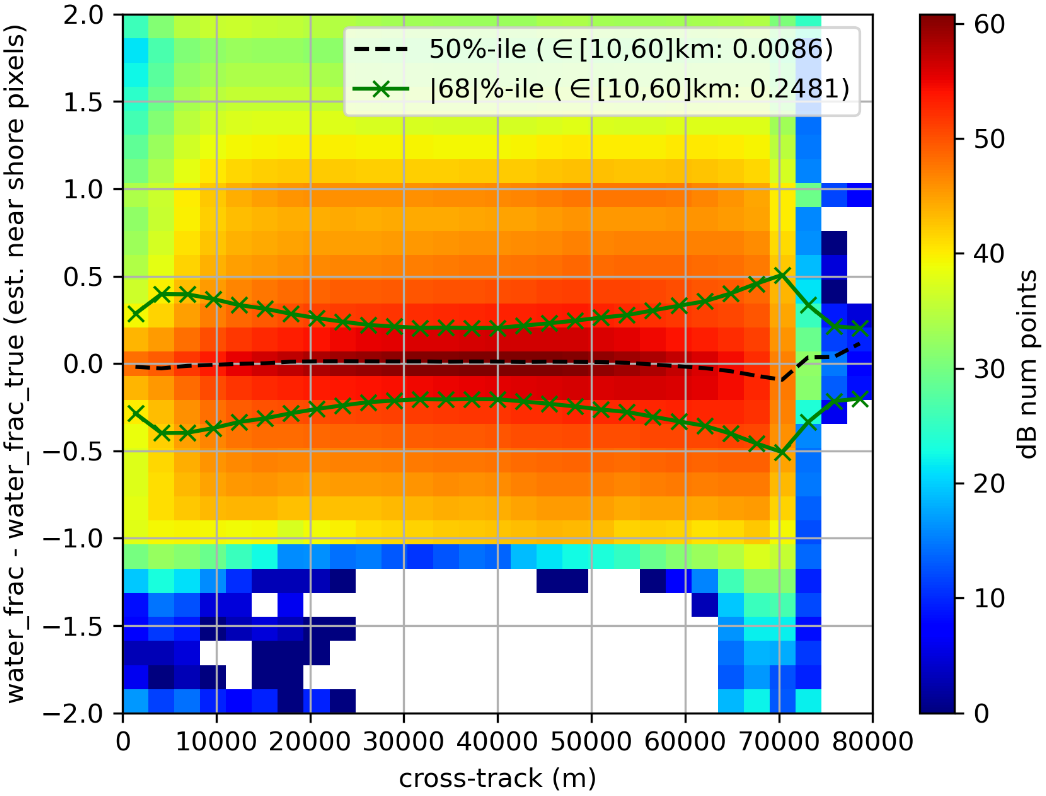


Figure 26. Simulated water fraction error for detected edge pixels (water and land near the shore).

3.3.4 DarkWaterFlagging

This function flags dark water using a prior water occurrence map (e.g., the Auxiliary data occurrence maps that give probability of water occurring for any given place on the ground [cite the aux data description doc?]) and the result of water detection. Water is dark when little or no signal returns to the radar, so the flagging algorithm is based on prior (not observed) water-occurrence information. Detected water is used to help refine the slant-plane projection of the prior information, as well as to help select an appropriate occurrence probability threshold. Recall that Figure 8 in Section 2.3.1 shows the prior water probability and the binary prior water probability mask slant-plane images for a simulated river scene.

When water is calm, the scattering becomes specular. In other words, most of the radar signal tends to bounce off the surface in the opposite direction that it hits the surface. This makes it so that the energy returning to the radar as a function of incidence angle drops off very sharply as the incidence angle moves away from nadir. When there is more water surface roughness, the backscattered energy falls off much slower as a function of incidence angle. Figure 27 shows an example of AirSWOT data collected at two wind speed conditions (low ~2-3 m/s and moderate ~5-7 m/s), as well as a geophysical model function (GMF) that represents σ_0 vs incidence angle curves as a function of wind speed. Note that at very low wind speeds, σ_0 is low even at SWOT-like incidence angles (i.e., the water looks dark in the radar images), and that the transition from bright σ_0 to dark σ_0 over water as a function of wind speed can be very sharp, occurring over a very narrow wind speed range. Thus, typical SWOT-like images are expected to be bright with dark patches, with very few pixels with "in-between" brightness.

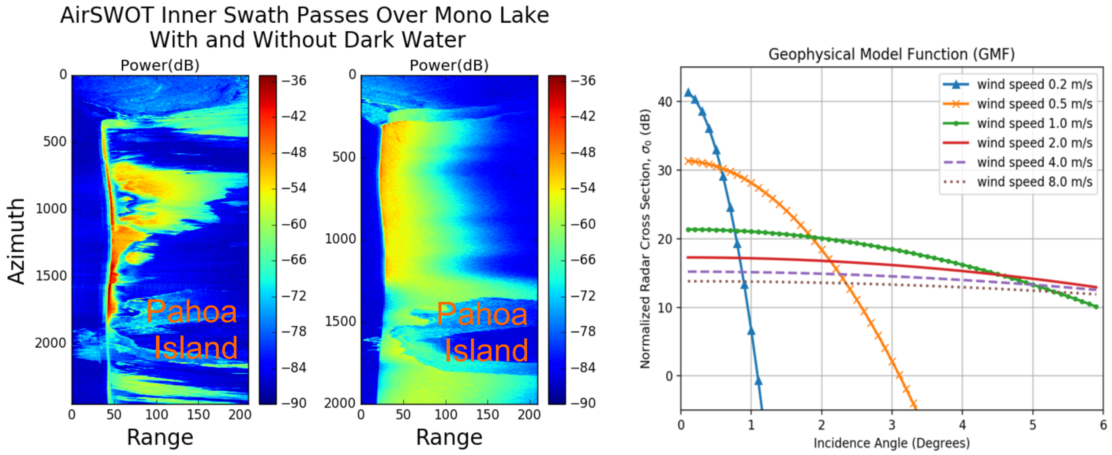


Figure 27. AirSWOT power images over Mono Lake for two different surface wind speed conditions (~3-4 m/s for left, ~5-7 m/s for middle), and Ka-band geophysical model function showing σ_0 versus incidence angle for several wind speeds (right).

Specular reflections are not the only reason that water returns can be dark. Atmospheric attenuation due to rain can also result in signal loss. Commonly, significant rain rates occur at relatively short spatial scales and the effect on the data is expected to appear similar to the specular water phenomenon – causing patches of the swath to go dark. Thus, the same method for dark-water flagging also serves to mitigate the effects of rain.

In the presence of dark water, there is little or no signal to detect in the radar images.

Furthermore, there is little information about the height in the phase. The purpose of this dark-water flagging algorithm, therefore, is to use prior information to try to recover the extent of inundation so that applications that use the water body area can still be applied with minimal impact on bias due to dark water occurrence.

The water occurrence probability, such as that produced by Pekel et al [14], can generally be used to obtain a binary water mask that corresponds to a given water height. To be more precise, contours of constant water probability approximately correspond to a particular water inundation mask for a given water height or river stage. Therefore, thresholding the probability map at a given threshold that fits the detected water mask best gives a mechanism for filling in missing parts of a given water feature caused by dark water.

3.3.4.1 Input Data

Description	Source
Detected water (WaterMap)	WaterDetection
Prior water occurrence map (PriorWaterMask)	Static Auxiliary Data
Satellite orbit TVP data (RareUnflatIntf)	MakeRareIntf
High resolution reference DEM (RefDEM)	Static Auxiliary Data

3.3.4.2 Output Data

Description: DarkWaterMap
Dark water flag binary slant-plane image
Prior water probability map image projected into slant-plane
Estimated probability threshold slant-plane image
Estimated range-shift slant-plane image
Range coregistration-updated reference DEM (CoregRefDem)
Range coregistration-updated reference locations

3.3.4.3 Mathematical Statement

The dark-water flagging algorithm projects a prior water occurrence probability map into the slant plane using reference locations that are recomputed here. The same conditioning method described in Section 3.3.1.3 is applied here, but both detected and prior water are used in this round of height conditioning. If there is a height bias in the reference DEM, there is the potential to have a mapping error that looks like a shift in the range direction. An attempt is made to correct this shift as a function of along-track, using a correlation of the projected water probability with the detected water mask. Next, the range coregistration fit is used to produce an estimate of the height error for water areas in the reference DEM, and an algorithm to adjust the water heights in the DEM is applied. This updated DEM is then conditioned in a similar manner as the other conditioning steps, but in the ground projection, then new reference locations are computed and the prior reprojected. Finally, a water probability threshold is selected to get a binary mask. The threshold is chosen to be the most consistent with the detected water mask.

The range-shift s is estimated for each slant-plane pixel (x) using an exhaustive search over a limited range-shift window where the quantity to be maximized is the correlation coefficient. This can be expressed as

$$s(x) = \underset{s}{\operatorname{argmax}}\{\rho(x, s)\} \tag{3.3.4.3.1}$$

where

$$\rho(x,s) = \frac{\sum_k M_{prior}(k,s)\, M_{meas}(k) W(k-x)}{\sum_k |M_{prior}(k,s)\,|^2 W(k-x) \sum_j |M_{meas}(j)\,|^2 W(j-x)} \qquad (3.3.4.3.2)$$

where $M_{prior}(x,s)$ is the binary prior mask shifted in the slant-range dimension by s pixels, $M_{meas}(x)$ is the measured or detected water mask, $W(x)$ is a uniform (or boxcar) smoothing window (default size 50 x 50 pixels). This window is chosen to help regularize the shift fit to be spatially smooth (note that the numerator is a 2-D convolution representing a spatial filtering operation). The assumption that the shift is spatially smooth is appropriate since the height biases in the reference DEM are expected to vary slowly in space. Note that the binary prior mask is obtained by thresholding the prior water probability map $\Pi(x)$ at zero probability (i.e. maximum water extent) after projecting it into the slant plane, i.e.,

$$M_{prior}(x) = \begin{cases} 1 \; if \; \Pi(x) > 0 \\ 0 \; if \; \Pi(x) = 0 \end{cases} \qquad (3.3.4.3.2)$$

The probability threshold is selected as follows. The slant-plane is broken up into overlapping windows (by default 201 by 201 pixels with 50% overlap). The best probability threshold is then computed for each window separately. The estimated thresholds over the windows are then combined using a Hamming window to average the overlapping probability threshold estimates. The best threshold t_n for each window n is selected by maximizing

$$\rho_n(t) = \frac{\sum_x M_{n,t}(x)\, M_n(x)}{\sqrt{\sum_x |M_{n,t}(x)\,|^2 \sum_x |M_n(x)\,|^2}} \qquad (3.3.4.3.3)$$

where $M_{n,t}(x)$ is a binary mask made by thresholding the prior water probability at t such that

$$M_{n,t}(x) = \begin{cases} 1 \; if \; \Pi(x) \geq t \\ 0 \; if \; \Pi(x) < t \end{cases} \qquad (3.3.4.3.4)$$

and $M_n(x)$ is the measured or detected mask.

The best threshold slant-plane image $t(x)$ is obtained from the best thresholds over the windows $t_n$by averaging with the Hamming window function $H_n(x)$ (the same size as the overlapping windows, 201 by default). This operation can be expressed as

$$t(x) = \frac{\sum_n t_n\, H_n(x)}{\sum_n H_n(x)} \qquad (3.3.4.3.5)$$

The final slant-plane binary prior water mask is obtained by thresholding the slant-plane projected probability map (with the corrected coregistration range shift) with the best threshold as per Equation (3.3.4.3.4). This binary prior mask is used in conjunction with the detected water slant-plane image (later in Section 3.3.8) to set as dark water the classification of the pixels that are not detected as water, but the prior mask says it should be water.

3.3.4.4 Accuracy

The dark-water flag is very dependent on the prior data (both the DEM and the water probability map). The conditioning of the reference locations and the estimation of the cross-track shift provide some robustness to the accuracy of the reference DEM, but if the prior water mask is not representative of the actual water surface extent, dark-water flagging is also not expected to be very accurate. The water mask being used is based on the Pekel water occurrence [14], which in general is representative of water occurrence probability, but in some areas these may be biased toward the dry season, or may not accurately describe water under vegetation. Subsequent reprocessing may involve using an occurrence map that has been updated using high spatial resolution active radar data, including SWOT.

Note that this dark-water flagging approach is not well suited to handle cases where large percentages of a water feature are dark, since it uses the detected (i.e., bright) water to aid in selecting a probability threshold and projecting into the slant plane. Nevertheless, for these cases where most of a water body is dark, there is effectively no measured signal or information about that water feature in the data. Generally, it is expected that the presence and structure of dark water will fluctuate at relatively short time scales compared to revisit times. Thus, multi-temporal, or multi-sensor, approaches could be applied by science users outside of the SWOT operational processing flow to detect and possibly interpolate over these cases.

Note that the dark water flagging algorithm is also sensitive to where in the water-body the dark patches occur. For example, the algorithm is expected to be less reliable of the entire perimeter of a water body is dark as opposed to the same amount of dark water occurring mostly interior to the water body (or even randomly distributed). This is because the estimation of the probability threshold that produces the variable-extent binary water mask from the probability map is particularly sensitive to the edges (or shorelines) of the water-bodies.

The dark water flagging performance is assessed using the representative dataset simulations. Figure 28 plots the detection error rates for various detected classifications given the various truth classifications as a function of cross-track distance from nadir. The plot shows the missed detection rate of dark water (i.e., the probability of detecting land given that the truth is dark water), as well as the false detection of dark water (i.e., the probability of detection dark water when the true class is land). This plot shows that when we have true dark water, land is incorrectly detected about 50% of the time, which shows some skill, but is not particularly great. The plot also shows the missed detection rate for water bodies wider than 100m. The error decreases suggesting that many of the missed detections are for small or narrow water bodies that are more likely to be in dark, but are not likely to be well represented in the prior mask. Note that although false detection rate of dark water is very low, this does not mean that there a few false detections. In fact, there are about as many false detections and missed detections, but because there are many more land pixels than dark water pixels, the false detection rate is very low. This can be seen in the Missouri River example in Figure 11 which shows a lake in the upper right detected as dark water, where there is no lake in the truth, as well as missed dark water in the the very narrow river channels. In this scene there are a few dark water patches of dark water that are correctly detected as dark but are a bit smaller (due to the overdetection case by the azimuth PTR smearing effect). This is not a problem since these bright/dark water edges are still detected as water. Note that although there are some algorithmic contributions to the dark water flagging error (e.g., slant-plane-projection or coregistration issues), the major contributor to the dark water flagging error is the inconsistency between the truth and prior water mask (and to a lesser extent the truth and reference DEM). The simulations used in this analysis are expected to be representative in terms of the difference between the truth and prior water masks and DEMs, suggesting that there is a fundamental, non-algorithmic, limit to the accuracy of such a dark water flagging approach. Nevertheless, improved accuracy may be possible to be obtained in future reprocessing campaigns after updating the water mask with SWOT data.

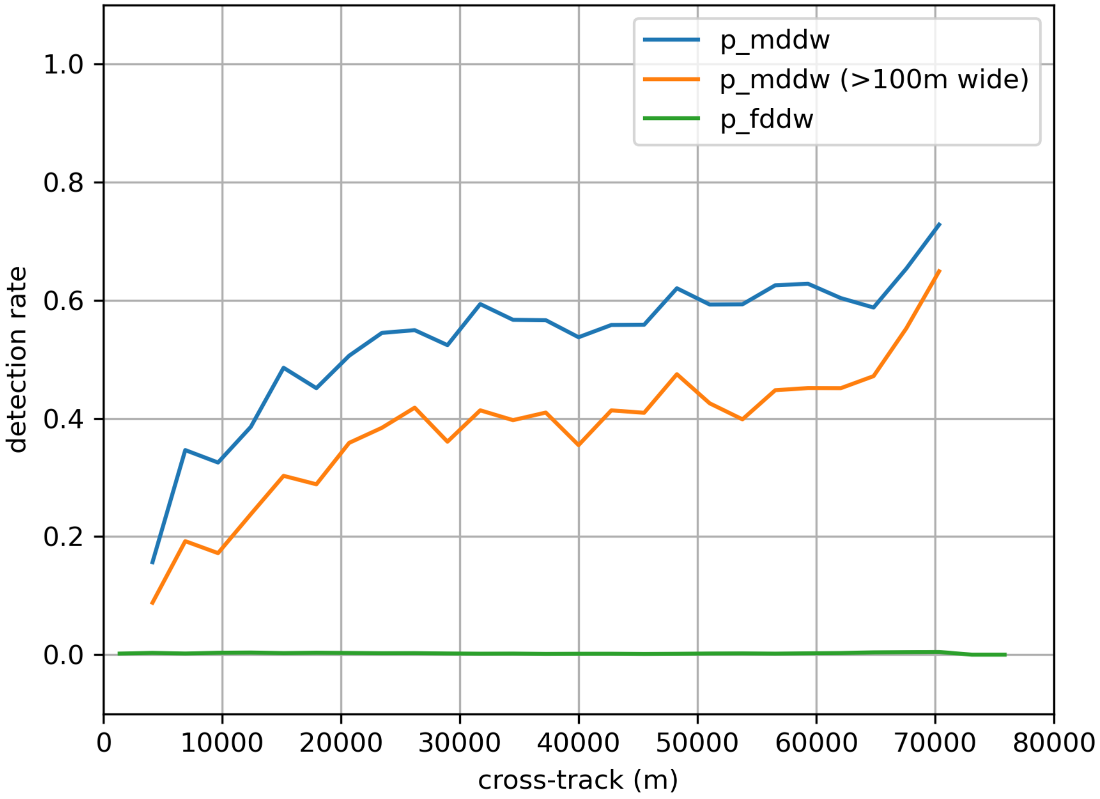


Figure 28. Detection rates for land, bright, and dark water given various truth classes. The probability of missed detection of dark water (p_mddw) is plotted for all data and all water features greter than 100m wide, while the probability of false detection of dark water is plotted only for all data.

3.3.5 LayoverFlagging

This function computes an estimate of the height error related to layover. This quantity can be used for augmenting uncertainty estimates and interpreting artifacts in the height data. This layover flag is not a binary indicator of where layover occurs, rather it is a continuous valued function predicting height error due to layover. Recall that Figure 9 in Section 2.3.1 shows the layover impact flag for a simulated river scene.

3.3.5.1 Input Data

Description	Source
Coregistration updated reference DEM (CoregRefDEM)	DarkWaterFlagging
Detected water map and background land and water power estimates (WaterMap)	WaterDetection
Satellite orbit TVP data (RareUnflatIntf)	MakeRareIntf
Prior water occurrence map (PriorWaterMask)	Static Auxiliary Data

3.3.5.2 Output Data

Description: LayoverMap
Estimated layover height error image

3.3.5.3 Mathematical Statement

Note that several approaches have been explored for flagging layover and excluding layover pixels in downstream processing when aggregating to water-body level features (e.g., lakes and river nodes). However, it has been shown that excluding these pixels, even if perfectly flagged, actually is often worse than just keeping the layover-corrupted pixels because having more pixels helps in reducing the height noise, and the layover impact is not generally that extreme. For this reason, the baseline layover flagging method implements a layover impact flag that is to be used only for providing rough information on the quality of the height data. This information can be folded into the height uncertainty estimates of the more aggregated products.

The baseline layover flagging algorithm computes an expected layover-induced height error (e.g., a continuous-valued impact flag) using a reference DEM and reference water mask with estimates of the background σ_0 for land and water. Although a simple flag for indicating land-land layover, land-water layover, and water-water layover based on a reference DEM and water mask has been explored and implemented, it is not currently exercised in the baseline processing.

The expected layover height error is computed for each detected water pixel by integrating a synthetic interferogram for all DEM pixels in the same range bin, weighted by the expected power for the given reference class in ground coordinates. This can be expressed as

$$\Delta h_l(x) = \left(\phi_s(x) - \phi_{ref}(x)\right) \cdot \frac{\partial h}{\partial \phi_{ref}}(x) \tag{3.3.5.3.1}$$

where $\phi_{ref}(x)$ is the phase of the reference location (which is assumed to be the truth location for this purpose), $\frac{\partial h}{\partial \phi_{ref}}(x)$ is the sensitivity of the height to the phase at the reference location, $\phi_s(x)$ is the phase of the simulated interferogram $I_s(x)$

$$\phi_s(x) = \angle I_s(x) \tag{3.3.5.3.2}$$

and $I_s(x)$ is computed from the power $P_s(g)$ in ground coordinates g and geometry between the reference DEM and the sensor assuming broadside imaging

$$I_s(x) = \sum_{g \to x} P_s(g) \cdot e^{-j\phi(g)} \, dg = P_s(x) e^{-j\phi_s(x)} \qquad (3.3.5.3.3)$$

where the $g \to x$ represents all the points on the ground (g) that map into the same slant plane pixel (x). Note that the power in the ground coordinates can be computed in various ways. Several methods have been investigated including using a known expected contrast between land and water, and using estimated background powers from the water detection step projected in ground coordinates. (Note that the power of land that lays over into water cannot be estimated directly because it is necessarily in the same slant-plane pixel as water if it is in layover, so it must be inferred from nearby land.) Also, investigations have been conducted using a prior water mask, the detected mask, and a combination of the two in order to determine classes in ground coordinates. Moreover, studies have been conducted involving smoothing the water heights to make water flat. The current baseline method uses a constant contrast of 10 dB between water and land brightness, applying the prior mask in ground coordinates, with no height smoothing of water.

3.3.5.4 Accuracy

The layover impact quantity provided in the product is for informational purposes only and there are no requirements on its accuracy. Layover quantities are very sensitive to the reference DEM, as well as the relative σ_0 values. Layover estimates are not sufficiently accurate to apply directly on a pixel-wise basis, although they are reported at that posting. However, layover quantities, such as expected layover height error, may become more robust after they are aggregated over larger extents; this is a potential subject of future work.

3.3.6 BrightLandFlagging

This function computes a flag for bright land. This quantity is heavily dependent on the prior information and can be used for qualitative assessments of false detection of water.

3.3.6.1 Input Data

Description	Source
High resolution reference DEM (RefDEM)	Static Auxiliary Data
Detected water map and background land and water power estimates (WaterMap)	WaterDetection
Satellite orbit TVP data (RareUnflatIntf)	MakeRareIntf
Prior water occurrence map (PriorWaterMask)	Static Auxiliary Data
Prior bright land map (BrightLandMask)	Static Auxiliary Data

3.3.6.2 Output Data

Description: LayoverMap
Estimated bright land flag

3.3.6.3 Mathematical Statement

The bright land flag applies similar principles to the dark water flag. That is, the prior bright land mask is projected into the slant-plane using the conditioned reference locations made from the coregistration-updated reference DEM. The prior bright land mask itself, however, has a higher resolution than the reference DEM or the Pekel occurrence mask, since it is based on various data sources that incorporate information such as urban development. The bright land mask is used only for providing quality context to the water detection algorithm (e.g., by marking the classification quality flag as suspect for areas known to be dense urban areas that are expected to be bright regardless of the presence of water).

3.3.6.4 Accuracy

The accuracy of the bright-land flag is very dependent on the prior data (both the DEM and the bright land map). The projection of the prior into the slant-plane as well as the consistency of the mask itself with truth has the same fundamental issues as the dark water flagging algorithm. Since this quantity is provided principally for added science user utility and is not used in subsequent processing algorithms, the accuracy is not as critical as other algorithms.

3.3.7 LowCohFlagging

This function computes a flag for low coherence water. This quantity is helpful for identifying bright but low-coherence water, which is expected to have different properties than the typical open water class (e.g., because of layover corruption or volume scattering from vegetation etc.). Recall that Figure 10 in Section 2.3.1 shows the low-coherence flag for a simulated river scene.

3.3.7.1 Input Data

Description	Source
Detected water map and background land and water power estimates (WaterMap)	WaterDetection
Rare flat interferogram, including the two channel powers and complex interferogram (RareUnflatIntf)	MakeRareIntf

3.3.7.2 Output Data

Description: LowCohMap
Estimated low coherence water map

3.3.7.3 Mathematical Statement

The interferometric coherence normalized by the SNR-induced coherence is computed, then a special filter is applied that accounts for spatial structure in the coherence and phase. Next, a threshold is applied to flag low coherence water regions.

The SNR-normalized coherence is computed as

$$\gamma'(x) = \frac{\gamma(x)}{\gamma_{SNR}(x)} \tag{3.3.7.3.1}$$

where

$$\gamma_{SNR}(x) = \frac{1}{\left(1+SNR_{ref}^{-1}\right)\left(1+SNR_{sec}^{-1}\right)} \tag{3.3.7.3.2}$$

Note that low SNR values are clipped to a non-zero small value (0.01) for both the reference and secondary SNR estimates before computing $\gamma_{SNR}(x)$. Furthermore, the $\gamma_{SNR}(x)$ term is also clipped (to 0.1) for moderately low values of reference or secondary channel SNR (0.5) before creating the SNR-normalized coherence estimate $\gamma'(x)$. This reduces anomalies due to normalizing with noisy estimates of SNR, while preserving some information relative to simple clipping of $\gamma'(x)$ (e.g., if only one of the channels is causing the anomaly only the SNR of that channel gets clipped). This clipping approach also results in more well-behaved statistical distributions of $\gamma'(x)$ for each pixel that reduce further anomalies when spatial filtering. The SNR-normalized coherence is spatially filtered as follows. First, a complex image is created with the SNR-normalized coherence as the magnitude, and the flattened interferometric phase as the phase. Then a complex vector median filter is applied to the slant-plane image, which is equivalent to independently median filtering the real and imaginary parts. The filter size is chosen to be a small number of slant-plane pixels (3, which is consistent with the medium smoothing described below). This minimal amount of filtering is helpful for getting a more robust estimate of the coherence, while preserving finer spatial structure as well as edges and discontinuities in the interferometric phase or coherence. The final smoothed coherence estimate is taken as the magnitude of the smoothed complex image.

The smoothed coherence is then thresholded and any pixels with a value less than the threshold value are flagged as low coherence areas. Finally, only the detected water pixels (i.e., the binary detected water mask output from the WaterDetection step above) that are low coherence are flagged as low coherence water.

3.3.7.4 Accuracy

The accuracy of coherence is particularly poor with low numbers of looks, which is one reason why we do the smoothing, but after tens of looks the coherence becomes asymptotically more unbiased. At this level of smoothing, there are still noticeably many individual pixels or small regions that are flagged as low coherence simply because of speckle. These tend to occur on the edges of water bodies where we have detected water, but the pixels are slightly less bright because of the coherence time smearing mixed with the particular speckle realizations. Nevertheless, for what this flag may be used for in down-stream processing, these “false” low coherence areas are tolerable.

Figure 29 shows an example of the low-coherence flagging algorithm in a simulated scene of a dam with a river outlet that exhibits water-water layover. Layover can impact the phase estimate because multiple targets from different places on the ground at different heights (and thus different phases) can map into the same range bin (slant-plane pixel). Because of the low incidence angle for SWOT layover occurs often; however, in most situations (e.g., when dark-scattering land lays over into bright-scattering water) the impact is generally not too severe [cite layover paper]. This is not the case when a water feature lays over into another water feature, where since both have a similar brightness, the phase is significantly corrupted. This phase corruption comes in two flavors: 1) it biases the phase estimate to be somewhere far from either water surface, and 2) it reduces the coherence and thus the phase noise or variability of the phase estimate. There is nothing that can be done to mitigate the bias (unless somehow already precisely know the height and the brightness of one of the water surfaces); however, the drop in coherence can be useful information e.g., for flagging purposes etc (e.g., at least we can know if the phase is corrupted and should not be trusted). In Figure 29 the normalized coherence is plotted for various smoothing candidates (unfiltered, simple median filter, and vector median filter) as well as a colored image with the final binary flag of the low-coherence, bright water classification. These images show the river outlet and the reservoir laying over (since the reservoir is higher than the outlet, but farther in cross-track, they map in to the same bins in the slant-plane images), as represented by the dark snaking feature (river) near the left bottom of the lake. The low-coherence flag seems to appropriately detect the water-water-layover regions in this case.

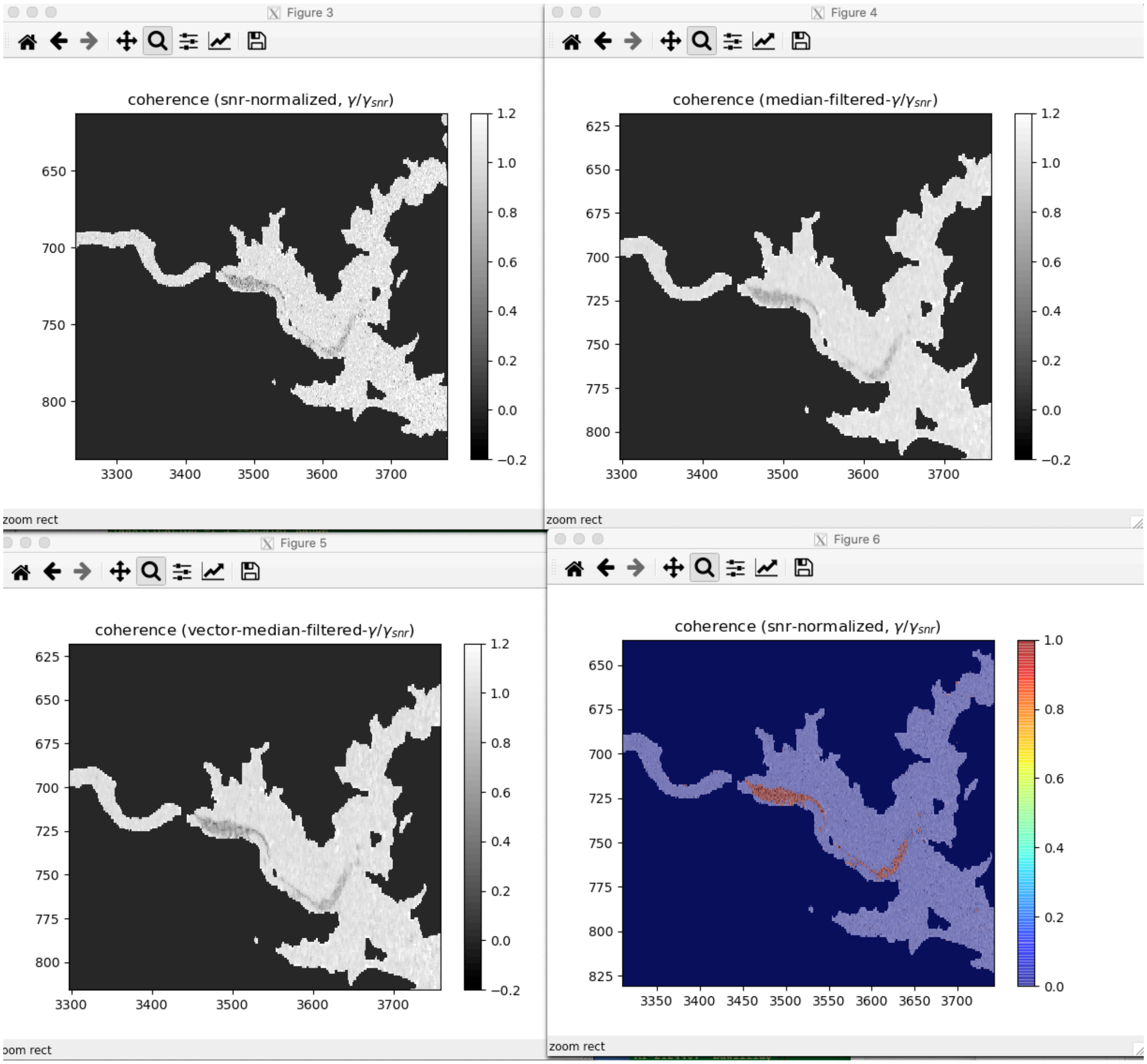


Figure 29. Example of detected low-coherence water that is caused by water-water layover.

3.3.8 BuildClassMaps

This function collects the fields from the water detection, dark-water flagging, layover, bright land flagging, and low-coherence flagging functions and computes a multivalued water mask classification with land/water edges identified. Recall that Figure 11 in Section 2.3.1 shows the multivalued classification for a simulated river scene.

3.3.8.1 Input Data

Description	Source
Detected water, detection rates (WaterMap)	WaterDetection
Water fraction, water fraction uncertainty (WaterFrac)	WaterDetection
Estimated layover height error image (LayoverMap)	LayoverFlagging
Dark water mask and conditioned reference locations (DarkWaterMap)	DarkWaterFlagging
Bright land map (BrightLandMap)	BrightLandFlagging
Low-coherence water map (LowCohMap)	LowCohFlagging

3.3.8.2 Output Data

Description: ClassMaps
Multivalued classification image representing detected water, dark water, low-coherence detected water, and land/water edges
Detection rate images (e.g., false detection and missed detection)
Water fraction estimate image
Water fraction uncertainty estimate image
Estimated layover height error image
Conditioned reference locations (passed from DarkWaterMap)
Bright land map (passed from BrightLandMap)

3.3.8.3 Mathematical Statement

The multivalued classification mask is produced by first flagging any pixels in the prior inclusion/pruning mask, which has been mapped into the slant-plane with the same reference locations as the dark water prior and the bright land prior masks. Next a slant-plane binary water mask that includes the detected (also low-coherence) and dark water is generated and a morphological dilation operation is applied with 10 iterations. This mask is combined with the inclusion mask to produce the “keep buffer” which includes all pixels that are kept in the final pixel cloud product. The multivalued classification slant-plane image is initialized to 0, and then those pixels in the keep buffer are set to 1. Next, the land edges near the detected water mask are identified using morphological binary dilation (one iteration with a simple 3x3 square kernel) and the classification image is set to 2 for those pixels. Then, water pixels (detected and dark) are set to value of 4 in the classification image (overwriting the values of 2 on the detected/dark water boundary). Then, water edge (detected, low-coherence, or dark) pixels are identified using morphological erosion (3x3 square kernel), and then to handle extreme azimuth smearing an extra erosion is performed with a kernel in the azimuth direction only, so that water pixels that are a distance of 2 pixels away in the azimuth direction are also flagged as water edges. These, pixels are set to 3 in the classification image (overwriting the values of 4 where applicable). Next, the dark water pixels are set to value of 5 (overwriting any other classes). The dark-water-outer edges are then detected using dilation and the pixels that are a dark-water outer edge and an interior water edge (previously set to 3) are set to the value 4 (interior water) in the classification image. Finally, the low-coherence values are set to 6 if the class value is 3, and 7 if the class value is 4. To summarize the multivalued classification is produced with the following classification mappings:

1 = land in keep buffer
2 = land on a water edge
3 = detected water on a land edge
4 = detected water not on an edge (i.e., interior water)
5 = dark water
6 = low coherence detected water edge
7 = low coherence detected water (not on an edge)

Other fields from the inputs are simply passed through to the output.

3.3.8.4 Accuracy

The BuildClassMaps operation only combines previously computed quantities into a multi-valued classification variable, and identifies edges using morphological operations. It does not introduce additional errors and is accurate to within floating point precision. Note that adjacent pixels in the slant-plane do not always come from adjacent regions on the ground. Since the land/water edge values are computed in the slant-plane, this property is preserved after edge masking.

3.3.9 Accuracy of MakeClassMaps

The accuracy of MakeClassMaps can be broken down into 3 parts: 1) the individual accuracy of the algorithms within it (e.g., water detection, dark water flagging etc.), 2) the interaction or dependency between these algorithms, and 3) interaction of the algorithms with the downstream modules and algorithms (e.g., phase unwrapping). The first item is discussed in the individual subsections above, the third item is discussed in the roll-up accuracy discussion below, and the second item is discussed here. The accuracy of each of the steps in MakeClassMaps is somewhat dependent on the water detection accuracy.

Consider the dark water flagging algorithm dependance on the detected water mask. If the water detection is poor, then the dark water flagging algorithm is expected to have more trouble precisely coregistering the mask in the slant-plane, resulting some cross-track shifts relative to where the true water is due to inaccuracies of the reference DEM used for the projection. Furthermore, the probability thresholding is also likely to have trouble if the detected water mask has significant errors. That being said, the dark water flag can often be relatively accurate even without the use of the detected water mask, especially where the reference DEM is good, and where the probability map is accurate and relatively consistent over most typical constant probability thresholds (which is a proxy for water-levels).

Although there is dependance of the bright land flagging on the detected water mask, the impact is minimal. This is because the bright land flag is primarily used as a quality indicator, attempting to flag false detections from bright, non-water targets (as opposed to an estimate of a geophysical quantity itself). For example, if bright targets are detected where no water is detected, then there is no harm done if the bright land mask had errors. If on the other hand water is detected where the bright land mask is set, it either correctly indicates a false detection, or it is actually water, but near a bright target. For the case where there is true water and the bright land flag is set (e.g., an urban development near the shore that perhaps is shifted into the true water pixels in slant-plane due to imprecise reference DEM), these are regions that likely have a suspect quality anyways (e.g., corrupted by urban layover).

Now consider the Layover flagging dependence on the water mask (both the detected and the prior/dark-water mask). Since the impact of layover is highly dependent on the relative brightness of the targets being laid-over and the brightness contrast between water and non-water surfaces is so large, errors in either the slant-plane detected water mask, or the prior water mask (dark water flag) will significantly impact the accuracy and utility of the layover impact quantity.

The low-coherence flagging is minimally impacted by the accuracy of the detected water. The purpose of this flag is to try to separate the detected water (not the truth water) into two different kinds of bright (e.g., detected water) targets: those that look like typical water, and those that are bright but decohere—primarily due to height/phase variations within a pixel (e.g., from water-water layover, land-land layover detected as water, or significant volume scattering from vegetation). Note in the case of missed detections of water, the low-coherence flag as implemented as depending of the detected water as opposed to the brightness will miss those regions. There may be some value in exploring the a low-coherence, bright target flag that is independent of the detected water mask.

3.4 MakeMedIntf

3.4.1 Purpose

This function applies additional, class-dependent multilooking on the rare interferogram to produce the so-called medium interferogram. The delta corrections to the rare geophysical and crossover calibration corrections are computed. The medium level of smoothing is needed to reduce the phase noise sufficiently so that geolocation and height reconstruction become quasi-linear over the width of the phase noise distribution, so that additional aggregation of the geolocated heights are approximately equivalent to additional multilooking before geolocation. The ideal medium level of smoothing takes on the order of 50 effective looks for typical water brightness in the central part of the swath. Since water features can generally be narrower than 50 looks worth of slant-range pixels, the medium smoothing must take into account the classification of the pixels that are being multilooked so as to not mix water with land pixels. Recall that Figure 12 in Section 2.3.1 shows the medium interferogram phase and coherence for a simulated river scene.

3.4.2 Input Data

Description	Source
Rare interferogram, including the two channel powers and complex interferogram (RareUnflatIntf)	MakeRareIntf
The multi-valued classification and conditioned reference locations from (ClassMaps)	MakeClassMaps
Geophysical inputs: wet troposphere, dry troposphere, ionosphere	DynamicGeophys
Crossover calibration inputs	XOverCal

3.4.3 Output Data

Description: MedFlatIntf
Medium Interferogram image on the rare-grid posting, including complex interferogram and the individual two channel powers
Conditioned reference locations on the rare-grid (simply passed through)
Satellite orbit TVP data (simply passed through)

3.4.4 Mathematical Statement

The adaptive multilooking approach is relatively simple. First, the phase of the rare interferogram is once again flattened using the conditioned reference locations created in the dark water flagging step. Next, each classification in the multivalued classification is filtered separately, using a moving window to multilook every pixel in the window in a predetermined list of class precedence. The class precedence enables, for example, a dark water (or also a low-coherence) pixel to incorporate neighboring bright water pixels, but for bright water pixels not to incorporate neighboring dark (or also low-coherence) pixels. This can be expressed as

$$Q_{med}(x) = \sum_{y \in w_x} Q(y) \cdot b_x(y) \qquad (3.4.4.1)$$

where Q represents the quantity to be multilooked (e.g., the complex interferogram, or reference

SLC or secondary SLC power images), where

$$b_x(y) = \frac{I_p\big(d(x),d(y)\big)}{\sum_{y\in w_x} I_p\big(d(x),d(y)\big)} \tag{3.4.4.2}$$

represents a weighting kernel composed of uniform, normalized weights for pixels y of the classes d that are allowed by the class precedence of the center pixel x over the local window w_x around the center pixel ($y \in w_x$), with $I_p\big(d(x), d(y)\big)$ representing an indicator function, which takes on only values of 1 if the class in the second argument, $d(y)$, is in the precedence allowed by the class in first argument $d(x)$ and a value of 0 otherwise. The following equations show the indicator function values for each combination of first and second arguments.

$$I_p(0,d) = \begin{cases} 1 \; if \; d \in \{0\} \\ 0 \; otherwise \end{cases} \tag{3.4.4.3}$$

$$I_p(1,d) = \begin{cases} 1 \; if \; d \in \{1\} \\ 0 \; otherwise \end{cases} \tag{3.4.4.4}$$

$$I_p(2,d) = \begin{cases} 1 \; if \; d \in \{2\} \\ 0 \; otherwise \end{cases} \tag{3.4.4.5}$$

$$I_p(3,d) = \begin{cases} 1 \; if \; d \in \{3,4\} \\ 0 \; otherwise \end{cases} \tag{3.4.4.6}$$

$$I_p(4,d) = \begin{cases} 1 \; if \; d \in \{4\} \\ 0 \; otherwise \end{cases} \tag{3.4.4.7}$$

$$I_p(5,d) = \begin{cases} 1 \; if \; d \in \{3,4,5\} \\ 0 \; otherwise \end{cases} \tag{3.4.4.8}$$

$$I_p(6,d) = \begin{cases} 1 \; if \; d \in \{6,7\} \\ 0 \; otherwise \end{cases} \tag{3.4.4.9}$$

$$I_p(7,d) = \begin{cases} 1 \; if \; d \in \{7\} \\ 0 \; otherwise \end{cases} \tag{3.4.4.10}$$

That is, all classes get averaged with their neighboring pixels of the same class, while the dark water class (5) also folds in the bright edge and interior water classes (3 and 4), the edge water (3) also folds in the interior water class (4), and the edge low-coherence class (6) also folds in the interior low-coherence class (7). This enables the data that is expected to be more reliable to be averaged in with the potentially less reliable classes, while enabling the aggregation of the more reliable classes not to be impacted by the potentially less reliable classes.

For our baseline case we use a 3x3 window for w_x, which corresponds to at most nine rare pixels, each of which corresponds to approximately 4 independent raw looks. Therefore, for water bodies that are large enough, a medium pixel includes approximately 36 independent looks (except at the edges). This provides a good enough trade to get close the ideal number of looks without overcomplicating the algorithm (because the water-body level aggregated heights become decreasingly sensitive to amount of multilooking as more looks are taken). Note that this adaptive multilooking preserves the original posting, and so the rare and medium interferograms are on the same slant-plane grid posting (i.e., the rare-grid, with no downsampling).

This adaptive multilooking is applied to the complex interferogram as well as the two channel powers. No coherent power is produced for the medium interferogram. Also, the reference locations are not multilooked, but the rare reference locations (produced in the dark water flagging function) are used in subsequent processing.

3.4.4.1 *Calibration and Geophysical Corrections*

Here the same corrections applied in Section 3.2.4.1 are computed (except for the atmospheric attenuation). That is, the corrections $r_{media,i}(x)$, $r_{xover}(x)$, $\phi_{media}(x)$, $\phi_{drift}(x)$, and $b_d(x)$ are recomputed using the more accurate reference locations based on the reference DEM (the conditioned reference locations). The rare corrections are subtracted from these newly computed corrections, producing the delta corrections. Note that for the media and crossover range corrections the bulk (constant) rare versions are subtracted, e.g.,

$$\Delta r_{media,i}(x) = r_{media,i}(x)|_{med} - r_{media} \tag{3.4.4.1.1}$$

$$\Delta r_{xover}(x) = r_{xover}(x)|_{med} - r_{xover} \tag{3.4.4.1.2}$$

while for the secondary media range correction the rare phase correction is also incorporated to remove the rare correction when computing the medium phase correction, i.e.,

$$\Delta r_{media,2}(x) = \Delta r_{media,2}(x) - \frac{\lambda}{2\pi} \cdot \phi_{media}(x)|_{rare} \tag{3.4.4.1.3}$$

Each of the other terms subtract the corresponding rare quantity directly, e.g.,

$$\Delta\phi_{media}(x) = \phi_{media}(x)|_{med} - \phi_{media}(x)|_{rare} \tag{3.4.4.1.4}$$

$$\Delta\phi_{drift}(x) = \phi_{drift}(x)|_{med} - \phi_{drift}(x)|_{rare} \tag{3.4.3.1.5}$$

$$\Delta b_d(x) = b_d(x)|_{med} - b_d(x)|_{rare} \tag{3.4.3.1.6}$$

where the notation $|_{med}$ and $|_{rare}$ refer to versions of those same quantities computed in the medium and rare multilooking steps, respectively. Note that these “delta” corrections are not applied to the corresponding quantities at this stage, but rather are used in the geolocation steps later on.

3.4.5 Accuracy

Adaptive multilooking technically changes the centroid of where the signal comes from on the ground if the classification shape is not symmetric. In general, one would need to smooth the reference locations similarly to avoid geolocation and height bias artifacts due to asymmetries in water-body feature shape. However, since the window we use is small and we are applying the reference phase to each pixel rather than to the entire feature, this effect is negligible and the same rare reference locations can be used as the medium reference locations.

The adaptive multilooking approach reduces resolution (destroys some high spatial-frequency information) within features, while preserving (or even sharpening) edges between different classes. More importantly, however, the multilooking reduces the phase noise over regions with quasi-homogeneous statistics in advance of geolocation.

(TODO: comment of the accuracy of the corrections at these new reference locations?)

3.5 UnwGeoloc

Due to its complexity, this algorithm is split into the different pieces that are shown in Table 3 and Figure 30. The primary objectives of this algorithm are to perform spatial phase unwrapping, region breaking, phase ambiguity resolution, and height reconstruction. This section summarizes the baseline implementation for the SWOT phase unwrapping approach.

In order to reconstruct the height and determine the geolocation of each pixel of the interferogram, information on the range difference between the ranges of that scatterer at the pixel to the two interferometric antenna phase centers is needed. That range difference is embodied in the phase of the interferogram but the phase is wrapped to be within 2π. The purpose of phase unwrapping is to determine how many multiples of 2π (or phase ambiguities) need to be added to the face value of the measured interferometric phase of each pixel to get the so-called "absolute phase".

The term "phase unwrapping" in this document refers generally to the first three steps of 1) spatial phase unwrapping of the interferogram into regions of self-consistent spatially relative phase, 2) region breaking to minimize remaining phase discontinuities within regions, and 3) ambiguity resolution to estimate what multiple of 2π should be added to each region. The geolocation step takes the absolute phase for each pixel and maps it to the corresponding location and height.

For SWOT, the first three "phase unwrapping" steps are only applied to detected water pixels (e.g., classes 3,4,6,7) as opposed to the full interferogram image, and not all detected water pixels will be successfully unwrapped (e.g., in low coherence regions). The remaining pixels that are not phase unwrapped are geolocated using the reference location to resolve the 2π phase ambiguity. Recall that Figure 13 in Section 2.3.1 shows the phase unwrapping regions and the absolute phase for a simulated river scene.

Table 3. High level description of the functions within the UnwGeoloc function.

Function Name	Description
SpatialUnwrap	Computes spatially unwrapped interferometric phase over connected regions (InitUnwRegions)
BreakRegions	Computes smaller unwrapping regions to minimize height inconsistencies within regions (MedUnwRegions)
AmbigSolver	Computes absolute ambiguity for each unwrapping region (MedUnwIntf)
Geolocate	Computes reconstructed heights and geolocations, applying media corrections (MedPixFull)

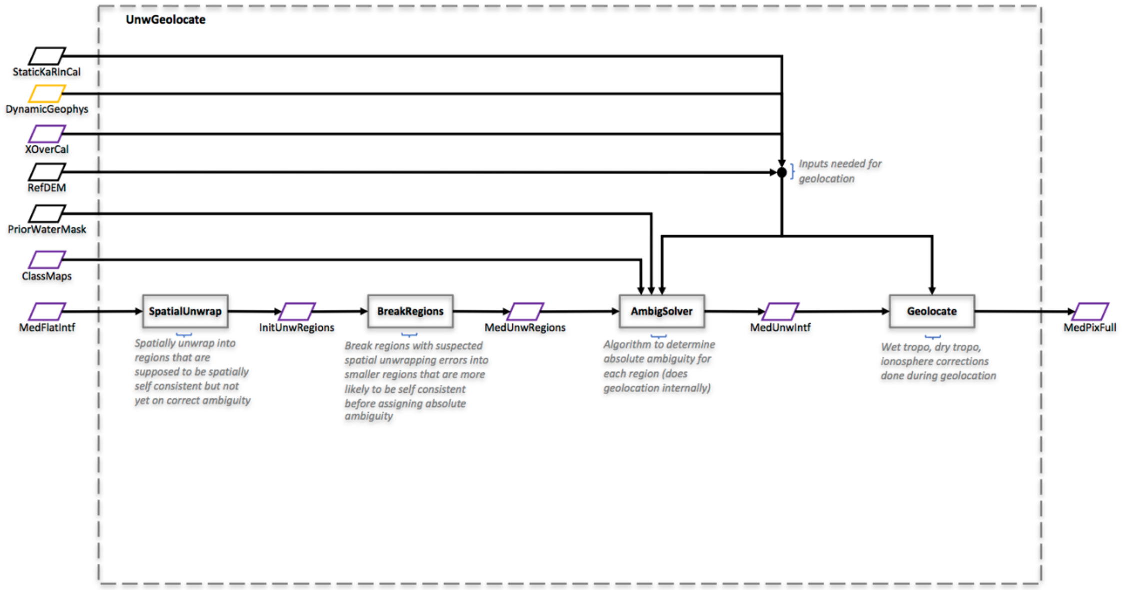


Figure 30. Flow diagram of lower-level algorithms within the UnwGeoloc function. (TODO: update this flow diagram with all inputs [except ClassMaps?] also going to BreakRegions, and I think ClassMaps going to SpatialUnwrap).

3.5.1 SpatialUnwrap

This function identifies spatially disjoint regions in the slant-plane and spatially unwraps the phase over those regions.

Note that the incidence angles, the high coherence of water for SWOT, and the fact that water tends to be relatively flat make this problem different than for typical phase unwrapping in other interferometric SAR systems. The most obvious idiosyncrasy with SWOT is that the spatial extents of the regions that can be unwrapped and geolocated can be relatively small and disjoint and correspond mainly to the bright surface water areas. Also, because of the low incidence angle viewing geometry, issues related to layover can be more significant for SWOT. Furthermore, land and dark water are expected to be often not bright enough to reliably unwrap and geolocate, because the phase is corrupted by the relatively high noise floor (and layover). Moreover, most hydrology applications are primarily interested in the water heights as opposed to the land heights (at least for the inline produced L2 products; the algorithms related to the floodplain DEM may have different considerations). Thus, the phase unwrapping method here focuses on properly geolocating the water pixels/regions.

3.5.1.1 Input Data

Description	Source
Medium interferogram phase, coherence, power (MedFlatIntf)	AdaptiveML
ClassMaps containing detected water	MakeClassMaps

3.5.1.2 Output Data

Description
Spatially unwrapped phase of each isolated region with the 2π ambiguity unresolved for each region (InitUnwRegions)

3.5.1.3 Mathematical Statement

The two-dimensional phase unwrapping problem has been studied for over 40 years. Several kinds of algorithms have been proposed to solve this problem. Most of these algorithms can be divided into three categories: Branch-Cut based algorithms [Goldstein 1988, Flynn 1997, Constantini 1998, Chen 2000], region-growth [1999 Xu] based algorithms and least-squares algorithms [Ghiglia 1997, 1998]. The most used algorithms for phase unwrapping in interferometry processors are branch-cut based. The original branch-cut algorithm [Goldstein 1988] and its modifications can find a solution quickly and still is the top choice for many interferometry applications. But it is not an optimal solution and in some cases it leaves many areas not unwrapped. For the SWOT mission, we have implemented two branch-cut based algorithms: PHASS [2017 WU] and SNAPHU [2000 and 2001 Chen], because they are optimal algorithms in the sense that they provide phase unwrapping solutions with a minimum cost. The baseline algorithm applies the PHASS algorithm for spatial phase unwrapping.

The general idea of spatial phase unwrapping is to start from a seed pixel and add a phase value of a multiple of 2π to each of its neighbor pixels so that the phase difference between any two neighboring pixels is smaller than π as often as possible. We say there is a phase discontinuity if the absolute difference of the phase between two neighboring pixels is bigger than π. The algorithms PHASS and SNAPHU aim to minimize some forms of the total phase discontinuity in the unwrapped interferogram. The details of these algorithms are described in the references [WU 2017] and SNAPHU [Chen 2000/2001].

For SWOT, pixels that are not detected as water or which have insufficient coherence are masked out of the spatial phase unwrapping algorithm. The output of the spatial phase unwrapping process includes the unwrapped phase image and a region index map, or simply region map. The region index value of -1 is assigned to each of the non-water pixels. Every pixel in each water region is assigned a unique, non-negative region index starting from 0. The unwrapped phase of the detected water areas coming out from the spatial unwrapping should have minimum phase discontinuity and the initial 2π ambiguity of zero is assigned to each water region since we assume the reference DEM with respect to which the interferogram is flattened is the best prior DEM available.

Note that there are some pixels that end up in the pixel cloud product, which may not be included in these phase unwrapping regions (e.g., dark water and land near the water or in the floodplain inclusion mask). The next few processing stages compute the absolute phase only for pixels in the spatial phase unwrapping regions. The remaining pixels simply compute the absolute phase by adding the noisy, wrapped, flattened phase to the absolute phase of the reference locations (which have been conditioned so that water and land near water are flat with approximately the same elevation as the water).

3.5.1.4 Accuracy

Spatial phase unwrapping is very sensitive to the structure of a given scene. It is difficult to characterize quantitatively how spatial phase unwrapping performs in general for hydrology purposes. Accuracy assessments are therefore made at a higher level in Section 3.5.3.4.

3.5.2 BreakRegions

The purpose of this function is to break up the phase unwrapping regions when they have pixels or subregions with conflicting 2π phase ambiguities with the rest of the region so that after breaking the regions, each smaller region is self-consistent. This attempts to ensure that the phase and geolocated heights are self-consistent within each region. This minimizes issues in the step that chooses the absolute ambiguity arising from having different portions of a single region that should be assigned to different ambiguity numbers.

3.5.2.1 Input Data

Description	Source
Initial phase unwrapping regions as well as the necessary quantities from the medium interferogram (InitUnwRegions)	SpatialUnwrap
Prior water occurrence maps (PriorWaterMask)	Static Auxiliary Data
High resolution reference DEM (RefDEM)	Static Auxiliary Data
XOverCal	XoverCal Intermediate Data
DynamicGeophys	Dynamic Auxiliary Data
StaticKaRinCal	Static Auxiliary Data

3.5.2.2 Output Data

Description
Refined phase unwrapping regions as well as the necessary quantities from the medium interferogram (MedUnwRegions)

3.5.2.3 Mathematical Statement

In this method, a local geolocation is performed on each region and the height statistics of each region are analyzed to determine if they should be broken down further into smaller regions. The local geolocation uses the same methods as described in the section describing the "Geolocate" function below. For more details on the geolocation portion of this algorithm see Section 3.5.4.

The method for breaking regions is based on statistical image segmentation. Several methods have been explored, including simple height histogram segmentation, superpixel segmentation, and minimum cost function segmentation. The baseline approach uses a minimum cost function approach specially designed for the SWOT measurement. The cost function has two components related to the two major types of errors caused by selecting an incorrect phase unwrapping ambiguity number: 1) a large residual height error with respect to the reference DEM, and 2) a large cross-track geolocation offset error.

In order to reconstruct the correct height from the unwrapped interferometric phase, one needs also to know the correct 2π ambiguity number. A 2π phase ambiguity error will cause a height error known as the ambiguity height. At the same time, it also causes a cross-track location error. The ambiguity height for SWOT geometry varies from about 10 m to about 60 m from near- to far-range of the swath, while the cross-track error for a single ambiguity error is approximately 750 meters, almost independent of where it is in the swath. Figure 31 shows the ambiguity height (top plot) and the cross-track shift (bottom plot) for a one-ambiguity (i.e., 2π) error as a function of cross-track. If our reference DEM is accurate to half of the ambiguity height, which is about 5 meters at near range, there would be no problem determining the ambiguity numbers based on the reference DEM height alone. However, the global reference DEM is not generally accurate to 5 m. For example, the accuracy of the DEM from the Shuttle Radar Topography Mission (SRTM) is on the order of 10 m. Furthermore, even if we had a perfect DEM, some water features may have elevations that vary several meters or more from pass to pass. Since the static DEM is not generally accurate enough to uniquely pick the ambiguity closest to the reference DEM everywhere in the swath, this makes the algorithm for resolving the phase ambiguity for each region somewhat complicated.

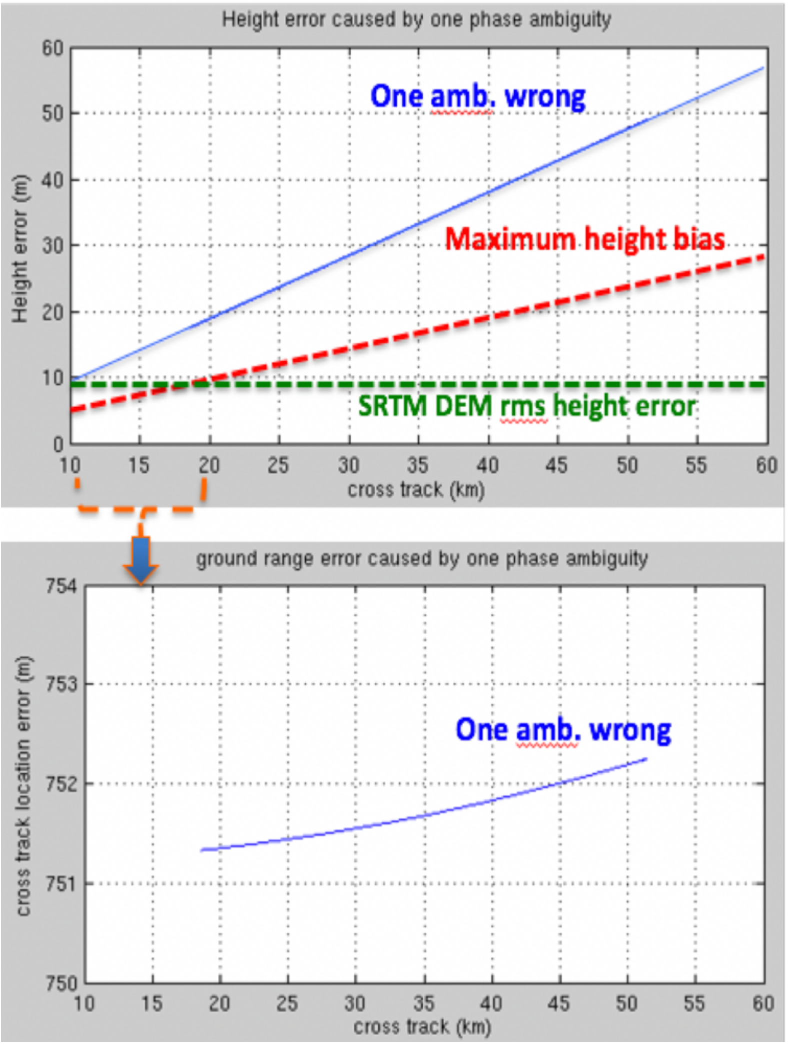


Figure 31 Ambiguity height versus cross-track (top) and cross-track shift for one ambiguity error versus cross-track (bottom).

The reference DEM can be used to help break up regions with ambiguity conflicts by making use of the residual height (reconstructed height – reference height) for each of the possible ambiguities. The residual height for a pixel x can be expressed as

$$\Delta_{h,x}(a) = h_x(a) - h_{x,ref}(a) \tag{5.3.2.3.1}$$

where $h_x(a)$ is the reconstructed height for ambiguity a (where a is the integer number of 2π cycles added to the wrapped phase to obtain the unwrapped phase), and $h_{x,ref}(a)$ is the height of the reference DEM at the reconstructed location for ambiguity a. Note that the residual height does not compare the reconstructed height to the true height but to the height of the reference DEM after geolocating with a possible incorrect value of a. That is, the reference DEM may be evaluated at the wrong cross-track location (shifted by a multiple of the 750 m cross-track geolocation error associated with each incorrect phase ambiguity). While imperfect, the residual height is nonetheless still a reasonable indicator for which ambiguity is likely to be correct for some part of the region.

Considering all the pixels of a region, we define the square root mean square deviation of the residual height to describe how close the reconstructed height is to the reference DEM height:

$$\Delta_{h,n}(a) = \sqrt{\frac{1}{N_n}\sum_{x=1}^{N_n}\left(h_x(a) - h_{x,ref}(a)\right)^2} \qquad (5.3.2.3.2)$$

where N_n is number of pixels x of region n, and a is the phase ambiguity.

The prior water mask can also be used to help break up regions by comparing the detected water mask with the prior mask in the same domain (either the slant-plane or the ground plane) assuming a particular ambiguity number for each region. The correlation coefficient between the detected and prior water masks for a particular ambiguity a for region n [denoted $\rho_n(a)$] is a good indicator of ambiguity error if the true (or detected) water mask is consistent with the prior. Since only detected water pixels are included in unwrapping regions, $\rho_n(a)$ is defined as

$$\rho_n(a) = \frac{\sum_{x=0}^{N_n} w_x(a)\cdot\Pi_x(a)}{\sqrt{\sum_{x=0}^{N_n} w_x^2(a)\cdot\sum_{x=0}^{N_n}\Pi_x^2(a)}} \qquad (5.3.2.3.3)$$

where $w_x(a)$ is flag of the reconstructed pixel x for the ambiguity a and $\Pi_x(a)$ is the corresponding prior water probability of $w_x(a)$. The value of the flag $w_x(a)$ is either 0 or 1, depending on whether the location of the pixel on the ground is occupied by pixels from another region whose ambiguity number has already been assigned. If the location is occupied, its value is 0, otherwise the value is 1. The prior probability map provides a water probability percentage between 0 and 100. Using O_n to represent the number of overlapped pixels of the region n with the reference water mask, the correlation coefficient $\rho_n(a)$ can be simplified as:

$$\rho_n(a) = \frac{\sum_{x=0}^{O_n}\Pi_x(a)}{\sqrt{O_n\cdot\sum_{x=0}^{N_n}\Pi_x^2(a)}} \qquad (3.5.3.3.4)$$

The region breaking algorithm uses both the residual height and the correlation between the water mask and the detected water bodies and applies a cost function for each pixel as a function of the phase ambiguity as follows:

$$J_n(a) = c_1\cdot\left(\frac{\Delta_{h,n}(a)}{\sigma_{DEM}}\right)^2 + c_2\cdot\left(1-\rho_n^2(a)\right) \qquad (5.3.2.3.5)$$

where a is the phase ambiguity index, and c_1 and c_2 are positive constant weights to balance the contributions of the first residual-height related term and the second correlation-with-the-water-mask related term. σ_{dem} is the height uncertainty of the reference DEM and is dependent on the quality of the DEM, which for the SRTM DEM is about 10 meters on average. Both the residual height and the correlation coefficient vary with the phase ambiguity number a, resulting in an ambiguity-dependent cost function. The cost will be small if the residual height is small and ρ is large. Therefore, the ambiguity corresponding to the minimum cost would likely be the correct ambiguity.

The determined ambiguity corresponding to the minimum cost is then used to reconstruct the height and geolocation of each pixel of the region. All the pixels overlapping with non-zero water probability in the water mask are assigned the minimum cost ambiguity. Those left-over pixels of the same region with no overlap with the water mask go through the same process again until all the left-over pixels formed regions have a region size smaller than a predefined region size threshold, whose default value is 1000 pixels and can be specified in the auxiliary parameter file. The spatially connected pixels with the same assigned minimum cost ambiguity are connected to form a (most likely smaller) region.

The above-described region-breakup process is done for each region sequentially, starting from the largest and the farthest range regions. This process produces many small regions, which makes the next step of resolving the absolute ambiguity more difficult. Therefore, an additional region merging routine is applied to minimize these small regions by merging them back into one of their neighbors in a judicious way. The region merging algorithm applied can be described as follows:

Region Merge Algorithm:
(1) Sort the regions in ascending order based on the total number of pixels of each region. Set a size threshold Th.
(2) Starting with the smallest region R. Skip this region if the region size is smaller than the threshold Th.
    (2-1) check all the border pixels of the region and find all the neighboring pixels from regions other than this region R. Find the neighboring region Rn with the most border pixels with region R.
    (2-2) merge all the pixels of R into the neighbor region Rn: the ambiguity of each pixel of R is assigned the ambiguity of region Rn.
(3) Repeat (2) until all the regions are checked.

As a third and last step in the region breakup process, each region is checked again for a possible ambiguity conflict. In this step, the height of each pixel is evaluated using the same ambiguity and the height differences between all 4 neighbor pixels are compared with the local ambiguity height. Only the pixels with height differences less than half of the ambiguity height are connected to form a region.

Once the 3-step region breaking process is finished, the final regions should have minimal ambiguity conflicts. This means that by assigning one (hopefully correct) ambiguity to each region, the height can be reconstructed for all the pixels in each region reliably. The next section focuses on assigning the correct ambiguity to each region of the interferogram and producing the correct unwrapped phase without ambiguity (or absolute phase).

3.5.2.4 Accuracy

The region breaking and remerging algorithm does not directly introduce additional error here; however, the choice of methods does have bearing on the accuracy of the final geolocated heights. For example, the bigger the region, the more likely it will find the correct absolute ambiguity (unless there are internal conflicts or artifacts). Breaking the regions into smaller, but more self-consistent regions, reduces the chance of a phase conflict causing a large portion of the region to be on the wrong ambiguity, while increasing the probability that the smaller regions will be placed on the wrong ambiguity simply because of noise variability.

3.5.3 AmbigSolver

The first 2 steps out of the 3-step phase unwrapping procedure have been described in the last two sections. This section focuses on the third step: resolving the ambiguity of each region, thereby estimating the absolute phase for each pixel in each region. After the first 2 steps of spatial phase unwrapping and region breakup, there are many regions. These regions are spatially separated, yet at the same time the pixels within each region are spatially connected. This step attempts to determine one ambiguity number for each region so that adding the same phase ambiguity to the original unwrapped phase (plus the known flattening phase) of each pixel of the region gives us the correct absolute phase of each pixel of the region.

As we know from Figure 31 that one ambiguity of phase error will cause height errors from about 10 m at near range to 60 m at far range and about 750 m of cross-track geolocation error. As a side effect of the height error caused by ambiguity error, there is also a cross-track slope error of about 50 m/50 km from one phase ambiguity error, as is shown also in Figure 31. The algorithm for resolving the phase ambiguity makes use of all three of these errors (residual height, cross-track shift, and cross-track slope) to attempt to compute the ambiguity number for each region.

Note that the cross-track slope is most useful for large flat water bodies (like larger lakes, as opposed to rivers) due to two reasons: (1) its cross-track slope can be measured more reliably because of its large size; (2) even if the reference DEM height at the lake is biased, the lake would likely remain flat (at least with respect to the geoid). The cross-track slope is, however, not so useful for long, narrow rivers because rivers may have their own slopes, and the reference DEM may not be good enough to be used for residual slope calculation. Because of this, the cross-track slope is an optional measure for ambiguity determination, and it is currently turned off by default in the baseline algorithm.

3.5.3.1 Input Data

Description	Source
ClassMaps containing detected water	MakeClassMaps
Prior water mask	Static Auxiliary Data
RareUnflatIntf containing the rare reference locations	MakeRareIntf
XOverCal	XoverCal Intermediate Data
DynamicGeophys	Dynamic Auxiliary Data
StaticKaRinCal	Static Auxiliary Data

3.5.3.2 Output Data

Description
MedUnwIntf

3.5.3.3 Mathematical Statement

This section mathematically describes the algorithm used for determining the phase ambiguity of each detected region. The idea is to define a cost function to jointly use the 3 pieces of information described in the previous sections so that the cost function is a function of the ambiguity, and its minimum value would likely correspond to the correct ambiguity.

First, consider one region independently. The minimum residual height corresponds to the correct ambiguity in most cases if the reference DEM is accurate, although this is not guaranteed (e.g., if the DEM height at the location of an incorrect ambiguity is close to the corresponding multiple of the ambiguity height). Therefore, a cost term related to the residual height is defined as the square root mean square deviation of the residual height:

$$\Delta_{h,n}(a) = \sqrt{\frac{1}{N_n}\sum_{x=1}^{N_n}\left(h_x(a) - h_{x,ref}(a)\right)^2} \qquad (3.5.3.3.1)$$

where $h_x(a)$ is the reconstructed height and $h_{x,ref}(a)$ is the reference DEM height, N_n is number of pixels x of region n, and a is the phase ambiguity. The corresponding reference DEM height $h_{x,ref}(a)$ of the reconstructed height $h_x(a)$ is found by calculating the geolocation of each pixel of the region and locating it on the reference DEM.

The second piece of information is the match between the prior water mask and the detected water regions. The best match likely corresponds to the correct ambiguity. The following correlation coefficient is used:

$$\rho_n(a) = \frac{\sum_{x=0}^{N_n} w_x(a)\cdot\Pi_x(a)}{\sqrt{\sum_{x=0}^{N_n} w_x^2(a)\cdot\sum_{x=0}^{N_n}\Pi_x^2(a)}} \qquad (3.5.3.3.2)$$

where $w_x(a)$ is flag of the reconstructed pixel x for the ambiguity a and $\Pi_x(a)$ is the corresponding prior water probability of $w_x(a)$. The value of flag $w_x(a)$ is either 0 or 1, depending on whether the location of the pixel on the ground is occupied by pixels from another region whose ambiguity number has already been assigned. If the location is occupied, its value will be zero, otherwise the value will be one. The prior probability map provides a water probability percentage between 0 and 100. Using O_n to represent the number of overlapped pixels of the region n with the reference water mask, the correlation coefficient $\rho_n(a)$ can be simplified as:

$$\rho_n(a) = \frac{\sum_{x=0}^{O_n}\Pi_x(a)}{\sqrt{O_n\cdot\sum_{x=0}^{N_n}\Pi_x^2(a)}} \qquad (3.5.3.3.3)$$

Since the best match between the region and the priori water mask corresponds to maximum correlation 1, the cost function is defined as $1 - \rho_n^2(a)$ for this term. In this way the minimum cost corresponds to the maximum match and most likely the correct ambiguity.

The maximum match between the detected water and the prior water mask can lead to an incorrect ambiguity in some cases mainly because of the following reasons: (1) the detected water region may be just a small part of the real water body due to dark water (low SNR in water areas) and higher image noise level; (2) nearby land areas are falsely detected as water; (3) a real water body may be broken into many smaller parts in the prior water mask map itself. Therefore, combining other information like the residual height helps to improve the chance of successful estimation of the ambiguity.

The third piece of information is the cross-track slope (although this information is neglected in the baseline implementation). One cross-track slope estimate is computed for each region. This estimation is more reliable when the region size is large. It can be used directly with a weighting to balance its contribution to the overall cost. Combing all three cost terms, we can define the cost function as follows:

$$J_n(a) = c_{n,1} \cdot \left(\frac{\Delta_{h,n}(a)}{\sigma_{DEM}}\right)^2 + c_{n,2} \cdot \left(1 - \rho_n^2(a)\right) + c_{n,3} \cdot s_n^2 \qquad (3.5.3.3.4)$$

where $c_{n,1}$, $c_{n,2}$, and $c_{n,3}$ are the weights to balance the three terms, σ_{DEM} is the height uncertainty of the reference DEM used for the phase ambiguity determination, s_n is the cross-track slope of region n. (If the slope information is turned off for this purpose, c_3 will be set to zero). The selection of the three weights $c_{n,1}$, $c_{n,2}$, and $c_{n,3}$ is currently based on a few hundred simulated interferograms. Specifically, $c_{n,1}$, $c_{n,2}$ are constants and 0.25 and 1.0, respectively. $c_{n,3}$ is a weighted roundness of the shape of the region considered. The minimum cost will correspond to the minimum residual height, maximum correlation with the prior water mask, and the flattest topography if the third term is included.

Since there are many regions in an image, the ideal way to find the correct ambiguity for each region is to find the set of ambiguities for each of the regions to minimize the total cost:

$$\hat{\vec{a}} = \underset{\vec{a} \in A}{min} \sum_{n=1}^{N_t} \left\{ c_{n,1} \cdot \left(\frac{\sigma_n(a)}{\sigma_{DEM}}\right)^2 + c_{n,2} \cdot \left(1 - \rho_n^2(a)\right) + c_{n,3} \cdot s_n^2 \right\} \qquad (3.5.3.3.5)$$

Where N_t is the total number of regions of the image considered, n is the region index, $\vec{a}$ is the ambiguity vector with N_t elements, A is the set of ambiguities to search over, and a is one element of $\vec{a}$ for one region out of the N_t regions. This is a complicated optimization problem. The simplest algorithm to find the optimal solution is to do exhaustive search. Any other algorithms are sub-optimal. Unfortunately, the required computation for this exhaustive search is too overwhelming to be realistic for the hundreds or thousands of regions typically encountered in an interferogram.

An alternative sub-optimal approach is to assign the ambiguity of each region sequentially. In that case, the Eq. (3.5.3.3.5) still applies but the total number of regions $N_t = 1$. Since bigger regions and regions far away from nadir have better separations between minimum cost and the second minimum cost and their minimum cost ambiguity estimations are more reliable, we select those regions with bigger size and in the far range first for ambiguity assignment. Basically, we rank the regions in a descending order based on the sum of the range bin number of each pixel of the region and do the ambiguity assignment once a region sequentially until either every region is assigned an ambiguity or some regions are marked as unwrappable.

3.5.3.4 Accuracy

The accuracy of the ambiguity assignment step relies on the accuracy of the reference DEM, the prior water mask, the water detection algorithm, as well as other characteristics of the measurement (e.g., interferometric coherence). Other factors like layover, the size and structure of the actual water bodies, and where the feature falls in the swath also can play a significant role in the accuracy of the ambiguity assignment algorithm.

Generally, the probability of selecting the wrong phase ambiguity is very low (a few percent of pixels) for SWOT-measurable water bodies (i.e., water bodies at least as large as the science requirements). Nevertheless, when an incorrect ambiguity is selected, the height and location errors are very large. The largest contributors that relate to a higher likelihood of selecting the wrong ambiguity are: 1) if there is water-water layover in a region, 2) if the region is in the near range part of the swath, closer to nadir [because of the more restrictive ambiguity height, larger pixels/less averaging for the same size feature, and lower SNR], or 3) the reference DEM is not sufficiently accurate.

The pixel-wise accuracy of the final ambiguity solver, including the use of reference locations to select the ambiguity number for pixels that are not spatially unwrapped, is evaluated using the representative dataset simulations. Figure 32 shows the pixel-wise phase ambiguity error rate as a function of cross-track distance for different truth classes of the pixels. The dark water phase ambiguity rates are almost as low as the bright water rates—due in large part to the reference location conditioning assumptions for dark water. However, for mixed and land pixels there is a significant chance of selecting the wrong phase ambiguity.

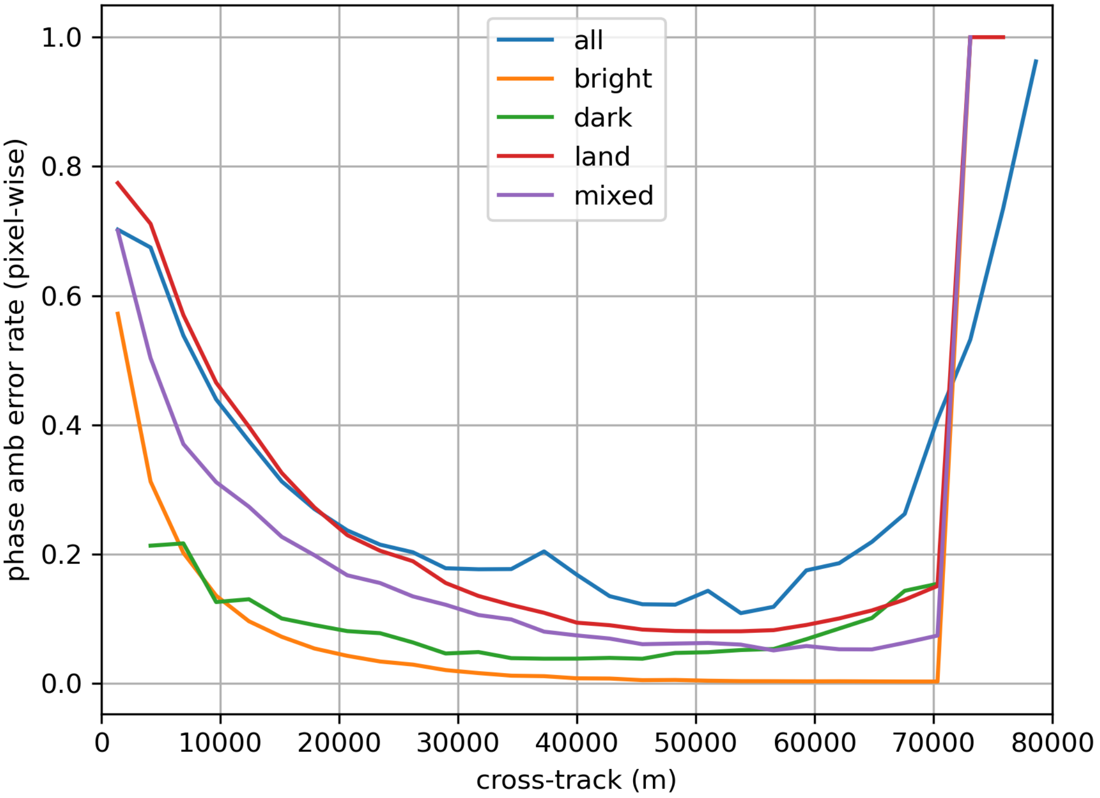


Figure 32. Pixel-wise phase ambiguity error rate as a function of cross-track distance (TODO: ignore after 60km...resolve issue).

3.5.4 Geolocate

This function reconstructs the height and location from the absolute phase, applying media corrections (e.g., wet troposphere delay), and calibration corrections (e.g., cross-over corrections, phase screen, phase biases and drifts, etc) that are obtained from the Auxiliary data input files [15] [16]. Recall that Figure 14 in Section 2.3.1 shows the geolocation latitude, longitude, and height images for a simulated river scene, while Figure 15 shows the phase noise standard deviation and estimated height standard deviation. Furthermore, Figure 16 shows the 3D scatterplots for a portion of the river.

3.5.4.1 Input Data

Description	Source
MedUnwRegions	SpatialUnwrap
Prior water mask	Static Auxiliary Data
RareUnflatIntf containing the rare reference locations	MakeRareIntf
XoverCal	XoverCal Intermediate Data
DynamicGeophys	Dynamic Auxiliary Data
StaticKaRinCal	Static Auxiliary Data

3.5.4.2 Output Data

Description
MedPixFull containing the geolocated pixels

3.5.4.3 Mathematical Statement

The geolocation and height reconstruction approach is not unique to SWOT. Height reconstruction from cross-track interferometry is well documented in published papers such as [9] [17]. Here, a high-level description of height reconstruction is provided as well as an explanation of the various corrections that must be applied to compute the absolute location of each pixel.

The location and height (e.g., the geolocation) of each slant-plane pixel is computed from the slant-range, the Doppler, and the absolute phase. In principle (neglecting the corrections described below to provide an intuitive understanding for now), when considering the geolocation of a single pixel, the set of all potential solutions with the same slant-range represent a sphere centered at the location of the reference antenna at the illumination time of the pixel. Similarly, all potential solutions with the same Doppler represent a cone centered on the velocity vector, with its apex at the center of the range sphere. Moreover, all potential solutions with the same absolute phase represent another cone centered along the interferometric baseline, with its apex at the center of the range sphere. The intersection of the range sphere, the Doppler cone, and the phase cone is the solution of the 3-D geolocation of that pixel (see Figure 33). This 3-D point is computed for each slant-plane pixel and then converted to latitude, longitude, and height above the ellipsoid. Note that the slant-range and Doppler "measurement" correspond to the pixel index in the slant-plane image, and each slant-plane pixel contains contributions from all points on the Earth's surface that intersect the circle of points on the range-sphere-Doppler-cone intersection (e.g., from layover). That is, the SLC processor effectively predefines the slant range and Doppler for each SLC image pixel and sorts (filters) the echo energy from the surface into the slant-range/Doppler bins of the SLC images. The slant-range and Doppler are therefore both known with relatively low uncertainty, whereas the interferometric phase has significantly higher uncertainty due primarily to noise variability and radar speckle. Note that the range sphere, Doppler cone, and interferometric phase cone may intersect in multiple places, but only one intersection is consistent with the gross knowledge of the viewing geometry (intersections above the radar or on the other side of the nadir track can be ignored).

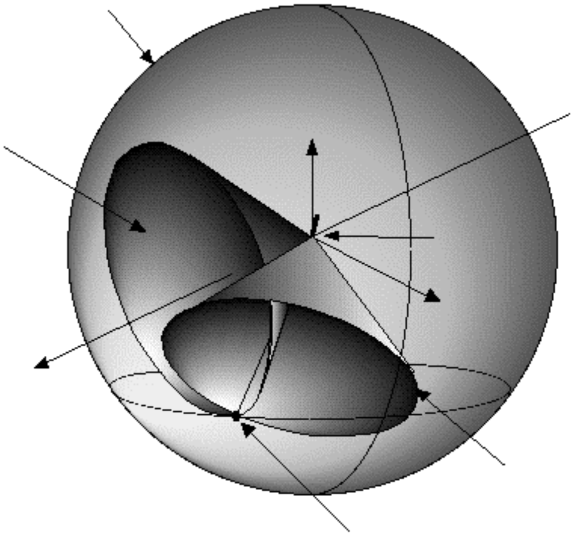


Figure 33 Range sphere, Doppler one, and phase cone used in geolocating each slant-plane pixel (TODO: annotate this figure with T, P, v, b etc.).

For each slant-plane pixel, the geolocation equation can be expressed mathematically as [9] [17]

$$\vec{T} = \vec{P} + \rho\left(\frac{\lambda f}{2v}\hat{v} + \frac{\frac{b^2}{2\rho}-\frac{\lambda\phi}{2\pi}\left(1+\frac{\phi}{4\pi\rho}\right)-\langle\vec{b},\hat{v}\rangle\frac{f}{2v}}{b\sqrt{1-\langle\vec{b},\hat{v}\rangle^2}}\frac{(\vec{v}\times\vec{b})\times v}{|(\vec{v}\times\vec{b})\times v|} \pm \sqrt{1-\langle\hat{l},\hat{v}\rangle^2-\langle\hat{l},\hat{n}\rangle^2}\,\frac{\vec{v}\times\vec{b}}{|\vec{v}\times\vec{b}|}\right) \quad (3.5.4.3.1)$$

where $\vec{T}$ is the scatterer position vector, $\vec{P}$ is the position vector of the reference antenna, $\vec{v}$ is the velocity vector, $\hat{v}$ is the unit vector in the velocity direction, $\vec{b}$ is the baseline vector (difference between the reference and secondary antennas), $\hat{l}$ is the unit look vector pointing from the reference antenna to the scatterer, ρ is the range from the scatterer to the scatterer, ϕ is the absolute (unwrapped and ambiguity assigned) interferometric phase between the two antenna channels, f is the Doppler frequency, and $\hat{n}$ is the normalized projection of the baseline vector into the plane normal to the velocity vector. Note that each of the terms in the geolocation expression generally differ for each slant-plane pixel, but we have suppressed this dependency in the notation for simplicity (e.g., $\rho(x)$ is expressed simply as ρ).

Note that the range, the Doppler, and the phase need to account for calibration as well as atmospheric effects before geolocation. The crossover and media delta corrections to the range and phase are applied here as well as a phase screen correction to the phase (which is a function of the absolute phase). Thus,

$$\rho(x) = r_1(x) + \Delta r(x) \tag{3.5.4.3.2}$$

$$\phi(x) = \phi_{abs}(x) + \Delta\phi(x) + \phi_{screen}\big(\phi_{abs}(x) + \Delta\phi(x)\big) \tag{3.5.4.3.3}$$

where $\phi_{screen}(\cdot)$ represents the function that maps the absolute phase to the phase screen correction, $r_1(x)$ is the range of the reference channel, $\phi_{abs}(x)$ is the absolute phase of the interferogram (after spatial phase unwrapping and phase ambiguity estimation), and $\Delta r(x)$ and $\Delta\phi(x)$ are the delta media and crossover corrections detailed in Section 3.2.4.1.

The geolocation sensitivity equation can be expressed mathematically as [TODO: cite]

$$\begin{aligned}\Delta\vec{T} = \Delta\vec{P} + \Delta\rho\hat{l} & \\ & + \frac{\rho}{\langle b, \hat{l}\times\hat{v}\rangle}\left[\left(-\frac{\lambda\Delta\phi}{2\pi b} - \frac{1}{b}\langle\hat{l}, \Delta\vec{b}\rangle\right)\hat{l}\times\hat{v} + \frac{1}{v}\langle\hat{l}, \Delta\vec{v}\rangle\hat{l}\times\hat{b}\right. \\ & \left. - \left(\hat{v}\times\hat{b} - \langle\hat{l}, \hat{v}\times\hat{b}\rangle\hat{l}\right)\frac{\Delta\lambda}{\lambda}\right]\end{aligned} \tag{3.5.4.3.4}$$

Where the notation with Δ prepended to a variable (e.g., $\Delta\rho$) indicates the error or small deviation in that variable (e.g., ρ). The sensitivity of the geolocation to the various measurements and system parameters can be obtained from this general equation. The specific equations for each variable can be somewhat messy, and so we exclude this level of detail from this document (see references [TODO: cite] for more details). The processor computes the sensitivities of height with respect to range $\frac{\delta h}{\delta \rho}$, baseline $\frac{\delta h}{\delta b}$, roll-angle $\frac{\delta h}{\delta \theta_{roll}}$, as well as the 3D geolocation sensitivities with respect to phase $\frac{\delta h}{\delta \phi}$, $\frac{\delta lat}{\delta \phi}$, $\frac{\delta lon}{\delta \phi}$. The sensitivity to phase can be scaled by an estimate of the phase noise standard deviation to obtain estimates of the height and location error bars.

The phase noise standard deviation is estimated from the knowledge of the distribution of the medium phase. The phase noise std (σ_ϕ) for each pixel (x) is related to the measured coherence (γ) and the medium number of looks (L_{med}) and can be expressed as [9]

$$\sigma_\phi(x) = \sqrt{L_{med}(x)\,\frac{(1-\gamma(x)^2)}{2\,\gamma(x)^2}} \tag{3.5.4.3.5}$$

Note that the estimate of the phase noise standard deviation is clipped to one ambiguity (any values greater than 2π are set to 2π). The coherence is obtained from the medium interferogram $I_{med}(x)$ and the reference $P_{1,med}(x)$ and secondary $P_{2,med}(x)$ channel powers by

$$\gamma(x) = \frac{|I_{med}(x)|}{\sqrt{P_{1,med}(x)P_{2,med}(x)}} \quad (3.5.4.3.6)$$

Note that for the purposes of computing σ_{ϕ} the estimate coherence is clipped to be within the range between 0 and 1 (e.g., $\gamma(x)\epsilon(0,1]$). Note that the estimate of coherence is generally biased, but the magnitude of the bias diminishes as the number of looks increases. The number of looks in the medium interferogram generally is large enough that the coherence bias is expected to sufficiently small.

3.5.4.4 Accuracy

The height reconstruction equations are accurate to floating point precision and do not introduce additional algorithm errors to the heights, or locations (assuming that the absolute phase is correctly resolved). Nevertheless, the overall accuracy of the heights and locations are dependent on the accuracy of several pieces of auxiliary data (including the orbit and attitude, the cross-over and media corrections) as well as system noise, radiometric calibration, phase screen and scene brightness (phase coherence), as well as layover contamination. Most of the errors that are directly related to the errors in the aux data produce what looks like a systematic bias for a given L2_HR_PIXC tile (since they are typically long wave number effects spatially), but tend to average out over multiple pass observations. The system noise errors that are related to the phase noise fluctuate from pixel to pixel and vary throughout the swath. These noise uncertainties are predictable and are directly related to the phase noise variance scaled by the height/location sensitivities to phase. An estimate of the phase noise variability, all the sensitivities, and all the information needed to aggregate the height and location uncertainties over larger spatial extents are provided in the L2_HR_PIXC product.

The geolocation sensitivities to phase, attitude, and baseline are reported using the derivatives at the reference locations. These sensitivities can be computed numerically using the geolocation equations with small perturbations to the inputs, or by using analytical mathematical expressions. Both approaches have been implemented and cross-verified, although the baseline is to use the analytical expressions [9].

3.5.5 Accuracy of UnwGeoloc

The accuracy of the UnwGeoloc function is primarily dependent on the phase noise standard deviation, since the phase unwrapping errors (which can cause extremely large height and geolocation errors) occur relatively rarely. Note that the phase noise errors are not algorithm considered algorithm errors, but are fundamental to the measurement process (i.e., no algorithm can mitigate them), nevertheless the phase noise contribution is included in the characterization of the height accuracy in this section.

Figure 34 shows the height error for pixels detected as bright water, Figure 35 shows the height error for near shore pixels (both land and water pixels near the shore), Figure 36 shows the height error for detected dark water pixels, and Figure 37 shows the height error for land pixels as a function of the cross-track distance over the entire representative dataset simulations. Note that the pixel-wise height errors for water typically fall within about half a meter from the truth, while dark water and land pixels have typical height errors on the order of several meters (because they are darker and thus have a lower coherence and phase noise standard deviation). The significantly larger height errors for regions with lower brightness (and thus lower coherence) is expected and is one of the main reasons for excluding most of the non-water pixels in the slant-plane images from the pixel cloud product.

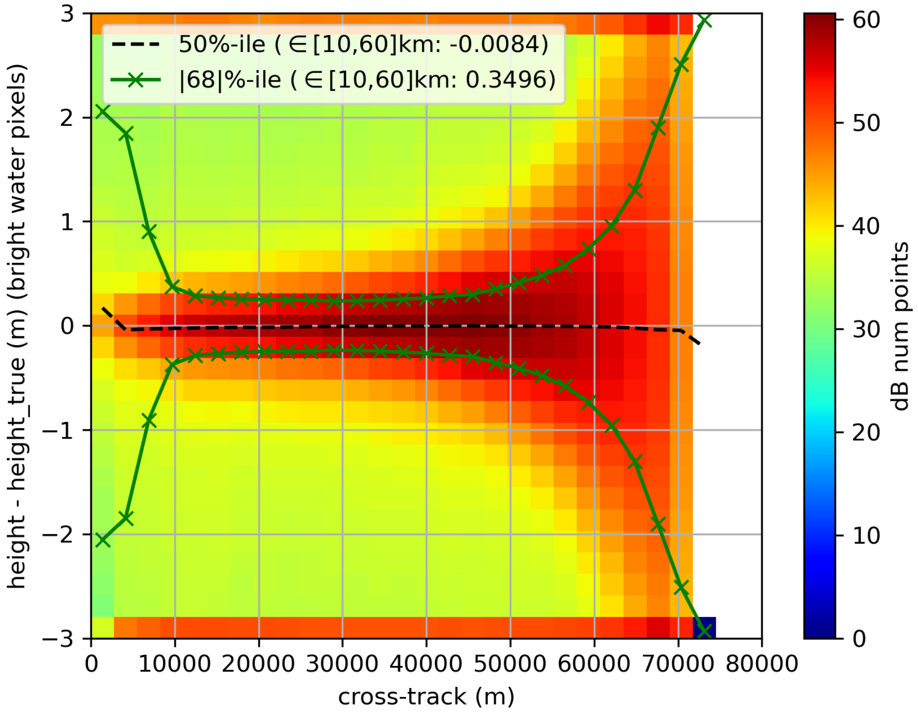


Figure 34. Height error for detected bright water as a function of cross-track distance.

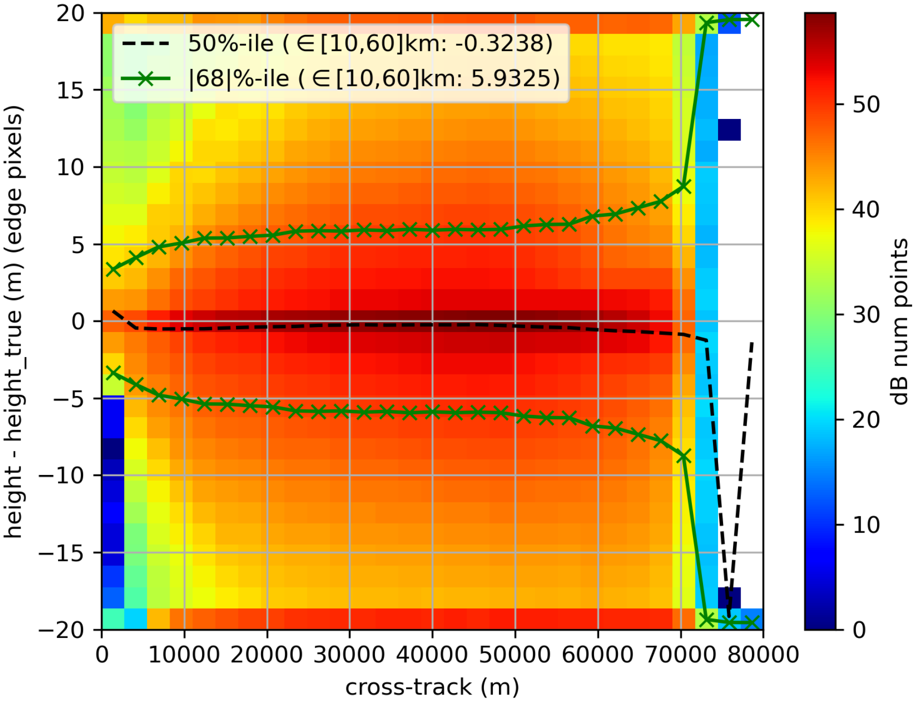


Figure 35. Height error for near shore pixels (both land and water near shorelines) as a function of cross-track distance.

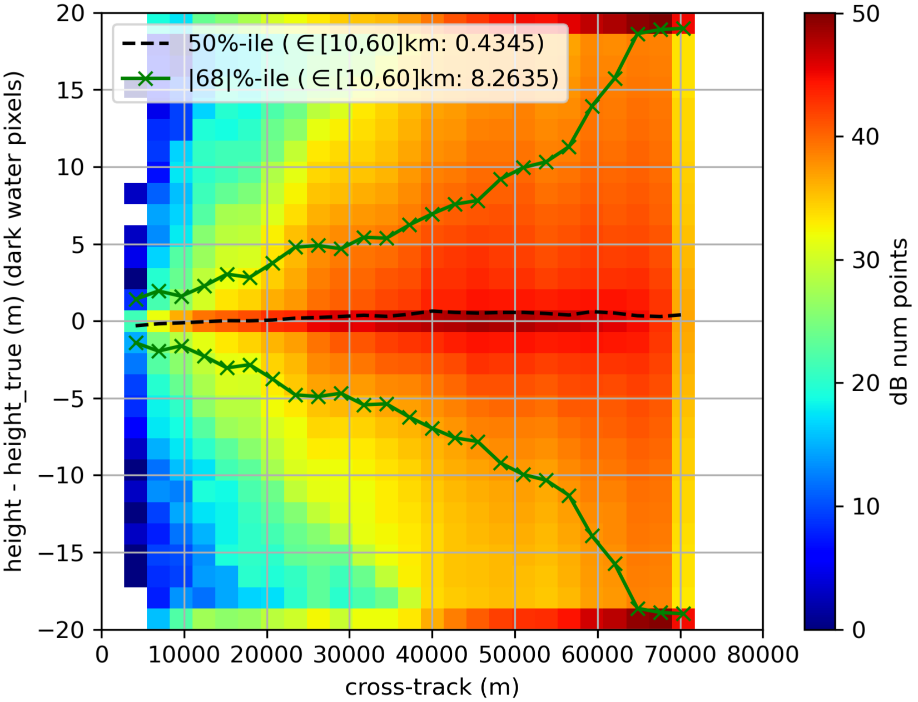


Figure 36. Height error for detected dark water as a function of cross-track distance.

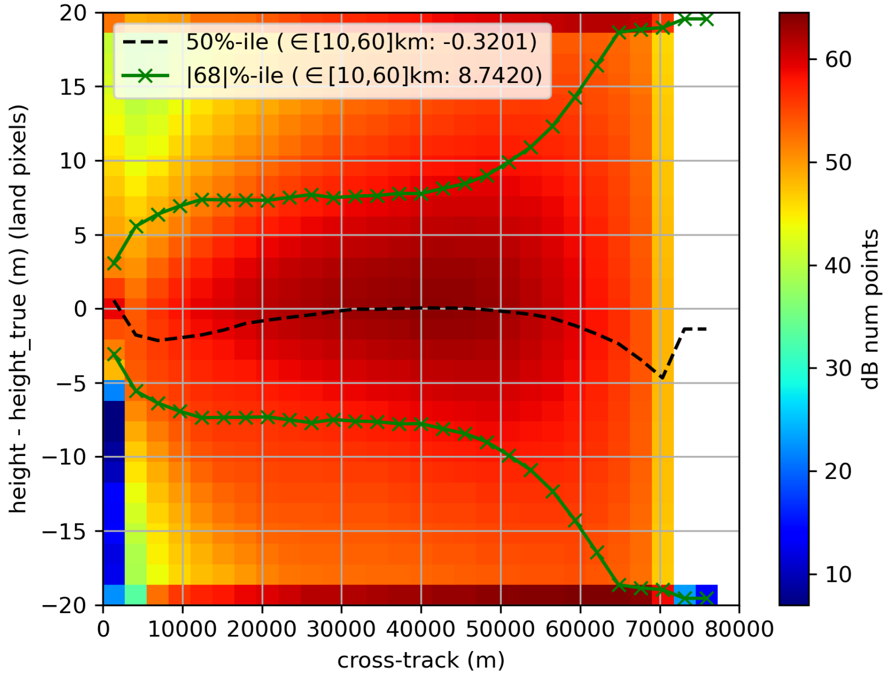


Figure 37. Height error for detected land as a function of cross-track distance.

3.6 CompGeophysCorr

3.6.1 Purpose

This function evaluates and interpolates various auxiliary data at each medium pixel location. There are two classes of variables computed here: 1) the effective height corrections that correspond to the geophysical corrections that are already applied before geolocation (e.g., the effective wet troposphere height correction), and 2) the various height references (e.g., the geoid, Earth tides etc.). Effective height corrections corresponding to the crossover calibration are also handled similarly here. Note that the particular corrections that are applied as range, phase, and/or interferometric baseline corrections during (or before) the geolocation function are reported in the product both as instrument range, phase, and baseline corrections as well as effective height corrections. This enables user to take out the effective height correction and put in some other correction as a delta height correction, without having to rerun the geolocation algorithms (e.g., using a different wet tropo model, or updated crossover calibration later on in the mission). These range, phase, and/or baseline corrections are interpolated at the reference locations during geolocation, whereas all other corrections are interpolated at the medium geolocated pixel locations.

3.6.2 Input Data

Description	Source
DynamicGeophys	Dynamic Auxiliary Data
StaticGeophys	Static Auxiliary Data

3.6.3 Output Data

Description
GeophysCorr

3.6.4 Mathematical Statement

The method evaluates and interpolates the aux data in space and time to the estimated pixel locations. Most of the corrections are interpolated to the geolocated positions in the MedPixFull data; however, some of the corrections are applied as range and/or phase corrections in the geolocation function and the corresponding effective height corrections are interpolated to the reference locations. Each data variable generally has a different implementation related to that specific quantity (e.g., the solid Earth tide is evaluated differently than the wet troposphere is interpolated). These algorithms have heritage from radar altimetry missions (e.g., Jason 1/2). The interpolation of the fields is done using the same algorithms applied in multiple processors and is detailed in the GECO software documentation [cite].

The following geophysical fields are evaluated from the input auxiliary data and interpolated to the geolocated positions of each slant-plane pixel. These corrections are not applied to the pixel height, but are reported in the product for the user's reference and to be applied in the downstream processors:

- Geoid

- Surface type
- Pole position
- Pole tide
- Solid Earth tide
- Finite Element Solutions (FES) load tide
- Goddard/Grenoble Ocean Tide (GOT) load tide

The following geophysical fields are evaluated from the input auxiliary data and interpolated to the reference locations for each slant-plane pixel. These corrections are applied during (or before) geolocation:

- Wet troposphere
- Dry troposphere
- Ionosphere total electron content
- Crossover calibration

The total range, phase, and interferometric baseline corrections applied during (or before) geolocation are reported in the pixel cloud product, as well as the corresponding effective height delta corrections related to these corrections for each individual term (e.g., effective wet troposphere height correction, effective dry topo height correction, and crossover height correction). The media delay height corrections are interpolated to slant-plane, and the crossover height correction is then computed as the sum of contributions from the timing height error term, phase height error term, and baseline height error term.

$$h_{xover} = h_{\ timing} + h_{phase} + h_{baseline} \tag{3.6.4.1}$$

The timing height error term is simply obtained as the height error term from the left or right swath. Note that this term is applied directly during processing as a range error, and the difference between height error, timing error, and range error are neglected (the difference is typically less than one millimeter).

$$h_{timing}(x) = \begin{cases} t_{r,xover}(x) & if\ swath\ is\ right \\ t_{l,xover}(x) & if\ swath\ is\ left \end{cases} \tag{3.6.4.2}$$

The phase height error term is computed from the slope terms as

$$h_{phase}(x) = \begin{cases} (s_{1,xover}(x) + s_{2,xover}(x)) * 10^{-6} \cdot D_{ct}(x) & if\ swath\ is\ right \\ (s_{1,xover}(x) - s_{2,xover}(x)) * 10^{-6} \cdot D_{ct}(x) & if\ swath\ is\ left \end{cases} \tag{3.6.4.3}$$

where D_{ct} is the cross-track distance in meters.

The baseline height error term is computed from the quadratic height error due to baseline as

$$h_{baseline}(x) = \mathrm{b_{xover}}(x) \cdot D_{ct}^2(x) \tag{3.6.4.4}$$

where D_{ct} is the cross-track distance in meters.

3.6.5 Accuracy

The methods implemented in the geophysical corrections module are sensitive to the accuracy and spatial scales represented in the input data. The accuracy is also fundamentally limited by the accuracy of the auxiliary data and models used in the geophysical algorithms. The errors due to interpolation implementation are considered to be negligible compared to the errors in the models themselves.

The accuracy of the geophysical quantities is also sensitive to errors in the location at which the quantity is interpolated. Therefore, the accuracy of corrections applied to range and/or phase is sensitive to errors in the reference locations, while the accuracy of all other corrections is sensitive to the geolocation errors for each pixel. However, the input data being interpolated is generally smooth relative to the pixel cloud grid and the geolocation errors (because the knowledge of the input data at high spatial frequencies is very limited). Thus, the errors caused by interpolating to the noisy measured locations and reference locations are generally negligible.

3.7 MakePixC

3.7.1 Purpose

This function collects the various data output from the other functions, prunes out the pixels that are not near water (either detected or in an inclusion mask) including trimming the data to the pixel cloud tile extent (removing the SLC product overlap in the slant-plane image pixels), computes the σ_0 estimate, populates the quality flags, and produces the L2_HR _PIXC standard product.

3.7.2 Input Data

Description	Source
InclusionMask	Static Auxiliary Data
MedPixcFull	UnwGeoloc
GeophysCorr	CompGeophysCorr
RareUnflatIntf	MakeRareIntf
Tile boundary defintions	RefOrbitTrack

3.7.3 Output Data

Description
Describe the output variable(s) generated by this function.
L2_HR_PIXC, the standard pixel cloud product

3.7.4 Mathematical Statement

For each pixel the many variables that have already been produced by other functions are collected, and additional quantities are computed. The output data can be grouped into a rare group or layer (not geolocated, but represents the rare interferogram water detection and flagging quantities), and a medium group or layer (which is primarily the heights and locations with various quantities needed for computing uncertainties, as well as geophysical corrections and height references).

For the rare layer variables, the following are collected: the rare interferogram with the two channel powers, the radiometric calibration, and the number of looks.

For the medium variables, the following data are collected:

- The geophysical corrections (e.g., wet troposphere effective height correction) and height references (e.g., geoid) are collected from the GeophysCorr data.
- The medium heights, locations, phase unwrapping region identifier, phase noise standard deviation, and various sensitivities to height, phase, baseline, range, and roll are collected from the MedPixcFull.
- An estimate of σ_0.

The σ_0 estimate is computed from the rare interferogram by taking the linear mean of the σ_0 of the two channels after subtracting the noise and applying the radiometric calibration as follows

$$\sigma_0(x) = \frac{1}{2}\left(\sigma_{0,1}(x) + \sigma_{0,2}(x)\right) = \frac{1}{2}\left(\frac{P_1(x)-P_{N,1}(x)}{X_1(x)} + \frac{P_2(x)-P_{N,2}(x)}{X_2(x)}\right) \quad (3.7.4.1)$$

where P_k is the rare interferogram power, $P_{N,k}$ is the measured noise power, and $X_k(x)$ is the X-factor (radiometric calibration), each for the two interferometric channels $k \in [0,1]$.

The method for pruning the pixels is based on the prior inclusion mask, the detected water, and the tile boundary information. First, all pixels in the multivalued classification ClassMaps that are non-zero (including the land in a buffer around water) are added to the list of output pixels. Recall that the classification variable already includes all the pixels in the inclusion mask. Finally, any pixels that are in the list that are beyond the tile boundary are excluded.

An estimate of the uncertainty or dispersion of the σ_0 estimate is also provided in the product. This quantity is computed from knowledge of the distribution of the signal and noise power measurements as follows.

$$var(\sigma_0(x)) = \frac{1}{4}\left(var(\sigma_{0,1}(x)) + var(\sigma_{0,2}(x)) + 2cov(\,\sigma_{0,1}(x)\sigma_{0,2}(x))\right) \quad (3.7.4.6)$$

$$= \frac{1}{4}\left(var(\sigma_{0,1}(x)) + var(\sigma_{0,2}(x))\right) + \frac{\rho_{1,2}}{2}\sqrt{var(\sigma_{0,1}(x))\; var(\sigma_{0,2}(x))} \quad (3.7.4.7)$$

where $\rho_{1,2}$ is the correlation coefficient between the σ_0 estimates of the two channels. Note that the two measurements are highly correlated so we do not get much variance reduction by taking the average, so we assume that $\rho_{1,2} = 1$. Each measurement variance is computed as

$$var(\sigma_{0,k}(x)) = \frac{var(P_k(x)) + var(P_{N,k}(x))}{X_k^2(x)} = \frac{1}{X_k^2(x)}\left(\frac{P_k^2(x)}{L(x)} + \frac{P_{N,k}^2(x)}{L_N(x)}\right) \quad (3.7.4.8)$$

where L_N is the number of looks for the noise measurement, respectively. The uncertainty reported in the product is the square-root of the variance $\sigma_{0,unc}(x) = \sqrt{var(\sigma_0(x))}$.

3.7.4.1 Quality Flagging

The quality flags are also produced in this function, although many of the quantities used to produce the final flags are generated at various stages within the processor. There are several quality flags explicitly set in the pixel cloud product:

1) the TVP quality flag, which is simply copied verbatim from the SLC product TVP
2) a per-line quality flag, similar to the SLC quality flag, but at the rare-slant-plane posting
3) a per-pixel quality flag indicating the quality of the rare interferogram quantities
4) a per-pixel quality flag indicating the quality of the heights/geolocations
5) a per-pixel quality flag indicating the quality of the classification results
6) a per-pixel quality flag indicating the quality of the σ_0

Each of these flags have multiple bits that indicate suspect, degraded, or bad quality of the relevant pixel cloud data, each bit preserving information as to why it is suspect, or degraded or bad. Flag values are generally zero if the corresponding data value(s) is/are believed to be nominal or good. The flag values are nonzero if the data are believed to be suspect, or degraded, or bad, with higher value bits corresponding to increasingly poor quality. Details on the bit values of each flag are expressed in the product description document [cite].

The per-line pixel cloud quality flag can be used to distinguish whether pixels in an interferogram row are excluded from the product because the interferogram was invalid (e.g., HR data not collected) or because valid data were collected by KaRIn, but no water was detected or otherwise thought to be present. This flag (q_{line}) is created by multilooking the SLC quality flag (q_{slc}) to the rare posting. The various bits are handled by independently multilooking a binary mask version of each bit ($M_{mask,b}$). For each bit, the corresponding rare-level flag bit is set for any rare line where the multilooked flag is nonzero. This can be expressed as

$$M_{mask,b}(a_{raw}) = q_{slc}(a_{raw}) \;\&\&\; 2^b \qquad (3.7.4.1.1)$$

$$M_{mask,b}(a_{rare}) = \begin{cases} 1 \; if \; \sum_{a_{raw}} M_{mask,b}(a_{raw})\, k_{ml}(a_{raw}, a_{rare}) \\ 0 \; otherwise \end{cases} \qquad (3.7.4.1.2)$$

$$q_{line}(a_{rare}) = \sum_b \, 2^b \, M_{mask,b}(a_{rare}) \qquad (3.7.4.1.3)$$

where a_{raw} is the 1D SLC/raw grid azimuth index, a_{rare} is the 1D rare interferogram azimuth index, and $k_{ml}(a_{raw}, a_{rare})$ is the 1D rare azimuth multilooking and downsampling kernel (i.e., shifted rect widows of length 7). There are per-line quality bits corresponding to suspect and bad quality of the tvp, suspect or bad quality due to spacecraft events, and whether there was a small or large along-track gap in the collection of KaRIn data.

The bits of the interferogram quality flag (on the rare posting) are initialized by taking the per-line quality flag and mapping to each pixel (e.g., sc_event_suspect, small_karin_gap etc). Then additional bits are set related to the quality of the rare power, and rare phase. There are two bits (suspect and bad) related to the interferogram power quality, as well as two other bits related to phase quality (also suspect and bad). For the power bits, sanity checks are applied to to each of the two interferometric channel power measurements in order to check for extreme values (linear unit values, not dB, outside [0, 10^50]) and invalid values (infinity, NaN, masked arrays etc). If all checks pass for both channels, no quality bits are set high. If exactly one of the channels power measurements do not pass the checks, the "rare_power_suspect bit" is set (e.g., one of the channel power measurements is still useful). If both power measurements do not pass the checks then the "rare_power_bad" bit is set. For the phase, both channel measurements must be valid (pass masked/inf/nan checks) otherwise the "rare_phase_bad" bit is set. The "rare phase_suspect" bit is set if the assumptions imposed when multilooking the phase (e.g., homogeneous statistics over the multilooking window) may be violated. A metric that quantifies dispersion of the phase (without considering variability of the magnitude), is computed by multilooking a complex unit vector with the measured phase (e.g., $e^{-j\phi}$). For multilooking windows with a large spread in the phase, the resulting magnitude of the multilooked quantity is attenuated (i.e., less than 1). For the "rare_phase_suspect" bit we apply a very small threshold (0.1) so that only regions with extreme phase dispersion end up getting flagged as suspect. Note that although one could also consider using a metric that weights the magnitude as well as the phase as a measure of phase dispersion (e.g., using the interferometric coherence like what is done in the geolocation quality flags later), the simple metric used here is sufficient for this purpose, and is less impacted by some of the subtleties related to the coherence.

The classification quality flag bits with the same "flag_meanings" attribute value as the interferogram quality flag (e.g., sc_event_suspect, small_karin_gap etc.) are inherited from the interferogram quality flag. Additional bit flags are then set depending on information relevant to the classification, water detection, and water fraction estimation. The "no_coherent_gain" bit is set for pixels where the coherent power is not amplified over the incoherent combination of the two channel powers. The "power_close_to_noise_floor" bit is set if the coherent power (in linear, not dB) is less than a scaled version (scale factor of 1.5) of the geometric mean of the thermal noise floor powers for the two interferometric channels. The "detected_water_but_no_prior_water" bit is set if the water is detected, but the prior probability map projected into the slant-plane (by the dark water flagging algorithm) is less than a threshold (0.2). The "detected_water_but_bright_land" bit is set if water is detected and the prior bright water mask projected into the slant-plane indicates bright land. The "water_false_detection_rate_suspect" flag is set by simply thresholding the false detection rate computed in the water detection algorithm (threshold is 0.05). The "coherent_power_suspect" bit is set if the interferogram qual "rare_power_suspect" bit is set or if the coherent_gain is less than or equal to 1. The "coherent_power_bad" bit is set if the interferogram qual "rare_power_bad" bit is set or if the coherent power fails sanity checks (i.e., within [0, 10^50], and not masked/inf/nan).

Like the classification quality flag, the geolocation quality flag sets the bits from the interferogram quality flag with the same "flag_meanings" then additional bits are set that pertain specifically to the geolocation quality. There are also flags related to the various geophysical correction data (e.g., wet/dry troposphere corrections, ionosphere corrections and crossover calibration corrections) that indicate whether those aux datasets are themselves suspect or missing. There is also a bit for "medium_phase_suspect" and "medium_phase_bad" that correspond to the rare versions in the interferogram quality. The "layover significant" bit is set if the absolute value of the layover_impact quantity is greater than some threshold (e.g., 5m). The "no_geolocation_bad" flag is set if there is no geolocation from the KaRIn data (e.g., bad or flagged, data, or no phase unwrapping solution), and no geolocation could be found using the reference locations (e.g., from the viewing geometry and the reference DEM). The "geolocation_is_from_refloc" bit is set if there is no geolocation from the KaRIn data, but there is a reference location solution. The "phase_noise_suspect" bit is set if the pixel is detected as water and the estimated height uncertainty is greater than a threshold (e.g., 5 m). The "phase_unwrapping_suspect" bit is set based on a threshold on the ratio of ambiguity cost (J_n) for the first (a_1) and second best (a_2) phase ambiguity solution. The ratio can be expressed as

$$R_a = \frac{J_n(a_2)}{J_n(a_1)} \tag{3.7.4.1.4}$$

The bit is set for pixels with values of R_a less than 1.6.

The σ_0 quality flag inherits many of the bits from the interferogram quality flag, and then sets a few extra bits. The xfactor and noise suspect and bad bits are set similarly to the interferogram quality power suspect and bad bit (i.e., they test for invalid values and then set to bad if both channels are invalid, and suspect if only one channel is invalid). The "sig0_cor_atmos_suspect" and "sig0_cor_atmos_bad" bits are set similarly to the geophysical correction bits in the geolocation qual flag (i.e., the suspect bit is set if the atmospheric correction data is suspect and the bad bit is set if the correction data are missing or invalid). The "sig0_uncert_suspect" bit is set if the sig0 uncertainty ($\sigma_{0,unc}(x)$, in linear, not dB) expressed above is greater than some threshold (10.0). or if the normalized sig0 uncertainty ($\sigma_{0,unc}(x)/|\sigma_0(x)|$) is greater than some threshold (10000.0 so that it is effectively unapplied, but can be updated with a parameter change, as opposed to a code change).

3.7.5 Accuracy

The majority of this function is simply collecting pieces from other functions in the processor and there is no additional error introduced for these quantities. The σ_0, which is computed here, is sensitive to the knowledge of the noise power and the radiometric calibration. Generally, this σ_0 estimate is expected not to be more accurate than a few dB. Furthermore, there is significant speckle at the medium level of smoothing, so the σ_0 will also have significant dispersion (which is captured by the uncertainty estimate $\sigma_{0,unc}(x)$).

Figure 38 shows the σ_0 error as a function of incidence angle for all the scenes in the representative dataset simulations. The σ_0 bias is generally less than 1 dB and the typical (1-sigma) errors are expected to be within about 3 dB for most of the swath for this simulation. Note that the σ_0 error in these simulations are primarily impacted by radiometric calibration errors (e.g., antenna pattern pointing alignment as well as absolute offset calibration), assuming simplistic modeling of many of the phenomena and calibration terms. Real SWOT data may exhibit more complex characteristics, but the errors are still expected be generally consistent with these results.

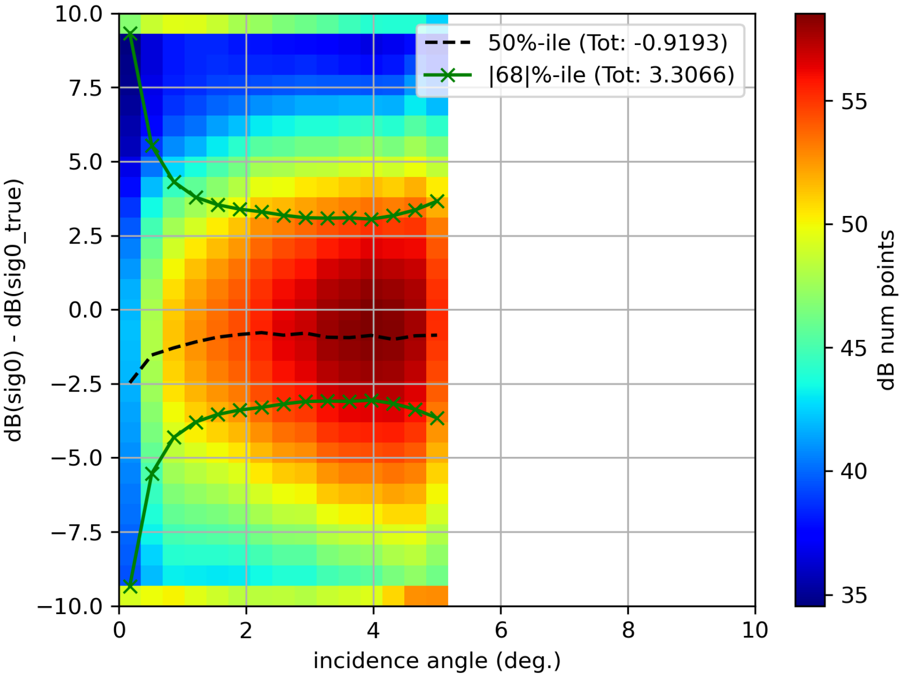


Figure 38. Normalized radar cross-section (σ_0) error as a function of incidence angle.

4 Accuracy of L2_HR_PIXC Processor

Note that there are no formal science requirements of performance at the pixel level, only once aggregated to river or lake features. Nevertheless, it is important to understand how pixel-wise errors propagate to rivers and lakes of various shapes and sizes, and in different parts of the swath. Here a simplified model is developed to propagate the pixel-level errors to the level of rivers and lakes. The purpose here is not to evaluate performance with respect to the science requirements. That is be done in the respective river and lake performance documents as those results also depend on trade-offs and errors introduced by the river and lake algorithms. Rather the point here is to provide the context for users to gain the intuition needed to interpret performance results as a function of 1) water feature size, 2) where it is in the swath, and 3) how much dark water exists in the feature and 4) how much of the water feature constitutes shoreline pixels. River and lake height and area, as well as river slope considerations are presented.

4.1 Height

At the pixel level, one of the largest errors contributing to the height errors is the noisiness or variability of the estimated height. This variability is significantly reduced when aggregating the pixel heights over larger spatial extents. There are many approaches that could be taken to aggregate the heights over a feature, but an optimal approach is to apply inverse variance weighting [cite uncertainty paper/memo]. This can be expressed as

$$\hat{h}_f = \sum_{x\in f} w(x)h(x) \tag{4.1.1}$$

where

$$w(x) = \frac{1}{\sigma_h^2(x)} \Big/ \sum_{x\in f} \frac{1}{\sigma_h^2(x)} \tag{4.1.2}$$

and $\sigma_h^2(x)$ is the variance of the height measurement $h(x)$ for each pixel x in the water feature f.

Assuming that the height measurements for each pixel are all statistically independent (which is not exactly true, but is a reasonable approximation if we account for the variance reduction as explained later) and defining the true feature height h_f as the non-weighted height the bias and 1-σ height error of the aggregated feature can be expressed as

$$B_{h_f} = E\{\hat{h}_f - h_f\} = \sum_{x\in f} w(x)\, B_h(x) + B_{w,h,f} \tag{4.1.3}$$

$$\sigma_{h_f} = \sqrt{E(\hat{h}_f - h_f)^2} = \sqrt{\sum_{x\in f} w^2(x)\, \sigma_h^2(x) + B_{h_f}^2 - \sum_{x\in f} w^2(x)\, B_h^2(x)} \tag{4.1.4}$$

where the pixel wise bias and 1- σ height error are

$$B_h(x) = E\{\hat{h}(x) - h(x)\} \tag{4.1.5}$$

$$\sigma_h(x) = \sqrt{E\left(\hat{h}(x) - h(x)\right)^2} \tag{4.1.6}$$

and the extra bias term $B_{w,h,f}$ that is due to the difference between the weighted and non-weighted average of the true height is

$$B_{w,h,f} = \sum_{x \in f} \left(w(x) - \frac{1}{N_p} \right) h(x) \qquad (4.1.7)$$

where $h(x)$ is the true pixel height, and N_p is the total number of pixels. Note that if the true height is constant over the feature, this extra bias term is identically zero. For relatively small features (small lakes and river nodes etc) setting this term to zero is a good approximation, but for larger features where there can be significant height differences within the feature (e.g., river reaches) this bias term can be large. For these larger features, we will assume that this bias term is accounted for in the estimation of the water feature height, so we will neglect it in the analysis presented in this section.

Now in a given water feature f there are generally multiple classes K of pixels (e.g., bright water, dark, water, edge water, low-coherence bright water etc), each with a distinct distribution (e.g., mean and variance). With this and assuming $B_{w,h,f} = 0$, the height bias and 1- σ height error of the aggregated feature can be expressed as

$$B_{h_f} = \sum_{k \in K} w_k \, N_k \, B_{h,k} \qquad (4.1.8)$$

$$\sigma_{h_f} = \sqrt{\sum_{k \in K} w_k^2 \, N_k \, \sigma_{h,k}^2 \; + B_{h_f}^2 - \sum_{k \in K} w_k^2 \, N_k \, B_{h,k}^2} \qquad (4.1.9)$$

where N_k is the number of pixels of class k in the feature, and $\sigma_{h,k}$ is the 1- σ height error of the pixels of klass k and

$$w_k = \frac{1}{\sigma_{h,k}^2} / \sum_{j \in K} \frac{N_j}{\sigma_{h,j}^2} \qquad (4.1.10)$$

Thus, the aggregated height over a water feature can be expressed as a random variable with mean and variance being functions of the mean and variance of the classes of pixels in the feature and the number of pixels of each class.

Now estimates of the errors as a function of cross-track distance (and averaged over 10km to 60km cross-track) can be computed for water features of various sizes in different parts of the swath using the pixel-wise statistics computed in the previous sections. Note that because the heights are estimated from the medium multilooked phase, the pixels are not independent. We account for this by scaling the pixel-level height standard deviation (or the 68th percentile, $P_{68,h,k}$) by the number of medium looks related to each feature and classification type (e.g., edge classes for simple water features are have 3 pixels in the multilooking window, while interior water classes can have up to 9 pixels if the water feature is wide enough). To be specific, we use

$$B_{h,k} = P_{50,h,k} \qquad (4.1.11)$$

$$\sigma_{h,k} = \sqrt{N_{med}} P_{68,h,k} \qquad (4.1.12)$$

where $P_{50,h,k}$ and $P_{68,h,k}$ are the pixel-wise 50th and 68th percentiles of the height error for class k, and N_{med} is the number of medium looks. Note that the number of edge vs bright vs dark pixels are approximated for each type of feature using very simple assumptions (e.g., square lakes and rectangular rivers oriented in the along-track direction, with 10% of pixels being dark water overall and 20% of edge pixels being dark water pixels).

Figure 39 shows the estimate of the bias of the height error over each kind of feature and Figure 40 shows the 1- σ height error for each particular feature. The kinks in the bias and the 1- σ height error in the lower cross-track distances are due to the fact that there are no interior water pixels for those water features at those cross-track distances because of the narrowness of the features relative to the pixel size, and they thus exhibit the statistics of the edge pixels more than the statistics of the interior, or non-edge water pixels. Note that the larger water features (river reaches and the larger lakes) are expected to produce results in much of the swath less than the 10cm level, while the smaller water features have much more uncertainty in the height estimates.

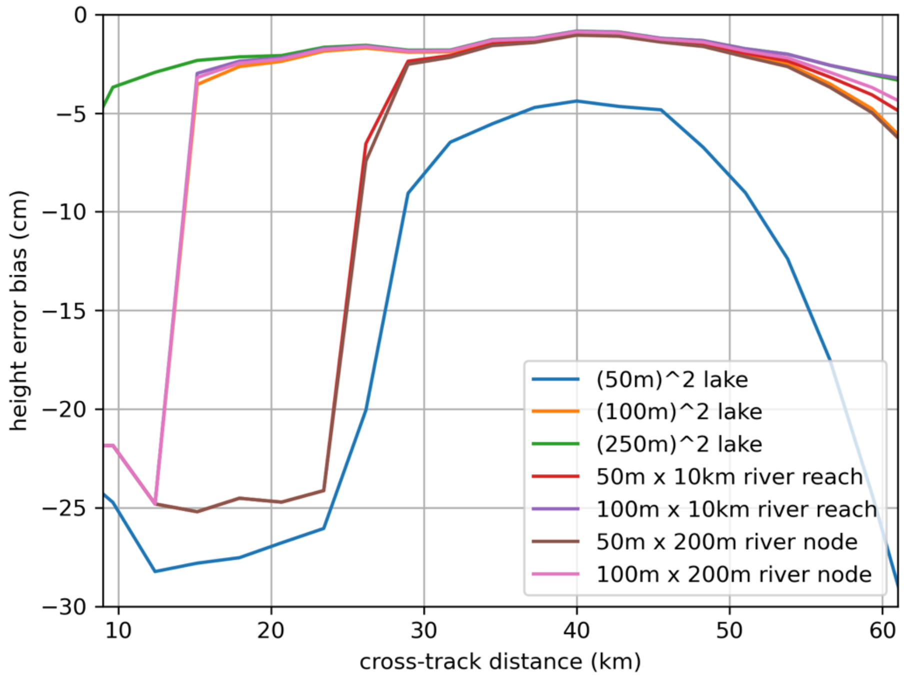


Figure 39. Projected water feature height bias as a function of cross-track distance.

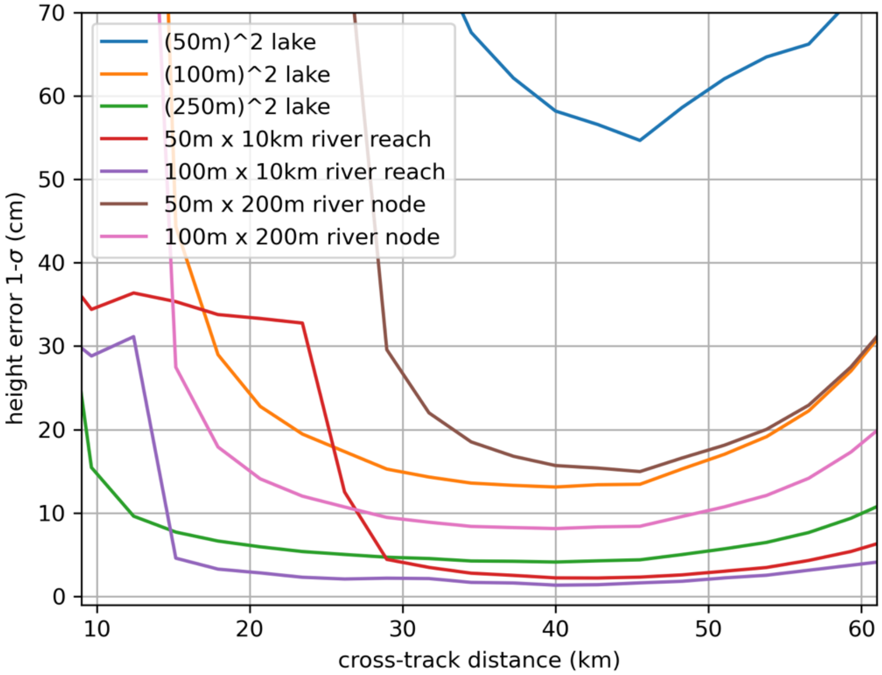


Figure 40. Projected water feature 1- σ height error as a function of cross-track distance.

4.2 Slope

To evaluate the river slope, two simple metrics are computed, one from taking the slope over a single reach using the heights of the end nodes (first to last node over 10km reaches), and one from taking the slope using the reach-level heights from consecutive reaches. Note that neither of these approaches are what are actually done for slope estimation in the river processor, but they provide two simple bounds (or rather reference points) to propagate the pixel-level errors to slope estimates under simple assumptions. Both of these two slope computations can be expressed as the height difference over 10km (10000m)

$$\hat{S}_{1,2} = \ (\hat{h}_2 - \hat{h}_1)/10000 \qquad (4.2.1)$$

where $h_2$is the upstream reference height (i.e., ether the upstream reach-level height, or the river node at the upstream side of the reach) and $h_1$is the downstream reference height (i.e., either the downstream reach-level height, or the node height of the downstream-most node in the reach).

The bias of the slope estimate can be expressed as

$$B_{s_f} = E\{\hat{S}_{1,2} - S_{1,2}\} = \frac{E\{\hat{h}_2 - h_2\} - E\{\hat{h}_1 - h_1\}}{10000} = \frac{B_{h_1} - B_{h_2}}{10000} \qquad (4.2.2)$$

and the 1- σ slope error of the slope estimate can be expressed as

$$\sigma_{s_f} = \sqrt{E(\hat{S}_{1,2} - S_{1,2})^2} = \frac{\sqrt{\sigma_{h_2}^2 + \sigma_{h_1}^2 - 2B_{h_2}B_{h_1}}}{10000} \tag{4.2.2}$$

When considering the standard deviation for rivers where the reference heights h_1 and h_2 occur in approximately the same cross-track location in the swath, and converting to units of cm/km, this can be simplified to

$$\sigma_{s_f} = \frac{\sqrt{2\left(\sigma_{h_f}^2 - B_{h_f}^2\right)}}{10000} \quad \frac{100cm/m}{1m/1000km} = \sqrt{\sigma_{h_f}^2 - B_{h_f}^2}10\sqrt{2}\ cm/km \tag{4.2.2}$$

where σ_{h_f} is the height 1- σ error (in meters) of either of the reference heights (since they are assumed to be the same), and $B_{h_f}$is the height bias of either reference height. Note that these expressions are simplified and only account for the measurement noise term. For slope errors there are other affects that are significant, including terms such as the residual roll bias and media effects that are often considered height biases, but actually vary spatially over larger extents (e.g., 10km reaches).

Figure 41 shows the slope 1- σ error for the same river features in the height plots above. The plot suggests that even at the node level for wider rivers, the slope can be estimated to within a few cm/km. However, this analysis only accounts for the measurement variability and does not account for some of the height errors (e.g., residual roll bias, and media delays) that exhibit low wavenumber characteristics (sometimes referred to as systematic errors) that can significantly affect the slope estimates (and often dominate the slope errors).

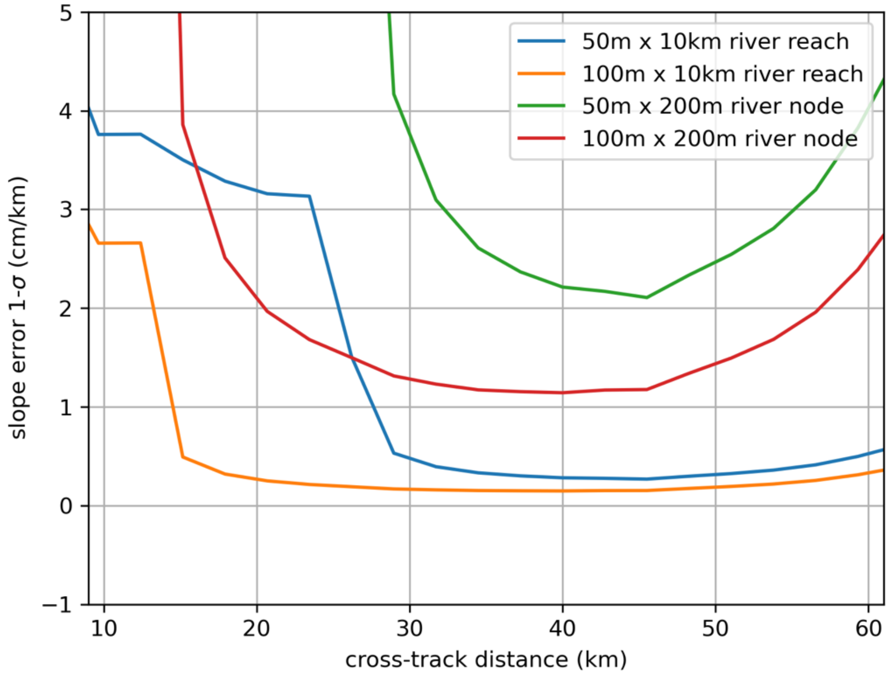


Figure 41. Projected water feature 1- σ slope error as a function of cross-track distance.

4.3 Area

Aggregating the area of the water pixels seems simple at first glance; however, special consideration needs to be applied to handle the effect of azimuth PTR smearing of the water edges. As discussed above, the pixel cloud product provides a noisy water fraction estimate of to scale the pixel area in order to handle this effect. Note that because of the large dispersion of the water fraction estimate, a less noisy estimate of the pixel area is obtained by simply aggregating the pixel area for pixels that are fully water (e.g., have a true water fraction close to 1); therefore, the approach that the downstream algorithms apply is to simply aggregate the pixel areas for water pixels not near the shorelines, while applying the water fraction estimate to the pixel area for both the water and land pixels near the shorelines. This is explained in more detail in the reference paper on uncertainties [cite it]. In the uncertainty paper, the area uncertainties over a feature (both a bias and variability term) can be expressed as a function of pixel-wise quantities that exist in the pixel cloud product. Applying similar assumptions as with the height, and neglecting measurement-to-water-feature assignment errors and pixel area errors, the area bias and 1- σ error can be expressed as

$$B_{A_f} = E\{\hat{A}_f - A_f\} = A_f \left[N_{de}\, B_\alpha + \textstyle\sum_{k \notin de} N_k \frac{P_{fa,k} - P_{md,k}}{2} \right] \tag{4.3.1}$$

$$\sigma_{A_f} = \sqrt{E\{\hat{A}_f - A_f)^2} = \sqrt{A_f^2 \left[N_{me}\, \sigma_{samp}^2 + N_{de}\sigma_\alpha^2 + \textstyle\sum_{k \notin de} N_k \frac{P_{fa,k} + P_{md,k}}{2} \right] + B_{A_f}^2 - B_{A_f,2}} \tag{4.3.2}$$

where

$$B_{A_f,2} = A_f^2 N_p \left[\frac{N_{de}}{N_p} B_\alpha + \textstyle\sum_{k \notin de} \frac{N_k}{N_p} \frac{P_{fa,k} - P_{md,k}}{2} \right]^2 \tag{4.3.1}$$

where $N_{de}$is the number of detected edge/shoreline pixels (both land and water edges), $B_\alpha = E\{\hat{\alpha} - \alpha\}$ is the bias in the water fraction estimates for the detected edge pixels, $k \notin de$ are all the non-edge water classes, $P_{fa,k}$ and $P_{md,k}$ are the flase alarm and missed detection rates for detected water class k, σ_{samp}^2 is the sampling error variance that is due to the finite pixel size (e.q., 1/12), N_{me} is the number of missed detected edges (e.g., $0.1 * N_{de}$), N_k is the number of pixels of detected water class k and A_f is the true area for the water feature.

In most analyses the area percent error is the metric of interest. The bias and the 1- σ error can be normalized by the true area and scaled by 100 to produce area percent error bias and 1- σ estimates from B_{A_f}, and σ_{A_f}. Figure 42 shows the area percent error bias for various features as a function of cross-track, and Figure 43 shows the corresponding 1- σ error. These plots suggest that the area percent errors over most features (e.g., except very small lakes) are expected to be less than the 15 % for much of the swath (although there is more trouble in the near-range). Note however, that this analysis does not account for all the components of the area error (e.g., pixel-to-water-feature assignment errors in the river and lake algorithms), and so the resulting performance for rivers and lakes through the actual processors can be somewhat worse.

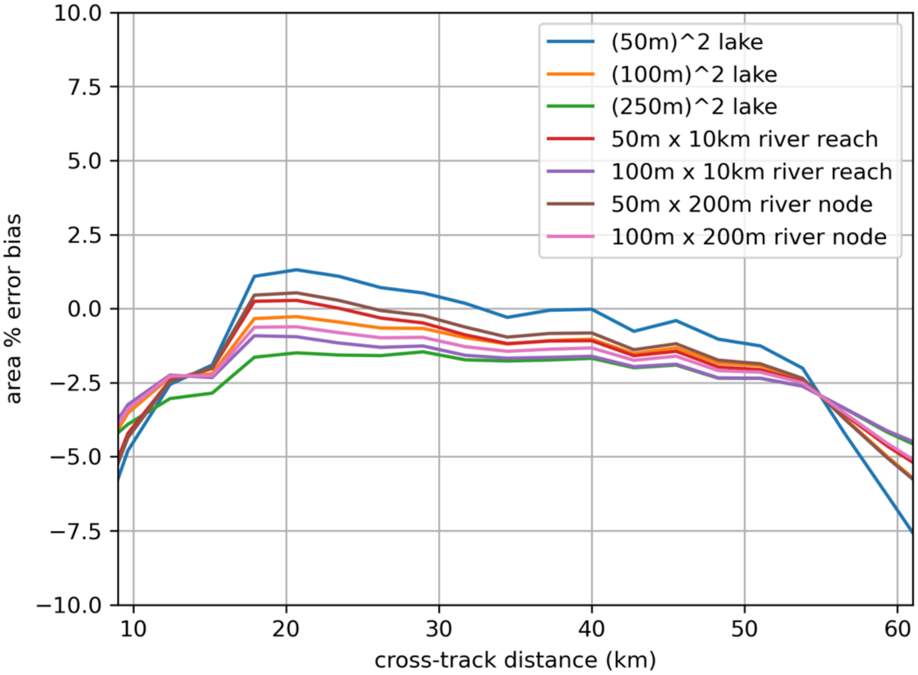


Figure 42. Projected water feature area % error bias as a function of cross-track distance.

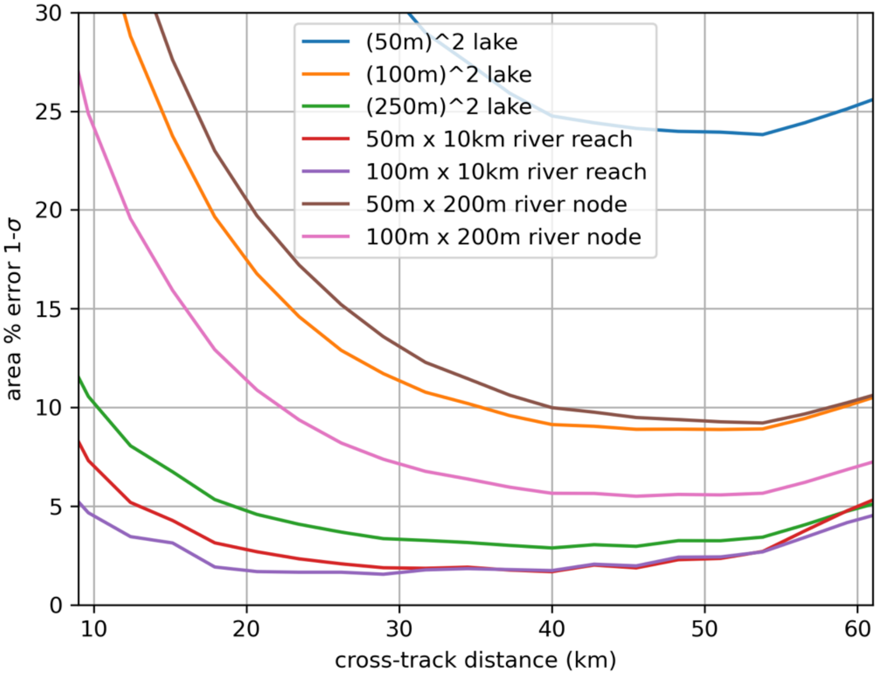


Figure 43. Projected water feature 1- σ area % error as a function of cross-track distance.

5 References

[1] JPL D-56411, "SWOT Level 2 KaRIn High Rate Water Mask Pixel Cloud Product," Jet Propulsion Laboratory Internal Document, Pasadena, 2023.

[2] JPL D-56410, "SWOT Level 1B KaRIn High Rate Single Look Complex Data Product," Jet Propulsion Laboratory Internal Document, 2023.

[3] JPL D-61923, "Surface Water and Ocean Topography Mission (SWOT) Project Science Requirements Document," Jet Propulsion Laboratory Internal Document, 2018.

[4] JPL D-56413, "SWOT Level 2 KaRIn High Rate River Single Pass Vector Product," Jet Propulsion Laboratory Internal Document, 2023.

[5] SWOT-TN-CDM-0673-CNES, Revision C (Draft) , "SWOT Product Description Document: Level 2 KaRIn High Rate Lake Single Pass Vector Product (L2_HR_LakeSP)," Centre National d'Etudes Spatiales, 2024.

[6] JPL D-56416, "SWOT Level 2 High Rate Raster Product," Jet Propulsion Laboratory Internal Document, 2022.

[7] E. Rodríguez, D. Esteban Fernandez, E. Peral, C. W. Chen, J.-W. De Bleser and B. Williams, "Wide-Swath Altimetry A Review," in *Satellite altimetry over oceans and land surfaces*, Boca Raton, Florida: Taylor & Francis, 2017, pp. 71-112.

[8] JPL D-79084, "SWOT Project Mission Performance and Error Budget," Jet Propulsion Laboratory Internal Document, 2017.

[9] P. A. Rosen, S. Hensley, I. R. Joughin, F. K. Li, S. N. Madsen, E. Rodríguez and R. M. Goldstein, "Synthetic Aperture Radar Interferometry," *Proceedings of the IEEE,* vol. 88, no. 3, pp. 333-382, 2000.

[10] SWOT-TN-CDM-0677-CNES, Revision C (Draft), "SWOT Product Description Document: Level 2 KaRIn High Rate Pixel Cloud Vector Product (L2_HR_PIXCVec)," Centre National d'Etudes Spatiales , 2024.

[11] JPL D-105503, "SWOT Level 1 High Rate Single Look Complex Algorithm Theoretical Basis Document," Jet Propulsion Laboratory Internal Document, 2023.

[12] S. Lobry, "Markovian models for SAR images: Application to water detection in SWOT satellite images and multi-temporal analysis of urban areas," TELECOM ParisTech, Paris, France, 2017.

[13] S. Lobry, L. Denis, R. Fjørtoft, F. Tupin, V. Poughon and B. Williams, "Water Detection in SWOT HR Images Based on Multiple Markov Random Fields," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* vol. 12, no. 11, pp. 4315-4326, 2019.

[14] J.-F. Pekel, A. Cottam, N. Gorelick and A. S. Belward, "High-resolution mapping of global surface water and its long-term changes," *Nature,* vol. 540, pp. 418-422, 2016.

[15] JPL D-75724, "SWOT Calibration / Validation Plan," Jet Propulsion Laboratory Internal Document, 2018.

[16] SWOT-NT-CDM-2150-CNES, "SWOT Algorithm Theoretical Basis Document: Crossover Calibration," Centre National d'Etudes Spatiales , 2022.

[17] S. Hensley, "A combined methodology for SAR interferometric and sterometric error modeling," in *Radar Conference*, 2009.

Appendix A. **Acronyms**

AD	Applicable Document
API	Application Interface
ATBD	Algorithm Theoretical Basis Document
CNES	Centre National d'Études Spatiales
JPL	Jet Propulsion Laboratory
NASA	National Aeronautics and Space Administration
OBP	On-Board Processor
PGE	Product Generation Executable
RD	Reference Document
SAS	Science Algorithm Software
SDS	Science Data System
SWOT	Surface Water and Ocean Topography
TBC	To Be Confirmed
TBD	To Be Determined
SLC	Single Look Complex image
SAR	Synthetic Aperture Radar
SNR	Signal-to-noise ratio

Appendix B. **Simulations**

This appendix presents the common simulation methodology that is used for assessments of the various algorithms presented in this document. Several scenes are simulated using the high-fidelity SWOT Hydrology SLC simulator. This simulator simulates the SLC images directly using a distributed target simulation scheme. The simulator uses the ideal SWOT orbit (not including orbit deviations), parameterizable SWOT system parameters (e.g., SNR, antenna pattern, attitude pointing etc), and a sophisticated modeling of the observation phenomenology.

The SLC simulations take as input: 1) the topography from a DEM, typically a high-resolution DEM (~5m) from lidar and 2) a surface classification map, called a land-type map, which includes a water mask as well as potentially multiple land classes (e.g., forest, grassland, urban etc). This DEM and land type map are made to be consistent with each other (e.g., steep land topography near water does not get mixed into the water areas, which should be mostly flat) and are placed on the same lat/lon grid. The DEM and land-type map over a particular "scene," or postage stamp area on the ground, is called the GDEM-DEM for that scene. Note that some simulations have been run with realistic variations of the water heights and extents by running multi-temporal hydrology models. This results in multiple GDEM-DEMs for a given scene (e.g., the Sacramento River at multiple stages). Given a particular GDEM-DEM, the SWOT orbits that overlap the GDEM-DEM are computed, and some (or all) of the passes can be used to simulate SLC products.

For a given pass, the σ_0 or brightness of each land type as a function of incidence angle is looked up from a table and the integrated complex interferometric returns from all the ground locations that map to a given slant-plane pixel are computed (weighted by the antenna gain pattern etc.). The tables of the brightness of each land type as a function of incidence angle are computed for each of the land classifications in the National Land Cover Dataset (NLCD) over the continental US using collocations with GPM data. Simplified constant σ_0 tables are also used in some cases (e.g., -5dB land and 10dB water). Note that the simulator also has the capability to produce spatial variability of the water brightness, including dark, specular water returns. This is accomplished by simulating a wind field with realistic statistics (k^-2 wavenumber spectrum and typical mean value), and then projecting this wind field through a Geophysical Model Function (GMF) that can map wind speed and incidence angle to ka-band σ_0. Note that dark water occurrence has also been parameterized as a function of distance to shore, which effect has been observed in AirSWOT data. This is implemented in the simulation by attenuating the wind field as a function distance to shore for a given truth water mask (while scaling to obtain the global expected dark water fraction). Figure 44 shows the dark water occurrence rate as a function of the distance to shore and incidence angle for the representative dataset used to evaluate the performance.

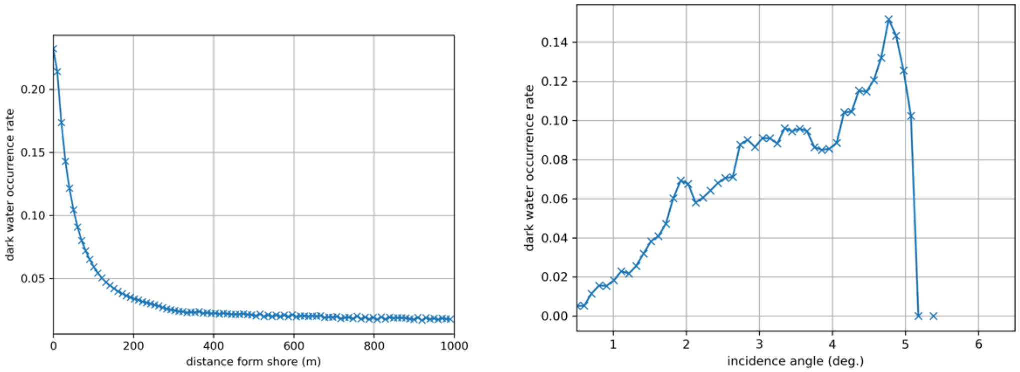


Figure 44. Dark water occurrence rate vs distance to shore and incidence angle for the simulation representative dataset.

Note that an azimuth and range point-target response are applied in simulating the interferogram. The returns from water-only pixels are handled differently than land pixels in order to simulate azimuth defocusing caused by surface movements during SAR processing (e.g., coherence-time smearing). To accomplish this, a water-only and a land-only interferogram are each produced, and a different azimuth point-target response is applied to the two interferograms, which are then added together to produce the “truth” interferogram.

Once a truth interferogram is obtained, thermal noise and radar speckle are simulated and two SLC images are produced that will result in an interferogram with the appropriate statistics. These SLC images are then formatted to the SLC product conventions, including flattening the phase with a global, relatively smooth or low-resolution DEM.

These SLC images are then processed by the pixel cloud processor using a reference DEM and reference water mask, which are generally different than those used in the truth GDEM-DEM. Note that there are several different pixel cloud processing options that can be run when using simulations in order simulate “ideal” processing of the various pixel cloud functions. For example, “ideal” water detection can be produced by using the “truth classification” field produced in the interferogram simulator. Furthermore, one can run the simulation with the truth DEM as the reference and simulate “ideal” geolocation with minimal phase unwrapping errors. Moreover, one can run with the “ideal” prior water mask by using the truth mask from the GDEM-DEM in the processing.

For the analysis in this report the so-called “representative dataset” has been used. This dataset is a collection of several scenes (32) simulated using lidar DEMs that have been curated by the hydrology science representatives of the SWOT Algorithm Development Team (ADT). There are two variants of this dataset: one that is representative for river hydrology and one that is representative for lake hydrology. The river scenes are obtained from the USGS as truth, using static water heights and masks. The water mask uses a combination of the National Hydrography Dataset (NHD), and SRTM water bodies and the heights over these water features have been filtered to result in smooth/flat heights over the water. All NLCD values are used as the land types for the continental US (though the NLCD open water class is remapped to deciduous forest), while for Alaska all land types are remapped to evergreen forest. For the reference DEM, SRTM v3 is used primarily, although Merit DEM is used for Alaska. 1-sigma worst case parameters are used for system parameters (e.g., SNR) and phenomenology (e.g., σ_0

scattering and coherence time smearing).

The lake hydrology representative scenes are generated in a similar manner but using some data from outside of the United States.