The Journal of Open Source Software

InsarViz: An open source Python package for the interactive visualization of satellite SAR interferometry data

Margaux Mouchene [1,2¶], **Renaud Blanch** [2], **Erwan Pathier** [1], **Romain Montel**[1], **and Franck Thollard** [1]

1 Univ. Grenoble Alpes, Univ. Savoie Mont Blanc, CNRS, IRD, Univ. Gustave Eiffel, ISTerre, 38000 Grenoble, France **2** Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, 38000 Grenoble France ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

The deformation of the Earth surface or of man-made infrastructures can be studied using satellite Synthetic Aperture Radar (SAR) Interferometry (InSAR). Thanks to new satellite missions and improvements in the complex data processing chains, large amounts of high-quality InSAR data are now readily available. However, some characteristics of these datasets make them unsuitable to be studied using conventional (geo)imagery softwares. We present InsarViz, a new Python tool designed specifically to interactively visualize and analyze large InSAR datasets.

Statement of needs

Satellite Synthetic Aperture Radar (SAR) Interferometry (InSAR) is a well-established technique in Earth Observation (EO) that enables very high precision monitoring of ground displacements (mm/year). This method combines high spatial resolution data (up to a few meters) and large coverage capabilities (up to continental scale) with a fairly high temporal resolution (a few days to a few weeks). It is used to study a wide range of phenomena that impact the Earth surface (e.g. earthquakes, landslides, permafrost evolution, volcanoes, glaciers dynamics, subsidence, building and infrastructure deformation, etc.).

For several reasons (data availability, non-intuitive radar image geometry, complexity of the processing, etc.), InSAR has long remained a niche technology and few free open-source tools have been dedicated to it compared to the widely-used multi-purpose optical imagery. Most existing tools are focused on data processing (e.g. ROI_PAC (Rosen et al., 2004), DORIS (*DORIS*, 2017), GMTSAR (Sandwell et al., 2016), StaMPS (*StaMPS*, 2018), ISCE (Rosen et al., 2012), NSBAS (Doin et al., 2011), OrfeoToolBox (*Orfeo ToolBox*, 2022), SNAP (*SNAP Toolbox*, 2022), LICSBAS (Morishita et al., 2020)). Generic remote-sensing or Geographic Information System (GIS) softwares are limited when used to visualize InSAR data because of their unusual geometry and formats. Some visualization tools with dedicated InSAR functionalities, like the pioneer MDX software (*MDX*, 2020), or the ESA SNAP toolbox (*SNAP Toolbox*, 2022), were designed to visualize a single radar image or interferogram.

However, recent spatial missions like the Sentinel-1 mission of the European program COPERNICUS, with a systematic background acquisition strategy and an open data policy, provide unprecedented access to massive SAR datasets. From these new datasets, a network of thousands of interferograms can be generated over a single area. The consecutive step is a time-series analysis which produces a spatiotemporal data cube: a layer of this data cube is a

2D map that contains the displacement of each pixel of an image relative to the same pixel in the reference date image. A typical data cube size is 4000x6000x200, where 4000x6000 are the spatial dimensions (pixels) and 200 is a typical number of images taken since the beginning of the spatial mission.

The aforementioned tools are not suited to allow fluid and interactive data visualization of such large and multifaceted datasets. If data cube visualization is a more generic problem and an active research topic in EO and beyond, some specifics of InSAR (radar geometry, wrapped phase, relative measurement in space and in time, multiple types of products needed for interpretation…) call for a new, dedicated visualization tool.

Overview of functionality

InsarViz was prototyped and designed, and is continuously developed, in close interaction with the geophysicists (end-users) through interviews and work observations by the developing team (UX-design). Our focus is on making this tool ergonomic and intuitive, and providing pertinent functionalities to explore the datasets, while maintaining performance and accuracy (stay true to data).

InsarViz allows visualization and access to data from the spatiotemporal data cube of InSAR time-series (displacement maps). When loading such a data cube, the user can visualize and navigate spatially (general view and synchronized zoomed-in view of a map from the series) and/or temporally (switch between maps), in radar or ground geometry. Hovering the cursor on the map directly gives access to the data from the map and from the whole temporal series (temporal profile drawn on-the-fly). A separate panel can be used to plot and extract data from selected points or profiles on the map. A parametrized trend can then be fitted and subtracted from the observed data to discern physical processes. Publication-ready figures of the maps and plots can easily be exported in multiple common formats.

In future versions of this tool, the user will be able to concurrently load other images (other products of the processing chain, DEM, etc.) for further analysis (quality assessment, etc.).

The main technical characteristics of the tool are:

- InsarViz is a standalone application that takes advantage of the hardware (i.e. GPU, SSD hard drive, capability to run on cluster). We choose the Python language for its well-known advantages (interpreted, readable language, large community) and we use QT for the graphical user interface and OpenGL for the hardware graphical acceleration.
- InsarViz uses the GDAL library (GDAL/OGR contributors, 2024) to load the data. This allows to handle all the input formats most widely used by the community (e.g. GeoTIFF). Moreover, we plan on developing a plug-in data loader template to easily manage custom data formats in the near future.
- We take advantage of the Python/QT/OpenGL stack to ensure efficient user interaction with the data. For example, they allow the fluid, rapid switching between large maps and on-the-fly plotting.
- Visualization tools commonly use aggregation methods (e.g. smoothing, averaging, clustering) to drastically accelerate image display, but they thus induce observation and interpretation biases that are detrimental to the user. To avoid those bias, we focus on staying true to the original data and allowing the user to customize the rendering manually (color-scale, outliers selection, level-of-detail).

The Journal of Open Source Software

Example Use Case

The following figure shows a screenshot of the `ts_viz` program of the `InsarViz` package on data provided by the Flatsim service (Thollard et al., 2021). This example shows the displacement of a point in the *Line of Sight* of the satellite in a period of time that covers the Pueblo Earthquake (2019/09/19).

Color on the map shows the displacement with respect to the previous date (yellow means going away from the satellite). The colorbar in the middle allows the user to interactively change the dynamic of the color on the map. The curve on the right shows the displacement, in the direction of the satellite, of the point under the mouse (cross). The curve is dynamically updated while the user moves the mouse on the map.

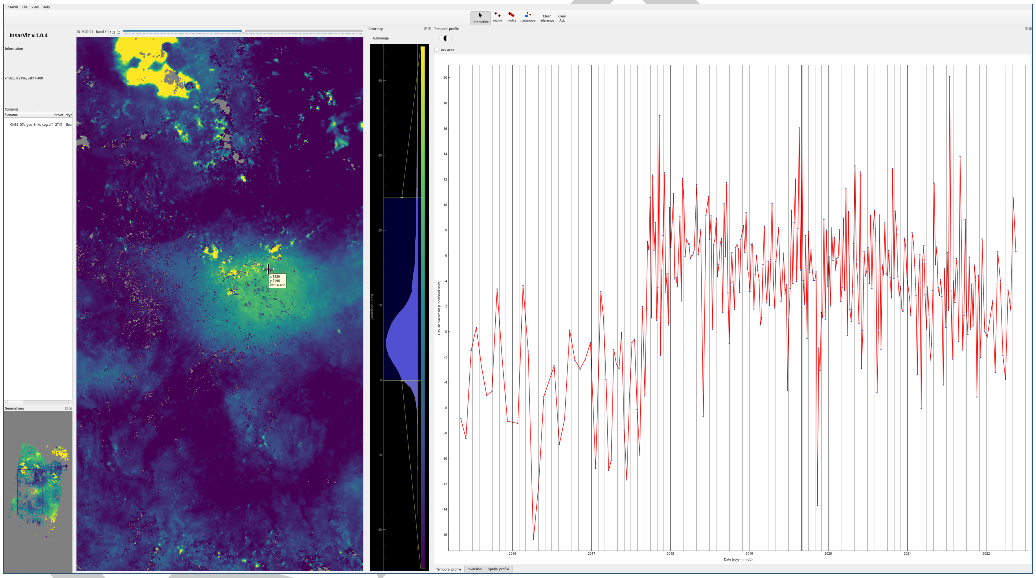

Figure 1: Visualisation of a data-cube of Mexico. Displacement at the localisation of the Puebla Earthquake, 2017/09/19

Development Notes

InsarViz is developed on the Université de Grenoble's GitLab as an open-source package, and the authors welcome feature suggestions and contributions. We use the pytest package to test and ensure the code quality.

Acknowledgements

This project was financially supported by CNES as an application of the SENTINEL1 mission. The authors would like to thank the Editor and the Reviewers for their time and comments that helped improve the manuscript and the code.

References

Doin, M.-P., Lodge, F., Guillaso, S., Jolivet, R., Lasserre, C., Ducret, G., Grandin, R., Pathier, E., & Pinel, V. (2011). Presentation of the small baseline NSBAS processing chain on a case example: The etna deformation monitoring from 2003 to 2010 using envisat data. *Fringe Symposium 2011*.

DORIS. (2017). Delft University of Technology. https://github.com/TUDelftGeodesy/Doris

GDAL/OGR contributors. (2024). *GDAL/OGR geospatial data abstraction software library*. Open Source Geospatial Foundation. https://doi.org/10.5281/zenodo.5884351

MDX. (2020). Jet Propulsion Lab NASA. https://software.nasa.gov/software/NPO-35238-1

Morishita, Y., Lazecky, M., Wright, T. J., Weiss, J. R., Elliott, J. R., & Hooper, A. (2020). LiCSBAS: An open-source InSAR time series analysis package integrated with the LiCSAR automated sentinel-1 InSAR processor. *Remote Sensing*, *12*(3). https://doi.org/10.3390/rs12030424

Orfeo ToolBox. (2022). [Computer software]. CNES. https://www.orfeo-toolbox.org/CookBook/

Rosen, P. A., Gurrola, E., Sacco, G. F., & Zebker, H. (2012). The InSAR scientific computing environment. *EUSAR 2012; 9th European Conference on Synthetic Aperture Radar*, 730–733.

Rosen, P. A., Hensley, S., Peltzer, G., & Simons, M. (2004). Updated repeat orbit interferometry package released. *Eos, Transactions American Geophysical Union*, *85*(5), 47–47. https://doi.org/10.1029/2004EO050004

Sandwell, D., Mellors, R., Tong, X., Xu, X., Wei, M., & Wessel, P. (2016). *GMTSAR: An InSAR processing system based on generic mapping tools*. http://topex.ucsd.edu/gmtsar/tar/GMTSAR_2ND_TEX.pdf

SNAP toolbox. (2022). European Space Agency. https://earth.esa.int/eogateway/tools/snap

StaMPS. (2018). https://github.com/dbekaert/StaMPS

Thollard, F., Clesse, D., Doin, M.-P., Donadieu, J., Durand, P., Grandin, R., Lasserre, C., Laurent, C., Deschamps-Ostanciaux, E., Pathier, E., Pointal, E., Proy, C., & Specht, B. (2021). FLATSIM: The ForM@ter LArge-scale multi-temporal sentinel-1 InterferoMetry service. *Remote Sensing*, *13*(18). https://doi.org/10.3390/rs13183734