NISAR L1 and L2 Algorithm Theoretical Basis Document

Revision A

JPL D-95677

Nov 12, 2023

Paper copies of this document may not be current and should not be relied on for official purposes. The current version is in TeamCenter (EPDM): https://epdm.jpl.nasa.gov

Authors: NISAR Algorithms Development Team

Jet Propulsion Laboratory
California Institute of Technology

SIGNATURE PAGE

Prepared by:

Electronic Signature on File *01-Dec-2023*
Heresh Fattahi, NISAR Algorithms Development Lead Date

Electronic Signature on File *30-Nov-2023*
Brian Hawkins, NISAR L1 Lead Date

Electronic Signature on File *30-Nov-2023*
Virginia Brancato, NISAR L2 Interferometry Lead Date

Email Approval on File *30-Nov-2023*
Gustavo H. X. Shiroma, NISAR L2 GCOV Lead Date

Email Approval on File *05-Dec-2023*
Bo Huang, NISAR ADT Algorithm Engineer Date

Electronic Signature on File *04-Dec-2023*
Geoff Gunter, NISAR ADT Algorithm Engineer Date

Email Approval on File *05-Dec-2023*
Jungkyo Jung, NISAR ADT Algorithm Engineer Date

Approved by:

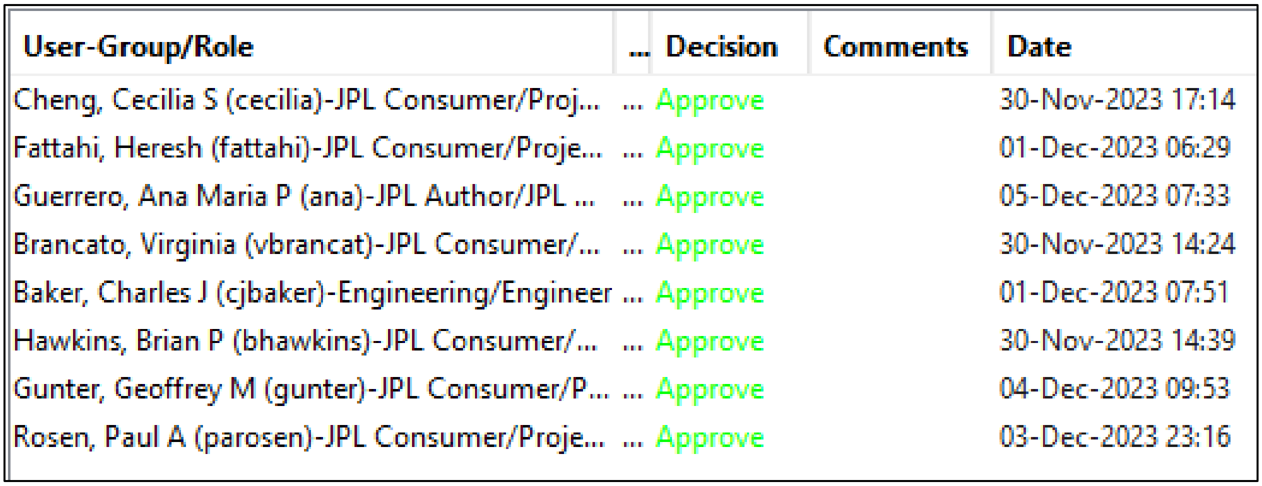
Electronic Signature on File *05-Dec-2023*
Ana Maria Guerrero, NISAR Mission System Manager Date

Electronic Signature on File *03-Dec-2023*
Paul Rosen, NISAR Project Scientist Date

Electronic Signature on File *30-Nov-2023*
Cecilia Cheng, NISAR SDS Manager Date

Electronic Signature on File *01-Dec-2023*
Chuck Baker, NISAR Mission System Engineer Date

EPDM Electronic Signatures



User-Group/Role	...	Decision	Comments	Date
Cheng, Cecilia S (cecilia)-JPL Consumer/Proj...	...	Approve		30-Nov-2023 17:14
Fattahi, Heresh (fattahi)-JPL Consumer/Proje...	...	Approve		01-Dec-2023 06:29
Guerrero, Ana Maria P (ana)-JPL Author/JPL ...	...	Approve		05-Dec-2023 07:33
Brancato, Virginia (vbrancat)-JPL Consumer/...	...	Approve		30-Nov-2023 14:24
Baker, Charles J (cjbaker)-Engineering/Engineer	...	Approve		01-Dec-2023 07:51
Hawkins, Brian P (bhawkins)-JPL Consumer/...	...	Approve		30-Nov-2023 14:39
Gunter, Geoffrey M (gunter)-JPL Consumer/P...	...	Approve		04-Dec-2023 09:53
Rosen, Paul A (parosen)-JPL Consumer/Proje...	...	Approve		03-Dec-2023 23:16

Document Change Log

Revision	Cover Date	Sections Changed	ECR #	Reason, ECR Title, LRS #*
Outline	May 9, 2016	All	N/A	New document
Draft	Sep-Dec 2016	All	N/A	Draft of separate ATBD documents
Preliminary	Dec 2016-Jan 2017	All	N/A	Merged into one document and revised LRR 028264
Initial Release	Sep 2019	All	N/A	Updated after Phase C design work LRR 044839 for NISAR ISRO LRR 044840 for NISAR JPL Sharma LRR 044841 for NISAR ADT Brancato & Shiroma
R3	Aug 23 2021	Sec. 2, Sec. 6, Sec 7	N/A	Removed references to RCOV products and updated description of the GCOV workflow
R3.1	July 25 2022	Sec. 2, Sec.4, Sec. 6	N/A	Updated document to include offset products. Updated InSAR workflow steps and description Updated Sec. 6 cross-references, Added RFI algorithm
Rev A	Nov 12 2023			Clearance number: CL#23-6412

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

This document has been reviewed and determined not to contain export controlled technical data

1 INTRODUCTION

1.1 Purpose of Document

This Algorithm Theoretical Basis Document (ATBD) provides the theoretical basis for the algorithms used to create the NASA-ISRO Synthetic Aperture Radar (NISAR) L-SAR Level 1 and Level 2 science products. The Algorithm Development Team (ADT) implements these algorithms in the Science Algorithm Software (SAS) and provides the SAS to the NASA Science Data System (SDS) for generation of the products.

1.2 Scope of the Document

This ATBD provides details of the algorithms used to transform the L-SAR L0B data to Level 1 and Level 2 products. This includes an overview of the contents of the products [RD1] and explanation of the theoretical basis behind the choice of the algorithms and their implementations. The NASA Project Science and Science Team Level 3+ product algorithms [RD2-RD8] as well as the ISRO S-SAR science product algorithms are out of scope of this document.

The calibration and validation of the L-SAR Level-1 and Level-2 science products are discussed in the SDS ADT Calibration and Validation Plan [AD2].

1.3 Applicable and Reference Documents

Applicable documents levy requirements on areas addressed in this document. Reference documents provide additional information to readers. In case of conflict between the applicable documents and this document, the Project shall review the conflict to find the most effective resolution.

NISAR-generated documents are listed here and a list of all other reference material is provided at the end of the document.

Applicable Documents

- AD1: NISAR Calibration and Validation Plan, JPL D-80829, September 6, 2018.
- AD2: NISAR SDS ADT Calibration and Validation Plan, JPL D-102256, September, 2019.
- AD3: NISAR Science Data Management and Archive Plan, JPL D-80828, June 1, 2016.
- AD4: NISAR Science Management Plan, JPL D-76340, May 29, 2016.
- AD5: NISAR NASA SDS Software Management Plan, JPL D-95656, January 2017.
- AD6: NISAR Project Coordinate Systems Definition, JPL D-80882, April 2018.
- AD7: DSI SweepSAR On-Board DSP Algorithms Description, JPL D-95646, April 2018.

- AD8: NISAR SDS ADT Cal/Val Plan, JPL D-102256, September 2019.

Reference Documents

- RD1: NISAR NASA SDS Product Description, JPL D-95672, September, 2019.
- RD2: Solid Earth Algorithm Theoretical Basis Document, April 14, 2016.
- RD3: Ice Sheet Algorithm Theoretical Basis Document, April 14, 2016.
- RD4: Sea Ice Algorithm Theoretical Basis Document.
- RD5: Biomass Algorithm Theoretical Basis Document, April 14, 2016.
- RD6: Disturbance Algorithm Theoretical Basis Document, April 14, 2016.
- RD7: Crop Area Algorithm Theoretical Basis Document, April 14, 2016.
- RD8: Inundation Algorithm Theoretical Basis Document, April 14, 2016.
- RD9: NISAR NASA SDS Product Specifications, JPL D-102266 – D-102274, D-105009 – D-105010, March, 2022.

1.4 Organization of this Document

Section 2 provides common background material. Sections 3-6 discuss the range-Doppler single look complex product, geocoded single look complex product, geocoded covariance product and the interferometry products including wrapped and unwrapped interferograms and pixel offsets.

2 BACKGROUND

2.1 Product Overview

The NASA SDS generates the L-SAR L0-L2 science products (Table 2-1 and Figure 2.1). The Range Doppler Single Look Complex (RSLC) is derived from L0B raw data. RSLC is the fundamental product which flows to all other L1 and L2 products. Geocoded SLC (GSLC) product, is single look complex data in a geocoded grid. The GSLC product enables users to perform backscatter amplitude as well as interferometric analysis directly on a geocoded grid.

The Geocode Covariance (GCOV) product provides radiometrically terrain corrected SAR backscatter data to primarily support the NISAR ecosystem requirements of biomass estimation, soil moisture estimation, disturbance detection, inundation mapping, and crop area delineation, as well as additional ecosystem and land-cover applications that may be developed during the NISAR mission.

The NISAR interferometry products include wrapped interferogram in Range-Doppler coordinates (RIFG), the unwrapped interferogram in Range-Doppler coordinates (RUNW), the unwrapped interferogram in geocoded map coordinates (GUNW), the dense pixel offsets in Range-Doppler coordinates (ROFF), and dense pixel offsets in geocoded map coordinates (GOFF). The interferometry products primarily support the solid earth and cryosphere requirements of ground surface displacement measurements and glacier movements.

Table 2.1. NISAR L-SAR L0-L2 products

Product	Level	Description
Radar Raw Science Telemetry (L0A) (RRST)	L0A	This L0A product is the raw downlinked data delivered to SDS. Communication wrapping has been removed
Radar Raw Signal Data (L0B) (RRSD)	L0B	This L0B product is corrected, aligned radar pulse data derived from the RRST products and used for further processing.
Range Doppler Single Look Complex (RSLC)	L1	Focused SAR Imagery in range-Doppler coordinates.
Range Doppler Interferogram (RIFG)	L1	Multi-looked flattened (ellipsoid) interferogram in range-Doppler coordinates with no removal of topographic fringes. Formed using high-res offsets.
Range Doppler Unwrapped Interferogram (RUNW)	L1	Multi-looked unwrapped differential interferogram in range-Doppler coordinates with topo fringes removed.
Range Doppler pixel Offsets (ROFF)	L1	Raw pixel offsets layers in range and azimuth directions derived by speckle tracking with different resolutions (e.g., different chip and search window size) in range-Doppler coordinates.
Geocoded Single Look Complex (GSLC)	L2	Geocoded SLC product using the MOE state vectors and a DEM.
Geocoded Unwrapped Interferogram (GUNW)	L2	Geocoded, multi-looked unwrapped differential Interferogram.
Geocoded Polarimetric Covariance (GCOV)	L2	Geocoded, multi-looked polarimetric covariance matrix.
Geocoded pixel Offsets (GOFF)	L2	Raw pixel offsets in range and azimuth directions derived by speckle tracking with different resolutions (e.g., different chip and search window size) in geocoded coordinates.

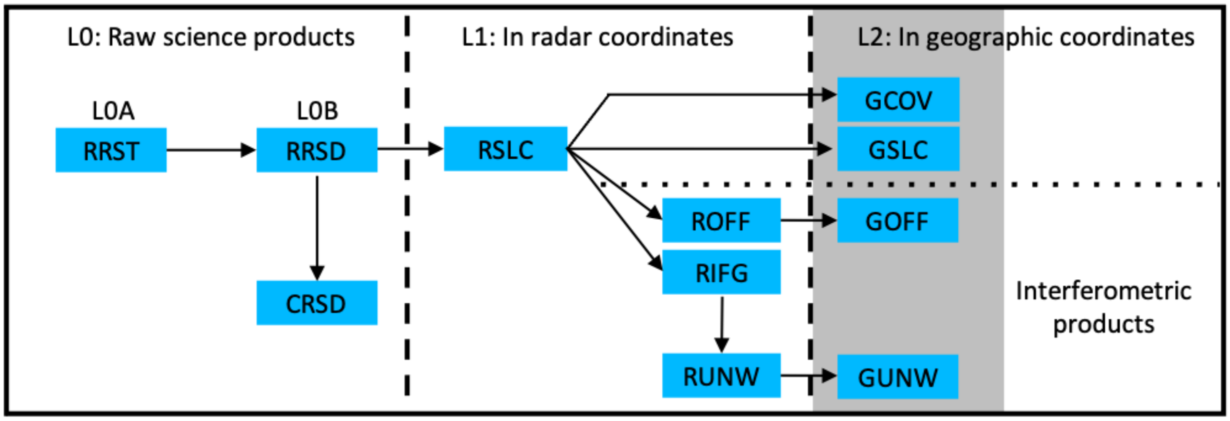


Figure 2.1. Product dependency

2.2 Product Format Overview

The product specification documents [RD9] defines and organizes the detailed content of the NASA SDS L-SAR science products. The products consist of two major components: science data and metadata.

The science data and metadata are stored based on the HDF-EOS5 specification [Klein and Taaheri, 2016], an HDF5-based format that has the following advantages:

- open, self-describing format
- supports hierarchical tree data arrangement
- supported by GIS and database software
- provides flexibility to support any binary data format making it scalable to support all levels of NISAR products

2.3 Data Product Characteristics

2.3.1 Range Doppler Single Look Complex Product

The RSLC product is the range-Doppler geometry SAR image that is operationally delivered with a global coverage. The RSLC product is distributed in the zero-Doppler radar geometry coordinates. The RSLC processor handles SweepSAR's PRF changes within a data granule and the output imagery is on a grid characterized by constant azimuth time interval and one-way slant range spacing. The output grid is characterized by a fixed set of starting slant range, zero-Doppler start time, slant range spacing and azimuth time interval values. Most NISAR science acquisition modes acquire the raw data in two frequency bands referred to as the main band and side bands at two different center frequencies. All the main band image layers for a multi-polarization or multi-frequency product are generated on a common azimuth-time slant-range grid.

2.3.2 Geocoded SLC Product

The GSLC product is derived from the RSLC product by projecting (i.e., interpolating) the RSLC data layers to a geocoded grid. The spacing of the GSLC product in East and North directions is comparable to the full resolution original RSLC product. The GSLC product can be directly overlaid on a map or combined with other similar GSLC products to derive interferograms and change maps.

The GSLC product contains individual binary raster layers representing complex signal return for each polarization layer. GSLC data corresponding to the auxiliary 5 MHz sub-band are also stored in a similar format but in a separate data group within the HDF-EOS5 product granule.

The GSLC product granule contains metadata similar to the source RSLC product but with the lookup tables referenced to geographic coordinates instead of radar coordinates.

The GSLC phase is flattened, i.e. the interferometric phase representing the geometrical range between the radar antenna phase center and the target on the ground is computed using a DEM and satellite state vector and removed from the GSLC phase. Accordingly, the interferograms that users will generate from two GSLCs will be flattened.

2.3.3 Interferometry Products

The range-Doppler pixel Offset (ROFF) product represents a collection of dense pixel offsets layers obtained from applying speckle tracking [Michel, 1999] to a pair of coarsely coregistered RSLCs in range-Doppler geometry of the earlier (i.e., reference) RSLC product. Each offset layer is obtained at a different resolution (i.e., different chip sizes, search windows) but all the pixel offsets layers share the same grid i.e., all the layers have the same spacing and starting pixel in the slant range and azimuth directions. Pixel offsets layers contained in ROFF are distributed as raw i.e., without applying any post-processing operation (e.g., offset culling, low-pass filtering to improve signal-to-noise ratio). The ROFF product is primarily meant for Cryosphere applications and is only generated for acquisitions north of 60 degrees North or south 60 degrees South latitudes, and pre-identified mountain glaciers.

The interferogram (RIFG) product represents the ellipsoid height-corrected, wrapped interferogram generated from two RSLCs in the range-Doppler geometry of the earlier acquisition. The RIFG product is primarily meant for detecting grounding lines and is only generated for acquisitions north of 60 degrees North or south of 60 degrees South latitudes. WGS84 ellipsoid is used as the reference surface for flat earth correction and the products are multi-looked to a posting of 30 meters on the ground. The RIFG product contains individual binary raster layers representing the non-flattened complex interferogram for each co-pol channel and the normalized interferometric correlation. The RIFG product also contains the layer of slant range and azimuth dense pixel offsets used to generate the wrapped interferogram. For frames for which ROFF product is available, the slant range and azimuth layers included in the RIFG product are obtained by optimally blending the pixel offset layers at different resolutions. If the ROFF product is not available for the processed frame, the slant range and azimuth pixel offset layers included in the RIFG product are obtained from running the speckle tracking algorithm with a single set of parameters (e.g., reference window size and search window size). In addition to the metadata of the original RSLC granules, lookup tables for the perpendicular and parallel baseline components are included.

The RUNW product represents the unwrapped, multi-looked differential interferogram generated from two RSLCs in the range-Doppler geometry of the earlier acquisition. For every ingested RSLC product, an archived RSLC product corresponding to the same imaging geometry and nearest in time is identified and an RUNW processing job is launched. The RUNW product is only generated in the latitude range 60 degrees North to 60 degrees South, and between co-pol channels. A DEM is used as the reference elevation model for processing and the products are multi-looked to a posting of 80 meters on the ground.

The RUNW product contains individual binary raster layers representing the single-precision floating point unwrapped phase for each co-pol channel. In addition, the RUNW product includes byte layers with quantized normalized interferometric correlation, geometry masks and connected components information. Like the RIFG product, the RUNW product also includes the slant range and azimuth pixel offsets layers used to generate the wrapped interferogram and lookup tables for the parallel and perpendicular components of the interferometric baseline. The RUNW product also includes layers with an estimate of the ionospheric phase screen and with the quantification of its uncertainty. The GUNW (GOFF) product is derived from the RUNW (ROFF) product by geocoding to a map coordinate system at 80 m posting. All the lookup tables including phase corrections are transformed from image coordinates to geographic coordinates.

2.3.4 Geocoded Covariance (GCOV) Product

The physical quantity encoded in the GCOV product is the square root of γ^0 (terrain-corrected gamma-naught). So, the modulus square of the GCOV product yields γ^0 (the radar cross section normalized to the area of the resolution cell projected on the plane normal to radar line of sight), which is a parameter of interest for ecosystem applications [Small, 2011]. Gamma-naught was chosen for the GCOV product to minimize the variation of the radar backscatter coefficient to changes of local incidence [Ulaby et al, 1986].

2.3.5 Posting

NISAR instrument records the acquisitions from all different radar modes with 1.2x oversampling factor in range direction. For example, the 20 MHz data are sampled at 24 MHz. Similarly, the NISAR RSLC products are oversampled by 1.2x in range direction. The range resolution of the RSLC products is determined by the acquisition mode. Table 2.3.1 lists the slant range resolution and posting for different radar modes. In azimuth direction, the RSLC products are sampled at 1520 Hz which oversamples the data by 1.2x. This specification leads to RSLC azimuth resolution and spacing of ~6m and ~5m respectively.

The GSLC product is posted on an oversampled grid that preserves the high-resolution information from the source RSLC product (Table 2.3-1).

Table 2.3-1. RSLC and GSLC product resolution and posting

Range bandwidth	RSLC Azimuth resolution	RSLC slant range resolution	RSLC azimuth posting	RSLC Slant range posting	GSLC North posting	GSLC East posting
5 MHz	~6 m	30 m	~5 m	25 m	5 m	40 m
20 MHz	~6 m	7.5 m	~5 m	6.25 m	5 m	10 m
40 MHz	~6 m	3.75 m	~5 m	3.12 m	5 m	5 m
80 MHz	~6 m	1.95	~5 m	1.56 m	5 m	2.5 m

The RIFG product is multi-looked to a posting of approximately 30 m and the RUNW products are multi-looked to a posting of approximately 80 m. The GUNW products are posted at 80 m spacing. The ROFF products are multi-looked to a posting of approximately 90 m and GOFF products are posted at 80 m.

The nominal posting of GCOV products is 20 m regardless of the radar mode except for 5 MHz data which will be provided at 100 m posting. The nominal posting of 20 m was chosen because it is a factor of the posting used for the final L3+ ecosystem products, 100 m (1ha). Aggregating 25 adjacent samples of GCOV products gives the desired posting of 100 m in both east and west directions.

2.3.6 Polarization

For multi-polarization imaging modes, each polarization is included in the HDF-EOS5 granule as a separate dataset. The various polarization datasets in the same product are processed with consistent parameters to ensure they are coregistered. All the polarizations of the RSLC product are carried to the level-2 GSLC product.

Interferometric products are generated only for the co-pol channels (HH or VV) of the main band. For quad-pol imaging modes, the interferogram corresponding to each co-pol channel is included in the HDF-EOS5 granule as a separate dataset. The various polarization bands are processed with consistent parameters to ensure they are coregistered.

NISAR has two major acquisition modes for ecosystem science: linear dual-pol (HH/VH) with H-pol transmission and simultaneous H/V-pol coherent reception, and linear quad-pol (HH/HV/VV/VH) with alternating H/V-pol transmission and simultaneous H/V-pol coherent reception. The scattering vector is the set of single-look complex layers obtained in all combinations of transmit/receive polarization. Depending on the mode (dual or quad), the scattering vector has 2 or 4 complex-valued elements. After calibration and symmetrization to force target reciprocity (i.e., HV=VH), the quad-pol scattering vector has 3 complex-valued elements [van Zyl et al, 1990; Cloude and Pottier, 1994].

The polarimetric covariance matrix represents the pair-wise multiplication and multi-looking of the elements of the polarimetric scattering vector. The polarimetric covariance matrix is Hermitian (i.e., it is invariant to transpose-conjugate operation) and positive semi-definite (i.e., it has non-negative eigenvalues) by construction. By virtue of these properties, only the upper triangular part of the matrix is distributed, as the lower triangular part can be obtained after complex-conjugation of the upper triangular part.

The GCOV algorithm is capable of producing 3 layers from dual-pol RSLC data: 2 real-valued layers (the diagonal elements of the matrix) and 1 complex-valued element (the single off-diagonal element of the matrix). The GCOV algorithm is capable of producing 6 layers from the quad-pol RSLC product, of which 3 are real-valued (the diagonal elements of the matrix) and 3 are complex-valued (the off-diagonal elements of the matrix). The product specification clarifies if the product is generated with off-diagonal terms or with only diagonal (real-valued) terms.

The linear lexicographic basis is used to represent the covariance matrices. We choose this basis because it is widely used and it is directly derived from the RSLC. The covariance matrices can be converted and represented in any other basis (e.g., Pauli) using external tools. The characteristics and algorithms of the dual-pol GCOV products also apply to other types of dual-pol modes besides HH/VH (e.g., VV/HV or compact-pol modes with circular-transmit/linear-receive).

2.3.7 Side Bands

Depending on the acquisition mode, the NISAR RSLC, GSLC and GCOV products may include side-band 5 MHz imagery as a separate layer in the HDF-EOS5 granule depending on the acquisition bandwidth. This auxiliary layer is processed with parameters consistent with the main imaging band and output on a grid that has an integer-scaled relationship with that of the main imaging grid. As an example, the range-Doppler grid of a 5 MHz SLC in the 20+5 MHz RSLC product has the same length as the 20 MHz layer in azimuth direction and a width of ¼ of the width of the 20 MHz layer.

3 RANGE DOPPLER SLC ALGORITHMPRODUCT

3.1 Introduction

The primary NISAR L-SAR L1 product is the range-Doppler single-look complex (RSLC) image. The RSLC is used to form the other L1 and L2 products.

The algorithms to generate an RSLC product (Figure 3.1-1) from the L0B RRSD product may be grouped into the following categories [Piantanida, 2016]:

- Pre-processing algorithms
- Doppler centroid estimation algorithms
- Azimuth gap filling & regridding algorithms
- SLC processing algorithms
- SLC post-processing algorithms
- Common and support algorithms

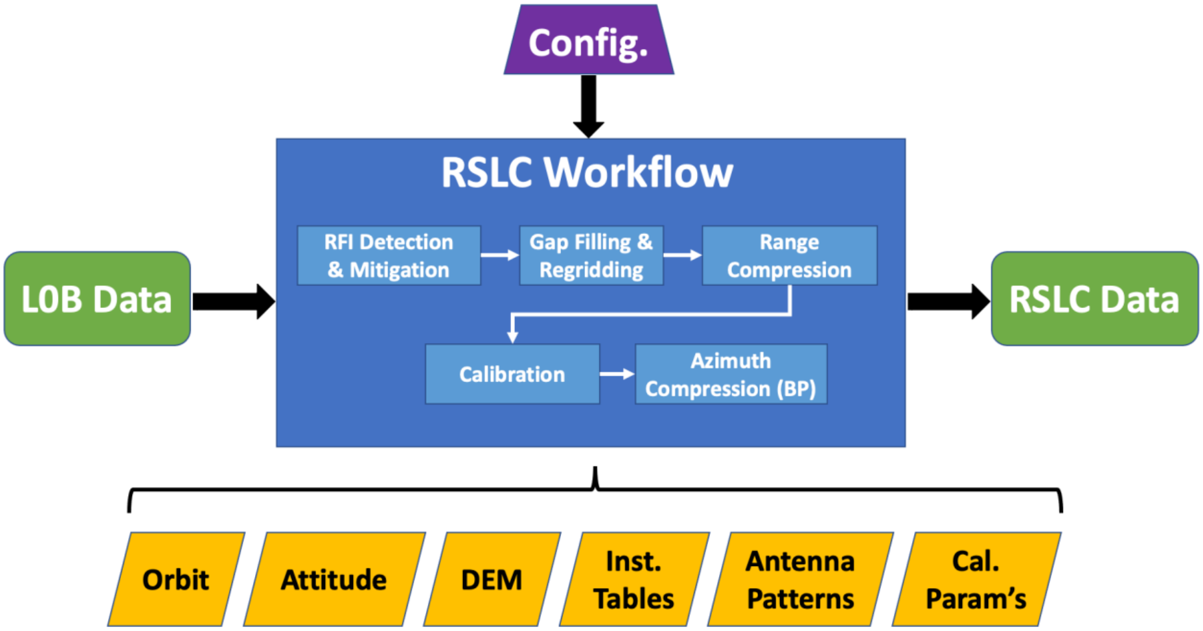


Figure 3.1-1. RSLC product generation workflow

In addition to the raw L0B data, the algorithm requires several ancillary inputs to function properly. These include

- Precise orbit and attitude files.
- Digital elevation model (DEM).
- Instrument tables (time-to-angle and angle-to-coefficient) that define the on-board digital beamforming and are necessary for antenna pattern correction.
- Antenna patterns for each transmit/receive module.
- Calibration parameters: Common and differential range delays, absolute calibration factor, complex channel imbalances, and crosstalk coefficients. The delays are applied during/after range compression and the other parameters comprise the calibration matrix described in section 3.10.

3.2 Raw Data Decoding

The NISAR L-SAR produces 16-bit raw data samples at a sampling rate of 240 MHz, which is further downsampled to 96, 48, 24, or 6 MHz depending on the mode. The data volume is reduced onboard using a block adaptive floating-point quantization (BFPQ) algorithm [Kwok and Johnson, 1989]. This produces 64-sample data blocks of 3/4/5-bit BFPQ samples with a common 5-bit exponent.

The BFPQ decoding is partially done in the NISAR SDS L0A to L0B processing. At this stage the BFPQ exponents are extracted for each block, left-shifted by the mantissa width, and ORed with the mantissa for storage in a 16-bit field in the L0B product. The HDF5 dataset applies a gzip compression filter to transparently eliminate the duplicated information. The L0B to RSLC processor then decodes the data to floating point using a single look-up table. Raw data statistical analysis and mitigation strategies in the SLC processing are used to minimize biases associated with bit errors and RFI [West, 2014].

3.3 Radio Frequency Interference (RFI)

The Radio Frequency Interference (RFI) is defined as the undesired signal in the form of wideband or narrowband interference that overlaps with the in-band frequency spectrum of the NASA-ISRO SAR (NISAR). The main sources of high-power RFI are electromagnetic signals from civilian and military ground-based communication or radar platforms. RFI may severely degrade SAR image quality. Furthermore, RFI may degrade the correlation of the interferometric products. The proposed top-level NISAR RFI mitigation process flow is shown in Figure 3.2-1 below. RFI processing is a part of the overall RSLC workflow. RFI detection is performed on L0B raw data before focusing. If significant RFI is detected in the L0B raw data, RFI mitigation is applied on the data. NISAR RFI detection and mitigation algorithms are based on the principal component analysis approach which utilizes Eigenvalue Decomposition (EVD).

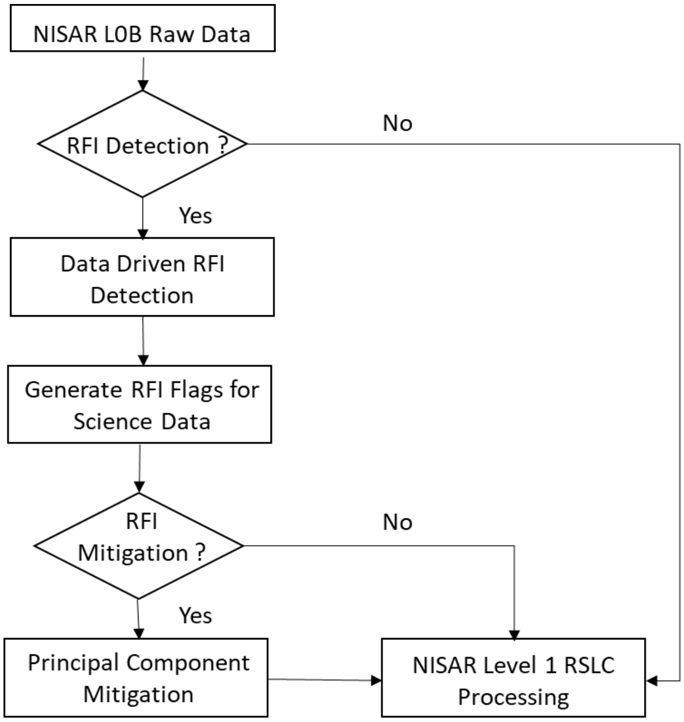

NISAR L0B Raw Data
RFI Detection ?
No
Yes
Data Driven RFI Detection
Generate RFI Flags for Science Data
RFI Mitigation ?
No
Yes
Principal Component Mitigation
NISAR Level 1 RSLC Processing

Figure 3.2-1. RFI Processing Workflow

3.3.1 Principal Component Approach

We assume that RFI has higher power than the signal of interest. We apply the Principal Component Analysis approach to separate RFI from the signal of interest. Figure 3.2-2 demonstrates the process of determining the dominant Eigenvalues (power of RFI) from Sample Covariance Matrix using an azimuth block of raw data defined as Coherent Processing Interval

(CPI). Since corresponding computed Eigenvectors are mutually orthogonal, raw data can be projected in the direction of signal and noise Eigenvectors to remove RFI.

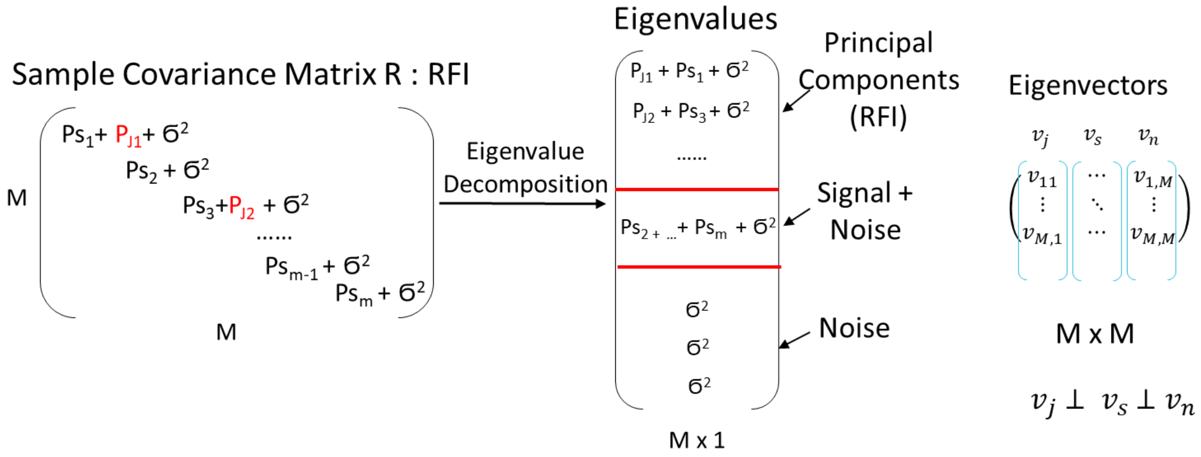


Figure 3.2-2. Determination of Principal Components

The variables defined in Figure 3.2-1are defined as follows:

- P_J represents RFI power.
- P_s represents signal power
- $σ^2$ represents noise power
- v_J represents Eigenvectors corresponding to RFI.
- v_S represents Eigenvectors corresponding to signal of interest
- v_n represents Eigenvectors corresponding to noise
- M = dimension of Sample Covariance Matrix

3.3.2 RFI Detection

The detection algorithm performs Maximum Eigenvalue Slope Test (MEST) on each CPI to determine the presence of RFI. The number of range lines in a CPI must be at least 2 to avoid EVD saturation which results in failure to separate the RFI Eigenvalue from that of the signal. In addition, the number of range lines in a CPI needs to be greater than the number of independent RFI present in a CPI. Once the number of range lines in a CPI is estimated, e.g., CPI = 20 range lines, MEST computes the difference of maximum Eigenvalue between adjacent CPIs for the entire raw data frame as shown in the equation below. The number of eigenvalues in a CPI is equal to that of range lines in a CPI.

$$\Delta\lambda_{max\_diff\_dB} = \lambda_{max\_dB,n} - \lambda_{max\_dB,\ n-1} \quad \text{(dB/CPI)}$$

where

- $\lambda\_max\_dB,\ n$ represents maximum Eigenvalue of CPI *n*
- $\lambda\_max\_dB,\ n-1$ represents maximum Eigenvalue of CPI *n-1*

- $\lambda\_max\_diff\_dB$ represents the difference in maximum Eigenvalue between CPI *n* and *n-1*

$\lambda\_max\_diff\_dB$ is compared against the detection threshold | in dB/CPI as shown in the hypotheses below to determine RFI presence.

$$H_0\text{: } \Delta\lambda_{max\_diff\_dB} < \tau \text{ (No RFI)}$$
$$H_1\text{: } \Delta\lambda_{max\_diff\_dB} > \tau \qquad \text{(RFI)}$$

RFI detection threshold is estimated by the standard deviation (STD) of minimum eigenvalues of all CPIs as shown in the equation below. The minimum Eigenvalues represent system noise response as a function of antenna input signal power. It has a similar standard deviation to that of the maximum Eigenvalues when no RFI is present. When RFI is present, minimum Eigenvalue standard deviation is significantly smaller than that of maximum Eigenvalues. Hence it is used to estimate detection threshold τ.

$$\tau = \sqrt{\frac{\sum\left(\lambda_{min\_dB,\, i} - \mu\right)^2}{N}}$$

where

- τ Represents RFI detection threshold in dB/CPI.
- $\lambda\_min\_dB,\ i$ represents minimum Eigenvalue of CPI i
- μ represents the mean of minimum Eigenvalues of all CPI
- N represents the total number of CPI

In order to perform ensuing RFI mitigation, an RFI mitigation threshold needs to be derived. Based on RFI detection threshold τ, adjacent CPIs with maximum eigenvalue slope greater than τ are flagged as RFI contaminated, and all of the eigenvalues of the contaminated CPI are removed from the Eigenvalue matrix of the entire raw data frame. The process continues until there are no more eigenvalue slopes between adjacent CPI greater than τ. The maximum remaining eigenvalue is the mitigation threshold ε.

3.3.3 RFI Mitigation

The following steps are performed to achieve RFI mitigation.

- Apply derived RFI mitigation threshold ε against all eigenvalues of a CPI.
- Remove Eigenvectors associated with Eigenvalues greater than ε.
- Project raw data of the CPI in the direction of remaining Eigenvectors (Apply remaining eigenvectors as filter weights).
- Generate RFI contamination flag for the CPI.
- Repeat the process until all CPIs are covered.

The ideal RFI mitigation results and mitigation artifacts are illustrated below in Figure 3.2-3.

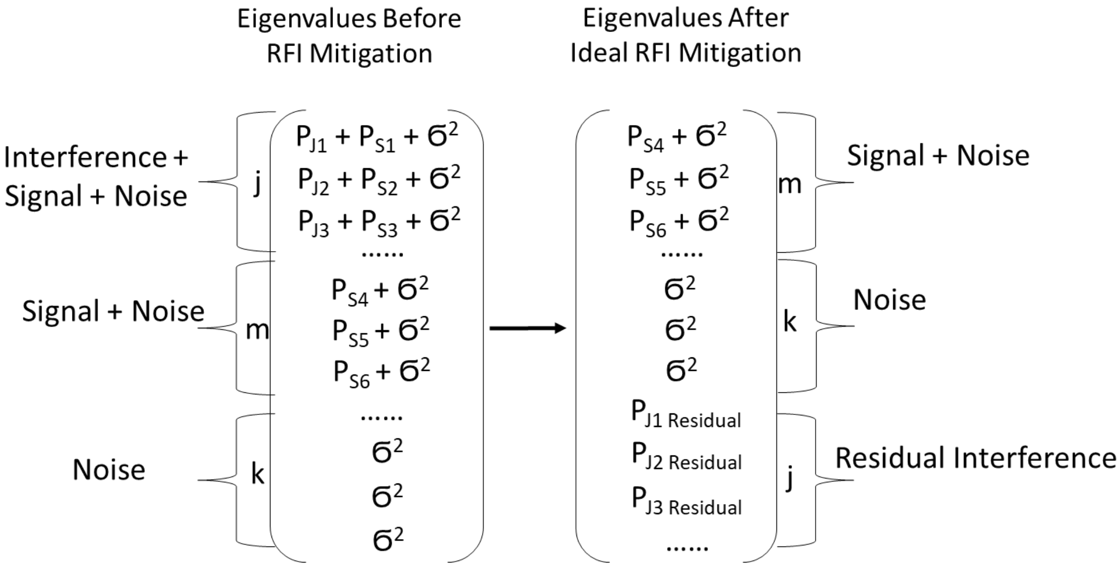


Figure 3.2-3. RFI Mitigation Artifacts

where

- P_J represents RFI power.
- P_s represents signal power
- б^2 represents noise power
- $P_{J\ Residual}$ represents residual RFI power.

Residual RFI power is close to zero which implies some signal power is also lost in the process of RFI mitigation. The amount of signal power loss is proportional to the number of independent RFI sources.

3.4 Imaging Geometry and Definitions

SAR focusing techniques combine information from numerous transmitted pulses to produce a high-resolution two-dimensional backscatter image of the area illuminated by the antenna footprint (Figure 3-2). Consequently, the observed amplitude and phase measurement at any single pixel in a SAR image cannot be attributed to any individual pulse in azimuth time or range bin in slant range. To better geolocate targets in focused SAR images, most processing approaches use various conventions based on the range-Doppler equation to set up reference functions for compressing energy in slant range and azimuth time domains.

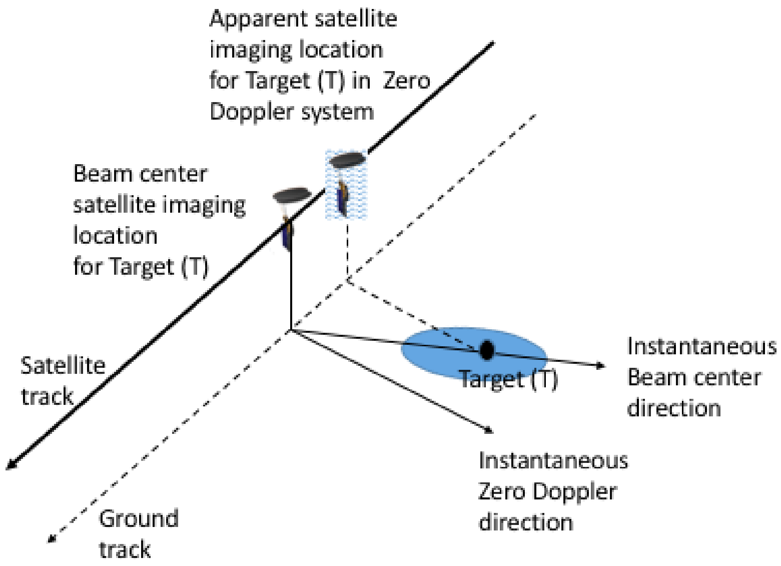

Apparent satellite imaging location for Target (T) in Zero Doppler system
Beam center satellite imaging location for Target (T)
Satellite track
Target (T)
Instantaneous Beam center direction
Instantaneous Zero Doppler direction
Ground track

Figure 3.2-1. Imaging geometry

The range-Doppler equation established the relationship between the Target T position $\vec{T}$ and the satellite imaging location:

$$\frac{2\cdot\overrightarrow{V_{sat}(\eta_{f,T})}\cdot\left(\vec{T}-\overrightarrow{R_{sat}(\eta_{f,T})}\right)}{\lambda\cdot R_{f,T}} = f(\eta_{f,T}, R_{f,T})$$

where

- | represents “slow time” domain, also referred to as azimuth time domain.
- R represents the slant range domain. R is Directly related to the “fast time” domain.
- $\overrightarrow{R_{sat}(\eta)}$ is the vector position of the satellite.
- $f_d(\eta, R)$ is the nominal Doppler centroid in Hz and is a function of azimuth time and slant range.
- $\eta_{0,T}$ is the zero Doppler azimuth domain position of Target T.
- $R_{0,T}$ is the zero Doppler slant range domain position of Target T.
- $\eta_{dc,T}$ is the beam center azimuth domain position of Target T.
- $R_{dc,T}$ is the beam center slant range domain position of Target T.

For a given Doppler frequency model $f\left(\eta_{f,T}, R_{f,T}\right)$, a Target T would show up at azimuth line location $\eta_{f,T}$ and slant range location $R_{f,T}$ in the focused image. Note that the choice of Doppler frequency model to describe the geometry of the SAR image can be arbitrary. However, there are two standard conventions widely used for easy interpretation of the imaging geometry: Native Doppler (or Beam Center) geometry and the Zero Doppler (or Tangential) geometry.

3.4.1 Native Doppler Geometry

The Native Doppler geometry system is the most natural system for representing SAR data. In this case, the Doppler frequency model is chosen to match the estimated Doppler centroid of the data, i.e.:

$$\frac{2 \cdot \overrightarrow{V_{sat}\left(\eta_{dc,T}\right)} \cdot \left(\vec{T} - \overrightarrow{R_{sat}\left(\eta_{dc,T}\right)}\right)}{\lambda \cdot R_{dc,T}} = f_{dc}\left(\eta_{dc,T}, R_{dc,T}\right)$$

The Doppler centroid at a given azimuth time and slant range determines the imaging geometry as well as the azimuth carrier on the data. The azimuth time and slant range correspond to the target’s passage through the center of the antenna along track footprint. The Native Doppler convention is ideal for applying antenna pattern and gain corrections. However, the Doppler centroid of the acquired data can vary in both azimuth time and slant range. Consequently, patch processing of the SAR pulses that accounts for updated processing parameters along-track introduces complications. The dependence on the varying Doppler centroid also makes it more complicated to mosaic SLC images on the same track that were processed with slightly different processing parameters.

3.4.2 Zero Doppler Geometry

The Zero Doppler geometry system is the most widely used convention for representing SAR data. In this case, Doppler frequency model is set to zero, i.e.:

$$\frac{2 \cdot \overrightarrow{V_{sat}\left(\eta_{0,T}\right)} \cdot \left(\vec{T} - \overrightarrow{R_{sat}\left(\eta_{0,T}\right)}\right)}{\lambda \cdot R_{0,T}} = 0$$

The imaging geometry can be determined independent of the Doppler centroid of the acquisition. The vector from the satellite to target is perpendicular to the instantaneous satellite velocity. Note that in case of the zero Doppler geometry, the azimuth time corresponding to a target can lie outside the interval defined by the imaging aperture. The SAR data still has an azimuth carrier defined by the Doppler centroid but this piece of information does not affect the geolocation or interpretation of the imaging geometry. This independence between Doppler centroid and imaging geometry allows one to mosaic images on the same track processed with different parameters easily.

3.5 Geometry Mapping Algorithms

At various stages in processing, we use algorithms to map targets in (azimuth time, slant range) domain to ECEF Cartesian domain and vice-versa. We describe these forward and inverse mapping algorithms here and reference them as needed in other sections.

3.5.1 Forward Mapping Algorithm

This algorithm maps a given target (T) located at azimuth time ($\eta_{f,T}$) and slant range ($R_{f,T}$) in radar image coordinates to map coordinates (X_{map},Y_{map},h(X_{map},Y_{map})). This is done by using a given Doppler model ($f_d(\eta, R)$) and a Digital Elevation Model (DEM) (z(X,Y)) as a function of horizontal datum coordinates X, Y. Details of various implementations of the forward mapping algorithm can be found in a number of reference [Kropatsch et al., 1990; Eineder, 2003; Sansosti et al., 2006; Nitti et al., 2011].

3.5.1.1Problem Formulation

We can formulate our forward problem as that of finding target position $\vec{T}$, such that the following two constraints are satisfied

$$\frac{2 \cdot \overrightarrow{V_{sat}(\eta_{f,T})} \cdot \left(\vec{T} - \vec{R}_{sat}(\eta_{f,T})\right)}{\lambda \cdot R_{dc,T}} = f_{dc}(\eta_{f,T}, R_{f,T})$$
$$\|\vec{T} - \vec{R}_{sat}(\eta_{f,T})\| = R_{f,T}$$

In the proposed implementation, the algorithm can be broken down into four steps:

1. Setting up a local TCN (tangential, cross-track, nadir) coordinate system at the location of the satellite where the $\hat{n}$ axis points from the satellite to the center of the Earth, $\hat{c}$ is perpendicular to the plane formed by $\hat{n}$ and the satellite velocity vector, and $\hat{t}$ completes the right-hand system and represents the tangential component of the velocity vector. The height of the target is initialized to some nominal value h_0.
2. Construct a local sphere with radius R_{geo} equal to the distance from the center of the Earth to the intersection of the satellite position vector with the ellipsoidal surface. The height of the target is initialized to some nominal value h_0 above this local sphere.
3. Solve the constrained optimization problem shown above for a point $\vec{T}$ on the local sphere. Convert the geocentric location of coordinates to map coordinates – X_{map}, Y_{map}.

Note that only the horizontal location information is used from this estimate for the next stage of the algorithm.

4. Interpolate the given DEM z(X,Y) to obtain z_{map}. Convert the coordinates (X_{map},Y_{map},z_{map}(X_{map},Y_{map})) to the local geocentric sphere system and estimate the height above local sphere, h_{est}.
5. Go back to Step 1, with h_0 = h_{est}. Continue iterating until h_{est} converges. When the algorithm converges (X_{map},Y_{map},z_{map}(X_{map},Y_{map})) from the latest iteration represents the target location in map coordinates.

Each of the steps is described in detail below. The algorithm can support analysis in both Native Doppler and Zero Doppler coordinate systems. For Zero Doppler coordinate system, the Doppler model ($f_d\left(\eta_{dc,T}, R_{dc,T}\right)$) is set to zero and by replacing $\eta_{dc,T}$ by $\eta_{0,T}$ and $R_{dc,T}$ by $R_{0,T}$.

3.5.1.2Local TCN Coordinate System

Let $\overrightarrow{R_{sat}}$ and $\overrightarrow{V_{sat}}$ represent the position of the satellite corresponding to the azimuth time of the target of interest. The local basis vectors for the TCN system are given by

$$\hat{n} = \frac{-\overrightarrow{R_{sat}}}{\|\overrightarrow{R_{sat}}\|}$$

$$c^{\wedge} = \frac{n^{\wedge} \times \overrightarrow{V_{sat}}}{\|n^{\wedge} \times \overrightarrow{V_{sat}}\|}$$

$$t^{\wedge} = c^{\wedge} \times n^{\wedge}$$

$$A = [t^{\wedge}, c^{\wedge}, n^{\wedge}]$$

3.5.1.3Range Doppler Equation Solution

Assuming that the target point is located at a height h_0 above the local sphere of radius (R_{curv}), the slant range vector can be represented in the local TCN basis as

$$\vec{T} = \overrightarrow{R_{sat}} + \alpha.\hat{t} + \beta.\hat{c} + \gamma.\hat{n}$$

Using the law of cosines on the local sphere, we can show that

$$\gamma = \frac{R_0}{2}.\left[\left(\frac{R_{sat}}{R_0}\right) + \left(\frac{R_0}{R_{sat}}\right) - \left(\frac{h_0 + R_{geo}}{R_{sat}}\right).\left(\frac{h_0 + R_{geo}}{R_0}\right)\right]$$

$$\alpha = \left(\frac{f_d(R_0)\lambda R_0}{2\|\overrightarrow{V_{sat}}\|} - \gamma(\hat{n} \cdot \hat{v})\right) \cdot \frac{1}{(\hat{t} \cdot \hat{v})}$$

where $R_{sat} = \|\overrightarrow{R_{sat}}\|$ and $\hat{v} = \frac{\overrightarrow{V_{sat}}}{\|\overrightarrow{V_{sat}}\|}$. β can be computed as

$$\beta = -L.\sqrt{R_0^2 - \gamma^2 - \alpha^2}$$

where $L = -1$ for right looking imaging geometry and $L = +1$ for left looking imaging geometry. Once, 〈, ® and © are computed we can compute the location of the target in Cartesian space ($\vec{T}$). The target location can be converted into map coordinates as $\left(X_{map}, Y_{map}, z_{map}\left(X_{map}, Y_{map}\right)\right)$ using standard transformations.

3.5.1.4 DEM Interpolation

DEMs are commonly provided in non-Cartesian coordinates (e.g., Lat-Long grid, UTM grid, EASE-2 grid) and contain heights above a geoid (e.g., EGM96 or EGM08). Since the NISAR orbit’s position, velocity and acceleration will be provided in Earth-Centered, Earth-Fixed coordinate system with respect to the WGS84 G1762 system, all the geometry mapping algorithms presented in this document explicitly assume that the DEMs are adjusted to represent heights above the representative ellipsoid, i.e., WGS84. Standard GIS tools offer numerous methods of interpolating height data (e.g., nearest neighbor, bilinear, and bicubic). Spline-based methods have the advantage of ensuring continuity of spatial derivatives (up to the order of the spline) over the domain of the spline knots, allowing for preservation of slope information for a finite region [Press et al., 2007]. Furthermore, spline-based methods tend to be more stable than polynomials and are resistant to significant oscillations between data points. We implement bivariate interpolating B-splines (preferably bicubic order) for DEM interpolation.

For the forward mapping algorithm, we interpolate the DEM at location (X_{map}, Y_{map}) to determine the new z_{map}. This new target location is then transformed into the local geocentric sphere system and estimate the height above local sphere, h_{est}, which becomes the initial height estimate h_0 for the next iteration of the algorithm. When trying to estimate the target location on a reference ellipsoid, the DEM is assumed to be of constant height z_{map} and the algorithm converges in two or three iterations.

3.5.2 Inverse Mapping Algorithm

This algorithm maps a given target (T) located at (X,Y, z(X,Y)) in map coordinates represented by horizontal datum (X,Y) to radar images coordinates - azimuth time (|) and slant range (R), using a given Doppler model ($f_d(\eta, R)$). Different implementations of the Inverse Mapping Algorithm can be found in several references [Eineder, 2003; Sansosti et al., 2006; Nitti et al., 2011].

The proposed implementation of the algorithm is based on the Newton-Raphson method and has three key steps:

1. We start with an initial guess for the azimuth time (| g), and interpolate the orbit state vectors provided in the metadata to estimate the satellite position $\overrightarrow{R_{sat}(\eta_g)}$ and velocity $\overrightarrow{V_{sat}(\eta_g)}$.
2. Perform a Newton Raphson iteration to determine the change to be applied to the azimuth time guess (⊗|).
3. Continue iterations until the change in azimuth time (⊗|) is negligible. When the algorithm converges, | g represents the azimuth time and the distance between the satellite location at | g and the given target represents the slant range in radar image coordinates.

Each step is described in detail below. The algorithm can support analysis in both Native Doppler and Zero Doppler coordinate systems. For Zero Doppler coordinate system, the Doppler model ($f_d(\eta_{dc,T}, R_{dc,T})$) is set to zero.

3.5.2.1 Orbit Interpolation

The different NISAR orbits, including the Near real-time Orbit Ephemeris (NOE), Medium precision Orbit Ephemeris (MOE) and Precise Orbit Ephemeris (POE), are uniformly sampled every 10 seconds [Ref]. To precisely map targets from map coordinates to radar image coordinates, we need to be able to interpolate the orbit state vectors with accuracy on the order of a few mm. Two possible interpolation methods satisfy this requirement:

1. Hermite polynomial interpolation

   A third-order Hermite polynomial can be used to interpolate the orbit information reliably. The Hermite polynomial is constructed using 4 state vectors spanning the azimuth time epoch of interest; and combines position and velocity information for interpolating the state vectors.
2. Legendre polynomial interpolation

   An eighth-order Legendre polynomial can also be used to interpolate the orbit information reliably. The Legendre polynomial is constructed using 9 state vectors spanning the azimuth time epoch of interest; and interpolates the position and velocity arrays independently of each other.

We use Hermite polynomials to interpolate the NISAR orbit state vectors.

3.5.2.2 Newton-Raphson Iteration

The function $y(\eta)$, whose zero crossing we are trying to determine using the Newton-Raphson method can be directly derived from the range-Doppler equation.

$$y(\eta) = \overrightarrow{V_{sat}(\eta)} \cdot \left(\vec{T} - \vec{R}_{sat}(\eta)\ \right) - \frac{\lambda}{2} . f_d\big(\eta, R_{dc}(\eta)\big). R_0(\eta)\ = 0$$

where

$$R_{dc}(\eta) = \|\vec{T} - \vec{R}_{sat}(\eta)\ \|.$$

The adjustment to the initial guess for the azimuth time epoch ($|_{\mathrm{g}}$) is given by

$$\eta_{new} = \eta_g - \frac{y\big(\eta_g\big)}{y'\big(\eta_g\big)}$$

where

$$y'(\eta) \approx \frac{\lambda}{2} . \left[\frac{f_d\big(\eta, R_{dc}(\eta)\big)}{R_{dc}(\eta)} + f_d'\big(\eta, R_{dc}(\eta)\big)\right] . \left(\overrightarrow{V_{sat}(\eta)} \cdot \left(\vec{T} - \vec{R}_{sat}(\eta)\right)\right) - \ \|\overrightarrow{V_{sat}}\|^2$$

The Newton-Raphson iterations are continued until the estimated azimuth time converges, i.e., the range-Doppler equation is satisfied. When the algorithm converges $R_{dc}(\eta_{new})$ represents the slant range to the target.

3.5.3 Terrain Height Function

For improved radiometric performance, we update the processing parameters for focusing in the along-track direction. These processing parameters are modified depending on the variation of topography in the antenna footprint. Before we start processing the RRSD to RSLC data, we determine a vector of terrain heights as a function of azimuth sensing time [Section 4.4 in Piantanida, 2016].

Each terrain height record entry is derived using the orbit information and a low-resolution DEM (e.g, GMTED2010) and represents the average height of terrain across the range swath and averaged over an aperture length in azimuth. This is done by assuming a zero Doppler imaging geometry and using the Forward Geometry Mapping algorithm. The impact of assuming a zero Doppler geometry is negligible since we are looking at aperture-wide averages of topography height. The L0 data is annotated with this information for use in focusing the data.

Note that a single set of terrain height vectors is used for all the acquired polarizations. This vector changes slowly in time due to large aperture length, and can be interpolated using linear interpolation in azimuth [Piantanida, 2016].

3.5.4 Effective Radar Velocity

To optimize the quality of the RSLC product and to reduce the impact of phase errors introduced by SAR focusing, the effective SAR platform velocity needs to be estimated to within 0.1 m/s. While this parameter is not used in the RSLC processor, it is included in the product metadata. A flat earth geometry approximates the one acquired from a satellite flying in a curved orbit. This equivalence is achieved by associating the true range history of a target on ellipsoidal earth with that observed from a flat earth-straight orbit imaging geometry.

The effective velocity is a function of slant range and azimuth time, and is computed using the method described in Wong et al. [2000]:

1. For the given azimuth time (|), estimate the terrain height $h_{ref}(\eta)$ using method described in Section 3.3.3 and satellite state vector by interpolating the orbit information.
2. For the given azimuth time (|) and slant range (R), estimate the geometric Doppler centroid $f_{dc}(\eta, R)$ (Section 3.6.1)
3. Using this information estimate the target location $\overrightarrow{T_{ref}}(\eta, R)$ on ground on the reference surface described by the estimated terrain height.
4. Choose a time-period corresponding the imaging aperture (azimuth bandwidth) of interest, and interpolate the orbit information to create an array of time-sorted state vectors.
5. Using the array of state vectors, estimate the range history to $\overrightarrow{T_{ref}}(\eta, R)$ and fit a hyperbola to estimate the effective velocity. The estimated effective velocity is a function of the processing azimuth bandwidth.

The effective velocity is typically estimated every aperture length (e.g., ~2.3 seconds for L-SAR).

3.6 Gap Filling and Regridding

The NISAR L-SAR instrument uses a PRF dithering scheme to overcome block bands in the SweepSAR imaging mode [Villano et al., 2014]. Consequently, the samples are not uniformly sampled in slow time and have gaps where transmit events overlap the receive window. Filling these gaps using data from adjacent pulses substantially improves the azimuth impulse response. The NISAR L-SAR processor uses the best linear unbiased (BLU) interpolation scheme, which is optimal for homogeneous clutter [Villano et al., 2014; Hawkins, 2019].

Before interpolation, the data are shifted to baseband in azimuth using a phase ramp conjugate to the Doppler centroid. The phase of the ramp is zero at an arbitrary time η_0 chosen at the center of the processing block. Then for each desired pulse time η, the output is determined by a linear combination of adjacent pulses

$$s_{out}(\eta) = \sum_i \quad w_i s_{in}(\eta_i)$$

The weights w_i are determined by solving the linear system

$$Aw = b$$

where the coefficients are

$$A_{ij} = K(\eta_i - \eta_j)$$
$$b_i = K(\eta - \eta_i)$$

and the kernel $K(\eta)$ is the azimuth autocorrelation function of the signal, given by the Fourier transform of the azimuth antenna pattern. For a perfect sinc antenna pattern, the kernel is a piecewise cubic polynomial. The indices are all those where the kernel is nonzero, typically 2-4 pulses depending on the PRF. After resampling, the data are shifted back to the Doppler carrier frequency by multiplying by a phase ramp with the same zero-phase point η_0 used earlier.

3.7 Range Processing

The first step in the SLC formation from the L0B RRSD product is range processing. SLC processing is performed in azimuth blocks. Range processing of each azimuth block consists of the following steps:

- Range reference function generation
- Drift compensation
- Range compression
- Range dependent gain correction
- SWST bias correction

3.7.1 Range Reference Function

Range compression involves the use of a matched filter commonly referred to as the range reference function [Section 6.1.1 in Piantanida, 2016]. The construction of this matched filter begins with a reference replica $Rep(t)$ that is generated from either the extracted chirp replica or the nominal chirp function (Section 3.4.3). The reference function is zero-padded to length N_{fft} that is large enough to accommodate the sum of the lengths of the raw signal data and twice the length of the reference replica. The reference function is then transformed to the frequency domain. The amplitude is flattened in the frequency domain to yield a reference spectrum $R(f)$ which is normalized to give matched filter with unit energy:

$$R_1(f) = \frac{R(f)}{\sqrt{\frac{1}{N_{fft}} \sum_{f=0}^{N_{fft}-1} \quad |R(f)|^2}}$$

For polarimetric processing, a linear phase ramp is applied in the frequency domain to introduce a time shift Δt equivalent to the polarization channel registration offset:

$$R_2(f) = R_1(f)\; exp(-j2\pi f \Delta t)$$

To align the chirp replicas, the co-pol channel is used as a reference for dual-pol data and the HH channel is used as a reference for qual-pol data. The matched filter is now normalized by the oversampling factor to ensure that the gain of the matched filter is unity:

$$R_3(f) = R_2(f) \frac{1}{N_r} \sqrt{\frac{B_r}{f_s}}$$

where B_r is the pulse bandwidth, f_s is the range sampling rate, and $N_r$is the number of non-zero samples in $Rep(t)$. Complex gain corrections due to the different polarizations are also applied at this stage to the matched filter.

The matched filter is now ready for use in range compression operation.

3.7.2 Drift Compensation

The decoded range line is now transformed into the frequency domain by performing an FFT of length N_{fft}. Note that N_{fft} accommodates the length of the received pulse as well twice the length of the nominal replica. Any delays due to instrument drift are accommodated by multiplying the range spectrum by a linear phase ramp.

3.7.3 Range Compression

The conjugate of the reference filter is multiplied with the range spectrum and an inverse FFT is performed. The valid portion of the range-compressed line is preserved for further processing.

3.7.4 Range-Dependent Gain Correction

After range compression, the data is compensated for the elevation antenna pattern and range spreading loss. A thermal noise estimate is also determined and stored in the RSLC metadata [Section 6.1.2 in Piantanida, 2016].

3.7.4.1 Elevation Antenna Pattern (EAP) Correction

Each range line is corrected using an EAP gain vector. This array is computed from the calibration and image information in the L0B product metadata as described in [Hawkins, 2022]. The EAP gain vector may change every pulse in order to track changes in the instrument state. The measured antenna orientation is used for the EAP correction.

First, the transmit pattern g_{TX} is synthesized using the twelve element patterns

$$g_{TX}(EL) = \sum_{i=1}^{12} \gamma_i g_i(EL)$$

where γ_i reflects the most recent internal calibration measurements (defined as the ratio of the HPA and bypass loopback calibration signals) and $g_i(EL)$ are the individual beam patterns provided by the system engineering team in an ancillary file (either simulated or measured patterns). Similarly, the received digitally beamformed (DBF) pattern g_{RX} is computed

$$g_{RX}(EL, \tau) = \sum_{i=1}^{12} w_i(\tau) g_i(EL)$$

where the fast-time weights $w_i(\tau)$ are computed using the instrument time-to-angle (TA) and angle-to-coefficient (AC) look-up tables. That is, each fast time is mapped to an elevation angle using the TA LUT, which is then mapped to a DBF weight using the AC LUT using the exact same LUTs as the instrument [AD7]. No additional internal calibration corrections are applied to the receive pattern because these have already been compensated on-board.

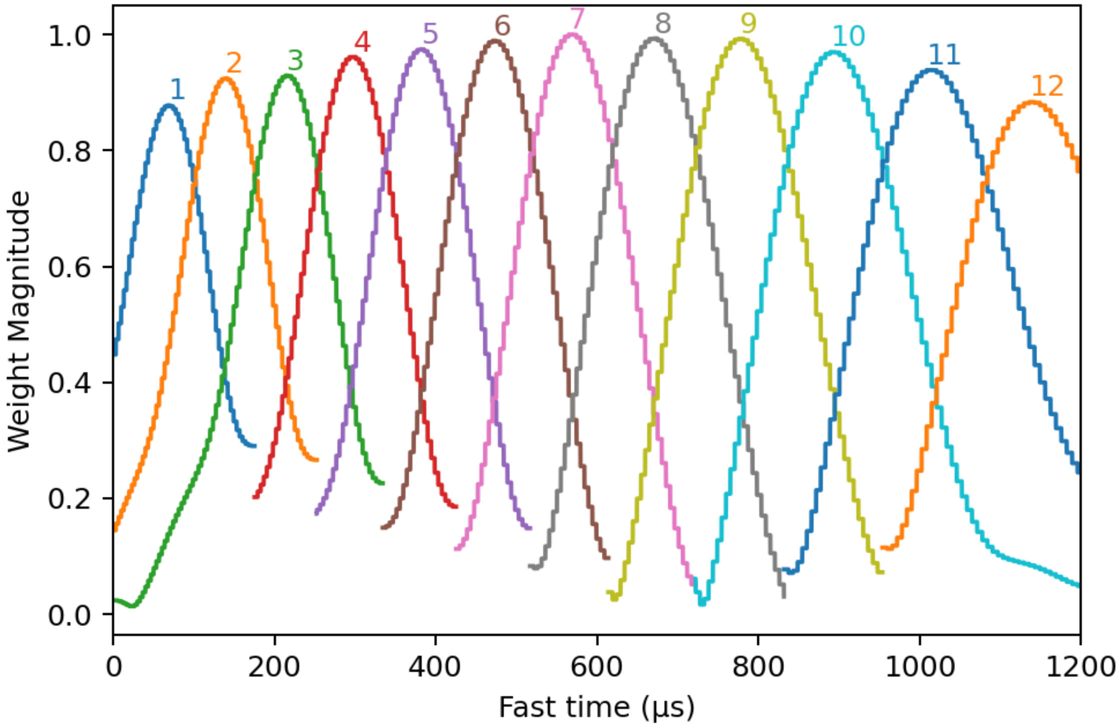


Figure 3.7-1 DBF weights for each beam obtained from instrument tables.

While the TA table is used to reconstruct the weights used by the onboard DBF processor, the RSLC processor has better knowledge of the scene geometry. Therefore the forward geometry mapping algorithm is used to construct the $EL(\tau)$ curves used to sample the gain patterns. The transmit and receive patterns are multiplied together, linearly interpolated at $EL(\tau)$, and used to normalize each range-compressed pulse.

3.7.4.2 Pulse Width Correction

An idiosyncrasy of the SweepSAR measurement technique is that the beam continues sweeping across the swath while the pulse from a target at a fixed elevation is being recorded. As a consequence, the time-domain envelope of each target is modulated by the dynamic antenna pattern [Ghaemi, 2014; Younis, 2015]. Since the waveform is an LFM pulse, there is an equivalent range frequency-domain modulation. Obviously the effect is more pronounced the longer the chirp.

Because the modulation is range-dependent, the processor does not attempt to undo the spectral windowing. However, the following simple model is used to correct the overall range-dependent radiometric loss [Hawkins, 2022].

$$G_{pw}(\tau) = \frac{1}{2\delta(\tau)}\sqrt{\frac{\pi}{k_2}}erf(\delta(\tau)\sqrt{k_2})$$

where $\delta(\tau)$ is the elevation angle extent of the pulse, k_2 is a parameter related to the instantaneous beam width (k_2=1.92 deg^{-2} for NISAR), and erf(x) is the Gauss error function. Note that

$$\delta(\tau) = T\frac{\partial EL}{\partial \tau}$$

where T is the pulse width and the second term is the slope of the time-to-angle curve $EL(\tau)$ found via the forward mapping algorithm.

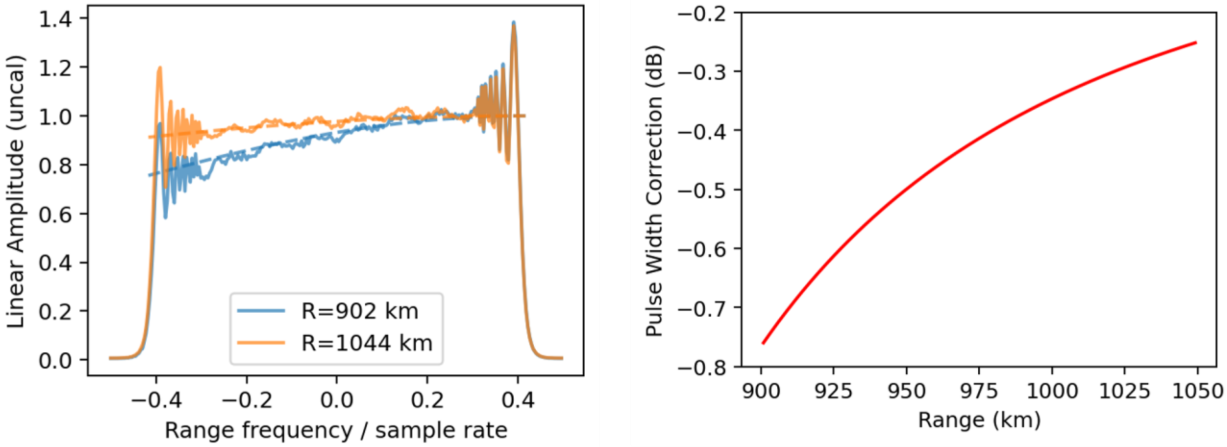


Figure 3.7-2 Modulation for a 40 μs pulse. On the left are two simulated point targets in the near range (blue) and far range (orange). Solid lines show the measured spectra while the dashed lines show the one parameter model. On the right is the resulting loss $G_{pw}(\tau)$.

3.7.4.3 Range Spreading Loss Correction

The radar equation shows that the total energy that is backscattered from a point target drops as the fourth power of the slant range (R^4). However, each range-compressed line represents the energy backscattered from the entire illuminated area on the ground. Since the illuminated area itself is proportional to the slant range, the correction factor can be expressed as [Section 9.6 in Piantanida, 2016]:

$$G_{rsl}(R) = \sqrt{\left(\frac{R_{ref}}{R}\right)^3}$$

where R_{ref} is a reference range, chosen for processing such that the the amplitudes / energies of the backscattered signal are normalized with respect to this reference range for further processing and R is the range to each sample. R_{ref} takes the fixed value 950 km for all data globally so that its contribution to the processor gain does not vary scene-to-scene. Note that the gain correction from the range spread loss is applied to the magnitude of SLC data only and the phase information is not affected.

3.7.5 SWST Correction

The Sampling Window Start Time (SWST) times as well as the maximum slant range times for all the range pulses in a given azimuth block are used to estimate the extent of valid data. Each range line is padded with zeros at the start and the end, as necessary to align the samples of each range line on a common grid with common SWST. Any common range gate biases are also applied at this stage to modify the metadata associated with the processed azimuth block.

3.8 Doppler Centroid Estimation

The overall DCE algorithm is split into two pieces. The image formation processor always uses a Doppler centroid computed from geometry. A separate processor measures the Doppler centroid from the raw L0B data over a carefully selected set of bright, homogeneous targets. This results in a product containing Doppler estimates as a function of range and time that is used by the guidance and control team to refine the attitude products. As a result the geometric Doppler used in image formation will maintain close agreement with measurements of the raw data.

3.8.1 Geometric Doppler Centroid

The RSLC processor determines the absolute DC from spacecraft orbit and attitude data. This information is expected to be good enough to determine the Doppler centroid to within a small fraction of the Doppler bandwidth. This does not depend on the presence of strong scatterers in the image when using a data-based approach. The DC calculation approach is similar to the one specified in Section 5.1 of Piantanida [2016].

The Doppler frequency f_d of a target located at the radar beam center is:

$$f_d(\eta, R) = -\frac{2}{\lambda}\vec{v}(\eta) \cdot \hat{r}(\eta, R)$$

where $\vec{v}$ is the platform relative (i.e., earth-centered, earth-fixed) velocity, $\hat{r}$ is the unit look vector from the antenna phase center to the target, and λ is the radar wavelength. The only unknown in this equation is the direction of the look vector, which can be parameterized by two angles. The NISAR antenna patterns are provided in an azimuth-and-elevation (AZ and EL) grid in the radar antenna Reflector Coordinate System (RCS) defined in [AD6]. In the RCS frame, the azimuth antenna pattern is centered at $AZ=AZ_c$ over the entire relevant range of EL angles (AZ_c=0.9° subject to on-orbit verification). Finally, we can constrain the EL angle using a DEM with a root-finding procedure like in the forward geometry mapping algorithm.

Specifically, we can generate a look vector using AZ=AZ$_c$ and some trial EL angle. These are related to the polar angles (θ, ϕ) via

$$\theta = \sqrt{EL^2 + AZ^2}$$

$$\phi = tan^{-1} \frac{AZ}{EL}$$

which define the look vector in the RCS frame

$$\hat{r}_{RCS} = [sin\ \theta\ cos\ \phi, sin\ \theta\ sin\ \phi, cos\ \theta]^T$$

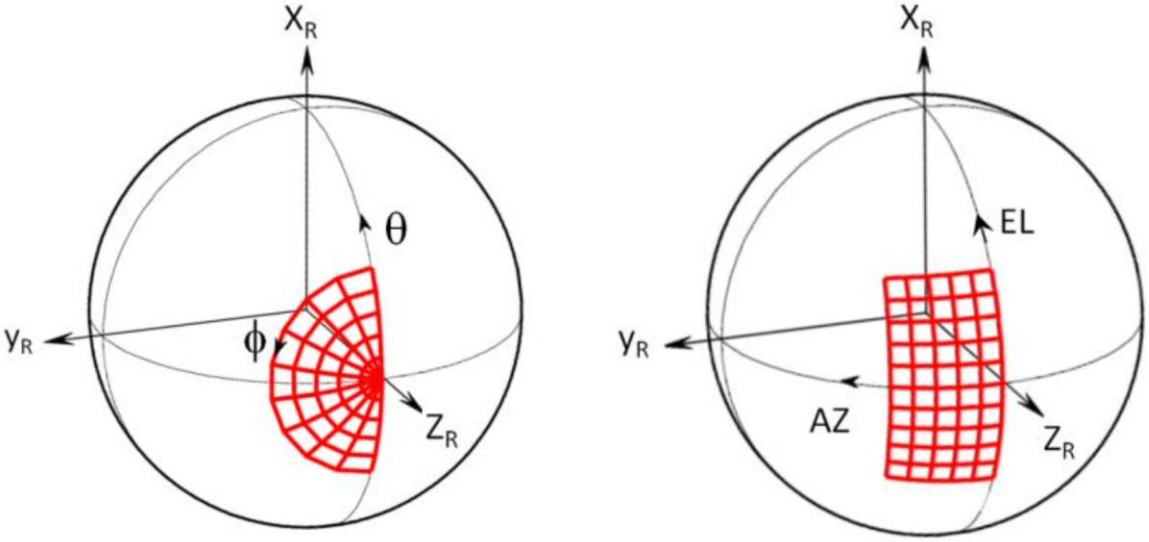


Figure 3.7-2 Angles used to parameterize line of sight vector in RCS frame.

The look vector can be expressed in ECEF XYZ coordinates using the quaternion $q(\eta)$ from the attitude data

$$\hat{r} = q\ \hat{r}_{RCS}\ q^{-1}$$

A target position $\vec{T}$ can then be generated

$$\vec{T}(\eta, R) = \vec{R}_{sat}(\eta) + R\ \hat{r}$$

Given the 3D target position one can compute the associated latitude, longitude, and height above the ellipsoid. The latitude and longitude can be used to query the DEM, from which we can define the height error

$$\Delta h(EL) = h - DEM(\phi_{lat}, \lambda_{lon})$$

We use a 1D root finding procedure to find the solution $\Delta h(EL_0) = 0$. Now the look vector is fully defined in terms of (AZ_c, EL_0) and the Doppler centroid can be calculated.

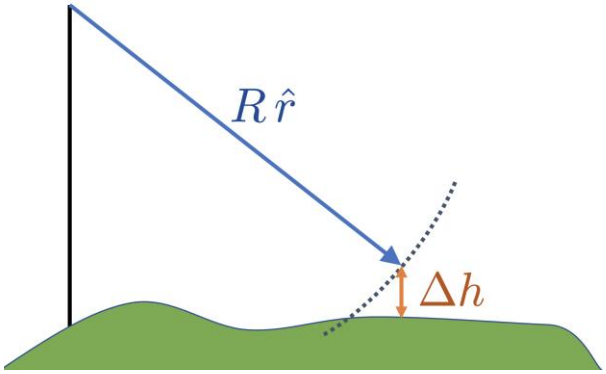


Figure 3.7-3 As the EL angle is varied the line of sight vector traces out an arc. The solution EL_0occurs where the arc intersects the DEM.

This computation is performed over a discrete set of azimuth times and slant ranges that span the entire imaging swath and results in a 2D look-up table (LUT) of the Doppler centroid. These estimates slowly change in space and are interpolated using bilinear interpolation as needed. The LUT is stored in the RSLC product metadata.

3.8.2 Measured Doppler Centroid

The DC estimate obtained using spacecraft orbit and attitude information may deviate from the true value due to systematic pointing errors. Therefore a separate process is utilized in the commissioning phase (and during calibration periods) to estimate these systematic errors using the radar data itself. The PRF is approximately 2 kHz and limits the observable range of Doppler frequencies. The measurement is therefore assumed to lie within ± PRF/2 of the value estimated from the orbit geometry.

We summarize here the description of the Correlation DC Estimation (CDCE) algorithm in Section 5.2 of Piantanida [2016]. The CDCE approach [Madsen, 1989] is implemented in ISCE3 and uses the average cross correlation coefficient (ACCC) between consecutive data lines. The CDCE is performed on raw data in the azimuth-time, range-time domain. The ACCC $c(\eta,\tau)$ for each line of data is calculated as:

$$c(\eta,\tau)=\sum_{\eta} s(\eta,\tau)s^{*}(\eta-\Delta\eta,\tau)$$

where $s(\eta,\tau)$ is the complex sample for a pulse at azimuth time | and slant range time |, $s(\eta-\Delta\eta,\tau)$ is the complex sample from the previous pulse and same slant range, and $\Delta\eta$ is the time separation between consecutive pulses. This operation is performed only when both consecutive pulses contain valid data.

The cross-correlation vector $c(\eta,\tau)$ is split into blocks and the averaged ACCC value is computed for each block. The DC frequency for each range block is then calculated:

$$f_d(\eta_i, \tau_j) = \frac{1}{2\pi\Delta\eta}\phi_{accc,i,j} + \frac{n_i}{\Delta\eta}$$

where $\phi_{accc,i,j}$ is the angle of each averaged ACCC after phase unwrapping in the range direction. The ambiguity number n_i is the integer that best aligns the median ACCC estimate with the geometric Doppler evaluated at mid-swath. Thus we consider the Doppler ambiguity, which may be nonzero since NISAR is expected to be squinted nearly one beamwidth. However, we assume that the variation across the swath is small relative to the PRF.

These Doppler measurements are delivered to the guidance and navigation team in order to be used to estimate biases to the pointing measurements. These biases are incorporated into the attitude products used for RSLC processing. Thus the RSLC products are always processed using the geometric Doppler, but the geometry has already been steered to match the available data.

3.9 Azimuth Processing

A time-domain back-projection algorithm is used to focus the data in azimuth [Frey et al., 2009]. The output image is computed pixel-by-pixel as the sum

$$s_{ij} = \sum_{k=k_c-N/2}^{k_c+N/2} exp(2\pi\sqrt{-1}f_c\tau_{ijk})x_k(\tau_{ijk})$$

where s_{ij} is the output pixel for grid point (i,j), τ_{ijk} is the two-way path delay between the radar at pulse k and grid point (i,j), $x_k(\tau)$ is the range-compressed pulse k interpolated at delay τ, and f_c is the radar center frequency. The sum is centered on pulse k_c nearest the Doppler centroid time, and the number of pulses N is determined by the azimuth resolution δs

$$N = \frac{\lambda R_{ij}(1 + h/a)}{2\delta s\Delta s}$$

where $\lambda = c/f_c$ is the radar wavelength, Δs is the average pulse spacing, $R_{ij} \equiv \tau_{ijk_c} \cdot c/2$ is the range to the pixel from the center of the aperture, h is the radar altitude, and a is the radius of the Earth. The precise target grid, delay model, and range interpolation scheme are described below.

3.9.1 Focusing Grid

The backprojection algorithm affords substantial flexibility in the selection of the output target grid. Nevertheless, the RSLC product is produced on a zero-Doppler grid in order to simplify downstream analysis and maintain compatibility with SAR workflows. A vector of azimuth times $\eta_i = \eta_0 + i\Delta\eta$ is constructed at a uniform PRF of 1520 Hz (configurable). Similarly, a vector of slant ranges is constructed $r_j = r_0 + j\Delta r$ with a range spacing equal to $\frac{c}{2f_s}$ where the radar sample rate f_s is typically 1.2 times the chirp bandwidth.

The forward mapping algorithm (with zero-Doppler) is used to determine the 3D target position $\vec{x}_{ij}$ associated with the zero-Doppler coordinate (η_i, r_j) of the output image pixel (i, j). The inverse mapping algorithm (with native Doppler) is used to determine the time η_c and range R_{ij} associated with the target's beam-center crossing. The aperture sum is therefore centered on pulse $k_c = (\eta_c - \eta_0)/\Delta\eta$ rounded to the nearest integer.

3.9.2 Delay Model

The range delay τ_{ijk} includes three main terms. The first is the geometric range

$$\rho_{ijk} = |\overrightarrow{x_{ij}} - \vec{p}_k|$$

where $\vec{p}_k$ is the location of the antenna phase center at the transmit time of pulse k. The second term is the delay due to the dry component of the troposphere τ_{atm}. In order to avoid dependence on external weather data, we utilize the same model that is employed for TerraSAR-X annotations:

$$\tau_{atm,ij} = \frac{2}{c}\frac{D_0}{cos\ \theta_{inc,ij}}\, exp\left(\frac{-h_{ij}}{H}\right)$$

where θ_{inc} and h are incidence angle and height at the pixel location, and the model parameters are the constant values $D_0 = 2.3$ m and $H = 6000$ m [Breit et al., 2010]. Note that the incidence angle does not change significantly during the synthetic aperture, so this term can be evaluated once at η_c ahead of the pulse loop. Finally, the delay model includes the so-called bistatic correction to the start-stop approximation [Breit et al., 2010; Schubert et al., 2015]. The complete delay is then

$$\tau_{ijk} = \tau_{atm,ij} + \frac{2\rho_{ijk} + 2(\vec{x}_{ij} - \vec{p}_k)\cdot\vec{v}_k/c}{c(1 - v_k^2/c^2)}$$

where $\vec{v}_k$ is the velocity of the antenna phase center at pulse k.

3.9.3 Range Interpolation

Having determined the range delay, it is necessary to interpolate the sampled, range-compressed pulse to produce the value at that exact range delay. The processor employs a Knab kernel for range interpolation [Knab, 1979; Knab, 1983; Migliaccio *et al*, 2007]. This function is defined

$$\varphi(t) \; = \frac{sin \;\; \pi t}{\pi t} \frac{cosh \left[\frac{\pi \nu L}{2} \sqrt{1 - \left(\frac{2t}{L}\right)^2}\right]}{cosh \left(\frac{\pi \nu L}{2}\right)}$$

where ν is the bandwidth, L is the length of the kernel, and all time units are normalized by the range sample rate.

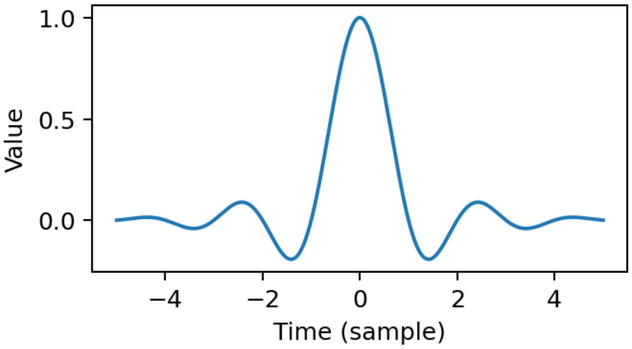


Figure 3.8-1. Range interpolation kernel for L=9 and f_s=1.2B

Accuracy improves with decreasing bandwidth and increasing kernel length. Range interpolation is achieved by computing the inner product with $L = 2m + 1$ samples of $\varphi(t - t_{out})$. Explicitly,

$$x_k(\tau_{ijk}) = \sum_{n=n_0-m}^{n_0+m} \quad \varphi(n - (\tau_{ijk} - \tau_0)f_s)\tilde{x}_{kn}$$

where $\tilde{x}_{kn}$ is the nth sample of pulse k, n_0 is the range sample nearest to τ_{ijk}, τ_0 is the sampling window start time, and f_s is the range sample rate.

3.10 Post-Processing

3.10.1 Polarimetric Calibration

After focusing, a final correction is applied to achieve polarimetric calibration. In its most general form, this is the linear transformation [Ainsworth *et al*, 2006]

$$\vec{s}_{cal} = M \, \vec{s}_{meas}$$

where we have arranged the scattering elements $\vec{s} = [s_{hh}, s_{hv}, s_{vh}, s_{vv}]^T$ and M is a 4x4 calibration matrix representing the absolute gain, channel imbalance, and crosstalk. In practice it is not possible to independently measure 16 complex-valued calibration parameters, so a reduced set of parameters is chosen and estimated from calibration targets as described in [AD8]. When the instrument is not operating in quad-pol mode, the crosstalk is ignored and only the diagonal elements of M are applied to the available polarimetric channels. The RSLC products are not compensated for Faraday rotation.

3.10.2 Quantization from CFloat32 to CFloat16

The complex image data processing is carried out in single precision complex floating-point format. However, in order to reduce data volume, the data are converted to half precision complex floating-point format, namely a pair of binary16 values [IEEE 754-2008] representing the real and imaginary parts. This format is well-suited to SLC data since it provides 11 bits of precision throughout the entire feasible dynamic range (-84 dB to +96 dB) without resorting to integer scale factors or lookup tables. Moreover, the short float format is supported by popular software tools including NumPy and HDF5.

4 GEOCODED SLC ALGORITHM

4.1 Introduction

The GSLC product is generated from the RSLC product using the following processing steps (Figure 4-1):

- Define the output geocoded grid of the GSLC product
- Inverse mapping of each pixel of the geocoded grid into slant range-Doppler coordinates
- Refine the estimated slant range for the ionospheric range delay using external TEC data
- Interpolate the complex RSLC signal into the derived range-Doppler coordinates in the previous step
- Flatten the phase of the GSLC using the geometrical slant range

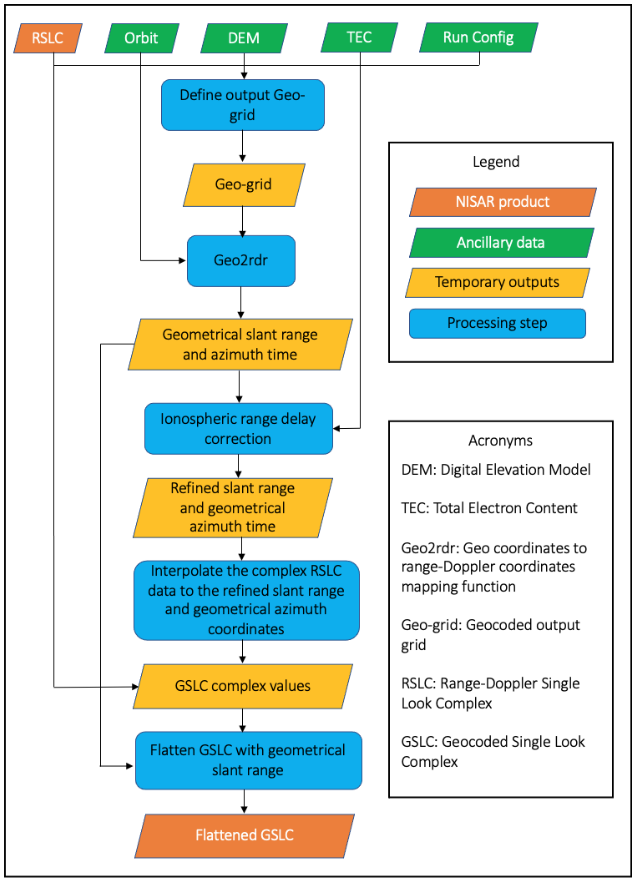

RSLC
Orbit
DEM
TEC
Run Config
Define output Geo-grid
Geo-grid
Geo2rdr
Geometrical slant range and azimuth time
Ionospheric range delay correction
Refined slant range and geometrical azimuth time
Interpolate the complex RSLC data to the refined slant range and geometrical azimuth coordinates
GSLC complex values
Flatten GSLC with geometrical slant range
Flattened GSLC
Legend
NISAR product
Ancillary data
Temporary outputs
Processing step
Acronyms
DEM: Digital Elevation Model
TEC: Total Electron Content
Geo2rdr: Geo coordinates to range-Doppler coordinates mapping function
Geo-grid: Geocoded output grid
RSLC: Range-Doppler Single Look Complex
GSLC: Geocoded Single Look Complex

Figure 4.1. Geocoded SLC product generation workflow

4.2 Define the output geocoded grid

In the first stage of the workflow, the geocoded grid of the output GSLC product is defined. This grid can be defined by the user through the run configuration parameters or may be determined using the bounding box of the input RSLC product. The NISAR RSLC product includes a polygon representing the bounding box of the product on the ground. NISAR data system processes the L1 and L2 products over predefined track and frames. The L2 geocoded products for each frame are processed from the L1 Range_Doppler products. Figure 4-2 demonstrates the relation between the L1 RSLC coverage and L2 GSLC products geocoded grid.

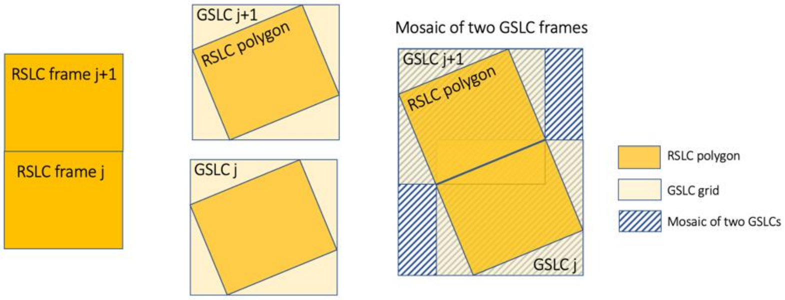


Figure 4-2. The representation of the L2 GSLC grid and it's relation with the L1 RSLC coverage. The mosaic of the GSLCs are shown for visualization only and will not be provided by the NISAR Data System.

4.3 Inverse mapping of geocoded grid pixels

In the second stage of the workflow, the geocoded grid of the output GSLC is mapped to the Range Doppler coordinates of the input RSLC using an existing DEM, orbit information and imaging geometry of the input RSLC. This inverse mapping of the geo coordinates to the Range-Doppler coordinates is done with the geo2rdr algorithm as described in section 3.4.2.2. For each pixel of the geocode grid, the output of this step is the slant range between the antenna phase center and the center of the geocoded grid and the azimuth time of the antenna phase center along the orbit given the zero Doppler grid of the RSLC. Since only geometrical information is used at this step, the estimated “slant range and azimuth time” are referred to “geometrical slant range and azimuth time”. The geometrical slant range and azimuth time can be biased by different sources of errors including DEM error, propagation delay of the microwave signal through troposphere and ionosphere, and motion of earth surface especially those caused by solid earth and ocean tidal loading (Yunjun et al, 2022; Gisinger et al, 2021).

Ionosphere is a dispersive media with respect to microwave signals. The propagation of the microwave signal through the ionosphere will cause phase advance and group delay. The latter translates to geolocation errors in the range direction if not accounted for. The ionospheric delay is inversely proportional to the radar frequency squared. Therefore, low frequency SAR data (such as L-band NISAR data) are more affected by ionospheric delay than the high frequency SAR data.

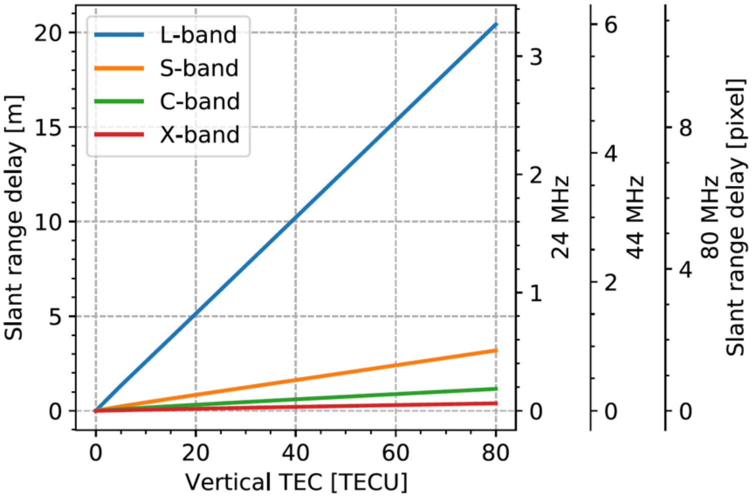


Figure 4-3. Slant range delay as a function of vertical TEC for different radar frequencies. The y-axis on the right side maps the delay to pixel shift for different signal bandwidths. (Yunjun et al 2022)

Figure 4-3 shows the slant range delay as a function of the ionospheric Total Electron Content (typically expressed as TECU = 10^16 electrons/m^2) for different radar frequencies. The ionospheric delay at L-band is ~6, ~17 and ~51 times larger than S-, C- and X-bands respectively.

The tropospheric delay can be divided into the hydrostatic and wet delay. The zenith hydrostatic delay (often called “dry”) is ~2.3 m at sea level and varies as a function of surface pressure (~2.273 mm/mb) and thus topographic height with smaller delay at higher altitudes. The zenith “wet” delay caused by water vapor is much smaller than the hydrostatic delay and may reach to ~40 cm in the equatorial regions. The temporal variation of the wet delay is generally larger than the hydrostatic delay. For NISAR GSLC products we correct the range geolocation errors due to ionospheric delay using ionospheric TEC estimates from Global Navigation Satellite Systems (GNSS) on the ground and on board the NISAR platform. The expected range delay can be computed from TEC estimates as

$$\partial r = K\frac{TEC}{f^2}$$

Where K=40.31 $m^3.s^{-2}$ is a constant, f is the radar carrier frequency and TEC is the ionospheric Total Electron Content along the radar slant range. In the next section we elaborate on the approach for converting the GNSS-based vertical TEC to slant TEC in the radar LOS direction. Ionospheric delay dominates the range delay of L-band NISAR data. A global analysis of GNSS-based TEC data during the last solar maxima in 2014 demonstrates that the L-band data from NISAR descending orbits (~6 pm local time) can potentially experience ionospheric range delay up to ~20 m in some regions close to the equatorial belt (Yunjun et al, 2022). The experimental results have demonstrated that using GNSS-based TEC data for correcting the ionospheric range delay can reduce the ionospheric range delay geolocation error to less than 2 m in NISAR L-band data. This error can be even further reduced when we correct for topside TEC which is the TEC between the NISAR orbit and the GNSS satellites orbit.

NISAR RSLC products already account for dry tropospheric delay using a static model as explained in section 3.8.4 of this document. By correcting for ionospheric delay and hydrostatic tropospheric delay, the absolute geolocation error of the NISAR GSLC products can reach less than 2 m excluding the elevation errors in DEMs used for the geocoding.

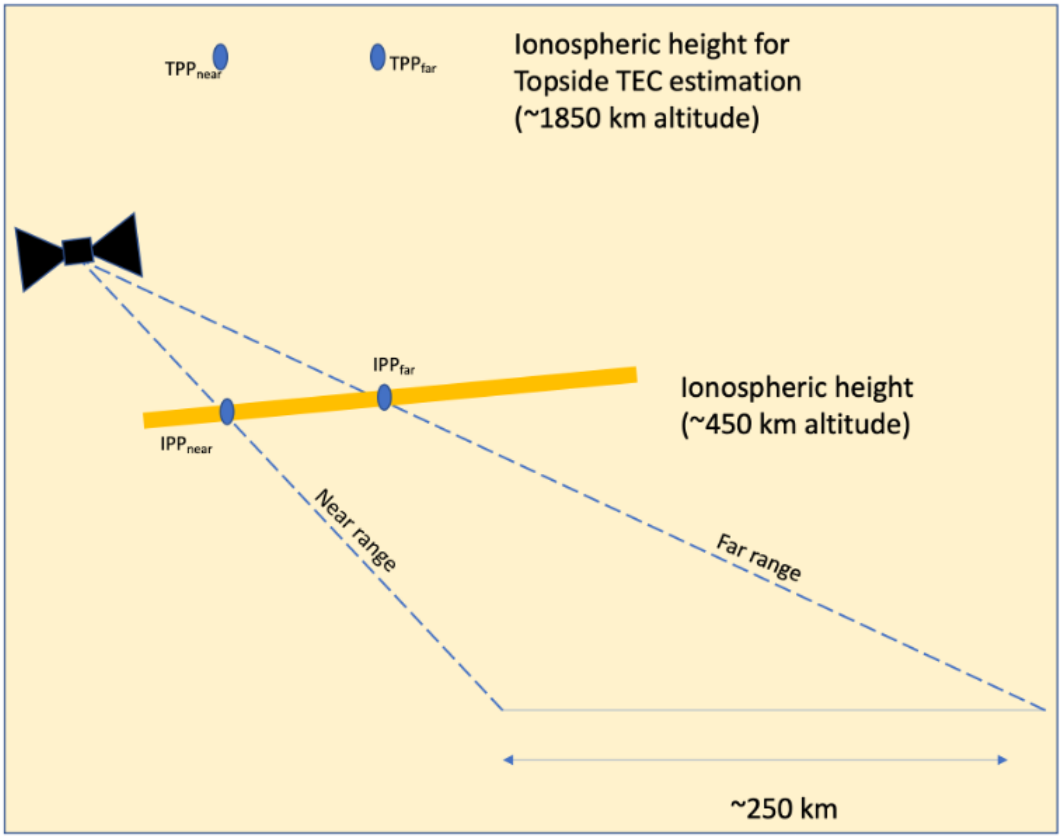

TPP$_{near}$
TPP$_{far}$
Ionospheric height for Topside TEC estimation (~1850 km altitude)
IPP$_{far}$
IPP$_{near}$
Ionospheric height (~450 km altitude)
Near range
Far range
~250 km

Figure 4-4 SAR acquisition geometry (right-looking) in the two-dimensional zero Doppler plane across the satellite track. The Total-TEC is provided at IPPnear and IPPfar and Topside-TEC provided at TPPnear and TPPfar

4.4 Ionospheric range delay correction

The NISAR GSLC product will use the NISAR TEC products to correct the ionospheric range delay. The NISAR TEC product contains two main components:

1- Total-TEC: vertical TEC between GNSS satellite at altitude around 20200 km and ground surface along a line passing through Ionospheric Piercing Point (IPP) at an altitude of ~450 km and normal to the surface of the WGS84 G1762 ellipsoid which is the reference ellipsoid for the NISAR orbit ephemeris.

2- Topside-TEC: vertical TEC between GNSS satellites altitude and SAR satellite altitude along a line passing through Topside Piercing Point (TPP) and normal to WGS84 G1762 ellipsoid.

In order to capture the first order variation of the ionospheric TEC across the NISAR track, each of total-TEC and topside-TEC are provided in two piercing points in near and far ranges. Figure 4-4 shows the IPP points at near and far ranges at ionospheric height of ~ 450 km at which the Total-TEC is provided. Similarly the Topside-TEC is provided at TPPnear and TPPfar which are assumed to be at the same latitude and longitude of IPPnear and IPPfar but at the altitude of ~1850 km where the topside-TEC is estimated at.

Starting from a given pixel of the geocoded grid of GSLC at location (x,y) we first compute the range-Doppler coordinates reporsented by azimuth time and geometrical slant range at (t_{az}, rng). The computed slant range, i.e. rng, is potentially biased by the ionospheric delay.

We then interpolate the GNSS-based NISAR TEC product as follows to estimate TEC for the range and azimuth coordinates of interest. Assuming the geometrical azimuth time and slant range at (t_{az}, *rng*) the GSLC algorithm follows the following steps to estimate the TEC:

1- Extract TEC for 1-2 minutes along the orbit with samples before and after the point of interest

2- Compute the sub-orbital TEC by subtracting the Topside-TEC from Total-TEC at each of the extracted samples at near and far ranges.

3- Fit a polynomial to the sub-orbital TEC along the orbit at near ranges

4- Fit a polynomial to the sub-orbital TEC along the orbit at far ranges

5- Evaluate the near and far range TEC polynomials at t_{az}

6- Compute the cross-track sub-orbital TEC-slope at t_{az}

$$\text{slope} = (\text{TEC}_{\text{far}} - \text{TEC}_{\text{near}})/(\text{far\_range} – \text{starting\_range})$$

7- Compute TEC at the slant range of *rng* as

$$\text{Sub orbital TEC } (t_{az}, \text{rng}) = \text{slope} * (\text{rng} – \text{starting\_range}) + \text{TEC}_{\text{near}}$$

The estimated sub-orbital TEC is then used to compute the range delay. The range delay is added to the computed geometrical slant range to obtain the refined slant range.

$$rng^c = rng + dr_{iono}$$

where rng^c is the refined slant range corrected for ionospheric delay.

4.5 Interpolation

After estimating the azimuth time and refined slant range, the RSLC data will be interpolated to estimate the SLC data in the position of (t_{az}, rng^c). The interpolation is performed as follows:

1- A block of data around (t_{az}, rng^c) is extracted from the RSLC product. The block should be large enough to accommodate the support of the chosen interpolator

2- Demodulate the block of data using the Doppler centroid polynomial to baseband the complex valued data

3- Interpolate the base-banded block of data using interpolator of choice. The choice of interpolator plays a significant role in determining the quality of the final interferometric products [Hanssen and Bamler, 1999]. The NISAR SAS uses a 16-point truncated sinc for resampling all SLCs.

4- Compute the azimuth carrier at the interpolated pixel location (t_{az}, rng^c) from the Doppler centroid Polynomial, and add it back to the interpolated value

5- Compute the geometrical phase corresponding to slant range between the target and the radar phase center as

$$\delta\varphi = \frac{4\pi}{\lambda} rng$$

Note that we use the slant range before ionospheric delay correction (rng) to compute the geometrical phase. This way the phase of the GSLC will be untouched in term of ionospheric delay and the consistency of GSLC products from different frames along the satellite orbit will be maintained

6- Remove the geometrical phase from the interpolated phase

5 INTERFEROMETRY PRODUCTS

5.1 Introduction

The RIFG, ROFF, RUNW, GOFF, and GUNW NISAR interferometric products are generated from two RSLC products using the following processing steps reported below and outlined in Figure 5-1:

- Bandpass filtering
- Coarse coregistration
- Dense cross-correlation
- Rubber-sheeting
- Resampling
- Coregistered RSLC cross-multiplication
- Interferometric phase filtering
- Phase unwrapping
- Geocoding
- Ionospheric phase screen estimation
- Tropospheric phase screen estimation

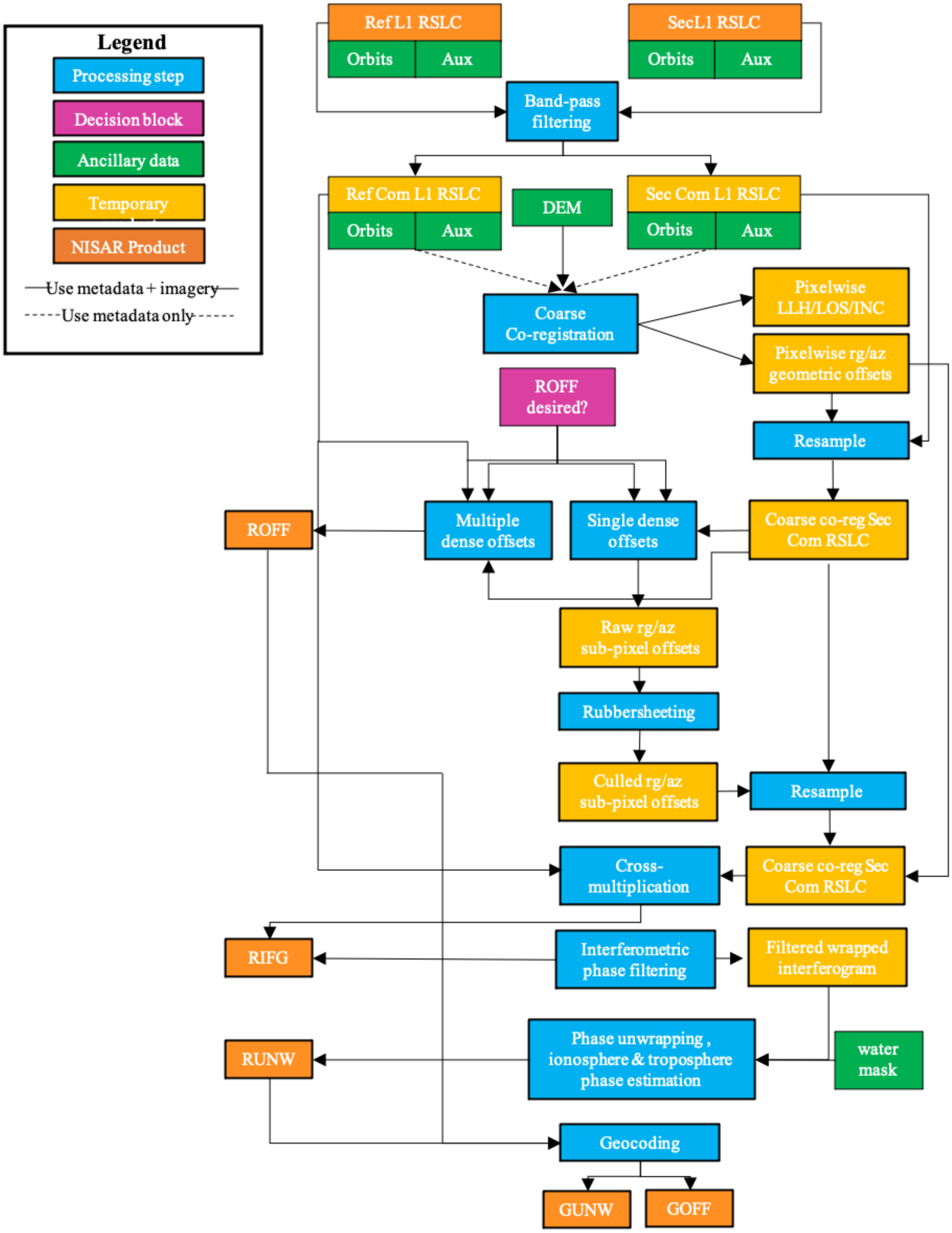


Figure 5.1. Interferometric product generation workflow

5.2 Bandpass filtering

Conventional SAR interferometry assumes that the input reference and secondary RSLCs share a common slant range and azimuth bandwidths, so that each resolution cell is characterized by the same scattering contribution [Simons & Rosen, 2007]. However, the input RSLCs for the NISAR L-band InSAR processing may exhibit different but overlapping slant range spectra. In such a case, the full-band reference and secondary RSLCs are bandpass-filtered to produce a new reference and/or secondary RSLC exhibiting a common slant range spectrum. During the bandpass filtering step, the slant range spectra of the full-band RSLCs are weighted in range using a Kaiser-Bessel window, as also used during the RSLC focusing step. To avoid band-passed RSLCs with nonuniform spectral weighting along slant range, a de-weighting operation is performed. The generated common slant range bandwidth RSLCs are then used in the downstream steps of the NISAR InSAR L-band processing chain.

5.3 Coarse-coregistration

The NISAR L-band baseline InSAR processing aligns the input RSLCs using a two steps coregistration algorithm [Yague-Martinez et al., 2010]. The first step of the algorithm consists in a geometry-based coregistration. Using an external DEM and the RSLCs orbit ephemeris, the geometry-based coregistration computes a regular grid of slant range and azimuth offsets of the secondary RSLC with respect to the reference RSLC [Sansosti, 2006; Yague-Martinez, 2010]. In a second step, described in Section 5.4, the geometry-based coregistration is further refined using incoherent cross-correlation performed on reference and coarse-coregistered secondary RSLC patches arranged on a grid [Michel et al., 1999].

The implementation of the geometry-based coregistration is based on the approach described in [Sansosti et al., 2006]. The algorithm can be broken down into the following steps:

1. For each pixel in the reference RSLC image, estimate the position on the ground using an external DEM, the reference RSLC orbit ephemeris, and a forward geometry mapping algorithm (see Section 3.5.1).
2. For each estimated ground coordinate from step-1, estimate the slant range and azimuth times in the secondary RSLC using the secondary RSLC orbit ephemeris and an inverse geometry mapping (see Section 3.5.2).

The geometry-based coregistration step generates high resolution slant range and azimuth offset maps. These offset maps accurately track the terrain topography and are usually accurate within a pixel [Sansosti et al., 2006]. The pixel-by-pixel slant range and azimuth offset maps are used to coarsely align the secondary RSLC in the geometry of the reference RSLC.

The geometry-based coregistration algorithm exploits only the reference and secondary RSLC metadata (e.g., timing information, orbit ephemeris) i.e., no SAR imagery is used in this coregistration step.

Consequently, these estimates are susceptible to relative timing errors between the reference and secondary RSLC. The impact of the absolute timing errors for the reference RSLC is ignored as these timing errors are expected to be less than 10 cm in the along-track direction.

5.4 Dense cross-correlation

The geometry-based coregistration allows to achieve a coregistration accuracy on the order of a pixel and it is usually sufficient to preserve interferometric correlation over regions of the world characterized by negligible surface deformation [Simons & Rosen, 2007]. On the contrary, this approach can cause interferometric phase aliasing over areas with fast-moving surfaces (e.g., polar ice sheets) and high strain rates (e.g., earthquakes) [Joughin, 2002; Rignot, 2011]. Over these areas, tracking pixel motion with incoherent cross-correlation [Michel, 1999] can help restitute interferometric phase correlation [Rignot, 2011]. In addition, incoherent cross-correlation can contribute to refine the accuracy of a geometry-based coregistration to a sub-pixel level [Michel et al., 1999; Simons & Rosen, 2007; Yague-Martinez et al., 2010; Fattahi et al., 2017].

The implementation of the incoherent cross-correlation algorithm follows the approach described in [Shafer et al., 1994; Zhu et al., 2022]. Its main processing steps are reported below and outlined in Figure 5-2.

1. Select a chip of shape $w_r \times w_a$ in the reference RSLC as an image template, where w_r is width in range direction and w_a is the length in azimuth direction. Each pixel of the template is a complex number of the RSLC data
2. Oversample the template by a factor of 2 to avoid aliasing in the cross-multiplication. This operation allows to have a default half-pixel resolution in the estimation of the offset
3. Take the amplitude of each pixel in the oversampled image template
4. Select a larger chip in the secondary RSLC (i.e., search template) around the location of the reference chip. The shape of the search template is $(w_r + 2 \times s_r) \times (w_a + 2 \times s_a)$.
5. Repeat Step #2 and #3 to get the oversampled amplitude of the search template
6. Perform the normalized cross-correlation between the reference and secondary chips for each offset in slant range and azimuth direction.
7. Find the cross-correlation maximum and its location
8. Choose a small window (i.e., zoom window [Shafer et al., 1994]) around the cross-correlation maximum and oversample the cross-correlation surface by a factor (e.g., 64). This procedure aims to gain sub-pixel resolution of the offsets. For example, with a correlation surface oversampling factor of 64, the offset resolution is on the order of 1/128 pixel accounting for an anti-aliasing oversampling of a factor of 2.
9. Find the oversampled cross-correlation maximum and its location.
10. Repeat this procedure for all the reference chips and locations. Typically, there are millions of chip pairs to be cross-correlated. The density of the estimated offset field is governed by the spacing between neighboring reference chips along the slant range and the azimuth i.e., the skip window. Smaller skip windows along slant range and azimuth results in the denser estimated slant range and azimuth offsets.

This procedure generates high-resolution maps of sub-pixel slant range and azimuth offsets that when added to the offset maps obtained from the geometry-based coregistration allow to precisely resample the secondary RSLC on the geometry of the reference RSLC .

The implementation of the cross-correlation algorithm supports [Zhu, 2022]:

1. Computation of the cross-correlation in the frequency or spatial domain of the signal
2. Oversampling/interpolation based on FFT or sinc interpolation kernel

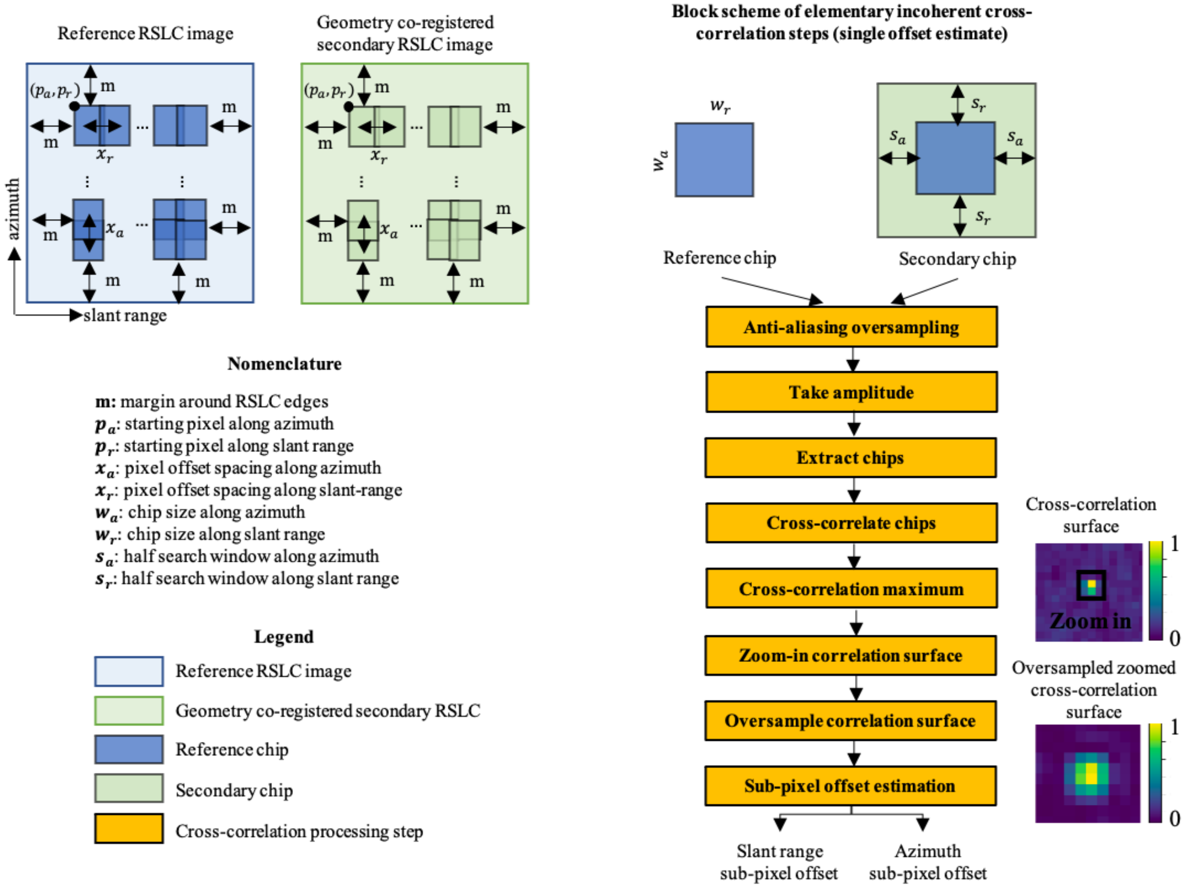

Reference RSLC image
Geometry co-registered secondary RSLC image
(p_a, p_r)
m
x_r
x_a
azimuth
slant range
Nomenclature
m: margin around RSLC edges
p_a: starting pixel along azimuth
p_r: starting pixel along slant range
x_a: pixel offset spacing along azimuth
x_r: pixel offset spacing along slant-range
w_a: chip size along azimuth
w_r: chip size along slant range
s_a: half search window along azimuth
s_r: half search window along slant range
Legend
Reference RSLC image
Geometry co-registered secondary RSLC
Reference chip
Secondary chip
Cross-correlation processing step
Block scheme of elementary incoherent cross-correlation steps (single offset estimate)
w_r
w_a
s_r
s_a
Reference chip
Secondary chip
Anti-aliasing oversampling
Take amplitude
Extract chips
Cross-correlate chips
Cross-correlation maximum
Zoom-in correlation surface
Oversample correlation surface
Sub-pixel offset estimation
Slant range sub-pixel offset
Azimuth sub-pixel offset
Cross-correlation surface
Zoom in
1
0
Oversampled zoomed cross-correlation surface
1
0

Figure 5-2. Details of the incoherent cross-correlation algorithm.

The offset spacing, the chip size, and the correlation surface oversampling directly affect the computational performance of the incoherent cross-correlation i.e., offset grids with denser spacings and/or big chip sizes and/or high surface correlation oversampling require more time to be processed [Shafer et al., 1994]. In addition, the accuracy of the incoherent cross-correlation is also affected by the shape of the chip size and the interferometric phase quality and it can be expressed as [De Zan, 2013]:

$$\sigma_I = \sqrt{\frac{3}{2N}}\sqrt{\frac{2+5\,\gamma^2-7\gamma^4}{\pi\gamma^2}}$$

Where σ_I represents the accuracy of the incoherent cross-correlation normalized to the resolution element, N is the number of samples in a chip while γ is the magnitude of the normalized interferometric correlation. A coarse offset grid (offset spacings on the order of 200 m x 200 m) with a chip size of 64x64 pixels is usually sufficient to refine the accuracy of the geometry-based coregistration [Yague-Martinez et al., 2010]. On the contrary, denser offset grids (e.g., 90 m x 90 m) with bigger chip sizes (e.g., 128 x128 pixels) are generally required to achieve a good coregistration accuracy over fast-deforming areas or over regions of the world characterized by strong ionospheric phase gradients [Joughin, 2002; Rignot et al., 2011].

5.4.1 Offset product generation

Running the cross-correlation algorithm with a set of hierarchically increasing chip sizes and search windows allows to generate the offset products ROFF and its geocoded version GOFF [Joughin, 2002]. The dense offset layers contained in ROFF and GOFF share the same starting pixel and the same spacing in both slant range and azimuth directions [Joughin, 2002]. The generated offset layers in ROFF and GOFF share the same matching centers at each location and they are stackable i.e., offset layers can be blended easily without any interpolation. All the produced layers are intended to be raw i.e., not corrected for offset outliers and/or not low-pass filtered, no post-processing operations [Joughin, 2002].

5.5 Rubber-sheeting

The slant range and azimuth offsets produced by the cross-correlation algorithm may be affected by outliers and noise [Joughin, 2002]. During the rubber-sheeting step, inaccurate offset estimates are removed and replaced with newly computed estimates in a neighborhood [Mouginot, 2012]. The implementation of the rubber-sheeting algorithm consists in the following steps:

1. At each offset location, apply a median filter with a kernel size of the same shape of the features that need to be removed (e.g., 9x9) [Mouginot et al., 2012].
2. Remove all the offsets having an absolute median deviation over a certain threshold (e.g., 1-3 standard deviations)
3. Iteratively fill outlier locations with offset estimates derived by averaging valid offset estimates in the neighborhood of the outlier location
4. Filter the culled and filled offset maps with a moving average filter with a kernel size of 5x5 in slant range and azimuth directions.
5. The culled and filtered slant range and azimuth offsets are then interpolated on the same grid of the corresponding geometric offsets using a bilinear interpolation.
6. The sum of geometry and culled dense offsets is then used to resample the original secondary (not coarsely coregistered) RSLC onto the same geometry of the reference RSLC.

In cases where an offset product ROFF is available, the rubber-sheeting algorithm blends the available offset layers at different resolutions. Starting from the offset layers at the finest resolution (i.e., with the smallest chip size), the implementation of the rubber-sheeting step follows a pyramidal filling algorithm consisting the following steps:

1. Identify offset outliers using a median filter and median absolute deviation thresholding
2. Fill each outlier location by using the offset layers at coarser resolution. For each outlier location
    a. Select a neighborhood in coarser offset layers centered at the same location of the offset outlier
    b. Average all the valid offset estimates in the neighborhood and assign this value to the outlier location in the offset layer at fine resolution
3. Average offset estimates of the different multi-resolution layers for all the locations containing valid offset estimates.
4. Filter the culled and filled high resolution offset maps with a moving average filter with kernel size of 5x5 in slant range and azimuth directions

5.6 Resampling

After refining the range and azimuth offsets as discussed in the rubber-sheeting step, the original secondary (not coarsely coregistered) RSLC is resampled to the range-Doppler grid of the reference RSLC. For a single pixel, the implementation of the resampling algorithm can be broken down into the following steps:

1. Read a block of data in the secondary RSLC in the neighborhood of the pixel of interest. The neighborhood should be large enough to support the chosen interpolator which is a 16-point truncated sinc for resampling RSLC.
2. Base-band (i.e., demodulate) the data block using the Doppler centroid look up table.
3. Interpolate the base-banded data block using the interpolator.
4. Compute the azimuth carrier at the interpolated pixel location from the Doppler centroid look up table and add it back to the interpolated value.

5.7 Coregistered RSLCs cross-multiplication

The complex wrapped interferogram is generated by cross-multiplying the reference RSLC with the finely coregistered secondary RSLC. The implemented algorithm can be broken down into the following steps [Davidson & Bamler, 1999]:

1. Up-sample the filtered reference and coregistered secondary RSLCs by a factor of two in slant range
2. Cross-multiply reference and coregistered RSLC and decimate the output of the cross-multiplication by a factor of two in range by averaging to generate the full resolution interferogram.

3. The interferometric phase is then flattened for the topographic and flat earth phase by removing the phase corresponding to the geometrical range offsets between the reference and secondary images as

$$I_{flattened} = I \times e^{-\frac{4\pi}{\lambda}\delta r}$$

The wrapped complex interferogram and the power of the reference and coregistered secondary SLCs are multi-looked to the desired posting. Afterwards, the magnitude of the normalized interferometric correlation, γ, is computed as

$$\gamma = \left| \frac{I_m}{\sqrt{R_m}\sqrt{S_m}} \right|$$

where I_m is the flattened and multilloked complex wrapped interferogram and R_m and S_m are the multi-looked power of the reference and coregistered secondary images, respectively.

For the RIFG product, the WGS84 ellipsoid is used as input for flattening and cross multiplication while the geometrical phase (including the topographic phase) is not removed from the generated complex interferogram.

5.8 Interferometric phase filtering

Flattened, multi-looked interferograms might be filtered to reduce phase noise before phase unwrapping. The baseline plan for the InSAR workflow is to use linear filters to filter the wrapped interferogram. The InSAR workflow supports different linear filter kernels e.g., moving average, Gaussian, and median filter.

5.9 Phase unwrapping

This section describes the baseline plan to perform interferometric phase unwrapping. Alternative approaches are being explored in an effort to improve the utilization of computational resources (e.g. processing time and memory usage).

The absolute interferometric phase ϕ_{abs} is the unambiguous phase proportional to the differential range between the interferometric pair. It is given by

$$\phi_{abs} = \frac{4\pi}{\lambda}\delta r$$

where λ is the radar wavelength and δr is the one-way path length difference between the two antenna and target positions [Simons & Rosen, 2007].

Since only the principal values of the phase function can be directly measured from the complex argument of the interferogram, the measured phase values are known only modulo 2π. The measured (“wrapped”) phase ψ is

$$\psi = \mathcal{W}\{\phi_{abs}\} = mod(\phi_{abs} + \pi, 2\pi) - \pi$$

where $\mathcal{W}\{\cdot\}$ is the wrapping operator and

$$-\pi \ \leq \psi < \pi.$$

The phase unwrapping process seeks to determine the relative phase between points within the interferogram in order to reconstruct the absolute phase values up to an overall constant multiple of 2π. The goal of phase unwrapping is to determine the two-dimensional array of unwrapped phase values ϕ_{unw} such that

$$\phi_{unw}(i, j) = \phi_{abs}(i, j) + 2\pi n$$

where $\phi_{unw}(i, j)$ is the unwrapped phase value of the point (i, j) in the sample grid, $\phi_{abs}(i, j)$ is the corresponding absolute phase value, and n is a fixed integer value.

An additional post-processing step, absolute phase determination, may be performed to estimate the offset multiple of 2π and reconstruct the absolute phase unambiguously [Simons & Rosen, 2007]. This step is not performed by the NISAR L-SAR processor. The unwrapped products RUNW and GUNW represent the unwrapped phase with respect to arbitrary, unspecified reference points. For notational convenience, throughout the remainder of this section, we refer to the unwrapped phase ϕ_{unw} as simply ϕ.

Phase unwrapping generally consists of two steps: (1) estimation of the discrete unwrapped phase gradients with respect to the sample grid, and (2) integration of the estimated unwrapped gradients to obtain the unwrapped phase surface.

We define the *wrapped* phase gradients as the discrete first-order differences between the measured phase values, wrapped to the interval $[-\pi, \pi)$,

$$\Delta\psi^{(i)}(i, j) = \mathcal{W}\{\psi(i + 1, j) - \psi(i, j)\}$$

$$\Delta\psi^{(j)}(i, j) = \mathcal{W}\{\psi(i, j + 1) - \psi(i, j)\}.$$

For much of the interferogram, the unwrapped phase gradients are expected to be equal to the wrapped phase gradients. However, due to decorrelation and undersampling, there may be regions where the unwrapped phase gradients differ from their wrapped counterparts [Bamler et al., 1998]. The main challenge for phase unwrapping algorithms is to determine the discrepancies between the unwrapped and wrapped phase gradients.

5.9.1 Primary phase unwrapping algorithm

The baseline plan for NISAR L-SAR processing is to use the statistical-cost network-flow algorithm for phase unwrapping (SNAPHU) [Chen & Zebker, 2001].

SNAPHU formulates the phase unwrapping problem as a maximum *a posteriori* (MAP) estimation problem. Its goal is to determine the set of unwrapped phase gradient values $\nabla\Phi$ that maximize the posterior distribution

$$f_{\nabla\Phi}(\nabla\Phi|\nabla\Psi)$$

given the set of wrapped phase gradients $\nabla\Psi$.

Individual unwrapped phase gradients are assumed to be statistically independent, given their wrapped counterparts, such that

$$f_{\nabla\Phi}(\nabla\Phi|\nabla\Psi) = \prod_{(i,j)} \quad f_{\Delta\phi^{(i)}}(\Delta\phi^{(i)}(i,j)|\Delta\psi^{(i)}(i,j)) * \prod_{(i,j)} \quad f_{\Delta\phi^{(j)}}(\Delta\phi^{(j)}(i,j)|\Delta\psi^{(j)}(i,j)).$$

Because the natural logarithm is a monotonic function, the maximization problem can be replaced by an equivalent minimization problem by taking the negative logarithm

$$\min_{\Delta\phi^{(i)},\Delta\phi^{(j)}} \sum_{(i,j)} \quad -log\,[f_{\Delta\phi^{(i)}}(\Delta\phi^{(i)}(i,j)|\Delta\psi^{(i)}(i,j))] + \sum_{(i,j)} \quad - log\,[f_{\Delta\phi^{(j)}}(\Delta\phi^{(j)}(i,j)|\Delta\psi^{(j)}(i,j))].$$

Different statistical models may be applied for the probability density functions (PDFs) $f_{\Delta\phi^{(i)}}$ and $f_{\Delta\phi^{(j)}}$ depending on whether the underlying interferometric phase signal represents topography or deformation. Some models incorporate other ancillary information such as the average SLC intensity I and the sample coherence γ. Thus, we may replace the posterior distributions $f_{\Delta\phi^{(i)}}$ and $f_{\Delta\phi^{(j)}}$ above with

$$f_{\Delta\phi^{(i)}}(\Delta\phi^{(i)}(i,j)|\Delta\psi^{(i)}(i,j), I(i,j), \gamma(i,j))$$

$$f_{\Delta\phi^{(j)}}(\Delta\phi^{(j)}(i,j)|\Delta\psi^{(j)}(i,j), I(i,j), \gamma(i,j)).$$

The minimization is subject to constraints which enforce that the unwrapped phase gradients represent a conservative vector field.

5.9.1.1Network flow optimization for phase unwrapping

The phase unwrapping problem can be posed as an equivalent network flow optimization problem in which we seek to minimize the total cost of transmitting flow of some commodity along a network from nodes with net supply to nodes with net demand [Costantini, 1998].

A flow network is a directed graph $G = (V, E)$ defined by a set of vertices, or nodes, $v \in V$ and directed edges, or arcs, $(u, v) \in E$. Each arc is associated with a flow x_{uv} and a cost function $c_{uv}(x_{uv})$ that represents the total cost of transmitting x_{uv} units of flow along arc (u, v). In the classical Minimum Cost Flow (MCF) problem, the total cost of flow along an arc is assumed to vary linearly with the amount of flow on that arc. However, in order to accomodate the nonlinear cost functions of SNAPHU, we consider costs that are arbitrary functions of the flow quantity. Flows are restricted to be nonnegative. Bidirectional flows can be represented by a pair of antiparallel directed arcs between adjacent nodes. Each node is associated with a value b_v representing its supply/demand. Positive values of b_v represent surpluses and negative values represent demands.

In the network flow representation of the phase unwrapping problem, nodes in the network represent 2×2 residue loop integrals. Each node stores a supply/demand equal to the value of the integral (either 1, 0, or -1). Directed arcs form a grid-like lattice connecting neighboring nodes. Arcs are associated with phase gradients in the original data. Flow on a particular arc represents the deviation, in cycles of 2π, between the corresponding unwrapped and wrapped phase gradients [Chen & Zebker, 2000].

The uncapacitated network flow cost minimization problem with generalized costs is

$$\min_{x} \sum_{(u,v)\in E} c_{uv}(x_{uv})$$

subject to

$$\sum_{v:(u,v)\in E} x_{uv} - \sum_{v:(v,u)\in E} x_{vu} = b_u, \qquad \forall\, u \in V$$

$$x_{uv} \geq 0,\ \forall\, (u,v) \in E.$$

The total cost of flow in all arcs of the network is minimized, subject to flow conservation and nonnegativity constraints. The flow conservation constraint states that the difference between the total outgoing flow and total incoming flow of each node must be equal to the supply/demand of the node. This constraint is equivalent to the requirement that the unwrapped phase gradient field must be residue-free. The nonnegativity constraint ensures that flow quantities must not be less than zero.

A desirable property of phase unwrapping algorithms is that the unwrapped and wrapped phase values should be *congruent*. That is, they should differ only by integer multiples of 2π. Within the network optimization context, this property can be enforced by restricting the flow network variables to take on only integer values. Due to the unimodularity property of the node-arc incidence matrix of G, the constrained network flow minimization problem always has at least one integer-valued optimal solution when the underlying data are integral. The integral optimal solutions are the *basic* solutions – those that correspond to a basis of the incidence matrix [Ahuja et al., 1993]. In the field of network flow optimization, there are efficient strategies for finding such solutions which can be adapted for use within the SNAPHU algorithm.

5.9.1.2Cost function

The NISAR L-SAR processor uses the “smooth” cost option of SNAPHU, which models the unwrapped phase as a smoothly-varying surface in the presence of additive Gaussian-distributed phase noise. The cost function is

$$c^{(i)}(i,j) = \frac{|\Delta\phi^{(i)}(i,j) - \mu_{\nabla\psi}(i,j)|^2}{{\sigma_\psi}^2(i,j)}$$

$$c^{(j)}(i,j) = \frac{|\Delta\phi^{(j)}(i,j) - \mu_{\nabla\psi}(i,j)|^2}{{\sigma_\psi}^2(i,j)}$$

where $\mu_{\nabla\psi}$(i,j) is the local mean of the wrapped phase gradients within a rectangular window about the point (i,j), and σ_ψ is the interferometric phase standard deviation estimated from the sample coherence and effective number of looks. The L-SAR processor uses a 7×7 window for averaging the wrapped phase gradients. σ_ψ is approximated via a simple model that was fit offline to curves provided by Rodriguez and Martin [1992] and Lee et al. [1994].

Intuitively, the cost function penalizes unwrapped phase gradients that are large in magnitude relative to the local variability of the phase values.

The cost values are scaled and quantized to integer values in order to ensure that the network flow optimization problem has integer-valued solutions. The integer costs are

$$C^{(i)}(i,j) = floor(K\, c^{(i)}(i,j))$$

$$C^{(j)}(i,j) = floor(K\, c^{(j)}(i,j))$$

where K is the scale factor (set to 100 in the L-SAR processor).

5.9.1.3Residue calculation

Residues are detected in the wrapped phase gradient field by evaluating the sum of the wrapped phase differences in clockwise loops around each 2×2 group of phase values (Goldstein et al., 1988]. The residue calculation is

$$residue(i,j) = round(\frac{1}{2\pi}[\Delta\psi^{(j)}(i,j) + \Delta\psi^{(i)}(i,j+1) - \Delta\psi^{(j)}(i+1,j) - \Delta\psi^{(i)}(i,j)])$$

where $round(\cdot)$ indicates rounding to the nearest integer.

The residue calculation results in three possible values: 1 (a positive residue), 0 (no residue), or -1 (a negative residue). Positive and negative residues indicate inconsistencies with respect to the assumption that the phase varies by no more than one half cycle between adjacent samples.

5.9.1.4Initialization

The initialization step seeds the network with an initial feasible flow that can be iteratively refined by the network flow minimizer. Since the objective function is non-convex, the choice of initialization may affect the final converged result of the solver.

SNAPHU provides options for initializing the network via a Minimum Cost Flow (MCF) solver or Minimum Spanning Tree (MST) solver. The NISAR L-SAR processor uses the MCF initializer, which selects an initial flow that minimizes the weighted total number of extra phase cycles added to the wrapped phase gradients.

The Minimum Cost Flow initializer uses the cost-scaling algorithm of Goldberg [1997].

5.9.1.5Network solver

The SNAPHU implementation uses the hybrid generalized-cost solver algorithm described by Chen & Zebker [2001], based on the network simplex algorithm for the MCF problem [Dantzig, 1951]. Starting from an initial feasible flow (a flow that meets the flow conservation constraints at each node), the algorithm iteratively improves the cost of the current flow while maintaining feasibility.

Since the cost surface is non-convex, the solver only approximately minimizes the cost by arriving at a locally optimal solution. Finding an exact solution that globally minimizes the cost is computationally intractable due to the *NP*-hardness of the network flow optimization problem with generalized costs [Chen & Zebker, 2001].

5.9.1.6Phase gradient integration

After the unwrapped phase gradients have been estimated, the final step to produce the unwrapped phase is to integrate the unwrapped phase differences, starting from some reference point. Since the unwrapped phase gradients are a conservative vector field, the integral is path-independent. Any scanning order that corresponds to a spanning tree of the original sample grid yields an equivalent unwrapped phase surface, up to a constant offset.

5.9.1.7Connected component labeling

In addition to the unwrapped phase array, the algorithm also produces a mask layer of connected components. Each connected component is a contiguous region of points in the unwrapped phase that is believed to have been reliably unwrapped in a self-consistent manner. Connected components are assigned unique positive integer labels. Invalid unwrapped phase values are assigned a label of 0, indicating that they could not be reliably unwrapped and are not members of any connected component.

Connected components are determined according to a region growing strategy, using the statistical cost functions to determine the boundaries of reliably unwrapped regions as described by Chen & Zebker [2002]. Given the probabilistic interpretation of SNAPHU's cost functions, the residual cost of adding or subtracting flow to an arc in the network relative to the unwrapped solution can be used as a measure of the statistical model's confidence in the unwrapped phase gradient. If the cost of augmenting flow along a particular arc is high then the model ascribes a low probability of error to the associated unwrapped phase gradient estimate.

The incremental costs $\Delta c^{(+)}$ and $\Delta c^{(-)}$ for flow increments $\delta = +1$ and $\delta = -1$ respectively are defined as

$$\Delta c = c(x_0 + \delta) - c(x_0)$$

where x_0 is the arc flow in the unwrapped solution.

Since lower incremental costs are associated with higher probability of unwrapping errors, we assign each arc a scalar cost c_s equal to the lesser of its two incremental costs,

$$c_s = min(\Delta c^{(+)}, \Delta c^{(-)}).$$

Regions are grown outward, starting from arbitrarily chosen seed pixels. Each region is allowed to expand by successively adding adjacent non-region pixels if they can be reached without crossing an arc whose scalar cost c_s is below some threshold. When no more pixels can be added to the current region, a new seed pixel is chosen from among the set of unvisited pixels and the region-growing procedure is repeated. This process continues until no more unvisited elements remain. Any resulting regions whose area is smaller than a predefined minimum size are deemed unreliable and subsequently masked out.

5.9.2 Alternative phase unwrapping algorithm

An alternative phase unwrapping algorithm under consideration is the ICU algorithm. ICU is a modification of the branch-cut algorithm of Goldstein et al. [1988] that attempts to construct branch cuts, or lines that cannot be crossed during the unwrapping process without resulting in phase inconsistencies in the unwrapped phase field. Once the lines are identified, the phase is unwrapped by integrating the phase differences within the delimited regions. The algorithm searches for residues in the wrapped phase gradient field, then grows branch cuts connecting residues in a tree-like manner by sequential search of nearby residues. Low-correlation regions are excluded by setting a coherence threshold. The tree growing process is guided by neutrons placed according to heuristics based on the range phase rate and radar backscatter intensity [Madsen & Rosen, 1993].

5.10 Geocoding

The geocoding algorithm leverages the inverse geometry mapping algorithm (see Section 3.5.2). The geocoding algorithm can be broken down in the following steps:

1. Identify the bounding box in the geocoded domain for a given interferometric product and fetch the relevant DEM.
2. Interpolate the DEM to the posting corresponding to the desired output.
3. For each pixel in the geocoded grid, estimate height by interpolating DEM at that point using a biquintic interpolation.

4. From the geocoded pixel, perform inverse mapping to estimate the corresponding range and azimuth coordinates in radar geometry.
5. Interpolate the raster layer in radar geometry at the estimated range and azimuth location and assign the value to the corresponding pixel in the geocoded domain.

The NISAR L-SAR InSAR processing uses a truncated sinc interpolator for interpolating complex wrapped interferogram products and bilinear interpolator for unwrapped phase, interferometric coherence, and dense offsets layers. For the interpolation of mask layers (e.g., layover/shadow, water mask), the InSAR workflow uses a nearest neighbor interpolator.

5.11 Interferogram correction layers

5.11.1 Ionospheric Phase Screen Estimation

Repeat-pass interferometric SAR measurements are typically affected by changes in the microwave propagation through the Earth's ionosphere [Rosen et al., 2010]. This differential ionospheric signal introduces, among other effects (e.g. Faraday rotation) a phase delay in repeat-pass interferograms [Rosen et al., 2010; Gomba et al., 2016]. If not adequately compensated, this spurious phase delay might drastically downgrade the accuracy of InSAR surface deformation estimates [Fattahi et al., 2017]. To mitigate the effect of the ionosphere phase delay, a common approach is to take advantage of the dispersive nature of the ionosphere, i.e., microwave propagation through the ionosphere is frequency-dependent [Rosen et al., 2010]. The baseline plan for InSAR L-band processing is to correct for the ionospheric phase screen using the range split-spectrum technique [Fattahi et al., 2017; Liang et al., 2019; Brancato & Fattahi, 2020].

The phase of a complex interferogram formed from two RSLCs acquired at times t_i and t_j can be expressed as the sum of several components:

$$\mathit{\Delta}\varphi = \frac{4\pi}{\lambda}\left(\mathit{\Delta}r_d + \mathit{\Delta}r_g + \mathit{\Delta}r_t + \mathit{\Delta}r_i + \mathit{\Delta}r_n\right)$$

where λ is the radar wavelength, $\mathit{\Delta}r_d$ is the surface displacement in the line-of-sight direction, $\mathit{\Delta}r_g$ represents the geometric range difference from radar to the target caused by a non-zero baseline between the two orbits from which the two RSLCs were acquired, and $\mathit{\Delta}r_t$ and $\mathit{\Delta}r_i$ are the tropospheric and ionospheric phase delay, respectively and $\mathit{\Delta}r_n$ is the term including all the unlisted phase noise contributions.

At a first-order, the ionospheric delay in the line-of-sight direction can be approximated as:

$$\mathit{\Delta}r_i = -\frac{K}{f_0^2}\Delta TEC$$

where *f0* is the center transmitted radar frequency, ΔTEC represents the variation of the ionospheric total electron content (TEC) along the slant range direction between the two acquisition times, and K=40.31 m^3/s^2 is a constant. Assuming that the ionosphere phase is the only dispersive component of the interferometric phase, the observed interferometric phase can be decomposed as the sum of the dispersive component dominated by the TEC variation and non-dispersive components:

$$\Delta\varphi_{f_0} = \frac{4\pi f_0}{c}\Delta r_{nd} - \frac{4\pi K}{cf_0} \quad \Delta TEC$$

where Δr_{nd} represent the non-dispersive interferometric phase component i.e., the sum of surface displacement, geometric phase, and tropospheric delay.

Most NISAR L-SAR science observations are acquired with a 20 or 40 MHz bandwidth of a main band and a 5 MHz sideband on the lower and upper parts of a total 77 MHz bandwidth. Interferograms at two different center frequencies can be formed for these main and side band acquisitions. The interferometric phase at side-band center frequency, *f1* , can be expressed in terms of the ionospheric and non-dispersive components of the interferometric phase at the main-band center frequency *f0*:

$$\Delta\varphi_{f_1} = \frac{f_1}{f_0}\Delta\varphi_{nd} + \frac{f_0}{f_1}\Delta\varphi_i$$

Thus, the phase difference between the main- and side-band can be defined as:

$$\Delta\varphi_{ms} = \Delta\varphi_{f_0} - \Delta\varphi_{f_1} = \frac{f_0 - f_1}{f_0}\Delta\varphi_{nd} + \frac{f_1 - f_0}{f_1}\Delta\varphi_i$$

The ionospheric phase screen can be estimated by utilizing the combinations of the unwrapped interferometric phases at *f0* and *f1* .

$$\Delta\varphi_i = \frac{f_1}{f_0 + f_1}\left(\Delta\varphi_{f_0} - \frac{f_0}{f_0 - f_1}\Delta\varphi_{ms}\right)$$

For NISAR L-SAR modes without the 5 MHz sideband, the slant range spectrum is band-pass filtered to sub-bands at the two ends of the slant range spectrum. The slant range bandwidth of the two sub-band images is selected to be 1/3 of the total available slant range spectrum. The sub-band SAR images from the two acquisition times are used to form sub-band interferograms. In this way, three interferograms are formed from two SAR acquisitions; one containing the entire range spectrum bandwidth and two at the lower and higher ends of the range spectrum. The low- and high-band interferograms can be combined to estimate the ionospheric phase:

$$\Delta\varphi_i = \frac{f_L f_H}{f_0(f_H^2 - f_L^2)}\left(f_H\Delta\varphi_{f_L} - f_L\Delta\varphi_{f_H}\right)$$

where f_L and f_H are the center frequencies of the low- and high-band interferograms, respectively, and $\Delta\varphi_{f_L}$ and $\Delta\varphi_{f_H}$ are the unwrapped interferometric phases at the low and high bands.

The standard deviation of the ionosphere phase estimate is approximated using interferometric phase variances. The interferometric phase variance is expressed as:

$$\sigma^2{}_{\Delta\varphi_k} = \frac{1}{2N}\frac{1-\gamma_k^2}{\gamma_k^2} \qquad k \ = \ f_L, f_H, f_0, f_1$$

where γ is the magnitude of the normalized interferometric correlation. Accordingly, the standard deviation of the estimated ionosphere phase from split-range-spectrum and main-side combination can be formulated as:

$$\sigma_{\Delta\varphi_{iono,split-band}} = \left(\frac{f_L f_H}{f_0(f_H^2 - f_L^2)}\right)\sqrt{f_L^2\sigma_{\Delta\varphi_{fL}}^2 + f_H^2\sigma_{\Delta\varphi_{fH}}^2}$$

$$\sigma_{\Delta\varphi_{iono,ms}} = \sqrt{\left(\frac{f_1^2}{f_1^2 - f_0^2}\sigma_{\Delta\varphi_{f0}}\right)^2 + \left(\frac{f_0 f_1}{f_1^2 - f_0^2}\sigma_{\Delta\varphi_{f1}}\right)^2}$$

The standard deviation of the estimated ionospheric phase is a function of the range bandwidth of the acquired data, the magnitude of the normalized interferometric correlation (reflected to the phase variance of interferograms), and the separation between the main- and side-band center frequencies or the frequency separation between the low- and high-band center frequencies. The frequency difference amplifies the interferometric noise in the estimated ionospheric phase. Therefore, the estimated ionospheric phase is usually noisy and requires low-pass filtering. As such, the estimated ionospheric phase screen only represents the low-frequency ionospheric phase while the higher frequency content is filtered out during the noise reduction process. To mitigate possible artifacts caused by low-pass filtering, an iterative masking-interpolation filtering approach is implemented [Fattahi et al., 2017]. Before applying low-pass filtering to the dispersive components, a mask layer for invalid regions (e.g. lakes, shadow, ocean) are created using the magnitude of the normalized interferometric correlation, connected components from phase unwrapping, or a data-driven approach based on the application of a median filter. Invalid regions are masked out from dispersive components and filled out with interpolated values from neighborhood pixels. After this step, a 2-D Gaussian weighted filter is applied to obtain the filtered ionospheric phase estimates. The ionosphere estimates are weighted with the inverse of their expected variance (calculated from magnitude of the normalized interferometric correlation of the interferferograms). This filtering procedure is applied iteratively. After the second iteration, the valid regions (e.g. high coherence areas) are reset with their original unfiltered values and the invalid regions are filled with filtered data.

5.11.2 Tropospheric Phase Screen Estimation

The L-SAR tropospheric phase screens are generated using the algorithms described in [Jolivet et al., 2011]. These interferogram correction layers consist of separate phase screens for the dry and wet tropospheric components, but do not attempt to capture the impact of atmospheric turbulence. The Interim analysis product from the European Center for Medium-Range Weather Forecasts (ECMWF) is used to generate the correction layers.

The tropospheric phase corrections are provided in data cubes consisting of 41 irregularly-spaced height layers from -500 m to 9000 m with each layer containing correction data at 25 x 25 km postings. The dry and wet corrections are determined by interpolating the appropriate cube using the line number, pixel number, and altitude of the interferogram point of interest.

6 GEOCODED POLARIMETRIC COVARIANCE (GCOV) PRODUCT

6.1 Introduction

The L2 Geocoded Covariance (GCOV) product is generated from the L1 RSLC product using the following general processing steps with workflow details described below (Figure 6-1):

- Generation of data, metadata, and auxiliary layers;
- Polarimetric symmetrization;
- Cross-multiplication;
- Radiometric terrain correction with the area projection algorithm (RTC-AP);
- Geocoding of data with area projection algorithm (GEO-AP);
- Geocoding of auxiliary layers.

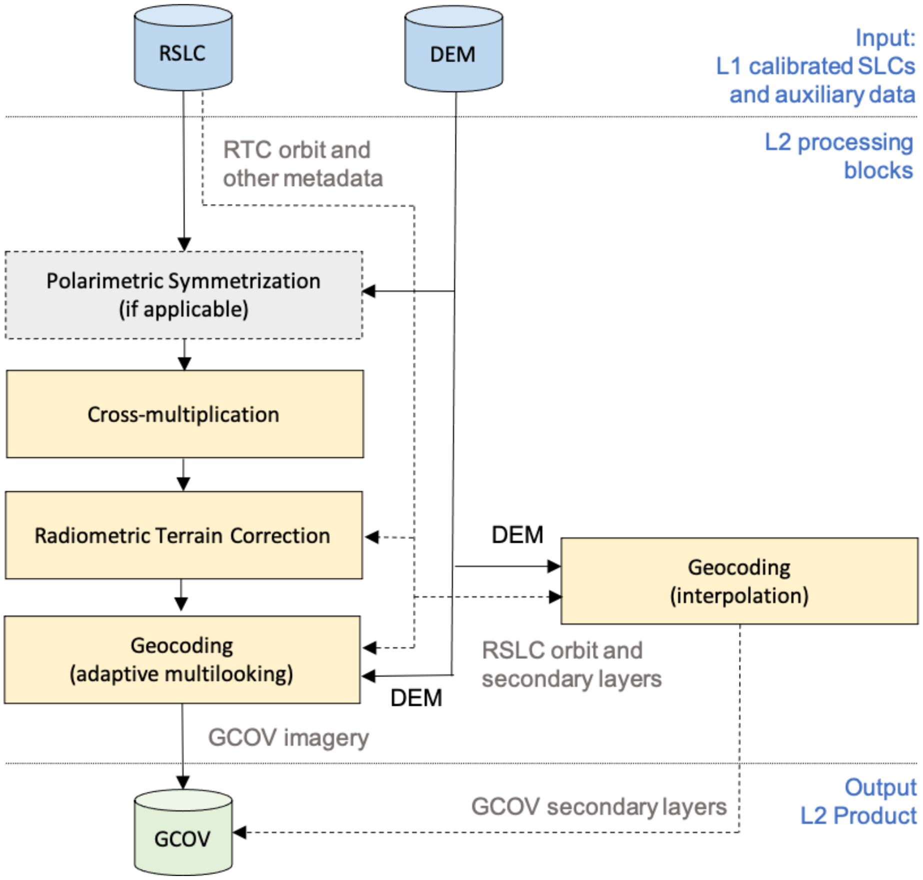


Figure 6.1-1. NISAR GCOV product generation workflow.

The input to the workflow is an RSLC product and a DEM. The input RSLC product is assumed to be calibrated by the polarization-dependent antenna pattern, range spread loss, and polarimetric cross-talk and channel imbalance [van Zyl, 1990]. All polarimetric HV/VV channels are assumed to be registered to the polarimetric HH/VH channels (half PRF azimuth shift). SAR SLC data s_{pn} for each polarimetric channel P_p can be arranged in the form of scattering vector k:

$$k = [s_{p1}, s_{p2}, \ldots, s_{pn}]^{\mathrm{T}}$$

6.2 Polarimetric Symmetrization

The polarimetric symmetrization step symmetrizes the cross-polarimetric channels VH and HV (in the linear canonical basis) forcing them to be equal due to target reciprocity [Cloude and Pottier, 1996]. Symmetrization is performed after polarimetric calibration, i.e., compensation for cross-talk and channel imbalances, expected to be performed during the L1 RSLC product generation. Symmetrization is only defined when VH and HV channels are available. The new $\underline{s}_{vh}$ value for each polarimetric calibrated RSLC image sample is replaced by the average of the s_{vh} and s_{hv} returns

$$\underline{s}_{vh} = \frac{s_{vh} + s_{hv}}{2}$$

In quad-pol mode, standard GCOV products produced by the NISAR SDS contain the 3x1 symmetrized scattering vector:

$$k_3 = [s_{hh}\ , \underline{s}_{vh}, s_{vv}]^{\mathrm{T}}$$

6.3 Cross-Multiplication

Cross-multiplication is a direct extension of the cross-multiplication step used to generate L-SAR interferograms. While interferograms are generated by cross-multiplying two SLCs of the same polarimetric channel from different datasets, the covariance matrix is generated by cross-multiplying two SLC layers of different polarimetric channels from the same RSLC product. Since the different polarimetric channels of a given frequency band in the RSLC product are already on the same range-Doppler grid, coregistration and resampling is not required before the cross multiplication. Given the scattering vector *k,* which contains SLC data at different polarizations, the covariance matrix is obtained as:

$$[C] \quad = k\, k^{*T}$$

The diagonal terms of the covariance matrix [C] are real-valued and represent the radar brightness of the scattering vector *k*. The off-diagonal terms are complex-valued and are only computed if the GCOV workflow is in the *full-covariance* mode.

The cross-multiplication of two polarimetric SLC layers operates typically line-by-line by (1) up-sampling the SLC data in range direction by a factor of two, and (2) cross-multiplying the SLCs lines to generate the associated elements of the range-Doppler covariance matrix, which also has the number of elements multiplied by two in the range direction. Because cross-multiplication is done in the time domain, up-sampling can be applied to avoid that the circular-convolution in the frequency domain affects the results of the cross-multiplication. Up-sampling can be efficiently implemented in the frequency domain by taking the Discrete Fourier Transform (DFT) and inverse DFT of the desired SLC line length (rounded up to a power of 2).

6.4 Radiometric Terrain Correction (RTC)

RTC is applied to L2 GCOV products to normalize the L1 RSLC backscatter to gamma-naught γ^0. The coefficient γ^0 is chosen to minimize the variation of the backscatter coefficient with terrain slope [Ulaby et al., 1986] as shown in Figure 6.4-1.

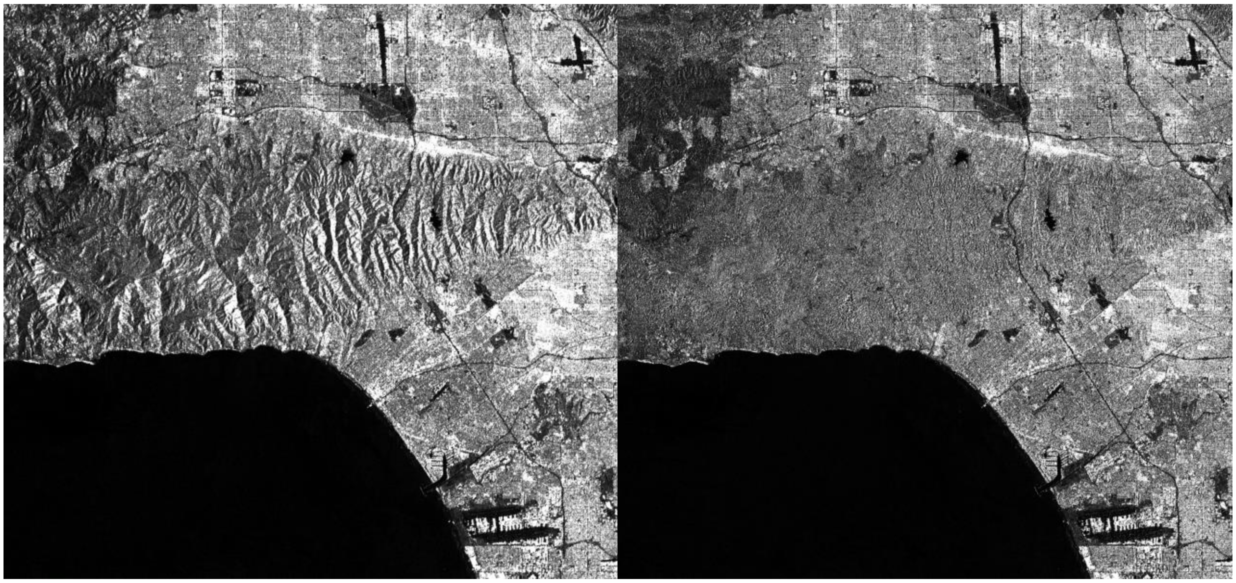


Figure 6.4-1.Geocoded (Left) beta-naught and (Right) gamma-naught backscatter (VV) from the Sentinel-1B Southern California dataset generated with the suggested workflow with area projection algorithm (ISCE3-AP). The area covers the Santa Monica Mountains in Greater Los Angeles. The radiometric terrain correction with the area projection algorithm (RTC-AP) corrects the radar backscatter for the effects of the terrain. The workflow also employs the area projection algorithm for geocoding (GEO-AP) providing an improved map projection with an adaptive multilooking. The color scale varies from 0 (black) to 0.35 (white). Credits: [Shiroma et al., 2022].

6.4.1 Introduction to Radiometric Terrain Correction

SAR radiometric correction is the process of retrieving the radar cross section (RCS) of a point target or a backscatter coefficient of a distributed target normalized to a reference area [Ulander, 1996, D. Small, 2011]. The backscatter coefficients radar brightness β^0, sigma-naught σ^0, and gamma-naught γ^0 represent the radar cross-section normalized to three reference surfaces A_β, A_σ, and A_γ, respectively [Ulander, 1996; Small, 2001, Shiroma et al., 2022], according to the acquisition geometry and the local topography as shown in Fig. 6.4-2 [Shiroma et al., 2022]. All three backscatter coefficients are unitless.

SAR SLC datasets are typically distributed as either beta-naught, β^0, or sigma-naught σ_e^0, also known as the sigma-naught-ellipsoid or ellipsoidal sigma-naught, calculated using the incidence angle θ, which is the angle between the look vector and the geodetic vertical normal to the ellipsoid, according to

~~$\sigma_e^0 = \beta^0 \, sin\, \theta \; = \beta^0 \frac{A_\beta}{A_{\sigma_e}}$~~

Starting from the radar brightness β^0 (conversion from σ_e^0 to β^0 can be applied as needed), the backscatter coefficients σ^0 and γ^0 are obtained for each image sample using the following equations:

$$\sigma^0 = \beta^0 \, cos\, \psi \; = \beta^0 \frac{A_\beta}{A_\sigma}$$

$$\gamma^0 = \beta^0 \frac{cos\;\, \psi}{cos\;\, \theta_i} = \beta^0 \frac{A_\beta}{A_\gamma}$$

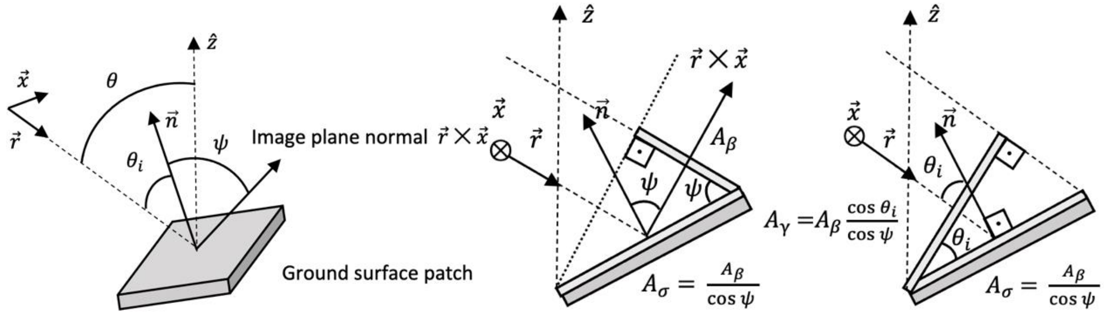
where θ_i is the local incidence angle defined as the angle between the unit vector normal to the local terrain slope $\hat{n}$ and the unit vector pointing from the target to the radar platform $\hat{r}$

$$\theta_i = acos\; (\hat{n} \cdot \; \hat{r})$$

and ψ is the projection angle, i.e., the angle between the normal to the image plane $\hat{g}$ (obtained as outer product between $\hat{r}$ and the platform velocity unit vector $\hat{v}$) and the normal to the local terrain slope $\hat{n}$

$$cos\, \psi \; = \hat{n} \cdot \; \hat{g} = \hat{n} \cdot (\hat{v} \; \times \; \hat{r})$$

Figure 6.4-2 shows the geometry of a ground surface patch projected into a SAR image describing the relationship between the incidence angle θ, the local-incidence angle θ_i, and the projection angle ψ and the reference areas A_β, A_σ, and A_γ.



(a) (b) (c)

Figure 6.4-2. Diagrams illustrating the SAR imaging geometry of a ground surface patch (a) in the 3-D space, and in simplified 2-D planes describing the relationship between the (b) image reference surface $\boldsymbol{A_\beta}$, and the ground surface $\boldsymbol{A_\sigma}$ and (c) the relationship between $\boldsymbol{A_\beta}$, $\boldsymbol{A_\sigma}$, and $\boldsymbol{A_\gamma}$. $\vec{\boldsymbol{x}}$ is the along-track vector, $\vec{\boldsymbol{r}}$ is the look vector, $\vec{\boldsymbol{n}}$ is the ground surface normal, θ is the ellipsoidal incidence angle, $\boldsymbol{\theta_i}$ is the local-incidence angle, and ψ is the projection angle. Credits: [Shiroma et al., 2022].

Amongst the different RTC methods, the approach proposed by D. Small [Small, 2011] and derived approaches [Frey et al. 2013, Simard et al., 2016, Shiroma et al., 2022] are usually preferred because it provides a solution to the lack of *bijective homomorphism* between ground and radar coordinates, i.e., ``many-to-one" and ``one-to-many" mapping between the two coordinate systems. This is achieved by first computing the reference surface, e.g. A_σ, and A_γ, over the geographical grid (geogrid) and further accumulating its area over radar coordinates. If multiple geogrid pixels (e.g., DEM pixels) fall into a single range-Doppler bin, all associated areas corresponding to the map pixels will be added to the radar sample correction factor.

More specifically, each geogrid gamma-naught area element is obtained by measuring the ground surface area A_σ directly from reference DEM and scaling to gamma-naught area A_γ by projecting the facet onto the cross-section plane using the local-incidence angle. The resulting gamma-naught area is projected and accumulated over the radar grid. Afterwards, the sum is divided by the beta-naught area A_β, and the resulting ratio, referred to as the *area normalization factor* (ANF), can then be used to scale the radar brightness β^0 to the backscatter coefficient γ^0.

[Small, 2011, Shiroma et al., 2022].

The projection of the gamma-naught area from the geogrid onto the radar grid proposed by [Small, 2011] uses a bilinear distribution. This approach often requires high DEM upsampling to avoid aliasing effects and unwanted artifacts, especially in areas with steep topography. Unfortunately, the DEM upsampling usually comes with a polynomial cost (linear in each direction), significantly increasing the processing time.

A more efficient approach uses the slant-range projection with the area projection algorithm [Shiroma et al., 2022]. This projection is able to adapt to different postings between the geogrid and the radar grid, filling all spaces covered by the projected area elements without leaving gaps, reducing or eliminating the need for DEM upsampling.

6.5 The area projection algorithm

The projection between radar range-Doppler and geographic coordinates, an operation shared by the radiometric terrain correction (RTC) and geocoding steps, is traditionally performed by interpolation, which assumes the representation of data as point elements. However, radar samples often contain information gathered not only from a point but from the integrated area or volume delimited by the resolution cell associated with the samples. The area projection algorithm consists of handling pieces of data as area elements, rather than point elements [Shiroma et al., 2022]. Each area element in one coordinate system is associated with a number of area elements in the other coordinate system. The values associated with the area elements are averaged and distributed from the source to the target grid. The number of averaged samples varies with the topography and data acquisition geometry. When applied to geocoding, the method represents an adaptive multi-looking that differs from the constant-window multi-looking that is traditionally applied to SAR data to reduce speckle. Analogously, the slant-range projection of geocoded data is improved by projecting geographic grid pixels onto the radar grid according to their corresponding location without leaving gaps. This approach is used to significantly reduce the computation time of previously published RTC algorithms [Shiroma et al., 2022].

6.5.1 Description of the area projection algorithm

The first step of the area projection algorithm consists of dividing the geographical grid, or geogrid, into area elements (AEs) to be mapped onto the range-Doppler domain. For geocoding, the area elements are defined as the geographical grid pixels (i.e., rectangles). For RTC, the area elements are defined as triangles (see Section 6.5.5).

After defining the area elements, the algorithm starts by visiting each vertex of the area element *x* perimeter in the clockwise order and locating the position of the vertex in the radar grid through an *inverse mapping algorithm* (described in Section 3.4.2).

If the AE has more than three vertices, the projected polygon is tested for *self-intersection*. If the polygon is self-intersecting (e.g., antiparallelogram), the corresponding AE is subdivided into smaller sub-AEs (two triangles) and the process is repeated for each sub-AE [Shiroma et al., 2022].

The projected polygon is then rasterized onto the range-Doppler grid through a polygon *rasterization algorithm* (described in Section 6.5.2). The rasterized polygon will contain, for each *r*-th radar-grid pixel, weights $w_{r,x}$ whose absolute values range from $|w_{r,x}| = 0$ if it is fully outside the projected AE and $|w_{r,x}| = 1$ if the pixel is fully contained, and intermediate absolute values between 0 and 1 calculated based on the normalized area intersection between the radar-grid pixel and the projected AE (output of the rasterization algorithm).

6.5.2 Rasterization algorithm

The suggested implementation of a rasterization algorithm is based on the integration of edges of a closed polygon. The polygon edges are represented as linear functions that can be integrated by definite integrals where the lower and upper limits are delimited by the polygon vertices. The algorithm is illustrated in Figure 6.5-1.

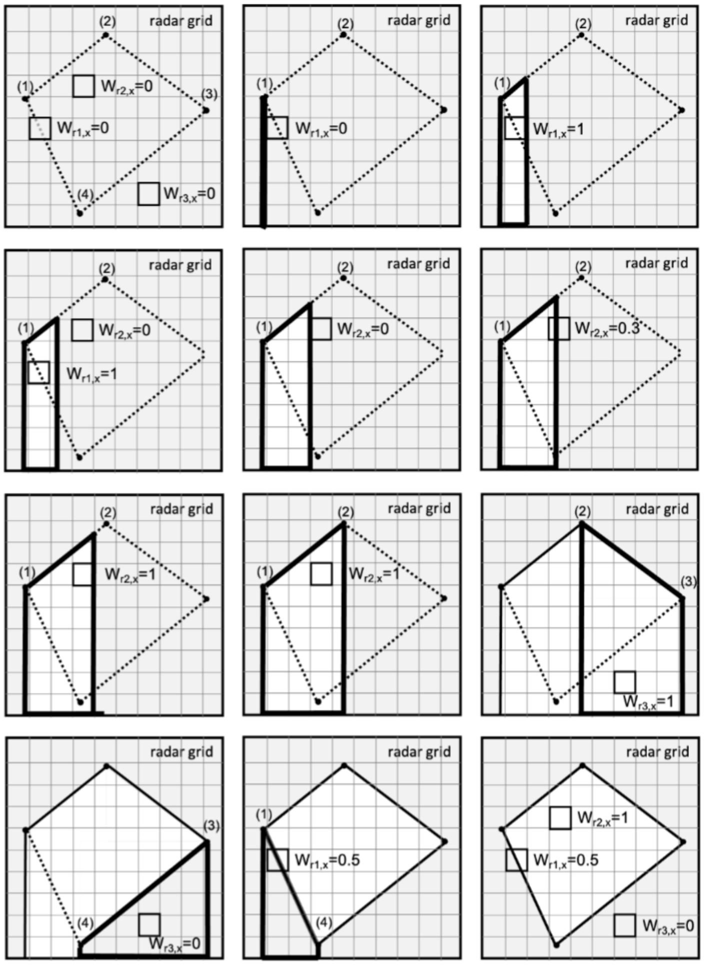


Figure 6.5-1. Suggested implementation of a rasterization algorithm based on the integration of a closed polygon. Each edge of the projected polygon is visited in the clockwise order. To each pixel r, intersected by or below the edge, a value representing the fraction of intersection, varying from 0 to 1, is added if the integration occurs from left to right, or subtracted otherwise. In the figure, this process is shown step-by-step between the edges associated with the vertices (1) and (2). The same process is repeated for the following edges. The integration stops when all edges are visited. Since the edges are visited in the clockwise order, the integrated value will be nonnegative and the resulting weight wr,x, associated with the area element x and radar-grid element r, will also be nonnegative ranging from 0 to 1.

The suggested rasterization algorithm starts by successively traversing each edge of the area element in a predefined order. If the polygon is traversed in the clockwise direction, all weights and integrated values will be nonnegative. On the other hand, if the polygon is traversed in the anti-clockwise direction, all rasterized values and integrated area will be nonpositive. Therefore, the suggested rasterization algorithm also allows for testing the visiting order of the polygon edges.

To each pixel *r*, intersected by or below the edge, a value representing the fraction of intersection, varying from 0 to 1, is added if the integration occurs from left to right or subtracted otherwise. The integration stops when all edges are visited.

Notice that one loop sequence iterates over the pixels intersected by the edge. For each intersection, another loop visits the pixels below the intersected pixel, adding the fraction of intersection as illustrated in Figure 6.5-1. Notice also that the same pixel may be visited multiple times while integrating a single edge or different edges. For instance, in Figure 6.5-1, the point $w_{r2,x}$, initially set to 0, is updated to 0.3, and successively to 1 during the integration of the first edge. The point $w_{r3,x}$ is set to 1 during the integration of the second edge and reset to 0 during the integration of the third edge.

6.5.3 Geocoding with the area projection algorithm

The association of radar grid pixels to the geogrid area elements and the corresponding weight map is used to project data from geo-coordinates to radar-grid coordinates and vice-versa.

If the projection occurs from the radar grid to the geo-coordinates, the geocoded value g_x is obtained from the weighted average of the set R of radar grid pixels r with value A_r that intersect completely or partially the area element x,

$$g_x = \frac{\sum_{r=1}^{R} \quad |w_{r,x}| A_r}{\sum_{r=1}^{R} \quad |w_{r,x}|}$$

The denominator of the equation above represents the number of looks n_x intersected by the area element x:

$$n_x = \sum_{r=1}^{R} \quad |w_{r,x}|$$

Note that n_x is not necessarily integer and varies with the shape of the AE x projected over the range-Doppler domain. Null radar samples or undefined elements should have their associated weights set to zero $w_{r,x} = 0$.

Notice that the weighted averaging performed by the proposed geocoding is equivalent to an adaptive multilooking operation that accounts for the topography and acquisition geometry.

The L2 geocoded polarimetric covariance (GCOV) matrix $[G_x]$ can be generated by applying the same adaptive multi-looking operation over the cross-product $c_{i,j}$ of the scattering vector k (Section 6.5) according to:

$$[G_x] = \frac{\sum_{r=1}^{R} \quad |w_{r,x}| \, c_{i,j}}{\sum_{r=1}^{R} \quad |w_{r,x}|}$$

where i and j are the line/column indices of the covariance matrix elements. When $i = j$, then C_x is real-valued and represents the averaged radar backscatter of the polarimetric channel associated with the i,j-element of the covariance matrix.

6.5.4 Slant-range projection with the area projection algorithm

The area projection algorithm also allows for the projection in the inverse direction, i.e., from the data G_x, in geo-coordinates, to the data A_r, in radar-grid coordinates. This operation, traditionally called *slant-range projection*, *inverse geocoding*, or *back geocoding*, is commonly employed for coregistering SAR images for interferometric purposes. The projected radar-grid value A_r is obtained by accumulating the projected values of G_x, associated with radar grid pixel r, according to:

$$A_r \leftarrow A_r + \frac{|w_{r,x}| G_x}{\sum_{r'=1}^{R} \quad |w_{r',x}|}$$

For radiometric terrain correction, we accumulate the gamma-naught area [Small, 2001], so no further normalization is required.

6.5.5 Radiometric terrain correction with the area projection algorithm

Since the slant-range projection with the AP algorithm is able to adapt to different postings between the geogrid and the radar grid, filling all spaces covered by the projected AEs without leaving gaps, it can be used to replace the slant-range projection performed by the bilinear distribution (RTC-BI). This new approach reduces or eliminates the need for DEM upsampling, which represents an important improvement of the RTC-AP algorithm over the traditional method, especially when the radar grid posting is much finer than the DEM posting (see Fig. 6.5-2).

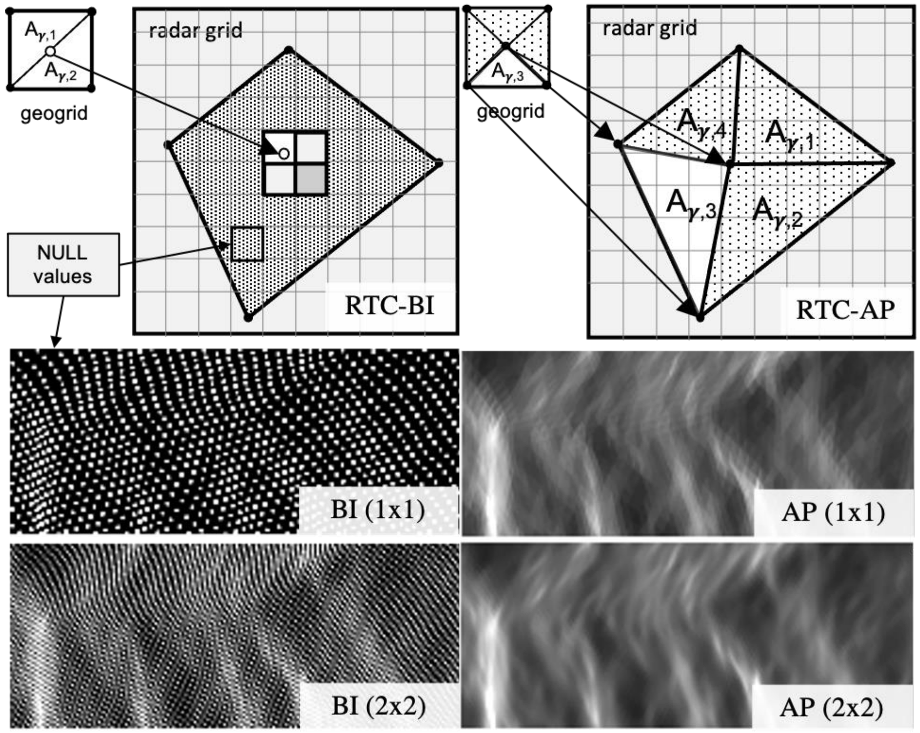


Figure 6.5-2. Diagrams illustrating the RTC with the (Top Left) bilinear distribution (RTC-BI) and (Top Right) AP (RTC-AP) algorithm, followed by images of the RTC ANF from NISAR-simulated single-look imagery (20-MHz mode) generated with the (Bottom Left) RTC-BI and (Bottom Right) RTC-AP algorithm. The level of DEM upsampling along easting and northing directions is indicated inside the parenthesis. The RTC-BI algorithm may leave gaps or suffer from aliasing effects, requiring higher DEM upsampling compared to the RTC-AP algorithm.

The algorithm to perform the RTC with the area projection algorithm starts by defining the geographical grid by means of the DEM geographical grid. Each element of the geogrid is then subdivided into four facets using the vertices and the center of the grid element (as shown in Fig. 6.5-2). Each of the four facets are visited, repeating the following steps:

1.1.Compute the cosine of the local-incidence angle cos θ_i. If it is negative, the facet does not face the sensor and the execution skips to the next facet.

1.2.Measure the facet ground surface area A_σ^F , which can be obtained from the DEM using the cross-product area:

$$A_\sigma^F = \frac{\| \overrightarrow{e_1} \times \overrightarrow{e_2} \|}{2}$$

where $\overrightarrow{e_1}$ and $\overrightarrow{e_2}$ are two of the three edges of the facet (triangle).

Another option is to use the Heron's formula to calculate the area of a triangle [Small, 2011]:

$$h = \frac{1}{2} P_a + P_b + P_c$$

$$A_\sigma^F = \sqrt{h\,(h - P_a)(h - P_b)(h - P_c)}$$

where h is the facet semi-perimeter and $P_a, P_b, and\ P_c$ are the lengths of the edges, measured from the distance (in the three-dimensional space) between the facet vertices. The heights of the vertices are obtained from the reference DEM.

1.3.Scale the facet ground surface area A_σ^F to the facet gamma-naught area A_γ^F by projecting the facet area onto the cross-section plane using the local-incidence angle:

$$A_\gamma^F = A_\sigma^F \, cos\ \theta_i$$

1.4.Run the rasterization algorithm (described in Section 6.5.2) to determine the association weights $w_{r,x}$.

1.5.Project and accumulate the gamma-naught area A_γ^F over the radar grid using the association weights $w_{r,x}$:

$$A_\gamma \leftarrow A_\gamma + \frac{|w_{r,x}| \,.\, A_\gamma^F}{\sum_{r'=1}^{R} \quad |w_{r',x}|}$$

After the steps above are completed, for each radar-grid element, divide the accumulated gamma-naught area A_γ by the beta-naught area A_β (i.e., the radar grid pixel area) and the resulting ratio, referred to as the RTC area normalization factor (ANF), can then be used to scale the radar brightness β^0 to the backscatter coefficient A_γ.

The RTC ANF does not change with polarization as it is derived from radar imaging geometry. It is therefore applied to all elements of the covariance matrix (and scattering vector) according to:

$$[C_x^\gamma] = \frac{A_\beta}{A_\gamma} \, [C_x \,]$$

As discussed in Sec. 6.5.3, null radar samples or undefined elements should have their associated weights set to zero $w_{r,x} = 0$. Radar samples with extremely low RTC ANF values (shadow areas) are expected to contain measurements with a very low signal-to-noise (SNR) ratio. These measurements are amplified by the low RTC ANF values. Therefore, radar samples with RTC ANF values below a threshold value should be masked out by setting its corresponding weight to zero, i.e., $w_{r,x} = 0$.

The geocoding equation updated with radiometrically terrain-corrected GCOV matrix $\left[G_x^\gamma\right]$ terms becomes

$$\left[G_x^\gamma\right] = \frac{\sum_{r=1}^{R} \quad |w_{r,x}|\, c_{i,j}^\gamma}{\sum_{r=1}^{R} \quad |w_{r,x}|}$$

6.6 Geocoding

Two approaches are available for geocoding polarimetric products. The first approach employs an interpolation algorithm that is used to resample the radar samples at the center of the output grid in map coordinates. The available algorithms are sinc, bilinear, bicubic, and biquintic interpolation. The generation of L2 GSLC products employs, by default, geocoding with the sinc interpolation. The process is described in Section 4.3. Secondary layers of L2 GCOV products, such as *noise-equivalent-sigma-zero (NESZ)* or *antenna pattern* are also geocoded from L1 RSLC metadata. These layers are geocoded using an interpolation algorithm.

The second approach uses the area projection algorithm, detailed in Section 6.5.3. The geocoding with the area projection algorithm can be performed directly from full-resolution SAR data providing an adaptive multi-looking that adjusts to the terrain and acquisition geometry. By default, GCOV products are generated using the adaptive multi-looking without the constant-window multi-looking step.

The number of looks n_x (Section 6.5.3) used in the geocoding with adaptive multi-looking varies with the radar geometry and the scene topography. A map with the number of looks used in the process is included as an auxiliary layer of the L2 GCOV product. The number of looks map is computed during geocoding and it is saved over the same geographical grid of L2 GCOV imagery. The map of the *effective number of looks (ENF)* can be retrieved by dividing the number of looks N_x layer by the *oversampling factor*, which can be calculated from the L1 RSLC or L2 GCOV products metadata. Another layer that is included as an auxiliary layer of the GCOV product is a geocoded RTC normalization layer to convert the backscatter coefficient from γ^0 to σ^0.

7 ACKNOWLEDGMENT

The research and development of the algorithms in this document was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration (80NM0018D0004).

7 REFERENCES

Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. 1993. Network Flows: Theory, Algorithms, and Applications. Prentice Hall.

Ainsworth, T. L, Ferro-Famil L., and Lee, J. S., "Orientation angle preserving a posteriori polarimetric SAR calibration," in IEEE Transactions on Geoscience and Remote Sensing, vol. 44, no. 4, pp. 994-1003, April 2006.

Bamler, R., N. Adam, G. W. Davidson, and D. Just. 1998. "Noise-Induced Slope Distortion in 2-d Phase Unwrapping by Linear Estimators with Application to SAR Interferometry." *IEEE Transactions on Geoscience and Remote Sensing* 36 (3): 913–21.

Bourbigot, M., H. Johnsen, and R. Piantanida, Sentinel-1 Product Specification, Ref S1-RS-MDA-52-7441, Issue/Rev 3/2, March 14, 2016.

Brancato, V., & Fattahi, H., "UAVSAR observations of InSAR polarimetric phase diversity: implications for NISAR ionospheric phase estimation", Earth and Space Science, vol. 8, no. 4, pp. e2020EA001445, March 2021.

Breit, H., Fritz, T., Balss, U., Lachaise, M., Niedermeier, A., and Vonavka, M., "TerraSAR-X SAR Processing and Products," in IEEE Transactions on Geoscience and Remote Sensing, vol. 48, no. 2, pp. 727-740, Feb. 2010.

Cafforio, C., P. Guccione, and A. M. Guarnieri, Doppler Centroid Estimation for ScanSAR Data, IEEE Trans. Geoscience and Remote Sensing, 42(1), January 2004.

Chen, C. W., & Zebker, H. A. (2000). Network approaches to two-dimensional phase unwrapping: intractability and two new algorithms. *JOSA A*, *17*(3), 401-414.

Chen, C. W., & Zebker, H. A. (2001). Two-dimensional phase unwrapping with use of statistical models for cost functions in nonlinear optimization. *JOSA A*, *18*(2), 338-351

Chen, C. W., & Zebker, H. A. (2002). Phase Unwrapping for Large SAR Interferograms: Statistical Segmentation and Generalized Network Models. *IEEE Transactions on Geoscience and Remote Sensing* 40 (8): 1709–19.

Cloude S.R., Pottier E., "A Review of Target Decomposition Theorems in Radar Polarimetry", *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 34 No. 2, pp 498-518, March 1996

Costantini, M. (1998). A novel phase unwrapping method based on network programming. *IEEE Transactions on Geoscience and Remote Sensing*, 36(3), 813-821.

Costantini, M., & Rosen, P. A. (1999). A generalized phase unwrapping approach for sparse data. In *Geoscience and Remote Sensing Symposium, 1999. IGARSS'99 Proceedings. IEEE 1999 International* (Vol. 1, pp. 267-269). IEEE

Dantzig, G. B. 1951. "Application of the Simplex Method to a Transportation Problem." *Activity Analysis and Production and Allocation*, 359–73.

Davidson, G. W., & Bamler, R. (1999). Multiresolution phase unwrapping for SAR interferometry. *IEEE transactions on geoscience and remote sensing*, *37*(1), 163-174.

De Zan, F., "Accuracy of incoherent speckle tracking for circular Gaussian signals", *IEEE Geoscience and Remote Sensing Letters*, vol. 11, issue 1, pp. 264-267, 2013.

Dutt, A. and Rokhlin, V., 1993. Fast Fourier transforms for nonequispaced data. *SIAM J. Sci. Computing*, *14*(6), pp.1368-1393.

Eineder, Michael. "Efficient simulation of SAR interferograms of large areas and of rugged terrain." *Geoscience and Remote Sensing*, IEEE Transactions on 41, no. 6 (2003): 1415-1427.

ESA (European Space Agency), ENVISAT ASAR Product Handbook, Issue 2.2, Feb. 27, 2007, https://earth.esa.int/pub/ESA_DOC/ENVISAT/ASAR/asar.ProductHandbook.2_2.pdf

Fattahi, H., Simons, M., and Agram, P., "*InSAR time-series estimation of ionospheric phase delay: an extension of the split range-spectrum technique*", IEEE Transactions on Geoscience and Remote Sensing, vol. 55, issue 10, pp. 5984-5996, 2017.

Frey, O., Magnard, C., Ruegg, M., and Meier, E., "Focusing of Airborne Synthetic Aperture Radar Data from Highly Nonlinear Flight Tracks," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 6, pp. 1844-1858, June 2009.

Ghaemi, H., Shaffer, S., and Hensley, S., "Onboard digital beamforming: Algorithm and results," in *IEEE Geoscience and Remote Sensing Symposium*, 2014, pp. 3838–3841.

Goldberg, Andrew V. 1997. "An Efficient Implementation of a Scaling Minimum-Cost Flow Algorithm." *Journal of Algorithms* 22 (1): 1–29.

Goldstein, R. M., & Werner, C. L. (1998). Radar interferogram filtering for geophysical applications. Geophysical Research Letters, 25(21), 4035-4038.

Goldstein, R. M., Zebker, H. A., & Werner, C. L. (1988). Satellite radar interferometry: Two-dimensional phase unwrapping. Radio science, 23(4), 713-720.

Gomba, G., Parizzi, A., De Zan, F., Eineder, M., & Bamler, R. (2016). Toward operational compensation of ionospheric effects in SAR interferograms: the split-spectrum method. *IEEE Transactions on Geoscience and Remote Sensing*, *54*(3), 1446-1461.

Hanssen, R., & Bamler, R. (1999). Evaluation of interpolation kernels for SAR interferometry. IEEE Transactions on Geoscience and Remote Sensing, 37(1), 318-321.

Hawkins, B., Ghaemi, H., Huang, B., & Veeramachaneni, C., "NISAR Antenna Pattern Calibration," JPL interoffice memorandum, 2022.

Hawkins, B., Processing NISAR data with non-uniform pulse spacing, JPL interoffice memorandum, 2019.

Hensley, Scott, Elaine Chapin, Adam Freedman, and Thierry Michel. "Improved processing of AIRSAR data based on the GeoSAR processor." In Airsar earth science and applications workshop, march. 2002.

"IEEE Standard for Floating-Point Arithmetic," in *IEEE Std 754-2008*, pp.1-70, Aug. 2008.

Hensley, Scott. "SCH coordinates and various transformations." *JPL Interoffice memorandum*. 2000.

Jolivet, R., Grandin, R., Lasserre, C., Doin, M. P., & Peltzer, G. (2011). Systematic InSAR tropospheric phase delay corrections from global meteorological reanalysis data. *Geophysical Research Letters*, *38*(17).

Joughin, I., "Ice-sheet velocity mapping: a combined interferometric and speckle-tracking approach", *Annals of Glaciology*, vol. 34, 2002.

Kidner, David, Mark Dorey, and Derek Smith. "What's the point? Interpolation and extrapolation with a regular grid DEM." In *Proc. of GeoComputation*, vol. 99, pp. 25-28. 1999.

Klein, L. and A. Taaheri, HDF-EOS5 Data Model, File Format and Library, ESDS-RFC-008v1.1, May 2016, https://cdn.earthdata.nasa.gov/conduit/upload/4880/ESDS-RFC-008-v1.1.pdf

Knab, J. "Interpolation of band-limited functions using the approximate prolate series (Corresp.)," in IEEE Transactions on Information Theory, vol. 25, no. 6, pp. 717-720, November 1979.

Knab, J. "The sampling window (Corresp.)," in IEEE Transactions on Information Theory, vol. 29, no. 1, pp. 157-159, January 1983.

Krieger G., Papathanassiou K.P., Cloude S.R., "Spaceborne Polarimetric SAR Interferometry: Performance Analysis and Mission Concepts", *EURASIP Journal of Applied Signal Processing*, Vol. 20, pp 3272-3292, 2005

Kropatsch, Walter G., and Dieter Strobl. "The generation of SAR layover and shadow maps from digital elevation models." *Geoscience and Remote Sensing, IEEE Transactions on* 28, no. 1 (1990): 98-107.

Kwok, Ronald, and William TK Johnson. "Block adaptive quantization of Magellan SAR data." *IEEE Transactions on Geoscience and Remote Sensing*, 27, no. 4 (1989): 375-383.

Kunis, S., and Potts, D. "Time and memory requirements of the Nonequispaced FFT." *Sampling Theory in Signal and Image Processing*, vol. 7. 2008.

Lee, Jong-Sen, Hoppel, Karl W., Mango, Stephen A., and Miller, Allen R. "Intensity and Phase Statistics of Multilook Polarimetric and Interferometric SAR Imagery," *IEEE Trans. Geoscience and Remote Sensing*, vol. 32, no. 5, 1994.

Lee J.S., Schuler D.L., Ainsworth T.L.,"Polarimetric SAR Data Compensation for Terrain Azimuth Slope Variation", *IEEE Trans. Geoscience and Remote Sensing*, Vol 38/5, pp 2153-2163, September 2000.

Liang, C., Agram, P., Simons, M., Fielding, E. J., "Ionospheric correction of InSAR time series analysis of C-band Sentinel-1 TOPS data", *IEEE Trans. Geoscience and Remote Sensing*, Vol. 57/9, pp. 6755-6773, September 2019.

Madsen, S. N., "Estimating The Doppler Centroid of SAR Data", *IEEE Transaction On Aerospace and Elect Sys*, March 1989.

Madsen, S. N., and Rosen, P. A. Proceedings of the Interferometric SAR Technology and Applications Symposium, Ft. Belvoir, VA, 159-178, 1993.

Michel, R., Avouac, J. P., and Taboury, J., "Measuring ground displacement from SAR amplitude images: Application to the Landers earthquake", *Geophysical Research Letters*, Vol. 26, 875-978, 1999.

Migliaccio, M., Nunziata, F., Bruno, F., and Casu, F., "Knab Sampling Window for InSAR Data Interpolation." *IEEE Geoscience and Remote Sensing Letters*, Vol. 4, No. 3, pp. 397-400, July 2007.

Mouginot, J., Scheuchl, B., and Rignot, E., "Mapping of ice motion in Antarctica using Synthetic Aperture Radar data", *Remote Sensing*, vol. 4, issue 9, pp. 2753-2767, 2012.

Nitti, Davide Oscar, Ramon F. Hanssen, Alberto Refice, Fabio Bovenga, and Raffaele Nutricato. "Impact of DEM-assisted coregistration on high-resolution SAR interferometry." *Geoscience and Remote Sensing, IEEE Transactions on*49, no. 3 (2011): 1127-1143.

O. Frey, M. Santoro, C. L. Werner and U. Wegmuller, "DEM-Based SAR Pixel-Area Estimation for Enhanced Geocoding Refinement and Radiometric Normalization," *IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 1, pp. 48-52, Jan. 2013, doi: 10.1109/LGRS.2012.2192093.

Piantanida, R., Sentinel-1 Level 1 Detailed Algorithm Definition, DI-MPC-IPFDPM, Ref MPC-0307, Issue/Ref 2/0, Feb. 29, 2016, https://sentinel.esa.int/documents/247904/1877131/Sentinel-1-Level-1-Detailed-Algorithm-Definition

Press, W.H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P., *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge University Press, 2007.

Rignot, E., Mouginot, J., and Scheuchl, B., “Antarctic grounding line mapping from differential radar interferometry”, *Geophysical Research Letters*, vol. 38, issue 10, 2011.

Rodriguez, E., and J. M. Martin. 1992. “Theory and Design of Interferometric Synthetic Aperture Radars.” *IEE Proceedings F Radar and Signal Processing* 139 (2, 2): 147.

Rosen, P., SRTM Critical Design Review Algorithm Development and Verification, December 2, 1997.

Rosen, P., Scott Hensley, Franz Meyer, and Tom Ainsworth. "Further developments in ionospheric mitigation of repeat-pass InSAR data." *Proceedings of IGARSS 2010* (2010).

Sansosti, E., Berardino, P., Manunta, M., Serafino, F., & Fornaro, G. (2006). Geometrical SAR image registration. *IEEE Transactions on Geoscience and Remote Sensing*, *44*(10), 2861.

Schubert, A., Small, D., Miranda, N., Geudtner, D., Meier, E., “Sentinel-1A product geolocation accuracy: Commissioning phase results.” *Remote Sensing*, vol. 7, pp. 9431-9449, 2015.

Shafer, S., Hensley, S., and Frankot, R. T., “An automatic matching algorithm for Magellan stereo data”, *Geoscience and Remote Sensing Symposyum (IGARSS), 1994 IEEE International*.

Shiroma, G. H. X., Lavalle, M., and Buckley, S., "An Area-Based Projection Algorithm for SAR Radiometric Terrain Correction and Geocoding," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1-23, 2022, Art no. 5222723, doi: 10.1109/TGRS.2022.3147472.

Simard, M., Riel, B. V., Denbina, M., Hensley, S., "Radiometric Correction of Airborne Radar Images Over Forested Terrain With Topography," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 8, pp. 4488-4500, Aug. 2016, doi: 10.1109/TGRS.2016.2543142.

Simons, M., and Rosen, P. A., “Interferometric synthetic aperture radar geodesy”, 2007.

Small, D. “Flattening gamma: Radiometric terrain correction for SAR imagery,” *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 8, pp. 3081–3093, Aug. 2011.

Small, D., Schubert A., Guide to ASAR geocoding, Issue 1.0, 19.03.2008.

Ulaby F.T., Moore R.K., Fung A.K., “Microwave Remote Sensing: Active and Passive Vol III: From Theory to Applications”, Artech House, 1986

Ulander, L. M. H. “Radiometric slope correction of synthetic-aperture radar images,” *IEEE Trans. Geosci.* Remote Sens., vol. 34, no. 5, pp. 1115–1122, Sep. 1996.

van Zyl J. J., "Calibration of Polarimetric Radar Images Using Only Image parameters and Trihedral Corner Reflectors", IEEE Transactions on Geoscience and Remote Sensing, Vol. GE-28, pp 337-348, 1990.

Villano, M., Krieger, G. and A. Moreira, "Staggered SAR: High-Resolution Wide-Swath Imaging by Continuous PRI Variation," IEEE Transactions on Geoscience and Remote Sensing, vol. 52, no. 7, pp. 4462-4479, 1 July 2014.

West, R., “Soil Moisture Active and Passive Mission L1B_S0, L1C_S0 Algorithm Theoretical Basis Document, 2014, http://smap.jpl.nasa.gov/system/internal_resources/details/original/280_L1C_S0_RevA_web.pdf

Wong, Frank H., Ngee Leng Tan, and Tat Soon Yeo. "Effective velocity estimation for spaceborne SAR." In Geoscience *and Remote Sensing Symposium, 2000. Proceedings. IGARSS 2000. IEEE 2000 International*, vol. 1, pp. 90-92. IEEE, 2000.

Wright P.A., Quegan S., Wheadon N. S., David Hall C., “Faraday Rotation Effects on L-Band Spaceborne SAR Data”, *IEEE Trans. Geoscience and Remote Sensing,* Vol. GRS-41, No. 12, pp 2735- 2744, Dec. 2003

Yague-Martinez, N., Eineder, M., Brcic, R., Breit, H., & Fritz, T. (2010, June). TanDEM-X mission: SAR image coregistration aspects. In *Synthetic Aperture Radar (EUSAR), 2010 8th European Conference on* (pp. 1-4). VDE.

Younis, M., Rommel, T., Bordoni, F., Krieger G., and Moreira, A., "On the Pulse Extension Loss in Digital Beamforming SAR," in *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 7, pp. 1436-1440, July 2015.

Zebker, H. A., & Chen, K. (2005). Accurate estimation of correlation in InSAR observations. *IEEE Geoscience and Remote Sensing Letters*, *2*(2), 124-127.

Zhu, L., Zhong, L. Simons, M. (2022). PyCuAmpcor: Amplitude cross-correlation with GPU. 10.5281/zenodo.6614209

8 Acronyms

ADT	Algorithm Development Team
ANF	Area Normalization Factor
CDR	Critical Design Review
DEM	Digital Elevation Model
ECMWF	European Center for Medium-Range Weather Forecasts
FR	Faraday Rotation
GOFF	Geocoded Pixel Offsets Product
GUNW	Geocoded Unwrapped Interferogram
ICU	Integrated Correlation and Unwrapping
InSAR	Interferometric Synthetic Aperture Radar
LUT	Look Up Table
MAP	Maximum *a Posteriori*
MCF	Minimum Cost Flow
MST	Minimum Spanning Tree
ML	Maximum Likelihood
PDF	Probability Density Function
PDR	Preliminary Design Review
PG	Power Gain
POC	Polarimetric Orientation Correction
RIFG	Range-Doppler Wrapped Interferogram
ROFF	Range-Doppler Pixel Offsets product

RSLC	Range-Doppler Single Look Complex
RUNW	Range-Doppler Unwrapped Interferogram
RTC	Radiometric Terrain Correction
SAR	Synthetic Aperture Radar
SDS	Science Data System
SDT	Science Definition Team
SLC	Single Look Complex
SNAPHU	Statistical-cost, Network-flow Algorithm for Phase Unwrapping
SNR	Signal-to-noise ratio
SRTM	Shuttle Radar Topography Mission
ST	Science Team
TCF	Terrain Correction Factor
TEC	Total Electron Content
UTM	Universal Transverse Mercator