ISSN 2029-6991 / eISSN 2029-7009

VILNIUS TECH
Vilnius Gediminas Technical University

GEODESY and CARTOGRAPHY

2025
Volume 51
Issue 2
Pages 88–99
https://doi.org/10.3846/gac.2025.21995

UDC 528.8.04

PHASE: A MATLAB-BASED SOFTWARE FOR THE DINSAR PS PROCESSING

Roberto MONTI ✉, Lorenzo ROSSI

Department of Civil and Environmental Engineering (DICA), Politecnico di Milano, Milan, Italy

Article History:
- received 08 August 2024
- accepted 28 May 2025

Abstract. The availability of free Synthetic Aperture Radar (SAR) images, like the ones delivered by the ESA Copernicus Sentinel-1 satellites, has led to the development of several processing tools, some of which are also free and open source. In this framework, when analyzing Sentinel data, the ESA SNAP software is usually required for data preprocessing, while, in the context of free and open source (FOS), Persistent Scatterer Interferometry (PSI) analysis can be performed by StaMPS (released by Stanford University). The workflow could be completed by the snap2stamps package, aiming at integrating the two main software packages. However, these tools are not designed to automate all the required steps to perform a complete PSI analysis. For this reason, the aim of this work is to develop PHASE (Persistent scatterer Highly Automated Suite for Environmental monitoring), a Matlab-based software suite that relies on already available FOS software, properly updated, enhanced and integrated, all accessible and customizable through a simple GUI. The focus is on minimizing the user interaction with the software, thus decreasing potential sources of error, while improving processing repeatability. The user will therefore primarily be responsible for configuring the processing parameters. Indeed, a streamlined procedure has been established, covering the entire process from downloading the SAR images to exporting the PS time series into a simple table format. In the paper, we present the developed software, highlighting its strengths compared to the status quo, while also providing a short example of successful application of the entire procedure.

Keywords: Persistent Scatterers, DInSAR, automatization, time series, displacement, Matlab, software, GUI.

✉Corresponding author. E-mail: ***roberto.monti@polimi.it***

1. Introduction

The Persistent Scatterer (PS) analysis based on Synthetic Aperture Radar (SAR) complex images represents a very popular and widely used tool for the identification, and subsequent study, of deformation time series associated with reflectors (scatterers) that materialize points characterized by stable reflective properties over time (Delgado Blasco et al., 2019). To perform this type of analysis several software packages exist, released both under a free and open source (FOS) or commercial license. Following the launch of the European Space Agency Copernicus Sentinel-1 mission (Fletcher, 2012) in 2014 SAR images also became available for free. Therefore, the interest in a completely free acquisition and processing procedure has largely increased for both research and corporate applications. As a consequence, two tools have emerged among others: the SeNtinel Application Platform (SNAP) software (European Space Agency [ESA], 2023) for the preprocessing of the SAR images, and the Stanford Method for Persistent Scatterers (StaMPS) software (Hooper et al., 2012) for the PS analysis. Subsequently, to seamlessly integrate these two software solutions, the snap2stamps package (Foumelis et al., 2018) has been developed. Leveraging the SNAP libraries, this package facilitates a cohesive and efficient workflow between SNAP and StaMPS. This combination has gained widespread adoption for a diverse array of studies centered on Persistent Scatterer analysis. Common applications span various fields, including seismology (Dalla Via et al., 2012; Massonnet et al., 1993, 1995; Peltzer & Rosen, 1995), volcanology (Antonielli et al., 2014; Massonnet & Sigmundsson, 2000), glaciology (Goldstein et al., 1993; Rignot et al., 1997), landslides (Carnec et al., 1996; Mateo-Garcia et al., 2021), ground subsidence and uplift (Amelung et al., 1999; Galloway et al., 1998), and many more.

SNAP, StaMPS and snap2stamps are well functioning, allowing the user to have detailed control of the whole process without the need to manually configure all the parameters at each iterative step. However, sometimes the multiple tools used in the process are not immediately integrated. Therefore, some manual interventions are required, e.g., to create the correct folder structure, launch each process, or manually perform some preliminary steps

New content has been added to this article's online version. Please see Addendum https://doi.org/10.3846/gac.2026.26826

before launching them. All these external steps can be critical, especially for people just approaching this activity field. In addition to that, the snap2stamps scripts have been developed for an older version of the SNAP software, leading the user to have to manually update each script after consulting the reference manual or forum. To overcome these issues and consequently improving the overall processing, PHASE (Persistent scatterer Highly Automated Suite for Environmental monitoring), a Matlab-based software suite, was developed in the framework of this work, with the aim of providing ready-to-use GUI applications to standardize and simplify the daily DInSAR PS processing. The proposed software leverages the already available snap2stamps package and both SNAP and StaMPS software, focusing on their enhancement, automation, and integration to provide the user with a simple and intuitive product able to streamline a complex procedure, such as the SAR images processing for PS analysis.

To improve clarity, enhance understanding, and engage the reader effectively, the upcoming section (Section 2) will provide essential theoretical insights for applying the Persistent Scatterer Interferometry (PSI) technique. Then, in Section 3, we will present the current state of the software involved, highlighting their functionalities and lacks. Section 4 will focus on the developed software suite PHASE, which aims to address and solve all the current limitations. In Section 5 we will conduct a comprehensive comparison between FOSS solutions and our custom software implementation, shedding light on their relative merits and limits. Finally, conclusions and future perspectives are drawn in the last section.

2. Persistent Scatterer Interferometry

Persistent Scatterer Interferometry is a remote sensing technique that can be used to quantify and monitor displacements of temporally coherent targets on the Earth's surface over time. They can be artificial, such as man-made structures (e.g., roads, buildings, dams) as well as natural, such as rock formations (e.g., for the monitoring of landslides). PSI belongs to the family of the Differential Interferometric Synthetic Aperture Radar (DInSAR) techniques (Crosetto et al., 2011), in which the different information contained in the phase of a coregistered stack of complex images is exploited. At least two images are required to perform an interferometric stack, but studies (e.g., Crosetto et al., 2015) recommend no less than twenty images for reliable and satisfactory results. The SAR images are usually acquired by repeated passes of satellites with the same sensor(s) over the same study area. Nevertheless, the combination of data coming from different satellites, and consequently different sensors, can also be performed, properly accounting for the correct integration methodology (Zhang et al., 2021).

Considering now, for the sake of simplicity, the case of two repeated acquisitions over the same target, the equations defining the DInSAR principle can be exploited. For the first image, acquired by a satellite at position S and time t, the measured phase contribution (φ), for a point P on the ground, can be written as:

$$\varphi_S = \varphi_{geom,S} + \varphi_{scatt,S} = \frac{4 \times \pi \times SP}{\lambda} + \varphi_{scatt,S}, \quad (1)$$

where SP is the distance from the sensor to the target, $\varphi_{scatt,S}$ is the contribution due to the interaction of the microwaves with the reflecting surfaces on the ground, λ is the radar wavelength, and 4π the factor associated to the two-ways nature of the emitted pulse. The second image is then acquired at time $t + \Delta t$, with P' representing the position of point P at this epoch. The following phase is observed:

$$\varphi_T = \varphi_{geom,T} + \varphi_{scatt,T} = \frac{4 \times \pi \times TP'}{\lambda} + \varphi_{scatt,T}. \quad (2)$$

In Equation (2) the satellite is at position T while the target is at position P', which could be different from the one of P. The differential interferometry exploits the difference of the two phases (i.e., the interferometric phase φ_{int}) (Equations (1) and (2)), resulting in the sum of a geometric and a scattering contribution, that is:

$$\varphi_{int} = \varphi_T - \varphi_S = \frac{TP' - SP}{\frac{\lambda}{4\pi}} + \varphi_{scatt,T} - \varphi_{scatt,S}. \quad (3)$$

If at time $t + \Delta t$, i.e., when the second image is acquired and the satellite was at position T, the target point moved from P to P' due to some external phenomenon (e.g. subsidence), it is still reasonable to expect that the scattering effects $\varphi_{scatt,S}$ and $\varphi_{scatt,T}$ are very similar when the PS exhibits good coherence. Therefore, their difference in Equation (2) can be neglected. As a consequence, Equation (3) can be rewritten as:

$$\varphi_{int} = \varphi_T - \varphi_S = \frac{TP - SP}{\frac{\lambda}{4\pi}} + \frac{TP' - TP}{\frac{\lambda}{4\pi}} = \varphi_{Topo} + \varphi_{Displ}. \quad (4)$$

In Equation (4), the first term can be referred to as the topographic phase (φ_{Topo}), including both the ellipsoidal and Digital Elevation Model (DEM) components, while the second one (φ_{Displ}) is the value of the phase describing the actual displacement along the Line-Of-Sight (LOS). So, if a proper ellipsoid and terrain models are available, the actual LOS displacement φ_{Displ} can be directly inferred from Equation (4). Obviously, the estimation of the topographic phase contribution is strongly dependent on the accuracy of the available data and model used for its description, leaving a residual topographic component (φ_{Topo_res}) in the data.

The equations describing the DInSAR observations (see (1) to (4)) up to now have been written in a simplified way, neglecting all the contributions that are not physically related to the position of the target on the ground. Therefore, to be more precise:

$$\varphi_{DInSAR} = \varphi_{int} - \varphi_{Topo} = \varphi_{Atm,T} - \varphi_{Atm,S} + \varphi_{Orb,T} - \varphi_{Orb,S} + \varphi_{Noise} + \varphi_{Amb}, \quad (5)$$

where the terms φ_{Atm} express the atmospheric phase component at the time of acquisition, φ_{Orb} the errors in the orbits' estimation, φ_{Noise} the noisy part plus all the unmodellable errors and φ_{Amb} the phase ambiguity (i.e., the integer number of cycles), which is a direct consequence of the wrapped nature of the interferometric phase.

The phase noise φ_{Noise} represents the instrumental noise. This quantity cannot be computed by numerical models, and, for this reason, every processing technique has the goal to reduce it as much as possible, recalling that the temporal and spatial decorrelations play a big role in its increase.

Over time, several algorithms and procedures have been developed with the aim of separating the φ_{Displ} component of the phase from the other components, thus estimating time series of deformation and velocities of targets showing good amplitude stability or coherence over time (Ferretti et al., 2000, 2001; Herrera et al., 2007; Van Leijen, 2014). Among them, the methodology proposed by Hooper et al., 2004 is of particular interest. Here, the PS selection is not performed by directly looking at the amplitude, e.g., choosing as candidate PS pixels those with the highest amplitude. Rather, the phase stability in time is considered, evaluating the ratio between the standard deviation and mean of the amplitude (D_A), and choosing as candidates PS pixels those with the lower values of D_A index. Therefore, this method becomes suitable also for low-amplitude natural targets. This work, and the related algorithm, led to the development of one of the most largely used free PSI software: StaMPS. On the other hand, also commercial software products have been developed, employing patented algorithms to perform the PS analysis. This is the case, for example of SARPROZ (https://www.sarproz.com), gamma-rs (https://www.gamma-rs.ch), and SARscape (https://www.sarmap.ch/index.php/software/sarscape/).

3. State of the art – PSI processing

In this section a summary of all the software and tools used during a standard – as proposed in literature – PSI processing with the StaMPS algorithm is presented, discussing the pros and cons of each step. It is worth noting that to apply the StaMPS algorithm effectively, preprocessing of SAR images is required, often accomplished within the SNAP environment, complemented by the snap2stamps package.

3.1. SNAP

The SeNtinel Application Platform is a toolbox developed and freely distributed by ESA with the aim of providing a powerful instrument capable of processing data coming from their Sentinel satellite constellation, as well as other SAR satellite platforms, like ENVISAT (Louet & Bruzzi, 1999), TerraSAR-X (Pitz & Miller, 2010) and Cosmo-SkyMed (Covello et al., 2010). In particular, the SAR segment of the Copernicus Sentinel program is composed by the Sentinel-1A and Sentinel-1B (which mission has been declared finished on August 3rd, 2022 (ESA, 2022)) twin satellites. The SAR complex images are freely distributed by ESA under the Copernicus program and can therefore be freely downloaded. For the PSI workflow, once downloaded, the images have to be imported in SNAP. It has the capability to autonomously recommend the master image for data processing, optimizing for minimal temporal and spatial baselines. However, by considering factors like weather and ground conditions in this selection process, the user could further enhance the choice. Then, the chosen master image must be manually processed to extract the area of interest, as well as the chosen polarization, and to apply the precise a-posteriori orbits. All the other images, denoted as slaves, can, instead, be processed using the snap2stamps scripts aimed at automating this repetitive procedure. Nevertheless, the impossibility to process the master image in an automated way, as for the slaves, represents a limitation in view of making the entire procedure easier, faster and completely automatic.

3.2. StaMPS

The Stanford Method for Persistent Scatterers is a Matlab-based software package that implements a DInSAR PS method developed to work even in terrains devoid of man-made structures and/or undergoing non-steady deformation. It is distributed under a FOS license, making it one of the most used software for this type of DInSAR analysis. The StaMPS algorithm uses the spatially correlated nature of the deformation rather than using a priori assumptions of the temporal nature of the deformation. The initial set of candidate PS pixels, those with lower D_A index, is further filtered, choosing only points with high coherence. The noise contribution is then estimated by subtracting from the interferometric phase all the spatially correlated components (e.g., the atmospheric contribution) and the spatially uncorrelated look angle (Narayan et al., 2018). After the iterative procedure to refine the PS selection, adjacent and noisy pixels are dropped and, finally, the unwrapping of the phase is performed. To estimate the phase-deformation time series all the phase contributions not depending on the displacement (see Equation (5)) need to be estimated and removed. They are the ones related to the atmospheric phase of the master and slaves, to the orbit error of the master and slaves and to the spatially correlated look angle. As last step, the phase is converted into metric displacement by applying the standard conversion factor ($\lambda/4\pi$). In addition to this complete and detailed procedure, this software also enables the possibility to use the external toolbox TRAIN (Toolbox for Reducing Atmospheric InSAR Noise). In particular, TRAIN has been developed and freely distributed to include current state of the art tropospheric correction methods into the default DInSAR processing chain (Bekaert et al., 2015a).

Notwithstanding the quality of this software and the

large number of configurable settings, some limitations are still present, especially in spite of making this processing faster. Probably, the biggest one is that, as it is distributed, the user has to manually set every variable for each execution of the code, which is quite time-consuming. Alongside this, there is no built-in method to automatically export the computed displacement time-series, which is not ideal as this is the final scope of the PS processing. Last but not least, the preparation of the data exported from snap2stamps has to always be done manually, a time demanding procedure considering the file paths that must be inserted, often leading to typing errors. So, these three main problems can limit the ability to obtain reliable results in a fast but correct way, which is always the requirement when performing a DInSAR processing, considering that it is a time-consuming procedure.

3.3. snap2stamps

The snap2stamps package consists of a series of python scripts (compatible with both Python 2 and 3 versions) and XML configuration files. The latter contain the parameters for the slave image stack pre-processing and are required to call the functions present in the SNAP software library, through the Python scripts. It was developed to avoid the need of user inputs during repetitive procedures for which only the file of the slave image is changing over the processing. As widely known, the snap2stamps procedure can be summarized by four steps: 1) images preparation, 2) images splitting and orbits correction, 3) images coregistration and interferogram formation, and finally 4) images export in a StaMPS compatible file format.

The snap2stamps toolbox was released in 2018 and only a single update was distributed over the years. In the meantime, instead, SNAP received several major updates, carrying also important modifications in the software libraries, thus arising several problems in the functionality of snap2stamps. In other words, the distributed package is no longer fully compatible with the SNAP software as it is distributed but requires modifications that have to be performed by the users according to the new releases of SNAP. In addition to that, other limitations are present. In particular, the lack of the possibility to include in the automatic process also the master image is quite constraining, especially when different choices would be compared. Moreover, the available customization options are limited, in contrast with the available settings present in SNAP. The most relevant missing ones are the choice of the polarization direction, the selection of the burst(s), the possibility to employ an external DEM and to save the terrain-corrected coherence band.

4. Proposed software: PHASE

As detailed explained in the previous Sections, the current PS procedure utilizing SNAP (through snap2stamps), and StaMPS faces significant limitations in computational time and configurability. To address these challenges and enhance the user experience, we have developed a Matlab-based software suite, with the goal to streamline the processing workflow and offer greater flexibility in setting parameters. Instead of replacing existing tools, our software builds upon the snap2stamps package and both SNAP and StaMPS, introducing full automation and enhancing their integration to provide a more intuitive and efficient solution for processing SAR images for PS analysis. We have minimized the required inputs and incorporated a user-friendly graphical interface for configuring settings. Furthermore, we have updated the original snap2stamps scripts to ensure compatibility with the latest SNAP release (v. 10.0.x). It is important to note that our software does not embed the original package of StaMPS that must be previously installed by the user, as detailed in the accompanying manual. We chose Matlab as the main programming language because it is already used by StaMPS. This decision aligns with our goal of creating a straightforward workflow without the added complexity of introducing another programming language. However, although the user will interact only with the PHASE GUI in Matlab, thus acting as a front-end, the software includes in his back end also the Python codes previously mentioned.

We named our software suite PHASE (Persistent scatterer Highly Automated Suite for Environmental monitoring) to encapsulate its key applications and advancements, ensuring clarity and simplicity. Although the acronym doesn't explicitly mention SAR, the term "PHASE" inherently evokes the essential observation used in SAR interferometry. The software package is organized into two distinct modules: one dedicated to preprocessing (utilizing SNAP), and the other focused on PSI analysis (leveraging StaMPS). The subsequent Sections will delve into the details of each module.

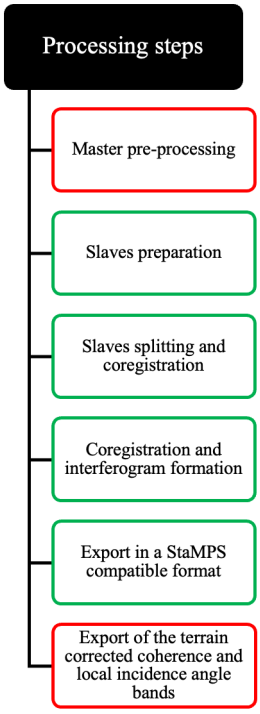
4.1. DInSAR PS pre-processing module

The first module is aimed at automating the whole pre-processing procedure by eliminating all the user-required inputs at different levels. Typically, the first step in SAR processing involves downloading the necessary images from a web portal. For the Sentinel-1 constellation, our software seamlessly integrates with the Python download script available on the Alaska Satellite Facility (ASF, https://asf.alaska.edu/) website, streamlining this process for users. Once the images are obtained, the provided tool takes over, orchestrating the entire workflow, from defining the master image to exporting data in a format compatible with StaMPS. This module of our software builds upon the snap2stamps framework, which partially automates the standard pre-processing steps required to generate interferometric products (co-registration, interferogram formation, and differential interferometry) while ensuring consistency with StaMPS input requirements. Given that snap2stamps is essential for structuring data in the correct format, we have

integrated an updated version of the original scripts, refining and expanding them to improve automation, capabilities, efficiency, and compatibility with the latest SNAP release. All configurable parameters (e.g., the date of the master, the path to the processing folder, the name of the project folder, the parameters for the snap2stamps scripts and so on) can be easily set directly in the proposed GUI of the developed application. Therefore, the given snap2stamps scripts represent an updated version of the original ones developed by the authors as they are both revised and newly created. The modifications were necessary for the previously reported reasons and include several improvements, such as:

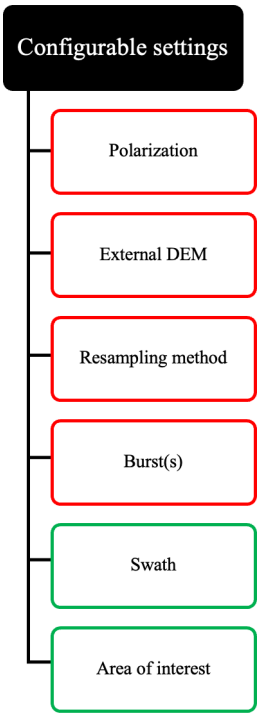
- Introduction of the choice of the type of polarization to be considered;
- Introduction of the choice of the initial burst to be used;
- Introduction of the choice of the final burst to be used;
- Introduction of the choice of using an external DEM;
- Introduction of the choice of the DEM resampling method;
- Adaptation of some variable names to match the updated version of SNAP;
- Extension of the workflow to also accommodate COSMO-SkyMed constellation.

Figures 1 and 2 show the already available processing steps and global configurable settings in the original snap2stamps codes (in green), as well as the additions in PHASE (in red).



Note: In green the steps already available in snap2stamps, while in red the proposed integration in PHASE.

Figure 1. Processing steps



Note: In green the settings already configurable in snap2stamps, while in red the proposed integration in PHASE.

Figure 2. Configurable settings of the entire processing

The implementation of these features occurs on the back end of the PHASE suite, which, for this first module, comprises the revised snap2stamps Python scripts and XML configuration files. The XML files interrogate the SNAP libraries, while the Python scripts compile them with the given inputs and iteratively repeat the procedure. Notably, the scripts that perform master splitting and orbit correction are among the most significant. This is of particular relevance because, as it stands, these two steps had to be manually performed in SNAP, slowing down the whole process and introducing a possible source of errors. Two other new scripts have been added to the collection: one for the estimation of the local incident angle and one for the estimation of the pixels coherence. Also, the code automatically creates the correct folder/subfolder structure for the next steps with the correct file names. Then, the standard snap2stamps procedure is applied to the slave images through its well-known four-step process: preparation, splitting and orbit correction, coregistration and interferogram computation, export for StaMPS. Finally, images of the terrain corrected coherence estimation (COH), local incident angle (LIA) are saved for each slave date, together with the average intensity (AI) of the scene. Additionally, two XML configuration files are proposed for the coregistration and interferogram computation step, in view of the different possible scenarios. One refers to the situation in which multiple bursts of the same swath are considered, while the other one is suitable for the selection of a smaller area of interest (AOI) – belonging to a single burst - inside the image. The only difference among these two cases is just the presence of the Enhance Spectral

Diversity (ESD) operator in the multi-swath procedure. As last step, the folder for the PS analysis is created and the required files for the next step are moved inside it. There is also the possibility to automatically remove all the downloaded slave files to save disk space since long time series can occupy significant storage.

Summarizing, all the steps performed by the developed software listed below.

1. Configuration of the input parameters through a GUI;
2. Download of all the SAR images required for the project;
3. Master processing:
   a. Preparation;
   b. Splitting and orbit correction;
   c. Creation of the correct folder structure and file names.
4. Slaves processing:
   a. Preparation;
   b. Splitting and orbit correction;
   c. Coregistration and interferogram formation;
   d. StaMPS export;
   e. Terrain corrected coherence, LIA and average intensity images export.
5. StaMPS processing folder preparation (with the required files) and optional removal of the slave files.

The significant innovation lies in the fact that users are no longer required to modify any part of the code or library files, greatly simplifying the entire procedure. Through the graphical user interface (Figure 3), users make selections that generate an input file consolidating all necessary variables. This file stores all user-dependent parameters in a single location, eliminating the need for scattered modifications. Additionally, configurable options include flags enabling users to easily determine whether certain steps are executed or skipped. Furthermore, the software compatibility with Windows, macOS and Linux environments offers users the freedom to operate within the preferred native system, maximizing resource utilization. This flexibility underscores the efficiency gains achieved and the impact of operating system choice on processing speed. Optimal workflow performance is thus realized by executing the code in the native environment, emphasizing the importance of platform selection.

Finally, we have extended the processing workflow to encompass the inclusion of Cosmo-SkyMed data in addition to Sentinel-1. This enhancement is made possible by developing a new set of back end tools (both XML configuration files and Python scripts) to meet the unique requirements of this satellite platform, following the same structure as those designed for Sentinel-1 image processing. Users can seamlessly choose between the two satellites via a simple toggle switch, enhancing flexibility and adaptability.

4.2. StaMPS processing module

The second module is designed to streamline StaMPS data preparation and processing by minimizing the manual inputs required from the user. Like the first module, users need to only set variables defining the project folder path and the parameters for StaMPS processing. However, it is crucial to note that while PHASE automates much of the workflow, users are strongly encouraged to inspect the outputs of each step to identify potential computational errors, which may stem from data quality issues or limitations within the source code. For troubleshooting and further guidance, users can refer to the STEP Forum (ESA, 2018b), where community members actively discuss and resolve various issues. At the core of this module is the PSI analysis, which is performed using StaMPS. Rather than modifying or replacing StaMPS, PHASE enhances its usability by providing a user-friendly GUI that consolidates all configurable parameters, including recommended values for environmental monitoring. This interface simplifies the setup process while preserving the flexibility of StaMPS for advanced users. For a detailed description of these parameters, users can refer to this resource (ESA, 2018a). Additionally, our software has been designed to

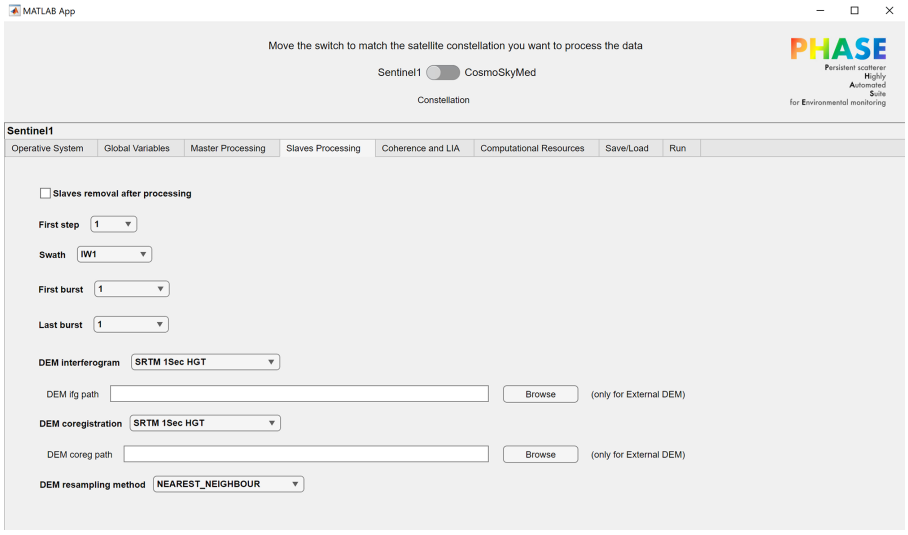


Figure 3. Example tab of the PHASE_Preprocessing Matlab application

Table 1. Example of data export from the PHASE_StaMPS module

Point	Longitude, degrees	Latitude, degrees	Mean LOS velocity, mm/yr	Displacements at days, mm			
				0	6	12	...
1	16.54539	39.24308	0.17843	–2.16447	–2.15390	1.34284	
2	16.54545	39.24309	0.31147	–2.41281	–2.98363	–0.48358	
...							

Note: The first column shows the point ID, the second and the third longitude and latitude, the fourth the mean LOS velocity. All the other columns refer to LOS the displacement at the incremental day (shown in the header) from the first epoch.

accommodate the utilization of the TRAIN (Bekaert et al., 2015b) tool for the atmospheric correction, provided it has been independently installed beforehand. Upon completion of the StaMPS processing steps, this module of PHASE seamlessly generates an Excel spreadsheet containing LOS deformation time series for each Persistent Scatterer point within the user-defined area of interest. For further details, users can refer to the accompanying user manual. Specifically, the output table, which encapsulates the results of the PSI analysis, adheres to the structural framework shown in Table 1.

After obtaining the results in Table 1 format, users can effortlessly import them into software like Matlab for temporal deformation analysis, or for conducting more advanced studies such as combining the ascending and descending orbits.

In summary, all the steps executed by this proposed code are outlined below.

1. Configuration of the input parameters through a GUI (Figure 4);

2. Preparation of the data exported in the last step of the DInSAR PS pre-processing module for the StaMPS processing;

3. StaMPS PS analysis upon parameters definition:

   a. Possibility to use the external tool TRAIN.

4. Temporal series export in an Excel table for further analysis.

It is important to clarify that while the pre-processing module is fully compatible with Windows, macOS, and Linux, StaMPS itself requires a Linux environment. Consequently, users must execute this second module of PHASE within a Linux system to ensure proper functionality. The benefits of the proposed software are significant, particularly in preventing common errors arising from the numerous operations required, from data preparation to export. Additionally, the inclusion of a built-in functionality for automatic data export is highly advantageous, since these data are invariably necessary for subsequent analyses.

5. Results

The validation rigorously evaluated the proposed PHASE software ability to handle diverse processing scenarios by testing a wide range of parameters across both modules. Specifically, for the preprocessing module, different configurations were applied to key parameters such as orbit type, burst selection, area of interest size, and digital elevation models (DEM). Similarly, for the StaMPS processing module, various amplitude thresholds and StaMPS-specific settings were tested. This comprehensive evaluation ensured that the software remained robust under different conditions and that any parameter modification effectively introduced a change in the workflow without causing failures. To ensure accuracy, the entire processing workflow underwent dual executions: one with the standard procedure (involving SNAP, snap2stamps and StaMPS with all intermediate manual steps) and one with PHASE. The primary objective was to test the equivalency of the two ap-

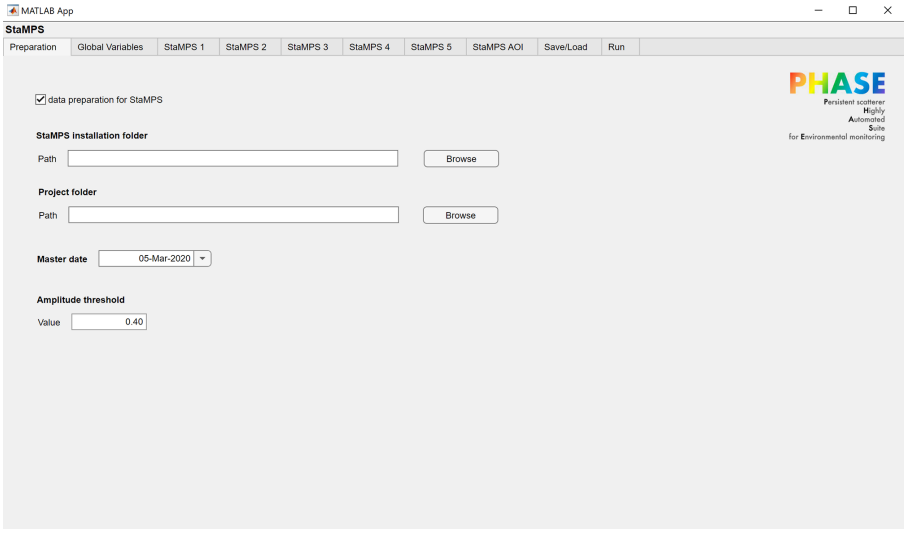
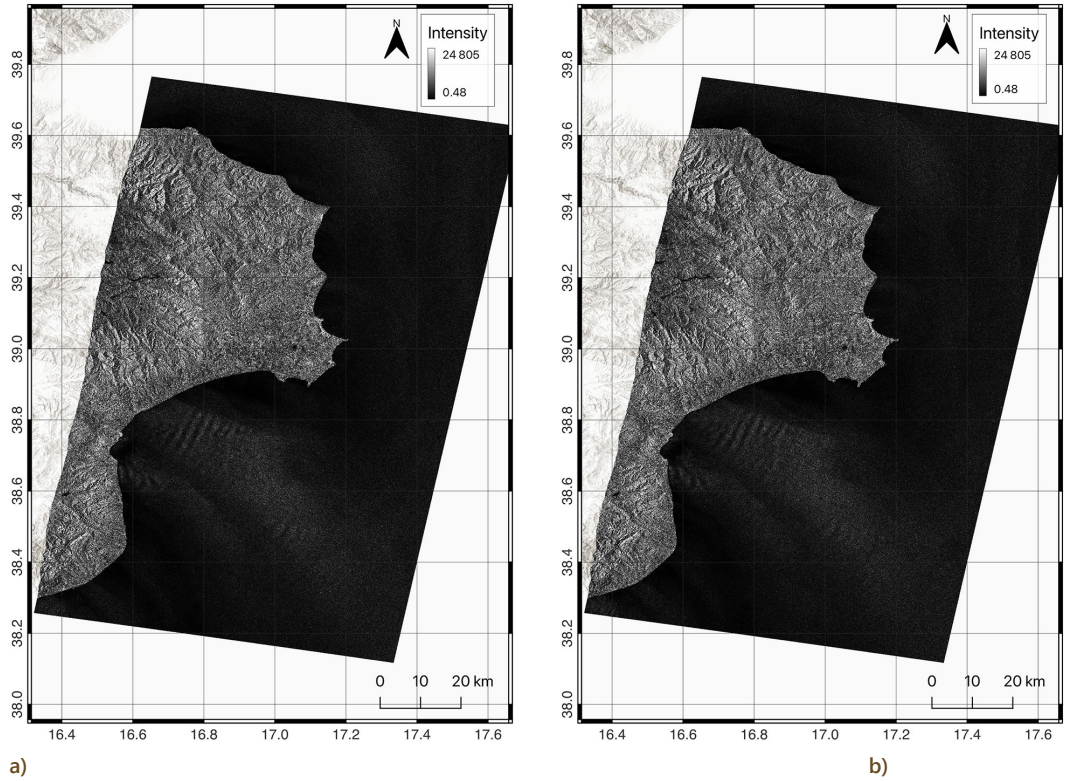


Figure 4. Example tab of the PHASE_StaMPS Matlab application

proaches. Both methods produced identical results across all stages, including intermediate products such as coregistered images and interferograms, as well as the final output (PS displacement time series). This alignment was expected, as PHASE is designed to automate and enhance the existing procedure while leveraging the same computational tools, albeit in an updated and more streamlined manner. To further support this validation, some processing products from the first module are presented, comparing the outcomes from both workflows. The complete set of validation scenarios, including figures and tested parameters configurations, are included in Annex A. For brevity, only a few representative examples are included in this manuscript. Figure 5 displays the master image used in one of the test cases, processed with orbit correction and splitting (along with debursting and terrain projection for visualization purposes). In this context, terrain projection refers to the conversion of SAR's native azimuth-range coordinates into cartographic projected coordinates (e.g., Universal Transverse Mercator (UTM) coordinate system) to facilitate geospatial analysis and interpretation. On the other hand, Figure 6 illustrates an interferogram computed over another test scenario, where the output represents a subset of a burst (subsequently terrain projected for visualization, as before), showcasing the capability of both procedures to handle such configurations.

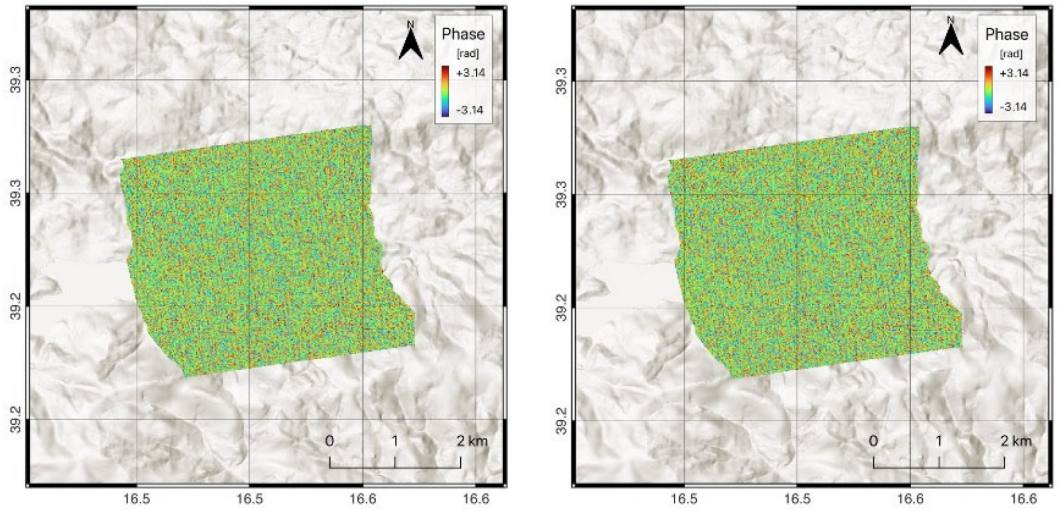
Next, a complete case study is presented, in which the entire processing workflow, encompassing both the preprocessing and PSI analysis modules, is executed and tested. The example focuses on an earth-filled dam in southern Italy, using a dataset of fifty Sentinel-1 SAR images spanning approximately 300 days. The dataset includes ascending orbit images from both Sentinel-1A and Sentinel-1B, resulting in a combined revisit time of six days. The SAR images were acquired through the ASF portal using the built-in download functionality of PHASE, further demonstrating the software's ability to streamline data acquisition. The same dataset, processed with identical settings, underwent a complete PS analysis using both the standard workflow and PHASE. For brevity, only the final results obtained using PHASE are plotted. Figure 7a shows the distribution of PS over the study area, while Figure 7b visualizes the time series associated to some of these PS.

The objective of this secondary validation activity is to provide a quantitative assessment of the advantages introduced by PHASE, with a focus on the reduction of error sources and time savings. This analysis is based on the case study presented before, where the entire processing workflow, from preprocessing to PSI analysis, was executed using both the standard procedure and PHASE. Addressing the first aspect, the PHASE software was meticulously crafted to minimize user inputs, directly correlating with a decrease in potential human errors. Throughout the processing workflow, over 500 potential error sources have been mitigated, largely through automation and simplified configuration via the graphical user interface. Table 2 delineates the approximate reduction in number of the required inputs for each step of the process chain.



Note: The boundaries of the area are expressed in latitude and longitude. The basemap is ESRI World Hillshade.

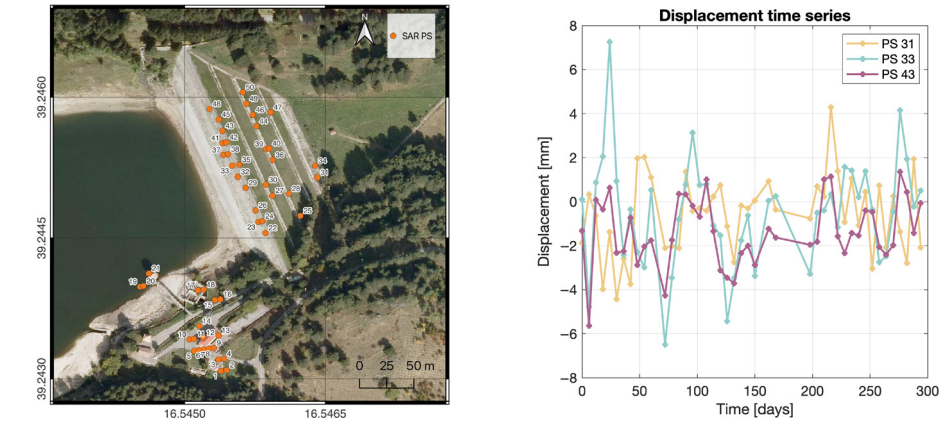
Figure 5. Intensity of the master image of one of the test scenarios. Sentinel-1 descending track D51, bursts 1 to 9: a – is the result with the standard procedure; b – with PHASE software



a) b)

Note: The boundaries of the area are expressed in latitude and longitude. The basemap is ESRI World Hillshade.

Figure 6. Interferometric (wrapped) phase for an area of interest subsampled from the Sentinel-1 ascending track A146, burst 7: a – is the result with the standard procedure; b – with PHASE software



a) b)

Figure 7. Case study area showing the results of the PSI processing: a – position of PS over the Calabria region (Italy) orthophoto; b – raw displacement time series of some PS

Table 2. Comparison of required user input

Processing step	Standard procedure implementation	PHASE implementation	Reduction in manual actions
Master preprocessing	Not implemented	Automated	~10
Parameters for slaves processing	Some from configuration file, some manually	Automated, with parameters from GUI	~100
Slaves processing	Manual execution	Automated	4
COH, LIA, AI	Not implemented	Automated	~n. images × 10
StaMPS	Manual execution	Automated, with parameters from GUI	~50
Time series export	Not implemented	Automated	~300

Now, moving to the aspect of time saving, notable enhancements have been achieved across all processing steps necessitating manual repetition, modification of distributed codes, or manual implementation of unavailable features. Naturally, computational time can fluctuate depending on the hardware configuration; the values presented herein are derived from a setup featuring an Intel Core i7-4770 processor with 8 cores and 32 GB of RAM. In Table 3 a summary of the main computational times is reported, highlighting the gains due to PHASE.

Table 3. Comparison of computational times

Processing step	Standard procedure computational time	PHASE computational time	Time reduction
Master preprocessing	10 min	2 min	80%
Parameters for slaves processing	5–15 min	2 min	60–87%
Slaves processing	610 min	600 min	1.5%
COH, LIA, AI	500 min	50 min	90%
StaMPS	75 min	60 min	20%
Time series export	5–20 min	1 min	80–95 %

The average time reduction stands around 60%, a substantial improvement by any measure. This reduction is particularly significant as it accompanies the near elimination of all potential error sources. Consequently, across all presented scenarios, there is a marked enhancement in processing quality from every angle. Such advancements yield a discernibly positive impact on the intricate task of SAR image processing for PS analysis.

6. Conclusions

The PHASE software suite represents an advancement in the daily processing of SAR images for Persistent Scatterers analysis, acting as bridge between SNAP and StaMPS software, while incorporating the snap2stamps package. As known, this methodology is widely employed for the monitoring of natural phenomena such as subsidence or active landslides, as well as man-made structures and infrastructures. With PHASE, we aim to enhance the capabilities of the existing free tools while preserving their accessibility. Rather than redistributing snap2stamps as a standalone package, we have integrated it into a comprehensive, structured workflow that improves automation and usability. As a result of our efforts, we developed a comprehensive suite centered around two Matlab modules: one for DInSAR pre-processing and one for StaMPS PSI processing. The first module is compatible with Windows, macOS, and Linux, while the second one runs only on Linux due to StaMPS compatibility. Both applications feature a user-friendly GUI that simplifies identifying and configuring the necessary workflow parameters. Additionally, we have updated snap2stamps scripts and configuration files to ensure full compatibility with the latest SNAP release and have developed new scripts with a consistent structure to further streamline the processing. Finally, to facilitate correct usage, we provide a detailed user manual with step-by-step instructions. The entire PHASE software suite is published and distributed through GitHub (https://github.com/robimonti/PHASE).

All the validation tests demonstrated that PHASE reliably produces results identical to those obtained using the standard procedure, as expected, since it is designed to optimize and improve rather than replace the existing workflow. The software successfully processed large-scale datasets with different configurations, confirming its robustness and adaptability. Additionally, the intermediate step examples highlighted PHASE's ability to streamline SAR image processing across various stages, including master image selection, interferogram generation, and PSI analysis, while maintaining accuracy at each step. Automation and user-friendly GUIs reduced manual inputs and potential errors, cutting processing time by an average of 60%, with some steps seeing up to a 95% reduction. These improvements enhance the efficiency and reliability of SAR image processing for PS analysis, making the workflow more accessible to users with varying levels of expertise.

In conclusion, PHASE provides a very useful help in the improvement of the SAR images processing for the PS analysis both in term of computational time and simplicity of operations, maintaining the whole process widely accessible by using free and open source software.

Acknowledgements

We would like to thank all the individuals who contributed to the development and maintenance of snap2stamps and StaMPS.

Funding

This work received no external fundings.

Author contributions

RM and LR conceived the study and were responsible for the design and development of the software structure. RM was responsible for software development. LR contributed to its improvement. RM wrote the first draft of the article. LR revised and improved it. RM and LR worked together for the final version.

Disclosure statement

The author has no competing financial, professional, or personal interest in the other party.

References

Amelung, F., Galloway, D. L., Bell, J. W., Zebker, H. A., & Laczniak, R. J. (1999). Sensing the ups and downs of Las Vegas: InSAR reveals structural control of land subsidence and aquifer-system deformation. *Geology*, *27*(6), Article 483. https://doi.org/10.1130/0091-7613(1999)027<0483:STUADO>2.3.CO;2

Antonielli, B., Monserrat, O., Bonini, M., Righini, G., Sani, F., Luzi, G., Feyzullayev, A., & Aliyev, C. (2014). Pre-eruptive ground deformation of Azerbaijan mud volcanoes detected through satellite radar interferometry (DInSAR). *Tectonophysics*, *637*, 163–177. https://doi.org/10.1016/j.tecto.2014.10.005

Bekaert, D. P. S., Hooper, A., & Wright, T. J. (2015a). A spatially variable power law tropospheric correction technique for InSAR data. *Journal of Geophysical Research: Solid Earth*, *120*(2), 1345–1356. https://doi.org/10.1002/2014JB011558

Bekaert, D. P. S., Walters, R. J., Wright, T. J., Hooper, A. J., & Parker, D. J. (2015b). Statistical comparison of InSAR tropospheric correction techniques. *Remote Sensing of Environment*, *170*, 40–47. https://doi.org/10.1016/j.rse.2015.08.035

Carnec, C., Massonnet, D., & King, C. (1996). Two examples of the use of SAR interferometry on displacement fields of small spatial extent. *Geophysical Research Letters*, *23*(24), 3579–3582. https://doi.org/10.1029/96GL03042

Covello, F., Battazza, F., Coletta, A., Lopinto, E., Fiorentino, C., Pietranera, L., Valentini, G., & Zoffoli, S. (2010). COSMO-SkyMed an existing opportunity for observing the Earth. *Journal of Geodynamics*, *49*(3–4), 171–180. https://doi.org/10.1016/j.jog.2010.01.001

Crosetto, M., Devanthéry, N., Cuevas-González, M., Monserrat, O., & Crippa, B. (2015). Exploitation of the full potential of PSI data for subsidence monitoring. *Proceedings of IAHS*, *372*, 311–314. https://doi.org/10.5194/piahs-372-311-2015

Crosetto, M., Monserrat, O., Cuevas, M., & Crippa, B. (2011). Spaceborne differential SAR interferometry: Data analysis tools for deformation measurement. *Remote Sensing*, *3*(2), 305–318. https://doi.org/10.3390/rs3020305

Dalla Via, G., Crosetto, M., & Crippa, B. (2012). Resolving vertical and east-west horizontal motion from differential interferometric synthetic aperture radar: The L'Aquila earthquake. *Journal of Geophysical Research: Solid Earth*, *117*(B2), Article B02310. https://doi.org/10.1029/2011JB008689

Delgado Blasco, J. M., Foumelis, M., Stewart, C., & Hooper, A. (2019). Measuring urban subsidence in the Rome Metropolitan Area (Italy) with Sentinel-1 SNAP-StaMPS persistent scatterer interferometry. *Remote Sensing*, *11*(2), Article 129. https://doi.org/10.3390/rs11020129

European Space Agency. (2018a, May 4). *How to prepare Sentinel-1 images stack for PSI/SBAS in SNAP – Microwave Toolbox / StaMPS* [Online post]. STEP Forum. https://forum.step.esa.int/t/how-to-prepare-sentinel-1-images-stack-for-psi-sbas-in-snap/4981/514

European Space Agency. (2018b, July 1). *Snap2stamps package: A free tool to automate the SNAP-StaMPS Workflow—Microwave Toolbox / StaMPS - STEP Forum* [Online post]. STEP Forum. https://forum.step.esa.int/t/snap2stamps-package-a-free-tool-to-automate-the-snap-stamps-workflow/10971

European Space Agency. (2022, March 8). *Sentinel-1B in-flight anomaly summary report*. https://sentinel.esa.int/documents/247904/4819394/Sentinel-1B+In-Flight+Anomaly+Summary+Report.pdf

European Space Agency. (2023, December 12). *STEP – Science Toolbox Exploitation Platform*. https://step.esa.int/main/

Ferretti, A., Prati, C., & Rocca, F. (2000). Nonlinear subsidence rate estimation using permanent scatterers in differential SAR interferometry. *IEEE Transactions on Geoscience and Remote Sensing*, *38*(5), 2202–2212. https://doi.org/10.1109/36.868878

Ferretti, A., Prati, C., & Rocca, F. (2001). Permanent scatterers in SAR interferometry. *IEEE Transactions on Geoscience and Remote Sensing*, *39*(1), 8–20. https://doi.org/10.1109/36.898661

Fletcher, K. (2012). *Sentinel-1: ESA's radar observatory mission for GMES operational services*. ESA Communications.

Foumelis, M., Delgado Blasco, J. M., Desnos, Y.-L., Engdahl, M., Fernandez, D., Veci, L., Lu, J., & Wong, C. (2018). Esa Snap – stamps integrated processing for Sentinel-1 persistent scatterer interferometry. In *IGARSS 2018 – 2018 IEEE International Geoscience and Remote Sensing Symposium* (pp. 1364–1367). IEEE. https://doi.org/10.1109/IGARSS.2018.8519545

Galloway, D. L., Hudnut, K. W., Ingebritsen, S. E., Phillips, S. P., Peltzer, G., Rogez, F., & Rosen, P. A. (1998). Detection of aquifer system compaction and land subsidence using interferometric synthetic aperture radar, Antelope Valley, Mojave Desert, California. *Water Resources Research*, *34*(10), 2573–2585. https://doi.org/10.1029/98WR01285

Goldstein, R. M., Engelhardt, H., Kamb, B., & Frolich, R. M. (1993). Satellite radar interferometry for monitoring ice sheet motion: Application to an Antarctic ice stream. *Science*, *262*(5139), 1525–1530. https://doi.org/10.1126/science.262.5139.1525

Herrera, G., Tomás, R., Lopez-Sanchez, J. M., Delgado, J., Mallorqui, J. J., Duque, S., & Mulas, J. (2007). Advanced DInSAR analysis on mining areas: La Union case study (Murcia, SE Spain). *Engineering Geology*, *90*(3), 148–159. https://doi.org/10.1016/j.enggeo.2007.01.001

Hooper, A., Bekaert, D., Spaans, K., & Arıkan, M. (2012). Recent advances in SAR interferometry time series analysis for measuring crustal deformation. *Tectonophysics*, *514–517*, 1–13. https://doi.org/10.1016/j.tecto.2011.10.013

Hooper, A., Zebker, H., Segall, P., & Kampes, B. (2004). A new method for measuring deformation on volcanoes and other natural terrains using InSAR persistent scatterers. *Geophysical Research Letters*, *31*(23), Article L23611. https://doi.org/10.1029/2004GL021737

Louet, J., & Bruzzi, S. (1999). ENVISAT mission and system. *IEEE 1999 International Geoscience and Remote Sensing Symposium. IGARSS'99 (Cat. No.99CH36293)*, *3*, 1680–1682. https://doi.org/10.1109/IGARSS.1999.772059

Massonnet, D., & Sigmundsson, F. (2000). Remote sensing of volcano deformation by radar interferometry from various satellites. *Washington DC American Geophysical Union Geophysical Monograph Series*, *116*, 207–221. https://doi.org/10.1029/GM116p0207

Massonnet, D., Briole, P., & Arnaud, A. (1995). Deflation of Mount Etna monitored by spaceborne radar interferometry. *Nature*, *375*(6532), Article 6532. https://doi.org/10.1038/375567a0

Massonnet, D., Rossi, M., Carmona, C., Adragna, F., Peltzer, G., Feigl, K., & Rabaute, T. (1993). The displacement field of the Landers earthquake mapped by radar interferometry. *Nature*, *364*, 138–142. https://doi.org/10.1038/364138a0

Mateo-Garcia, G., Veitch-Michaelis, J., Smith, L., Oprea, S. V., Schumann, G., Gal, Y., Baydin, A. G., & Backes, D. (2021). Towards global flood mapping onboard low cost satellites with machine learning. *Scientific Reports*, *11*(1), Article 7249. https://doi.org/10.1038/s41598-021-86650-z

Narayan, A. B., Tiwari, A., Dwivedi, R., & Dikshit, O. (2018). Persistent scatter identification and look-angle error estimation using similar time-series interferometric pixels. *IEEE Geoscience and Remote Sensing Letters*, *15*(1), 147–150. https://doi.org/10.1109/LGRS.2017.2778421

Peltzer, G., & Rosen, P. (1995). Surface displacement of the 17 May 1993 Eureka Valley, California, Earthquake observed by SAR interferometry. *Science, 268*(5215), 1333–1336. https://doi.org/10.1126/science.268.5215.1333

Pitz, W., & Miller, D. (2010). The TerraSAR-X satellite. *IEEE Transactions on Geoscience and Remote Sensing*, *48*(2), 615–622. https://doi.org/10.1109/TGRS.2009.2037432

Rignot, E. J., Gogineni, S. P., Krabill, W. B., & Ekholm, S. (1997). North and northeast Greenland ice discharge from satellite radar interferometry. *Science*, *276*(5314), 934–937. https://doi.org/10.1126/science.276.5314.934

Van Leijen, F. J. (2014). *Persistent Scatterer Interferometry based on geodetic estimation theory*. https://repository.tudelft.nl/islandora/object/uuid%3A5dba48d7-ee26-4449-b674-caa8df93e71e

Zhang, B., Chang, L., & Stein, A. (2021). Spatio-temporal linking of multiple SAR satellite data from medium and high resolution Radarsat-2 images. *ISPRS Journal of Photogrammetry and Remote Sensing*, *176*, 222–236. https://doi.org/10.1016/j.isprsjprs.2021.04.005