Surface Water and Ocean Topography Project

Algorithm Theoretical Basis Document
Long Name: Level 2 KaRIn high rate lake single pass science algorithm software: Level 2 Processing
Short Name: SAS_L2_HR_LakeSP: Level 2 Processing

Initial Release

Prepared by:

Claire Pottier CNES Algorithm Engineer	8/28/2023 Date	Cassie Stuurman JPL Algorithm Engineer	2023-08-29 Date

Approved by:

Roger Fjørtoft CNES Algorithm System Engineer	8/28/2023 Date	Curtis Chen JPL Algorithm System Engineer	2023-08-29 Date

Concurred by:

Jean Francois Cretaux CNES Hydrology Lead	8/29/2023 Date	Tamlin Pavelsky JPL Hydrology Lead	2023-08-29 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

July 26, 2023
SWOT-NT-CDM-1753-CNES

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

Contributing Authors

Name	Affiliation
Cécile Cazals	CS-Group
Manon Delhoume	CS-Group
Damien Desroches	CNES

Science Team Reviewers

Name	Affiliation
Herve Yesou	Service Régional de Traitement d'Image et de Télédétection (SERTIT), Université Louis Pasteur de Strasbourg Ecole Nationale Supérieure de Physique de Strasbourg (France)
Jida Wang	Department of Geography & Geographic Information Science, University of Illinois Urbana-Champaign (USA) Department of Geography and Geospatial Sciences, Kansas State University (USA)
Jean-François Crétaux	Laboratoire d'Études en Géophysique et Océanographie Spatiales (LEGOS) (France)

Change Log

VERSION	DATE	SECTIONS CHANGED	REASON FOR CHANGE
Initial Release	2023-07-26	ALL	Initial Release

Table of Contents

Table of Figures

Table of Tables

List of TBC Items

These items are to be completed when document is ready to enter configuration control.

Page	Section

List of TBD Items

These items are to be completed when document is ready to enter configuration control.

Page	Section
17	§2.3.1: The use of quality flags to discard pixels from lake processing remains TBD and will only be implemented post-launch.
29	§3.2.4: The use of these flags remains TBD and will be decided post-launch based on what we observe in real SWOT data.
42	§3.6.3.2.4.2: The current baseline is to only use open water pixels (i.e. classification 4) to compute *wse*, but this may evolve post-launch (TBD)

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 lake single pass vector (L2_HR_LakeSP) science data product [1], as well as Level 2 KaRIn high-rate pixel cloud vector attribute (L2_HR_PIXCVec) science data products [2]. These algorithms are applied in the SWOT Level 2 KaRIn high-rate lake tile vector (L2_HR_LakeTile) science data software (SAS), and in the L2_HR_LakeSP SAS.

For detailed information on the product file types, attribute definitions, and metadata fields, the reader is directed to the production description documents [1] [2] [3].

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_LakeTile and L2_HR_LakeSP SASs 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, as well as the functional flow of the primary functions.

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

Section 4 summarizes the overall accuracy of the L2_HR_LakeTile and L2_HR_LakeSP processors

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 describes the simulated data used for accuracy assessment.

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 [4]. The broad scientific goals can be summarized as follows:

Oceanography: To characterize the ocean mesoscale and submesoscale circulation determined from the ocean surface topography at spatial resolutions of 15 km (for 68% of the ocean).

Hydrology: To provide a global inventory of all terrestrial surface water bodies whose surface area exceeds $(250m)^2$ (goal: $(100m)^2$, threshold: $1km^2$) (lakes, reservoirs, wetlands) and rivers whose width exceeds 100m (goal: 50m, threshold: 170m). To measure the global storage change in terrestrial surface water bodies at sub-monthly, seasonal, and annual time scales. To estimate the global change in river discharge at sub-monthly, seasonal, and annual time scales.

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

The lowest level high-rate data product available to science users is the L1B_HR_SLC product [5], 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 precisely geolocated. The water mask pixel cloud (L2_HR_PIXC) product [6] is the first level of HR products that has several quantities that are directly useful for hydrology (primarily geolocated water heights). L2_HR_PIXC products are then used to generate standard vector products specific to rivers (L2_HR_RiverSP [7], L2_HR_RiverAvg [8]) and lakes (L2_HR_LakeSP [1], L2_HR_LakeAvg [9]), as well as raster products (L2_HR_Raster) [10], which are easier to use and likely better suited for most hydrology applications.

This document describes the algorithms that are used to generate the L2_HR_LakeTile intermediate products [3] and the L2_HR_LakeSP [1] and L2_HR_PIXCVec [2] standard products. First, an algorithm overview is provided, and thereafter a description of the functional flow of the algorithms of the L2_HR_LakeTile and L2_HR_LakeSP SASs. Section 3 contains more detailed descriptions of these algorithms. Section 4 presents the overall accuracy assessment.

2.2 Conceptual Processing Description

The handling of input pixel cloud [6] [11] tiles (~64×64 km^2) and tile boundaries represents a considerable part of the processing to generate an L2_HR_LakeSP product [1] covering an entire continent-pass (left and right half swaths across a continent [12]). Especially the fact that large lakes may cover more than two consecutive tiles makes it complex. However, it is done in a way that should give the same final result as if all the input tiles were simply concatenated and processed together (in practice this is not feasible because it would require too much memory).

In this overview section, we place ourselves in this imaginary case and let all the along-track tile handling logistics aside, and we also skip the specific handling of near-range and far-range tile boundaries, to provide a conceptual description of the principal algorithm steps to generate the three shapefiles of the L2_HR_LakeSP product [1], as well as L2_HR_PIXCVec products [2]:

1. Information from the pixel cloud vector attribute river (L2_HR_PIXCVecRiver) intermediate products [11] are used to eliminate certain L2_HR_PIXC pixels from further lake processing, i.e., pixels that the river processing has assigned to reaches defined in the Prior River Database (PRD) [13], except for reaches that are so-called "connected lakes" (lakes connected to the river network).
2. The remaining water pixels are segmented into features, i.e., distinct, connected groups of pixels in radar geometry [12] [14]. Note that there is not necessarily a one-to-one relationship between these features and actual lakes, nor lakes represented in the Prior Lake Database (PLD) [15] (there may be false water detection in the L2_HR_PIXC product, errors in the PLD…).
3. For each of these features
    a. Height-constrained (further regularized) pixel geolocations [16] are computed.
    b. A concave hull polygon is computed based on the height-constrained geolocations of the edge pixels.
    c. Attributes are computed (average water surface elevation, area…).
    d. The polygon of the feature is compared to the polygons of the PLD (illustrated as dashed polygons in Figure 1 (a)), and the feature is assigned to one or more PLD lakes in case of overlap.
4. Features assigned to one or more PLD lakes are gathered in the L2_HR_LakeSP_Obs shapefile, as shown in Figure 1 (b), and those that have not been assigned to any PLD lake constitute the L2_HR_LakeSP_Unobserved shapefile, illustrated in Figure 1 (d).
5. The assignment to PLD lakes is done at the pixel level, so when a detected feature (connected group of water pixels in radar geometry) is assigned to several PLD lakes, each of its pixels is only assigned to one PLD lake, based on distance. To populate the L2_HR_LakeSP_Prior shapefile, illustrated in Figure 1 (c), we consider each set of water pixels assigned to the same PLD lake (one or more connected regions) and compute:
    a. Its observed geometry (one or more polygons)
    b. Its attributes (overall average water surface elevation, overall area…)
    c. Water storage change w.r.t. a reference state, based on information in the PLD (storage change is only available in the L2_HR_LakeSP_Prior shapefile)
6. PLD lakes covered by the input data, but not observed, are added to the L2_HR_LakeSP_Prior shapefile with no geometry and only prior data attributes populated.
7. For each input L2_HR_PIXC and L2_HR_PIXCVecRiver product, the river and lake identifiers, height-constrained geolocations etc. of the pixels are written to the corresponding L2_HR_PIXCVec product.

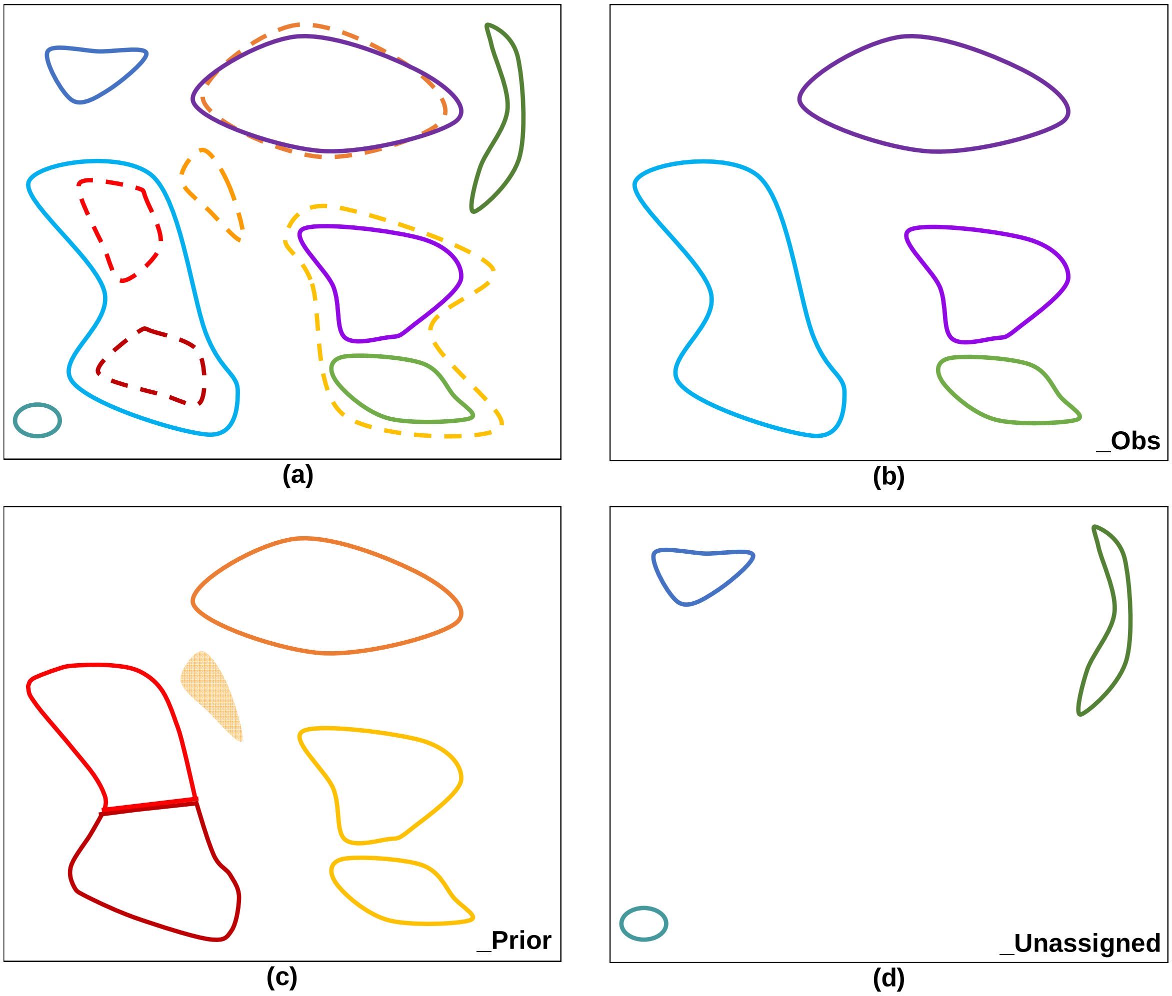


Figure 1. Illustration of how L2_HR_LakeSP [1] (and L2_HR_LakeTile [3]) products are organized in three shapefiles. (a) Example of observed features (solid polygons) and PLD lakes (dashed polygons) in an area. Different colors indicate different observation identifiers or PLD identifiers. (b) Polygons of the observation-oriented lake shapefile. (c) Polygons of the PLD-oriented lake shapefile. The unobserved PLD lake is an empty geometry with only prior attributes, here shown as a filled polygon. An observed lake intersecting two PLD lakes (red and dark red) is split into two polygons. Two observed lakes intersecting the same PLD lake (yellow) are grouped in a multipolygon. (d) Polygons of the observation-oriented unassigned features shapefile.

2.3 Functional Flow

Figure 2 summarizes the overall flow of the L2_HR_LakeTile and L2_HR_LakeSP processors. Each L2_HR_LakeTile intermediate product [3] is generated from the corresponding L2_HR_PIXC standard data product [6] and L2_HR_PIXCVecRiver intermediate product [11], as well as the LakeDatabase (or PLD) auxiliary data product [15]. Thereafter, an L2_HR_LakeSP product [1] is generated from all the L2_HR_LakeTile intermediate data products (left and right half swaths) of a continent-pass [12]. The L2_HR_LakeSP processor also generates one L2_HR_PIXCVec standard product [2] per input L2_HR_LakeTile product.

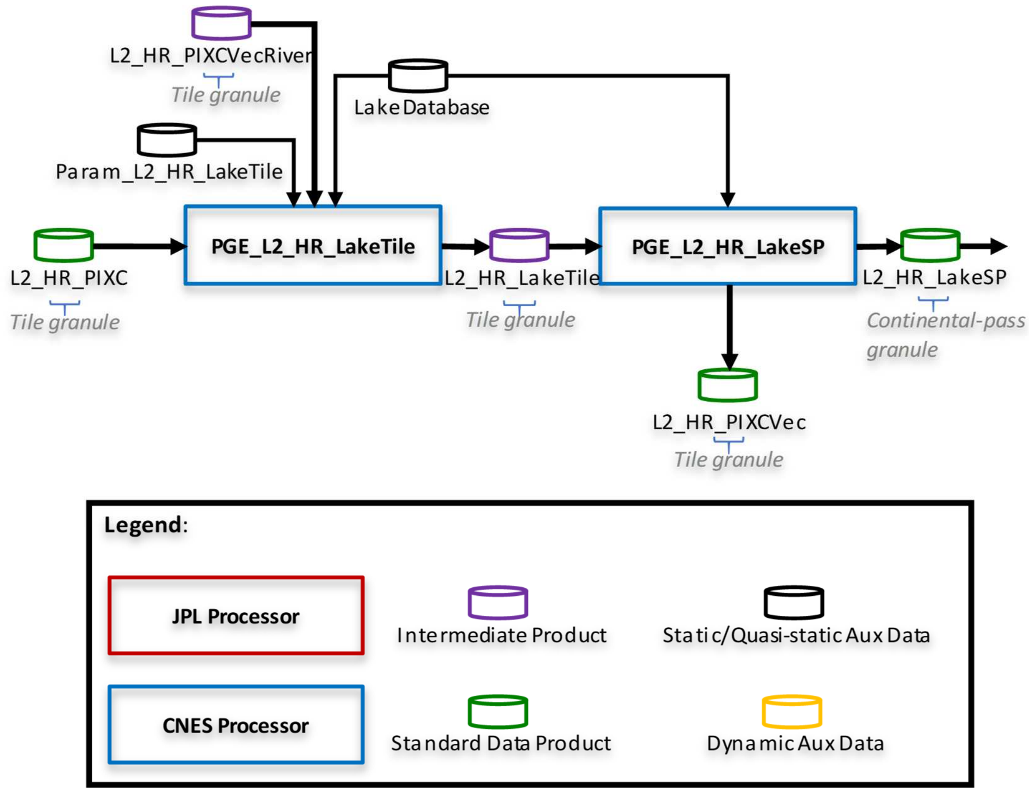


Figure 2. Overall flow diagram for the L2_HR_LakeTile and L2_HR_LakeSP processors.

2.3.1 L2_HR_LakeTile

Table 1 provides a high-level description of each of the processing functions that are used to generate the L2_HR_LakeTile intermediate product and Figure 3 illustrates the flow of these processing steps.

Table 1. High level description of the functions used to generate the L2_HR_LakeTile product.

Function Name	Description
`proc_pixc_vec.compute_pixc_to_reject`	Select pixels to reject from lake processing, i.e., those processed by river processing, except connected lakes
`proc_pixc.extract_from_pixc`	Select pixels to use in lake processing
`proc_pixc.compute_separate_regions`	Compute water mask and label separate water regions
`proc_pixc.localize_regions_wrt_tile`	Separate labels of detected water regions entirely inside the tile, from labels of those located at along-track edges of the tile
`proc_pixc.compute_edge_pixels`	Get pixels corresponding to water regions crossing the along-track edges of the tile
`proc_lake.compute_lake_features`	Compute lake features corresponding to the list of water regions not crossing the along-track edges of the tile

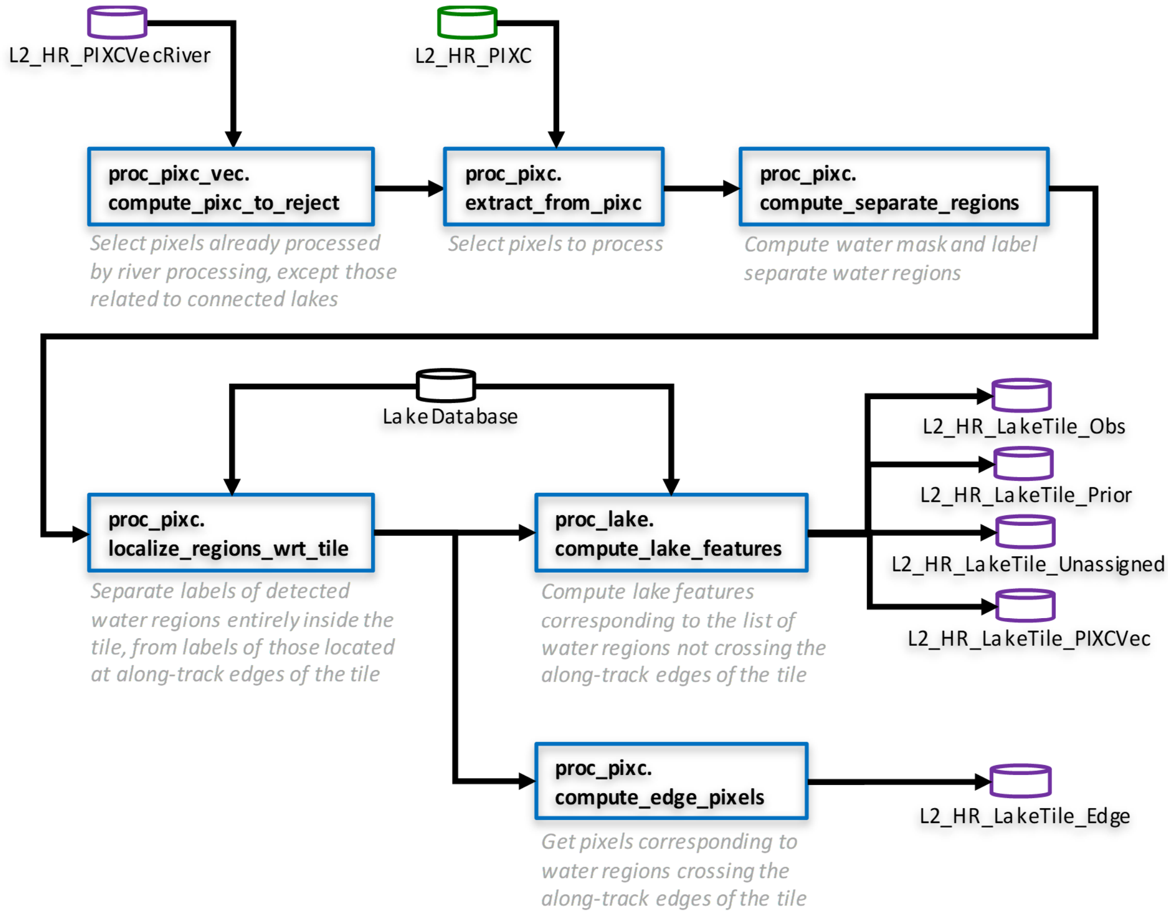


Figure 3. Flow diagram of the L2_HR_LakeTile processing steps (functions) used to generate the L2_HR_LakeTile product.

The L2_HR_LakeTile processing begins with the selection of the water pixels of L2_HR_PIXC product that it should handle. Based on information in the L2_HR_PIXCVecRiver product, water pixels already assigned to reaches by the river processing (except for reaches that are connected lakes) are discarded. The use of quality flags to discard pixels from lake processing remains TBD and will only be implemented post-launch.

Then, the remaining water pixels are gathered into separate features through segmentation in radar geometry [12] [14], where we have a regular grid and therefore a notion of connectivity (as opposed to the pixel cloud in ground geometry). Pixels belonging to features at along-track L2_HR_PIXC tile boundaries (i.e. at the edge between the current and the previous and/or next tile) are put apart and written to the L2_HR_LakeTile_Edge file for further handling in the L2_HR_LakeSP SAS. This processing organization is made necessary by the fact that large lakes can span several tiles along-track.

Afterwards, each remaining feature (entirely inside the tile granule in the along-track direction) is processed separately. They will populate the L2_HR_LakeTile_Obs and L2_HR_LakeTile_Unassigned shapefile layers, illustrated in Figure 1 (b) and (d), respectively.

- First, a height-constrained geolocation is computed. By averaging the pixel heights at the scale of the feature, and using it to constrain the geolocation, it becomes far less noisy. This updated geolocation is stored in the L2_HR_LakeTile_PIXCVec file.

- Then, concave hull polygons, delineating the outer boundary of the detected water feature, and inner boundaries in case of islands, are extracted from the water edge pixels, assigning height-constrained geolocations to the nodes.
- The attributes of the feature are computed from the L2_HR_PIXC variables.
- The next step is to establish the link between the observed feature and PLD lakes: if the polygon of an observed feature intersects one or more PLD polygons, it will go into the L2_HR_LakeTile_Obs shapefile; otherwise it will go into the L2_HR_LakeTile_Unassigned shapefile.

The identifier of the observed feature and the identifier of the associated PLD feature (if any), are stored for the corresponding pixels in the intermediate L2_HR_LakeTile_PIXCVec file. Although some detected features are linked to several PLD polygons, each pixel is linked to a single PLD lake: in case of two or more intersecting PLD polygons, a given pixel is assigned to the closest PLD lake (based on influence zones defined in the PLD, see Section 3.6.3.3 for more details). This enables the processor to populate the L2_HR_LakeTile_Prior shapefile layer, illustrated in Figure 1 (c).

- For each observed PLD lake:
    - One or more patches of pixels assigned to it are used to compute its geometry (a single polygon or a multi-polygon) and attributes.
    - Prior information from the PLD are used to populate additional attributes.
    - Water storage change w.r.t. a reference is computed.

2.3.2 L2_HR_LakeSP

Table 2 provides a high-level description of each of the processing functions that are used to generate the L2_HR_LakeSP product and Figure 4 illustrates the flow of these processing steps.

Table 2. High level description of the functions used to generate the L2_HR_LakeSP product.

Function Name	Description
`proc_pixc_sp.` `swath_global_relabeling`	Gather edge pixels in separate regions for all the tiles of the continent-pass for one swath
`proc_lake.` `compute_lake_features`	Compute lake features corresponding to the list of water regions crossing the along-track edges of the tiles of the continent-pass for one swath
`proc_pixc_vec_sp.` `update_pixcvec`	Update the L2_HR_LakeTile_PIXCVec variables of the pixels related to water regions crossing the along-track edges of the tiles of the continent-pass for one swath
`lake_db.` `init_prior_layer`	Initialize L2_HR_LakeSP_Prior layer with the attributes of the PLD lakes located over the continent-pass granule
`my_shp_file.` `merge_shp`	Gather all features of several shapefiles into a single shapefile
`locnes_product_shapefile.` `merge_duplicate_features`	Merge attributes of PLD lakes observed by both swaths into a single feature

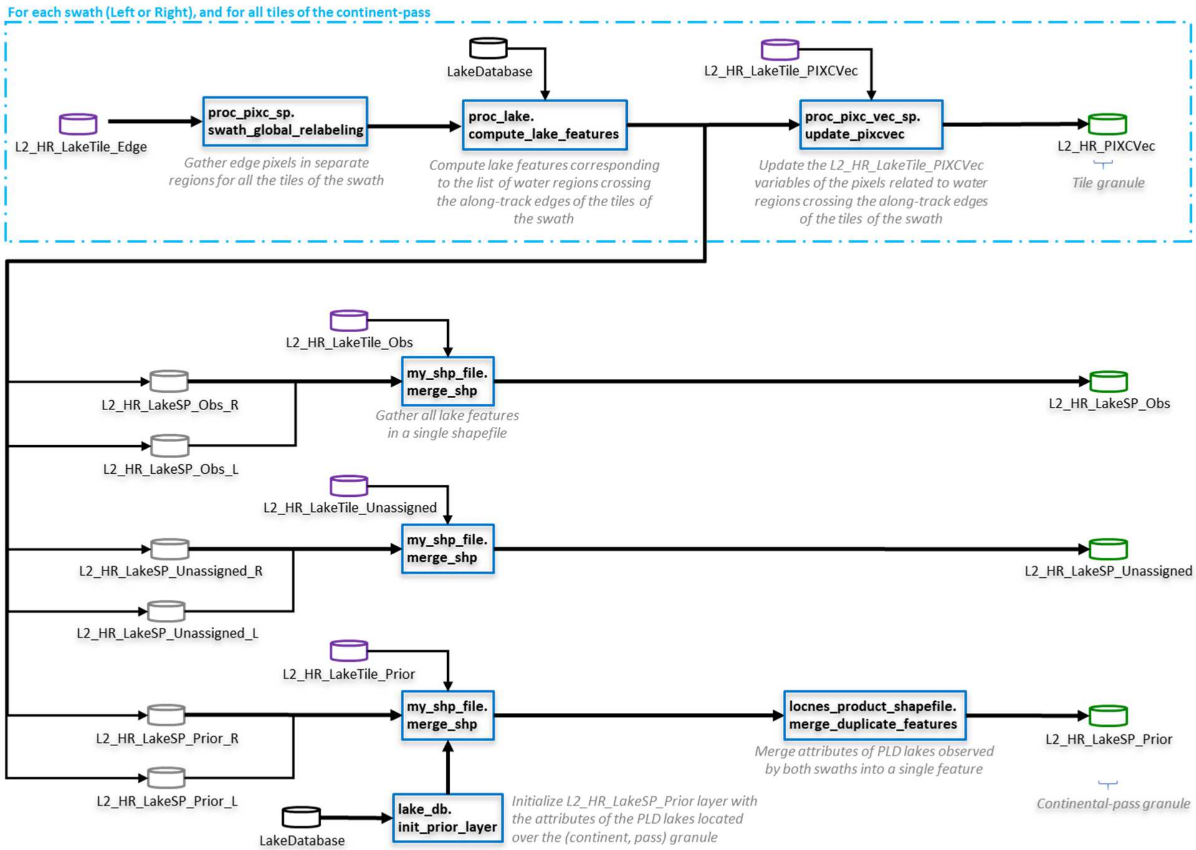


Figure 4. Flow diagram of the L2_HR_LakeSP processing steps (functions) used to generate the L2_HR_LakeSP and L2_HR_PIXCVec products.

The main steps of the L2_HR_LakeSP processing are:

- For each swath (Left or Right):
    - Process detected water features situated at the along-track boundaries between consecutive L2_HR_LakeTile products (i.e. corresponding to pixels present in the L2_HR_LakeTile_Edge files) into polygons and attributes.
    - Update each input L2_HR_LakeTile_PIXCVec file with information on the pixels related to these features to generate the corresponding L2_HR_PIXCVec product.
- Merge these edge lake polygons and attributes with those already present in the input L2_HR_LakeTile products into L2_HR_LakeSP_Obs|Prior|Unassigned shapefile layers. The L2_HR_LakeSP_Prior shapefile layer has been previously initialized with the attributes of the PLD lakes located within the continent-pass granule; therefore, PLD lakes covered by the input L2_HR_LakeTile tiles, but that have not been detected as water, are included in the L2_HR_LakeSP_Prior shapefile layer with no geometry and only prior data attributes populated.
- For the L2_HR_LakeSP_Prior shapefile layer, merge attributes of PLD lakes observed by both swaths into a single feature.

The L2_HR_LakeSP processing begins with the handling of the L2_HR_LakeTile_Edge files of the continent-pass, i.e., all features crossing along-track tile edges. The left and right half swaths are processed independently. First, we relabel these features for the entire continent-pass. Then, each feature is processed separately, in the same way as in the L2_HR_LakeTile processing detailed in the previous section. Temporary L2_HR_LakeSP_Obs and L2_HR_LakeSP_Unassigned shapefile layers (illustrated in Figure 1 (b) and (d), respectively) are created, and thereafter the L2_HR_LakeSP_Prior shapefile layer (illustrated in Figure 1 (c)) and the L2_HR_PIXCVec file.

After the processing of L2_HR_LakeTile_Edge files, the results of the two half swaths are merged. Then, they are gathered with the L2_HR_LakeTile files containing water features entirely inside the tile granule in the along-track direction. PLD lakes covered by both half swaths are merged into a single feature (for ex. two separate polygons become one multi-polygon). We obtain the final L2_HR_LakeSP product.

Likewise, one standard L2_HR_PIXCVec product is generated per input L2_HR_LakeTile product, by combining L2_HR_LakeTile_PIXCVec files with height-constrained geolocations, identifiers etc. obtained from the L2_HR_LakeTile_Edge files.

2.3.3 Illustrative Example

An example based on data generated with the SWOT Large Scale simulator [17] is presented here to illustrate the data as they propagate through the L2_HR_LakeTile and L2_HR_LakeSP processors. While all the tiles of an entire continent-pass are simulated, only a small extract of a tile, and the boundary with the previous tile, are shown. The simulation is based on a synthetic water mask (Figure 5 (a)) containing a river and several lakes, some of them identified in the PLD (Figure 5 (b)).

Figure 6 shows the input products of the L2_HR_LakeTile processor, which are the L2_HR_PIXC and L2_HR_PIXCVecRiver products. Figure 7 illustrates the main output:

- The L2_HR_LakeTile_Obs shapefile contains the observed features that intersect at least one PLD lake. An observed lake may intersect two or more PLD lakes, for example if the observed water stage is higher than the one reflected by the PLD.
- The L2_HR_LakeTile_Prior shapefile contains the PLD lakes covered by the tile, including those that are not observed (detected). Two or more PLD lakes may correspond to the same observed lake, which will then be split into several features, one per PLD lake. Inversely, two or more observed lakes may intersect the same PLD lake and be grouped in a single feature, with a single geometry (multi-polygon) and summed or averaged attributes (area, water surface elevation…).
- The L2_HR_LakeTile_Unassigned shapefile contains the observed features NOT linked to any PLD lake.

Recall that the L2_HR_LakeTile processor does not handle water bodies situated at the along-track edges of the tile. Instead, pixels at the along-track edges are stored in L2_HR_Lake_Tile_Edge files, which are further handled in the L2_HR_LakeSP processor.

Figure 8 illustrates the L2_HR_LakeTile_PIXCVec output file, which includes the additional information on river pixels contained in the L2_HR_PIXCVecRiver input product (Figure 5 (b)) as well as the equivalent information for the water pixels processed by the L2_HR_LakeTile processor.

Figure 9 (a) illustrates the L2_HR_LakeTile_Edge product, containing pixels of water features located at the along-track tile edges.

The edge pixels of all tiles of the continent-pass granule are then handled by the L2_HR_LakeSP processor, leading to temporary L2_HR_LakeSP_Obs_tmp, L2_HR_LakeSP_Prior_tmp and L2_HR_LakeSP_Unassigned_tmp shapefiles. Figure 9 (b) and Figure 10 illustrate the results over the studied area. Afterwards, these temporary files are combined with their corresponding L2_HR_LakeTile shapefiles (Figure 11).

Figure 12 illustrates the L2_HR_PIXCVec output file, which includes the additional information on pixels contained in the related L2_HR_LakeTile_PIXCVec file (reach and lake identifiers, height-constrained geolocations etc.) as well as the equivalent information for the water pixels at the along-track edges processed by the L2_HR_LakeSP processor.

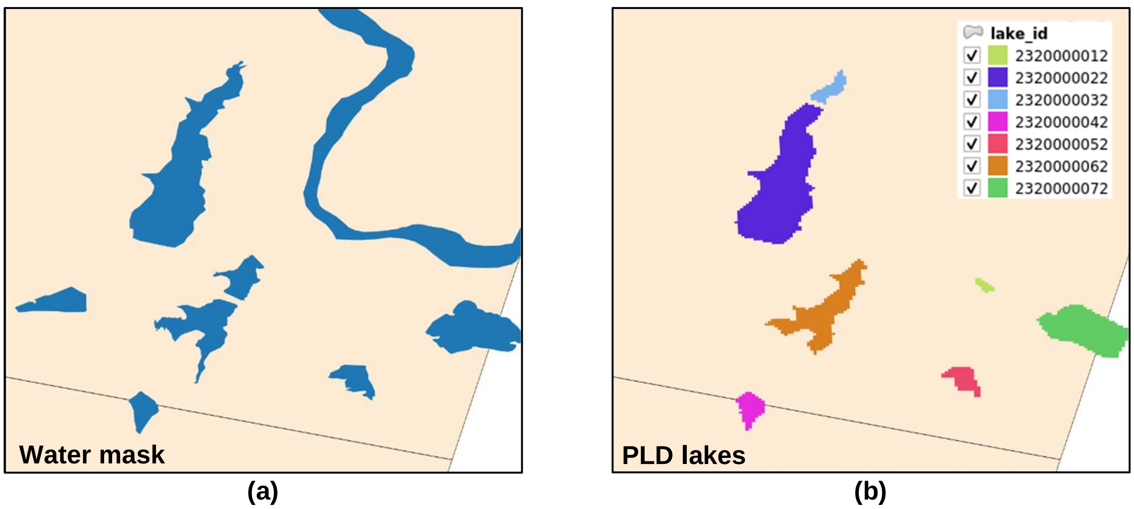


Figure 5. (a) Water mask of the synthetic scene (only a small extract is shown): the top-right water body corresponds to a river, the others to lakes, most of which are identified in the PLD. (b) PLD over the same area: the *lake_id* identifiers of the PLD lakes are shown in the legend. The rectangles in pale yellow correspond to L2_HR_PIXC tiles. We will mainly focus on the principal tile (228R), although a small part of the previous tile (227R) is also shown. The water feature corresponding to the green PLD lake crosses the far range edge of the tile, whereas the water feature corresponding to the pink PLD lake crosses the along-track edge of the tile.

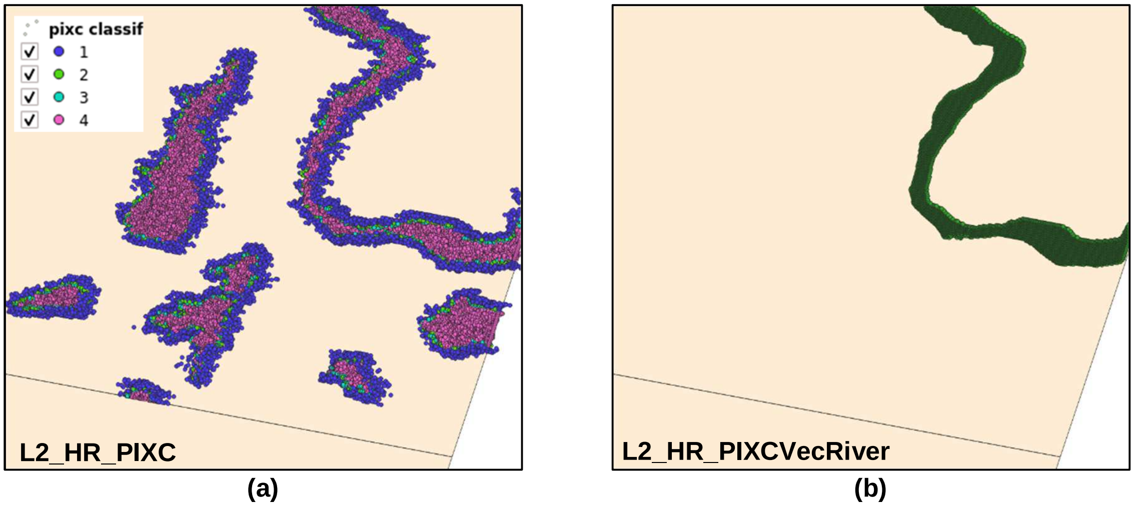


Figure 6. Input products of the L2_HR_LakeTile processor. (a) The L2_HR_PIXC product. The classification of the pixels is shown in the legend: 1=land; 2=land-near-water; 3=water-near-land; 4=open water (refer to Section 3.1.4 for a complete list of classes and information about which classes used in the lake processing). (b) The L2_HR_PIXCVecRiver product indicates the pixels that have already been processed by the river processor, and those assigned to river reaches (except for connected lakes) will be excluded from the lake processing.

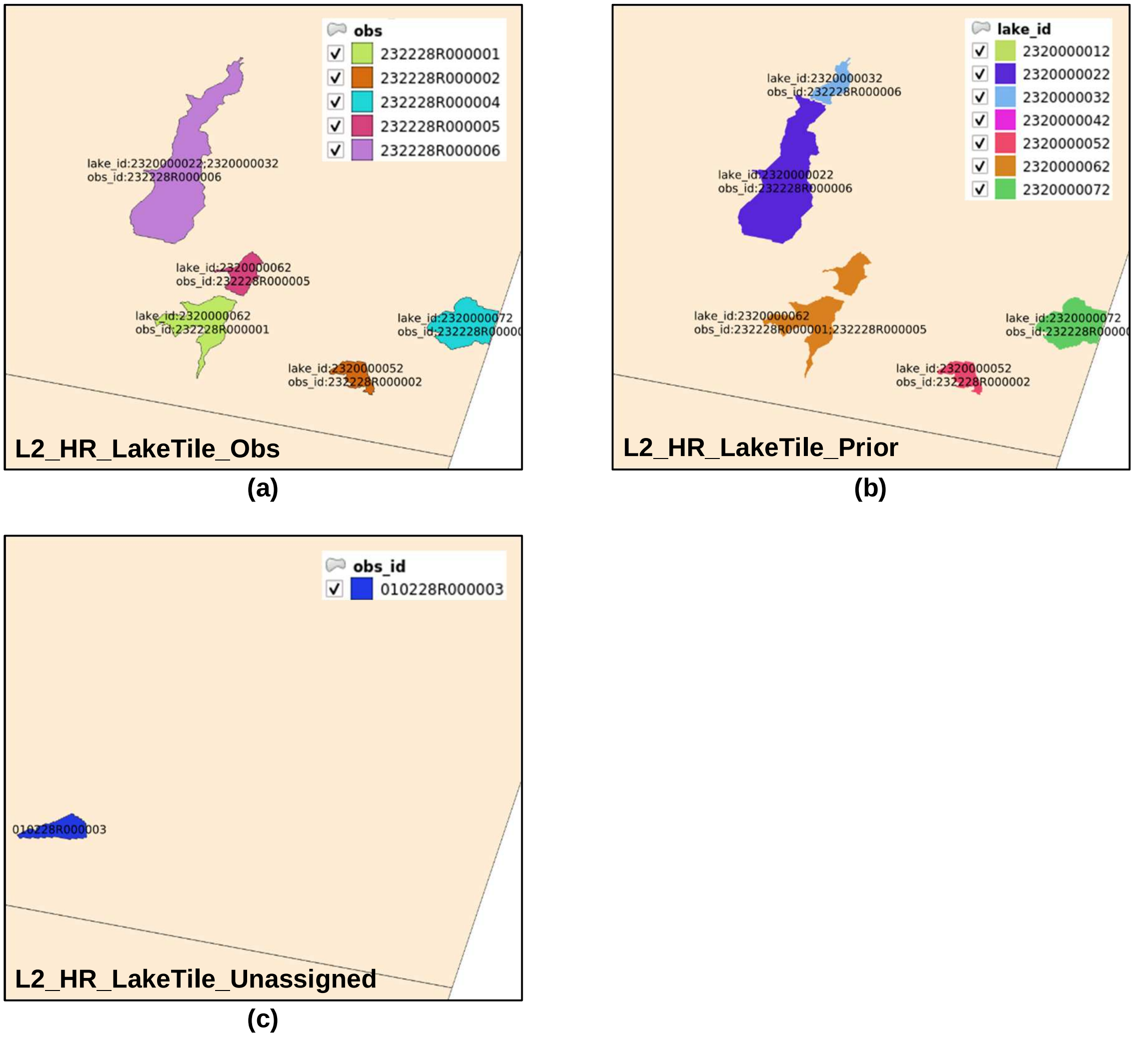


Figure 7. Output of the L2_HR_LakeTile processor (1/3). (a) The L2_HR_LakeTile_Obs shapefile contains the observed features linked to at least one PLD lake. The pink and green features are stored as two distinct features (with their own geometry and attributes) in the shapefile, even though they are related to the same PLD lake (lake_id=2320000062). (b) The L2_HR_LakeTile_Prior shapefile contains the observed PLD lakes covered by the tile. The purple observed feature is therefore split into two prior features in this shapefile, one for each intersecting PLD lake. Inversely, the pink and green observed features are stored as one single prior feature, with a multi-polygon geometry and aggregated attributes (area, water surface elevation…). PLD lake 2320000012 (in pale green in Figure 5 (b)) has not been observed. It is therefore not stored in the L2_HR_LakeTile_Prior shapefile (but will later be added by the L2_HR_LakeSP processor). PLD lake 2320000042 (in pink in Figure 5 (b)) is not processed by the L2_HR_LakeTile processor as the observed geometry attached to it crosses the along-track edge of the L2_HR_PIXC tile (but it will be processed later by the L2_HR_LakeSP processor). (c) The L2_HR_LakeTile_Unassigned shapefile contains the observed features not intersecting any PLD lake (and that have not been assigned to a river reach by the river processing).

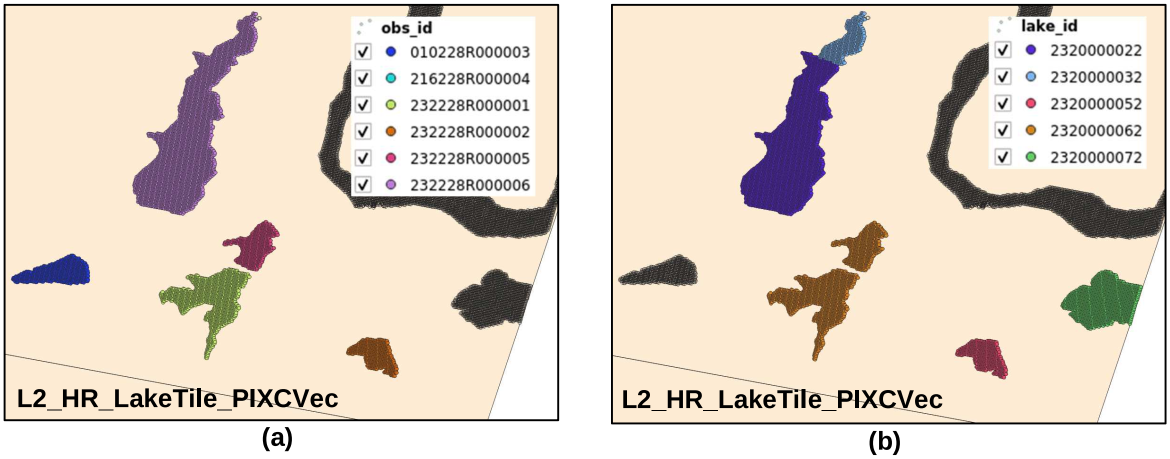


Figure 8. Output of the L2_HR_LakeTile processor (2/3): The L2_HR_LakeTile_PIXCVec file includes the information on river pixels of the L2_HR_PIXCVecRiver product, and is completed with information on pixels belonging to water features handled by the L2_HR_LakeTile processor, which have both an *obs_id* identifier (a) and a *lake_id* identifier (b).

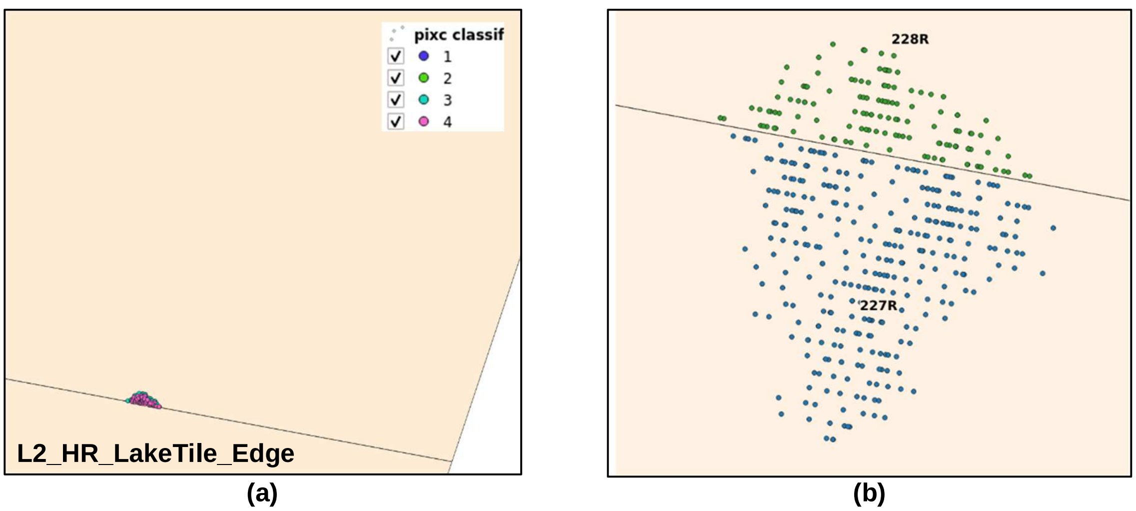


Figure 9. Output of the L2_HR_LakeTile processor (3/3): The L2_HR_LakeTile_Edge file contains pixels belonging to water features located at the along-track edge of the L2_HR_PIXC tile. (a) For tile 228R only. (b) For tiles 227R (blue dots) and 228R (green dots).

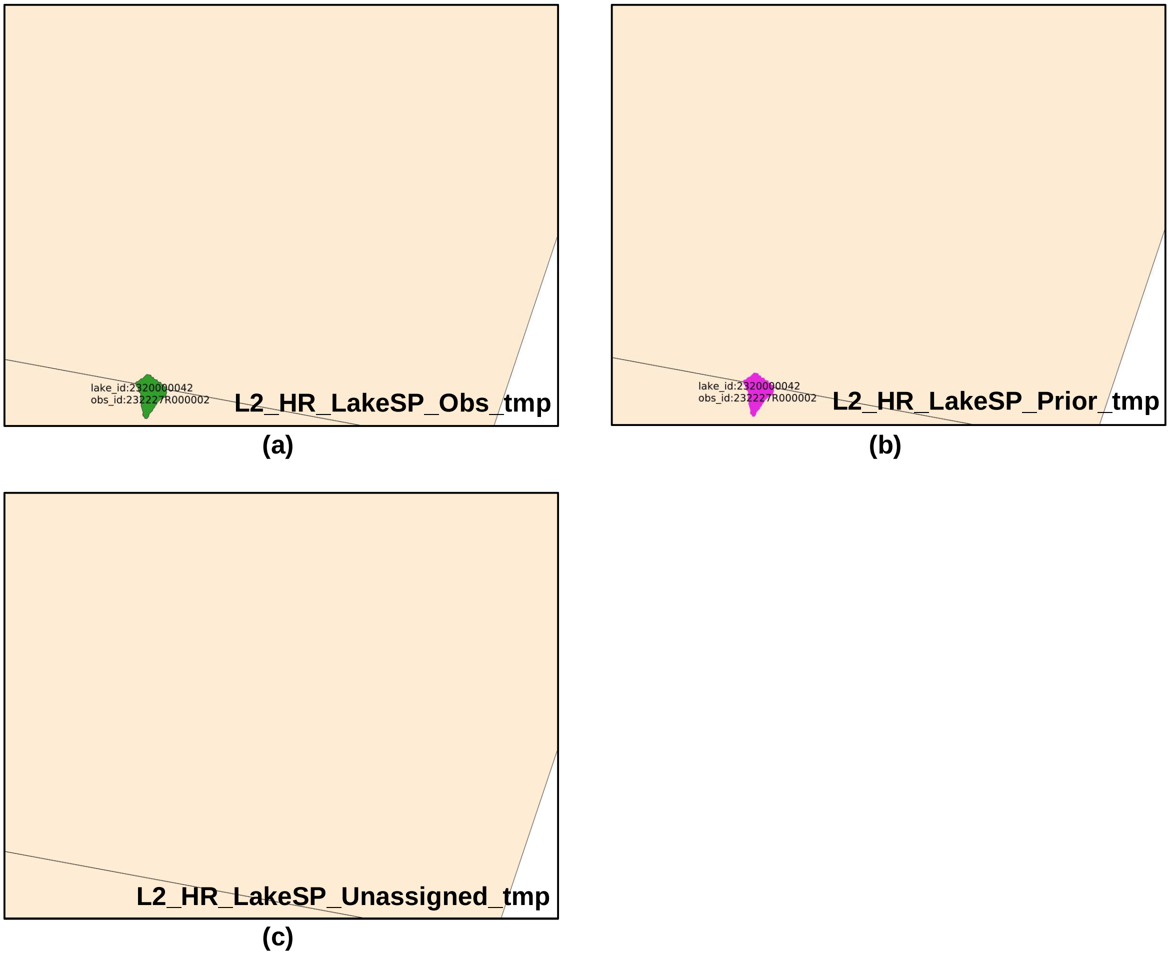
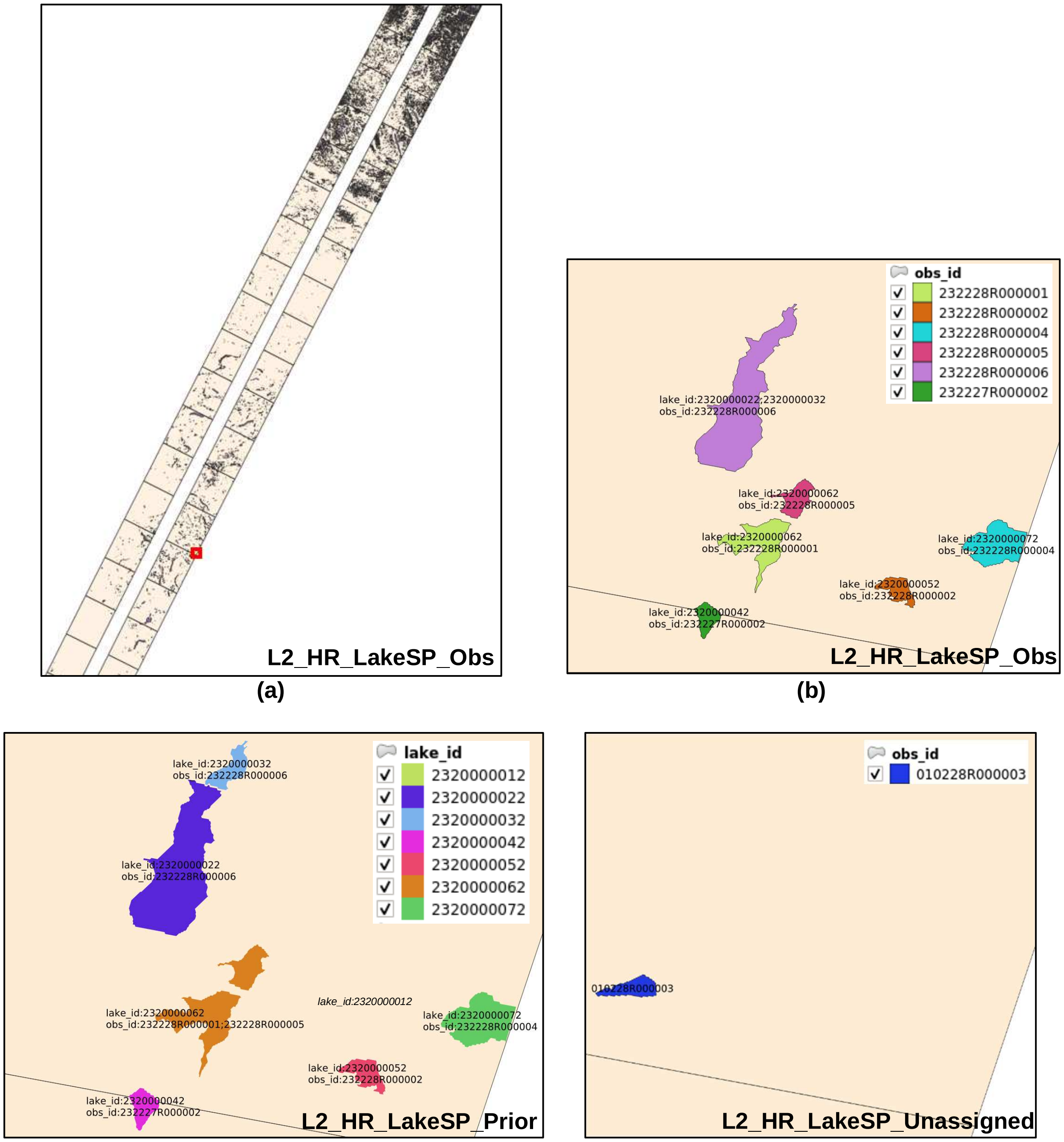


Figure 10. Temporary output of the L2_HR_LakeSP processor. The water feature located at the along-track edge of tiles 227R and 228R is related to a PLD lake. It is therefore assigned to the temporary L2_HR_LakeSP_Obs_tmp (a) and L2_HR_LakeSP_Prior_tmp (b) shapefiles, whereas the L2_HR_LakeSP_Unassigned_tmp (c) shapefile here remains empty.



(c) (d)

Figure 11. Output of the L2_HR_LakeSP processor (1/2). (a) Both swaths of the L2_HR_LakeSP_Obs granule. The area zoomed for the subsequent illustrations is delineated as a red rectangle. (b) The L2_HR_LakeSP_Obs shapefile contains the observed features linked to at least one PLD lake. If there are observed lakes in both half swaths assigned to the same PLD lake, they will have separate geometry and *obs_id*, but the same *lake_id*. (c) The L2_HR_LakeSP_Prior shapefile contains the observed PLD lakes. If there are observed lakes in both half swaths assigned to the same PLD lake, they will have a common geometry (multi-polygon) and the same *lake_id*, whereas *obs_id* will be a list of the individual identifiers. Unobserved PLD lakes are also added, as features with no geometry and only prior attributes populated (here lake_id 2320000012 has been added to the figure to symbolize the PLD lake location). (c) The L2_HR_LakeSP_Unassigned shapefile contains the observed features not intersecting any PLD lake (and that have not been assigned to a river reach by the river processing).

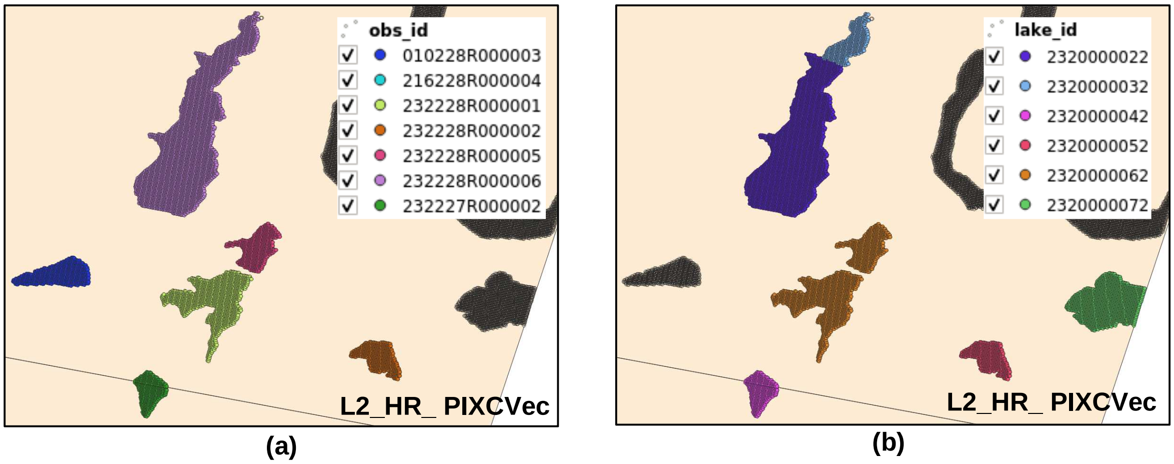


Figure 12. Output of the L2_HR_LakeSP processor (2/2): The L2_HR_PIXCVec file includes information on river pixels from the L2_HR_PIXCVecRiver intermediate product and is completed with information on pixels belonging to lakes and unassigned water features from L2_HR_LakeTile_PIXCVec intermediate file or computed by the L2_HR_LakeSP processor, which have both an *obs_id* identifier (a) and a *lake_id* identifier (b). Note that the L2_HR_PIXCVec products are split in tiles and perfectly match the L2_HR_PIXC products.

3 Algorithm Descriptions

This section describes the algorithms in the L2_HR_LakeTile and L2_HR_LakeSP processing flows. Each of the following subsections describes a module in one of the flow diagrams.

- Subsections 3.1 to 3.6 describe the L2_HR_LakeTile functions illustrated in Figure 3.
- Subsections 3.6 to 3.11 describe the L2_HR_LakeSP functions illustrated in Figure 4. They first gather pixels corresponding to water regions localized at the along-track edges of a pixel cloud tile (Section 3.7), and then compute for each of them the lake geometry and attributes (Section 3.6). Both functions are called independently for the left and right swaths. In the end, lake features previously computed by the L2_HR_LakeTile processor are gathered with those computed by the L2_HR_LakeSP processor to form the L2_HR_LakeSP product [1] (Sections 3.9 to 3.11) and the L2_HR_PIXCVec products [2] (Section 3.8).

3.1 proc_pixc_vec.compute_pixc_to_reject

3.1.1 Purpose

The L2_HR_PIXCVecRiver product identifies pixels in the L2_HR_PIXC product which have been processed by the river processor. These pixels are related to river reaches defined in the PRD [13], but note that some of these reaches are lakes that are connected to rivers.

The objective of this function is to identify pixels to remove from the list of pixels to process as lakes, i.e., those related to river reaches, except for those assigned to connected lakes.

3.1.2 Input Data

Description	Source
Indices of pixels in the L2_HR_PIXC product (*pixc_index* variable) and identifier of the reach to which each pixel has been assigned by the river processing (*reach_id* variable)	L2_HR_PIXCVecRiver

3.1.3 Output Data

Description
Indices of pixels in L2_HR_PIXC that have been assigned to a reach by the river processing and that should be excluded from lake processing (i.e., except for those assigned to connected lakes)

3.1.4 Mathematical Statement

The water body type codes [13] [15] used by the river and lake processors are indicated in Table 3. A *reach_id* ending with a type code set to 3 corresponds to a connected lake. Therefore, all pixels present in the L2_HR_PIXCVecRiver product whose *reach_id* ends with a digit other than 3, should be excluded from the lake processing. Note that there cannot be pixels with type code 2 in the L2_HR_PIXCVecRiver product.

Table 3. Water body type codes for river and lake processing.

Type Code	Water Body Type	River Processing	Lake Processing
1	River	Yes	No
2	Disconnected lake	No	Yes
3	Connected lake	Yes	Yes
4	Dam	Yes	No
5	No topology	Yes	No

3.1.5 Accuracy

The lake processing depends strongly on the river/lake assignment of the river processing. If the river processing erroneously assigns part of a disconnected lake (or all of it) to a river reach, only the remaining part of that lake (or none of it) will be included in the lake processing.

3.2 proc_pixc.extract_from_pixc

3.2.1 Purpose

The L2_HR_PIXC product contains all detected water pixels, pixels flagged as "dark water", as well as a buffer area of land pixels around them [14]. The purpose of this function is to select and extract the pixels and associated variables to keep for the lake processing based on this pixel classification, and to exclude those already assigned to a river reach as described in Section 3.1.

3.2.2 Input Data

Description	Source
Pixel cloud product	L2_HR_PIXC
Indices of pixels in L2_HR_PIXC that have been assigned to a river reach and that should be excluded from lake processing	`proc_pixc_vec.compute_pixc_to_reject`

3.2.3 Output Data

Description
Subset of L2_HR_PIXC product in terms of selected pixels and needed variables for lake processing

3.2.4 Mathematical Statement

Pixels already assigned to a river reach (according to the list produced by the `proc_pixc_vec.compute_pixc_to_reject` function) are first removed.

Second, the quality flags are considered to exclude some more pixels from further processing. The list of flags and their associated mask are given by the parameters PIXC_QUAL_LIST and PIXC_QUAL_MASK in the parameter file [18]. The current baseline is to consider all pixels, whatever their quality value for *interferogram_qual*, *classification_qual*, and *geolocation_qual*, but this will evolve post-launch. The *bright_land_flag* may also be considered (parameter EXCLUDE_BRIGHT_LAND in [18]), but the current baseline is to keep these pixels. The use of these flags remains TBD and will be decided post-launch based on what we observe in real SWOT data.

Then, the *classification* value is considered to select which of the remaining pixels to use in the lake processing. The classification flags of interest are given by the parameter CLASSIF_LIST in the parameter file [18] and chosen as indicated in Table 4. Only “Land” pixels are excluded from lake processing.

Table 4. List of classification flags values and names, indicating water detection results for L2_HR_PIXC [6].

Classification flag	Name	Kept for lake processing
1	Land	
2	Land near water	X
3	Water near land	X
4	Open water	X
5	Dark water	X
6	Low coherence but bright water	X
7	Low coherence but bright open	X

3.2.5 Accuracy

This operation does not introduce additional errors.

3.3 proc_pixc.compute_separate_regions

3.3.1 Purpose

The purpose of this function is to identify all separate water regions in the water mask previously computed.

3.3.2 Input Data

Description	Source
Range and azimuth indices of pixels kept for lake processing	`proc_pixc.extract_from_pixc`

3.3.3 Output Data

Description
Label of separate water regions for all pixels kept for lake processing

3.3.4 Mathematical Statement

First, all pixels retained for lake processing in the `proc_pixc.extract_from_pixc` step are represented in radar geometry, using their *range_index* and *azimuth_index* attributes, to obtain a binary water mask (Figure 13 (a)). Then, using the `scipy.ndimage.measurements.label` function and 4-connectivity, a unique label is given to each separate water region (Figure 13 (b)).

Figure 13. (a) Example of water mask (in radar geometry) based on the pixels retained for lake processing, and (b) segmentation into separate water regions with their labels (colors).

In a second step, an additional segmentation based on height is performed to handle lakes that are mixed in radar geometry. Indeed:

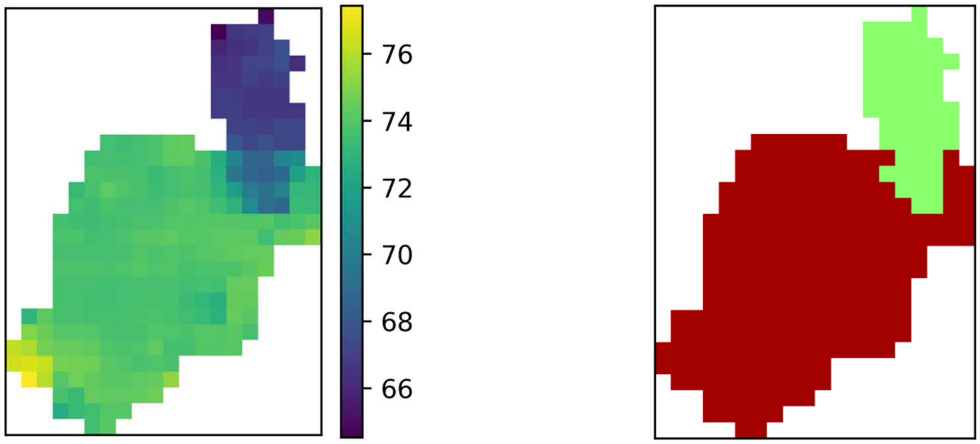
- Two or more lakes with different heights aligned in the range direction may be partly overlapping in SAR geometry because of layover.
- Two nearby lakes at different heights can also appear as one single water region because they are marginally connected through a small river segment (that may not be identified in the river database and therefore not handled by the L2_HR_RiverTile processing).

To separate such mixed lakes with different heights, the Otsu method [19] is used to perform automatic height histogram thresholding. This algorithm determines a threshold that separate pixels into two classes, by minimizing the intra-class variance and maximizing the inter-class variance. A split in two classes (A and B) is performed if

$$\mu_A + 2\sigma_A < \mu_B - 2\sigma_B$$

where μ_A and μ_B are the average heights of the two classes (assuming $\mu_A < \mu_B$), and σ_A and σ_B their standard deviations, and provided that the aggregated area of the pixels assigned to each class is larger than MIN_SIZE, the smallest allowed size for a lake feature [18]. Each of these two classes is then analyzed in the same way and possibly split, so that a maximum of four classes can be defined. Note that each class can correspond to several spatially separate groups of pixels in radar geometry, so that the observed water body may be split in more than four regions. Each resulting region must be larger than MIN_SIZE, otherwise it is regrouped with the neighbor region having the highest number of neighbor pixels. The implementation of this method in the L2_HR_LakeTile processing uses the `skimage.filters` library.

As an illustration, Figure 14 (a) shows the height of a detected water body in slant range radar geometry, composed of two separate lakes in ground geometry. One part of it, in dark blue, has an estimated water surface elevation of ~68 m while the other part, in green, has a computed water surface elevation of ~72 m. There is also a tiny part with an estimated water surface elevation of ~77 m. Figure 14 (b) shows the result of the segmentation, giving two different labels. These two resulting water bodies will be considered as two different lakes in all further lake processing. The small part of the lake with an estimated height of ~77 m has not been labelled separately as its area is smaller than MIN_SIZE.

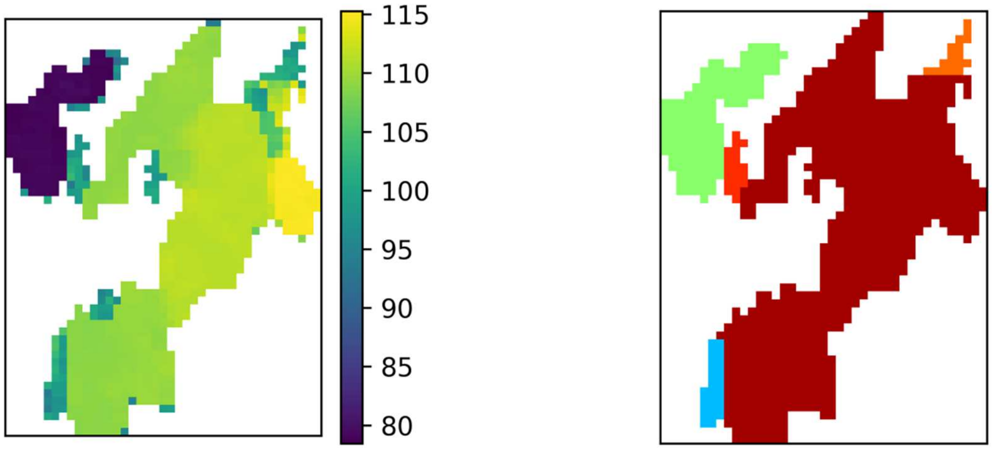


(a) (b)

Figure 14. (a) Water surface elevation of a detected water body in slant range and (b) resulting lake segmentation. The water region in (a) has been extracted from a L2_HR_PIXC product computed on a pair of L1B_HR_SLC images simulated by the HR Science Simulator.

3.3.5 Accuracy

This operation may introduce over-segmentation, as illustrated in Figure 15. This may occur in case of real height variations across very big lakes, land/water layover, etc.



(a) (b)

Figure 15. Example of over-segmentation. (a) Water surface elevation of a detected water body in slant range and (b) resulting lake segmentation. The water region in (a) has been extracted from a L2_HR_PIXC product computed on a pair of L1B_HR_SLC images simulated by the HR Science Simulator. There are two separate lakes in reality, but they are observed as connected in radar geometry because part of them are at the same distance from the radar (water/water layover). However, there are also other observed height variations that could be due to effects such as land/water layover, and that here lead to over-segmentation into altogether 5 regions.

3.4 proc_pixc.localize_regions_wrt_tile

3.4.1 Purpose

The purpose of this function is to separate labels of detected water regions entirely inside the tile, from labels of regions located at along-track edges of the tile. The former will be processed by the L2_HR_LakeTile processor, whereas the latter will be processed by the L2_HR_LakeSP processor (see Section 3.7). Regions intersecting PLD lakes located at along-track edges of the tile are also identified, and will be discarded from further L2_HR_LakeTile processing.

3.4.2 Input Data

Description	Source
Label of separate water regions for all pixels kept for lake processing	`proc_pixc. compute_separate_regions`
Azimuth index of pixels kept for lake processing	*azimuth_index* of L2_HR_PIXC
Pixel cloud rare radar grid line index to TVP index mapping	*pixc_line_qual* of L2_HR_PIXC
List of identifiers and geometries of very large PLD lakes	LakeDatabase
ogr.MultiPolygon of the PLD lake polygons intersecting the along-track edge at the beginning of the tile	`lake_db. build_border_geometry`
ogr.MultiPolygon of the PLD lake polygons intersecting the along-track edge at the end of the tile	`lake_db. build_border_geometry`
ogr.MultiPolygon of the PLD lake polygons crossing the tile, i.e. intersecting the along-track edges both at the beginning and at the end of the tile	`lake_db. build_border_geometry`

3.4.3 Output Data

Description
List of labels of water regions intersecting the line having *azimuth_index=pixc_first_line_index* or linked to a PLD lake intersecting the along-track edge at the beginning of the tile (but NOT those also intersecting the opposite along-track tile edge).
List of labels of water regions intersecting the line having *azimuth_index=pixc_last_line_index* or linked to a PLD lake intersecting the along-track edge at the end of the tile (but NOT those also intersecting the opposite along-track tile edge).
List of labels of water regions intersecting both the line having *azimuth_index=pixc_first_line_index* and the line having *azimuth_index=pixc_last_line_index*, or being linked to a PLD lake intersecting the along-track edges both at the beginning and at the end of the tile.
List of labels of water regions NOT intersecting the line having *azimuth_index=pixc_first_line_index* NOR the line having *azimuth_index=pixc_last_line_index*, NOR being linked to a PLD lake intersecting the along-track edges both at the beginning and at the end of the tile.

3.4.4 Mathematical Statement

While the L2_HR_PIXC product contains pixels that are within its tile boundaries, it is important to note that the *azimuth_index* variable extends beyond the along-track tile boundaries, as it inherits the overlap of the L1B_HR_SLC tiles, as illustrated in Figure 1 of [6]. This implies that the first and last lines within the L2_HR_PIXC tile do NOT correspond to *azimuth_index*=0 and *azimuth_index*=*interferogram_size_azimuth*-1, but need to computed. They correspond to the first and the last indices of the *pixc_line_qual* variable of the *tvp* group for which the *not_in_tile* bit is set to 0 [6], and are hereafter called *pixc_first_line_index* and *pixc_last_line_index*, respectively.

First, labels of water regions having pixels with *azimuth_index=pixc_first_line_index* are added to a list, and those having pixels with *azimuth_index=pixc_last_line_index* are put in a second list. Labels of water regions having both pixels with *azimuth_index=pixc_first_line_index* and pixels with *azimuth_index=pixc_first_line_index* are stored in a third list.

Then, the bounding box of each separate water region is compared with the collection of PLD lakes that intersect the along-track edges at the beginning and/or end of the tile. If they intersect, the label of the water region is added to the corresponding list among those cited above (unless already present). Any overlap between the two first lists and the third one is eliminated.

A fourth list is composed of all the region labels that are not in any of the three other lists. These are the regions that are fully within the tile in the along-track direction and that can therefore be fully processed by the L2_HR_LakeTile processor.

As a special case, water regions related to very large PLD lakes, spreading over more than MAX_NB_TILES_FULL_AZ tiles (see [18]), are removed from the first three lists and added to the fourth one. These water regions are therefore fully processed by the L2_HR_LakeTile processor, and will later be gathered by the L2_HR_LakeSP processor (see Section 3.7). This is slightly sub-optimal, but necessary to limit the memory usage.

3.4.5 Accuracy

This operation does not introduce additional errors.

3.5 proc_pixc.compute_edge_pixels

3.5.1 Purpose

The purpose of this function is to retrieve the indices of pixels corresponding to detected water regions located at along-track edges of the tile, or linked to PLD lakes at along-track tile edges, and to associate them with their label value and a location flag. These pixels will be written in the L2_HR_LakeTile_Edge file, to be further processed by the L2_HR_LakeSP processor (see Section 3.7).

3.5.2 Input Data

Description	Source
Label of separate water regions for all pixels kept for lake processing	`proc_pixc.` `compute_separate_regions`
List of labels of water regions intersecting the line having *azimuth_index=pixc_first_line_index* or linked to a PLD lake intersecting the along-track edge at the beginning of the tile (but NOT those also intersecting the opposite along-track tile edge).	`proc_pixc.` `localize_regions_wrt_tile`
List of labels of water regions intersecting the line having *azimuth_index=pixc_last_line_index* or linked to a PLD lake intersecting the along-track edge at the end of the tile (but NOT those also intersecting the opposite along-track tile edge).	`proc_pixc.` `localize_regions_wrt_tile`
List of labels of water regions intersecting both the line having *azimuth_index=pixc_first_line_index* and the line having *azimuth_index=pixc_last_line_index*, or linked to a PLD lake intersecting the along-track edges both at the beginning and at the end of the tile.	`proc_pixc.` `localize_regions_wrt_tile`

3.5.3 Output Data

Description
Index of pixels related to detected water regions located at one or both along-track edges of the tile.
Label of the water region corresponding to each of these pixels.
Location flag of the water region corresponding to each of these pixels.

3.5.4 Mathematical Statement

Based on the list of region labels for all pixels retained for L2_HR_LakeTile processing, and the three lists indicating the labels of all regions associated with along-track edges as described in Section 3.4, lists are created containing the index of each pixel associated with a region at one or both along-track edges, as well as its label, and a location flag which takes the following values:

- 0 if the water region intersects the line having *azimuth_index=pixc_first_line_index* or is linked to a PLD lake along-track edge at the beginning of the tile (but NOT intersecting the opposite along-track tile edge)
- 1 if the water region intersects the line having *azimuth_inde=pixc_last_line_index* or is linked to a PLD lake intersecting the along-track edge at the end of the tile (but NOT intersecting the opposite along-track tile edge)
- 2 if the water region intersects both the line having *azimuth_index=pixc_first_line_index* and the line having *azimuth_index=pixc_last_line_index*, or is linked to a PLD lake intersecting the along-track edges both at the beginning and at the end of the tile.

Thereafter, the L2_HR_LakeTile Product Generation Executable (PGE) retrieves the subset of the L2_HR_PIXC product corresponding to the indices computed by this function and writes it in the L2_HR_LakeTile_Edge file, along with the three output variables described above.

3.5.5 Accuracy

This operation does not introduce additional errors.

3.6 proc_lake.compute_lake_features

Due to its complexity, this function is split into several sub-functions that are shown in Table 5 and Figure 16. The objectives are to compute lake features for all the water regions that have been retained for L2_HR_LakeTile processing (i.e. except those assigned to river reaches, or situated at the along-track edges), to separate them into L2_HR_LakeTile_Obs, L2_HR_LakeTile_Prior, and L2_HR_LakeTile _Unassigned shapefiles, and to populate the L2_HR_LakeTile_PIXCVec variables for all the corresponding pixels.

Note that while the proc_lake.compute_lake_features function handles a tile, the sub-functions address individual water regions within the tile, as indicated in Figure 16.

Only pixels of the water region that are inside a specific cross-track range are selected as input for the following steps. This cross-track range is given by the parameters MIN_XTRACK et MAX_XTRACK provided in the parameter file [18]. If its size is consequently reduced, then the water body is considered as partially observed, and its *partial_f* flag set to 1.

Table 5. High-level description of the sub-functions within the proc_lake.compute_lake_features function.

Function Name	Description
proc_lake. compute_hconstr_geoloc	Compute the height-constrained geolocation of the pixels composing the observed water region.
proc_lake. update_pixcvec_with_hconstr_geoloc	Update the L2_HR_LakeTile_PIXCVec variables with the height-constrained geolocation of these pixels.
proc_lake. add_obs_feature	Add the observed feature to the L2_HR_LakeTile_Obs shapefile layer.
proc_lake. update_pixcvec_with_ids	Update the L2_HR_LakeTile_PIXCVec variables of the pixels composing the observed water region with the identifier of the observed feature and, eventually, of the PLD lake(s) related to them.
proc_lake. add_prior_feature	Add the prior feature to the L2_HR_LakeTile_Prior shapefile layer.

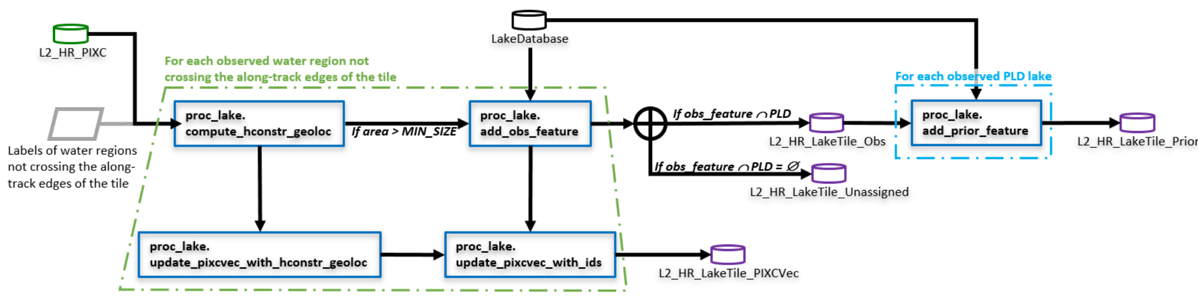


Figure 16. Flow diagram of the sub-functions within the proc_lake.compute_lake_features function.

3.6.1 proc_lake.compute_hconstr_geoloc

3.6.1.1 Purpose

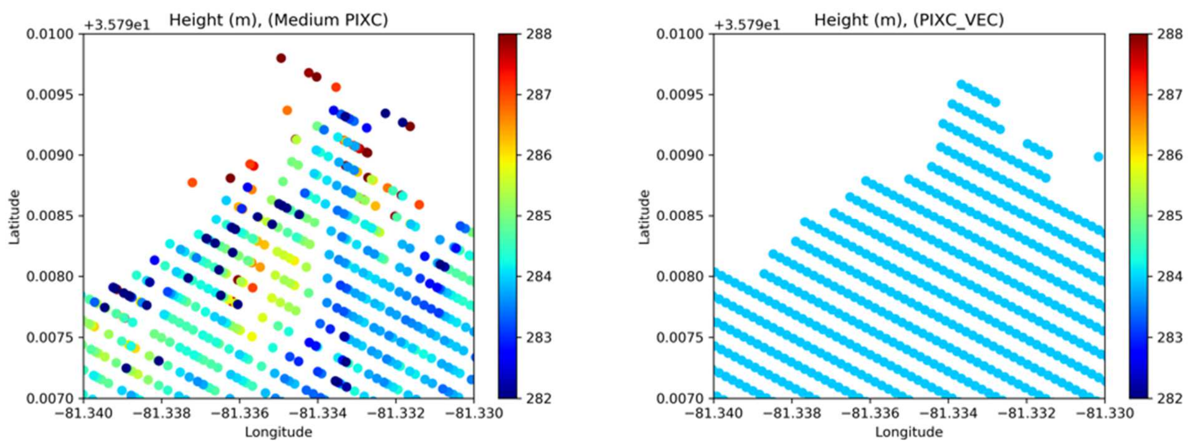
The purpose of this function is to reduce the geolocation noise of the pixels, by exploiting the fact that lakes are generally quite flat. The heights of the pixels are constrained with the average value computed over the water region [16], which results in a more regular projection of the pixels into geographical coordinates, as illustrated in Figure 17. On big lakes, a sliding window is used to compute a local average height, to allow for slow height variations across the lake.

3.6.1.2 Input Data

Description	Source
Indices of the pixels related to the observed water region, having their *cross_track* between MIN_XTRACK and MAX_XTRACK	`proc_pixc.` `compute_separate_regions`
Total area of this water region	`proc_lake.` `compute_lake_features`
Pixel cloud	L2_HR_PIXC

3.6.1.3 Output Data

Description
Height-constrained longitude, latitude and height of each pixel of the water region



(a) (b)

Figure 17. Illustration of (a) geolocation of the pixels of a lake in the L2_HR_PIXC product, and (b) height-constrained geolocation in the L2_HR_LakeTile_PIXCVec product (that will ultimately end up in the L2_HR_ PIXCVec product).

3.6.1.4 Mathematical Statement

The first step of this method is to compute the heights h_{target} that will be used to constrain the geolocation of the pixels of the water region:

- If the ground-projected area of the water region is below BIGLAKE_MIN_SIZE given in [18], the uncertainty-weighted average height [20] of the region is used for all its pixels.
- If it is larger, possible height variations across the water region need to be accounted for. The baseline algorithm is to fit a second-degree bi-dimensional polynomial model to the pixel heights over the entire water region [16]. The parameters of the method are given in [18].

The geodetic coordinates of each pixel of the observed region (i.e. longitude, latitude, and height) are converted into Cartesian coordinates (i.e. x, y, z). The distance from the satellite to each pixel center is computed based on information in the L2_HR_PIXC product:

$$R[i] = near\_range + range\_index[i] * my\_var.GEN\_RANGE\_SPACING$$

where:

- *near_range* is the slant range for the first pixel
- *range_index[i]* is the range index of pixel *i*
- *my_var.GEN_RANGE_SPACING* is the range spacing (~0.75m)

The height-constrained geolocation can now be computed with the function `geoloc.lib.geoloc.pointcloud_height_geoloc_vect` [16].

The inputs of the function are as follows:

- Coordinates (*x*, *y*, *z*) of the point *N* corresponding to the noisy pixel geolocation
- Corresponding sensor position $\overrightarrow{r_{sat}}$ and motion vector $\overrightarrow{v_{sat}}$
- Constrained height of the pixel h_{target}
- Range distance from pixel to sensor *R*
- Doppler value Z_{dop} corresponding to the plane perpendicular to $\overrightarrow{v_{sat}}$ containing *N*

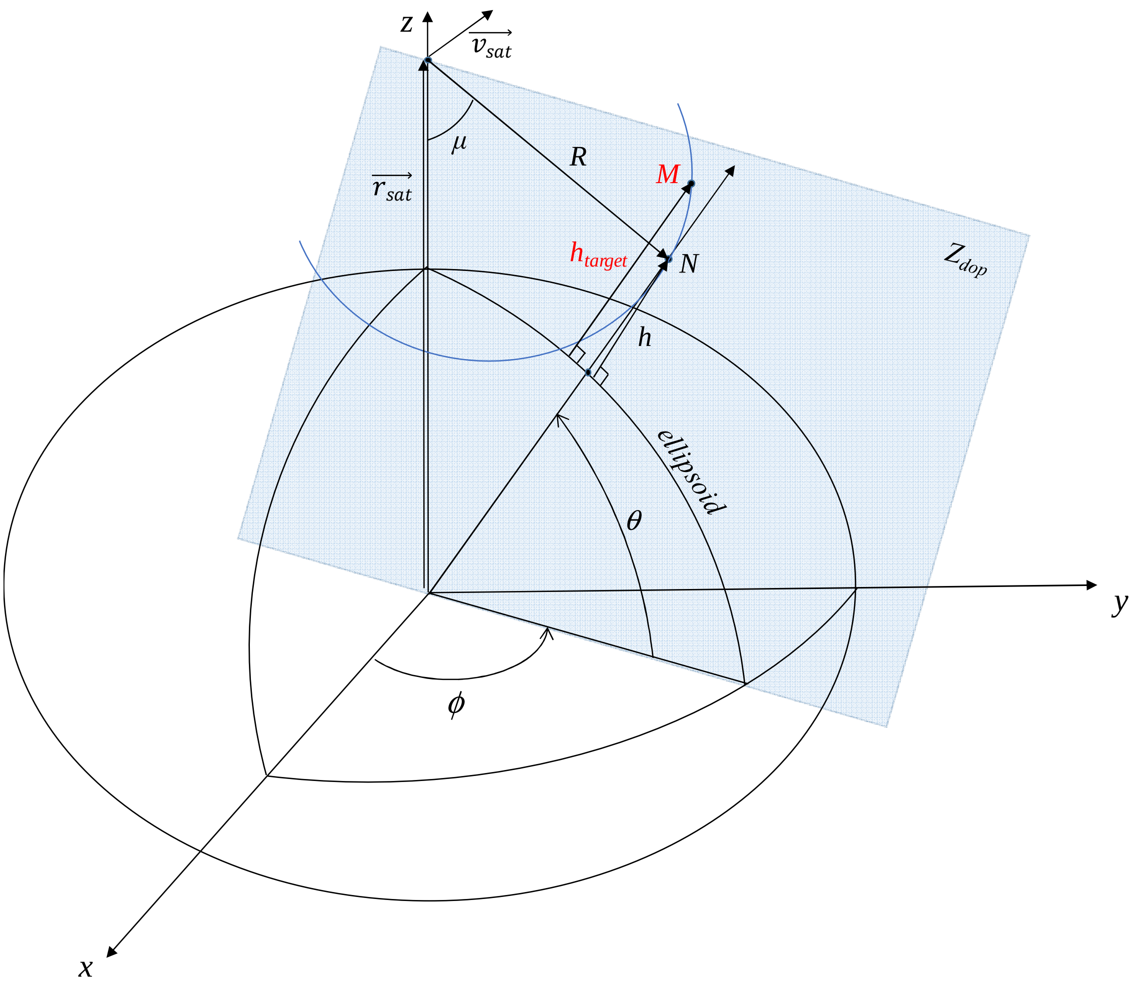


Figure 18. Illustration of the geometry of the height-constrained geolocation problem

For a given pixel, the noisy geolocation of its center *N* is defined by its range *R*, Doppler Z_{dop} and height *h*, as illustrated in Figure 18. The range *R* describes a sphere, whereas the Doppler defines a cone. Their intersection describes a circle, so that the height-constrained geolocation problem becomes 1D: We browse the range/Doppler circle (angle μ) to find the point *M* whose height projected on the ellipsoid has the value h_{target}. The latitude, longitude and height h_{target} of the point *M* represent the height-constrained geolocation of the pixel.

The full analytical description and resolution of the problem is described in [16].

3.6.1.5 Accuracy

The error introduced by this algorithm is given in [16].

3.6.2 proc_lake.update_pixcvec_with_hconstr_geoloc

3.6.2.1 Purpose

The purpose of this function is to add the height-constrained geolocation coordinates of the pixels of the current observed feature (computed in the previous step) to the L2_HR_LakeTile_PIXCVec structure.

3.6.2.2 Input Data

Description	Source
Indices of the pixels related to the current observed water region, having their *cross_track* between MIN_XTRACK and MAX_XTRACK	`proc_lake.compute_lake_features`
Height-constrained longitude, latitude and height of each pixel of the water region	`proc_pixc.compute_hconstr_geoloc`

3.6.2.3 Output Data

Description
Updated longitude, latitude and height of each pixel related to the water region in the L2_HR_LakeTile_PIXCVec object.

3.6.2.4 Mathematical Statement

For the pixels of indices specified in input, the *longitude_vectorproc*, *latitude_vectorproc*, and *height_vectorproc* variables of L2_HR_LakeTile_PIXCVec are updated with the corresponding input values.

3.6.2.5 Accuracy

This operation does not introduce additional errors.

3.6.3 proc_lake.add_obs_feature

Due to its complexity, this algorithm is split into the different sub-algorithms as shown in Table 6 and Figure 19. The objectives of this algorithm are to build the geometry (polygon) of the observed water region, compute its common attributes (water surface elevation, area, uncertainties, etc.), retrieve the PLD lake(s) related to it (if any), and add the resulting feature to the L2_HR_LakeTile_Obs shapefile layer (if the feature is related to at least one PLD lake) or the L2_HR_LakeTile_Unassigned shapefile layer (if the observed feature is not related to any PLD lake).

Table 6. High-level description of the functions within the proc_lake.add_obs_feature function.

Function Name	Description
`my_hull.compute_lake_boundaries`	Build the geometry of the observed feature.
`proc_lake.compute_common_attributes`	Compute its common attributes.
`lake_db.link_to_db`	Link the observed geometry to PLD lake geometries.

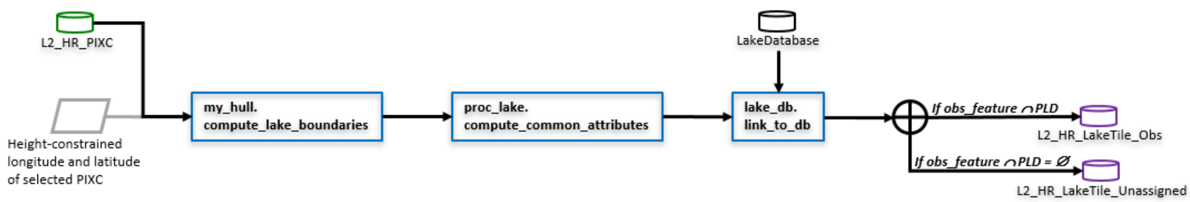


Figure 19. Flow diagram of the sub-functions of the proc_lake.add_obs_feature function.

3.6.3.1 my_hull.compute_lake_boundaries

3.6.3.1.1 Purpose

The purpose of this function is to build the polygon representing the boundary of the feature. The external ring and potential internal rings (delineating islands inside the lake) are built in radar geometry, and thereafter projected into geographical coordinates.

3.6.3.1.2 Input Data

Description	Source
Height-constrained longitude and latitude of each pixel related to the feature, having their *cross_track* between MIN_XTRACK and MAX_XTRACK	`proc_pixc.compute_hconstr_geoloc`
Azimuth and range indices of each pixel related to the feature	L2_HR_PIXC

3.6.3.1.3 Output Data

Description
Polygon delineating the boundary of the feature

3.6.3.1.4 Mathematical Statement

First, a binary mask of the water region is formed in radar geometry, using the L2_HR_PIXC range and azimuth coordinates of the pixels related to it. Then, the `skimage.measure.find_contours` function builds the external and potential internal contours (concave hull) of the region from this water mask. At last, the resulting contours, are projected in ground geometry, using their height-constrained longitude and latitude coordinates. Only the classification values listed in the CLASSIF_4HULL parameter [18] contribute to the feature boundary. Given the pixel classes included in the water mask, as described in Section

3.2.4, the polygon node positions correspond to the center of the *water near land* pixels (those next to *land near water* pixels if the layer of *water near land* pixels is more than one pixel wide). An example of a water region and the resulting polygon is given in Figure 20.

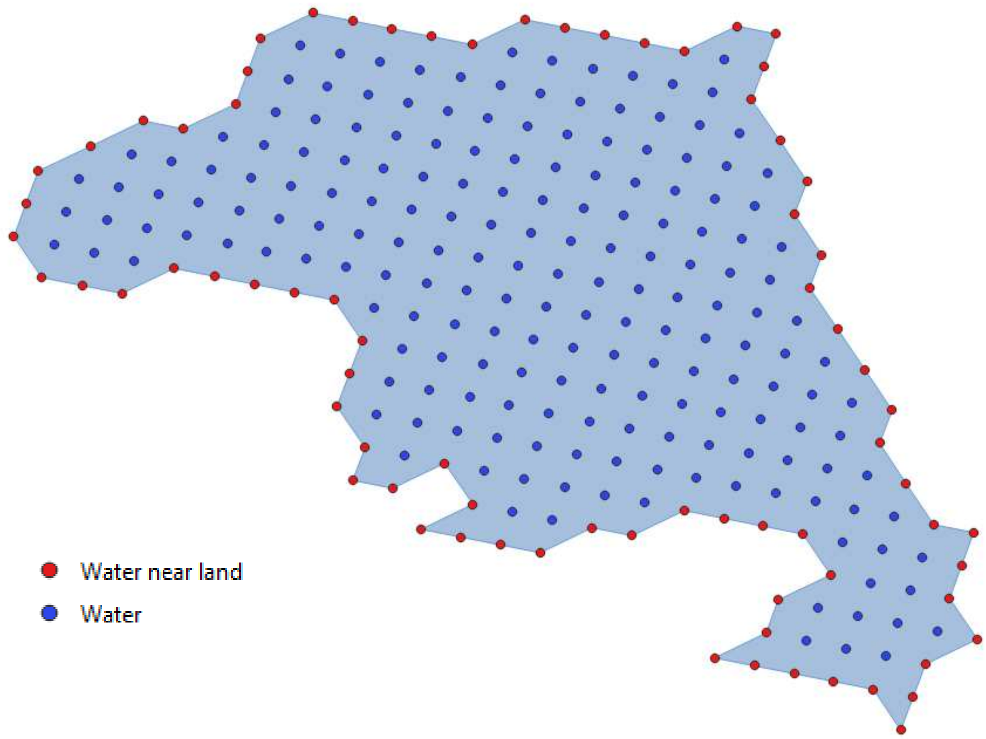


Figure 20. Polygon of a water region (in pale blue), whose nodes correspond to the center of the height-constrained water near land pixels (in red).

For very large water regions, the processing time becomes prohibitive with this approach. Therefore, water regions containing more than 50 000 pixels are split into sub-regions. Boundaries are first computed over these sub-regions and then merged.

3.6.3.1.5 Accuracy

The resulting polygons are intended to give a good indication of the actual extent of the lakes. However, because of mixed pixels, and azimuth blurring due to the limited coherence time of water, the exact position of the water/land boundary is not precisely known. There can also be errors in the water/land classification. We have for simplicity chosen to use the center of the outer *water near land* pixels (see Table 4) as node positions for the polygons. It should be noted the computation of lake water surface area is based on the areas of the underlying L2_HR_PIXC pixels, and includes water fraction estimates for boundary pixels, so that the area of the polygons will not be fully consistent with the water surface area attribute.

3.6.3.2 proc_lake.compute_common_attributes

3.6.3.2.1 Purpose

The purpose of this function is to compute the common attributes (water surface elevation, area, uncertainties…) of the feature related to the given water region.

3.6.3.2.2 Input Data

Description	Source
Pixel cloud	L2_HR_PIXC
Indices of the pixels assigned to the given feature, having their *cross_track* between MIN_XTRACK and MAX_XTRACK	`proc_lake.compute_lake_features`
partial_f flag	`proc_lake.compute_lake_features`

3.6.3.2.3 Output Data

Description
Computed common attributes

3.6.3.2.4 Mathematical Statement

3.6.3.2.4.1 Median date time

The *time* and *time_tai* attributes correspond to the mean values of the *time* and *time_tai* variables of the pixels of the feature.

The *time_str* attribute corresponds to the *time* attribute, written as a string.

3.6.3.2.4.2 Measured hydrology parameters

The water surface elevation is first computed for each pixel related to the given feature as follows:

$$wse_p = height_p - geoid_p - solid\_tide_p - load\_tide\_fes_p - pole\_tide_p$$

The following attributes of the L2_HR_PIXC product are involved [6]:

- $height_p$ is the geocentric height of the water surface with respect to the reference ellipsoid after applying corrections for media delays (due to propagation in the atmosphere)
- $geoid_p$ is the geoid height above the ellipsoid
- $solid\_tide_p$ is the solid Earth tide height
- $load\_tide\_fes_p$ is the geocentric load tide height (FES)
- $pole\_tide_p$ is the geocentric pole tide height.

The water surface elevation *wse* of the lake is the uncertainty-weighted average of the wse_p of the pixels with the classifications listed in the CLASSIF_4WSE parameter [18]. The current baseline is to only use open water pixels (i.e. classification 4) to compute *wse*, but this may evolve post-launch (TBD). The weights are computed as follows:

$$w_p = 1 / height\_std_p^2 \text{ where } height\_std_p = phase\_noise\_std_p * dheight\_dphase_p$$

These two variables are also available from the L2_HR_PIXC product [6]:

- $phase\_noise\_std_p$ is the phase noise standard deviation
- $dheight\_dphase_p$ is sensitivity of height estimate to interferogram phase.

The total uncertainty in lake water surface elevation is given by:

$$wse\_u = \sqrt{\frac{1}{nb\_pixels} * \frac{\sum_p eff\_num\_medium\_looks_p / eff\_num\_rare\_looks_p}{nb\_pixels}} * \sqrt{\frac{\sum_p w_p * \left(height_p - height\_mean\right)}{\sum_p w_p}}$$

with $$height\_mean = \frac{\sum_p w_p * height_p}{\sum_p w_p}$$

and:

- $eff\_num\_medium\_looks_p$ is the effective number of medium looks
- $eff\_num\_rare\_looks_p$ is the effective number of rare looks
- *nb_pixels* is the number of pixels involved in the water body.

The random-only component of the uncertainty in the lake water surface elevation is given by:

$$wse\_r\_u = \sqrt{phase\_var} * \left|\frac{\sum_p dheight\_dphase_p^2 * w'_p}{\sum_p dheight\_dphase_p * w'_p}\right|$$

where:

- *phase_var* is the phase noise variance of pixels involved in the water body
- w_p' are the normalized weights w_p.

wse_std is the standard deviation of wse_p of interior water pixels, after excluding ouliers (i.e. values beyond than 2-sigma of the median *wse*).

The total estimated water surface area *area_total* of the lake is computed by summing the ground-projected areas of the individual pixel's *pixel_area* in the L2_HR_PIXC product [6]. Water near land pixels (Table 4) are weighted by their estimated water fraction *water_frac* [6].

The actual SWOT-detected water surface area *area_detct* is computed similar to *area_total*, except that dark water pixels are excluded.

For further details on how these hydrological parameters and associated uncertainties are computed, refer to [21].

layover_val corresponds to the mean value of the *layover_impact* variable of the pixels related to the feature (outliers excluded).

xtrk_dist is the mean value of the *cross_track* variable of the pixels belonging to the feature (outliers excluded).

3.6.3.2.4.3 Quality indicators

The *dark_frac* attribute is the fraction of the feature that is flagged as dark water, computed as follows:

dark_frac = (*area_total* – *area_detct*) / *area_total* * 100.

The *quality_f* flag depends on *classification_qual* and *geolocation_qual* of the pixels involved in the water body. If the ratio between the number of good pixels (i.e., having *classification_qual*=0 and *geolocation_qual*=0) and the total number of pixels is above THRESHOLD_4NOMINAL (typically 70%; see [18]), then the lake is considered as good, and *quality_f* is set to 0. Otherwise, the lake is considered as suspect and *quality_f* is set to 1.

The *ice_clim_f* and *ice_dyn_f* flags are directly retrieved from the PLD [15].

The quality of the cross-over calibrations *xovr_cal_q* directly depends on the value of the *geolocation_qual* flags of the pixels involved in the water body:

- *xovr_cal_q* = 1 if *geolocation_qual* = *xovercal_suspect* [6]
- *xovr_cal_q* = 2 if *geolocation_qual* = *xoverqual_missing* [6]
- *xovr_cal_q* = 0 otherwise

3.6.3.2.4.4 Other attributes

The other common attributes of the feature are computed from their counterpart in the L2_HR_PIXC product (Table 7).

Table 7. Variable name in the L2_HR_PIXC product from which the corresponding attribute name in the lake product is computed.

Attribute name in the lake product = [lake_att]	Counterpart in the L2_HR_PIXC product = [pixc_att]
geoid_hght	*geoid*
solid_tide	*solid_earth_tide*
pole_tide	*pole_tide*
load_tidef	*load_tide_fes*
load_tideg	*load_tide_got*
dry_tropo_c	*model_dry_tropo_cor*
wet_tropo_c	*model_wet_tropo_cor*
iono_c	*iono_cor_gim_ka*
xovr_cal_c	*height_cor_xover*

The aggregation algorithm used for these attributes is equivalent to the one used for the computation of *wse* (see Section 3.6.3.2.4.2). As for wse, the five first attributes are computed by considering only pixels with classification listed in the CLASSIF_4WSE parameter [18] (see section 3.6.3.2.4.2). On the contrary, the other attributes are computed using all pixels retained for lake processing as described in Table 4.

3.6.3.2.5 Accuracy

The uncertainties are expressed for the attributes related to the elevation and area. The way to compute them is given above.

This operation does not introduce additional errors for the other attributes.

3.6.3.3 lake_db.link_to_db

3.6.3.3.1 Purpose

This function establishes the link between an observed water region and the PLD (i.e., the global database of known lakes). Any PLD lake intersecting the polygon of the water region, and whose overlap is above a threshold, is considered as linked to it. In this case, the water region is considered as a lake, and the corresponding feature will be stored in the L2_HR_LakeTile_Obs shapefile (and in the L2_HR_LakeTile_Prior shapefile). If not, it will be stored in the L2_HR_LakeTile_Unassigned shapefile.

3.6.3.3.2 Input Data

Description	Source
Water region polygon geometry	`my_hull.compute_lake_boundaries`
Height-constrained longitude and latitude of each pixel related to the water region	`proc_pixc.compute_hconstr_geoloc`
PLD lakes polygons, identifier, and polygons of their influence area	LakeDatabase

3.6.3.3.3 Output Data

Description
List of the identifiers of the PLD lakes intersecting the observed water region, ordered by decreasing overlapping area
List of the fractions of observed polygon covered by each intersecting PLD lake, ordered by decreasing overlapping area
List of the identifiers of the PLD lakes for each pixel related to the water region

3.6.3.3.4 Mathematical Statement

The first step is to determine which PLD lakes sufficiently overlap the observed polygon to be linked to it. To do so, only PLD lakes intersecting the polygon of the observed water region are selected (as explained in section 3.6.3.1, the polygon is based on height-constrained geolocations of the pixels in the observed region). Then, an intersecting PLD lake is retained only if the ratio between the area of the intersection of the PLD lake and the observed polygon, and the total area of the observed polygon is above MIN_OVERLAP (typically in the order of 2%) [18]. The prior attributes of the observed lake are set to the values corresponding to the PLD lake having the largest overlapping area.

(a) (b) (c)

Figure 21. Association of observed water regions (a) to PLD lakes (b). PLD lakes 232142722 and 232132172 are not observed. The observed polygon on the left is linked to no PLD lake: therefore, this feature will end up in the L2_HR_LakeTile_Unassigned shapefile. The upper-right observed polygon is linked to both PLD lakes 232008092 and 232009412: therefore, it will become a single feature in the L2_HR_LakeTile_Obs shapefile, and two distinct features in the L2_HR_LakeTile_Prior shapefile. The observed feature related to PLD lake 232123812 will be identical in the L2_HR_LakeTile_Obs and L2_HR_LakeTile_Prior shapefiles. The zoom (c) illustrates how the pixels of the upper right observed water region are split between two overlapping PLD lakes based on their influence areas.

The second step is to attach each pixel related to the input water region to its corresponding PLD lake. This is directly set if a single PLD lake is attached to the input water region. If there are two or more PLD lakes, the “influence area” polygons of these PLD lakes are retrieved. One influence area polygon corresponds to a single PLD lake and delineates a region around it. These polygons have been precomputed based on Voronoï diagrams and are available in the PLD [15]. The polygons are distinct (Figure 22) and entirely cover the continents. Each pixel of the input water region is associated with the PLD lake corresponding to the influence area polygon it falls in (Figure 21 (c)). In the rare cases where there is no influence area polygon (for example, badly geolocated coastal pixels that fall in the ocean where there is no influence area polygon), the pixel is associated to the PLD lake with the closest influence area polygon, in terms of Euclidian distance from pixel geolocation.

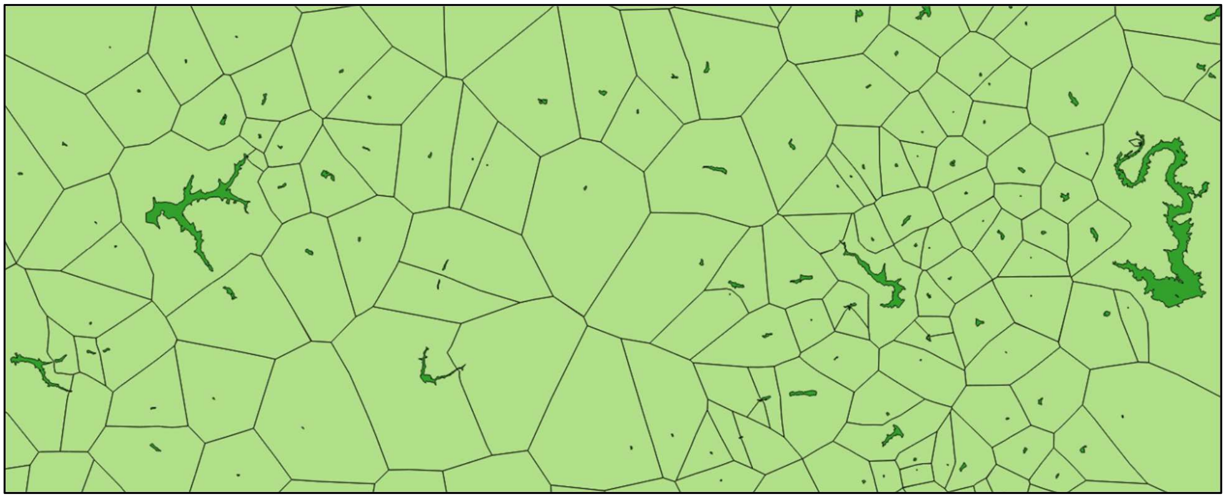


Figure 22. PLD lakes (deep green) and their related influence area polygon (light green).

3.6.3.3.5 Accuracy

The link between an observed water region and the PLD is based on polygon intersection. Therefore, it is highly dependent on the accuracy of the detected water mask and the geolocation of its pixels, and also on the accuracy of the extent and geolocation of the PLD lake polygon. Such errors may lead to assignment errors.

In the same way, the correspondence between the pixels of the water region and one single PLD lake is highly dependent of the construction of the lake influence area polygons, which depends itself of the accuracy of the PLD lake polygons (location, shape, etc.). Moreover, these Voronoï diagrams are purely geometrical, and do not reflect the actual catchment basins.

Missing river reaches in the PRD [13] may also have a strong impact on the assignment of pixels to lakes: pixels that actually correspond to rivers, but that are not assigned to a PRD reach during river processing (because that river reach is not included in the PRD), will not be eliminated from further lake processing as described in section 3.1, and may therefore be erroneously assigned to a connected PLD lake, thereby degrading its estimated water surface area, WSE, etc.

3.6.4 proc_lake.update_pixcvec_with_ids

3.6.4.1 Purpose

The purpose of this function is to store, in the L2_HR_LakeTile_PIXCVec structure, the PLD and observation identifiers of all the pixels of the current observed feature.

3.6.4.2 Input Data

Description	Source
Indices of the pixels related to the current observed water region	`proc_lake.compute_lake_features`

Observation identifier of the current observed feature	`proc_lake.compute_lake_features`
PLD identifier of the pixels of the current observed feature	`lake_db.link_to_db`

3.6.4.3 Output Data

Description
Updated *obs_id* and *lake_id* of each pixel related to the water region in the L2_HR_LakeTile_PIXCVec object.

3.6.4.4 Mathematical Statement

For the pixels of indices specified in input, the *obs_id*, and *lake_id* variables of L2_HR_LakeTile_PIXCVec are updated with the corresponding input values.

3.6.4.5 Accuracy

This operation does not introduce additional errors.

3.6.5 proc_lake.add_prior_feature

Due to its complexity, this function is split into the sub-functions shown in Table 8 and Figure 23. It is equivalent to the `proc_lake.add_obs_feature` function described in Section 3.6.3 and reuses some of its sub-functions, with different input parameters. The objectives of this algorithm are to build the (multi-) polygon for a given PLD lake, to compute its common attributes and its storage change, and to add the resulting feature to the L2_HR_LakeTile_Prior shapefile layer.

Table 8. High-level description of the functions within the proc_lake.add_prior_feature function.

Function Name	Description
`proc_lake.build_prior_boundary`	Build the geometry of the prior feature.
`proc_lake.compute_common_attributes`	Compute its common attributes.
`proc_lake.compute_storage_change`	Compute storage change values for this prior feature.

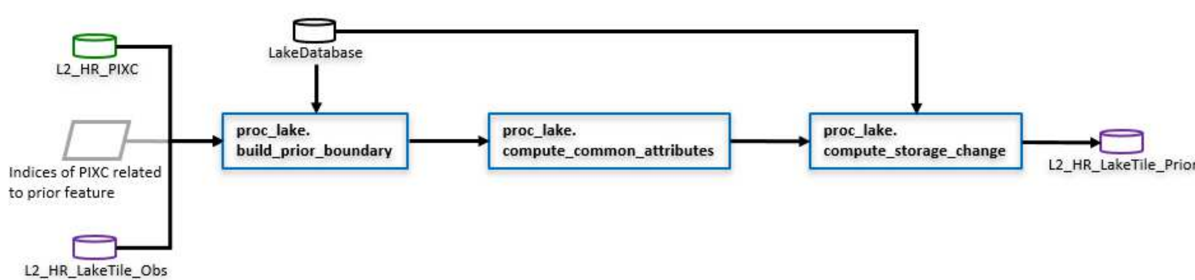


Figure 23. Flow diagram of the lower-level algorithms within the proc_lake.add_prior_feature function.

3.6.5.1 *proc_lake.build_prior_boundary*

3.6.5.1.1 Purpose

The purpose of this function is to build the geometry of the prior feature, from the pixels related to it or directly from the observed feature related to it if it is linked to only one PLD lake.

3.6.5.1.2 Input Data

Description	Source
PLD lake object	`lake_db.PriorLake`
All L2_HR_LakeTile_Obs features linked to the PLD lake	`proc_lake.compute_lake_features`
Indices of the pixels related to the prior feature	`proc_lake.compute_lake_features`

3.6.5.1.3 Output Data

Description
Polygon delineating the external and potential internal boundaries of the prior feature

3.6.5.1.4 Mathematical Statement

The processing depends on the correspondence between the current PLD lake and its associated observed lake(s):

- If the PLD lake corresponds to a single L2_HR_LakeTile_Obs feature:
  - If this L2_HR_LakeTile_Obs feature is only related to this PLD lake, the geometry of the prior feature is the geometry of the L2_HR_LakeTile_Obs feature
  - If not, the geometry of the prior feature is computed with the function `my_hull.compute_lake_boundaries` (Section 3.6.3.1), with, as input, the height-constrained longitude and latitude, and azimuth and range indices of each pixel related to the prior feature (computed in the `lake_db.link_to_db` step described in Section 3.6.3.3).
- If the PLD lake corresponds to two or more L2_HR_LakeTile_Obs features, the processing is the same as described above, for each of these L2_HR_LakeTile_Obs features, and the resulting geometries are stored in a single ogr.MultiPolygon geometry.

3.6.5.1.5 Accuracy

Refer to Section 3.6.3.1.5 concerning the accuracy of the polygon nodes, and Section 3.6.3.3.5 concerning the accuracy of the assignment of pixels to PLD lakes.

3.6.5.2 *proc_lake.compute_common_attributes*

This function is already described in Section 3.6.3.2. The only difference here is that the input pixel indices correspond to pixels assigned to the prior feature.

3.6.5.3 proc_lake.compute_storage_change

3.6.5.3.1 Purpose

This function computes the storage change between the observed state of a PLD lake (water surface elevation and area) and a reference state.

3.6.5.3.2 Input Data

Description	Source
PLD lake object	`lake_db. PriorLake`
Attributes of the prior feature related to the PLD lake	`proc_lake. add_prior_feature`
List of observed features linked to the PLD lake	`proc_lake. compute_lake_features`
Indices of the pixels related to the prior feature	`proc_lake. compute_lake_features`
Pixel cloud	L2_HR_PIXC

3.6.5.3.3 Output Data

Description
Storage change values computed with linear and quadratic equations, and associated uncertainties

3.6.5.3.4 Mathematical Statement

The volume of a lake is highly dependent on its bathymetry. As the bathymetry is unknown for the large majority of the lakes in the world, approximate bathymetry models are used to compute storage change. Two models (i.e., linear and quadratic) are considered. Furthermore, two approaches (i.e., direct and incremental) are used to compute storage change, as described below. This leads to four different storage change estimates which are added to the _Prior shapefile layer.

3.6.5.3.4.1 Direct approach

With the linear hypothesis, the volume change is approximated by the volume of a trapezoid (Figure 24). This formula is appropriate to lakes that are narrow or landlocked in relief.

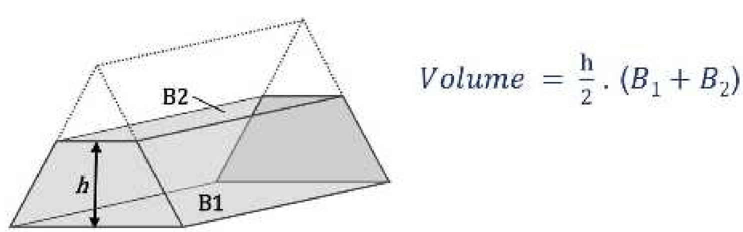


Figure 24. Volume of a trapezoidal prism.

This hypothesis leads to the following equation for the volume change between the state of the observation t_i and a reference state *ref* (Figure 26):

$$\Delta V_l(t_i) = \frac{wse - wse_{ref}}{2} * \left(area\_total + area\_total_{ref}\right) \textbf{ [Eq. 1]}$$

where *wse* (in m) and *area_total* (in km^2) are the observed water surface elevation and total area of the prior feature at time t_i (Section 3.6.5.2), and wse_{ref} and $area\_total_{ref}$ the corresponding reference elevation and area from the PLD.

With the quadratic hypothesis, the volume change is approximated by the volume of a truncated pyramid (Figure 25) [22]. This formula is appropriate to lakes having a convex shape.

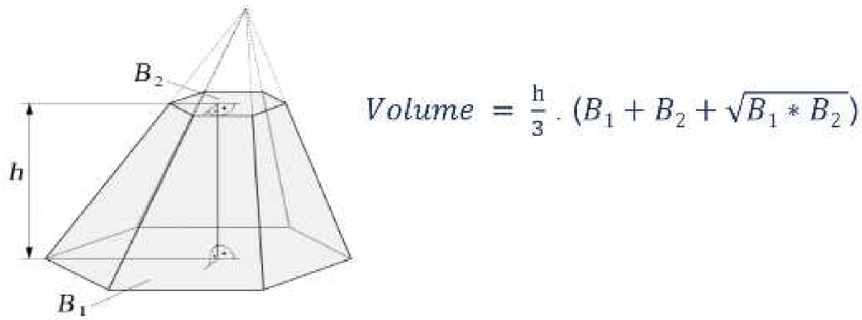


Figure 25. Volume of a truncated pyramid.

This hypothesis leads to the following equation for the volume change between the state of the observation t_i and a reference state *ref* (Figure 26):

$$\Delta V_q(t_i, ref) = \frac{wse - wse_{ref}}{3} * \left(area\_total + area\_total_{ref} + \sqrt{area\_total * area\_total_{ref}}\right)$$
[Eq. 2]

The *delta_s_l* and *delta_s_q* attributes should indicate the storage change between the current observation t_i and the first valid observation by SWOT at time t_0, rather than with respect to the reference state mentioned above (Figure 26).

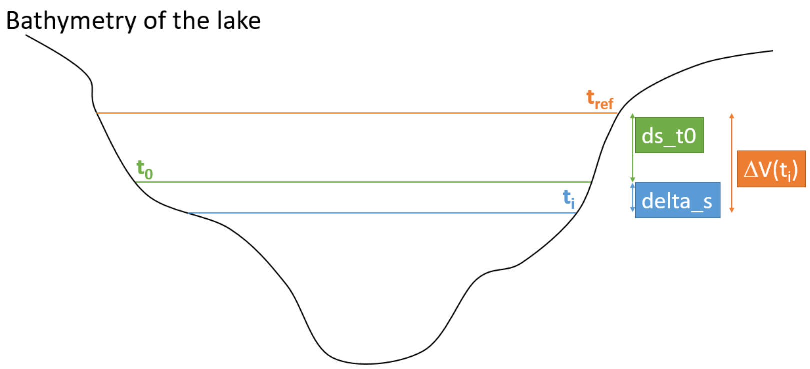


Figure 26. Illustration of the different states that occur in the storage change equations for the direct approach.

Note that the storage change attributes are given in km^3, whereas *wse* are given in m and *area_total* in km^2. Therefore, the attributes available in the L2_HR_LakeTile_Prior shapefile layer are derived as follows:

$$delta\_s\_[l|d](t_i, ref) = {\Delta V_{l|q}(t_i)}/{1000} - ds\_t0$$

where *ds_t0* is the storage change between the first valid observation by SWOT and the reference state, which will be added to the PLD when it is updated post-launch [15] (up to then *ds_t0* is set to zero and the storage change (*delta_s_l* or *delta_s_q*) is given with respect to the reference state).

The corresponding uncertainty is given by:

$$ds\_u\_[l|q] = \sqrt{\begin{array}{c} wse_u{}^2 \left(\frac{\partial \Delta \mathrm{V}}{\partial \mathrm{wse}}\right)^2 + \sigma\left(wse_{ref}\right)^2 \left(\frac{\partial \Delta \mathrm{V}}{\partial \mathrm{wse_{ref}}}\right)^2 \\ + area\_tot\_u^2 \left(\frac{\partial \Delta \mathrm{V}}{\partial \mathrm{area\_total}}\right)^2 + \sigma(area\_total_{ref})^2 \left(\frac{\partial \Delta \mathrm{V}}{\partial area\_total_{ref}}\right)^2 \end{array}}$$

where *wse_u* and *area_tot_u* are attributes of the prior feature (Section 3.6.5.2).

With the linear hypothesis:

$$\frac{\partial \Delta \mathrm{V}}{\partial \mathrm{wse}} = -\frac{\partial \Delta \mathrm{V}}{\partial \mathrm{wse_{ref}}} = \frac{1}{2}(\mathrm{area\_total} + \mathrm{area\_total_{ref}})$$

$$\frac{\partial \Delta \mathrm{V}}{\partial \mathrm{area\_total}} = \frac{\partial \Delta \mathrm{V}}{\partial \mathrm{area\_total_{ref}}} = \frac{1}{2}(\mathrm{wse} - \mathrm{wse_{ref}})$$

With the quadratic hypothesis:

$$\frac{\partial \Delta \mathrm{V}}{\partial \mathrm{wse}} = -\frac{\partial \Delta \mathrm{V}}{\partial \mathrm{wse}_{\mathrm{ref}}} = \frac{1}{3}\left(\mathrm{area\_total} + \mathrm{area\_total}_{\mathrm{ref}} + \sqrt{\mathrm{area\_total} * \mathrm{area\_total}_{\mathrm{ref}}}\right)$$

$$\frac{\partial \Delta \mathrm{V}}{\partial \mathrm{area\_total}} = \frac{1}{3}(\mathrm{wse} - \mathrm{wse}_{\mathrm{ref}})\left(1 + \frac{1}{2}\sqrt{\frac{\mathrm{area\_total}_{\mathrm{ref}}}{\mathrm{area\_total}}}\right)$$

$$\frac{\partial \Delta \mathrm{V}}{\partial \mathrm{area\_total}_{\mathrm{ref}}} = \frac{1}{3}(\mathrm{wse} - \mathrm{wse}_{\mathrm{ref}})\left(1 + \frac{1}{2}\sqrt{\frac{\mathrm{area\_total}}{\mathrm{area\_total}_{\mathrm{ref}}}}\right)$$

3.6.5.3.4.2 Incremental approach

The second approach uses the *hypso_curve* table that will be added to the PLD [15] approximately one year into the SWOT mission. This table will contain discrete points on a curve fitting the (*wse*, *area_total*) pairs observed so far, allowing an incremental volume computation, as illustrated in Figure 27. In this case, the volume change equations [Eq. 1] and [Eq. 2] become:

$$\Delta V_{l|q}(t_i) = \sum_{j=i}^{N=ref-1} \Delta V_{l|q}\left(t_{j+1}, t_j\right)$$

where *j* indicates the states of the *hypso_curve* table that are between the state *i* of the observation and the reference state *ref*. All other equations above are applied in the same way.

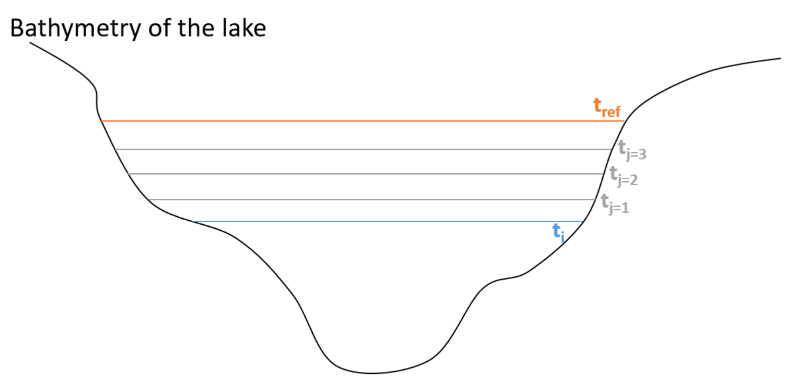


Figure 27. Illustration of the states j to compute storage change using the *hypso_curve* table, for the incremental approach.

Note that the results of this approach will become available during the first reprocessing campaign, projected approximately one year after launch.

3.6.5.3.5 Accuracy

The accuracy depends on the method used to compute volume change.

If the difference in water surface elevation between the observation and the reference is small, the results obtained with the direct approach are expected to be good approximations, but for higher elevation differences, and if the linear and quadratic models do not well reflect the bathymetry, the estimated storage change will be less accurate.

The incremental approach is expected to be more accurate. Indeed, the stepwise computation between the observation and the reference, based on the *hypso_curve* table of the PLD, is thus less sensitive to the shape of the bathymetry, reducing the error.

3.7 proc_pixc_sp.swath_global_relabeling

Some detected water bodies are divided into two or more parts because they are situated at the border between consecutive L2_HR_PIXC tile granules [6]. Pixels related to them have been put aside in L2_HR_LakeTile_Edge files [3] for further processing.

The objective of this function is to gather these pixels, so that pixels belonging to the same connected region have a unique label, at the continent-pass scale and for one swath (left or right).

Due to its complexity, this function is split into several sub-functions that are shown in Table 9 and Figure 28.

After the processing of pixels at the edge of two consecutive tiles, and thereafter at the continent-pass scale (for one swath) with these sub-functions, the height-based lake segmentation is run (as for LakeTile, see section 3.3.4).

Table 9. High-level description of the sub-functions within the proc_pixc_sp.swath_global_relabeling function.

Function Name	Description
proc_pixc_sp. compute_range_variation_between_tiles	Compute near range variation of the first pixels between two consecutive tiles.
proc_pixc_sp.gather_regions_at_edge	Identify regions split at the edge of two consecutive tiles.
proc_pixc_sp.gather_regions_of_swath	Gather split regions at the continent-pass scale (for one swath).

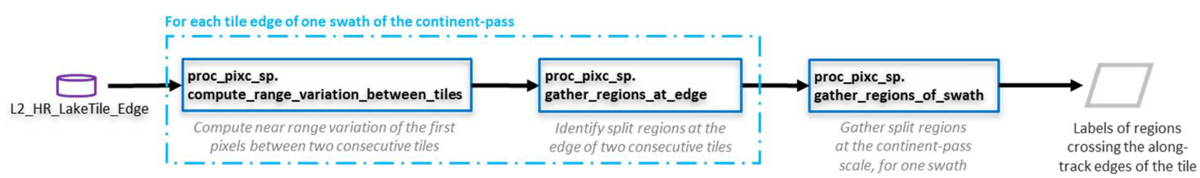


Figure 28. Flow diagram of the sub-functions within the proc_pixc_sp.swath_global_relabeling function.

3.7.1 proc_pixc_sp.compute_range_variation_between_tiles

3.7.1.1 Purpose

The purpose of this function is to compute the difference between the first pixels in range between two consecutives tiles (tile “N” and tile “N+1” hereafter).

3.7.1.2 Input Data

Description	Source
For both tiles, slant range to the first range bin and spacing between range samples in the underlying 2-D arrays upon which the 1-D pixel cloud samples are taken; these correspond to *near_range* and *nominal_slant_range_spacing* global attributes of the L2_HR_LakeTile_Edge product	L2_HR_LakeTile_Edge files related to tiles “N” and “N+1”

3.7.1.3 Output Data

Description
Near range variation between the two consecutive tiles

3.7.1.4 Mathematical Statement

At the connection between processed tile “N” and its consecutive tile “N+1”, the range variation is computed by:

$$\Delta near\_range = \frac{near\_range_{tile_N} - near\_range_{tile_{N+1}}}{nominal\_slant\_range\_spacing_{tile_N}}$$

3.7.1.5 Accuracy

This operation does not introduce additional errors.

3.7.2 proc_pixc_sp.gather_regions_at_edge

3.7.2.1 Purpose

The purpose of this function is to gather detected water regions that are split across the edge between two consecutive tiles, i.e. establish a list of corresponding region labels.

3.7.2.2 Input Data

Description	Source
rangex_index, *azimuth_index,* and *edge_label* of pixels at the edge of both tiles “N” and “N+1”	L2_HR_LakeTile_Edge files related to tiles “N” and “N+1”
pixc_last_line_index of tile “N”, i.e. *azimuth index* corresponding to the last slant range interferogram line that is inside tile “N”	L2_HR_LakeTile_Edge file related to tile “N”
pixc_first_line_index of tile “N+1”, i.e. *azimuth index* corresponding to the first slant range interferogram line that is inside tile “N+1”	L2_HR_LakeTile_Edge file related to tile “N+1”
Near range variation between the two consecutive tiles	`proc_pixc_sp.compute_range_variation_between_tiles`

3.7.2.3 Output Data

Description
List of LakeTile_Edge labels that correspond to the same detected water region across the edge between two consecutive tiles

3.7.2.4 Mathematical Statement

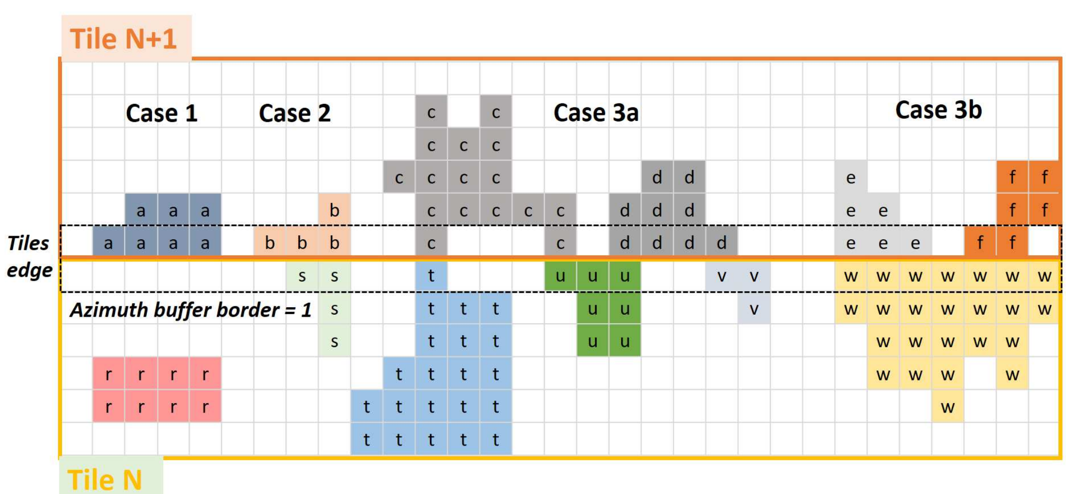
Pixels at the last line of L2_HR_LakeTile_Edge tile “N” (i.e. with *azimuth_index* =

pixc_last_line_index) and at the first line of L2_HR_LakeTile_Edge tile "N+1" (i.e. with *azimuth_index* = *pixc_first_line_index*) are first aligned by making their *range_index* compatible, using the near range difference computed in the previous step (this is necessary to make pixels in the same column correspond to the same distance from the radar in the merged mask). The obtained binary water mask (in radar geometry) is then processed to identify connected water regions, as explained in Section 3.3.4. However, this step is here applied only to the last azimuth line within tile "N" and the first azimuth line within tile "N+1".

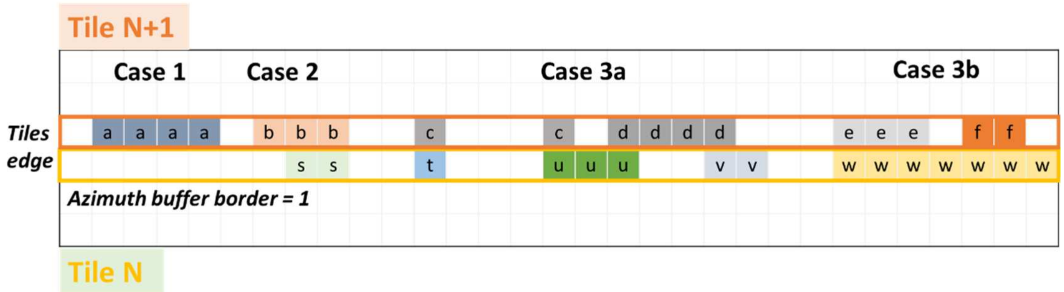
The last step identifies labels on both sides of the tile edge that belong to the same detected water region. Three cases may occur (see example in Figure 29):

- Case 1: No label of tile "N" corresponds to a label of tile "N+1" or reversely. This happens when a lake is entirely located at the boundary of one tile, but does not cross the border. In this case, the label of tile "N" or "N+1" is the single element of the list added to the output list. This is the case of label *a* in Figure 29 (a) and (b).
- Case 2: One label of tile "N" corresponds to one label of tile "N+1". In this case, the list composed the old label of tile "N" and the old label of tile "N+1" is added to the output list. This is the case of labels *b* and *s* in Figure 29 (a) and (b).
- Case 3: Several labels of tile "N" match one or more labels of tile "N+1", or reversely. This case rarely occurs. It means the lake "meanders" along the border between two tiles. In this case, we obtain a list of multiple matches. This is the case of labels *c*, *d*, *u*, and *v*, and labels *e*, *f* and *w* in Figure 29 (a) and (b).

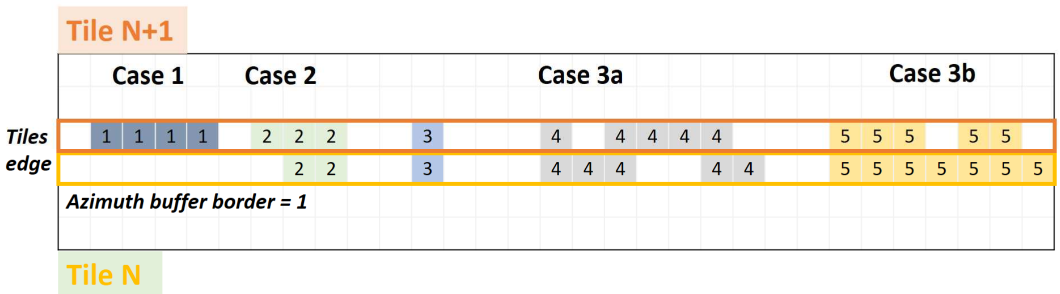
In this example, we first identify *5* separate regions as illustrated in Figure 29 (c). The matching labels for the two tiles are shown in the following list: [[*a*], [*b*, *s*], [***c***, *t*], [***c***, *u*, *d*, *v*], [*e*, *w*, *f*]]. Region *3* and Region *4* both contain label *c*. Therefore, sub-lists [***c***, *t*] and [***c***, *u*, *d*, *v*] are gathered to obtain the resulting output list: [[*a*], [*b*, *s*], [*c*, *d*, *t*, *u*, *v*], [*e*, *w*, *f*]]. We can notice that the region with label *r*, corresponding to a PLD lake crossing the tile edge, and therefore present in LakeTile_Edge (see Section 3.4), is not taken into account here, as it is not located at the tile border.



(a)



(b)



(c)

Figure 29. Illustration of LakeTile labels of consecutive tiles "N" and "N+1", (a) at the scale of the tiles, and (b) at the along-track edge. (c) A new segmentation and temporary relabeling (*1-5*) at the tile edge permit to identify labels that belong to the same detected water body (e.g. *b* and *s*).

3.7.2.5 Accuracy

See Section 3.3.5.

3.7.3 proc_pixc_sp.gather_regions_of_swath

3.7.3.1 Purpose

The purpose of this function is to reorganize all LakeTile_Edge labels at the scale of the continent pass (for one swath), gathering also water regions covering more than two tiles.

3.7.3.2 Input Data

Description	Source
Lists of LakeTile_Edge labels that correspond to the same detected water region across consecutive tiles	`proc_pixc_sp.gather_regions_at_edge`
LakeTile_Edge label for each pixel contained in features at the edges of all tiles of the continent-pass (*edge_label* variable)	L2_HR_LakeTile_Edge related to all tiles of the continent-pass (for each swath)

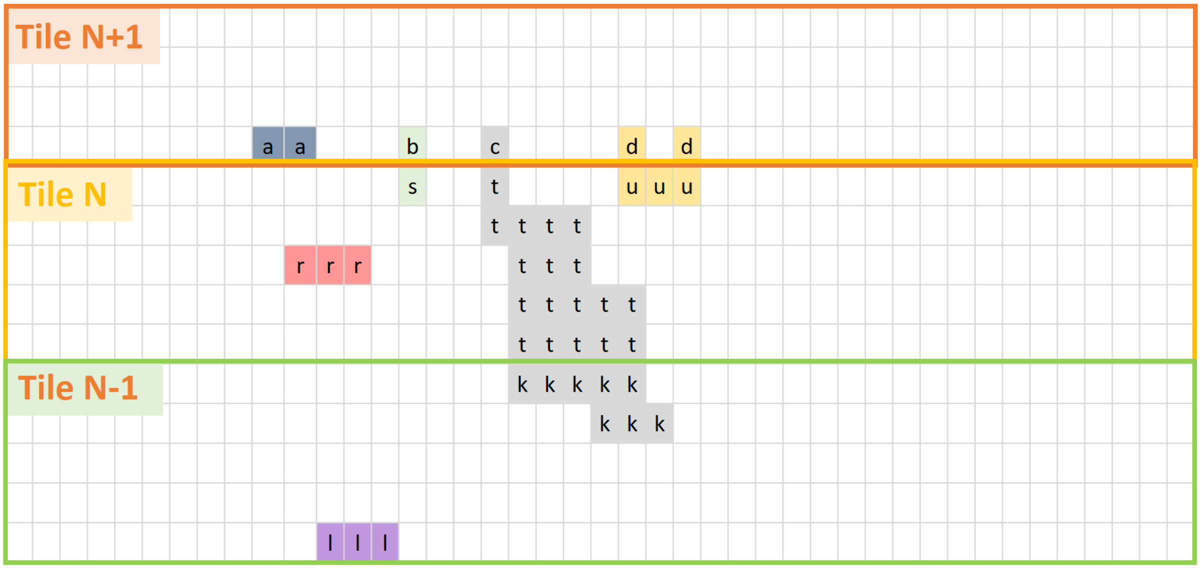
3.7.3.3 Output Data

Description
Updated labels for each pixel with new labels gathering pixels by regions within whole swath

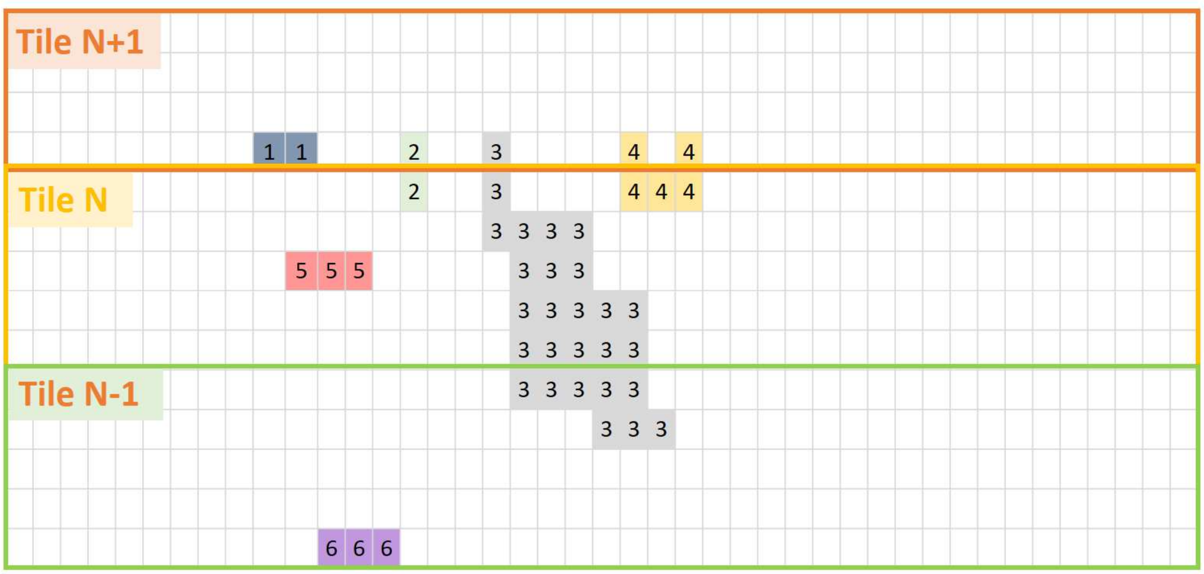
3.7.3.4 Mathematical Statement

This function groups previously obtained labels corresponding to the same waterbody at the swath scale and add LakeTile_Edge labels of features not directly involved into a tile edge. Then, a new label is given for each separate feature.

In the example shown in Figure 30, `gather_regions_at_edge` (Section 3.7.2) has provided matching labels for all along-track tile edges in the continent pass (for one swath): [[*a*], [*b*, *s*], [*c*, ***t***], [*d*, *u*], [***t***, *k*]]. Label *t* belongs to two sub-lists. Labels *r* and *l* are not directly involved in any tile edge. Therefore, the final list becomes [{*a*}, {*b*, *s*}, {*c*, *t*, *k*}, {*d*, *u*}, {*r*}, {*l*}]. Finally, LakeTile label *a* is relabeled *1*, *b* and *s* are relabeled *2*, and so forth, as shown on Figure 30 (b).



(a)



(b)

Figure 30. Illustration of (a) the original LakeTile_Edge labels at the continent-pass scale, for one swath, and (b) their new labels. (Connected groups of pixels are shown with the same colors.)

3.7.3.5 Accuracy

This operation does not introduce additional errors.

3.8 proc_pixc_vec_sp.update_pixcvec

3.8.1 Purpose

The purpose of this function is to update each input L2_HR_LakeTile_PIXCVec file with height-constrained geolocation, and PLD and observation identifiers of the pixels related to water regions crossing the along-track edges of the tiles, to prepare the L2_HR_ PIXCVec standard products.

3.8.2 Input Data

Description	Source
Pixel cloud vector attribute products for rivers, lakes and unassigned features inside each tile of the swath	L2_HR_LakeTile_PIXCVec
Height-constrained longitude, latitude, height, PLD and observation identifiers of each pixel related water regions crossing the along-track edges of the tiles	`proc_pixc.compute_lake_features`

3.8.3 Output Data

Description
Updated longitude, latitude, height, *obs_id* and *lake_id* of each pixel related to these water regions in the L2_HR_ PIXCVec objects.

3.8.4 Mathematical Statement

For the pixels related to water regions crossing the along-track edges of the tiles, the *longitude_vectorproc*, *latitude_vectorproc*, *height_vectorproc*, *obs_id* and *lake_id* variables of L2_HR_PIXCVec objects are updated with the corresponding input values.

3.8.5 Accuracy

This operation does not introduce additional errors.

3.9 lake_db.init_prior_layer

3.9.1 Purpose

The purpose of this function is to initialize the L2_HR_LakeSP_Prior shapefile layer with prior attributes of the PLD lakes located over the continent-pass granule.

3.9.2 Input Data

Description	Source
PLD lake object	`lake_db.PriorLake`
Continent-pass granule	Input parameter

3.9.3 Output Data

Description
L2_HR_LakeSP_layer initialized with the prior attributes of the PLD lakes located over the continent-pass granule

3.9.4 Mathematical Statement

The PLD lake object is reduced to the continent-pass granule, and needed prior attributes are passed to the L2_HR_LakeSP_Prior shapefile layer.

3.9.5 Accuracy

This operation does not introduce additional errors.

3.10 my_shp_file.merge_shp

3.10.1 Purpose

The purpose of this function is to merge L2_HR_LakeTile_[Obs|Prior|Unassigned] and L2_HR_LakeSP_[Obs|Prior|Unassigned]_[R|L] input shapefile layers into a single shapefile layer.

3.10.2 Input Data

Description	Source
L2_HR_LakeTile_[*] shapefile layer	L2_HR_LakeTile product
Both L2_HR_LakeTile_[*]_R and L2_HR_LakeTile_[*]_L shapefile layers	`proc_lake.compute_lake_features`

[*] means one among (Obs, Prior, Unassigned).

3.10.3 Output Data

Description
Shapefile layer containing all features from input shapefile layers

3.10.4 Mathematical Statement

The input shapefile layers are combined using the `gdal.ogr2ogr` library.

3.10.5 Accuracy

This operation does not introduce additional errors.

3.11 locnes_product_shapefile.merge_duplicate_features

3.11.1 Purpose

A PLD lake can be observed in both swaths (L and R), and in rare cases more than once per swath (for example a large PLD lake shaped as the letter “C”, where the middle part is in the left half swath, and the two ends are in the right half swath, but in different tiles in the along-track direction). This initially leads to separate features in the L2_HR_LakeSP_Prior shapefile layer.

The purpose of this function is to combine these features into a single one.

3.11.2 Input Data

Description	Source
L2_HR_LakeSP_Prior shapefile layer	`my_shp_file.merge_shp`

3.11.3 Output Data

Description
Updated input shapefile layer, with duplicate features merged

3.11.4 Mathematical Statement

The first step is to identify the *lake_id* attributes that occur more than once in the input shapefile layer, and retrieve their corresponding internal feature identifier (FID).

Then, the input features related to each identified PLD lake are merged:

- The resulting geometry is the union of the input geometries.
- The output *obs_id* list is the aggregation of the input *obs_id* lists, removing duplicates. The input *overlap* lists are merged in the same way. *n_overlap* is the resulting number of elements in each of these lists.
- The areas (*area_total* and *area_detct* attributes), the storage change attributes (*ds[1|2]_[l|q]* attributes) are the sum of their corresponding input attributes.
- The flags (*quality_f*, *ice_clim_f*, *ice_dyn_f*, *partial_f* attributes) are the maximum of their corresponding input attributes.
- All other attributes (e.g. *wse*) are the weighted average of their corresponding input attributes. The weights w_i are defined as:

$$w_i = \frac{area\_total_i}{\sum_i area\_total_i}$$

3.11.5 Accuracy

This operation does not introduce additional errors.

4 Accuracy of L2_HR_LakeTile/SP Algorithms

This section summarizes the overall accuracy of the L2_HR_LakeTile and L2_HR_LakeSP algorithms.

Table 10 describes the performance statistics at the lake level, for all PLD lakes in the simulated representative dataset to which the science requirements [4] are applicable. We refer to Appendix B for further information on the representative dataset and how it is used for L2_HR_LakeTile/SP performance assessment, by comparing nominal products to the so-called "truth" (reference) products. Note that only lakes within the nominal swath (10-60 km) and larger than 250×250 m^2 are considered with respect to the science requirements, and that completely unobserved lakes are not taken into account.

Table 10. Summary statistics for the L2_HR_LakeSP lake-level performances based on simulated data from the representative dataset.

Metric	Lake size	|68%ile|	(sci. req.)	50%ile	Lake count
area_total (%)	>250×250 m^2	20.6	(<15)	11.9	8783
area_detct (%)	>250×250 m^2	16.7	(NA)	10.2	8783
WSE (m)	>250×250 m^2, <1 km^2	0.066	(<0.25)	0.041	8239
WSE (m)	>1 km^2	0.067	(<0.10)	0.042	544

We see from Table 10 that the measured 1σ (or |68 percentile|) relative water surface area error (based on the *area_total* attribute) is not within the science requirements (20.6% > 15%), whereas the 1σ WSE error is well within the science requirements (6.6 cm < 25 cm and 6.7 cm < 10 cm, respectively).

4.1 Water Surface Area

Figure 31 shows the 1σ |relative error| in total water surface area for several lake size categories (see Appendix B for further details on how it is computed). While lakes larger than 1 km^2 are within the requirement (<15%), smaller lakes have errors above this limit, especially those between 250×250 m^2 and 500×500 m^2. The result of this, together with the fact that the smaller lakes are more numerous (see Figure 38 in Appendix B), is that the overall 1σ |relative error| in total surface area reported in Table 10 (20.6%) is above the requirement. Although there is no science requirement attached to it, we also computed the 1σ |relative error| for the detected water surface area (i.e. before adding surfaces flagged as dark water, both in the nominal and the truth water mask). The error is significantly lower (16.7%), though still above the requirement.

Figure 32 represents as a dot plot the individual relative errors in total area as a function of cross-track distance, and lake size category (different colors as indicated in the legend). While these signed errors have a lower bound (-100%), there is no upper bound, so we have clipped values above 200% to make the graph more readable. As expected, larger lakes generally seem to have smaller relative area errors than smaller ones, but all size categories have strong outliers. The dependency of the errors on the cross-track distance is not easy to see in this graph.

Figure 33 shows the median (50 percentile) value of the relative errors in total area for cross-track distances ranging from 10 to 60 km (nominal swath). Here we see a clearly falling trend with distance. All the median values are positive, which probably reflects the fact that there is a lower bound on the relative error (lake not detected), but no upper bound (the observed, i.e. detected and dark-water-flagged area may become much bigger than the actual lake).

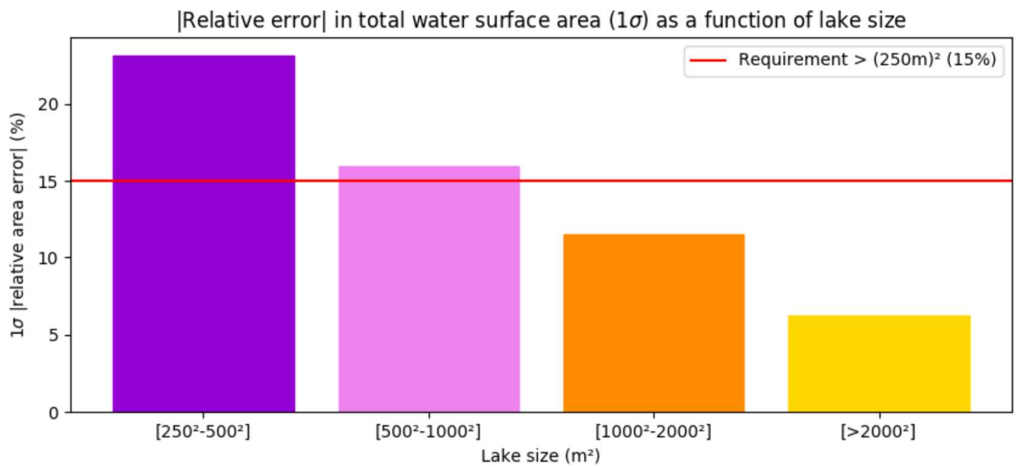


Figure 31. |Relative error| in total surface area (1σ) as a function of lake size.

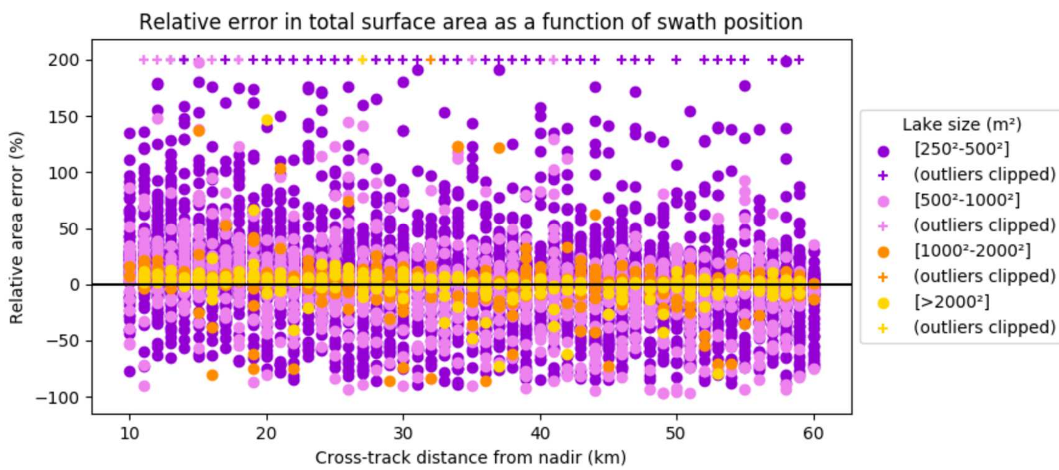


Figure 32. Relative error in total surface area as a function of cross-track distance from nadir, and for different lake size categories.

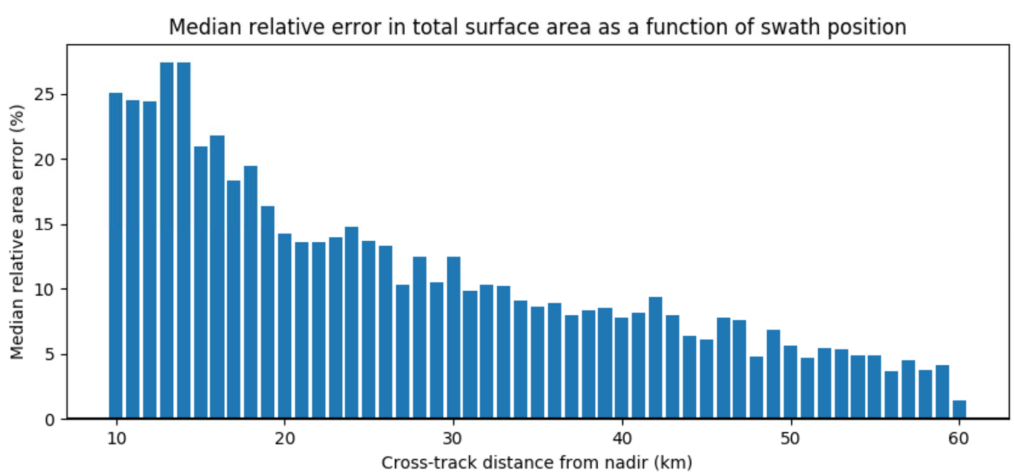


Figure 33. Median relative error in total surface area as a function of cross-track distance from nadir.

A comprehensive discussion of the area errors at the L2_HR_PIXC product level are given in [14]. Important error sources are related to the water detection accuracy (risk of false detection and missed detection) and to the uncertainty of the estimated water fraction for water/land edge pixels. The water detection has a regularization term that reduces the risk of missed detection within the water bodies, but that may also smoothen (over-regularize) the water body edges. Small lakes have a larger proportion of edge pixels and are therefore more exposed to errors in both water detection and water fraction estimation.

When part of a waterbody cannot be detected because of so-called “dark water” (very weak backscattering of the water surfaces at winds speeds below ~2 m/s), they are tentatively completed through dark water flagging based on a prior water occurrence map [14]. However, the dark water model used in the current version of the simulated data makes dark water more probable close to the boundaries of the lakes than in the middle of it, and this is something that makes it more difficult to flag dark water successfully. The fact that the relative error of the detected water surfaces is lower than for the total surfaces, indicates that the dark water flagging is part of the problem.

Assignment errors are an even more important error source when going from the pixel level to the lake level: these can be related to inaccuracies in the PLD (lake polygons, influence area polygons), but also in the PRD. In particular missing river reaches in the PRD can cause over-attribution of pixels to a spatially connected lake.

4.2 Water Surface Elevation (WSE)

Figure 34 shows the 1σ |error| in average WSE for several lake size categories. As indicated in Table 10, the results are well within the requirements both for lakes <1 km^2 and lakes >1 km^2, but while the expected trend is lower error as the lake size increases, Figure 34 reveals an anomaly for lakes >4 km^2, for which the 1σ WSE |error| is bigger than for smaller lakes, and is slightly above the requirement (< 10cm).

The dot plot of the individual signed lake WSE errors in Figure 35 has been clipped to ±1 m for improved readability. Larger lakes generally have smaller WSE errors than smaller ones, as expected, but all size categories have strong outliers. This graph does not reveal any strong dependency of the errors on the cross-track distance.

Figure 36 displays the median (50 percentile) value of the WSE errors for cross-track distances ranging from 10 to 60 km (nominal swath). The median errors are mm- or cm-scale, and there are both negative and positive values. There is no strong trend with distance in the first half of the swath, but then the median errors increase towards far range (60 km).

The WSE error sources inherits those of the water surface area described above, except that pixels flagged as dark water are not used to compute WSE. There are also specific errors related to phase unwrapping and the computation of geolocated heights (depending on the accuracy of the reference DEM, random phase noise, impact of layover…).

In addition, assignment issues play an important role in WSE errors, in combination with other error sources. An example is shown in Figure 37, where the river reach downstream of a PLD lake is not present in the PRD. As the truth (reference) water mask has a higher resolution, the connectivity between the lake and the river reach is preserved, and as the river reach has not

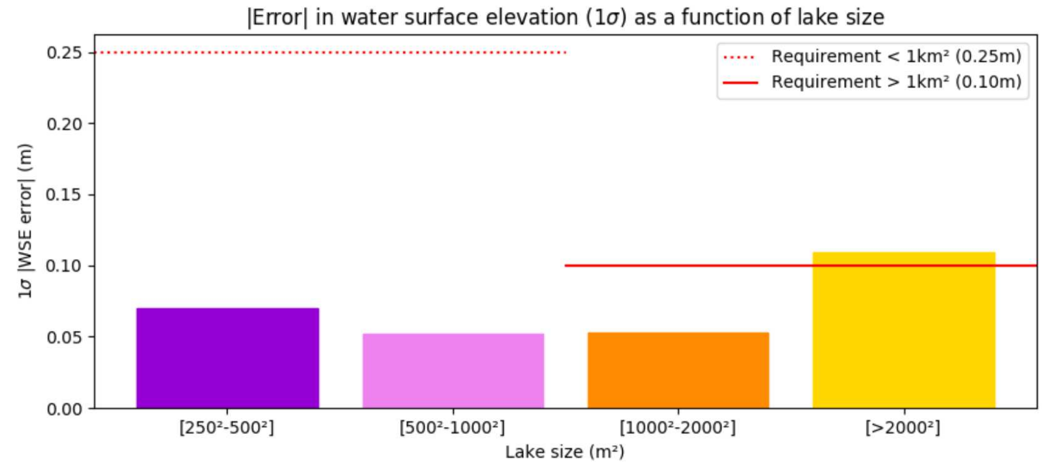


Figure 34. |WSE error| (1σ) as a function of lake size.

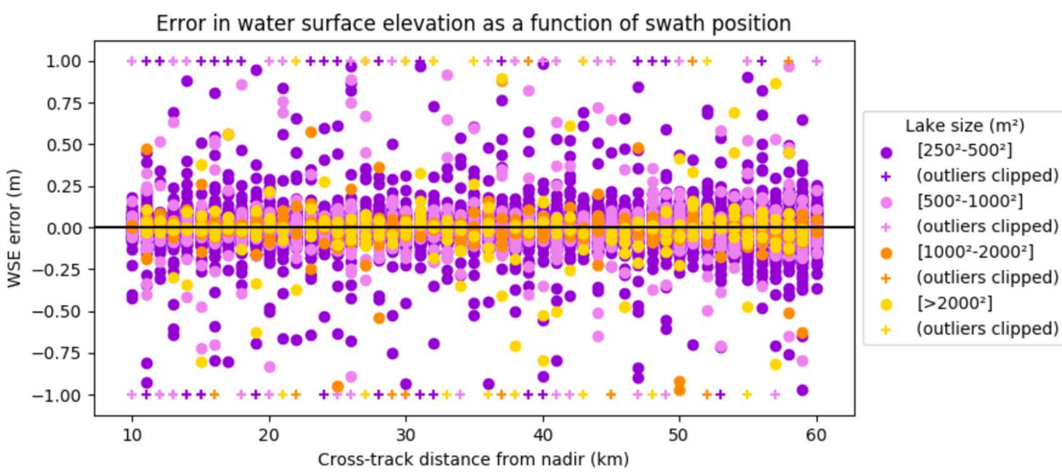


Figure 35. WSE error as a function of cross-track distance from nadir, and for different lake size categories.

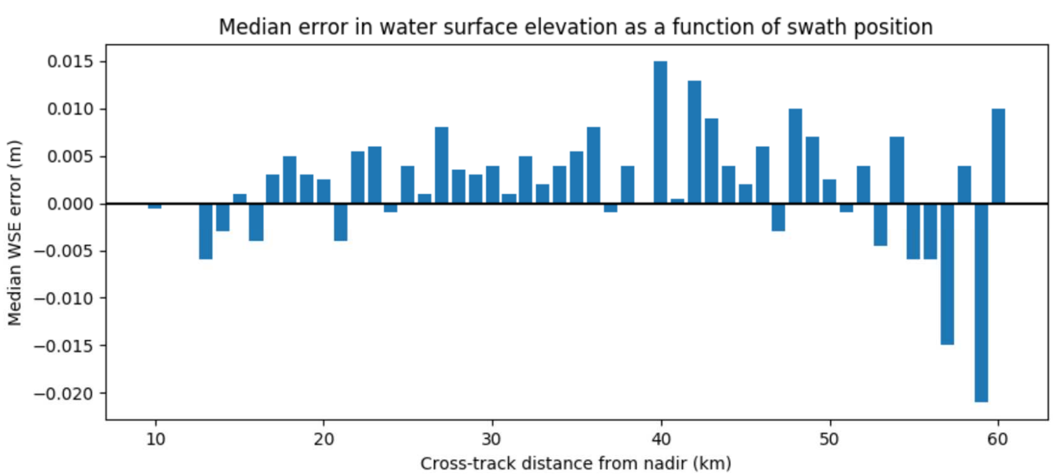


Figure 36. Median WSE error as a function of cross-track distance from nadir.

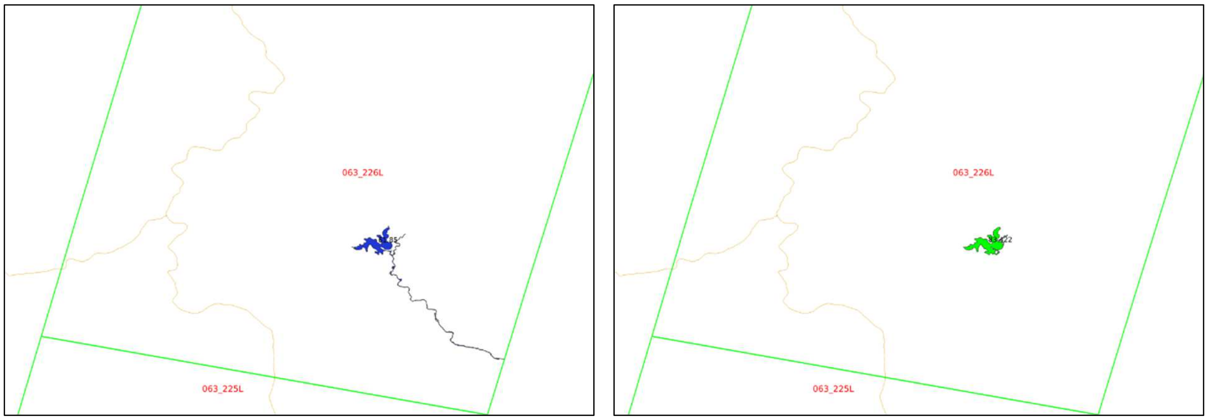


Figure 37. Example of assignment differences between the truth L2_HR_LakeSP product (left) and the nominal L2_HR_LakeSP product (right). The underlying L2_HR_LakeTile tile numbers are indicated in red. A lake and a connected river reach (not present in the PRD) are assigned to a PLD lake (lake_id 7310042752) in the truth product, whereas only the observed lake is associated with the PLD lake in the nominal product (river not detected in the simulated SWOT image).

been assigned to a PRD reach through the river truth processing, it is assigned to the PLD lake (together with the actual lake). In the nominal product, however, as the resolution is coarser and the observability of the river weaker, the lake is not connected to the river reach (in radar geometry), so only the lake itself is assigned to the PLD lake. In this particular case, the impact on the relative water surface area is relatively moderate (-11.4%), whereas the WSE error is large (21.5 m) compared to the science requirements. In other cases, we inversely have large area errors and minor WSE errors.

In the case described above, the assignment issue only concerns the truth water mask, but in many cases it also concerns the water mask of the nominal product, and the general problem is that there may be assignment differences between the two masks that increase the measured errors.

As explained in Appendix B, it is not practically feasible to filter out all such spurious cases based on visual inspection, because of the large number of simulated lakes. Improvements in the PRD and PLD throughout the SWOT mission will reduce the assignment errors and thereby also the water surface area and WSE errors.

5 References

[1] C. Pottier, "Product Description Document - Level 2 KaRIn high rate lake single pass vector product (L2_HR_LakeSP) - SWOT-TN-CDM-0673-CNES," CNES, 2023.

[2] C. Pottier, "Product Description Document - Level 2 KaRIn high rate pixel cloud vector product (L2_HR_PIXCVec) - SWOT-TN-CDM-0677-CNES," CNES, 2023.

[3] C. Pottier, "Product Description Document - Level 2 KaRIn high rate lake tile vector product (L2_HR_LakeTile) - SWOT-TN-CDM-0672-CNES," CNES, 2023.

[4] S. Desai, "SWOT Science Requirements Document, JPL D-61923," Jet Propulsion Laboratory, Pasadena, 2018.

[5] X. Wu, "Product Description Document - Level 1B KaRIn high rate single look complex data product (L1B_HR_SLC) - D-56410," JPL, 2023.

[6] B. Williams, "Product Description Document - Level 2 KaRIn High Rate Water Mask Pixel Cloud Product (L2_HR_PIXC) - D-56411," JPL, 2023.

[7] C. Stuurman, "Product Description Document - Level 2 KaRIn high rate river single pass vector product (L2_HR_RiverSP) - D-56413," JPL, 2023.

[8] C. Stuurman, "Product Description Document - Level 2 KaRIn high rate river average vector product (L2_HR_RiverAvg) - D-56414," JPL, 2023.

[9] C. Pottier, "Product Description Document - Level 2 KaRIn high rate lake average vector product (L2_HR_LakeAvg) - SWOT-TN-CDM-0676-CNES," CNES, 2023.

[10] A. Corben, "Product Description Document: L2_Raster, JPL D-56416," Jet Propulsion Laboratory, 2023.

[11] C. Stuurman, "Product Description Document - Level 2 KaRIn high rate pixel cloud vector attribute river product (L2_HR_PIXCVecRiver) - D-56415," JPL, 2023.

[12] C. W. Chen, "SWOT Project Science Data Product Granule Boundary and Sampling Definition," Jet Propulsion Laboratory, D-102104, 2022.

[13] E. H. Altenau, T. M. Pavelsky, M. T. Durand and R. Frasson, "SWOT Prior River Database (SWORD)," 2019.

[14] B. Williams, "Algorithm Theoretical Basis Document: L2_HR_PIXC Level 2 Processing," JPL D-105504, 2023.

[15] C. Pottier, "Auxiliary Data Description - Prior Lake Database (PLD)," CNES, 2023.

[16] D. Desroches, "Height-constrained geolocation," Technical note, DTN/TPI/TR-2022/00500, CNES, 2022.

[17] CNES, "SWOT Large Scale Simulator," [Online]. Available: https://github.com/CNES/swot-hydrology-toolbox.

[18] C. Pottier, "Auxiliary Data Description Document - L2_HR_LakeTile auxiliary parameter file (Param_L2_HR_LakeTile)," CNES, 2023.

[19] N. Otsu , "A threshold selection method from gray-level histograms," *IEEE Trans. Sys. Man. Cyber.,* vol. 9, no. 1, p. 62–66, 1979.

[20] C. Stuurman, "Algorithm Theoretical Basis Document (ATBD) - Level 2 KaRIn High Rate River Single Pass Science Algorithm Software (SAS_L2_HR_RiverSP)," JPL, 2023.

[21] B. Williams, "SWOT Hydrology Height and Area Uncertainty Estimation," JPL Project Document, 2018.

[22] J.-F. Cretaux, R. A. D. Rio, M. Berge-Nguyen, A. Arsen, V. Drolon, G. Clos and p. Maisongrande, "Lake volume monitoring from Space,," *Survey in Geophysics,* vol. 37, no. doi 10.1007/s10712-016-9362-6, pp. 269-305, 2016.

[23] C. Pottier, "SAS Design Document - Level 2 high rate lake tile science algorithm software

(SAS_L2_HR_LakeTile) - SWOT-DD-CDM-0561-CNES," CNES, 2022 .

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**

The performance assessment of L2_HR_LakeSP products is based on a representative simulated dataset of L2_HR_SLC products, processed to L2_HR_PIXC products as described in [14] (Appendix B), thereafter to L2_HR_RiverTile and L2_HR_RiverSP products as described in [20] (Appendix B), and eventually to L2_HR_LakeTile and L2_HR_LakeSP products as detailed below.

The representative dataset contains a total of 63 unique simulated scenes in the United States, Canada and France, yielding 598 scene-pass-tile combinations in total (~64×64 km^2 tiles). When intersecting these simulated data with the PLD [15], the representative dataset contains 11606 unique lakes and 21521 lake-passes in total. Of these, there are 4027 unique lakes and 8783 lake-passes that meet filtering criteria based on the applicability of the science requirements for SWOT [4]:

- Simulated lakes must be located between 10 km and 60 km cross-track. That is, if a portion of the lake is outside these limits, this part is not considered in the performance statistics. Lakes outside this nominal swath are excluded from the very beginning in the above figures.
- They must meet a minimum area of 250×250 m^2 to be included in the lake performance statistics. This criterion discards 9552 of the 21521 lake-passes within the nominal swath (note that detected lakes smaller than 100×100 m^2 are excluded from the very beginning). Figure 38 illustrates how the number of lakes continue to diminish with increasing lake size.
- Lakes that are not observed at all are let out of the statistics. This criterion eliminates 4092 PLD lake-passes, of which 769 lakes were bigger than 250×250 m^2.

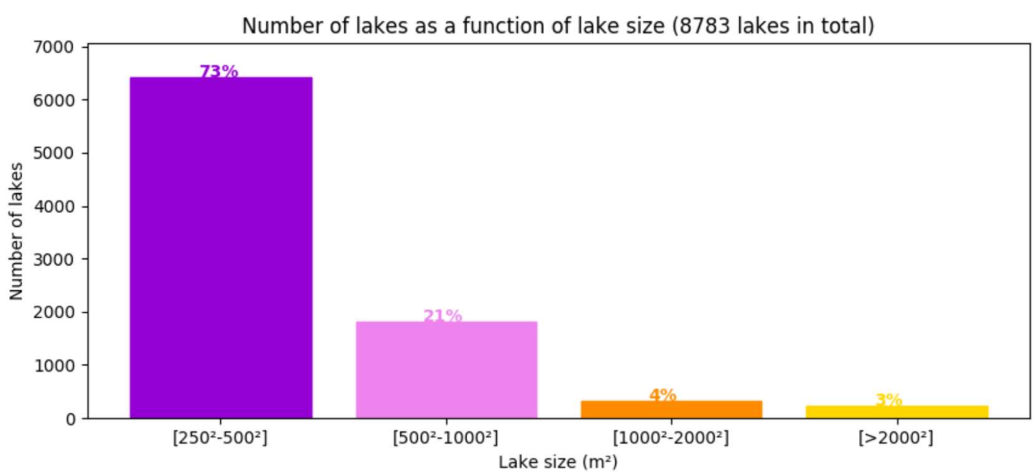


Figure 38 Number of PLD lakes (lake-passes) in the representative dataset as a function of the lake size (only lakes larger than 250×250 m^2 are shown here).

Performance assessment based on simulated SWOT data require both “truth” and nominal processed data. River and lake “truth” data were generated by evenly distributing (resampling) water observation pixels over the truth water masks and assigning WSEs to each pixel from the truth heights (based on airborne lidar data) used as inputs to the simulation, in order to form an artificial L2_HR_PIXC “truth” product [14]. Directly mapping “truth” heights to pixel heights eliminates sources of error due to LR_HR_PIXC or L1_HR_SLC processing. These artificial L2_HR_PIXC products are then processed through L2_HR_RiverTile processing to create "truth" L2_HR_RiverTile products [20], which allows us to exclude pixels assigned to river reaches (except connected lakes) from the subsequent L2_HR_LakeTile and L2_HR_LakeSP processing used to generate L2_HR_LakeTile and L2_HR_LakeSP "truth" products.

While there may be discrepancies between the simulation inputs (and thus the "truth" products) and the PRD and PLD, and also non-physical artifacts or inaccuracies in the "truth" data themselves, it has not been judged practically feasible to visually inspect the 4027 unique PLD lakes (8783 lake-passes) covered by the dataset (after filtering), to eliminate such spurious cases. The result is an increase in the measured errors as discussed in section 4.

The two main error metrics for lakes with respect to the science requirements [4] are computed as follows:

- The 1σ |relative error| in total surface area is computed by first subtracting the reference total area from the observed total area of the individual PLD lakes, based on the "truth" and nominal L2_HR_LakeSP products, respectively, then dividing by the reference total area and taking the absolute value of the result, and finally computing the 1σ value (68 percentile), either globally or separately for different lake size intervals.
- The 1σ |WSE error| is computed in a similar manner, except that it is simply based on the difference between the SWOT-observed WSE and the reference WSE.

Signed individual errors and their median values are also computed and displayed in section 4.