California Institute of Technology

Keck Institute for Space Studies
Post-doctoral Fellowship Report

Piyush Agram

Mar 30, 2013

Summary

The Keck Institute for Space Studies (KISS) fellowship has played an important role in my development as an engineer- scientist. The fellowship gave me an opportunity to pursue exciting new ideas and work on novel concepts leading to important scientific advancements in monitoring surface deformation using space-based radar satellites. My experiences from the research work as a KISS fellow have played an important role in shaping my decision to further pursue research on effectively using space-based radar instruments for monitoring and studying surface deformation at JPL. The KISS fellowship also allowed me, as an engineer, to reach out and collaborate directly with geophysicists on problems of mutual interest. My work consisted of developing new algorithms and analysis techniques for studying the spatial and temporal evolution of surface deformation using data from space-borne synthetic aperture radar (SAR) systems. The key contributions of my research during the fellowship are:

1. **A theoretical noise model for synthetic aperture radar interferometric data.**

   The new noise model will drive the next generation of algorithms for estimating the spatial and temporal evolution of surface deformation from radar interferometric observations. The noise model will also enable us to provide realistic uncertainty estimates for our observations and hence, will influence the design of future interferometric SAR missions. We expect this work to be submitted for publication in a research journal soon.

2. **Improvements to various aspects of time-series radar interferometry.**

   In collaboration with members of Prof. Mark Simon's research group:

   (a) Algorithms for correcting effects due to the stratified atmosphere in interferometric data.

   (b) Improve and optimize the Multi-scale Interferometric Time Series (MInTS) algorithms for practical implementation on large datasets.

   (c) Development of an automated processing chain for processing large volumes of SAR data into interferograms. For the first time, it is now possible to automatically process hundreds of interferograms in a reasonable amount of time with open source software. The implications for automated monitoring of various geophysically active regions around the world are numerous.

3. **Development of the Generic InSAR Analysis Toolbox (GIAnT).**

   In collaboration with members of Prof. Mark Simon's group, I developed an automated, open source interferometric data analysis package called Generic InSAR Analysis Toolbox (GIAnT). GIAnT enables geophysicists to quickly and efficiently derive surface deformation maps from large volumes of SAR data. GIAnT has already been applied to study some of the largest SAR datasets ever compiled in the United States, e.g. over Long Valley caldera in California, Los Angeles and the Mojave desert. GIAnT is expected to become the time-series analysis tool of choice in the future, and has been developed with the JPL's next generation InSAR Scientific Computing Environment (ISCE) in mind. An article announcing the public release of this software package has already been published in American Geophysical Union's EOS newsletter. GIAnT can be downloaded for free at http://earthdef.caltech.edu.

4. **Algorithms for optimizing GPS networks.**

   I also developed network optimization algorithms for optimially extending GPS networks for monitoring specific seismogenic zones. An article describing the application of these algorithms for augmenting the continuous GPS network in Sumatra, Indonesia will be submitted to a scientific journal.

5. **Modelling creep along the San Andreas Fault.**

   In collaboration with Sylvain Barbot and Romain Jolivet of the Tectonics Observatory at Caltech, the surface deformation time-series products over Parkfield, CA derived using various algorithms that I developed are currently being used to study variations in creep along the San Andreas Fault. This is a project in progress and is expected to result in a couple of journal publications.

Outline

This report is composed of articles documenting my research work during the fellowship, that have either been published or are being prepared for submission to a scientific journal. The attachments include

1. P. S. Agram, R. Jolivet, B. Riel, Y. N. Lin, M. Simons, E. Hetland, M. -P. Doin and C. Lasserre, "New radar interferometric time-series package released", *Eos, Transactions American Geophysical Union* 92.28 (2012): 234.

2. P. S. Agram, R. Jolivet and M. Simons, "Generic InSAR Analysis Toolbox (GIAnT) - User Guide", 2012, http://earthdef.caltech.edu.

3. P. S. Agram and M. Simons, "A noise model for InSAR time-series", In preparation.

4. S. Barbot, P. S. Agram, M. Simons and M. De Michele, "Segmentation of the San Andreas Fault from space geodetic data", In preparation.

5. S. Barbot, P. S. Agram and E. Hill, "Augmenting the space geodetic monitoring of the Mentawai seismic gap of the Sundra megathrust", In preparation.

Acknowledgement

I am grateful to Prof. Mark Simons, his research group members and the members of the Seismological Laboratory for their encouragement, support and guidance. I would also like to thank Prof. Jean-Philippe Avouac and members of the Tectonics Observatory for their support. I am grateful for having the opportunity to be part of the group of people comprising KISS and hope that these relationship will be long-lasting.

EOS
EOS, TRANSACTIONS, AMERICAN GEOPHYSICAL UNION

VOLUME 94 NUMBER 7
12 FEBRUARY 2013
PAGES 69–76

New Radar Interferometric Time Series Analysis Toolbox Released

PAGES 69–70

Interferometric synthetic aperture radar (InSAR) has become an important geodetic tool for measuring deformation of Earth's surface due to various geophysical phenomena, including slip on earthquake faults, subsurface migration of magma, slow-moving landslides, movement of shallow crustal fluids (e.g., water and oil), and glacier flow. Airborne and spaceborne synthetic aperture radar (SAR) instruments transmit microwaves toward Earth's surface and detect the returning reflected waves. The phase of the returned wave depends on the distance between the satellite and the surface, but it is also altered by atmospheric and other effects. InSAR provides measurements of surface deformation by combining amplitude and phase information from two SAR images of the same location taken at different times to create an interferogram. Several existing open-source analysis tools [*Rosen et al.*, 2004; *Rosen et al.*, 2011; *Kampes et al.*, 2003; *Sandwell et al.*, 2011] enable scientists to exploit observations from radar satellites acquired at two different epochs to produce a surface displacement map.

The past decade has seen the development and verification of numerous algorithms that combine phase information from multiple radar interferograms to produce internally consistent time series of land surface deformation [e.g., *Ferretti et al.*, 2001; *Berardino et al.*, 2002; *López-Quiroz et al.*, 2009; *Hetland et al.*, 2012]. Combining multiple interferograms allows detection and quantification of both secular and transient displacements. These methods also help to mitigate the effects of change in scatterer properties and phase delay introduced by the atmosphere between SAR acquisitions, resulting in measurements of surface deformation with subcentimeter accuracy.

BY P. S. AGRAM, R. JOLIVET, B. RIEL, Y. N. LIN, M. SIMONS, E. HETLAND, M.-P. DOIN, AND C. LASSERRE

A new repeat interferometry time series analysis toolbox, Generic InSAR Analysis Toolbox (GIAnT) 1.0, was released in December 2012. GIAnT 1.0 is a user-friendly, open-source, documented framework for rapid generation of time series of surface displacement using InSAR data. GIAnT 1.0 includes numerous published time series techniques, in some cases with improvements, allowing geophysicists to efficiently analyze the large and ever-increasing archive of SAR data acquired over the past 2 decades as well as allowing scientists to test the sensitivity of results to different analysis approaches.

A typical processing chain for generating InSAR time series products consists of (1) assembling a stack of phase-unwrapped interferograms; (2) optionally applying corrections, also known as atmospheric phase screens (APS), to mitigate the differential path delay effects due to the stratified atmosphere; (3) optionally estimating residual long-wavelength errors (e.g., due to imprecise orbits) empirically or through the use of other prior information such as surface displacement fields provided by dense GPS networks; and (4) estimating time series of line-of-sight displacements and residual turbulent APS using one of several time series analysis methods.

GIAnT 1.0 addresses steps 2 to 4 in the processing chain and includes implementations of various time series analysis methods for step 4, while allowing users to implement step 1 using their favorite processing tools [e.g., *Rosen et al.*, 2004; *Rosen et al.*, 2011; *Kampes et al.*, 2003; *Doin et al.*, 2011; *Sandwell et al.*, 2011].

GIAnT 1.0 enables mitigation of the effects of signal delays due to the stratified troposphere in each interferogram using either an empirical approach or estimates from global atmospheric models. Empirical estimates are based on the evaluation of the dependency of interferometric phase on topography and the stratification of the lower atmosphere [e.g., *Lin et al.*, 2010]. Alternatively, global atmospheric models provide daily estimates of atmospheric variables, including temperature, pressure, and water vapor partial pressure, which in turn can be used to derive the phase delay related to spatial and temporal variations in the refractivity index of air [e.g., *Jolivet et al.*, 2011]. GIAnT 1.0 implements atmospheric corrections as a stand-alone Python module named PyAPS (Python-based Atmospheric Phase Screen) and includes support for automatic download of meteorological data sets (European Centre for Medium-Range Weather Forecasts (ECMWF) Re-Analysis (ERA) Interim, North American Regional Reanalysis (NARR), and Modern-Era Retrospective Analysis for Research and Applications (MERRA)). GIAnT 1.0 can optionally correct each interferogram from residual orbit errors by removing a simple parametric function determined empirically or using GPS-derived time series of displacements or velocities. All corrections are consistently applied within a given interferometric data set and generally increase the signal-to-noise ratio of inferred time series.

GIAnT 1.0 implements four existing InSAR time series approaches, and new ones are easily added to the toolbox. These approaches are the Small Baseline Subset (SBAS) [*Berardino et al.*, 2002], the New-SBAS (NSBAS) [*López-Quiroz et al.*, 2009], a temporally parameterized inversion (TimeFun), and the Multiscale Interferometric Time-Series (MInTS) [*Hetland et al.*, 2012] algorithms. In SBAS and NSBAS algorithms the temporal evolution of the phase is derived assuming each interferogram is the linear combination of each SAR acquisition's phase value. Additionally, the NSBAS method takes advantage of a user-defined functional form of the phase evolution to overcome the issue of missing links in the interferometric network due to temporal and spatial decorrelation. The MInTS approach allows the characterization of the temporal behavior of surface deformation using a dictionary of user-defined functions, including linear trends, seasonal oscillations, steps, exponential and logarithmic decays, and various splines [*Hetland et al.*, 2012]. MInTS also transforms InSAR observations into the spatial wavelet domain and allows for distinction between different spatial scales of deformation and atmospheric noise. Within GIAnT 1.0 the temporal inversion component of MInTS has also been adapted to the conventional nonwavelet approaches (TimeFun).

Each of these methods includes a data-driven bootstrapping approach to estimate uncertainties associated with time series products. While numerous variants of published time series algorithms exist, GIAnT provides several tools in a simple and efficient framework so that users can test a variety of techniques and customize their processing chain, specific to a given data set. Users are encouraged to make their modifications or even new algorithms available for inclusion in future distributions of GIAnT. The goal is to make GIAnT an open collaborative environment for InSAR time series analysis.

GIAnT 1.0 is primarily an ensemble of Python routines but includes an interface for some optimized C and Fortran 90 routines. GIAnT 1.0 relies extensively on numerical Python libraries to develop an object-oriented, flexible, and generalized framework for InSAR time series applications. The user manual describes available scripts and functions and includes detailed instructions for installing the set of prerequisite libraries using standard repository management tools on Linux and OS X platforms. The developers are heavy users of GIAnT for their own geophysical projects, and they will attempt to fix software bugs as they arise.

GIAnT 1.0 is available from http://earthdef.caltech.edu. The Web site includes details regarding access to the version-controlled software repository and a user discussion forum and wiki. Other related packages can also be obtained from the same Web site. While not designed to match the standards of a well-maintained commercial package, GIAnT 1.0 provides a set of tools to be used by researchers who need the flexibility and access to various stages of processing in InSAR time series applications. Future versions of GIAnT will include support for working directly with wrapped interferometric data, persistent scatterer algorithms, improved constrained and regularized solvers, automatic correction of elastic ocean tidal load response, and direct download of APS maps from third-party projects. An immediate gain from using GIAnT is the ability for able and willing users to easily share large interferometric data sets in a standard format and to compare the performance of various time series approaches on any data set in a common framework. The rich suite of library functions that is distributed with GIAnT 1.0 should also facilitate faster development and prototyping of new InSAR time series processing algorithms.

Acknowledgments

GIAnT has been developed through collaboration with many researchers. In particular, we acknowledge Pablo Muse and Gilles Peltzer. We also thank two anonymous reviewers and the editor for their valuable comments. This work was supported by the Keck Institute for Space Studies and the Tectonics Observatory at Caltech. In addition, this work was partially supported by NASA grant NNX09AD25G and by the French Agence Nationale pour la Recherche grant EFIDIR. This is Caltech's Seismological Laboratory contribution 10085 and Tectonics Observatory contribution 214.

References

Berardino, P., G. Fornaro, R. Lanari, and E. Sansosti (2002), A new algorithm for surface deformation monitoring based on small baseline differential SAR interferograms, *IEEE Trans. Geosci. Remote Sens.*, *40*(11), 2375–2383.

Doin, M. P., S. Guillaso, R. Jolivet, C. Lasserre, F. Lodge, G. Ducret, and R. Grandin (2011), Presentation of the small baseline NSBAS processing chain on a case example: The Etna deformation monitoring from 2003 to 2010 using Envisat data, paper SP-697 presented at FRINGE Symposium, Eur. Space Agency, Frascati, Italy.

Ferretti, A., C. Prati, and F. Rocca (2001), Permanent scatterers in SAR interferometry, *IEEE Trans. Geosci. Remote Sens.*, *39*(1), 8–20.

Hetland, E. A., P. Muse, M. Simons, Y. N. Lin, P. S. Agram, and C. J. DiCaprio (2012), Multiscale InSAR Time Series (MInTS) analysis of surface deformation, *J. Geophys. Res.*, *117*, B02404, doi:10.1029/2011JB008731.

Jolivet, R., R. Grandin, C. Lasserre, M. P. Doin, and G. Peltzer (2011), Systematic InSAR tropospheric phase delay corrections from global meteorological reanalysis data, *Geophys. Res. Lett.*, *38*, L17311, doi:10.1029/2011GL048757.

Kampes, B., R. Hanssen, and Z Perski (2003), Radar interferometry with public domain tools, paper presented at FRINGE Symposium, Eur. Space Agency, Frascati, Italy.

Lin, Y. N., M. Simons, E. A. Hetland, P. Muse, and C. DiCaprio (2010), A multiscale approach to estimating topographically correlated propagation delays in radar interferograms, *Geochem. Geophys. Geosyst.*, *11*, Q09002, doi:10.1029/2010GC003228.

López-Quiroz, P., M.-P. Doin, F. Tupin, P. Briole, and J.-M. Nicolas (2009), Time series analysis of Mexico City subsidence constrained by radar interferometry, *J. Appl. Geophys.*, *69*, 1–15, doi:10.1016/j.jappgeo.2009.02.006.

Rosen, P., S. Hensley, G. Peltzer, and M. Simons (2004), Updated repeat orbit interferometry package released, *Eos Trans. AGU*, *85*(5), 47, doi:10.1029/2004EO050004.

Rosen, P., E. Gurrola, G. Sacco, and H. A. Zebker (2011), InSAR scientific computing environment—The home stretch, poster IN43B-1397 presented at 2010 Fall Meeting, AGU, San Francisco, Calif., 13–17 Dec.

Sandwell, D., R. Mellors, X. Tong, M. Wei, and P. Wessel (2011), Open radar interferometry software for mapping surface deformation, *Eos Trans. AGU*, *92*(28), 234, doi:10.1029/2011EO280002.

—P. S. Agram, R. Jolivet, B. Riel, Y. N. Lin, and M. Simons, Seismological Laboratory, California Institute of Technology, Pasadena, Calif.; E-mail: piyush@gps.caltech.edu; E. Hetland, Earth and Environmental Science, University of Michigan, Ann Arbor; and M.-P. Doin and C. Lasserre, Institut des Sciences de la Terre, Centre National de la Recherche Scientifique (CNRS), Université Joseph Fourier, Grenoble, France

Attachment 1:

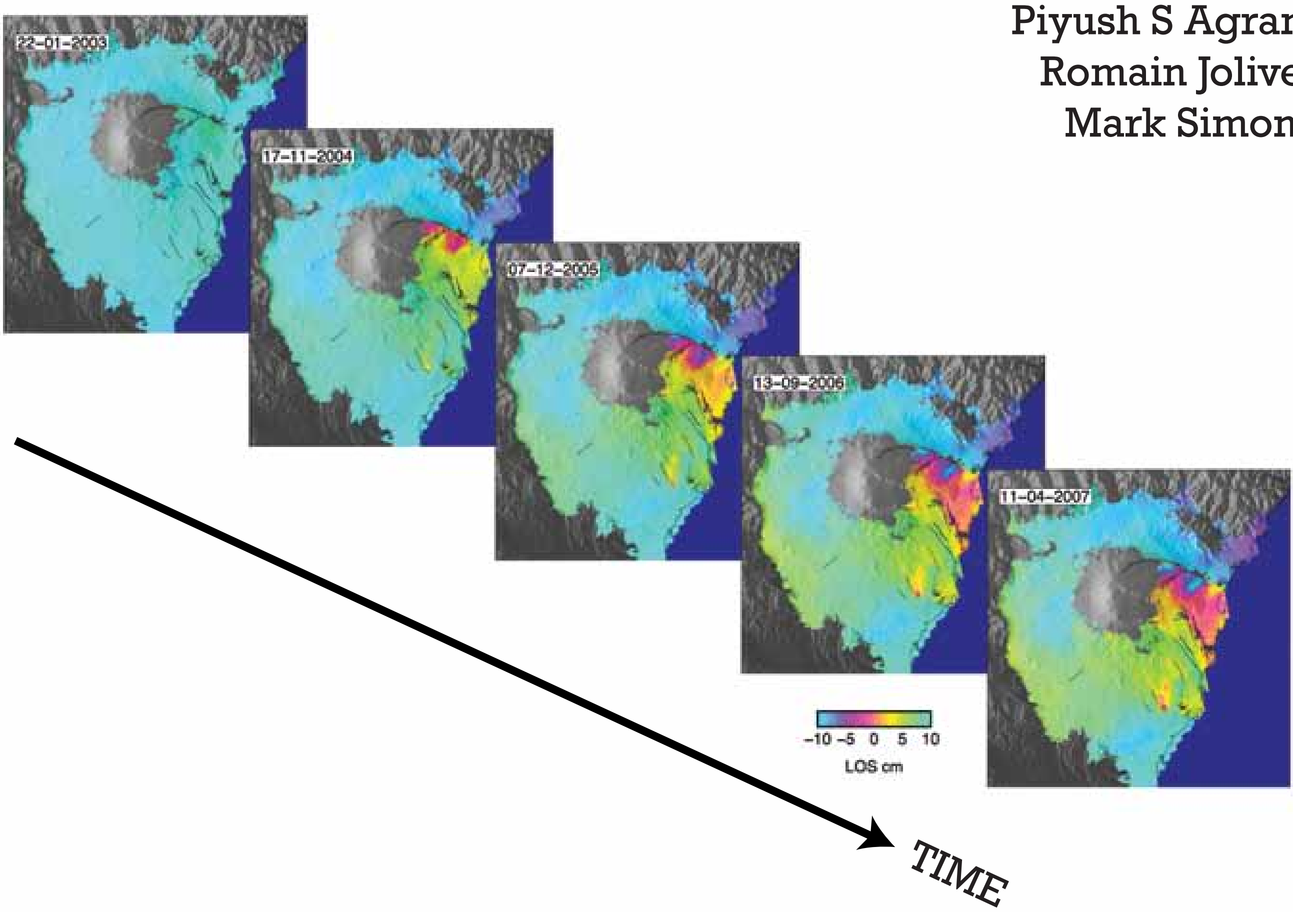
AGU EOS article on GIAnT

GIAnT

The Generic InSAR Analysis Toolbox

User Manual

Piyush S Agram
Romain Jolivet
Mark Simons



earthdef@caltech.edu

Contents

Contents **2**

0.1 About This Document . . . 7
0.2 Who Will Use This Documentation . . . 7
0.3 Citation . . . 7
0.4 Acknowledgements and Credits . . . 8
0.5 Request for feedback . . . 8

1 Introduction **11**

1.1 Overview . . . 11
1.2 Features . . . 12
1.3 Algorithms . . . 12
1.4 Programming philosophy . . . 13
1.5 Future work . . . 13

2 Installation **15**

2.1 Pre-requisites . . . 16
2.1.1 Python 2.6 or higher . . . 16
2.1.2 Numerical Python (NumPy) . . . 16
2.1.3 Scientific Python (SciPy) . . . 17
2.1.4 Cython . . . 17
2.1.5 Matplotlib . . . 18
2.1.6 h5py . . . 18
2.1.7 pygrib . . . 18
2.1.8 pywavelets . . . 19
2.1.9 LXML . . . 19
2.2 Optional . . . 20
2.2.1 ffmpeg or mencoder . . . 20

2.2.2 pyresample 20
2.2.3 HDFview 20
2.2.4 iPython 21
2.2.5 bpython 21
2.2.6 pykml 21
2.2.7 ImageMagick 22
2.2.8 xmlcopyeditor 22
2.3 Installation 22
2.3.1 Building extensions 23
2.3.2 Non-standard installations 24

3 Using GIAnT 25
3.1 Python 27
3.2 Working with HDF5 27
3.3 Matplotlib 28
3.4 GIAnT conventions 28

4 Preparing data for processing 29
4.1 Inputs 29
4.1.1 Using geocoded stacks [0] 31
4.2 Outputs 32
4.3 Directory structure 32
4.4 Preparing XML files 33
4.4.1 Writing the **data.xml** file 34
4.4.2 Writing the **sbas.xml** file 37
4.4.3 Writing the **mints.xml** file 39
4.4.4 Writing more XML files 41
4.5 Preparing the Interferogram Stack 41
4.6 Checklist 42

5 Removing orbital ramps and stratified tropospheric artifacts 45
5.1 Inputs 46
5.2 Outputs 46
5.3 Atmospheric corrections 48
5.3.1 Theory and methodology 48
5.3.2 Implementation and possible options 49
5.3.3 Empirical corrections 51
5.4 Orbital errors estimation 51
5.4.1 Network De-Ramping 51
5.4.2 GPS De-Ramping 52

Theory 52
Implementation 54
5.5 Checklist 55

6 Time-series: SBAS 57
6.1 Inputs 57
6.2 Outputs 58
6.3 The SBAS method 58
6.3.1 Inversion Strategy 58
6.3.2 Uncertainties estimation 59
6.3.3 Outputs 59
6.3.4 Checklist 60
6.4 The NSBAS method 60
6.4.1 Inversion strategy 60
6.4.2 Traditional stacking [1] as special case 62
6.4.3 Uncertainties estimation 62
6.4.4 Outputs 63
6.4.5 Checklist 63
6.5 The TimeFun method 64
6.5.1 Inversion strategy 64
6.5.2 Uncertainties estimation 65
6.5.3 Outputs 66
6.5.4 Checklist 66
6.6 Important Note 66

7 Time-series: MInTS 67
7.1 The MInTS Algorithm 67
7.2 Forward Wavelet Transform 68
7.3 Time-series inversion of the wavelet coefficients 69
7.3.1 Inversion strategy 69
7.3.2 Uncertainties estimation 71
7.4 Inverse Wavelet Transform 71
7.5 Note 72
7.6 Checklist 72

8 Visualization and Geocoding 73
8.1 Interactive visualization 73
8.2 Movies 74
8.3 KML 75
8.4 Geocoding 76

[1] Velocity estimates

Appendices **76**

A I/O Utilities **77**
A.1 Text files 77
A.2 Binary files 78
A.3 HDF5 Files 78
A.4 GMT netcdf files 79
A.5 XML Files 79
A.5.1 XML format 79
A.5.2 Creating XML file 79
A.5.3 Reading XML file 80
A.6 DEM RSC file for non ROI_PAC data 80
A.7 GPS Input Files 81
A.7.1 Option 1: Velocity type station list 81
A.7.2 Option 2: Station-wise time series 81
Option a: Classic Station List 81
Option b: SOPAC station list 81
Station File example 81

B Time-series Utilities **83**
B.1 Temporal characterization 83
B.1.1 Dictionary of functions 83
B.1.2 Putting functions together 85
B.1.3 SBAS Formulation 85
B.2 Network utilities 85

C Image utilities **87**
C.1 MInTS image routines 87
C.2 Stack routines 87

D Wavelets **89**
D.1 Convention 89
D.2 Routines 90
D.3 Other wavelets 91

E Solvers **93**
E.1 Least squares 93
E.2 Regularized L_2 norm 93
E.3 Iteratively reweighted least squares 94

Bibliography **95**

Preface

0.1 About This Document

This document describes GIAnT. It is organized following the steps most users will follow for their interferometric SAR time series processing. We begin with a description of how to install the software, with extensive details for Linux and OS-X users. We have not tested this software on a Windows platform. We then describe, step by step, the different techniques implemented in GIAnT.

0.2 Who Will Use This Documentation

This documentation is designed for both scientists who are content to use pre-packaged tools for analysing their synthetic aperture radar (SAR) interferogram stack and for experienced interferometric SAR (InSAR) time-series algorithm developers. The latter are likely to want to study the source code, compare the performance of numerous algorithms and even incorporate additional processing approaches into GIAnT. Users who want to modify the source will need to have familiarity with scripting, software installation, and programming, but are not necessarily professional programmers. We hope that any user-improvements will be shared with the developers so that all users can benefit from them.

0.3 Citation

We make this source code available at no cost in hopes that the software will enhance your research. Commercial use of any part of this software is not allowed without express permission from the authors.

A number of individuals have contributed significant time and effort towards the development of this software. Please recognize these individuals by citing the relevant peer-reviewed papers and making relevant acknowledgements in talks and publications.

[[[Include citations when ready.]]]

0.4 Acknowledgements and Credits

The authors of this software were supported by NASA solid earth and natural hazards program (NNX09AD25G), the Keck Institute for Space Studies (KISS) and the Caltech Tectonics Observatory (CTO). The Authors thank the Gordon and Betty Moore foundation for financial support of the CTO.

The authors thank all those people who contributed in helping to develop this program and those who volunteered to be guinea pigs, among them B. Riel, G. Peltzer, C. Lasserre and M.-P. Doin.

The data used to generate the front page graphics were provided by Marie-Pierre Doin, ISTerre, France. The data processing from the raw SAR data to the interferograms is described in Doin et al. [2011]. We produced the time series using GIAnT. The snapshots are created using the Generic Mapping Tool [Wessel and Smith, 1995].

0.5 Request for feedback

Your suggestions and corrections are essential to improve this software suite and the documentation. Please report any errors, inaccuracies or typos to the GIAnT development team at earthdef@gps.caltech.edu.

License

```
THE ACCOMPANYING PROGRAMS ARE PROVIDED UNDER THE TERMS OF THIS
    PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR
    DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE
    OF THIS AGREEMENT.

a. Generic InSAR Analysis Toolbox (GIAnT) - Copyright (c) 2012,
    California Institute of Technology
b. Python-based Atmospheric Phase Screen estimator (PyAPS) -
    Copyright (c) 2012, California Insitute of Technology
c. Variable Resolution Interferogram Resampler (VarRes) - Copyright
     (c) 2012, California Institute of Technology


1. Definitions

"Contribution" means:

a) in the case of the initial Contributor, the initial code and
    documentation distributed under this Agreement, and

b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from
    and are distributed by that particular Contributor. A
    Contribution 'originates' from a Contributor if it was added to
    the Program by such Contributor itself or anyone acting on such
    Contributor's behalf.

"Contributor" means any person or entity that distributes the
    Program.

"Program" means the Contributions distributed in accordance with
    this Agreement.
```

```
"Recipient" means anyone who receives the Program under this
    Agreement, including all Contributors.


2. Grant of Rights

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
    are met:

a) Program and derivative works may not be sold, nor may they be
    used in a commercial product or activity.

b) Derivative works of the Program, in both source and binary forms
    , must reproduce the above copyright notice, this list of
    conditions and the following disclaimer in the documentation and
    /or other materials provided with the distribution.

c) Contributors may not remove or alter any copyright notices
    contained within the Program.

d) Each Contributor must identify itself as the originator of its
    Contribution, if any, in a manner that reasonably allows
    subsequent Recipients to identify the originator of the
    Contribution.


3. Usage

The Program is the work of many developers, each of whom owns the
    copyright to the code they wrote. There is no central copyright
    authority you can license the code from. The proper way to use
    the Program is to examine it to understand how it works, and
    then write your own modules. Sorry, there is no free lunch here.

4. No Warranty

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
     "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT
    NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
    FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
```

Chapter 1

Introduction

1.1 Overview

The Generic InSAR Analysis Toolbox (GIAnT) is a suite of commonly used time-series interferometry algorithms in a common Python framework. Improvements in synthetic aperture radar (SAR) interferometry over the past couple of decades allow accurate measurement of ground surface deformation with unprecedented spatial coverage. Various packages are available to compute one or several interferograms, e.g, ROI_PAC [Rosen et al., 2004a], DORIS [Kampes et al., 2003], the new ISCE [Rosen et al., 2011], GMTSAR [Sandwell et al.] or variants, like NSBAS [Doin et al., 2011]. Observing large amplitude deformation signals, such as surface deformations due to earthquakes, is now a routine process. However, the detection of lower amplitude signals, such as interseismic accumulation, creeping faults, seasonal subsidence, etc., is more challenging. Time series analysis methods are used to enhance the signal-to-noise ratio and to study the temporal variability of surface deformation. The InSAR community uses a wide variety of methods and algorithms for interferometric time-series analysis. However, no common modular framework exists that allows researchers to quickly apply these wide range of algorithms on a single data set and compare their relative merits or limitations.

Development of GIAnT is primarily motivated by:

1. The need for standardization of data formats for ease in sharing time-series products.

2. Benchmarking of time-series InSAR algorithms.

3. Direct comparison of performance of various published algorithms.

4. A modular framework for efficient development of future time-series approaches.

GIAnT provides utility libraries of functions that are commonly used in various time-series approaches and includes demonstration scripts of various techniques. All the functions and scripts are documented in detail, allowing users to use these directly for data analysis or as building blocks for any time-series approach they would like to implement in the same framework.

1.2 Features

Some of key features of GIAnT include:

1. It is free.
2. It provides a Python-based framework.
3. Source code is distributed along with the package.
4. Use of memory mapped files facilitating analysis of large interferogram stacks.
5. A simple interface to weather model based atmospheric phase screen corrections [`http://earthdef.caltech.edu` Jolivet et al., 2011].
6. Direct calls to optimized linear algebra libraries like LAPACK, BLAS etc that can be optimized for speed using packages like ATLAS/ Intel MKL.
7. A set of interactive data visualization scripts.
8. Simple parallelization using Python's multiprocessing (shared memory) module for performance.

1.3 Algorithms

The time-series analysis routines in GIAnT can be broken down into two stages - spatial analysis and temporal analysis. GIAnT users can choose to analyze their data sets in radar (range, azimuth) coordinates directly or transform their data into wavelet domain before analysis. For the temporal analysis, the users can choose to work with the traditional piece-wise linear SBAS formulation [Berardino et al., 2002] or use a parameterized functional form of their choice [Hetland et al., 2011]. GIAnT modules have been designed in a manner that allows users to combine various types of spatial and temporal analysis algorithms as desired. The following time-series techniques have already been implemented and are provided with GIAnT for immediate use.

1. Small Baseline Subset (SBAS) [Berardino et al., 2002].
2. N-SBAS [Lopez-Quiroz et al., 2009, Doin et al., 2011, Jolivet et al., 2012].
3. Multiscale Insar Time-Series (MInTS) [Hetland et al., 2011].
4. Timefn (Temporal analysis part of MInTS applied directly in data domain).

1.4 Programming philosophy

GIAnT consists of two types of programs - modules and scripts. In GIAnT, modules are building blocks combined in a preferred order within a script. We use modules to implement processing steps that are common to various time-series techniques. We have consciously chosen to implement the various time-series algorithms as scripts for ease of understanding. We expect that most of the end users will not be expert programmers and the processing flow should be easy to comprehend by studying the scripts directly.

Even though we want to keep an eye on the evolution of the future versions of GIAnT, these tools are OpenSource, and therefore, we accept, with great pleasure, any contributions you could make to add algorithms, improve existing routines or develop new methods.

1.5 Future work

This first release of GIAnT focuses on building a framework and implementing some of the basic time-series InSAR algorithms. The package will constantly be maintained and updated. Bug reports and fixes can be reported on the website `http://earthdef.caltech.edu`. GIAnT in its current form does not address all aspects of time-series InSAR and the future versions are expected to address the following issues:

1. The processing chain is expected to start with wrapped data. Phase unwrapping will be included as a processing step. If starting with

unwrapped data, consistency checks will be incorporated to identify unwrapping errors.

2. Algorithms for estimation of DEM Error in the wrapped domain that are already available in the public domain [e.g, Hooper et al., 2007, Ducret et al., 2011] will be incorporated into GIAnT.

3. Tidal load corrections [DiCaprio and Simons, 2008] will be included as a package similar to the atmospheric corrections. This component will be important for the analysis of coastal regions and continental scale data sets.

4. Incorporation of Persistent Scatter algorithms [e.g, Hooper et al., 2007, Shanker and Zebker, 2007, Hooper, 2008].

5. Inclusion of more complete error covariance models and propagation of uncertainty estimates through various stage of time-series InSAR processing.

6. Optimization of various algorithms, including the option of analysis on distributed systems.

Chapter 2

Installation

Most of GIAnT consists of python programs that just need to be copied to a specified location. We include a simple Python setup script for components of GIAnT that are in Fortran or C. A large number of other Python modules need to be installed in order for GIAnT to work on your machine. Installing these pre-requisites are relatively easy.

Generic Linux
We recommend using a package manager like apt or yum to install the pre-requisites before installing GIAnT. We provide command lines to install the required Python libraries on a Linux Ubuntu machine.

OS-X
A convenient way to install all the pre-requisites is to use the package manager MacPorts (free)[1]. Installing MacPorts on OS-X machines is straightforward but requires Xcode[2] (free). We provide command lines to install the required Python libraries using MacPorts. Please be sure to run these commands as root. Another package manager called Fink is available[3] but the installation of all the libraries required by GIAnT has never been tested with Fink.

[1] `http://www.macports.org`
[2] `https://developer.apple.com/xcode/`
[3] `http://www.finkproject.org`

2.1 Pre-requisites

All the following pre-requisites may be installed from source. Although, we strongly advise the use of a package manager for beginners.

2.1.1 Python 2.6 or higher

GIAnT uses Python (`http://www.python.org`) and you will need a minimum of Python 2.6, for various pre-requisite packages to work seamlessly. If for some reason, you choose to build Python from source, please ensure that you use the same set of compilers for building any of the other packages for Python. Also ensure that you get the development package for Python for Linux.

On OS-X, all required libraries for GIAnT are available on MacPorts, for Python 2.6 or Python 2.7. The suggested commands are for Python 2.7 but can de adapted by changing "27" to "26" in the commands.

```
Ubuntu - 12.04:
>> apt-get install python2.7 python2.7-dev

OS-X:
>> port install python27
>> port select python python27
```

2.1.2 Numerical Python (NumPy)

GIAnT makes extensive use of NumPy (`http://numpy.scipy.org`) for representing datasets as arrays and for many array manipulation routines. We also use some FFT and linear algebra routines provided with NumPy. numpy.int, numpy.float and numpy.float32 are the most common data formats used at various stages of processing arrays and data.

```
Ubuntu - 12.04:
>> apt-get install python-numpy

OS-X:
>> port install py27-numpy
```

If you want to improve the performance of Numpy, we suggest using LAPACK, BLAS and ATLAS libraries. For more details on installing numpy from source using these libraries, see `http://docs.scipy.org/doc/numpy/`

user/install.html. On OS-X, a variant of Numpy that includes LAPACK, BLAS and the optimization ATLAS libraries is available on MacPorts. We suggest users to install the variant including compilation by gcc 4.5:

```
OS-X:
>> port install py27-numpy +atlas +gcc45
```

2.1.3 Scientific Python (SciPy)

SciPy (http://scipy.org) contains many functions for linear algebra operations, FFTs and optimization. SciPy also includes support for sparse matrices and provides solvers for various types of optimization problems.

```
Ubuntu - 12.04:
>> apt-get install python-scipy

OS-X:
>> port install py27-scipy
```

Vanilla distributions of SciPy obtained through utilities like yum and apt are typically not optimized. For best performance on large Linux computers, SciPy must be compiled with ATLAS / Intel MKL support. On Apple computers, the optimized SciPy distribution can be installed by typing:

```
OS-X:
>> port install py27-scipy +atlas +gcc45
```

2.1.4 Cython

Cython (http://www.cython.org) is a language that makes writing C extensions for the Python language as easy as Python itself. Cython is ideal for wrapping external C libraries and for writing fast C modules that speeds up the execution of Python code.

```
Ubuntu - 12.04:
>> apt-get install cython

OS-X:
>> port install py27-cython
```

2.1.5 Matplotlib

Matplotlib (`http://matplotlib.sourceforge.net`) is a python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. We use matplotlib for displaying output and for our interactive time-series viewers.

```
Ubuntu - 12.04:
>> apt-get install python-matplotlib

OS-X:
>> port install py27-matplotlib
```

2.1.6 h5py

h5py (`http://code.google.com/p/h5py`) provides a NumPy interface to Hierarchial Data Format 5 (HDF5) memory mapped files. We use h5py for storage and retrieval of named variables during various stages of processing. A big advantage of h5py is it allows us to access slices of large matrices directly from a file, without having to use up memory resources needed to read the entire matrices. The latest version of MATLAB also uses the HDF5 format and it is possible to directly read in .mat files into Python using scipy.io.loadmat.

```
Ubuntu - 12.04:
>> apt-get install python-h5py

OS-X:
>> port install py27-h5py
```

2.1.7 pygrib

GIAnT can directly interact with PyAPS modules to use weather model data for atmospheric phase screen corrections. pygrib (`http://code.google.com/p/pygrib`) provides the interface for directly reading in GRIB-format weather data files in Python. Successful installation of pygrib needs grib_api, openjpeg, jasper, libpng, zlib (including all development versions) which can all be obtained using standard repository management tools. Pygrib also needs the basemap or pyproj module for python to be installed.

```
Ubuntu - 12.04:
>> apt-get install zlib1g zlib1g-dev
```

```
>> apt-get install libpng12-0 libpng12-dev
>> apt-get install libjasper1 libjasper-dev
>> apt-get install libopenjpeg2 libopenjpeg-dev
>> apt-get install libgrib-api-1.9.9 libgrib-api-dev libgrib-api-tools
>> apt-get install python-mpltoolkits.basemap
>> apt-get install pyproj
>> easy_install pygrib
```

Unfortunately, pygrib is not directly available using a package manager on all Linux machines. You will have to follow instructions on the Google code page to install pygrib after installing all the required packages.

On OS-X computers, you can install pygrib using macports (all the dependencies will follow with that command):

```
OS-X:
>> port install py27-pygrib
```

2.1.8 pywavelets

The MInTS Hetland et al. [2011] time-series approach uses wavelets for spatial analysis. We provide our own Meyer wavelet library for analysis with the original MInTS approach. However, GIAnT also allows one to use other wavelets for spatial decomposition of unwrapped interferograms using the pywt (`http://github.com/nigma/pywt`) package.

```
Ubuntu - 12.04:
>> apt-get install python-pywt

OS-X:
>> port install py27-pywavelets
(or)
>> easy_install pywavelets
```

2.1.9 LXML

GIAnT uses XML files for setting up data and processing parameters. Specifically, we use the eTree module from lxml to construct input XML files and the objectify module from lxml to read in XML files. LXML (`http://lxml.de`) should be available as a standard repository on most linux distributions.

```
Ubuntu - 12.04:
>> apt-get install python-lxml

OS-X:
>> port install py27-lxml
```

2.2 Optional

We would also recommend installing the following packages before installing GIAnT.

2.2.1 ffmpeg or mencoder

We will need one of the two packages to use the matplotlib.animation submodule for making movies.

```
Ubuntu - 12.04:
>> apt-get install ffmpeg mencoder

OS-X:
>> port install ffmpeg
```

Mencoder is not available through MacPorts (maybe through Fink).

2.2.2 pyresample

Pyresample is a Python package that allows for easy geocoding of swath data (interferograms etc). We use pyresample to generate movies in the geocoded domain. Pyresample can be downloaded from `http://code.google.com/p/pyresample/`. If pyproj is already installed on your machine, you can install pyresample using the command

```
Ubuntu - 12.04 and OS-X:
>> easy_install pyresample
```

2.2.3 HDFview

HDFview is open source software for exploring the contents of an HDF file. The latest version can be downloaded from `http://www.hdfgroup.org/hdf-java-html/hdfview/index.html`.

```
Ubuntu - 12.04:
>> apt-get install hdfview
```

HDFview does not exist through MacPorts but can be easily installed following the instructions on the HDFview website.

2.2.4 iPython

Interactive Python (iPython) [Pérez and Granger, 2007] provides a rich toolkit for Python that allows users to work with the python interpreter in an environment similar to MATLAB or IDL.

```
Ubuntu - 12.04:
>> apt-get install ipython

OS-X:
>> port install py27-ipython
```

2.2.5 bpython

bpython `http://bpython-interpreter.org/` is a simple interface to the python interpreter. We recommend bpython when iPython cannot be used, for example when you are on a NFS partition.

```
Ubuntu - 12.04:
>> apt-get install bpython

OS-X:
>> port install py27-bpython
```

2.2.6 pykml

pykml (`http://packages.python.org/pykml/`) is a Python library for creating, parsing and manipulating KML files. GIAnT can output time-series products to KML files for immediate visualization in Google Earth.

```
Ubuntu - 12.04 and OS-X:
>> easy_install pykml
```

2.2.7 ImageMagick

This is typically a part of the standard Linux distributions. We use ImageMagick's **c**onvert utility to make parts of PNG files transparent for display usign KML/KMZ files.

```
Ubuntu - 12.04:
>> apt-get install imagemagick

OS-X:
>> port install imagemagick
```

2.2.8 xmlcopyeditor

xmlcopyeditor `http://xml-copy-editor.sourceforge.net/` is a simple editor for XML. The XML files used in GIAnT or ISCE can be easily modified using a text editor but xmlcopyeditor makes the task a little simpler. We recommend installing the package from source.

2.3 Installation

GIAnT has the following directory structure.

```
GIAnT (INSTALL directory)
|
|---- tsinsar   (Time-series library)
|
|---- pyaps     (Atmospheric corrections)
|
|---- SCR       (Time-series analysis scripts)
|
|---- geocode   (Geocoding library and scripts)
|
|---- solvers   (Solvers)
|
|---- setup.py  (Installation script)
|
|---- setup.cfg (Setup configure file)
```

Identify the directory in which you want to install **GIAnT** and make it your current working directory. Checkout the latest version of the code as

```
svn co http://earthdef.caltech.edu/svn/giant
```

Do not move the contents of the repository directory to another location as that may affect automatic updating of the repository using svn in the future. C and Fortran modules need to be built using C and Fortran compilers (see Section 2.3.1). The final step is to include the full path of the **giant/GIAnT** directory to the environment variable **PYTHONPATH**. This will allow python to import these modules whenever they are used in any script.

```
Using Bash:
>> export GIANT=/directory/where/you/did/copy/GIAnT
>> export PYTHONPATH=$GIANT:$PYTHONPATH

Using Csh:
>> setenv GIANT '/directory/where/you/did/copy/GIAnT'
>> setenv PYTHONPATH $GIANT:$PYTHONPATH
```

These commands should be included in your .bashrc or .cshrc files.

2.3.1 Building extensions

The setup script builds the **gsvd** module which contains our interface to the generalized SVD decomposition from LAPACK, similar to SciPy's interface to LAPACK and BLAS in this directory. The gsvd is used for L_2 norm regularized inversions in GIAnT.

The default settings uses the default C and fortran compilers to build extensions. The setup.cfg file can also be modified to force the machine to use a specific fortran compiler. If you have multiple Fortran and C compilers on your machine, you should specify the version compatible with your installation of python as shown below:

```
>>CC=Compiler python setup.py build_ext
--fcompiler=compiler-options
```

On OS-X, the default compiler will be clang. This will cause some problems if you use any regularized inversions. Therefore, on OS-X, if you linked Numpy and Scipy to Atlas, as mentioned previously, you want to compile gsvd using:

```
>> CC=gcc-mp-4.5 python setup.py build_ext
 --fcompiler=gnu95
```

The compiler options can also be included in the setup.cfg file before executing setup.py .

If your LAPACK/BLAS libraries were not built with gfortran, readup on the "–fcompiler" option for numpy distutils.

Alternate installation

Alternately, identify the directories in which the LAPACK, BLAS and ATLAS libraries are located. Compile using f2py in the gsvd directory.

```
>> f2py gensvd.pyf dggsvd.f -LPATH_TO_LIBS -llapacklib
     -lblaslib -latlaslib
```

```
On Ubuntu - 12.04:
>> f2py gensvd.pyf dggsvd.f -llapack -lblas -latlas
```

Test the compiled module using the provided test.py. Ensure that you are using the f2py corresponding to the numpy version you want to use.

2.3.2 Non-standard installations

If you happened to install any of the above pre-requisite libraries yourself and if they are not located in the Standard library directories for your OS, include the paths to the shared object files (libxxxxxxxxx.so) to the environment variable **LD_LIBRARY_PATH**. This set of tools has not been tested, or even installed, on Windows operated machines.

Chapter 3

Using GIAnT

GIAnT is distributed with implementations of SBAS [Berardino et al., 2002, Doin et al., 2011] and MInTS [Hetland et al., 2011] techniques. The prepackaged implementations are meant to work with outputs from ROI_PAC [Rosen et al., 2004b], ISCE [Rosen et al., 2011], DORIS [Kampes et al., 2003] or GMTSAR [Sandwell et al.]. In our description of the various processing steps, we used the term "stack" to denote a three-dimensional cube of data, the dimensions typically corresponding to range direction, azimuth direction and either number of interferograms or SAR acquisitions. Our usage of the term "stack" should not be interpreted as some form of velocity estimate. Figure 3.1 describes the work flow in the current implementation of various time-series InSAR algorithms implemented in GIAnT. However, the main strength of GIAnT lies in its modular implementation of the algorithms which allows users to implement their own version of the different time-series techniques and to extend GIAnT. As shown in Figure 3.1, there are multiple stages in the analysis:

1. **Stack preparation**
   Unwrapped interferograms are read from various locations on disk and are consolidated into a data cube. The data cube is stored along with other auxiliary information in a hierarchical data format (HDF5) file. Chapter 4 describes this step in detail.

2. **Stack preprocessing**
   Preprocessing of the stack including orbit deramping and topo-correlated atmospheric phase correction. The outputs are stored in a HDF5 file. Chapter 5 describes each of the preprocessing steps in detail.

3. **Time-series estimation**
   The time-series is estimated from the processed stack using a technique of choice. Currently, you can choose between various SBAS techniques (Chapter 6) and MInTS (chapter 7).

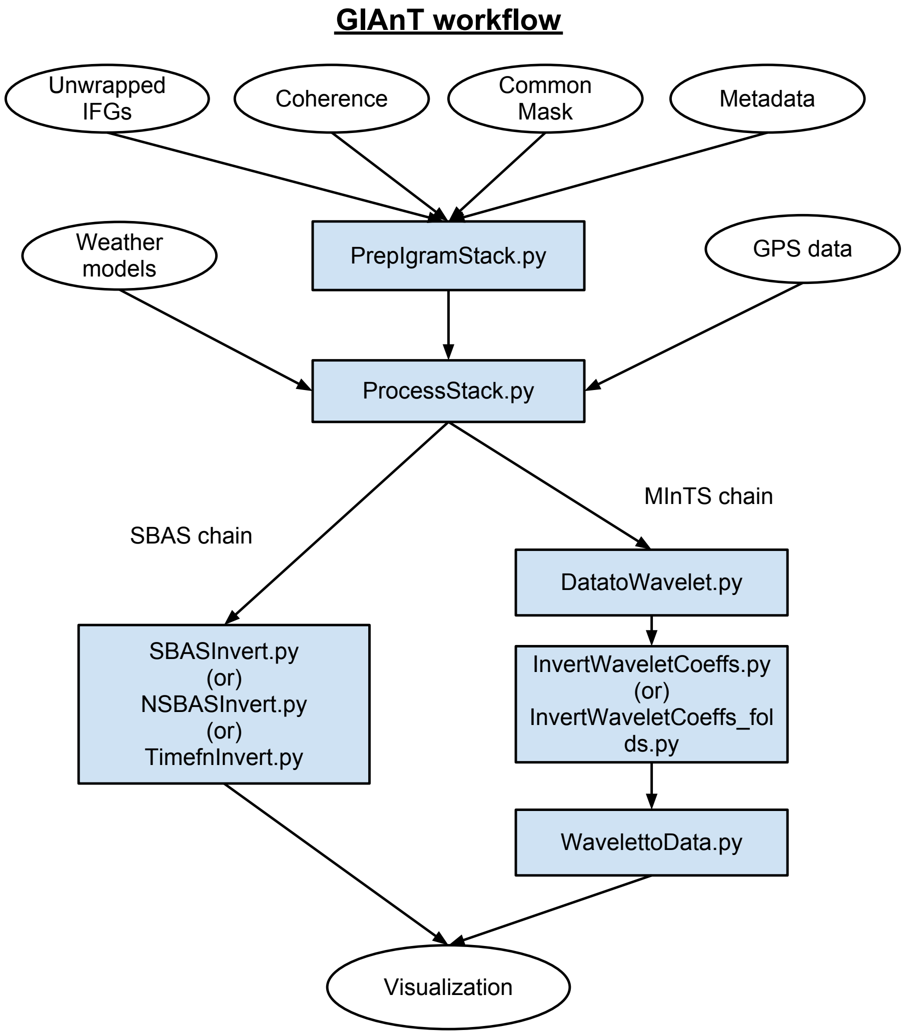


Figure 3.1: GIAnT workflow for using InSAR time-series analysis. The main strength of GIAnT is the modular implementation of these various stages. The modules can themselves be used to extend GIAnT and implement other time-series techniques.

Before processing any data, we strongly advise the users to familiarize themselves with various features of GIAnT, that will help them use the toolbox more effectively.

3.1 Python

For users who are familiar with MATLAB or IDL, the transition to Python should be fairly easy. Two resources that we recommend for users who are new to Python are

- The tentative Numpy turorial- `http://scipy.org/Tentative_NumPy_Tutorial`
- Numpy for MATLAB users `http://mathesaurus.sourceforge.net/matlab-numpy.html`

3.2 Working with HDF5

The Hierarchical data format (HDF) allows us to save large amounts of data in an organized and easy to access manner. GIAnT uses HDF5 files in the same way that users use .mat files in MATLAB or .sav files in IDL. In GIAnT, all HDF5 files and the datasets they contain are stored with a "help" attribute. We recommend that the users become familiar with the h5py [Collette, 2008] module interface in Python for reading and displaying data sets using matplotlib. A good introductory tutorial can be found at `http://h5py.alfven.org/docs-2.0/intro/quick.html`.

To determine the overview of the contents of a HDF5 file, use

```
####Overview description of file
$ h5dump -a help Filename.h5
```

To determine the contents of the HDF5 and their description

```
#####List contents
$ h5ls Filename.h5

#####List contents with description
$ h5ls -v Filename.h5 | grep Data
```

To dump an array from HDF5 to a binary file, use **h5dump**. This command can also be used to crop the array before writing it to a file. HDFview is another tool that we strongly recommend for browsing through the contents of a HDF file.

3.3 Matplotlib

GIAnT uses matplotlib [Hunter, 2007] for plotting. Some familiarity with matplotlib will allow users to write their own visualization codes for debugging and understanding the processing chain. Two useful tutorials can be found at

- http://matplotlib.sourceforge.net/users/pyplot_tutorial.html
- http://matplotlib.github.com/users/image_tutorial.html

3.4 GIAnT conventions

Various processing stages in GIAnT conform to few simple rules.

Units The unwrapped interferograms are converted to millimeters (mm) before any preprocessing. The atmospheric phase screen from PyAPS and GPS data are also converted to millimeters.

Master-slave Typically, users prefer to use the most recent amongst the pair of SAR acquisitions as the master scene when generating interferograms. GIAnT has been designed to be flexible and automatically figures out the correct connectivity matrix even if the order of master and slave acquisitions do not adhere to a consistent convention. We would still recommend the users to the latest SAR acquisition as the master scene in every pair consistently.

Ground-to-satellite The line-of-sight direction corresponds to the vector from ground position to the satellite. All the estimated time-series products are always estimated with respect to the ground, i.e, in the case of purely vertical deformation positive values represent uplift and negative values represent subsidence.

Angles The incidence angle provided by the user as a file or as a constant input is always measured with respect to the ground. This information is used for projecting estimated atmospheric path delays and GPS data along the line-of-sight.

Help attributes All the metadata values in input XML files must include a `<help>` field. All datasets in HDF5 files must be stored with a help attribute. This requirement is designed to aid users in understanding the relevance of input, output and intermediate products.

Chapter 4

Preparing data for processing

During this first step of the processing, all the necessary metadata and the unwrapped interferograms are sorted and organized into an HDF5 file and a few additional binary files, specially formatted for GIAnT. Inputs are data from outputs of ROI_PAC, DORIS, GMTSAR or ISCE. As we would like to provide the most generic toolbox, we tried to implement readers for the most common datasets, but slight changes might be needed to adapt GIAnT to your case. In this section we describe how to use two scripts located in `GIANT/SCR`:

- **prepxml_SBAS.py** and **prepxml_MInTS.py** create the XML files needed to specify processing options to GIAnT.
- **PrepIgramStack.py** reads the input data and creates files for GIAnT.

4.1 Inputs

To run these steps, the user needs to gather the following files:

- The unwrapped interferograms in radar coordinates (range, azimuth) and in radians produced by ROI_PAC, DORIS, GMTSAR or ISCE. If the users have geolocated unwrapped interferograms that they wish to analyze, they should take a look at Section 4.1.1 before proceeding.
- The corresponding coherence files.
- A list of the Interferograms, with the perpendicular baseline and the sensor code name.

- A Radar simulation file containing the pixels elevation, i.e, the DEM in radar coordinates. If you are using a processor other than ROI_PAC, you will have to generate a ROI_PAC style rsc file including the approximate latitude and longitude values for the four corners (see Appendix A). This information is needed for subsetting weather model data.

- An example.rsc file(ROI_PAC) (or) interferogram.out and master.res files (DORIS) (or) flat.xml and isce.log files (ISCE) or image.PRM and unwrap.grd files (GMTSAR).

- Optional: Two files containing each pixels latitude and longitude (binary files, same size as the interferograms, real single precision). These files can be GRD files in the case of GMTSAR.

- Optional: A mask file (binary file, same size as the interferogram, real single precision, 0 for False, 1 for True). This file can be a GRD file in case of GMTSAR.

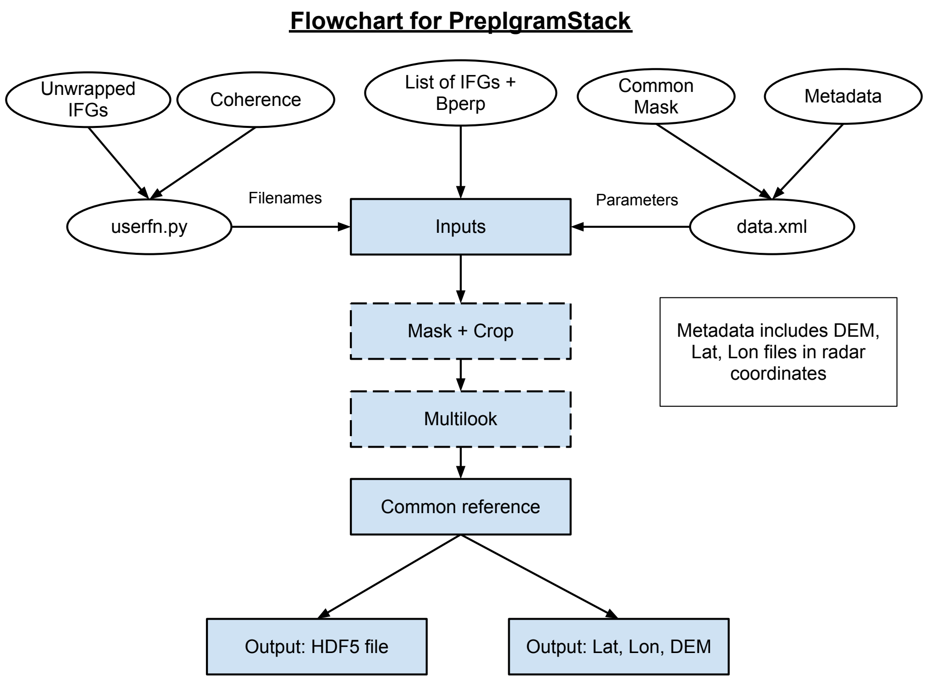


Figure 4.1: Work flow for preparing an interferogram stack.

We use the following setup for processing time-series with GIAnT. It is relatively simple to modify the stack [1] preparation script to use a different setup if desired. We pass the list of interferograms as an ASCII with four columns: the two first columns are the interferograms dates of acquisition coded over 6 or 8 digits (yymmdd or yyyymmdd), the third one is the perpendicular baseline while the last one is the sensor code name. Please make sure that the perpendicular baselines specified in here are consistent within the interferometric network, otherwise, the implemented Digital Elevation Model error estimation will produce incorrect results. A little example:

```
030104 031011 -384.6568006539 ENV
030104 031220 -412.2747552730 ENV
030104 041204 -346.6693732534 ENV
19920917 19920604 -158.938138675459 ERS
19921126 19920604 -194.461966044758 ERS
19921126 19920917 -35.4173555703392 ERS
....
```

You can optionally not provide the sensor name if you are using data from a single satellite. If you choose to mix data from multiple sensors, please ensure that the corresponding wavelength fields are populated in the **rdict** dictionary in **PrepIgramStack.py**. If all data is from a single sensor, the wavelength is automatically read in from the example rsc file. You will also need to provide a mapping from the dates to a filename on disk as a python function (Section 4.5).

We would also like to point out that most users will probably end up customizing the **P**repIgramStack.py script according to their own needs. Some users may prefer to read in the filenames and the associated wavelengths directly from an ASCII file. Implementing such changes should be trivial and we leave it up to the users to implement their favorite strategy.

4.1.1 Using geocoded stacks [1]

If the users have processed their interferograms individually and plan to combine them in geocoded domain, they should make the following changes to make GIAnT treat the data as if it was radar coded (range, azimuth).

1. Create latitude and longitude files of the same size as your geocoded unwrapped interferograms.

[1] Three-dimensional dataset and not a velocity estimate

2. Crop your DEM to the same size as your geocoded unwrapped interferograms.

3. Make sure that undefined data is set to NaN in **PrepIgramStack.py** where the interferograms are read into a HDF5 file.

4. You will also need to include entries for the four corners of the DEM in a ROI_PAC style RSC file. See Appendix **??**.

4.2 Outputs

A HDF5 file containing the following datasets is computed:

- `Jmat`: Matrix linking the interferograms to the acquisition dates, made of 0, 1 and -1.
- `bperp`: Vector of the perpendicular baseline values.
- `cmask`: Mask map of the pixels.
- `dates`: Dates of acquisition.
- `tims`: Time of acquisition in the specific ordinal reference.
- `usat`: Satellite sensor code name.
- `igram`: 3D matrix of size $n \times l \times w$, where n is the number of interferograms, l and w are the interferograms length (number of pixels along azimuth) and width (number of pixels along range).

Additionally, the binary files containing each pixels latitude, longitude and elevation are also output if the corresponding inputs are provided. Finally, PNG format previews of the interferograms are created.

4.3 Directory structure

The first step is to create a working directory. The sub-directories are automatically created as needed by the different scripts. The default file and directory names can be modified by the different scripts if needed. The typical structure of the working directory is:

```
|-Working directory (Xml + scripts + metadata)
|
|---Figs
|    |
|    |----- Igrams (Raw interferograms)
|    |
|    |----- Atmos  (Atmospheric corrections)
|    |
|    |----- Ramp   (Deramped)
|
|---Stack (All the h5 files)
|
|---Atmos (All the weather model data)
|
|---RESP (Wavelet Impulse Response, Only if you use MInTS)
```

4.4 Preparing XML files

The inputs to the SBAS and MInTS scripts are controlled using XML files. We currently use three XML files - one for data parameters (**data.xml**), one for the processing parameters of SBAS style chains (**sbas.xml**) and one for processing parameters of MInTS (**mints.xml**). An example of these files can be found in the `GIAnT/example` directory and in the Appendices.

These XML files are prepared using the **prepxml_SBAS.py** or the **prepxml_MInTS.py** scripts. These scripts can be modified to add additional processing parameters as desired. The parameter values in the XML files should be modified as per requirement and the processing strategy. In the following, we describe the possible options included. Currently, the scripts are designed to work with data in radar (range, azimuth) coordinates. The XML files produced by the **prepxml_XXXX.py** scripts can also be modified in a text editor.
To run these scripts, type:

```
>> python prepxml_SBAS.py
```

or

```
>> python prepxml_MInTS.py
```

Note that the structure of the XML files changes as we add more options to the processing chain and need more control parameters. The help fields in the XML files have been added to describe each of the options for the benefit of users.

4.4.1 Writing the data.xml file

Both **prepxml_XXXX.py** scripts start by writing the XML file corresponding to your dataset (data format, additional looks, crop region etc). In the scripts, these parameters are parsed through the routine `prepare_data_xml`. You should modify the arguments of this routine according to your needs. For the data to be read in successfully, a meaningful reference region has to be provided. If a reference region is not provided, average bias is corrected from each interferogram.

The syntax is:

```
prepare_data_xml(self, fname, proc='RPAC',looks=1,
cohth=0.0, mask='',xlim=None, ylim=None,
rxlim=None, rylim=None, latfile='',
lonfile='', hgtfile='', inc=23.0, h5dir='Stack',
atmosdir='Atmos', figsdir='Figs', respdir='RESP',
unwfmt='RMG', demfmt='RMG', corfmt='RMG',
chgendian=False, endianlist=['UNW','COR','HGT']),
masktype='f4')
```

The only non-optional argument is:

- **fname** - The name of the example rsc file from ROI_PAC. This file contains parameters like the width of the image, its length, the sensors wavelength.... An example can be found in the directory `GIAnT/example`.

. The optional arguments are:

Param	Type	Help	Default
proc	STR	Processor used for generating the interferograms. Can be RPAC, DORIS, ISCE or GMT.	RPAC
looks	INT	Number of additional looks. No looks is default.	1
cohth	FLOAT	Coherence threshold for SBAS pixel selection. This parameter allows to throw out pixel with a coherence value lower than **cohth**. No pixels are rejected by default.	0.0

mask	STR	File for common mask for pixels. This mask has to be the same size as your interferograms, and consist on a binary grid of 1 and 0 or NaN.	None
file_length	INT	Number of azimuth lines in the unwrapped files	From rsc file
width	INT	Number of range pixels in the unwrapped files	From rsc file
xmin	INT	First pixel of cropping limit in Range direction	None
xmax	INT	Last pixel of cropping limit in Range direction	None
ymin	INT	First line of cropping limit in Azimuth direction	None
ymax	INT	Last line of cropping limit in Azimuth direction	None
rxmin	INT	First pixel of reference region in Range after cropping	None
rxmax	INT	Last pixel of reference region in Range after cropping	None
rymin	INT	First line of referene region in Azimuth after cropping	None
rymax	INT	Last line of reference region in Azimuth after cropping	None
latfile	STR	Latitude file. The latitude file is a binary file, that has the same size as your interferograms, that specifies the latitude of each pixel. It is encoded has real, single precision. If this file is not provided, then, GIAnT uses a crude and simple linear transformation to calculate the geographic coordinates of the pixels. This is needed only by the PyAPS module. If you specify a latfile, you need to specify a lonfile.	None

lonfile	STR	Longitude file. The longitude file is a binary file, that has the same size as your interferograms, that specifies the latitude of each pixel. It is encoded has real, single precision. If this file is not provided, then, GIAnT uses a crude and simple linear transformation to calculate the geographic coordinates of the pixels. This is needed only by the PyAPS module. If you specify a lonfile, you need to specify a latfile.	None
hgtfile	STR	Altitude file. The hgt file is a RMG file (ROI_PAC style) or a binary file, that has the same size as your interferograms. It specifies the altitude, in meters, of each pixels.	None
inc	FLOAT (or) STR	Incidence angle. Can be a constant or a file of the same dimensions as lat/lon. Used only by PyAPS.	23.0^o.
h5dir	STR	Directory where all the HDF5 files will be stored.	`./Stack`
atmosdir	STR	Directory where all the atmospheric data will be stored.	`./Atmos`
figsdir	STR	Directory where all the figures will be stored.	`./Figs`
respdir	STR	Directory where the Impulse response for wavelet transform are stored (MInTS case).	`./RESP`
unwfmt	`RMG/FLT`	Specifies the format of the interferogram files. `RMG` is the standerd ROI_PAC output for unwrapped interferograms (real, single precision, amplitude and phase). `FLT` is a simple binary file (real, single precision).	`RMG`
demfmt	`RMG/FLT`	Specifies the format of the Digital Elevation Model file.	`RMG`
corfmt	`RMG/FLT`	Specifies the format of the correlation files.	`RMG`
chgendian	BOOL	Swaps bytes if true for the file types specified in the **endianlist**	`False`
endianlist	List of STR	Specifies the file type concerned by byte swapping if **chgendian** is `True`.	'UNW', 'COR', 'HGT'

masktype	STR	Type of data in the provided mask file (optional) Can be any `dtype` used in Numpy.	`'f4'`

4.4.2 Writing the sbas.xml file

If you intend to use any of the three SBAS style methods provided here, you will need to create a **sbas.xml** file by modifying the script **prepxml_SBAS.py** according to your needs. The informations about your processing strategy are parsed through the routine `prepare_sbas_xml`. The syntax is:

```
prepare_sbas_xml(self, netramp=False, atmos='', demerr=False,
nvalid=0, regu=False, masterdate='', filt=1.0, gpsramp=True,
stnlist='', stntype='', gpspath='', gpstype='', gpsvert=False,
gpspreproc=False, gpsmodel=False, unwcheck=False,
gpspad=3, gpsmin=5, tropomin=1, tropomax=None, tropolooks=8)
```

The optional arguments are:

Param	Type	Help	Default
netramp	BOOL	Network deramping: True/False. Interferograms are flattened by estimating and removing a best fit ramp. The best fit ramp parameters are re-estimated in a network sense. For more details, refer to section 5.4.1.	False
gpsramp	BOOL	GPS deramping: True/False. Interferograms are flattened using displacement informations from a GPS network.	False
atmos	STR	Atmospheric corrections: ECMWF/ERA/NARR/MERA/TROPO. The stratified component of the tropospheric is mapped from Global Atmospheric Models or estimated from the phase/elevation relationship (TROPO) and removed to the interferograms. For more details, refer to section 5.3.	None
demerr	BOOL	DEM Error estimation: True/False. Estimation of the Digital Elevation Model error during the Time Series analysis.	False

nvalid	INT	Minimum number of interferograms where a pixel is coherent. The pixel will be included in the Time Series analysis only if its coherence is higher than **cohth** in more than **valid** interferograms.	0
regu	BOOL	Regularization of time functions: True/-False. Activate the automatic determination of the regularization parameter in the Time Function inversion. For more details, refer to chapter 6.5.1.	False
masterdate	STR	Time to be used as reference (yyyymmdd). Default is the first date of the time series if not specified.	None.
stnlist	STR	Path to the GPS station list.	None
stntype	BOOL (or) STR	Code name for the type of station list. Can be True/ False/ velocity.	None
gpspath	STR	Directory where GPS data are stored.	None
gpstype	STR	Type of GPS file that can be used (sopac or velocity).	None
gpsvert	BOOL	Use the verticals of GPS data.	False
gpspreproc	BOOL	Preprocess GPS time series with splines.	False
gpsmodel	BOOL	Use the modeled GPS time series as input.	False
gpspad	INT	Half-width of the window around a GPS station.	3
gpsmin	INT	Minimum number of GPS stations per scene to proceed to the GPS de-ramping process. The process stops if less than `gpsmin` stations are found for one scene.	5
tropomin	INT	Minimum scale for empirical estimation of topography correlated atmosphere.	1
tropomax	INT	Minimum scale for empirical estimation of topography correlated atmosphere. If not defined, determined from dimensions.	None
tropolooks	INT	Number of looks to be applied to interferogram before empirical estimation of topography correlated atmosphere.	8

4.4.3 Writing the mints.xml file

If you intend to use the MInTS method, you will need to create a **mints.xml** file by modifying the script **prepxml_MInTS.py** according to your needs. The informations about your processing strategy are parsed through the routine `prepare_mints_xml.py`. The syntax is:

```
prepare_mints_xml(self, netramp=False, atmos='', demerr=False,
minscale=2, regu=True, masterdate='', minpad=0.1, shape=True,
smooth=3, skip=2, wvlt='meyer', gpsramp=True, stnlist='',
stntype='', gpspath='', gpstype='', gpsvert=False, gpspreproc=False,
gpsmodel=False, uwcheck=False, kfolds=1, lamrange=[-5,5,50],
tropomin=1, tropomax=None, tropolooks=8)
```

The optional arguments are:

Param	Type	Help	Default
netramp	BOOL	Network deramping: True/False. Interferograms are flattened by estimating and removing a best fit ramp. The best fit ramp parameters are re-estimated in a network sense. For more details, refer to section 5.4.1.	False
gpsramp	BOOL	GPS deramping: True/False. Interferograms are flattened using displacement informations from a GPS network.	False
atmos	STR	Atmospheric corrections: ECMWF/ERA/NARR/MERA. The stratified component of the tropospheric is mapped from Global Atmospheric Models and removed to the interferograms. For more details, refer to chapter 5.3.	None
demerr	BOOL	DEM Error estimation: True/False. Estimation of the Digital Elevation Model error during the Time Series analysis.	False
minscale	INT	Number of smallest scales (or) highest frequency components to be ignored during reconstruction.	1
maxscale	INT	Number of largest scales (or) smallest frequency components to be ignored during reconstruction.	0

regu	BOOL	Regularization of time functions: True/-False. Activate the automatic determination of the regularization parameter in the Time Function inversion. For more details, refer to chapter 6.5.1.	False
masterdate	STR	Time to be used as reference (yyyymmdd). Default is the first date of the time series.	None
minpad	FLOAT	Minimum amount of mirroring to be applied to images before converting to Dyadic length for wavelet transforms. Expressed as fraction of the width of the image.	0.1
shape	BOOL	Shape smoothing to be applied to the regularization matrix. See Hetland et al. [2011] for details.	False
smooth	INT	Spatial smoothing of the regularization parameter in k-fold cross validation.	3
skip	INT	Number of pixels to skip during estimation of the penalty parameters in k-fold cross validation. This reduces the execution time.	2
wvlt	STR	Name of the wavelet used for MInTS. Can be meyer or any valid string from pywt.	meyer
stnlist	STR	Path to the GPS station list.	None
stntype	BOOL (or) STR	Code name for the type of station list. Can be True/ False/ velocity.	None
gpspath	STR	Directory where GPS data are stored.	None
gpstype	STR	Type of GPS file that can be used (sopac or velocity).	None
gpsvert	BOOL	Use the verticals of GPS data.	False
gpspreproc	BOOL	Preprocess GPS time series with splines.	False
gpsmodel	BOOL	Use the modeled GPS time series as input.	False
gpspad	INT	Half-width of the window around a GPS station.	3
gpsmin	INT	Minimum number of GPS stations per scene to proceed to the GPS de-ramping process. The process stops if less than `gpsmin` stations are found for one scene.	5
kfolds	INT	Number of folds for k-fold cross validation in the MInTS inversions.	8

lamrange	LIST	Range of penalty parameter values to search across in logspace. First and second elements represent the min and max values in log space and the last values represents the number of steps.	[-5, 5, 50]
tropomin	INT	Minimum scale for empirical estimation of topography correlated atmosphere.	1
tropomax	INT	Minimum scale for empirical estimation of topography correlated atmosphere. If not defined, determined from dimensions.	None
tropolooks	INT	Number of looks to be applied to interferogram before empirical estimation of topography correlated atmosphere.	8

4.4.4 Writing more XML files

Another routine called `prepare_gen_xml` exists and can be easily modified to produce any XML file. Any other time series analysis method, radically different from the SBAS methods or the MInTS method provided here should include its own XML files and its own XML writer. Users are strongly encouraged to implement their own techniques and the associated XML file structure.

4.5 Preparing the Interferogram Stack

PrepIgramStack.py script is used to prepare the stack[2] of raw interferograms. We strongly encourage the user to copy this script in the working directory and to modify it according to the demands of the dataset. This script first reads the parameters in the **data.xml** file and the interferogram list in the **ifg.list** file. Then the script uses a user-defined function to map the interferogram dates to a physical file on disk. The function must be defined in a standalone python file called **userfn.py** (default). The users can start with the template provided in the example directory.

PrepIgram.py uses the user-defined function to read the unwrapped interferograms and the coherence files, crops them to the desired region, excludes the pixels with a coherence lower than **cohth**, removes the mean from the reference region and stores them in the output HDF5 file. It

[2]Three dimensional data set and not velocity estimates

finally formats the latitude, longitude, elevation and incidence (optional) files if provided.

You can run **PrepIgramStack.py** as follows:

```
>> python PrepIgramStack.py -h -i INAME -o ONAME -x DXML
                        -f FIGS -u USERPY
```

Like all the scripts included in GIAnT, using the `-h` flag as input provides a detailed description and input options for the script. The input options for **PrepIgramStack.py** are:

- **-h**: Ask for help.
- **-i INAME**: INAME Input file name containing interferograms acquisition dates, perpendicular baseline and sensor code name. Default is `ifg.list`.
- **-o ONAME**: ONAME Output HDF5 file name. Default is `RAW-STACK.h5`.
- **-x DXML**: DXML data.xml filename. Default is `data.xml`.
- **-f FIGS**: FIGS Directory in the general figure directory where interferograms previews are stored. Default is `Igrams`.
- **-u USERPY**: USERPY Python script with the user defined function `makefnames`.

PrepIgramStack.py currently supports RMG or plain binary files which are either in short, integer, float32 or float64 format. Any other format would require some additional changes in the **PrepIgramStack.py** script. The input formats for the files are read in from the **data.xml** file. If users develop readers for data in other formats, we strongly encourage them to share these with the community.

4.6 Checklist

Here is a summary of the actions and commands to prepare your dataset:

1. Create a working directory.
2. Gather the necessary files: interferograms, coherence files, radar simulation (DEM in radar coordinates) file, example.rsc/interferogram.out files, interferogram list, latitude and longitude files (optional), mask file (optional), incidence angle file (optional).

3. Copy to the working directory and modify the **prepxml_SBAS.py** or **prepxml_MInTS.py** files.

4. Run: `python prepxml_SBAS.py` or `python prepxml_MInTS.py`.

5. Copy **PrepIgramStack.py** to the working directory and modify it if needed.

6. Copy **userfname.py** to the working directory and modify it.

7. Run: `python PrepIgramStack.py [Options]`

8. Check that you have a new HDF5 file and have a look at the PNG previews just created.

9. If you provided lat, lon and incidence angle files as inputs make sure that equivalent binary files are created in the **h**5dir directory.

Chapter 5

Removing orbital ramps and stratified tropospheric artifacts

Once the data are has been read into a HDF5 file, the user may proceed to the stack pre-processing by applying atmospheric corrections and the estimation of residual orbit errors. These corrections are performed by the **ProcessStack.py** script. No major change is required in this script, unless the users wants to implement their own correction strategy. Again, we ask users to share their extensions of this script with the community.

ProcessStack.py uses the parameters provided in the **data.xml** and **sbas.xml** (default) or **mints.xml** files. It processes the data stored into the previously created HDF5 file (default: Stack/RAW-STACK.h5) and the latitude, longitude, elevation and incidence (optional) files. To execute, type:

```
>> python ProcessStack.py -h -i INAME -o ONAME -x DXML -p PXM
```

The command line options are:

- **-h**: Ask for help.
- **-i INAME**: INAME Input HDF5 file. Default is `RAW-STACK.h5`.
- **-o ONAME**: ONAME Output HDF5 file. Default is `PROC-STACK.h5`.
- **-x DXML**: DXML Data XML file. Default is `data.xml`.
- **-p PXML**: PXML SBAS/MInTS XML file. Default is `sbas.xml`.

5.1 Inputs

To run this script, the user needs to make sure the following files are available:

- The output HDF5 file from the **PrepIgramStack.py** script.
- The **sbas.xml** or the **mints.xml** files.
- The latitude, longitude, elevation and incidence (optional) files produced by the **PrepIgramStack.py** script.
- The password section of model.cfg in the pyaps directory to be filled out, if it is desired to used weather models for correcting stratified tropospheric delay.
- The GPS data files, as described below.

5.2 Outputs

The output is, by default, stored in the HDF5 file `Stack/PROC-STACK.h5`. The file name may be modified using the command line `-o` flag. This file contains the following datasets:

- `Jmat`: Matrix linking the interferograms to the acquisition dates, made of 0, 1 and -1.
- `bperp`: Vector of the perpendicular baseline values.
- `cmask`: Mask map of the pixels.
- `dates`: Dates of acquisition.
- `tims`: Time of acquisition in the specific ordinal reference.
- `figram/igram`: 3D matrix containing the corrected interferograms. Its size is $n \times l \times w$, where n is the number of interferograms, l and w are the interferograms length (number of pixels along azimuth) and width (number of pixels along range).
- `ramp`: Array of ramp parameters. Its size is $n \times p$ where n is the number of interferograms and p the number of ramp parameters per interferogram.

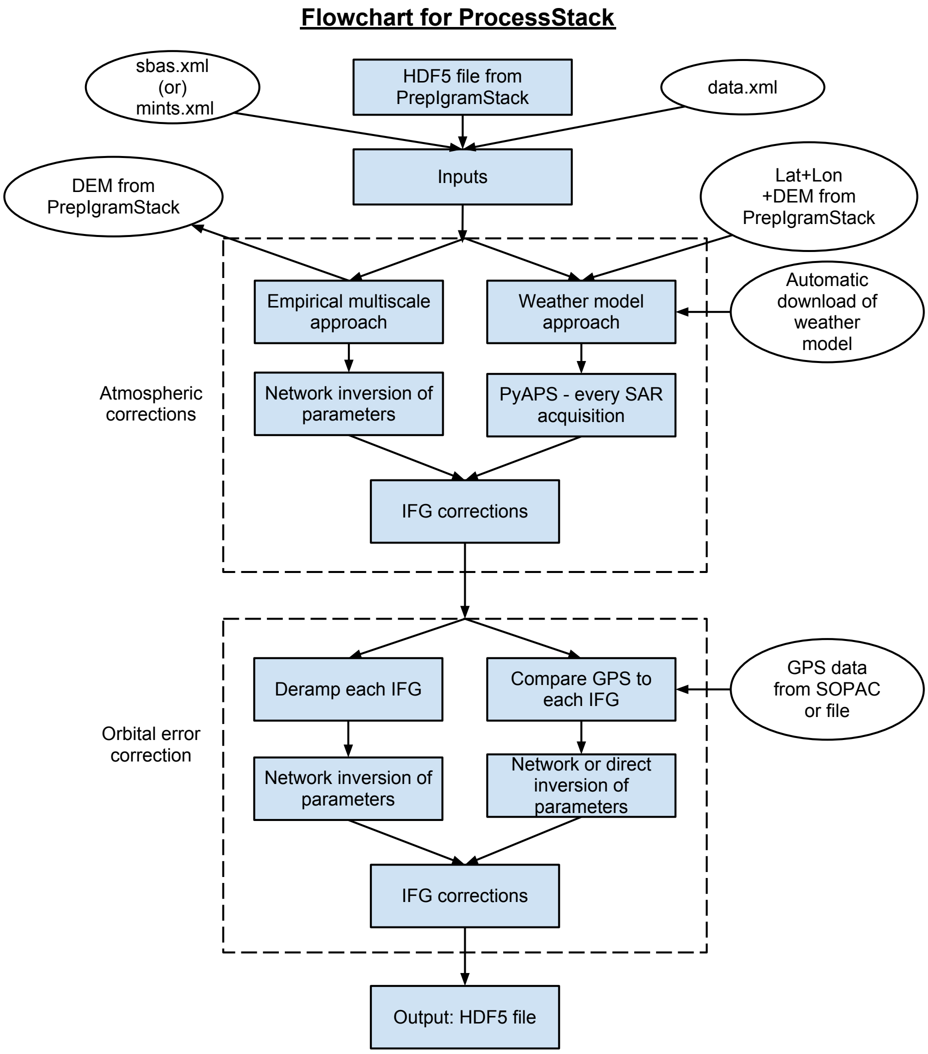


Figure 5.1: Work flow for processing an interferogram stack.

- `igram_aps`: Synthetic delay maps of each interferograms from the selected Global Atmospheric Model (Optional).

- `sar_aps`: Synthetic delay maps of each SAR acquisitions from the selected Global Atmospheric Model (Optional).

5.3 Atmospheric corrections

We use the PyAPS [Jolivet et al., 2011, Jolivet and Agram, 2012] module for implementing weather model based interferometric phase delay corrections. The python module is well documented, maintained and can be freely downloaded [1]. To use it in GIAnT, there is no need to download it as it is part of the package. PyAPS currently includes support for ECMWF's ERA-Interim, NOAA's NARR and NASA's MERRA weather models. The outputs from our processing modules include phase screen simulations for individual SAR scenes as well as phase corrections for each interferogram. PNG previews of the atmospheric corrections are saved (by default in the directory `Figs/Atmos`).

5.3.1 Theory and methodology

Following Doin et al. [2009] and Jolivet et al. [2011], we produce tropospheric delay maps from atmospheric data provided by Global Atmospheric Models. This method aims to correct differential atmospheric delay correlated with the topography in interferometric phase measurements. Global Atmospheric Models (hereafter GAMs), such as ERA-Interim (European Center for Medium-Range Weather Forecast), MERRA (Modern-Era Retrospective Analysis, Goddard Space Flight Center, NASA) or regional models such as NARR (North American Regional Reanalysis, National Oceanographic and Atmospheric Administration) provide estimates of the air temperature, the atmospheric pressure and the humidity as a function of elevation on a coarse resolution latitude/longitude grid. In PyAPS, we use this 3D distribution of atmospheric variables to determine the atmospheric phase delay on each pixel of each interferogram.

For a given GAM dataset, we select grid points overlapping with the spatial coverage of the SAR scene. Atmospheric variables are provided at precise pressure levels. We vertically interpolate these values to a regular grid between the surface and a reference altitude, z_{ref}, above which the delay is assumed to be nearly unchanged with time (~ 30000 m). We then compute the delay function on each of the selected grid points of the GAM as a function of height. The LOS single path delay $\delta \mathrm{L}^s_{LOS}(z)$ at an elevation

[1] `http://pyaps.googlecode.com`

z is given by [Doin et al., 2009, Jolivet et al., 2011]:

$$\delta \mathrm{L}^s_{LOS}(z) = \frac{10^{-6}}{cos(\theta)} \left\{ \frac{k_1 R_d}{g_m}(P(z) - P(z_{ref})) + \int_z^{z_{ref}} \left(\left(k_2 - \frac{R_d}{R_v} k_1 \right) \frac{e}{T} + k_3 \frac{e}{T^2} \right) dz \right\}, \tag{5.1}$$

where θ is the local incidence angle, $R_d = 287.05$ J.kg^{-1}.K^{-1} and $R_v = 461.495$ J.kg^{-1}.K^{-1} are respectively the dry air and water vapor specific gas constants, g_m is a weighted average of the gravity acceleration between z and z_{ref}, P is the dry air partial pressure in Pa, e is the water vapor partial pressure in Pa, and T is the temperature in K. The constants are $k_1 = 0.776$ K.Pa^{-1}, $k_2 = 0.716$ K.Pa^{-1} and $k_3 = 3.75 \cdot 10^3$ K^2.Pa^{-1}.

We compute the absolute atmospheric delay at each SAR acquisition date. For a pixel a_i at an elevation z at acquisition date i, we select the 4 surrounding grid points, 1, 2, 3 and 4. At each of these four grid points, we compute the delays at elevation z: $d^i_1(z)$, $d^i_2(z)$, $d^i_3(z)$ and $d^i_4(z)$. The resulting delay at the pixel a_i is the bilinear interpolation of $d^i_1(z)$, $d^i_2(z)$, $d^i_3(z)$ and $d^i_4(z)$. As the latitude/longitude grid of the NARR is based on a Lambert Conic sampling, we add a spatial linear resampling of the delay functions to match with a regular grid.

Finally, we combine the absolute delay maps corresponding to the SAR scenes to produce the differential delay maps used to correct the interferograms. Details and validation of our approach are available in Doin et al. [2009] and Jolivet et al. [2012].

5.3.2 Implementation and possible options

The script **ProcessStack.py** automatically downloads the atmospheric data into the directory `Atmos` (by default) before estimation the corrections to be applied to the interferograms. If the data has already been downloaded, it will not download it again. Each weather model has a different file naming convention, thus allowing users to determine the applicability and the effectiveness of different weather models on their interferogram stack[2]. PyAPS does not interpolate the weather model data in time and uses model information from the epoch closest to the SAR acquisition times. PyAPS can use three different GAMS (including automatic download) and the preferred model needs to be specified in the **sbas/mints.xml** file:

[2]Three dimensional dataset and not velocity estimates

- **ECMWF**: ERA-Interim Re-Analysis products are downloaded from the ECMWF repository in Europe[3] . We use the variables Temperature, Geopotential Height and Relative Humidity (default) at each Pressure Levels. If you are working on a machine with a non-US IP address, you should use this option. You need to register on the ECMWF website and provide your password in the file `GIAnT/pyaps/model.cfg` (a template is provided). To get your personalized access key from ECMWF, by read and agree to this license `http://data-portal.ecmwf.int/data/d/license/interim_full/`. Once you fill out the form, you will be directed to a page with a link to Perl/Python scripts on top. You will need to copy the registered email address and the associated key from the example scripts into the ECMWF fields in model.cfg. PyAPS only downloads the fields of interest - Humidity and Temperature as a function of pressure level. Each file (global for single time eopch) is about 28MB in size.

- **ERA**: ERA-Interim Re-Analysis products are available from the repository hosted by the University Corporation for Atmospheric Research (UCAR), Boulder, CO, USA[4]. We use the variables Temperature, Geopotential Height and Relative Humidity (default) at each Pressure Levels. If you are located in the US, you can download the ERA-Interim data faster from the UCAR archives. You need to register on the ERA Interim page at the UCAR website and provide your password in the file `GIAnT/pyaps/model.cfg` (a template is provided). You will need to register for access to the ERA Interim full resolution data set here `http://rda.ucar.edu/datasets/ds627.0/`. Once you have an active registered login, scroll down to the bottom of the same webpage and click on data access. You will need to agree to the terms and conditions before proceeding. You might have to wait for a email confirming your access to the data set. This dataset can only be accessed from within the United States. Each file is about 89MB in size. Our scripts do not attempt to subset this dataset.

- **NARR**: NARR Re-Analysis product is downloaded from the repository hosted by the National Oceanic and Atmospheric Administration (NOAA), USA. We use the variables Temperature, Geopotential Height and Relative Humidity (default) at each Pressure Levels. NARR covers North America at a resolution of about 30 km and has a temporal resolution of 3 hours. The original data is distributed on

[3]`http://data-portal.ecmwf.int/`

[4]`http://rda.ucar.edu/`

a non-uniform grid. PyAPS computes delay functions on the original grid and reinterpolates these functions onto a regular lat-lon grid before estimating corrections. No special login is needed to access this dataset. Each file is about 56MB in size.

- **MERRA**: MERRA Re-Analysis product is downloaded from the repository hosted by the NASA Goddard Space Flight Center, USA[5]. We use the variables Temperature, Pressure level height and Relative humidity. No special login is needed to access this dataset.

Users are strongly encouraged to report on the effectiveness of GAMs, send us feedback on PyAPS or implement their own modules in PyAPS to share it with the community. More details can be found in the PyAPS manual available online. In future versions of GIAnT, we plan to provide direct access to OSCAR's atmospheric phase delay maps (Jet Propulsion Laboratory, NASA), produced by merging GAMs and MODIS (Moderate Resolution Imaging Spectroradiometer, NASA) measurements and third party GPS-based tropospheric corrections.

5.3.3 Empirical corrections

If users prefer assuming a simple linear relationship with topography, the stratified atmospheric phase contributions can be estimated from the data itself. We optionally provide an implementation of a multi-scale approach Lin et al. [2010b]. The atmos parameter in the XML file needs to be set to **TROPO**.

5.4 Orbital errors estimation

Users are strongly encouraged to flatten the interferograms and correct for orbital errors prior to any time series analysis. We provide two methods for estimating orbital effects on interferograms, a network de-ramping and a GPS-based estimation method. If atmospheric corrections are activated, orbital errors will be estimated on corrected interferograms.

5.4.1 Network De-Ramping

This option is activated by setting the `netramp` parameter to `True` in the **sbas.xml** or **mints.xml** files. The process has been described by multiple

[5] `http://goldsmr3.sci.gsfc.nasa.gov`

studies, including Biggs et al. [2007], Cavalié et al. [2008], Lin et al. [2010a], Jolivet et al. [2012].

First, orbital errors are estimated independently on each interferogram using a least square scheme. By default, for the interferogram composed of two SAR acquisitions with indices i and j, we minimize the l_2 norm, $||d_{ij}(x,y) - R_{ij}(x,y)||_2$, where $d_{ij}(x,y)$ is the value of the pixel at range x and azimuth y, and

$$R_{ij}(x,y) = e_{ij} \cdot xy + a_{ij} \cdot x + b_{ij} \cdot y + c_{ij}, \tag{5.2}$$

where a_{ij}, b_{ij}, c_{ij} and e_{ij} are referred to as orbital parameters for the interferogram ij. The orbital term equation can be modified by changing the inputs of the `estramp` function in the **ProcessStack.py** script, so that,

$$\text{if poly} = 1, \qquad R_{ij}(x,y) = c_{i,j}, \tag{5.3}$$
$$\text{if poly} = 3, \qquad R_{ij}(x,y) = a_{ij} \cdot x + b_{ij} \cdot y + c_{ij}, \text{(default)} \tag{5.4}$$
$$\text{if poly} = 4, \qquad R_{ij}(x,y) = e_{ij} \cdot xy + a_{ij} \cdot x + b_{ij} \cdot y + c_{ij}. \tag{5.5}$$

Then, to ensure consistency in the interferometric network, we re-estimate the orbital parameters in a network sense. We estimate the orbital parameters for each SAR acquisition i, by inverting the linear systems given by,

$$a_{ij} = a_i - a_j, \tag{5.6}$$
$$b_{ij} = b_i - b_j, \tag{5.7}$$
$$c_{ij} = c_i - c_j, \tag{5.8}$$
$$d_{ij} = d_i - d_j. \tag{5.9}$$

Finally, we combine the re-estimated orbital parameters to produce orbital correction maps consistent with interferometric network and correct each interferogram. Other ramp functions can be easily incorporated in the `GIAnT/tsinsar/stackutils.py` file.

5.4.2 GPS De-Ramping

This option is activated by setting the `gpsramp` to `True` in the **sbas.xml** or **mints.xml** files. GPS velocities can be provided using a single ASCII file or in the SOPAC format. An example is available in the Appendices.

Theory

The GPS-based de-ramping technique starts by choosing the GPS stations overlapping with the SAR scene out of a station list provided by the user. The selected GPS stations latitude and longitude coordinates are then projected into the Radar geometry (Range, Azimuth) using the latitude and longitude files, if provided, or the SAR scene 4 corners (provided in the example.rsc/interferogram.out file). The GPS displacements are projected onto the Line-Of-Sight direction using the incidence angle and the heading and compared to the surrounding pixels (default is a 3 pixels wide window). We estimate the orbital parameters by minimizing $||d_{ij}(x, y) - R_{ij}(x, y) - D_{ij}^{GPS}(x, y)||_2$, where, $D_{ij}^{GPS}(x, y)$ is the GPS displacement projected into the LOS at a range x and azimuth y, $d_{ij}(x, y)$ is the phase value averaged over a 3 pixels wide (default) window centered on x and y. Only those GPS stations with colocated coherent InSAR observations are used to estimate the ramps. The orbital function is specified by the option `poly` (default, `poly`= 3) in the function `estramp_gps`.

Two options are available. If the `netramp` option is set to `True` in the Xml file, we estimate each SAR scene orbital parameter at once, ensuring consistency of the orbital parameters in the network sense. Orbital errors are given by,

$$\text{if poly=1,} \quad R_{ij}(x, y) = c_i - c_j, \tag{5.10}$$

$$\text{if poly=3,} \quad R_{ij}(x, y) = (a_i - a_j) \cdot x + (b_i - b_j) \cdot y + c_i - c_j, \text{(default)} \tag{5.11}$$

$$\text{if poly=4,} \quad R_{ij}(x, y) = (e_i - e_j) \cdot xy + (a_i - a_j) \cdot x + (b_i - b_j) \cdot y + c_i - c_j, \tag{5.12}$$

where, a_i, b_i, c_i and e_i are referred to as the orbital parameters for the scene i. Orbital parameters are then combined to produce orbital error maps and correct each interferogram.

If the `netramp` option is set to `False`, orbital errors are estimated independently on each interferogram. In such case, orbital errors are given by:

$$\text{if poly=3,} \qquad R_{ij}(x, y) = a_{ij} \cdot x + b_{ij} \cdot y + c_{ij}, \tag{5.13}$$

$$\text{if poly=1,} \qquad R_{ij}(x, y) = c_{ij}, \tag{5.14}$$

$$\text{if poly=4,} \qquad R_{ij}(x, y) = e_{ij} \cdot xy + a_{ij} \cdot x + b_{ij} \cdot y + c_{ij}, \tag{5.15}$$

where, a_{ij}, b_{ij}, c_{ij} and d_{ij} are referred to as the orbital parameters for the interferogram ij. We strongly advise the user to set both `gpsramp` and `netramp` options to `True`.

GPS displacements can be computed in three different ways:

- By using the actual raw GPS time series (not recommended).

- Using a smoothed version using cubic splines by setting the `gpspreproc` option to `True` in the **sbas.xml** or **mints.xml** file (recommended).

- By using modeled GPS time series. One can use modelled time series (SOPAC) or only a GPS velocity field.

Using modelled GPS time series is very convenient because it allows to use any functional form. It is also possible to use only the GPS velocities as input to flatten the interferograms. If a crude deformation model is available, it is possible to simulate a dense network of GPS stations and flatten the interferograms.

Implementation

Using the actual GPS time series, smoothed or not, is the default behavior, when the `gpsderamp` code name is set to `True` in the **sbas.xml** or **mints.xml** files. The `netramp` option in the XML file ensures consistency of the orbital parameters in the network sense. The option `gpsvert` will force the software to use vertical GPS displacements. The option `gpspreproc` will smooth the GPS time series before using them (recommended).

The user needs to specify the path to the GPS station list, using the option `stnlist` in the XML file, and what type of list it is, using the option `stntype` in the XML file. The station list type can be

- **False**: This specifies the default station list type, based on the SOPAC format. The station list file is a 14 columns file, as the ones one can download from SOPAC[6]. Only the columns 1, 5, 6 and 8 are used and are station code names in 4 digits, latitude, longitude and starting date of the time series, respectively.

- **True**: The station list is a 3 columns Ascii file specifying the station code name (4 digits) and its latitude and longitude coordinates.

- **velocity**: The station list is a 6 columns Ascii file specifying the station code name (4 digits), its latitude and longitude coordinates and the North, East and Up velocities. An example can be found in Appendices. In that case, no GPS station files will be used.

The GPS station files corresponding to the code names in the station file list need to be in the directory specified by the option `gpspath` in the XML file. In the SOPAC standard, each station file name has to be the 4

[6] `http://garner.ucsd.edu/pub/timeseries/measures`

digits station code followed by `CleanFlt.neu` (ex: `ctmsCleanFlt.neu` for the station `ctms`). Each file is an ASCII file starting by a header, describing the modeled time series, followed by 9 columns, specifying the date (floating point), the year (integer), the day of the year, North displacement, East displacement, Up displacement, North uncertainty, East uncertainty and Up Uncertainty. An example can be found in the Appendices and in the example directory.

5.5 Checklist

1. Check the outputs from the previous step.
2. Gather GPS data in a directory and create a station list file.
3. Copy the **ProcessStack.py** script to your working directory.
4. Run: `python ProcessStack.py [Options]`
5. Weather model data will be automatically downloaded as needed.
6. Check the new HDF5 file and the newly created PNG previews.

Chapter 6

Time-series: SBAS

Small Baseline Subset-like Time Series analysis are among the most common strategies to describe the ground displacements from a pile of interferograms. The name Small Baseline Subset (hereafter SBAS) was primarily chosen by Berardino et al. [2002], but now represents a wide range of methods [e.g, Usai, 2003, Schmidt and Bürgmann, 2003, Cavalié et al., 2007, Lopez-Quiroz et al., 2009]. We have included three implementations of such algorithms in GIAnT - SBAS, N-SBAS and TimeFun. We also provide a way to estimate uncertainties in the estimated time-series products using a cross-validation based approach.

6.1 Inputs

The input files are outputs from **ProcessStack.py** or **PrepIgramStack.py**, including:

- The HDF5 file from **ProcessStack.py** (default is `Stack/PROC-STACK.h5`) or **PrepIgramStack.py** (default is `Stack/RAW-STACK.h5`).
- The XML file containing the informations about the dataset (typically, **data.xml**).
- The XML file containing the informations about the processing strategies (typically, **sbas.xml**).

6.2 Outputs

The output dataset is organized in a single HDF5 file (default is `Stack/LS-PARAMS.h5` for the SBAS chain, `Stack/NSBAS-PARAMS.h5` for the NSBAS chain and `Stack/TIME-FUN.h5` for the time function inversion).

6.3 The SBAS method

6.3.1 Inversion Strategy

The SBAS method is implemented in the scripts **SBASInvert.py** located in the directory `GIAnT/SCR`. This script uses the informations enclosed in the **sbas.xml** and **data.xml** files and the datasets in the HDF5 input file from either **PrepIgramStack.py** or **ProcessStack.py**.

In the traditional SBAS approach, the set of interferometric phase observations writes as a linear combination of individual SAR scene phase values for each pixel independently:

$$d = \mathbf{G}m \iff \Phi_{ij} = \sum_{n=i}^{j-1} \delta\varphi_n, \tag{6.1}$$

where Φ_{ij} is the pixel phase of the interferogram combining acquisition i and j (i.e. in the data vector d) and $\delta\varphi_n$ is the pixel phase increment between acquisition time n and $n+1$ (.i.e in the model vector m). Here, we select pixels where the dataset is complete (i.e. all interferograms and all acquisition dates are available). This way, we form the linear operator $\mathbf{G}$ once and invert it only once using a least squares scheme.
To run this script, type:

```
>> python SBASInvert.py -h -i FNAME -o ONAME -d DXML -p PXML
```

The command line options are:

- **-h**: Ask for help
- **-i FNAME**: FNAME input HDF5 file. Default is `PROC-STACK.h5`.
- **-o ONAME**: ONAME output HDF5 file. Default is `LS-PARAMS.h5`.
- **-d DXML**: DXML data XML file. Default is **data.xml**.
- **-p PXML**: PXML parameter XML file. Default is **sbas.xml**.

6.3.2 Uncertainties estimation

The script **SBASxval.py** provides an uncertainty estimate for each pixel and epoch. For each pixel, a Jacknife test is performed using subsets generated on the basis of SAR acquisitions. For all SAR acquisitions, we form and invert for the linear system $d_n = \mathbf{G}_n m_n$ corresponding to the dataset without the nth acquisition date. For a dataset with M interferograms, combining N acquisitions, we invert for N linear systems. The standard deviation of the m_n vectors represents the uncertainty. Note that the master SAR acquisition is included in all the subsets and is used as the temporal reference.
To run this script, type:

```
>> python SBASxval.py -h -i FNAME -o ONAME -d DXML -p PXML
```

The command line options are:

- **-h**: Ask for help
- **-i FNAME**: FNAME input HDF5 file. Default is `PROC-STACK.h5`.
- **-o ONAME**: ONAME output HDF5 file. Default is `LS-PARAMS.h5`.
- **-d DXML**: DXML data XML file. Default is **data.xml**.
- **-p PXML**: PXML parameter XML file. Default is **sbas.xml**.

6.3.3 Outputs

Outputs are stored in a HDF5 file. Default is `Stack/LS-PARAMS.h5` for **SBASInvert.py** and `Stack/LS-xval.py` for **SBASxval.py**. The variables are:

- `cmask`: Mask map of the pixels.
- `dates`: Dates of acquisitions.
- `mName`: Name of each model parameter function.
- `masterind`: Index of the master acquisition date.
- `parms`: Output parameter vectors of each pixels inversion.
- `rawts`: Raw time-series for each pixel.
- `recons`: Filtered Time Series for each pixel.

- `regF`: Regularization indicator for model parameters.
- `tims`: Time vector in years, with respect to the master date.
- `error`: Error estimation (only with **SBASxval.py**).

For visualization, please refer to section 8.

6.3.4 Checklist

1. Copy the **SBASInvert.py** script to your working directory.
2. Run: `python SBASInvert.py [Options]`.
3. Wait a minute or two.
4. Copy the **plotts.py** script to your working directory.
5. Check the results by running: `python plotts.py [Options]`. See section 8.
6. If you are happy, copy the **SBASxval.py** script to your working directory.
7. Run: `python SBASxval.py [Options]`

6.4 The NSBAS method

The NSBAS method is implemented in the scripts **NSBASInvert.py** and **NSBASxval.py** located in the `GIAnT/SCR` directory. These scripts use the parameters in **sbas.xml** and **data.xml** files and the datasets in the HDF5 input file from either **PrepIgramStack.py** or **ProcessStack.py**. Extended descriptions of the inversion can be found in Lopez-Quiroz et al. [2009], Doin et al. [2011] and Jolivet et al. [2012].

6.4.1 Inversion strategy

This script **NSBASInvert.py** estimates the LOS phase change of each pixel independently using a linear system built with an ensemble Γ of interferograms and a set of a priori constraints combining N acquisition dates:

$$d = \mathbf{G}m \iff \begin{cases} \forall (i,j) \in \Gamma & \Phi_{ij} = \sum_{n=i}^{j-1} \delta\varphi_n \\ \forall k \in [2, N] & 0 = \sum_{n=1}^{k-1} \delta\varphi_n - f(\Delta t_k) + e\mathrm{B}_{perp}^k \end{cases} \tag{6.2}$$

where, Φ_{ij} is the pixel phase value for the interferogram combining acquisition i and j (.i.e in the data vector d), $\delta\varphi_n$ is the phase increment between acquisition n and $n+1$, $\Delta t_k = t_k - t_0$, e is a Digital Elevation Model error estimation, B^k_{perp} is the perpendicular baseline between satellite paths at acquisition 1 and k. f is a parametric representation of the temporal form of the deformation. This function needs to be specified in the `userfn.py` file, using the `NSBASdict` function. By default, it is assumed to be of the form:

$$f(t) = at^2 + vt + c, \tag{6.3}$$

where a is the pixel acceleration, v is the pixel velocity and c is a constant. The resulting linear operator **G** can be written,

$$\left(\begin{array}{|cccccc|cc}
 & & & & & & 0 & 0 \\
 & & & D & & & \vdots & \vdots \\
 & & & & & & 0 & 0 \\
\hline
1 & 0 & & & \cdots & 0 & -f(\Delta t_1) & -B^1_{perp} \\
1 & 1 & & & & 0 & -f(\Delta t_2) & -B^2_{perp} \\
1 & 1 & 1 & & & 0 & -f(\Delta t_3) & -B^3_{perp} \\
\vdots & & & & & \vdots & \vdots & \vdots \\
1 & 1 & 1 & \cdots & 1 & 0 & & \\
1 & 1 & 1 & 1 & \cdots & 1 & -f(\Delta t_N) & -B^N_{perp}
\end{array}\right), \tag{6.4}$$

where D is the design matrix relating acquisitions to interferograms through equation 6.2.

The function f is used as a regularization function. Its contribution in the linear operator **G** is weighted by a parameter γ, small enough so that, if the SBAS network is complete (i.e. no link between acquisitions is missing), the bottom part of **G** does not influence the inversion and is a fit to the data. If the SBAS network is incomplete and disconnected subsets arise, then the functional form links these subsets. By default, we set the γ parameter to 1e-4. This value can be modified using the command line option `-gamma`.

Practically, the **NSBASInvert.py** script reads the HDF5 file, creates the full linear operator **G**. Then, pixel by pixel, it selects the lines and columns of **G** corresponding to this particular pixel's interferometric network and invert the system using this new operator. Each model parameters are stored in the output HDF5 file, together with the phase filtered and un-filtered evolution. This script is parallelized and multiple threads can be used on a single machine.

To run this script, type:

```
>> python NSBASInvert.py -h -i FNAME -o ONAME -d DXML -p PXML
                         -nproc NPROC -gamma GAMMA
```

The command line options are:

- **-h**: Ask for help.
- **-f FNAME**: Name the of the input HDF5 file. Default is `Stack/PROC-STACK.h5`.
- **-o ONAME**: Name of the output HDF5 file. Default is `Stack/NSBAS-PARAMS.h5`.
- **-d DXML**: DXML is the data Xml file. Default is **data.xml**.
- **-p PXML**: PXML is the parameter Xml file. Default is **sbas.xml**.
- **-nproc NPROC**: Number of processes. Default is 1.
- **-gamma GAMMA**: Weighting parameter. Default is 1e-4.

6.4.2 Traditional stacking [1] as special case

The simplest method to analyze a pile of interferograms is to estimate the Line-of-Sight velocities. Even though such an analysis does not take into account temporal variations in deformation rates compared to robust time-series methods, it is still a good way to quickly look at one's dataset. The traditional velocity estimation by stacking is just a special case of NSBAS inverstion that uses a linear function in time to tie observations between disconnected interferogram clusters. The velocity map is stored as one of the estimated model parameters in the output HDF5 file.

6.4.3 Uncertainties estimation

The script **NSBASxval.py** has not been written for now, but should be available soon.

[1] Velocity estimates

6.4.4 Outputs

Outputs are stored in a HDF5 file. Default is `Stack/NSBAS-PARAMS.h5` for **NSBASInvert.py**. The variables are:

- `cmask`: Mask map of the pixels.
- `dates`: Dates of acquisitions.
- `mName`: Name of each model parameter function.
- `gamma`: Weighting parameter value.
- `ifgcnt`: Number of interferogram used for each pixel.
- `masterind`: Index of the master acquisition date.
- `parms`: Output parameter vectors of each pixels inversion.
- `recons`: Reconstructed filtered Time Series of each pixels.
- `rawts`: Reconstructed un-filtered Time Series.
- `regF`: Regularization indicator for model parameters.
- `tims`: Time vector in years, with respect to the master date.

For visualization, please refer to section 8

6.4.5 Checklist

1. Copy the **NSBASInvert.py** script to your working directory.
2. Modify the `userfn.py` function, if you want a constraint function different from the default one.
3. Run: `python NSBASInvert.py [Options]`.
4. Copy the **plotts.py** script to your working directory.
5. Check the results by running: `python plotts.py [Options]`. See section 8.

6.5 The TimeFun method

The TimeFun method is implemented in the **TimefnInvert.py** script, located in the `GIAnT/SCR` directory. These scripts use the information enclosed in the **sbas.xml** and **data.xml** files and the datasets in the Hdf5 files from either **PrepIgramStack.py** or **ProcessStack.py**. This inversion method is extensively described in Hetland et al. [2011], where it is applied in the wavelet domain.

6.5.1 Inversion strategy

The TimeFun method is an implementation of the temporal inversion scheme developed originally for MInTS [Hetland et al., 2011] directly in the data domain. This method allows one to describe each pixel's phase evolution using a dictionary of user defined functions. For each pixel (m, n), we invert the following linear system,

$$d^{mn} = \mathbf{G}m^{mn} \iff \forall(i,j) \in \Gamma \Phi_{ij}^{mn} = \sum \alpha_k^{mn}(f^k(t_i) - f^k(t_j)) + e^{mn}B_{perp}^{ij}, \tag{6.5}$$

where Γ is the set of all interferograms ij combining the SAR acquisitions with indices i and j, t_i and t_j are the SAR acquisition times, Φ_{ij}^{mn} is the pixel's phase value of interferogram ij (i.e. in the d^{mn} vector). f^k are a set of user defined functions chosen in the provided library that includes seasonal oscillations, polynomial forms, spline functions, integrated spline functions, step functions etc, α_k^{mn} are the corresponding coefficients (i.e. in the vector m^{mn}). B_{perp}^{ij} is the perpendicular baseline of the interferogram ij and e^{mn} is the Digital Elevation Model error term. You can turn on the estimation of e^{mn} using the `demerr` option in the XML file. We build the linear operator $\mathbf{G}$ once and only for pixels that have valid phase observations in all the interferograms.

To set up your functional form, you need to modify the file `userfn.py`. Unless you already did this before, copy this file from `GIAnT/SCR` to your working directory and modify the function `timedict` using the possible keywords (see Appendices, section B.1.1). This file needs to be provided as no default behaviour is assumed by the script.

Two options are possible to invert this linear system and need to be set in the **sbas.xml** file (see section 4.4.2):

- if `regu` is `False`, then, for each pixel mn, we minimize the cost function, F_c^{mn} given by,

$$F_c^{mn} = ||\mathbf{G}m^{mn} - d^{mn}||_2^2. \tag{6.6}$$

The resulting model is the classic non-regularized least-square solution.

- if `regu` is `True`, then, for each pixel mn, we minimize the cost function, F_c^{mn} given by,

$$F_c^{mn} = ||\mathbf{G}m^{mn} - d^{mn}||_2^2 + \lambda^2 ||Hm^{mn}||_2^2, \tag{6.7}$$

where, H is the laplacian operator. In that case, the damping parameter λ is chosen using a generalized singular value decomposition approach. The regularization is only applied on interpolating functions such as the spline functions and the integrated spline functions, and not on the other functions.

To run this script, type:

```
>> python TimefnInvert.py -h -i FNAME -o ONAME -d DXML -p PXML
                                -nproc NPROC -u USER
```

The command line options are:

- **-h**: Ask for help.
- **-i FNAME**: Name of the input Hdf5 file. Default is `Stack/PROC-STACK.py`.
- **-o ONAME**: Name of the output Hdf5 file. Default is `Stack/TS-PARAMS.py`.
- **-d DXML**: DXML is the data Xml file. Default is **data.xml**.
- **-p PXML**: PXML is the parameter Xml file. Default is **sbas.xml**.
- **-nproc NPROC**: Number of processes. Default is 1.
- **-u USER**: The python script with the user defined time dictionary function. Default: userfn.py.

6.5.2 Uncertainties estimation

The script **Timefnxval.py** provides an estimation of the uncertainties on the model parameters, using a Jacknife approach and can be used the same way as **TimefnInvert.py**.

6.5.3 Outputs

The output datasets are stored in a HDF5 file (default is `Stack/TS-PARAMS.h5`). The datasets are:

- `parms`: Inverted parameters.
- `recons`: Reconstructed phase evolution.
- `mName`: Name of the functions specified in `userfn.py`.
- `regF`: Vector indicating wether a function is regularized or not.
- `tims`: Time vector.
- `dates`: Acquisition dates.
- `cmask`: Mask map of the pixels.
- `masterind`: Index of the master SAR acquisition.

For visualization, please refer to section 8

6.5.4 Checklist

1. Copy the script **TimefnInvert.py** to your working directory.
2. Copy (if not done already) and modify the file `userfn.py`.
3. Run: `python TimefnInvert.py [Options]`.
4. Check Results using plotts.py (see section 8).

6.6 Important Note

We have described three different implementations of SBAS-type algorithms in this manual. One of the key strengths of GIAnT is its modularity. It should be fairly simple for the user to incorporate any features from the MInTS processing chain into their SBAS-type processing strategy A simple example would be to explore the use of Covariance matrices and the shape smoothed regularization in Timefun similar to MInTS.

Chapter 7

Time-series: MInTS

Multiscale InSAR Time-Series (MInTS) [Hetland et al., 2011] was originally developed at Caltech and is different from traditional SBAS approaches in two significant ways.

1. The interferometric phase data is transformed into wavelet domain before temporal inversion.
2. A dictionary of user-defined functions is used to describe the temporal evolution of surface deformation.

The original MInTS software was developed in a MATLAB framework and is available for download at `https://secure.earth.lsa.umich.edu/groups/lithosphere/wiki/eb455/MInTS.html`. We have reimplemented the entire package in Python for speed and efficiency. Some of the important components of GIAnT like the Meyer wavelet library and the Tikhonov solver were written primarily for implementing the various MInTS algorithms in Python.

7.1 The MInTS Algorithm

The MInTS processing chain has been extensively described in Hetland et al. [2011]. The processing steps include wavelet transform of the interferograms, parametrized inversion of the wavelet coefficients and inverse wavelet transform of the coefficients. These steps are implemented in the scripts called **DatatoWavelet.py**, **InvertWaveletCoeffs.py** and **WavelettoData.py**. The MInTS processing chain typically follows the atmospheric removal and the flattening of the interferograms, using **ProcessStack.py**.

7.2 Forward Wavelet Transform

The forward wavelet transform process is done using the **DatatoWavelet.py** script, located in the `GIAnT/SCR` directory. This step includes:

1. **Inpainting** - Small decorrelated holes in the interferograms are filled in with an inpainting algorithm. We use a Python version of the inpaint_nans utility [D'Errico, 2006].

2. **Mirroring** - Fast Wavelet Transform (FWT) algorithms are set up to work on matrices of dyadic lengths. We mirror our inpainted images to dyadic lengths before transforming the data. The bounds of the original data in the mirrored array are also saved for future use. The option `minpad` in the **mints.xml** file ensures a minimum fraction of the interferogram width and length will be used for mirroring (default is 0.1).

3. **Forward Wavelet Transform** - We apply the FWT to the mirrored data. We provide support for Meyer wavelets through our own Python library derived from the Wavelab package [Buckheit and Donoho, 1995] and for other wavelets through the pywt package [Wasilewski, 2012]. To use the Meyer wavelets, set the option `wvlt` to `meyer` in the **mints.xml** file.

4. **Reliability Measure** - We also compute the reliability measure of wavelet coefficients by convolving the absolute value of the impulse response with a binary mask representing original and interpolated data.

This script has been parallelized using Python's multiprocessing module and the results of this processing step are stored in **Stack/WAVELET.h5**. To run **DatatoWavelet.py**, type:

```
>> python DatatoWavelet.py  -h -i INAME -o ONAME -nchunk NCHUNK
                            -nproc NPROC -d DXML -p PXML
```

The command line options are:

- **-h**: Ask for help.
- **-i INAME**: input Hdf5 file. Default is defined by the parameters in the parameter Xml file (i.e. if no pre-processing is asked, default is `Stack/RAW-STACK.h5`, otherwise, it is `Stack/PROC-STACK.h5`).
- **-o ONAME**: output Hdf5 file. Default is `Stack/WAVELET.h5`.

- **-d DXML**: data XML file. Default is **data.xml**.
- **-p PXML**: parameter Xml file. Default is **mints.xml**.
- **-nchunk NCHUNK**: Number of interferograms processed in parallel. Default is 40.
- **-nproc NPROC**: Number of processes to operate on a chunk. Default is 1.

The outputs datasets are stored in a HDF5 file (default is `Stack/WAVELET.h5`). The datasets are:

- `Jmat`: Matrix linking interferograms to the acquisition dates, made of 0, 1 and -1 (also called connectivity matrix).
- `bperp`: Vector of the perpendicular baseline values.
- `cmask`: Mask map of the pixels.
- `dates`: Dates of acquisition.
- `offset`: Bounds of the mirrored array.
- `tims`: Vector of time of acquisition with respect to the master.
- `wvlt`: Stack of wavelets from the interferograms.
- `wts`: Stack of the Wavelet weights.

7.3 Time-series inversion of the wavelet coefficients

7.3.1 Inversion strategy

The time-series inversion of the wavelet coefficients is done using the script **InvertWaveletCoeffs.py** or **InvertWaveletCoeffs_folds.py**, located in the `GIAnT/SCR` directory. The inversion scheme is similar to the one used in the TimeFun method (Section 6.5.1).

$$\begin{aligned} d^{mn} &= \mathbf{G}m^{mn} \iff \forall (i,j) \in \Gamma \\ W_{ij}^{mn} &= \sum \alpha_k^{mn}(f^k(t_i) - f^k(t_j)) + e^{mn} B_{perp}^{ij} \end{aligned} \tag{7.1}$$

where W_{ij}^{mn} refers to the particular wavelet coefficient with index mn in interferogram ij. Once again the dictionary of temporal functions is passed to our inversion script using the **userfn.py**. If the user decides to use a non-parametric inversion scheme using splines or integrated splines, the solutions are regularized as described in Appendix E. **InvertWaveletCoeffs_folds.py** represents our optimized implementation of the original MInTS, using a k-fold cross validation approach to chose the damping parameter.

To run the inversion, type:

```
>> python InvertWaveletCoeffs.py -h -i INAME -o ONAME -d DXML -p PXML
                                 -nproc NPROC
```

The command line arguments are:

- **-h**: Ask for help.
- **-i INAME**: input Hdf5 file. Default is `Stack/WAVELET.h5`.
- **-o ONAME**: output Hdf5 file. Default is `Stack/WAVELET-INV.h5`.
- **-d DXML**: data XML file. Default is **data.xml**.
- **-p PXML**: parameter Xml file. Default is **mints.xml**.
- **-nproc NPROC**: Number of processes. Default is 1.

The outputs datasets are stored in a Hdf5 file (default is `Stack/WAVELET-INV.h5`). The datasets are:

- `Jmat`: Matrix linking interferograms to the acquisition dates, made of 0, 1 and -1 (also called connectivity matrix).
- `bperp`: Vector of the perpendicular baseline values.
- `cmask`: Mask map of the pixels.
- `dates`: Dates of acquisition.
- `model`: Description of the function used in the inversion.
- `offset`: Bounds of the mirrored array.
- `tims`: Vector of time of acquisition with respect to the master.
- `wvlt`: Stack of wavelets from the interferograms.

7.3.2 Uncertainties estimation

The script **InvertWaveletCoeffs_xval.py** provides an estimation of the uncertainties on the model parameters using the generalized singular value decomposition approach and may be used the same way as **InvertWaveletCoeffs.py**.

7.4 Inverse Wavelet Transform

The wavelet coefficients estimated in the previous step are transformed into model parameter space using the Inverse Wavelet Transform (IWT). This step is implemented in the **WavelettoData.py** script, located in the `GIAnT/SCR` directory. The image is cropped back to dimensions of the original data set and the deformation time-series is then recreated by putting together the temporal functions and the estimated model parameters. The results are stored in **Stack/WS-PARAMS.h5**.To run this step, type:

```
>> python WavelettoData.py -h -i INAME -o ONAME -d DXML -p PXML
                    -nchunk NCHUNK -nproc NPROC -u USER
```

The command line arguments are:

- **-h**: Ask for help.
- **-i INAME**: input Hdf5 file. Default is `Stack/WAVELET-INV.h5`.
- **-o ONAME**: output Hdf5 file. Default is `Stack/WS-PARAMS.h5`.
- **-d DXML**: data XML file. Default is **data.xml**.
- **-p PXML**: parameter XML file. Default is **mints.xml**.
- **-nchunk NCHUNK**: Number of interferograms processed in parallel. Default is 40.
- **-nproc NPROC**: Number of processes to operate on a chunk. Default is 1.
- **-u USER**: Python script with the user defined python function. Default is userfn.py .

The outputs datasets are stored in a HDF5 file (default is `Stack/WS-PARAMS.h5`). The datasets are:

- `Jmat`: Matrix linking interferograms to the acquisition dates, made of 0, 1 and -1 (also called connectivity matrix).
- `cmask`: Mask map of the pixels.
- `dates`: Dates of acquisition.
- `masterind`: Index of the master date.
- `model`: Model parameters maps.
- `modelstr`: Model description.
- `recons`: Reconstructed phase at each date of acquisition.
- `tims`: Vector of time of acquisition with respect to the master.

7.5 Note

One of the strengths of GIAnT is it's modularity. We currently provide the original implementation of MInTS with the GIAnT. However, it should be trivial for the users to reuse modules from the SBAS-type algorithms and apply them to the wavelet coefficients directly.

7.6 Checklist

1. Convert interferograms to the wavelet domain: `python DatatoWavelet.py [Options]`
2. Check the Hdf5 file produced.
3. Copy (if you have not done so before) the `userfn.py` file in your working directory, and modify the MInTS dictionnary of functions.
4. Run `python InvertWaveletCoeffs.py [Options]`.
5. Check the Hdf5 file produced.
6. Run `python WavelettoData.py [Options]`.
7. Check the results using the script `plotts.py`.
8. Run `python InvertWaveletCoeffs_xval.py` to get the uncertainties.
9. Run `python WavelettoData.py -f WAVELET-INV-xval.h5`.

Chapter 8

Visualization and Geocoding

8.1 Interactive visualization

We provide a visualization tool, called **plotts.py**, located in `GIAnT/SCR`. This tool can be used with any time series output from GIAnT. You can run **plotts.py**, typing:

```
>> python plotts.py -h -e -f FNAME -i TIND -m MULT -y YINF YSUP
                    -ms MSIZE -mask MASKFILE MASKXML -raw
                    -model -zf
```

The command line options are:

- **-h**: Ask for help
- **-f FNAME**: FNAME input HDF5 file with time-series estimates. Default is `Stack/LS-PARAMS.h5`.
- **-i TIND**: index of the slice to display.
- **-m MULT**: multiplicative factor. Default is 0.1 (i.e. results in cm).
- **-y YINF YSUP**: the lower and upper colorbar plot limits. Default is -25 25.
- **-ms MSIZE**: Marker size on the phase evolution plot. Reduce if error bars are too small. Default is 5.
- **-mask MASKFILE MASKXML**: MASKFILE is a binary file used for masking output data, if needed. MASKXML is the Xml file that contains the width and length of the binary file. Default is no mask.

- **-raw**: Flag to display the filtered and un-filtered time series of a pixel (Only for NSBAS outputs).
- **-model**: Plot the modeled phase value, along with the phase change (works only with NSBAS,TimeFun and MInTS) .
- **-zf**: Change the reference time for plotting.
- **-e**: Plot the errorbars if available (works with Xval scripts).

8.2 Movies

We include a simple python script **G**IANT/SCR/make_movie.py to build a quick movie of your estimated time-series in radar coordinates. This script can be used as a template in combination with the geocoding library to make movies in geocoded domain as well. You can generate movies of your time-series using

```
>> python make_movie.py -h -i FNAME -o ONAME -nslice NSLICE
                       -y YINF YSUP -win GWIN -model
                       -pix I0 J0 -fps FPS -geo DXML
```

The command line options are:

- **-h**: Ask for help
- **-i FNAME**: FNAME input HDF5 file with time-series estimates. Default is `Stack/LS-PARAMS.h5`.
- **-o ONAME**: output movie name. Default: movie.mp4 .
- **-nslice NSLICE**: Number of time-slices to divide your total time span into. Default: 100.
- **-y YINF YSUP**: the lower and upper colorbar plot limits. Default is -25 25.
- **-win GWIN**: Width of the Gaussian window in years used for interpolation. Default: 0.33.
- **-pix I0 J0**: (Azimuth,Range) coordinates of pixel in radar coordinates, whose time-series will also be plotted.
- **-model**: If you use a parameteric inversion, use the model to interpolate rather than the reconstructed time-series. Default: False

- **-fps**: Frames per second for the movie. Default: 10.
- **-geo DXML**: The data.xml file. If provided, the movie will be generated in geo-coded domain.

8.3 KML

The estimated time-series can also be spit out as a KML file with a time-slider for immediate visualization in Google Earth. The script is similar to the one that generates movies. The code does include system calls to **ImageMagick's convert** command for making parts of the image transparent and **zip** command to generate a KMZ file from the generated KML and PNG files. A colorbar is also automatically generated as an overlay.

```
>> python make_kml.py -h -i FNAME -x XNAME -o ONAME
                      -nslice NSLICE -y YINF YSUP
                      -win GWIN -model -dir DIRI -trans
```

The command line options are:

- **-h**: Ask for help
- **-i FNAME**: FNAME input HDF5 file with time-series estimates. Default is `Stack/LS-PARAMS.h5`.
- **-o ONAME**: output movie name. Default: movie.mp4 .
- **-nslice NSLICE**: Number of time-slices to divide your total time span into. Default: 100.
- **-y YINF YSUP**: the lower and upper colorbar plot limits. Default is -25 25.
- **-win GWIN**: Width of the Gaussian window in years used for interpolation. Default: 0.33.
- **-model**: If you use a parameteric inversion, use the model to interpolate rather than the reconstructed time-series. Default: False
- **-x DXML**: The data.xml file. This is needed as the file includes information about the lat/lon files.
- **-trans**: The undefined data in the images (NaNs) are made transparent. Default: False

8.4 Geocoding

GIAnT includes geocoding library routines in **G**IAnT/geocode directory. GIAnT uses the pyresample library to transform data to and from radar domain and geocoded domain. Currently, PyAPS and our geocoding modules only support the WGS84 format. Extending support to other projections should be relatively simple using pyresample.

An example script to geocode results from our output HDF5 files has been included in **G**IANT/SCR/rdr2geo.py. The users should copy this script and suitably modify it for their applications. Besides flat binary files, the users should also be able to output data to GMT netcdf format and OGR VRT files using the provided library.

Appendix A

I/O Utilities

A.1 Text files

We include a bunch of functions in **tsio.py** to read in data from ASCII files into python arrays and lists.

- **textread(fname,strformat)**
  This is very similar to textread utilities from MATLAB®.

  ```
  [fname,index,bperp] = textread('input.txt','S I K F')

  S - String, I - Integer, F - Float, K - Skip

  The function returns a single list of values that are read
   in form a space/tab delimited text file.
  ```

- **read_rsc(fname)**
  This allows us to read in a ROI_PAC style RSC file into a python dictionary.

  ```
  rdict = read_rsc('example.unw')

  Note that the .rsc extension in the file name is optional.
  Elements of the dictionary can be accessed as
  rdict['WIDTH'], rdict['FILE_LENGTH'] etc.
  ```

- **write_rsc(rdict,fname)**
  Writes the contents of the dictionary to file give by fname. The keys of the dictionary become the headings for the entries of the RSC file.

A.2 Binary files

We include the following functions to read in files from binary files into numpy arrays.

- **load_mmap(fname, nxx, nyy, map, nchannels, channel, datatype, quiet, conv)**
  This allows to memory map data in a binary file to a numpy array. This function can handle BIL, BIP and BSQ data formats. The default values are (map='BSQ', nchannels=1, channel=1, datatype=numpy.float32, conv=False). Mapping the file to a numpy array allows us to access the contents of the file directly like in an array.

```
#To map a simple float file of size 100 x 100
arr = load_mmap(fname, 100, 100)

#To map an RMG file of size 100 x 100 with phase in channel 2
arr = load_mmap(fname, 100, 100, map='BIL',
         nchannels=2, channel=2)
```

- **load_flt(fname, nxx, nyy, datatype, scale, conv)**
  Load a simple flat binary file into memory. Unlike the memory map, all the data is loaded into memory.

- **load_rmg(fname, nxx, nyy, datatype, scale, conv)**
  Load both channels of an RMG file into memory.

A.3 HDF5 Files

We have two simple utilities for reading and writing HDF5 files in **tsio.py**. These should only be used for simple debugging applications and for small datasets. HDF5 file interface through h5py is already fairly simple and we mostly make direct calls to h5py functions to deal with HDF5 files.

- **saveh5(fname,rdict)**
  Save the contents of specified dictionary (rdict) to a given HDF5 file (fname). The keys of the dictionary automatically become dataset names.

- **loadh5(fname)**
  Returns the contents of a HDF5 file (fname) in a dictionary. The names of the datasets becomes the keys in the dictionary.

A.4 GMT netcdf files

We include simple utilities to read data directly from GMT-style netcdf (grd) files. These are used to load data from GMTSAR products.

- **load_grd(fname, var='z')**
  Load a variable (var) from a given GMT-style netcdf file (fname).

- **get_grddims(fname, var='z')**
  Get the dimensions of a variable (var) from a given GMT-style netcdf file (fname).

A.5 XML Files

We use XML files to set processing parameters for our scripts. We include utilities for reading and writing XML files in **tsxml.py**. All XML processing is done through the TSXML class defined in tsxml.py. We use the lxml.eTree package to generate XML files and lxml.objectify to read them.

A.5.1 XML format

All data entries in our XML files are of format.

```
<params>
<width>
<value>500</value>
<type>INT</type>
<help>Width of interferograms.</help>
</width>
</params>
```

By design, we force all XML field entries (other than branch nodes) to have a help string describing the parameter. This has been enforced to ensure that processing parameters are easily understood by users from different backgrounds. Currently, we support INT, FLOAT, BOOL and STR data types.

A.5.2 Creating XML file

```
#Creating XML with params as root node.
g = TSXML('params')
```

```
br = g.addnode(g.root,'params')
g.addsubelement(br,'width',100,'Width of interferograms.')
g.writexml('data.xml')
```

The TSXML class includes utilities to prepare **data.xml**, **sbas.xml** and **mints.xml**. These different XML files are used for processing and are created using the **prepxml.py** utility provided with the scripts. We also provide a method to create a generic XML file using keywords.

A.5.3 Reading XML file

```
#Reading an XML file.
h = tsinsar.TSXML('data.xml',File=True)
wid = h.data.master.width
```

All the data in XML is converted into a structured object that can be directly used in your scripts.

A.6 DEM RSC file for non ROI_PAC data

Here is an example RSC file to be included with the radar simulation or DEM in case, the data was processed using a processor other than ROI_PAC. The dimensions are the most important part of this file. The latitude and longitude values are read in to determine the approximate bounds of the scene for subsetting weather model data.

```
WIDTH     350
FILE_LENGTH 500
LAT_REF1 38.5
LON_REF1 -119.5
LAT_REF2 37.3
LON_REF2 -118.0
LAT_REF3 38.5
LON_REF3 -119.5
LAT_REF4 37.3
LON_REF4 -118.0
AZIMUTH_PIXEL_SIZE 180.000000
RANGE_PIXEL_SIZE 63.300000
```

A.7 GPS Input Files

A.7.1 Option 1: Velocity type station list

The station locations and average velocities are provided in a single file.

```
a001    11.80000    42.80000    0.00604    0.01106    -0.00000
a002    12.00000    42.80000    0.00591    0.01106    -0.00000
a003    12.00000    43.00000    0.00592    0.01119     0.00000
a004    12.00000    43.20000    0.00592    0.01131     0.00000
....
```

A.7.2 Option 2: Station-wise time series

The coordinates of the GPS station can be provided in two different ways:

Option a: Classic Station List

```
7odm  34.11640714 -117.09319315
ab01 52.20950520 -174.20475602
ab02 52.97060620 -168.85467007
ab04 63.65686535 -170.56744305
...
```

Option b: SOPAC station list

This data can be retrieved by e-mail from the SOPAC archive.

```
7odm -2407750.9707 -4706536.6674  3557571.4197  34.11640714 -117.09319315  \\
 762.0806 2004.4932 0.0048 0.0059 0.0046 0.0024 0.0043 0.0074
ab01 -3896562.8770  -395471.6423  5017141.8417  52.20950520 -174.20475602   \\
 25.4568 2004.4932 0.0031 0.0019 0.0037 0.0018 0.0019 0.0045
ab02 -3776808.0832  -744083.8296  5068728.1267  52.97060620 -168.85467007  \\
 192.7802 2004.4932 0.0024 0.0014 0.0029 0.0017 0.0014 0.0034
ab04 -2799600.4279  -465105.4035  5692966.4183  63.65686535 -170.56744305   \\
136.5690 2004.4932 0.0025 0.0011 0.0044 0.0015 0.0010 0.0048
....
```

Station File example

The time-series of individual stations must be described in the SOPAC format.

```
#################################################################
# Refined Model Terms:
#
#                   n component
#     slope 1: -0.0088 +/- 0.0000 m/yr (1999.1932 - 2010.0644)
#    offset 1: -0.0002 +/- 0.0002 m (1999.6178)
#      annual:  0.0012 +/- 0.0001 m; phase: 4.10
# semi-annual:  0.0003 +/- 0.0000 m; phase: 2.84
#
#                   e component
#     slope 1: -0.0153 +/- 0.0000 m/yr (1999.1932 - 2010.0644)
#    offset 1:  0.0013 +/- 0.0002 m (1999.6178)
#      annual:  0.0007 +/- 0.0001 m; phase: 4.51
# semi-annual:  0.0004 +/- 0.0001 m; phase: 1.48
#
#                   u component
#     slope 1:  0.0000 +/- 0.0001 m/yr (1999.1932 - 2010.0644)
#    offset 1: -0.0031 +/- 0.0006 m (1999.6178)
#      annual:  0.0048 +/- 0.0002 m; phase: 3.98
# semi-annual:  0.0008 +/- 0.0001 m; phase: 2.53
#
# ---------------------------------------------------------------
#
##################################################################
1999.1932 1999 071  0.0486  0.0834 -0.0011  0.0025  0.0020  0.0026
1999.1959 1999 072  0.0488  0.0828 -0.0004  0.0024  0.0019  0.0025
1999.1986 1999 073  0.0484  0.0829  0.0000  0.0024  0.0019  0.0026
1999.2014 1999 074  0.0492  0.0820 -0.0030  0.0025  0.0020  0.0026
1999.2041 1999 075  0.0491  0.0827 -0.0006  0.0026  0.0021  0.0027
1999.2068 1999 076  0.0490  0.0821 -0.0015  0.0024  0.0019  0.0025
1999.2096 1999 077  0.0493  0.0828  0.0006  0.0024  0.0019  0.0026
1999.2123 1999 078  0.0496  0.0835  0.0008  0.0027  0.0021  0.0028
1999.2151 1999 079  0.0497  0.0827  0.0025  0.0026  0.0021  0.0028
1999.2178 1999 080  0.0489  0.0821 -0.0012  0.0025  0.0020  0.0026
...
```

Appendix B

Time-series Utilities

B.1 Temporal characterization

One of the strengths of GIAnT is the freedom for representing the temporal behaviour of surface deformation as a combination of functions from a predefined dictionary, including b-splines and integrated b-splines. We provide a simple framework to call this dictionary of functions and build design matrices. The related functions are included in the file **tsutils.py**. The most relevant function is **tsutils.Timefn(rep,t)**.

B.1.1 Dictionary of functions

We currently provide support for the following functions. It is trivial to customize and add functions to the dictionary.

- **Linear rate**
  Python representation `['LINEAR',[t1,t2,...,tN]]` is equivalent to N rows of design matrix such that

$$f_k\left(t\right) = \left(t - t_k\right) \tag{B.1}$$

- **Polynomial**
  Python representation `['POLY',[p1,p2,...,pN],[t1,t2,...,tN]]` is equivalent to `sum([p1,..,pN])+N` rows of the design matrix such that

$$f_{i,k}\left(t\right) = \left(t - t_i\right)^k \quad i \in [0, 1, ..., p_i] \tag{B.2}$$

- **Power**
  Python representation `['POW',[p1,p2,...,pN],[t1,t2,...,tN]]` is

equivalent to `N` rows of the design matrix such that

$$f_k(t) = (t - t_k)^{p_k} \tag{B.3}$$

- **Exponential decay**
  Python representation `['EXP',[t1,t2,...,tN],[tau1,tau2,...,tauN]]` is equivalent to N rows of the design matrix such that

$$f_k(t) = \left[1 - \exp\left(\frac{t - t_k}{\tau_k}\right)\right] \cdot u(t - t_k) \tag{B.4}$$

- **Logarithmic decay**
  Python represetntation `['LOG',[t1,t2,...,tN],[tau1,tau2,...,tauN]]` is equivalent to N rows of the design matrix such that

$$f_k(t) = \log\left(1 + \frac{t - t_k}{\tau_k}\right) \cdot u(t - t_k) \tag{B.5}$$

- **Step function**
  Python representation `['STEP',[t1,t2,...,tN]]` is equivalent to N rows of the design matrix such that

$$f_k(t) = u(t - t_k) \tag{B.6}$$

- **Seasonal**
  Python representation `['SEASONAL',[tau1,tau2,...,tauN]]` is equivalent to 2N rows of the design matrix such that

$$\begin{aligned} f_{2k-1}(t) &= \cos\left(2\pi\frac{t}{\tau_k}\right) \\ f_{2k}(t) &= \sin\left(2\pi\frac{t}{\tau_k}\right) \end{aligned} \tag{B.7}$$

- **B-Splines**
  Python representation `['BSPLINE',[Ord1,...,OrdN],[n1,...,nN]]` is equivalent to `prod([n1,...nN])*len(t)` rows of the design matrix. "Ord$_k$" represents the order and "n$_k$" represents the number of equally spaced splines between minimum and maximum values of the time vector.

- **Integrated B-Splines**
  Python representation `['ISPLINE',[Ord1,...,OrdN],[n1,...,nN]]` is equivalent to `prod([n1,...nN])*len(t)` rows of the design matrix. "Ord$_k$" represents the order and "n$_k$" represents the number of equally spaced splines between minimum and maximum values of the time vector.

B.1.2 Putting functions together

```
rep = [ ['LINEAR',[0.0,9.5]],

       ['EXP',[4.0,8.0],[1.0,3.0]],

       ['BSPLINE',[3],[16]],

       ['LOG',[2.0,9.0],[1.0,3.0]],

       ['SEASONAL',[0.5,1.0]],

       ['ISPLINE',[3],[16]]]

H,vname,rflag = tsutils.Timefn(rep,t)
```

H represents the design matrix, **vname** is a list with a unique name for each parameter in our temporal model and **rflag** is a vector indicating if the particular parameter needs to be regularized. The rflag vector is automatically used to construct the regularization operator. If multiple families of splines are used with different scales, each family has a unique regularization flag. Thus, the regularization operator automatically takes this into account.

B.1.3 SBAS Formulation

We also use the same Timefn to generate the design matrix for implementing SBAS.

```
H,vname,rflag = Timefn(['SBAS',[masterind]])
```

We allow the users to choose a master scene other than first SAR acquisition for formulation their time-series inversions. The users may need to make minor adjustments to the returned matrix depending on the exact implementation of the inversion scheme. Examples of such adjustments can be found in various SBAS-style scripts distributed with GIAnT.

B.2 Network utilities

We also provide a set of functions to create interferogram network related matrices.

- **ConnMatrix(dates,sensor)**
  Creates the connectivity matrix $[1, -1]$ for the interferogram network using the dates and satellite sensor as inputs.

- **conntoPairmat(Jmat)**
  Returns an edge list of interferograms using the connectivity matrix as input.

- **conntoAdjmat(Jmat)**
  Returns the adjacency matrix using the connectivity matrix as input.

- **adjmattoAdjlist(Amat)**
  Returns the adjacency list using the adjacency matrix as input.

- **simpleCycle(Jmat)**
  Returns a list of all simple cycles using the connectivity matrix as input.

Appendix C

Image utilities

GIAnT also includes a few 2D array manipulation routines, typically used in preparing interferograms before processing.

C.1 MInTS image routines

These are routines used in preprocessing interferograms before wavelet transforms in MInTS.

- **mints.inpaint(matrixwithnans)**
  The routine inpaints nans in the input matrix similar to the inpaint_nans (`http://www.mathworks.com/matlabcentral/fileexchange/4551-inpaintnans`) package developed by John D' Errico. We have only implemented the spring metaphor.

- **mints.Mirrortodyadic(matrix, frac)**
  Mirrors the input matrix to a dyadic length such that a specfied fraction is always mirrored. This is done to reduce edge effects in the wavelet transforms. Mirroring fraction is typically 20%.

C.2 Stack routines

These routines are used for processing stacks in general and are included in **stackutils.py** .

- **Lookdown(matrix,looks,method,var$\bar{\text{F}}$alse)**
  Multi-looking of interferograms. Method can either be - mean or median. If desired, the variance of multi-looked chips are also returned.

- **estramp(phs,mask,poly)**
  Estimates the ramp polynomial for an unwrapped interferogram using a mask. Poly can be either 1, 3 or 4.

- **deramp(phs,ramppoly)**
  Deramping of interferograms using a polynomail. Poly can be of length 1 (constant), 3 (Planar) or 4 (Product of linear terms).

- **nanmean(x)**
  Returns the mean of an array while taking care of NaNs.

Appendix D

Wavelets

The original implementation of MInTS Hetland et al. [2011] used the Meyer wavelet routines in the Wavelab850 package (`http://www-stat.stanford.edu/~wavelab`) for all wavelet operations. We have rewritten the Meyer wavelet routines in Python and have added new ones for analyzing rectangular datasets. We have also added routines that efficiently compute the impulse response and the reliability score of wavelet coefficients over interpolated holes. All routines related to the Meyer wavelet transforms can be found in **meyer.py**.

We have also included support for different wavelet functions using the pyWavelets package. The interface to **meyer.py** has been replicated for these set of wavelet basis in **wvlt.py**. Though, we prefer to work with Meyer wavelets in our work these other wavelets can be used in the development of future applications.

In our approach, we use wavelets to reduce the effect of spatially correlated noise in the estimated time-series. We explicitly avoid interpreting the information at various spatial scales as different components of deformation/ orbits / atmosphere like Shirzaei and Walter [2011], as these may be dependent on the family of wavelets used for analysis.

D.1 Convention

The original version of MInTS used cells to store matries of wavelet coefficients. We have decided to retain all the wavelet coefficients in a single 2D matrix (same size as original data matrix) for faster access during time-series inversions. This also allows us to use other features of MInTS directly for time-series analysis of InSAR data directly, instead of the wavelet coefficients if needed. Instead of the cell structure, we provide a lookup function that returns the four corners of the sub-matrix that contains the coefficients at a particular scale and quadrant. We have also reversed the labelling of scales in MInTS to directly relate the scale to the number of wavelet coefficients at any particular scale. Figure D.1 describes the convention used in MInTS to store the wavelet coefficients.

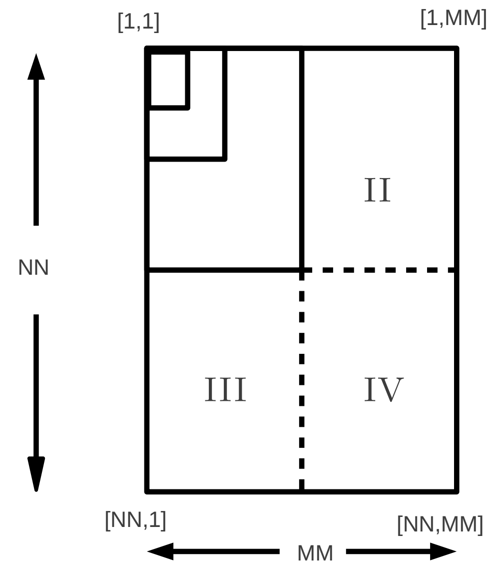


Figure D.1: Wavelet transform convention of an image of size (NN $\times$ MM). We always assume that NN $\geq$ MM in the routines. The smallest scale for analysis is 3 (corresponding to 8 pixels along width) and the highest scale corresponds to $\log_2(\text{MM})-1$ (corresponding to half the width of the image). The quadrants are named according to the convention shown above.

D.2 Routines

The functions in our custom written Meyer wavelet library (meyer.py) are

- **w = meyer.fwt2_meyer(matrix,degree,minscale)**
  2D wavelet transform of rectangular matrix. Scale definition is with

respect to the width of the image. Dyadic lengths assumed. "degree" and "minscale" always set to 3, for all the functions.

- **mat = meyer.iwt2_meyer(matrix,degree,minscale)**
  2D inverse wavelet transform of rectangular matrix. Scale definition is with respect to the width of the image. Dyadic lengths assumed. "degree" and "minscale" always set to 3, for all the functions.

- **ii,jj = meyer.get_corners(matrix.shape,scale,quadrant)**
  Returns the minimum and maximum values of rows and columns of the sub-matrix corresponding to the specified scale and quadrant, from a given wavelet coefficient matrix. This mechanism is the alternative to the cell structure used in the original MInTS software.

- **meyer.impulse_resp(matrix.shape,fname)**
  Writes the impulse response for 2D wavelet transform for a matrix of given shape to a specified HDF5 file. This is an important change to MInTS and now allows us to apply MInTS to very large matrices with ease. The impulse response for a matrix of given size (dyadic lengths) must be computed and stored before the analysis of the wavelet coefficients.

- **wt = meyer.CoeffWeight(mask,responsefname)**
  Computes the reliability of wavelet coefficients using the mask of real and interpolated data, and the HDF5 file containing the appropriate impulse response as inputs.

D.3 Other wavelets

The functions in our generalized wavelet library (wvlt.py) are designed to be compatible with our original Meyer wavelet library. The corresponding functions are

- **w = wvlt.fwt2(matrix,wvlt='db12')**
- **mat = wvlt.iwt2(matrix,wvlt='db12')**
- **ii,jj = wvlt.get_corners(matrix.shape,scale,quadrant)**
- **wvlt.impulse_resp(matrix.shape,fname,wvlt='db12')**
- **wt = wvlt.CoeffWeight(mask,responsefname,wvlt='db12')**

Appendix E

Solvers

Various inversion algorithms are used to various stages of processing in GIAnT. We have included a set of customized solvers in the solver directory.

E.1 Least squares

Numpy and Scipy have inbuilt optimized least squares solvers. We use these solvers with the default conditional parameter of **rcond**$= 1.0e-8$ for stability.

E.2 Regularized L_2 norm

We use Tikhonov regularization for inversions in our MInTS (Chapter 7) implementation and for TimefnInvert (Chapter 6) . Our implementation of Tikhonov regularization is included in **tikh.py** in the solver directory. We use the class TIKH to setup and solve our problems. We set up the regularized minimization formulation as

$$\underset{m}{\operatorname{argmin}} \; ||G \cdot m - d||^2 + \lambda ||H \cdot m||^2 \tag{E.1}$$

where G is the design matrix, m represents model parameters, H is the smoothing or damping matrix and d represents the set of observations. The regularization parameter λ can be chosen in multiple ways:

1. Generalized Cross Validation (GCV)

2. Curvature of the L-curve

3. Quasi optimality condition

4. K-fold cross validation

Our implementation of the first three methods is based on the regutools toolbox Hansen [2007]. We have adopted LAPACK's dggsvd routine to compute the generalized SVD. Our generalized SVD (gsvd module) returns a factorization that is different from the one returned by cgsvd from regutools. We have suitably modified our solver functions to account for this. As suggested on the regutools webpage, we also implemented a pre-processor for the regularization operator (H) using a rank revealing QR decomposition from UTV tools Fierro et al. [1999]. All the four algorithms can be ranked according to performance as follows:

Table E.1: Performance of various regularization parameter selection algorithms.

Method	Speed	Smoothness of solution
GCV	Fast	Roughest
L-curve	Fast	Rough
Quasi optimality	Fastest	Smoothest
K-folds	Slow	Moderately smooth

E.3 Iteratively reweighted least squares

We also included an implementation of the IRLS solver in the script **iterL1.py**. We use IRLS for empirical correction of topography-correlated atmosphere [Lin et al., 2010b].

Bibliography

P. Berardino, G. Fornaro, R. Lanari, and E. Sansosti. A new algorithm for surface deformation monitoring based on small baseline differential sar interferograms. IEEE TGRS, 40:2375–2383, 2002.

J. Biggs, T. Wright, Z. Lu, and B. Parsons. Multi-interferogram method for measuring interseismic deformation: Denali fault, alaska. GEOPHYSICAL JOURNAL INTERNATIONAL, 170(3):1165–1179, Sep 2007. ISSN 0956-540X.

Jonathan B. Buckheit and David L. Donoho. Wavelab and reproducible research, 1995. URL `http://www-stat.stanford.edu/~wavelab`.

O. Cavalié, M. P. Doin, C. Lasserre, and P. Briole. Ground motion measurement in the Lake Mead area, Nevada, by differential synthetic aperture radar interferometry time series analysis: Probing the lithosphere rheological structure. J. Geophys. Res., 112(B3), MAR 13 2007. ISSN 0148-0227. doi: {10.1029/2006JB004344}.

O. Cavalié, C. Lasserre, M. P. Doin, G. Peltzer, J. Sun, X. Xu, and Z. K. Shen. Measurement of interseismic strain across the Haiyuan fault (Gansu, China), by InSAR. Earth Planet. Sci. Lett., 275(3-4):246–257, NOV 15 2008. ISSN 0012-821X. doi: {10.1016/j.epsl.2008.07.057}.

Andrew Collette. Hdf5 for python, 2008. URL `http://h5py.alfven.org`.

John D'Errico. Inpaint_nans, 2006. URL `http://www.mathworks.com/matlabcentral/fileexchange/4551`.

C. DiCaprio and M. Simons. Importance of ocean tidal load corrections for differential insar. Geophys. Res. Lett., 35(L22309), 2008. doi: 10.1029/2008GL035806.

M. P. Doin, S. Guillaso, R. Jolivet, C. Lasserre, F. Lodge, G. Ducret, and R. Gradin. Presentation of the small baseline nsbas processing chain on a case example: The etna deformation monitoring from 2003 to 2010 using envisat data. In FRINGE 2011 ESA Conference, Frascati, Italy, September 2011. ESA.

M. P. Doin, C. Lasserre, G. Peltzer, O. Cavalie, and C. Doubre. Corrections of stratified tropospheric delays in SAR interferometry: Validation with global atmospheric models. J. App. Geophys., 69(1, Sp. Iss. SI):35–50, SEP 2009. ISSN 0926-9851. doi: {10.1016/j.jappgeo.2009.03.010}.

G. Ducret, M.P. Doin, R. Grandin, C. Lasserre, and S. Guillaso. Dem corrections before unwrapping in a small baseline strategy for insar time series analysis. In Geoscience and Remote Sensing Symposium (IGARSS), 2011 IEEE International, pages 1353–1356. IEEE, 2011.

R. D. Fierro, P. C. Hansen, and P. S. K. Hansen. Utv tools: Matlab templates for rank-revealing utv decompositions. Numerical Algorithms, 20:165–194, 1999.

P. C. Hansen. Regularization tools version 4.0 for matlab 7.3. Numerical Algorithms, 46:189–194, 2007.

E. A. Hetland, P. Muse, M. Simons, Y. N. Lin, P. S. Agram, and C. J. DiCaprio. Multiscale insar time series (mints) analysis of surface deformation. J. Geophys. Res., 117(B02404), 2011. doi: 10.1029/2011JB008731.

A. Hooper. A multi-temporal insar method incorporating both persistent scatterer and small baseline approaches. Geophys. Res. Lett., 35(L16302), 2008. doi: 10.1029/2008GL034654.

A. Hooper, P. Segall, and H. Zebker. Persistent scatterer insar for crustal deformation analysis, with application to volcán alcedo, galápagos. Journal of Geophys. Res., 112(B07407), 2007. doi: 10.1029/2006JB004763.

John D. Hunter. Matplotlib: A 2d graphics environment. Computing in Science & Engineering, 9(3):90–95, 2007. URL `http://link.aip.org/link/?CSX/9/90/1`.

R. Jolivet and P. S. Agram. Python-based atmospheric phase screen mitigation using atmospheric re-analysis, 2012. URL `http://pyaps.googlecode.com`.

R. Jolivet, R. Grandin, C. Lasserre, M. P. Doin, and G. Peltzer. Systematic insar tropospheric phase delay corrections from global meteorological reanalysis data. Geophys. Res. Lett., 38(L17311), 2011. doi: 10.1029/2011GL048757.

R. Jolivet, C. Lasserre, M.-P. Doin, S. Guillaso, G. Peltzer, R. Dailu, and J. Sun. Shallow creep on the haiyuan fault revealed by insar. JGR, In Press, 2012.

B. Kampes, R. Hanssen, and Z. Perski. Radar interferometry with public domain tools. Frascati, Italy, December 2003. ESA.

Y. N. Lin, M. Simons, E. A. Hetland, P. Muse, and C. DiCaprio. A multiscale approach to estimating topographically-correlated propagation delays in radar interferogram. Geochem. Geophys. Geosys., 2010a.

Y. N. Lin, M. Simons, E. A. Hetland, P. Muse, and C. DiCaprio. A multiscale approach to estimating topographically correlated propagation delays in radar interferograms. G-Cubed, 11(Q09002), 2010b.

P. Lopez-Quiroz, M.-P. Doin, F. Tupin, P. Briole, and J.-M. Nicolas. Time series analysis of Mexico City subsidence constrained by radar interferometry. J. App. Geophys., 69(1, Sp. Iss. SI):1–15, SEP 2009. ISSN 0926-9851. doi: {10.1016/j.jappgeo.2009.02.006}.

Fernando Pérez and Brian E. Granger. Ipython: A system for interactive scientific computing. Computing in Science & Engineering, 9(3):21–29, 2007. URL `http://link.aip.org/link/?CSX/9/21/1`.

P. Rosen, S. Hensley, G. Peltzer, and M. Simons. Updated repeat orbit interferometry package released. Eos Trans. AGU, 85((5)):47, 2004a.

P. Rosen, S. Hensley, G. Peltzer, and M. Simons. Updated repeat orbit interferometry package released. EOS Trans. AGU, 85(5):47, 2004b. doi: 10.1029/2004EO050004.

P. Rosen, E. Gurrola, G. Sacco, and H. A. Zebker. Insar scientific computing environment- the home stretch. San Francisco, CA, USA, December 2011. AGU. AGU Fall Meeting.

D. Sandwell, R. Mellors, X. Tong, M. Wei, and P. Wessel. Open radar interferometry software for mapping surface deformation. EOS Trans. AGU, 28(92). doi: 10.1029/2011EO280002.

D. A. Schmidt and R. Bürgmann. Time-dependent land uplift and subsidence in the Santa Clara valley, California, from a large interferometric synthetic aperture radar data set. J. Geophys. Res., 108:2416–+, September 2003. doi: 10.1029/2002JB002267.

P. Shanker and H. Zebker. Persistent scatterer selection using maximum likelihood estimation. Geophysical Research Letters, 34(22):L22301, 2007.

M. Shirzaei and T. R. Walter. Estimating the effect of satellite orbital error using wavelet-based robust regression applied to insar deformation data. IEEE TGRS, 49(11):4600 – 4605, 2011.

S. Usai. A least squares database approach for SAR interferometric data. IEEE Transactions on Geoscience and Remote Sensing, 41(4):753–760, 2003.

Filip Wasilewski. Pywavelets - discrete wavelet transforms in python, 2012. URL `http://www.pybytes.com/pywavelets/`.

P. Wessel and W. Smith. New version of the generic mapping tools released. Eos Trans. AGU, 76(329), 1995.

Attachment 2:

GIAnT User Manual

A noise model for InSAR time-series

P. S. Agram[*,a], M. Simons[a]

[a]*Seismological Laboratory, California Institute of Technology, Pasadena, CA-91125,USA*

Abstract

Interferometric Synthetic Aperture Radar (InSAR) time-series methods estimate the spatio-temporal evolution of surface deformation by incorporating information from multiple SAR interferograms. While various models have been developed to describe the interferometric phase and correlation statistics in individual interferograms, efforts to model the generalized covariance matrix that is directly applicable to joint analysis of networks of interferograms have been limited in scope. In this work, we build on existing decorrelation and atmospheric phase screen models and develop a simple covariance model (over space and time) for interferometric phase noise that can be directly applied to analyse the performance of time-series InSAR techniques. In particular, we present a simple model for describing phase noise covariance due to decorrelation processes in a network of interferograms that could be exploited to develop better time-series techniques.

Key words: InSAR, SBAS, PSI, Error budget

1. Introduction

Differential synthetic aperture radar interferometry is now regularly used to generate hundred kilometer scale surface deformation maps with centimeter-to-millimeter accuracy (e.g, Rosen et al., 2000). While single interferograms have been successfully used to study large deformation events (Massonnet et al., 1993; Peltzer et al., 1994; Zebker et al., 1994; Simons et al., 2002; Pritchard and Simons, 2002), their application to studying smaller amplitude

*Corresponding Author

Email addresses: `piyush@gps.caltech.edu` (P. S. Agram), `simons@caltech.edu` (M. Simons)

surface deformation events have been hampered due to effects of temporal and geometric decorrelation (Zebker and Villasenor, 1992).

An extensive and ever-expanding archive of SAR data acquired over the last two decades and data from future SAR missions with shorter repeat periods allow us to consider the temporal evolution of surface deformation by combining information from multiple interferograms. Time-series InSAR techniques (e.g, Ferretti et al. (2001); Berardino et al. (2002); Hooper et al. (2004); Shanker and Zebker (2007); Hooper (2008); Doin et al. (2011); Hetland et al. (2011)) estimate the temporal evolution of surface deformation in areas that are characterized by reasonably large signal-to-noise ratio and are less affected by temporal and geometric decorrelation. Simple models for the effect of decorrelation phenomena (Zebker and Villasenor, 1992; Bamler and Just, 1993) and the atmospheric phase screen (Hanssen, 1998; Emardson et al., 2003; Lohman and Simons, 2005; Onn and Zebker, 2006; Knospe and Jónsson, 2010) in individual interferograms have been well studied. Atmospheric phase noise is spatially correlated (Hanssen, 2001; Emardson et al., 2003; Lohman and Simons, 2005; González and Fernández, 2011) and phase noise in interferograms with common image acquisition are also correlated (Emardson et al., 2003).

Hanssen (2001) developed a simple mathematical framework to describe most common sources of error in individual interferograms and introduced a simple functional model focusing on three-pass and four-pass differential interferometry. Guarnieri and Tebaldini (2007) and Rocca (2007) proposed similar noise models for interferogram networks and derived associated Cramer-Rao bounds on velocity estimates from time-series methods. In this work, we attempt to extend the ideas from these works and focus on building a simple covariance model for interferogram networks over space and time in order to analyse the techniques used in time-series InSAR. In particular, we show that the covariance structure of interferometric phase observations in the temporal domain is significantly more complicated than previously assumed. We derive a simple method to estimate the contribution of temporal and spatial decorrelation to the overall noise covariance, thus extending previously published models which assumed high coherence to resolution elements with moderate signal to clutter ratio.

Using our derived covariance model as a reference, we also discuss the effects of various processing steps in time-series analysis on the noise covariance structure of the interferograms. We summarize some of the typical processing steps in time-series InSAR analysis and their effect on the noise covariance in Table 1.

Table 1: Summary of typical processing steps in time-series InSAR analysis and their effect on the noise covariance of the interferograms and time-series products.

Processing step	Implications
Multi-looking of interferograms	Breaks the closure of interferometric phase over a closed circuit in the interferogram network.
Adaptive filtering of interferograms	Decreases the impact of decorrelation noise but at the cost of resolution.
Empirical stratified-troposphere corrections	Decreases bias in estimated time-series. Covariance largely unaffected.
Wavelet transforms in spatial domain.	Exploits spatial correlation. Can reduce uncertainty, only when large parts of the interferograms are coherent.

This paper is organized as follows. In Section 2, we discuss various aspects of existing approaches to modelling uncertainties in interferometric observations and point out their respective shortcomings. In Section 3, we lay out the mathematical framework that we use to describe the covariance structure of interferometric phase noise over space and time. We derive our covariance model from first principles and from single-interferogram phase models in Section 4. We finish with a discussion of implications of using the proposed covariance models in time-series InSAR techniques.

2. Previous models

When multi-looked to resolution of few hundreds of meters, as is the case for most geophysical applications, InSAR data suffers from lack of “closure over a circuit”. For example, in a network of multi-looked interferograms generated using three SAR scenes labelled A,B and C, the multi-looked interferometric phase for interferogram BC cannot be recreated exactly using the multi-looked interferometric phase observations from interferograms AB and AC. Consequently, effective reduction of a set of interferometric phase observations to geophysical parameters of interest such as relative deformation or deformation velocity requires knowledge or a model of the uncertainty associated with each phase observation.

InSAR coherence is a widely used statistical measure developed to quantify the associated uncertainty with every interferometric phase observation. Reduction of InSAR coherence, also known as decorrelation, with increasing geometric and temporal baselines or with change in surface scattering properties due to vegetation or precipitation (Zebker and Villasenor, 1992) is well documented. However, coherence is insufficient as the only quantitative estimate of noise as it fails to capture the effect of spatially correlated long wavelength noise sources like atmospheric propagation delay. Consequently, all interferogram network models to date (Hanssen, 2001; Guarnieri and Tebaldini, 2007; Rocca, 2007; González and Fernández, 2011) use a combination of decorrelation noise and an atmospheric phase term.

Hanssen (2001) advocated the use of empirical covariance functions derived from each interferogram to model the contribution of turbulent atmosphere. He also provided a framework where any generalized stochastic covariance model could be incorporated. Emardson et al. (2003) and Guarnieri and Tebaldini (2007) also derived similar expressions for the covariance of the atmospheric propagation delay taking into account the correlation of interferograms with common SAR acquisitions. Rocca (2007) modelled the atmospheric phase contribution as additive noise in each interferogram.

Hanssen (2001) assumed the decorrelation noise component to be independent for each interferogram. Guarnieri and Tebaldini (2007) and Rocca (2007) argued that if temporal decorrelation could be modeled by a brownian motion process in urban areas, the temporal decorrelation noise terms need to be temporally correlated. We build on this idea and further show that decorrelation noise for a given pixel could be correlated in interferograms even if they are not composed from common SAR acquisitions, as this term represents the effect of changing scattering behavior of the pixel over time and imaging geometry. The covariance model presented in this work builds on ideas from all these interferogram network models and generalizes many aspects of modelling the contributions from various noise sources.

3. Mathematical Notation

We consider a set of N synthetic aperture radar (SAR) images acquired using a similar geometry at time epochs $(t_1, \cdots, t_N)$. A network of M co-registered and unwrapped interferograms is generated using the ensemble of SAR scenes. We also assume that there are P pixels for which we estimate a deformation time-series. We also assume, without loss of generality, that the all SAR acquisitions are part of at least one interferogram in a network of interferograms (i.e, $M \geq \frac{N}{2}$). Similar to Hanssen (2001) and Guarnieri and Tebaldini (2007), we model the individual phase terms and not the complex interferogram for tractability. The unwrapped phase ($\Delta\phi_{ifg}$) of a pixel x, in an interferogram with master acquisition index i and slave acquisition index j can be represented as

$$\begin{aligned} \Delta\phi_{ifg}^{x,i,j} &= \Delta\phi_{defo}^{x,i,j} + \Delta\phi_{aps}^{x,i,j} + \Delta\phi_{decor}^{x,i,j} + \phi_{n}^{x,i,j} \\ & \quad i, j \in [1, \cdots, N] \ \text{ and } \ x \in [1, \cdots, P] \end{aligned} \tag{1}$$

where $\Delta\phi_{defo}$ represents the phase contribution due to the cumulative surface deformation in the time spanning the SAR acquisitions, $\Delta\phi_{aps}$ represents the phase contribution due to the difference in propagation delay through the atmosphere between SAR acquisitions, $\Delta\phi_{decor}$ represents the phase noise due to change in surface scattering properties of the resolution element and ϕ_n represents the phase contribution from all other uncorrelated noise sources. Phase ramps introduced by orbital errors can be reasonably estimated using a best fitting planar or bilinear model (e.g, Pollitz et al., 2000; Simons et al., 2002) and any residual ramps cannot be distinguished from the atmospheric phase screen (Hooper et al., 2007). In any case, an orbit error term can be included in the physical modelling process. Hence, we do not include an orbital ramp term in Equation 1. We also note that a DEM error term (Hanssen, 2001; Berardino et al., 2002) represents a systematic effect in our formulation and that the range to the scattering centres of the resolution elements are precisely known. We model the phase contribution due to orbit errors and DEM errors as deterministic terms, and assume their contribution to the noise covariance model to be negligible. We will discuss the effects of these terms again in Section 5.

Comparing with the model in Hanssen (2001), Equation 1 does not include an integer ambiguity term as we assume that the wrapped phase observations can be unwrapped accurately. For realistic modelling of phase unwrapping errors, the spatial distribution of the coherent scatterers and the gradients in time and space of the deformation signal needs to be taken into account. The former is terrain-dependent and estimating the latter is the goal of our time-series technique. Hence, without complicating our model, we assume that we have a reasonably dense network of coherent scatterers and that the SAR images are acquired sufficiently often over an area that is deforming at a reasonable rate, allowing the interferogram network to be unwrapped consistently and accurately in space and time. This assumption may not be valid under all circumstances but is needed at present to allow our model to be mathematically tractable (Guarnieri and Tebaldini, 2007; Rocca, 2007). We discuss the implications of ignoring phase unwrapping errors in Section 5.

In typical InSAR time-series algorithms, the deformation phase ($\Delta\phi_{defo}$) represents the primary signal of interest and is parametrized as a combination of individual SAR phases depending on the connectivity of the interferogram network (Berardino et al., 2002) or using a pre-determined dictionary of temporal functions (Hetland et al., 2011) . All other phase terms in Equation 1 are considered to be nuisance or noise terms and contribute to our model of the covariance matrix of the interferometric phase noise. Following Berardino et al. (2002), we formulate our time-series inversion problem for all the P pixels in our network as

$$\begin{bmatrix} \Delta\phi_{ifg}^{1,i_1,j_1} \\ \vdots \\ \Delta\phi_{ifg}^{1,i_M,j_M} \\ \vdots \\ \Delta\phi_{ifg}^{P,i_1,j_1} \\ \vdots \\ \Delta\phi_{ifg}^{P,i_M,j_M} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \cdots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \cdots & \mathbf{A} \end{bmatrix} \cdot \begin{bmatrix} \phi_{SAR}^{1,1} \\ \vdots \\ \phi_{SAR}^{1,N} \\ \vdots \\ \phi_{SAR}^{P,1} \\ \vdots \\ \phi_{SAR}^{P,N} \end{bmatrix} + \begin{bmatrix} \phi_{n}^{1,i_1,j_1} \\ \vdots \\ \phi_{n}^{1,i_M,j_M} \\ \vdots \\ \phi_{n}^{P,i_1,j_1} \\ \vdots \\ \phi_{n}^{P,i_M,j_M} \end{bmatrix} \tag{2}$$

where $\phi_{SAR}^{x,i}$ represents the unwrapped SAR phase contribution from each SAR scene to an interferogram (with respect to the mean SAR phase), $\mathbf{A}$ represents the interferogram network incidence matrix ($M \times N$) and (i_k, j_k) represent the master and slave scene indices for interferogram k. We note that the formulation in Equation 2 is inclusive of the Persistent Scatterer (PS) problem (e.g, Ferretti et al., 2001) which corresponds to the specific case where only a common-master interferogram network is considered. We also note that Equation 2 is a simplified form of the generic functional model described in Section 3.1.3 of Hanssen (2001) for a network of interferograms.

We represent the unwrapped SAR phase contribution at pixel x in SAR acquisition i as the sum of deformation, atmospheric phase screen and decorrelation components.

$$\phi_{sar}^{x,i} = \phi_{defo}^{x,i} + \phi_{aps}^{x,i} + \phi_{decor}^{x,i} \tag{3}$$

Subsequently, we rewrite Equation 2 as

$$\begin{aligned}
\begin{bmatrix} \Delta\phi_{ifg}^{1,i_1,j_1} \\ \vdots \\ \Delta\phi_{ifg}^{1,i_M,j_M} \\ \vdots \\ \Delta\phi_{ifg}^{P,i_1,j_1} \\ \vdots \\ \Delta\phi^{P,i_M,j_M} \end{bmatrix} &= \overline{\mathbf{A}}\cdot \begin{bmatrix} \phi_{defo}^{1,1} \\ \vdots \\ \phi_{defo}^{1,M} \\ \vdots \\ \phi_{defo}^{P,1} \\ \vdots \\ \phi_{defo}^{P,M} \end{bmatrix} + \overline{\mathbf{A}}\cdot \begin{bmatrix} \phi_{aps}^{1,1} \\ \vdots \\ \phi_{aps}^{1,M} \\ \vdots \\ \phi_{aps}^{P,1} \\ \vdots \\ \phi_{aps}^{P,M} \end{bmatrix} \\
&+ \overline{\mathbf{A}}\cdot \begin{bmatrix} \phi_{decor}^{1,1} \\ \vdots \\ \phi_{decor}^{1,M} \\ \vdots \\ \phi_{decor}^{P,1} \\ \vdots \\ \phi_{decor}^{P,M} \end{bmatrix} + \begin{bmatrix} \phi_{n}^{1,i_1,j_1} \\ \vdots \\ \phi_{n}^{1,i_M,j_M} \\ \vdots \\ \phi_{n}^{P,i_1,j_1} \\ \vdots \\ \phi_{n}^{P,i_M,j_M} \end{bmatrix}
\end{aligned} \tag{4}$$

where $\overline{\mathbf{A}} = \mathbf{I}_{P,P} \otimes \mathbf{A}$ represents the block diagonal matrix resulting from the kronecker delta product, $\otimes$, of an identity matrix of size P, $\mathbf{I}_{P,P}$, and the network incidence matrix $\mathbf{A}$. ϕ_{defo} represents the SAR phase contribution due to surface deformation, ϕ_{aps} represents the phase contribution from the atmospheric phase screen and ϕ_{decor} represents the phase noise due to decorrelation in the SAR phase of each of the acquisitions. The SAR phase terms in Equation 4 represent deviations from the pixel-wise averages and not the absolute phase contributions. The representation of phase terms other than deformation as zero-mean random variables is reasonable as the common bias terms for each of these cancel out when computing the interferometric phase. Following (Zebker and Villasenor, 1992), we vectorize Equation 4 in the absence of surface deformation and rewrite phase noise in an interferometric network as

$$\overline{\Delta\phi_{ifg}} = \overline{\mathbf{A}}\cdot \overline{\phi_{aps}} + \overline{\mathbf{A}}\cdot \overline{\phi_{decor}} + \overline{\phi_n} \quad . \tag{5}$$

We tabulate all the mathematical symbols and notation used here in Appendix A.

4. Covariance model

All the terms in Equation 5 have distinct spatio-temporal characteristics (Hooper, 2006) that allows us to estimate their relative contributions from a set of observed interferometric phases. The atmospheric phase screen (ϕ_{aps}) in every SAR scene (not just interferograms) is correlated over space but uncorrelated in time for scenes with time-separation longer than approximately a day (Emardson et al., 2003). Phase noise due to the scatterers (ϕ_{decor}) in a resolution element is correlated for interferograms with common SAR acquisitions but is uncorrelated spatially. ϕ_n represents the combined contribution of all uncorrelated noise sources in each interferogram and is uncorrelated over space and time. The phase components in Equation 5 are statistically independent as they represent unrelated physical processes. Consequently, the total covariance matrix (for all pixels in all the interferograms), $\mathbf{\Sigma}_{ifg}$, as

$$\mathbf{\Sigma}_{ifg} = \mathbf{\Sigma}_{aps} + \mathbf{\Sigma}_{decor} + \mathbf{\Sigma}_{n} \quad . \tag{6}$$

Equation 6 is the same stochastic model presented in Hanssen (2001). Hanssen (2001) referred to the atmospheric phase screen component as C_s and combined $\mathbf{\Sigma}_{decor}$ and $\mathbf{\Sigma}_n$ into C_φ, as he did not distinguish between decorrelation phase noise and processing errors. C_s and C_φ represent the spatially correlated path delay component and the scatterer noise components of the interferometric phase error, respectively.

4.1. Atmospheric Phase Screen

The dominant contribution to the path delay component of phase error in many interferograms is from the spatial heterogeneity of the wet component of atmospheric refractivity, resulting in excess path length of the radar signal propagating through the neutral atmosphere (Goldstein, 1995; Emardson et al., 2003; Onn and Zebker, 2006). The atmospheric phase signal varies gradually over space and is often modelled as a long wavelength component in unwrapped phase (Onn and Zebker, 2006; Hooper, 2006). The spatial covariance function for interferometric phase has been studied in detail (Hanssen, 2001; Lohman and Simons, 2005; Knospe and Jónsson, 2010) and can be derived from the data itself. The structure function can be easily derived for coherent interferograms with short temporal baselines that are not affected by significant deformation. However, their method cannot be applied to all types of terrain and interferograms affected by large deformation events.

Alternately, Emardson et al. (2003) used GPS data to empirically determine the average delay structure function due to the troposphere as

$$\sigma_{aps}^{x,y} = c \cdot L_{x,y}^{\alpha} + k \cdot H_{x,y} \quad . \tag{7}$$

where $\sigma_{x,y}^2$ is the variance of the difference in SAR atmospheric phase between pixels x and y, $L_{x,y}$ represents the distance between the pixels and $H_{x,y}$ represents the difference in altitude between the pixels. The structure function (Equation 7) can be directly related to the covariance function ($\eta_{aps}^{x,y}$) of atmospheric phase of two pixels x and y as

$$\eta_{aps}^{x,y} = \frac{1}{2} \cdot \left[\left(\sigma_{aps}^{x,ref}\right)^2 + \left(\sigma_{aps}^{y,ref}\right)^2 - \left(\sigma_{aps}^{x,y}\right)^2 \right] \quad . \tag{8}$$

If the interferograms were calibrated using a set of independent geodetic observations like a GPS network and without an explicit reference region, we would use the covariance model directly derived using techniques suggested by Hanssen (2001) and Lohman and Simons (2005) to compute $\eta_{aps}^{x,y}$ in Equation 8.

Given a functional form for $\eta_{aps}^{x,y}$, the covariance matrix for the atmospheric phase components in an individual SAR acquisition can then be written as

$$\mathbf{\Sigma}_{aps}^{sar} = \begin{bmatrix} \eta_{aps}^{1,1} & \cdots & \eta_{aps}^{1,P} \\ \vdots & \eta_{aps}^{x,y} & \vdots \\ \eta_{aps}^{P,1} & \cdots & \eta_{aps}^{P,P} \end{bmatrix} \tag{9}$$

where Σ_{aps}^{sar} is of size $P \times P$. Assuming that the atmospheric conditions are temporally uncorrelated and the spatial covariance structure remains the same in any SAR acquisition, the atmospheric phase covariance matrix for the entire network can be written as

$$\mathbf{\Sigma}_{aps} = \overline{\mathbf{A}} \cdot \left[\mathbf{\Sigma}_{aps}^{sar} \otimes \mathbf{I}_{P,P} \right] \cdot \overline{\mathbf{A}}^T \quad . \tag{10}$$

This equation can be suitably modified if the parameters governing the functional covariance model (Equation 8) for each SAR acquisition are known. Alternatively the interferograms themselves can be directly analyzed to estimate the covariance function ($\eta_{aps}^{x,y}$) using a network approach (Biggs et al., 2007; González and Fernández, 2011). The derivation of Equation 10 is similar to the one presented in Emardson et al. (2003) and has been extended to include all the coherent pixels simultaneously in order to exploit the spatially correlated nature of the atmospheric phase screen.

InSAR phase measurements are typically correlated over a scale of a kilometer or two (Hanssen, 2001; Emardson et al., 2003; Lohman and Simons, 2005). Consequently, $\mathbf{\Sigma}_{aps}$ matrix is made up of large number of non-zero elements and cannot be efficiently represented as a sparse matrix. Our atmospheric phase covariance is consistent with the model suggested by Hanssen (2001).

4.1.1. Ionospheric effects

In deriving our covariance model for the atmospheric phase screen, we have currently neglected path delay introduced by ionospheric heterogeneities which can behave significantly differently compared to the troposphere (Chapin et al., 2006; Meyer, 2010; Meyer and Watkins, 2011). A fundamental difference is that the ionospheric contributions are strongly dependent on the sensor wavelength, whereas the tropospheric delay component is almost independent of wavelength. In the future, we expect to be able to mitigate ionospheric effects to a large extent using a multi-frequency approach (Meyer, 2010; Rosen et al., 2010).

4.2. Decorrelation

The achievable accuracy of any SAR interferogram is affected by temporal decorrelation caused by change in surface scattering properties over time, geometric decorrelation or spectral misalignment of received echoes due to different imaging angles and radar receiver noise (Zebker and Villasenor, 1992; Bamler and Just, 1993). The amount of phase noise affecting the interferometric phase measurement at pixel x in an interferogram composed of SAR acquisitions with indices i and j is commonly characterized by its coherence ($\gamma^{x,i,j}$) defined as

$$\gamma^{x,i,j} = \frac{\left\| \mathrm{E}\left(z_{x,i} \cdot z_{x,j}^{*}\right) \right\|}{\sqrt{\mathrm{E}\left(\|z_{x,i}\|^2\right) \cdot \mathrm{E}\left(\|z_{x,j}\|^2\right)}} \tag{11}$$

where $z_{x,i}$ and $z_{x,j}$ represent the complex return for pixel x in SAR acquisitions with indices i and j, and $\mathrm{E}(\cdot)$ represents the expectation function. A coherence value of one indicates noise-free observations whereas a value of zero indicates pure noise observations. The coherence as estimated using Equation 11 has been shown to be biased towards higher values (Touzi et al.,

1990) and often needs to be corrected before use for practical applications. Assuming gaussian scatterers the coherence values can be related to interferometric phase standard deviation using the Cramer-Rao bound relation (Rodriguez and Martin, 1992)

$$\gamma^{x,i,j} = \frac{1}{\sqrt{1 + 2 \cdot \mathrm{L} \cdot {\sigma_{\Delta\phi}^{x,i,j}}^2}} \tag{12}$$

where L represents the number of looks used to estimate the coherence and $\sigma_{\Delta\phi}$ is the associated interferometric phase standard deviation. We prefer to use the observed phase standard deviation to characterize phase noise directly as it can be directly used for building covariance matrices. To reduce the effect of gradients introduced by deformation or orbital errors, the observed phase values are corrected for a local slope component over the estimation window before the standard deviations are estimated (Zebker and Chen, 2005). The coherence estimate is assumed to be independent of the atmospheric phase screen as the physical scale of the atmospheric signal (1-2km) is much larger than the estimation window (100-200 m).

Hanssen (2001) preferred to model decorrelation noise (ϕ_{decor}) terms using spatially and temporally uncorrelated random variables. This assumption also lies in the heart of short baseline techniques e.g, Berardino et al. (2002). However, not all components of decorrelation noise are uncorrelated. We illustrate this with an example.

Assume we use a set of interferograms generated from four SAR images (labelled A, B, C and D) representing a set of tandem multi-baseline acquisitions as shown in Figure 1. For resolution cells characterized by distributed scatterers, Zebker and Villasenor (1992) showed that coherence for any pair is a function of the separation between the antenna centers of the receivers and is not affected by temporal decorrelation. The total spatial decorrelation affecting the pair BC also affects the pair AD, though they do not share any common SAR acquisitions. This simple example illustrates that the phase noise terms affecting the interferograms BC and AD are correlated. The covariance in the decorrelation phase for two interferograms is given by

$$\mathrm{cov}\left(\phi_{ij}, \phi_{kl}\right) = \sigma^2_{\phi_{pq}} \cdot I\left(B_{\perp,pq}\right) \quad \text{where} \quad B_{\perp,pq} = B_{\perp,ij} \cap B_{\perp,kl} \tag{13}$$

where i, j, k, l represent SAR scene indices, $B_{\perp,pq}$ represents the geometric overlap in the baselines of pairs (i, j) and (k, l), $B_{\perp,ij}$ represents the perpendicular baseline of interferogram (i, j) with respect to a single master

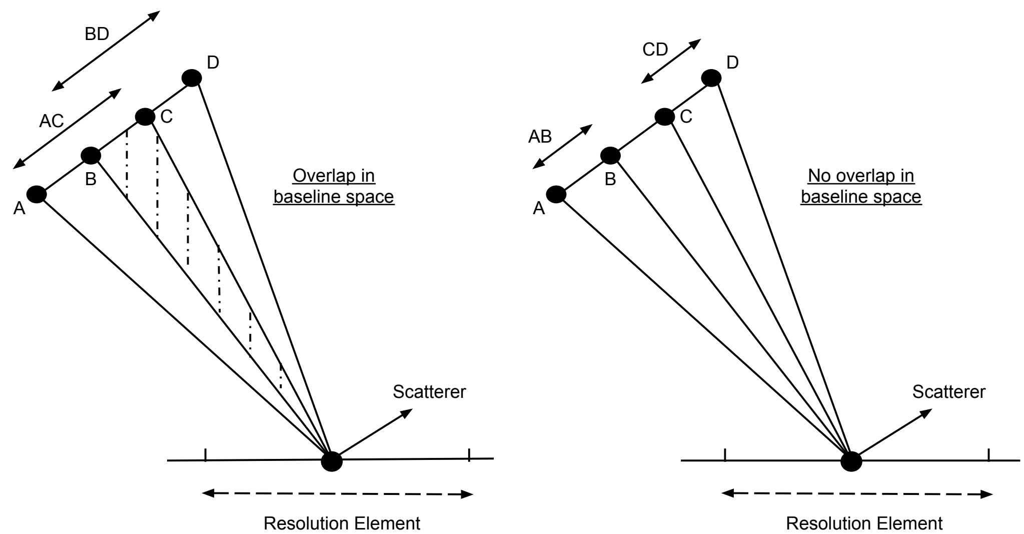


Figure 1: Example geometry of multi-baseline tandem SAR acquisitions - A, B, C and D represent the receiving antenna centers. We use this example configuration to argue that noise in interferograms AC and BD are correlated even though they do not share any common acquisitions, due to the overlap in baseline space. Interferograms AB and CD would not share common decorrelation noise terms as they do not overlap in baseline space.

acquisition, σ_{ij}^2 represents the phase noise variance of pair (i, j) and $I(\cdot)$ represents the indicator function which is one if the baselines overlap or zero otherwise. The covariance matrix corresponding to our example in Figure 1 as given by Equation 13 is

$$\begin{aligned}
\overline{\phi}^T &= [\phi_{AB}\ \phi_{AC}\ \phi_{AD}\ \phi_{BC}\ \phi_{BD}\ \phi_{CD}]^T \\
\mathbf{\Sigma}_{decor} &= \mathrm{E}\left[\overline{\phi}\cdot\overline{\phi}^T\right] \\
&= \begin{bmatrix}
\sigma_{AB}^2 & \sigma_{AB}^2 & \sigma_{AB}^2 & 0 & 0 & 0 \\
\sigma_{AB}^2 & \sigma_{AC}^2 & \sigma_{AC}^2 & \sigma_{BC}^2 & \sigma_{BC}^2 & 0 \\
\sigma_{AB}^2 & \sigma_{AC}^2 & \sigma_{AD}^2 & \sigma_{BC}^2 & \sigma_{BD}^2 & \sigma_{CD}^2 \\
0 & \sigma_{BC}^2 & \sigma_{BC}^2 & \sigma_{BC}^2 & \sigma_{BC}^2 & 0 \\
0 & 0 & \sigma_{BD}^2 & \sigma_{BC}^2 & \sigma_{BD}^2 & \sigma_{CD}^2 \\
0 & 0 & \sigma_{CD}^2 & 0 & \sigma_{CD}^2 & \sigma_{CD}^2
\end{bmatrix}
\end{aligned} \tag{14}$$

A similar argument can be made for the temporal decorrelation noise term when four SAR scenes (A, B, C and D) are acquired from the same point in space a few days apart such that they are not affected by spatial decorrelation. Assuming a Brownian motion model (Zebker and Villasenor, 1992; Rocca, 2007), the temporal decorrelation noise affecting the interferometric pair BC also affects the interferometric pair AD as they share a common time interval. Consequently, the decorrelation noise for these two pairs are correlated even though they share no common SAR scenes. For this example, the covariance matrix for the temporal decorrelation noise has the same structure as Equation 14. In this section, we derive a model for phase noise covariance that is consistent with observed baseline and temporal decorrelation effects.

Hanssen (2001) used the Gaussian signal model to describe the interferometric phase variance as a function of the observed interferometric coherence. However, the fact that high interferometric coherence also suggests that the decorrelation phase terms for the SAR acquisitions are similar or highly correlated was not highlighted. Subsequently, it was assumed that the decorrelation phase terms were independent in all the interferograms. We have already argued that the SAR decorrelation phase terms must be correlated to be consistent with observed geometry/temporal baseline effects, i.e, the scatterer noise increases with increasing baseline and time-separation between acquisitions (Zebker and Villasenor, 1992). We show that it is possible to derive an approximate covariance matrix that is consistent with geometric and temporal baseline effects in three steps:

1. We first populate a SAR decorrelation phase correlation matrix ($\mathbf{\Omega}_{sar}^{x}$) for a single pixel x using a coherence model as follows

$$\mathbf{\Omega}_{sar}^{x} = \begin{bmatrix} 1 & \gamma^{x,1,2} & \cdots \\ \gamma^{x,i,j} & 1 & \cdots \\ \vdots & \vdots & 1 \end{bmatrix} \tag{15}$$

   Zebker and Villasenor (1992) suggested the following form for the coherence model

$$\gamma^{x,i,j} = \gamma_{spatial} \cdot \gamma_{temporal} \cdot \gamma_{thermal} \tag{16}$$

   Our formulation can accommodate any general form of the coherence function (not necessarily stationary) to account for time-dependent phenomena like seasonal variation in the nature of scatterers (Lauknes et al., 2010). In general, we assume coherence to be a function of the two SAR acquisition times and the perpendicular baseline for each pixel.

$$\gamma^{x,i,j} = \zeta\left(t_i, t_j, B_{\perp,ij}\right) \tag{17}$$

   We use the singular value decomposition to generate a positive semi-definite approximation of $\mathbf{\Omega}_{sar}^{x}$ if needed.
2. Assuming that the scatterer noise levels are the same in every SAR acquisition, we transform this matrix into a pseudo-correlation matrix for interferometric phase of a single pixel x using the incidence matrix as

$$\tilde{\mathbf{\Omega}}_{ifg}^{x} = \mathbf{A} \cdot \mathbf{\Omega}_{sar}^{x} \cdot \mathbf{A}^{T} \tag{18}$$

   Theoretically, $\tilde{\mathbf{\Omega}}_{ifg}^{x}$ is not a correlation matrix as the matrix elements are not normalized to lie in the interval $[-1, 1]$.
3. We again use the pseudo-correlation estimates to scale the InSAR correlation matrix to the InSAR covariance matrix using

$$\begin{aligned} \mathbf{\Sigma}_{decor}^{x} &= \mathbf{D} \cdot \tilde{\mathbf{\Omega}}_{ifg}^{x} \cdot \mathbf{D} \\ \mathbf{D} &= \begin{bmatrix} \frac{\sigma_{\Delta\phi}^{x,i_1,j_1}}{\sqrt{\tilde{\mathbf{\Omega}}_{ifg}^{x}(1,1)}} & 0 & 0 \\ \vdots & \frac{\sigma_{\Delta\phi}^{x,i_k,j_k}}{\sqrt{\tilde{\mathbf{\Omega}}_{ifg}^{x}(k,k)}} & \vdots \\ 0 & 0 & \frac{\sigma_{\Delta\phi}^{x,i_M,j_M}}{\sqrt{\tilde{\mathbf{\Omega}}_{ifg}^{x}(M,M)}} \end{bmatrix} \end{aligned} \tag{19}$$

where $\sigma_{\Delta\phi}^{x,i_k,j_k}$ represents the InSAR phase standard deviation in the k^{th} interferogram (Equation 12) and $\tilde{\mathbf{\Omega}}_x^{ifg}(k,k)$ represents the k^{th} diagonal element of $\tilde{\mathbf{\Omega}}_x^{ifg}$. This step ensures that the diagonal terms of $\mathbf{\Sigma}_{decor}^x$ corresponds to ${\sigma_{\Delta\phi}^{x,i,j}}^2$ and hence, consistent with our coherence model.

The complete covariance matrix for all the pixels in the network can be then be written as

$$\mathbf{\Sigma}_{decor} = \begin{bmatrix} \mathbf{\Sigma}_{decor}^1 & 0 & 0 \\ \vdots & \mathbf{\Sigma}_{decor}^k & \vdots \\ 0 & 0 & \mathbf{\Sigma}_{decor}^P \end{bmatrix} \tag{20}$$

From Equation 20, it is clear that $\mathbf{\Sigma}_{decor}$ has a block diagonal structure and can be efficiently represented as a sparse matrix. This approach is only true when the resolution of the SAR system is same as the pixel spacing (Hanssen, 2001). If the pixel spacing is smaller than the imaging resolution, neighboring pixels are bound to be correlated. Also, the impulse response of a point target on the ground in a SAR image is represented by a sinc function (Cumming and Wong, 2005) and the observations from adjacent pixels may be correlated. Such an effect is particularly observed in urban areas with strongly reflecting structures (Cumming and Wong, 2005). However, we currently ignore this effect in our simple model.

$\mathbf{\Sigma}_{aps}$ term is commonly used in the covariance model when setting up parametric inversions of InSAR phase observations (Biggs et al., 2007; Hetland et al., 2011) but $\mathbf{\Sigma}_{decor}$ is often ignored primarily because:

1. Conventional time-series techniques, e.g persistent scatterers (Ferretti et al., 2001) and small baseline subset Berardino et al. (2002), operate on a subset of pixels that are coherent through out an interferogram network. $\mathbf{\Sigma}_{decor}^x$ is negligible for a pixel that is consistently coherent over a network of interferograms because the entries of matrix $\tilde{\mathbf{\Omega}}_{ifg}^x$ and $\sigma_{\Delta\phi}^{x,i,j}$, all tend to be nearly zero. We demonstrate this with a numerical example in Appendix B.
2. The interferograms are often filtered in the spatial domain before analysis for many geophysical applications that focus on estimating relative deformation at a scale of few kilometers and are willing to compromise on the finer details on the scale of few meters (Goldstein and Werner, 1998; Baran et al., 2003). The adaptive filtering process, artificially

boosts the coherence of the interferometric phase observations by reducing the resolution and the associated interferometric phase variance. An example of this artificial boosting of coherence is shown for a C-band interferogram in Figure 2. Use of filtered interferometric products presents a strong case for using observed interferometric phase variance for quantifying noise as opposed to traditional InSAR coherence (Equation 11).

Modeling and using $\mathbf{\Sigma}_{decor}$ is most useful in the case of estimating deformation parameters from partially coherent scatterers (Perissin and Wang, 2011), which is beyond the scope of this manuscript. $\mathbf{\Sigma}_{decor}$ represents a natural weighting matrix that values coherent InSAR phase observations more than the noisy observations (see Appendix B for an example). Our model of decorrelation noise motivates the beneficial effect of filtering interferograms for time-series InSAR studies over large areas.

4.3. Uncorrelated noise

$\overline{\phi_n}$ represents the vector of uncorrelated phase noise terms affecting interferometric phase measurements at all pixels and in all the interferograms of the network (Equation 5). Ideally from an information theory point of view, a given redundant interferogram network contains the same amount of information as any other connected interferogram network, as the corresponding incidence matrices $\left(\mathbf{A} \text{ or } \overline{\mathbf{A}}\right)$ have the same rank.

However, in practice, the same interferogram produced in different master geometries differ by a small random phase component that is spatially uncorrelated. Same interferograms produced using different InSAR processors also differ by a small random phase component. Hence, we include the uncorrelated noise term in our interferometric phase model (Equation 5). Hanssen (2001) attributed this error term to coregistration and interpolation errors during processing, and assumed it to be uncorrelated in space and time.

$\mathbf{\Sigma}_n$ in Equation 6, thus has a diagonal matrix structure in our model. The magnitudes of the diagonal entries in this work have been determined empirically by comparing the phase difference between an interferogram and another obtained by switching the slave and master. Table 4.3 tabulates the phase noise values for various interferograms computed for a set of ALOS PALSAR interferograms over Parkfield, CA as estimated using the Stanford University mocomp processor (Zebker et al., 2010).

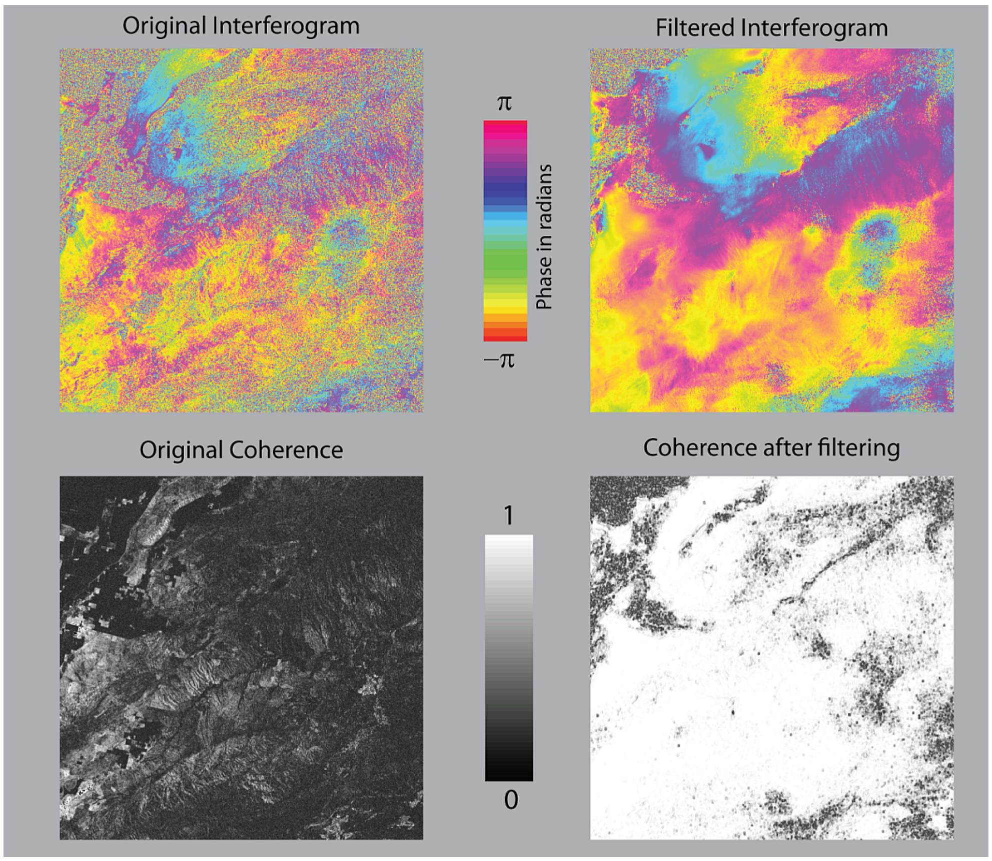


Figure 2: Example of improvement in perceived coherence due to adaptive filtering of interferograms. The data corresponds to pair of Envisat SAR scenes (Track 256, Frame 2889) acquired on Feb 24, 2006 and Feb 29, 2008. The interferogram was filtered using a Goldstein filter (Goldstein and Werner, 1998) of strength 0.6. The estimated phase variance was transformed to equivalent coherence assuming Gaussian scatterers (Equation 12).

Table 2: Standard deviation in radians of uncorrelated phase noise sources in an interferogram for a set of L-band ALOS PALSAR interferograms (Path 220, Frame 710) over Parkfield, CA.

Master date (yyyymmdd)	Slave date (yyyymmdd)	Bperp (m)	Phase noise (in radians)
20070909	20071025	410	0.22
20071025	20071210	98	0.16
20071227	20080211	932	0.47
20080211	20080328	102	0.19
20081113	20081229	39	0.45
20090614	20090730	159	0.43

4.4. Properties of $\mathbf{\Sigma}_{ifg}$

Following are the characteristic properties of $\mathbf{\Sigma}_{ifg}$:

- $\mathbf{\Sigma}_{ifg}$ cannot be efficiently represented as a sparse matrix, primarily due to the non-sparse structure of $\mathbf{\Sigma}_{aps}$.
- $\mathbf{\Sigma}_{aps}$ (Section 4.1) and $\mathbf{\Sigma}_{decor}$ (Section 4) are not full rank for a connected network of interferograms since, they are computed using linear transformations involving matrices $\mathbf{A}$ or $\overline{\mathbf{A}}$ which are not full rank. However, the diagonal structure of $\mathbf{\Sigma}_n$ (Section 4.3) ensures that the total covariance matrix $\mathbf{\Sigma}_{ifg}$ is full rank and hence, invertible for any connected subset of all possible interferograms.
- Like any general covariance matrix, $\mathbf{\Sigma}_{ifg}$ is symmetric. Its full rank property also ensures that it is positive-definite. These properties allow us to design a simple and efficient method to efficiently prune and augment interferogram networks (Agram and Simons, 2012).

5. Discussion

In Section 3 and Section 4, we described our proposed noise covariance model in detail. In this section, we discuss the implications of the various aspects of our noise-covariance model on InSAR time-series estimates.

5.1. Exploiting spatial correlation

We described a simple model in Section 4 to derive the noise covariance matrix over space and time for interferometric phase observations. The most important aspect of our noise covariance model is that phase noise is correlated over the spatial domain in interferograms. Consequently, our understanding of deformation at the pixel scale can be significantly improved if we take into account information from neighbouring pixels. Conventional time-series InSAR approaches, both PS and SB, rely on individual pixel-based inversion techniques for computational tractability. As a result, the estimated parameters are also affected by noise with the same spatial structure as the one derived in our covariance model.

Recently developed methods, like the Multi-scale Interferometric Time-Series (MInTS) technique (Hetland et al., 2011), attempt to exploit the spatially correlated nature of the atmospheric signal. The interferometric observations are transformed into wavelet space, where the coefficients are less correlated in space and over scale. Consequently inversion of wavelet coefficients various spatial scales is more effective compared to the direct inversion of interferometric observations themselves. The MInTS approach results in the diagonalization of the covariance matrix over the spatial domain resulting in significant reduction of spatially correlated error terms in the estimated time-series parameters while still allowing for an reasonably efficient implementation of parameter estimation for large data sets.

We illustrate the strength of wavelets using an example network of 80 ERS and Envisat interferograms (Track 27, Frame 2871, descending geometry) acquired over the creeping section of the San Andreas fault in Central California and spanning the time-period from November 1992 to July 2004 (Figure 3).

We used a threshold of 0.25 on the coherence value of pixels in each interferogram and restricted our analysis to pixels that were coherent in all interferograms. The deformation was modelled as a combination of a constant velocity term and sinusoidal terms with a time period of one year. First, we applied a parameterized inversion in the temporal domain (Hetland et al., 2011), hereby referred to as Timefn, technique to estimate the temporal model parameters on a pixel-by-pixel basis. We also applied the MInTS approach, i.e parameterized inversion in the temporal domain of the wavelet coefficients of the interferograms, on the same dataset. To reduce the impact of the decorrelation noise on our results, we used the same set of filtered interferograms and coherent pixels for our MInTS and Timefn analysis. We

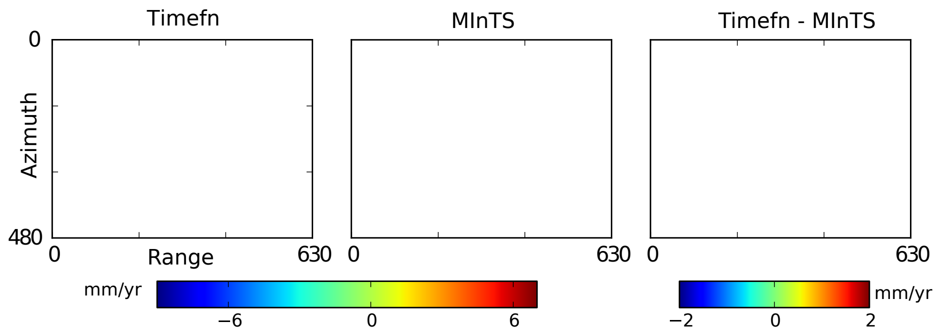


Figure 3: LOS velocity map derived using the Timefn (left) and MInTS (Middle) techniques. A temporal model consisting of a constant velocity term and sinusoidal terms with a period of one year was used to invert the dataset. The difference between the datasets is also shown (Right). The standard deviation of the LOS velocity over the entire image is approximately 1.8 mm/yr.

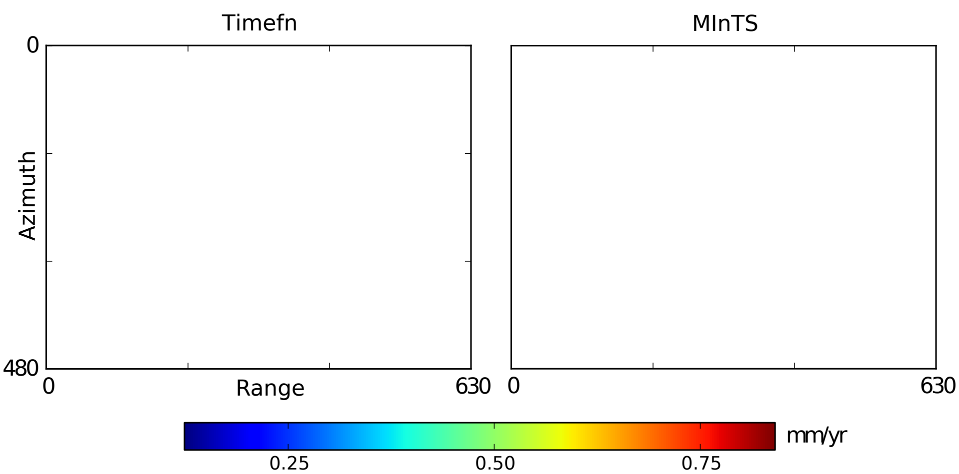


Figure 4: Estimated uncertainties in the LOS velocity map derived by applying a jackknife technique based on SAR acquisitions for Timefn (left) and MInTS (Right) techniques. The estimated variance of the LOS velocity estimates is reduced by 4 dB on average when spatial correlation between deformation is taken into account (Figure 5).

also used a temporal covariance model for inverting the data in the temporal domain in both the approaches (Emardson et al., 2003). The spatial covariance model for the wavelet coefficients in the MInTS processing is beyond the scope of this paper. To estimate the uncertainty in our time-series estimates, we used a jackknife statistical approach. Sub-networks were constructed from the original interferogram network by excluding one SAR scene at a time, and the linear velocity and seasonal sinusoidal terms were re-estimated for each of the sub-networks. The standard deviation of the estimated time-series of the sub-networks represents the corresponding uncertainty. All data analysis was performed using the Generic InSAR Analysis Toolbox (GIAnT) (Agram et al., Submitted), available for free at `http://earthdef.caltech.edu`. Figure 3 shows that both techniques estimate similar LOS velocity fields with few differences.

Figure 5 shows the ratio of the estimated uncertainty associated with the estimated time-series for our network of C-band interferograms using Timefn and MInTS on a logarithmic scale. Figure 5 shows that accounting for the spatially correlated nature of the atmospheric signal decreases the uncertainty in estimated time-series by roughly 4dB. However, we do note that such gains are only possible when large sections of the interferograms are coherent.

5.2. Empirical corrections

Another technique that is gaining popularity in time-series InSAR studies is the use of GPS wet delay and meteorological data sets to estimate the atmospheric phase screen and correction of interferograms before time-series analysis (e.g., Delacourt et al., 1998; Onn and Zebker, 2006; Foster et al., 2006; Cavalié et al., 2007; Jolivet et al., 2011). Auxiliary datasets like GPS or meterological models can be used to correct biases introduced by the stratified troposphere and the quality of these corrections are heavily dependent on the spatial resolution of the auxiliary data sets used. Figure 6 shows an ERS interferogram over Parkfield, CA that was corrected using the North American Regional Reanalysis (Mesinger et al., 2006) weather model and PyAPS software (Jolivet et al., 2011; Agram et al., Submitted). We also show the covariance function estimated before and after correction as estimated from the data itself (Jónsson, 2002; Lohman and Simons, 2005) to emphasize that auxiliary data does not account for effects of turbulence which contributes most to the covariance estimates (Hanssen, 1998). If the corrections due to the atmospheric model are not exact, it increases the covariance between pixels at larger distances as seen in Figure 6. The increased covariance at large distances is due to the fact that the phase corrections over the entire scene are derived from a fine set of meteorological grid points. Hence, a

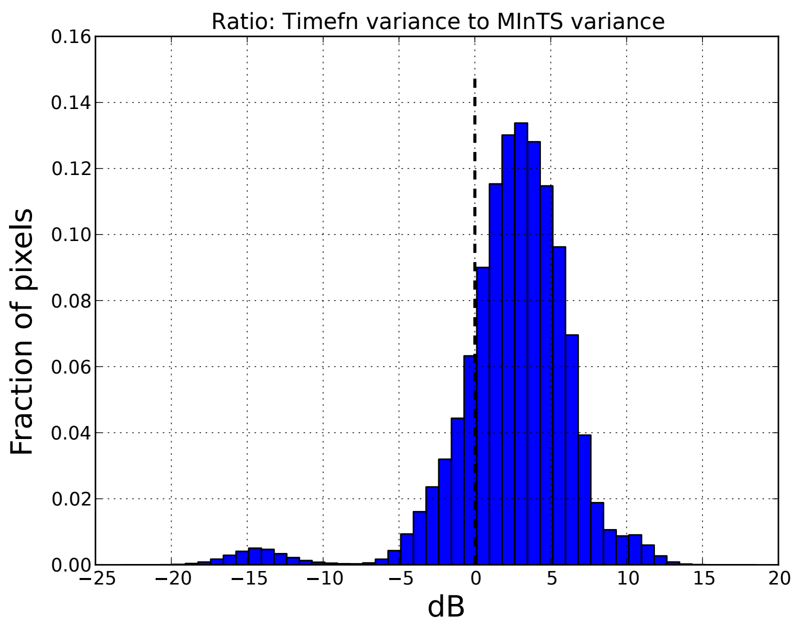


Figure 5: Ratio of the estimated uncertainties in LOS time-series using the SBAS and MInTS techniques for all pixels and all time-epochs. A jackknife approach based on the SAR acquisitions was used to determine the uncertainties.

MInTS-like approach (Hetland et al., 2011) would still be needed to diagonalize the covariance structure. Our observation of increased covariance over large distances also holds true for the case when the stratified tropospheric delay corrections (Lin et al., 2010; Lauknes et al., 2010) or the orbital ramp functions are determined empirically from the data itself. Using a wrong set of coefficients for either type of empirical corrections, introduces correlation between phase observations separated by large distances.

5.3. Decorrelation noise

Emardson et al. (2003) pointed out that the atmospheric phase screen will contribute to correlated noise between pairs of interferograms with common SAR images. We further argue that the scatterer noise terms could be correlated across interferograms that span some common time interval for consistency with single interferogram decorrelation models (Zebker and Villasenor, 1992). We described a simple way to derive a model the covariance matrix for scatterer noise in Section 4.2. Incorporating the decorrelation noise covariance can potentially allow us to develop better time-series algorithms that exploit partially coherent scatterers. Our model for decorrelation covariance also allows us to formally explain the motivation behind using short baseline interferograms for time-series analysis (Berardino et al., 2002). Recent work by Hooper (2008) and Ferretti et al. (2011), further illustrates the need for using networks with short temporal and spatial baseline interferograms to improve time-series estimates for deformation studies. As explained in the example in Appendix B, the contribution of the scatter noise term is negligible for a set of coherent observations.

5.4. DEM error

In deriving our covariance model, we have assumed that the contribution of the DEM error term is a systematic effect and can be estimated from the data itself for tractability. However, this is not always true. In case of a sensor that exhibits a systematic drift in baseline with time, like ALOS PALSAR, the perpendicular baselines are correlated with the temporal baseline causing leakage between the velocity and DEM error estimates (Samsonov, 2010). DEM error term also has an important effect when estimating time-series using partially coherent scatterers (Doin et al., 2011; Perissin and Wang, 2011). The correlation between the temporal baseline and the perpendicular baseline vectors, an indicator of trade-off between the parameters, potentially changes for each pixel. Consequently, the uncertainty

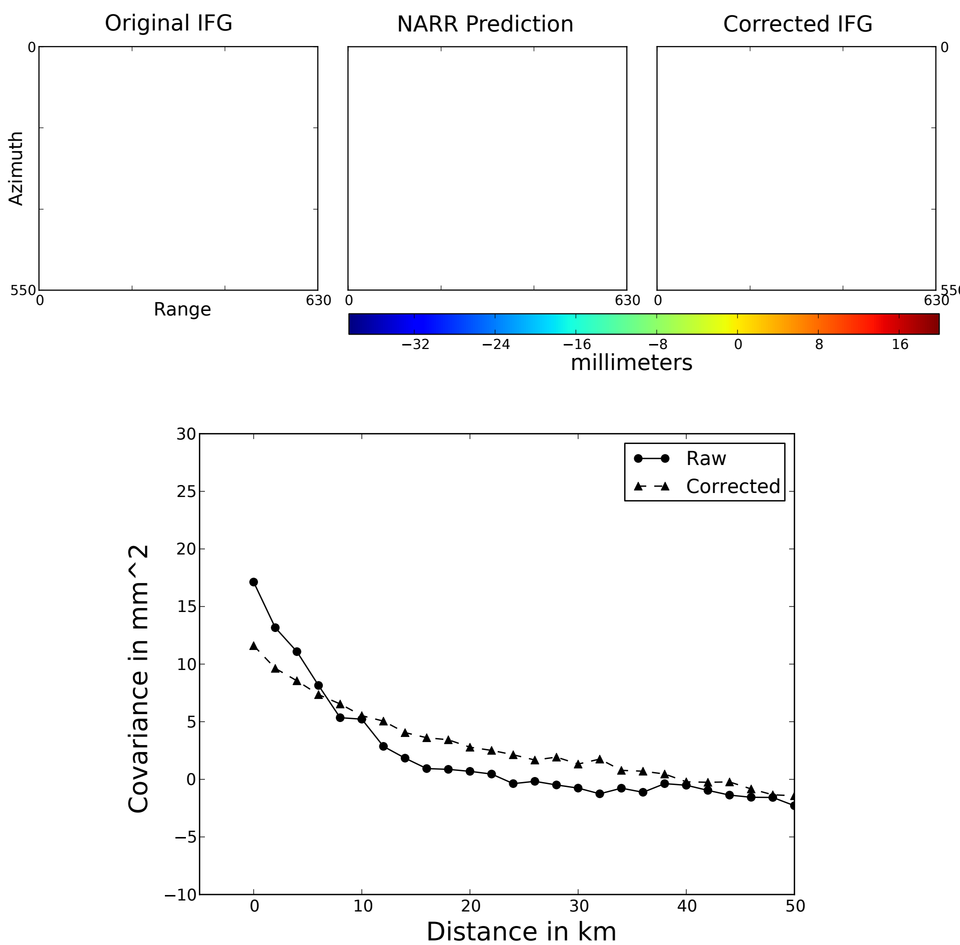


Figure 6: Comparison of the spatial phase covariance structure before and after correction using a NARR reanalysis based estimate of differential path delay due to temporal variations in the stratification of the atmosphere (Jolivet et al., 2011) over Parkfield, CA for an ERS interferogram spanning 1993-10-26 to 1993-11-30. Deformation in this time-period is assumed to be negligible. The interferogram was analyzed at a spatial resolution of 200 m.

and the covariances associated with the inferred time-series parameters will have different characteristics.

5.5. Phase unwrapping

From a purely statistical point of view, using all available interferograms with the correct covariance information should decrease the uncertainty in our deformation estimates. In practice, however, we are restricted by our inability to reliably unwrap highly decorrelated interferograms. Subsequently, we restrict ourselves to analyzing interferograms whose average coherence exceeds a certain threshold assuming that they can be reliably unwrapped. Our covariance model can potentially model the interaction between the phase noise terms in a network of interferograms, allowing us to potentially develop better statistical cost models (Chen and Zebker, 2001) for phase unwrapping.

5.6. Noisier SAR acquisitions

In our examples, we assumed that the noise in all SAR images are statistically similar. Our framework allows us to modify the interferogram networks suitably to compensate for noisier SAR acquisitions. It should be noted that reducing uncertainty in velocity and estimated deformation at a particular time instance can be competing objectives. In case of the former, the interferograms involving the noisier SAR observations tend to be pruned from the network in the attempt to improve a global estimate whereas in case of the latter, the number of interferograms involving the noisier SAR images need to be increased to reduce the uncertainty of the deformation estimate at the corresponding time epochs.

5.7. Persistent scatterers

All the examples and interferogram networks presented in this manuscript so far have dealt with distributed scatterers and the Gaussian signal model (Lee et al., 1994; Just and Bamler, 1994). Our technique can easily be extended for persistent scatterer analysis by using the phase statistics corresponding to the constant signal model (Ferretti et al., 2001; Agram, 2010) and suitably modifying the decorrelation model in Equation 17.

6. Conclusions

Most current time-series InSAR techniques operate under the assumption that interferometric phase observations are spatially and temporally independent. In this paper, we have built on simple models that describe phase statistics in single interferograms and developed a simple noise covariance model that shows that phase observations are both spatially and temporally correlated. Our model extends the work of Hanssen (2001) by formally deriving the covariance for interferograms with common SAR acquisitions from first principles. The most important implication of our covariance model is that accounting for spatial correlation in the inversion process can significantly improve the InSAR time-series parameter estimates (Hetland et al., 2011). We have also extended decorrelation models of individual interferograms (Zebker and Villasenor, 1992) to describe the phase noise statistics in a network of interferograms. The decorrelation covariance model can potentially be used to improve the spatial coverage of current time-series InSAR techniques by allowing us to better exploit the information from partially coherent scatterers.

7. Acknowledgements

This work was supported by the Keck Institute of Space Studies Postdoctoral fellowship.

A. List of symbols

N	Number of SAR acquisitions.
M	Number of interferograms in the network.
P	Number of coherent pixels in time-series analysis.
$\mathbf{A}$	Incidence matrix corresponding to the interferogram network of size $M \times N$.
$\mathbf{I}_{P,P}$	Identity matrix of order P.
$\overline{\mathbf{A}}$	Block diagonal matrix ($PM \times PN$) obtained by repeating $\mathbf{I}_{P,P} \otimes \mathbf{A}$.
$\Delta\phi_{ifg}^{x,i,j}$	InSAR phase of pixel x in IFG of SAR acquisitions with indices i and j.
$\Delta\phi_{defo}^{x,i,j}$	Contribution of deformation in InSAR phase of pixel x in IFG of SAR acquisitions with indices i and j.

$\Delta\phi_{aps}^{x,i,j}$	Contribution of atmospheric phase screen in InSAR phase of pixel x in IFG of SAR acquisitions with indices i and j.
$\Delta\phi_{decor}^{x,i,j}$	Contribution of decorrelation factors in InSAR phase of pixel x in IFG of SAR acquisitions with indices i and j.
$\phi_{n}^{x,i,j}$	Noise from uncorrelated sources in InSAR phase of pixel x in IFG of SAR acquisition with indices i and j.
$\phi_{sar}^{x,i}$	SAR phase that contributes towards the unwrapped interferometic phase observations for pixel x in SAR acquisition with index i.
$\phi_{defo}^{x,i}$	Contribution of deformation in SAR phase of pixel x in acquisition with index i.
$\phi_{aps}^{x,i}$	Contribution of atmospheric phase in SAR phase of pixel x in acquisition with index i.
$\phi_{decor}^{x,i}$	Contribution from decorrelation sources in SAR phase of pixel x in acquisition with index i.
$\overline{\Delta\phi_{ifg}}$	Vector of InSAR phases for all coherent pixels in all interferograms ($PM \times 1$).
$\overline{\phi_{defo}}$	Vector of SAR deformation phases for all pixels and all SAR acquisitions ($PN \times 1$).
$\overline{\phi_{aps}}$	Vector of SAR atmospheric phases for all pixels and all SAR acquisitions ($PN \times 1$).
$\overline{\phi_{decor}}$	Vector of SAR decorrelation phases for all pixels and all SAR acquisitions ($PN \times 1$).
$\overline{\phi_{n}}$	Vector of random phase noise for all pixels and all SAR acquisitions in the IFG network ($PM \times 1$).
$\mathbf{\Sigma}_{ifg}$	Covariance matrix of InSAR phase of all pixels in the IFG network ($PM \times PM$).
$\mathbf{\Sigma}_{aps}$	Covariance matrix of atmospheric InSAR phase of all pixels in the IFG network ($PM \times PM$).
$\mathbf{\Sigma}_{decor}$	Covariance matrix of decorrelation InSAR phase of all pixels in the IFG network ($PM \times PM$).
$\mathbf{\Sigma}_{n}$	Covariance matrix of uncorrelated InSAR phase noise for all pixels in the IFG network ($PM \times PM$).
$\sigma_{aps}^{x,y}$	Standard deviation of difference in atmospheric phase screen between two pixels x and y
$L_{x,y}$	The distance between two pixels x and y.

$H_{x,y}$	The difference in altitude between two pixels x and y.
$\eta_{aps}^{x,y}$	The covariance of atmospheric phase of two pixels x and y in the same SAR acquisition.
$\mathbf{\Sigma}_{aps}^{sar}$	The covariance matrix of atmospheric phase of all pixels in the same SAR acquisition.
$\gamma^{x,i,j}$	Interferometric coherence of pixel x in IFG of SAR acquisitions with indices i and j.
$z_{x,i}$	Complex signal return from a pixel x in SAR acquisition with index i.
$\mathbf{\Omega}_{sar}^{x}$	Correlation matrix of decorrelation phases of pixel $x$in all SAR acquisitions with respect to master scene $(N \times N)$. All the values of the matrix have been normalized to lie in the interval $[0,1]$.
$\tilde{\mathbf{\Omega}}_{ifg}^{x}$	Pseudo-correlation matrix of InSAR decorrelation phase of pixel x in the IFG network. The values have not been normalized to lie in the interval $[-1,1]$.
$\sigma_{\Delta\phi}^{x,i,j}$	Observed interferometric phase standard deviation around pixel x in interferogram (i,j).
$\mathbf{D}$	Diagonal matrix used to normalize $\mathbf{\Omega}_{ifg}^{x}$.
$\zeta(\cdot)$	Decorrelation model that maps perpendicular baseline and SAR acquisition times to interferometric phase standard deviation.

B. Coherence and decorrelation phase covariance

We demonstrate the applicability of our model to high and coherence pixels using a simple set of 3 SAR scenes A, B and C. In the first case, we assume that a pixel is coherent in all three interferogram pairs - AB, BC and CA. In the second case, we assume that A and C represent summer SAR acquisitions and B represents a winter SAR acquisition. Consequently, interferogram AC is more coherent than the pairs AB and BC. For our numerical example, we assume 20 looks and use Equation 12 for mapping coherence to phase standard deviation. A more accurate mapping from the coherence to the phase standard deviation can be obtained using the probability distribution functions for interferometric phase from appropriate signal models (Just and Bamler, 1994; Lee et al., 1994; Agram, 2010).

$$\mathbf{A} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 1 & 0 & -1 \end{bmatrix} \rightarrow \begin{bmatrix} AB \\ BC \\ CA \end{bmatrix}$$		
	Case I	Case II
Coherence in SAR image domain ($\mathbf{\Omega}_{sar}$)	$\begin{bmatrix} 1 & 0.9 & 0.7 \\ 0.9 & 1 & 0.8 \\ 0.7 & 0.8 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0.3 & 0.8 \\ 0.3 & 1 & 0.35 \\ 0.8 & 0.35 & 1 \end{bmatrix}$
Standard phase deviation from coherence. ($\sigma_{\Delta\phi}$)	$\begin{bmatrix} 0.012 \\ 0.028 \\ 0.052 \end{bmatrix}$	$\begin{bmatrix} 0.506 \\ 0.358 \\ 0.028 \end{bmatrix}$
Pseudo-correlation of the interferograms. ($\tilde{\mathbf{\Omega}}_{ifg}$)	$\begin{bmatrix} 0.2 & 0 & 0.2 \\ 0 & 0.4 & 0.4 \\ 0.2 & 0.4 & 0.6 \end{bmatrix}$	$\begin{bmatrix} 1.4 & -1.15 & 0.25 \\ -1.15 & 1.3 & 0.15 \\ 0.25 & 0.15 & 0.4 \end{bmatrix}$
Normalization matrix ($\mathbf{D}$)	$\begin{bmatrix} 2.12 & 0 & 0 \\ 0 & 1.41 & 0 \\ 0 & 0 & 1.08 \end{bmatrix}$	$\begin{bmatrix} 0.46 & 0 & 0 \\ 0 & 0.52 & 0 \\ 0 & 0 & 1.41 \end{bmatrix}$
Covariance matrix ($\mathbf{\Sigma}_{decor}$)	$\begin{bmatrix} 0.012 & 0 & 0.014 \\ 0 & 0.028 & 0.031 \\ 0.014 & 0.031 & 0.052 \end{bmatrix}$	$\begin{bmatrix} 0.506 & -0.363 & 0.040 \\ -0.363 & 0.358 & 0.021 \\ 0.040 & 0.021 & 0.028 \end{bmatrix}$

From our example, it is clear that the contribution of decorrelation to a coherent pixel covariance (Case I) is negligible. However, for a partially coherent pixel (Case II), the noise terms cannot be ignored. Moreover, our covariance model can capture the effects like the negative correlation between interferogram AB and BC, that can potentially allow us to exploit the phase information from these noisy interferograms.

References

Agram, P. S., 2010. Persistent scatterer interferometry in natural terrain. Ph.D. thesis, Stanford University.

Agram, P. S., Jolivet, R., Riel, B., Lin, Y.-N., Simons, M., Hetland, E., Doin, M.-P., Lasserre, C., Submitted. EOS Trans.

Agram, P. S., Simons, M., 2012. A sequential approach to optimizing geodetic networks. IEEE Geosci.Remote Sensing Lett.In preparation.

Bamler, R., Just, D., August 1993. Phase statistics and decorrelation in sar interferograms. In: Proceedings of Geoscience and Remote Sensing Symposium, IGARSS' 93. Vol. 3. pp. 980–984.

Baran, I., Stewart, M. P., Kampes, B. M., Perski, Z., Lilly, P., 2003. A modification to the goldstein radar interferogram filter. IEEE Trans. on Geosci. Remote Sensing 41 (9), 2114–2118.

Berardino, P., Fornaro, G., Lanari, R., Sansosti, E., 2002. A new algorithm for surface deformation monitoring based on small baseline differential sar interferograms. IEEE Trans. on Geosci. Remote Sensing 40 (11), 2375–2383.

Biggs, J., Wright, T., Lu, Z., Parsons, B., 2007. Multi interferogram method for measuring interseismic deformation: Denali fault, alaska. Geophys. J. Int. 170, 1165–1179.

Cavalié, O., Doin, M.-P., Lasserre, C., Briole, P., 2007. J. Geophys. Res. 112 (B03403), 18.

Chapin, E., Chan, S., Chapman, B., Chen, C., Martin, J., Michel, T., Muellerschoen, R., Pi, X., Rosen, P., april 2006. Impact of the ionosphere on an l-band space based radar. In: Radar, 2006 IEEE Conference on. p. 8 pp.

Chen, C. W., Zebker, H. A., 2001. Two-dimensional phase unwrapping with use of statistical models for cost functions in nonlinear optimization. J. Opt. Soc. Am. A 18 (2), 338–351.

Cumming, I., Wong, F., 2005. Digital processing of SAR data. Artech House, Inc., 685 Canton Street, Norwood, MA 02062.

Delacourt, C., Briole, P., Achache, J. A., 1998. Geophys. Res. Lett. 25 (15), 2849–2852.

Doin, M. P., Guillaso, S., Jolivet, R., Lasserre, C., Lodge, F., Ducret, G., Gradin, R., September 2011. Presentation of the small baseline nsbas processing chain on a case example: The etna deformation monitoring from 2003 to 2010 using envisat data. In: FRINGE 2011 ESA Conference. ESA, Frascati, Italy.

Emardson, T. R., Simons, M., Webb, F. H., 2003. Neutral atmospheric delay in interferometric synthetic aperture radar applications: Statistical description and mitigation. J. Geophys. Res. 15 (B5), 2231.

Ferretti, A., Fumagalli, A., Novali, F., Prati, C., Rocca, F., Rucci, A., 2011. A new algorithm for processing interferometric data-stacks: Squeesar. IEEE Trans. on Geosci. Remote Sensing 49 (8).

Ferretti, A., Prati, C., Rocca, F., 2001. Permanent scatterers in sar interferometry. IEEE Trans. on Geosci. Remote Sensing 39 (1), 8–20.

Foster, J., Brooks, B., Cherubini, T., Shacat, C., Businger, S., Werner, C., 2006. Geophys. Res. Lett. 33 (L16304), 5.

Goldstein, R. M., 1995. Atmospheric limitations to repeat-track radar interferometry. Geophys. Res. Lett. 22 (18), 2517–2520.

Goldstein, R. M., Werner, C. L., 1998. Radar interferogram filtering for geophysical applications. Geophys. Res. Lett. 25 (21), 4035–4038.

González, P. J., Fernández, J., 2011. Error estimation in multitemporal insar deformation time series, with application to lanzarote, canary islands. J. Geophys. Res. 116 (B10404).

Guarnieri, A. M., Tebaldini, S., 2007. Hybrid cramer-rao bounds for crustal displacement field estimators in sar interferometry. IEEE Signal Processing Letters 14 (12).

Hanssen, R., 1998. Atmospheric Heterogeneties in ERS Tandem SAR Interferometry. Delft University Press, Delft, The Netherlands.

Hanssen, R. F., 2001. Radar Interferometry: Data Interpretation and Error Analysis, 1st Edition. Springer.

Hetland, E. A., Muse, P., Simons, M., Lin, Y. N., Agram, P. S., DiCaprio, C. J., 2011. Multiscale insar time series (mints) analysis of surface deformation. J. Geophys. Res. 117 (B02404).

Hooper, A., May 2006. Persistent scatterer radar interferometry for crustal deformation studies and modeling of volcanic deformation. Ph.D. thesis, Stanford University.

Hooper, A., 2008. A multi-temporal insar method incorporating both persistent scatterer and small baseline approaches. Geophys. Res. Lett. 35, L16302.

Hooper, A., Segall, P., Zebker, H., 2007. Persistent scatterer interferometric synthetic aperture radar for crustal deformation analysis, with application to volcn alcedo, galpagos. J. Geophys. Res. 112, B07407.

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. Geophys. Res. Lett. 31, L23611.

Jolivet, R., Grandin, R., Lasserre, C., Doin, M.-P., Peltzer, G., 2011. Systematic insar tropospheric phase delay corrections from global meteorological reanalysis data. Geophys. Res. Lett. 38 (L17311).

Jónsson, S., 2002. Modeling volcano and earthquake deformation from satellite radar interferometric observations. Ph.D. thesis, Stanford University.

Just, D., Bamler, R., 1994. Phase statistics of interferograms with applications to synthetic-aperture radar. Appl. Opt. 33 (20), 4361–4368.

Knospe, S. H.-G., Jónsson, S., 2010. Covariance estimation for dinsar surface deformation in presence of anisotropic atmospheric noise. IEEE Trans. on Geosci. Remote Sensing 48 (4), 2057–2070.

Lauknes, T. R., Shanker, A. P., Dehls, J. F., Zebker, H. A., Henderson, I. H. C., Larsen, Y., 2010. Detailed landslide mapping in northern norway with small-baseline and persistent scatterer interferometric sar time series methods. Remote Sensing of Environment 114 (9), 2097–2109.

Lee, J. S., Hoppel, K. W., Mango, S. A., Miller, A. R., 1994. Intensity and phase statistics of multilook polarimetric and interferometric sar imagery. IEEE Trans. on Geosci. Remote Sensing 32 (5), 1017–1028.

Lin, Y. N., Simons, M., Hetland, E. A., Muse, P., DiCaprio, C., 2010. A multi-scale approach to estimating topographically correlated propagation delays in radar interferograms. G-cubed 11 (Q09002).

Lohman, R. B., Simons, M., 2005. Some thoughts on the use of insar data to constrain models of surface deformation. G-cubed 6 (1).

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.

Mesinger, F., DiMego, G., Kalna, E., Mitchell, K., Shafran, P., Ebisuzaki, W., Jović, D., Woollen, J., Rogers, E., Berbery, E., Ek, M., Fan, Y., Grumbine, R., Higgins, W., Li, H., Lin, Y., Manikin, G., Parrish, D., Shi, W., 2006. North american regional reanalysis. Bull. Amer. Met. Soc. 87 (3), 343–360.
URL `http://ci.nii.ac.jp/naid/30022376287/en/`

Meyer, F., july 2010. A review of ionospheric effects in low-frequency sar; signals, correction methods, and performance requirements. In: Geoscience and Remote Sensing Symposium (IGARSS), 2010 IEEE International. pp. 29 –32.

Meyer, F., Watkins, B., july 2011. A statistical model of ionospheric signals in low-frequency sar data. In: Geoscience and Remote Sensing Symposium (IGARSS), 2011 IEEE International. pp. 1493 –1496.

Onn, F., Zebker, H. A., 2006. Correction for interferometric synthetic aperture radar atmospheric phase artifacts using time series of zenith wet delay observations from a gps network. J. Geophys. Res. 111 (B09102).

Peltzer, G., Hudnut, K., Feigl, K., 1994. Analysis of coseismic surface displacement gradients using radar interferometry: New insights into the landers earthquake. J. Geophys. Res. 99 (21), 971–981.

Perissin, D., Wang, T., 2011. Repeat-pass SAR interfeormetry with partially coherent targets. IEEE Trans. on Geosci. Remote Sensing (99), 1–10.

Pollitz, F., Peltzer, G., Bürgmann, R., 2000. Mobility of continental mantle: Evidence from postseismic geodetic observations following the 1992 landers earthquake. J. Geophys. Res. 105 (B4), 8035–8054.

Pritchard, M. E., Simons, M., 2002. A satellite geodetic survey of large scale deformation of volcanic centers in the central andes. Nature 418 (6894), 167–171.

Rocca, F., 2007. Modeling interferogram stacks. IEEE Trans. on Geosci. Remote Sensing 45 (10).

Rodriguez, E., Martin, J. M., 1992. Theory and design of interferometric synthetic aperture radars. In: Radar and Signal Processing, IEE Proceedings F. Vol. 139. pp. 147–159.

Rosen, P., Hensley, S., Chen, C., may 2010. Measurement and mitigation of the ionosphere in l-band interferometric sar data. In: Radar Conference, 2010 IEEE. pp. 1459 –1463.

Rosen, P. A., Hensley, S., Joughin, I. R., Li, F. K., Madsen, S. N., Rodriguez, E., Goldstein, R. M., 2000. Synthetic aperture radar interferometry. Proc. IEEE 88 (3), 333–382.

Samsonov, S., july 2010. Topographic correction for alos palsar interferometry. Geoscience and Remote Sensing, IEEE Transactions on 48 (7), 3020 –3027.

Shanker, A. P., Zebker, H., 2007. Persistent scatterer selection using maximum likelihood estimation. Geophys. Res. Lett. 34, L22301.

Simons, M., Fialko, Y., Rivera, L., 2002. Coseismic deformation from the 1999 msubw 7.1 hector mine, california, earthquake as inferred from insar and gps observations. Bull. Seism. Soc. Am. 92 (4), 1390–1402.

Touzi, R., Lopes, A., Bruniquel, J., Vachon, P. W., 1990. Coherence estimation for sar imagery. IEEE Trans. on Geosci. Remote Sensing 28 (1), 88–97.

Zebker, H., Chen, K., april 2005. Accurate estimation of correlation in insar observations. Geoscience and Remote Sensing Letters, IEEE 2 (2), 124 – 127.

Zebker, H., Hensley, S., Shanker, P., Wortham, C., 2010. Geodetically accurate insar data processor. IEEE Trans. on Geosci. Remote Sensing 48 (12).

Zebker, H. A., Rosen, P. A., Goldstein, R. M., Gabriel, A., Werner, C., 1994. On the derivation of coseismic displacement field using differential radar interferometry : The landers earthquake. J. Geophys. Res. 99, 19618–19634.

Zebker, H. A., Villasenor, J., 1992. Decorrelation in interferometric radar echoes. IEEE Trans. on Geosci. Remote Sensing 30 (5), 950–959.

Attachment 3:

A noise model for time-series InSAR

Segmentation of the San Andreas Fault around Parkfield from Space Geodetic Data

Sylvain Barbot, Piyush Agram, Mark Simons and Marcello De Michele

California Institute of Technology, 1200 E California Blvd, California, USA

Abstract. Sequences of earthquakes are commonly represented as a succession of periods of interseismic stress accumulation followed by co- and postseismic phases of stress release. Because the recurrence time of large earthquakes is often greater than the available span of space geodetic data, it has been challenging to monitor the evolution of interseismic loading in its entire duration. Here, we analyze large datasets of surface deformation at different key episodes around the Cholame, Parkfield, and creeping segments of the San Andreas Fault that show evidence of significant deceleration of fault slip during the interseismic period. We compare the average fault slip rates before and after the 2004 Mw 6 Parkfield earthquake, in the 1986-2004 and 2006-2012 periods, respectively, avoiding two years of postseismic deformation after 2004. Using a combination of GPS data from the Plate Boundary Observatory, the SCEC Crustal Motion Map and the Bay Area Velocity Unification networks and interferometric synthetic aperture radar from the ALOS and ENVISAT satellites, we show that the area of coupling at the transition between the Parkfield and Cholame segments appears larger later in the interseismic period than it does earlier on. While strong plate coupling is uniform across the Parkfield and Cholame segments in the 1986-2004 period, creep occurs south of the 2004 epicenter after 2006, making segmentation of the San Andreas Fault south of Parkfield more clearly apparent. These observations indicate that analyses of surface deformation late in the earthquake cycle may over-estimate the area of plate coupling. A fault surface creeping much below plate rate may in some case be a region that does not promote earthquake nucleation, but rather just be at a slower stage of its evolution. Large variations of creep velocity above and below plate rate indicates that a cycle of weakening and hardening can also be at play in areas of velocity-strengthening friction. Our analysis also shows signs of large variation of slip velocity in the creeping segment, which is another indication that interseismic velocity can exhibit complex variations in time.

1. Introduction

The Cholame, Parkfield, and creeping segments of the San Andreas Fault (SAF) form the northern boundary of the great M 7.9 1857 Fort Tejon earthquake and its potential foreshocks [*Wood*, 1955]. The Cholame segment was ruptured during the 1857 event [*Sieh*, 1978a] and there is evidence of three similar earthquakes since 1000 A. D. [*Young et al.*, 2002]. Offset of geological markers indicates a late Holocene slip rate of $26.2 + 6.4/-4.3$ mm/yr near Parkfield [*Toké et al.*, 2011], and 33.9 ± 2.9 mm/yr near Wallace Creek [*Sieh and Jahns*, 1984]. Since the last rupture, about 5 m of slip deficit accumulated at crustal depths and Cholame may soon be the host of another large devastating earthquake [*Wesnousky*, 1986]. The Parkfield segment is a transition zone between the locked Cholame segment and the creeping section to the northwest and is the site of at least 6 Mw $\sim$ 6 earthquakes since 1857 in 1881, 1901, 1922, 1934, 1966 and 2004 [*Bakun and McEvilly*, 1984; *Bakun and Lindh*, 1985; *Bakun et al.*, 2005]. Because of the short recurrence time of the Mw 6 earthquakes, between 12 and 38 years, Parkfield is an ideal place to study the earthquake cycle and test the potential of earthquake forecast [e.g., *Roeloffs and Langbein*, 1994; *Barbot et al.*, 2012]. The creeping segment, starting a few kilometers north of the 1966 Mw 6 Parkfield epicenter, has had no earthquakes larger than Mw 4 in the past 65 years and has been creeping at depth at a rate of 33 mm/yr, based on geodetic data in the 1969-1976 period [*Thatcher*, 1979; *Burford and Harsh*, 1980].

It is important to understand the mechanics of faulting at this crucial location of the SAF to address both fundamental questions about earthquake physics and mitigate seismic hazards, but many aspects of the kinematics of the SAF remain unexplained.

Shallow creep in the creeping section is high, but in some places markedly slower than at greater depths [*Thatcher*, 1979; *Johanson and Bürgmann*, 2005; *Rolandone et al.*, 2008; *Ryder and Bürgmann*, 2008]. The implications of this shallow slip deficit is not fully understood and

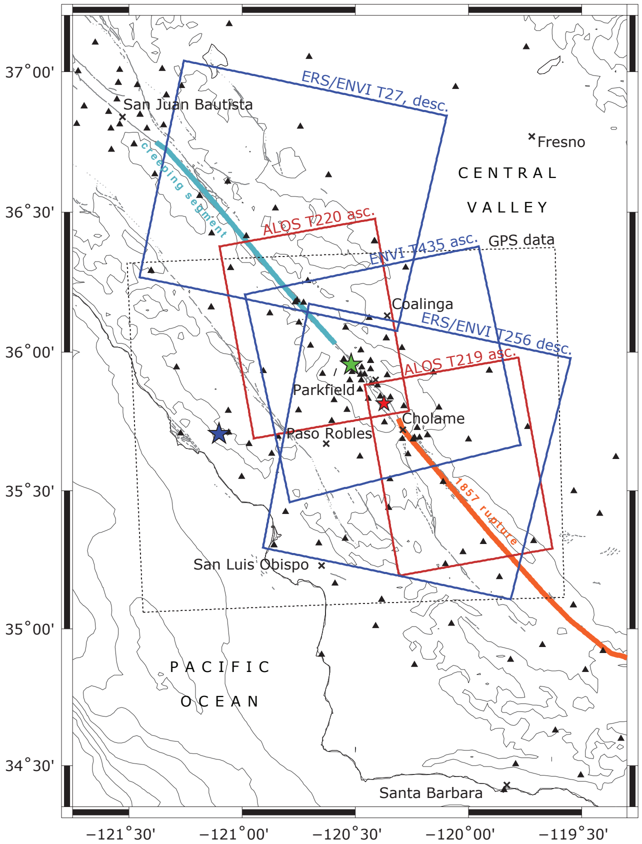


Figure 1. Area of study, including the Cholame-Carrizo, Parkfield and creeping segments of the San Andreas Fault. The 1966 and 2004 Parkfield and the 2003 San Simeon earthquakes are showed in green, red and blue stars, respectively. The local topography/bathymetry is shown in grey profiles. The subset of continuous GPS stations used in this study is contained in the black dashed box. The footprints of the ENVISAT, ERS and ALOS interferograms used in the study are indicated by the colored boxes.

may be the result of stress build up released in earthquakes [*Toppozada et al.*, 2002], spontaneously non-steady fault slip due to a particular friction behavior in a manner similar to slow slip events of slow earthquakes [e.g., *Rogers and Dragert*, 2003; *Liu and Rice*, 2005; *Miyazaki et al.*, 2006; *Ito et al.*, 2007], or transient loading from earthquakes on neighboring segments [e.g., *Ben-Zion et al.*, 1993].

Paleoseismic studies indicate that the 1857 Fort Tejon earthquake was preceded by two foreshocks in an area that includes the Parkfield and creeping segments [*Sieh*, 1978a, b; *Toppozada et al.*, 2002]. There is a possibility that the foreshock activity includes events similar to the Mw 6 earthquake sequence at Parkfield and that these earthquakes can trigger a larger rupture along the Cholame and Carrizo segments. Because many Parkfield earthquakes did not trigger a larger event, the mechanics behind this scenario implies a soft barrier between the Parkfield and Cholame segments, which would function as an efficient arrest to rupture only on occasion in history [e.g., *Kaneko et al.*, 2010].

Recently, *Barbot et al.* [2012] presented a physical model of earthquakes at Parkfield based on rate-and-state friction that can explain many aspects of seismological and geodetic observations. The model assumes that the seismogenic zone is delineated by persistent streaks of seismicity and that the 1966 and 2004 events started at the northern and southern boundaries of the seismogenic zone, respectively. The southern end of the seismogenic zone is not fully apparent from inversion of geodetic data before the 2004 earthquake [*Segall and Du*, 1993; *Bakun et al.*, 2005; *Murray and Segall*, 2005], and documenting the presence of this potential boundary can have important implications on the mechanics of the earthquake cycle at the Parkfield and Cholame segments. In particular, it is important to know how the series of Mw 6 earthquakes at Parkfield are arrested and initiated at the southern end and what are the mechanical properties of the boundary between the Cholame and Parkfield segments.

To address these questions, we use geodetic data of the interseismic period of the Parkfield earthquake cycle (Fig. 1). We use average velocities of surface displacement derived from analysis of continuous GPS times series and interferometric synthetic aperture radar (InSAR). Many analyses of InSAR and GPS have been successful at constraining slow interseismic deformation across faults [*Wright et al.*, 2004; **?**; *Cavalié et al.*, 2008; *Jolivet et al.*, 2008; *Elliott et al.*, 2011; *Wang et al.*, 2009; *Fay and Humphreys*, 2005; *Lundgren et al.*, 2009; *Bell et al.*, 2011; *Lindsey and Fialko*, 2012; *Jolivet et al.*, 2012] and our work extends studies focused on the northern termination of the central SAF section [*Rolandone et al.*, 2008; *Ryder and Bürgmann*, 2008; *Johanson and Bürgmann*, 2010]. Because the apparent interseismic velocity can change appreciably during the interseismic cycle [e.g., *Tse and Rice*, 1986; *Lapusta et al.*, 2000; *Barbot et al.*, 2012; **?**], we compare interseismic fault slip rates before and after the 2004 Parkfield earthquake. This allows us to build a more complete picture of the range of possible behavior in the interseismic period.

The manuscript is organized as follows. In Sections 2 and 3, we describe the processing of InSAR data and our inversion method. In Sections 4 and 5, we present the

Table 1. Table of all SAR data used in this manuscript.

Sensor	Geometry	Track	Frame	N_{SAR}	N_{InSAR}
Pre-earthquake (1992 - 2004)					
ERS	desc	27	2871	43	47
ERS	desc	256	2889	41	80
ENVI	asc	435	711	5	6
Post-earthquake (2006-2010)					
ENVI	desc	27	2871	23	34
ERS	desc	256	2889	4	2
ENVI	desc	256	2889	11	20
ENVI	asc	435	711	22	66
ALOS	asc	220	710	16	83
ALOS	asc	219	700	16	59

fault slip rates in the 1986-2004 and 2006-2012 periods, respectively. We discuss the implications for earthquake mechanics in Section 6.

2. InSAR data and processing

In this work, we use a rich dataset of interferograms constructed from SAR images acquired by the ERS, ENVISAT and ALOS satellites and spanning more than 15 years. Figure 1 shows the frame boundaries of the 5 different sets of SAR images used in this study. All C-band data, from the ERS and ENVISAT satellites, are obtained from the WInSAR and the Earthscope ESA SAR archives and the ALOS PALSAR data are obtained from the Alaska SAR Facility's (ASF) DAAC. Pre-earthquake data (1992 to 2004) were mostly acquired by the ERS-1 and ERS-2 satellites, whereas ENVISAT and ALOS satellites account for the dense temporal coverage after the Parkfield earthquake (2006 to 2010). Table 1 summarizes all the SAR data used in this work and the corresponding baseline plots are included in Figure 2 and the supplementary materials.

Subsets of our large SAR data set have been used in previously published studies over our region of interest. *Johanson et al.* [2006] studied the coseismic and immediate postseismic deformation after the September 28, 2004 Parkfield earthquake using Radarsat and a more limited set of ENVISAT interferograms. *Ryder and Bürgmann* [2008] applied stacking techniques to a set of 12 interferograms covering the creeping section of the San Andreas Fault (Track 27) and observed that the study area (Figure 1) is characterized by absence of strongly reflecting urban targets and severe decorrelation due to vegetation. *De Michele et al.* [2011] analyzed data from Track 256 to derive a detailed surface velocity field prior to the Parkfield earthquake. We increase the number of interferometric observations before the earthquake by including a larger set of interferograms from Track 27 and a few ENVISAT interferograms from Track 435 (See Table 1 and and Figs. 2, A1, A2, A3, A4 and A9). The SAR images prior to the earthquake were primarily acquired on descending passes. The number of post-EQ C-band interferograms per track is smaller than the corresponding pre-EQ data set. However, ascending pass L-band images from the ALOS PALSAR instrument, launched in 2006, makes up for reduced number of observations by imposing better geometric constraints.

2.1. InSAR Processing

The ERS and ENVISAT interferograms are individually processed using the ROI-PAC package [*Rosen et al.*, 2004] and then co-registered against a master ERS SAR scene. ERS scenes affected by the gyroscope failure in 2001 are preprocessed to determine the correct doppler ambiguities before processing with ROI-PAC. Despite monthly acquisitions, a large number of the ERS-2 scenes acquired after 2001 could not be used due to large doppler baselines. All the C-band interferograms are processed at a posting of roughly 100 meters (4 looks in range and 20 looks in azimuth). The ALOS PALSAR interferograms are processed using Stanford University's *mocomp* processor [*Zebker et al.*, 2010] to a common imaging geometry. All the FBD mode scenes are upsampled to full resolution after focusing and before coregistration. The ALOS interferograms are also generated at a posting of 100 meters (12 looks in range and 28 looks in azimuth).

All the interferograms are filtered using a Goldstein filter [*Goldstein and Werner*, 1998] of moderate strength and individually unwrapped using SNAPHU [*Chen and Zebker*, 2002]. All the post-EQ earthquakes are de-ramped using daily GPS solutions from the SOPAC archive (*sopac.ucsd.edu*). All the pre-EQ interferograms are de-ramped by removing the best fitting plane from the unwrapped phase, due to the absence of continuous GPS stations between 1992-1994. A large number of interferograms were initially generated and only a subset (Table 1) with greater than 60% spatial coverage (coherence greater than 0.25) over the entire frame are retained for analysis with the Multi-scale Interferometric Time Series (MInTS) technique [*Hetland et al.*, 2012].

2.2. MInTS time series processing

We first analyzed the C-band interferogram stacks using an integral spline formulation similar to *Hetland et al.* [2012] and we observed that the associated uncertainties are larger than 1 cm possibly due to 1) insufficient redundancy of the C-band interferogram networks (compare number of coherent interferograms versus number of SAR images in Table 1) and 2) presence of independent sub-networks of coherent interferogram clusters (supplementary material) particularly after 2001. Direct time-series estimation for the ALOS PALSAR stacks is not carried out due to the limited number of SAR acquisitions. We apply the MInTS technique to determine a constant line-of-sight velocity term and a seasonal signal amplitude with a 1-year period for each of the interferogram stacks at a spatial resolution of 100 m. Assuming a simplified temporal model for deformation allows us to overcome the rank deficiency issues arising due to disconnected interferogram sub-networks. The MInTS technique has two advantages: 1) it allows us to interpolate over small decorrelated regions in space and 2) it uses wavelets to reduce the impact of atmospheric phase contributions in the estimated deformation parameters. The interpolation capability, allowed us to increase the number of viable interferograms compared to other stacking techniques applied over the same region. We also estimate uncertainties associated with our simple temporal model and use the information to mask out noisy pixels in the velocity maps before modelling. Finally, all the InSAR data are down-sampled to a posting of 2 km before modelling.

2.3. Estimation of uncertainties

We use a data-driven leave-one-out bootstrap approach [?] to determine the uncertainties in the estimated temporal model parameters. These model parameters could represent piecewise linear functions [?] leading to direct estimation of the deformation time-series or a set of temporal functions as is typically used in GPS processing [*Bock et al.*, 1997; *Wdowinkski et al.*, 1997; ?] or MInTS [*Hetland et al.*, 2012]. For each frame, subsets of interferograms are generated by leaving out observations corresponding to a SAR scene one at a time, and a set of temporal model parameters are estimated for each of the subsets. The mean and the standard deviation of the estimated parameters are interpreted as the nominal value and the associated uncertainty.

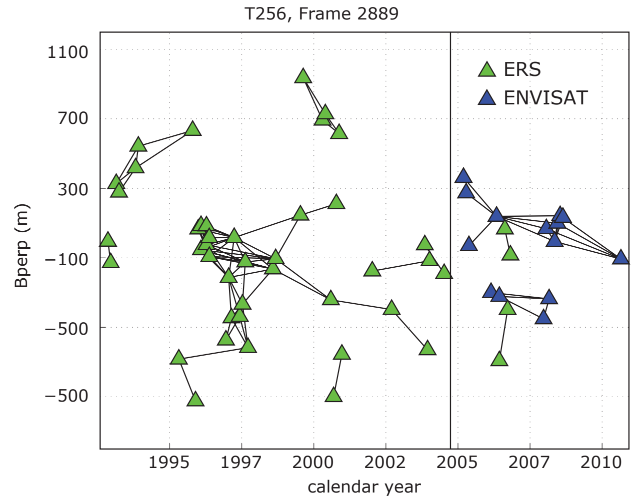


Figure 2. SAR acquisitions (triangles) and interferograms (black segments) of ERS and ENVISAT data considered in this study for track 256 and frame 2889. See Figs. A1, A2, A3, A4 and A9 for a description of other SAR data used in the study.

3. Joint inversion of GPS and InSAR data

In this section, we explain our method to estimate the distribution of slip rates on various faults using InSAR and GPS data simultaneously. The GPS average velocity is relative to an arbitrary reference frame, which adds a component to the velocity vectors that cannot be explained by local fault motion. Likewise, InSAR LOS measurements are relative to an unknown range and may suffer from orbital errors. We test two methods to mitigate these effects. In a first method, we include the reference frame of the GPS networks and the orbital error of InSAR in the inversion so that the effects of fault displacement and other contributions can be separated. In a second method, we use the baseline velocity between pairs of GPS stations and the gradient of the LOS displacements to constrain fault slip.

3.1. Case of absolute GPS and InSAR velocity

For each period considered, we jointly invert InSAR data from all tracks and GPS average velocities from up to two separate networks for slip rate on faults, dilatational opening of point sources and other non-tectonic parameters. We estimate the optimal orbital errors and GPS reference frames part of a global inversion where slip on faults and other parameters are optimized simultaneously. The geodetic data reduction can be formulated as the minimization problem

$$\tilde{\mathbf{m}} = \min_{\mathbf{m}} \left\{ \|\mathbf{h} - \mathbf{H}\,\mathbf{m}\|^2 \right\} \tag{1}$$

subject to

$$\mathbf{A}\,\mathbf{m} \geq 0 \tag{2}$$

bounded in the range $\mathbf{l} \leq \mathbf{m} \leq \mathbf{u}$, where $\tilde{\mathbf{m}}$ is the vector of optimized parameters, including fault slip rates, and $\mathbf{m}$ is the model space. The target vector is formed by a combination of data and constraints

$$\mathbf{h} = \begin{pmatrix} \mathbf{d} \\ \mathbf{0} \end{pmatrix} \tag{3}$$

Similarly, the Hessian is a combination of the design matrix $\mathbf{G}$ and the smoothing operator $\mathbf{D}$

$$\mathbf{H} = \begin{pmatrix} \mathbf{G} & \\ \mathbf{D} & \mathbf{0} \end{pmatrix} \ . \tag{4}$$

We use lower and upper bounds on the solution, $\mathbf{l}$ and $\mathbf{u}$, respectively, as a form of regularization to avoid spurious numerical instabilities. We invoke the constraints of eq. (2) to impose the rake of fault slip, while letting non-tectonic parameters unbounded.

The data vector is a combination of the GPS horizontal velocity components $\mathbf{d}^{\mathrm{GPS}}$, and the InSAR LOS measurements $\mathbf{d}^{\mathrm{SAR}}$. When jointly inverting P InSAR stacks and N GPS networks, we add a relevant subscript to the vectors, and form the global data vector in series

$$\mathbf{d} = \begin{pmatrix} w\,\mathbf{d}_1^{\mathrm{SAR}} \\ \vdots \\ w\,\mathbf{d}_P^{\mathrm{SAR}} \\ \mathbf{d}_1^{\mathrm{GPS}} \\ \vdots \\ \mathbf{d}_N^{\mathrm{GPS}} \end{pmatrix} \tag{5}$$

The InSAR data consist typically of a few thousand points, while the GPS vectors, just a few hundreds. To compensate for the difference in the number of point measurements, we introduce a weight w on the InSAR data. The model parameters are then organized as follows

$$\mathbf{m} = \begin{pmatrix} \mathbf{s} \\ \mathbf{o}_1 \\ \vdots \\ \mathbf{o}_P \\ \mathbf{r}_1 \\ \vdots \\ \mathbf{r}_N \end{pmatrix} \tag{6}$$

where $\mathbf{s}$ is a vector of slip rates on fault patches and opening rates of dilatation sources (typically a few hundreds of parameters). The vectors $\mathbf{o}_i$ are three orbital parameters for InSAR stack i and each $\mathbf{r}_i$ contains two reference frame parameters for GPS network i. Data and model space are connected through the design matrix

$$\mathbf{G} = \begin{pmatrix} w\,\mathbf{G}_1^{\mathrm{SAR}} & w\,\mathbf{G}_1^{\mathrm{orb}} & & & & \\ \vdots & & \ddots & & & \\ w\,\mathbf{G}_P^{\mathrm{SAR}} & & & w\,\mathbf{G}_P^{\mathrm{orb}} & & \\ \mathbf{G}_1^{\mathrm{GPS}} & & & & \mathbf{G}_1^{\mathrm{ref}} & \\ \vdots & & & & & \ddots \\ \mathbf{G}_N^{\mathrm{GPS}} & & & & & & \mathbf{G}_N^{\mathrm{ref}} \end{pmatrix} \tag{7}$$

where the weight w on InSAR data is taken into account. The $\mathbf{G}^{\mathrm{SAR}}$ and $\mathbf{G}^{\mathrm{GPS}}$ matrices are computed using unitary slip on rectangular fault patches using a combination of strike- and dip-slip prescribed by an assumed rake of the slip vector. We use the analytic expression of *Okada* [1985] for a homogeneous medium or the semi-analytic solution of *Wang et al.* [2003] for a layered model. We ignore the effects of lateral heterogeneities in the Earth's crust that can also affect distribution of slip velocity [e.g., *Fay and Humphreys*, 2005; *Lundgren et al.*, 2009; *Lindsey and Fialko*, 2012]. For those patches that are infinitely long to represent relative plate motion, we use the solution for a two-dimensional buried screw dislocation

$$u = \frac{1}{\pi} \arctan\left(\frac{y}{D}\right) \ , \tag{8}$$

where u is the surface fault parallel displacement, y is the fault perpendicular coordinate and D is the locking depth, or the solution of *Segall* [2010] for a layered medium

$$u = \frac{2}{\pi}\frac{\gamma}{1+\gamma}\Bigg[\arctan\frac{y}{D} + \sum_{m=1}^{\infty}\left(\frac{1-\gamma}{1+\gamma}\right)^m \arctan\frac{y}{D+2Hm}\Bigg], \tag{9}$$

where H is the bottom depth of the top layer and $\gamma = \mu_{\text{top}}/\mu_{\text{down}}$ is the rigidity ratio. Here, we ignore the viscoelastic effects [e.g., *Johnson and Segall*, 2004] and the deep-seated deformation accommodated by both localized fault slip and more distributed strain are modeled using elastic dislocation [*Fay and Humphreys*, 2005; *Lundgren et al.*, 2009].

The smoothing between slip patches is obtained through a finite difference approximation $\mathbf{L}$ of the Laplacian operator for irregular surfaces [*Huiskamp*, 1991; *Kositsky and Avouac*, 2010]. Fault estate is separated into K segments that are smoothed independently of each other so we form the smoothing operator $\mathbf{D}$ as follows

$$\mathbf{D} = \mathbf{\Lambda}\begin{pmatrix} \mathbf{L}_1 & & \\ & \ddots & \\ & & \mathbf{L}_K \end{pmatrix} \tag{10}$$

The strength of smoothing $\mathbf{\Lambda}$ is determined based on the resolution of fault patches. Using a singular value decomposition, we first determine the resolution matrix of the inversion

$$\mathbf{R} = \mathbf{G}^\dagger \mathbf{G} \tag{11}$$

where $\mathbf{G}^\dagger$ is the generalized inverse of the forward operator obtained by truncated singular value decomposition as a form of Tikhonov regularization [e.g., *Pritchard et al.*, 2002; **?**]. The eigenspectrum is truncated at eigenvalues lower than a threshold defined as the ratio of the expected noise to the expected fault velocity (eq. 15). Then, the smoothing weight is determined from the following equation

$$\Lambda_{ii} = \lambda_1 + \lambda_0 \cos\left(\frac{\pi R_{ii}}{2}\right)^{10} \tag{12}$$

where the coefficients λ_1 and $\lambda_0 + \lambda_1$ are obtained empirically and correspond to the smoothing weights for well- and poorly-resolved parameters, respectively.

3.2. The Case for GPS baseline velocity and InSAR velocity gradients inversions

We now consider the inversion of the baseline velocity between pairs of GPS stations and of the horizontal gradient of the LOS data. For sufficiently small GPS networks, it is often sufficient to assume a constant vector to represent the velocity of the reference frame, but for larger networks, a radial velocity and an Euler pole are a more adequate representation. However, simultaneously inverting for these two parameters is not a linear inversion, which complicates the analysis of large networks. It seems therefore advantageous to use the relative velocity between pairs of nearby stations as data constraints. In general, taking the difference between two time series of displacement increases the noise by a factor of $\sqrt{2}$. But fortunately, as a strong component of the noise in GPS time series is spatially correlated [*Dong et al.*, 2006; *Williams et al.*, 2004; *Langbein*, 2008; **?**], time series of baseline displacements have a much reduced noise, here by a factor of $\sim$2.5 (Fig. 3).

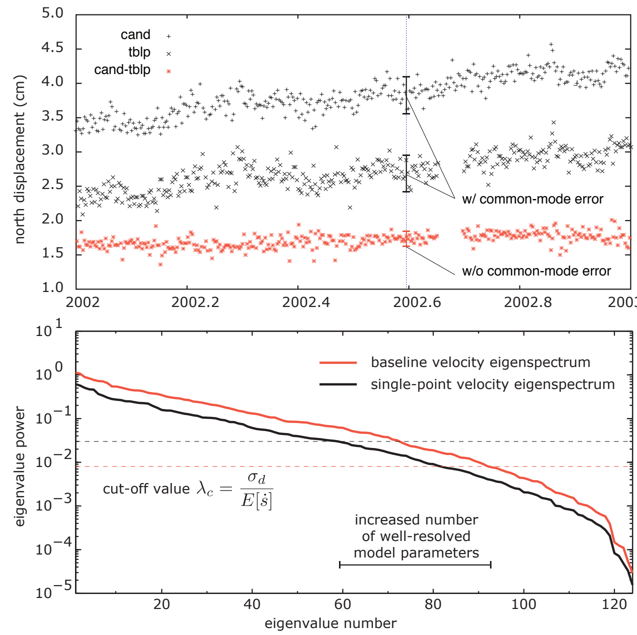


Figure 3. Merits of GPS baseline inversions. (top) raw time series of GPS stations CAND and TBLP, 10 km apart, and their difference. A coherent noise is removed in the differencing process and the noise level of the baseline time series is smaller than the noise in individual time series. (bottom) The eigenspectrum of the Green's function matrix of GPS velocity and GPS baseline velocity. The eigenvalues of the baseline velocity are approximately $\sqrt{2}$ higher than the ones for the velocity and the noise is reduced by at least a factor of two, leading in this case to about 15 more well-resolved parameters.

To determine the baseline velocities and the LOS gradient, we operate as follows. First, we perform of Delaunay triangulation of the point coordinates and identify the unique edges forming the triangular mesh. For

GPS, each point is a GPS station; for InSAR, each point is a coherent reflector. Then, we discard those pairs separated by less than 500 m or by more than 50 km. The difference between the velocities of two connected points forms the basis of the data vector **d** in our inversion. Because the velocity of the reference frame does not contribute to the baseline velocities, the model parameters reduce to

$$\mathbf{m} = \begin{pmatrix} \mathbf{s} \\ \bar{\mathbf{o}}_1 \\ \vdots \\ \bar{\mathbf{o}}_P \end{pmatrix} \quad (13)$$

where the relative range of the LOS is absent of the orbit parameters $\bar{\mathbf{o}}_i$. The forward model of fault slip and dilatation opening corresponds to the difference between predicted displacements at each end of the connecting edge and is marked by an upper bar symbol ($\bar{\ }$). The design matrix is written

$$\mathbf{G} = \begin{pmatrix} w\,\bar{\mathbf{G}}_1^{\mathrm{SAR}} & w\,\bar{\mathbf{G}}_1^{\mathrm{orb}} & & \\ \vdots & & \ddots & \\ w\,\bar{\mathbf{G}}_P^{\mathrm{SAR}} & & & w\,\bar{\mathbf{G}}_P^{\mathrm{orb}} \\ \bar{\mathbf{G}}_1^{\mathrm{GPS}} & & & \\ \vdots & & & \\ \bar{\mathbf{G}}_N^{\mathrm{GPS}} & & & \end{pmatrix} \quad (14)$$

An effect of using baseline data is to increase the number of lines in **G** by a factor of two to three. The rest of the inversion procedure, such as the definition of **A**, **D**, **H** and **h**, is the same as explained in Section 3.1.

The eigenvalues of the design matrix (14) is increased by a factor of about $\sqrt{2}$ on average compared to the ones obtained with definition (7) (Fig. 3). The increase in sensitivity of the design matrix could compensate an increase of the noise of the relative velocity. But because the latter decreases, the result is an improved robustness of the inversion to data noise, and an increased resolution of model parameters. The number of well-resolved model parameters can be estimated from the eigenspectrum of the design matrix using the cut-off eigenvalue [**?**]

$$\lambda_c = \frac{\sigma_d}{E[\dot{s}]} \quad (15)$$

where σ_d is a characteristic noise level of the data and $E[\dot{s}] = 30$ mm/yr is the expected value of fault slip velocity. Using $\sigma_d = 1$ mm/yr for velocity and $\sigma_d = 0.4$ mm/yr for baseline velocity, we find that inversion using baseline velocity can resolve about 15 more model parameters than the one using point-wise velocity within their respective noise level (Fig. 3). In addition, two model parameters per GPS network and one for each InSAR image are removed from the model space. As the inversion of baseline velocity offers significant improvements over the inversion of absolute velocity, we will only present results from the former method. For illustration purposes, we present both the fit to the baseline and absolute velocities. In the later case, the best fitting velocity of the reference frame is estimated in a post-processing step.

3.3. Bootstrap uncertainty estimation

To describe the sensitivity of the model parameters to data noise, we use a bootstrap technique where we perturb the data used in the inversion with a random Gaussian noise. After generating and inverting N modified datasets, we can describe the statistics of a large population of best-fitting models. In the following, we use $N = 100$. This method allows us to describe the model uncertainties including the effect of smoothing, non-negativity constraints and data coverage that are not included in other analytic estimates. For example, if redundancy is large in the data, model parameters may not be largely affected by data noise or potential outliers. To identify the expected amplitude of noise on a dataset basis, we estimate the data variance based on the residuals of our best-fitting model.

4. Interseismic creep before the 2004 Mw 6 Parkfield earthquake

To estimate fault slip rates before the 2004 Mw 6 Parkfield earthquake, we combine GPS data from the BAVU [*d'Alessio et al.*, 2005] (Fig. 4A) and PBO networks (PBO compilation *pbo.final_snf01.vel* released in August 2011 and available at *ftp://data-out.unavco.org*, Fig. 4B). We select the stations located within in a rectangular domain between point coordinates $(-100, -80)$ and $(70, 60)$ expressed in kilometers relative to the reference coordinates (W120.3740, N35.8150). We complement these observations with the interferogram stacks described in Section 2, using ascending and descending ENVISAT acquisitions (Fig. 5). Before inverting, we subsample InSAR data every 1 km and discard areas deemed biased by noise or non-tectonic signals. We use a cut-off value of the noise estimate when available but we also reject some areas by visual inspection.

The surface displacements are assumed to be the re-

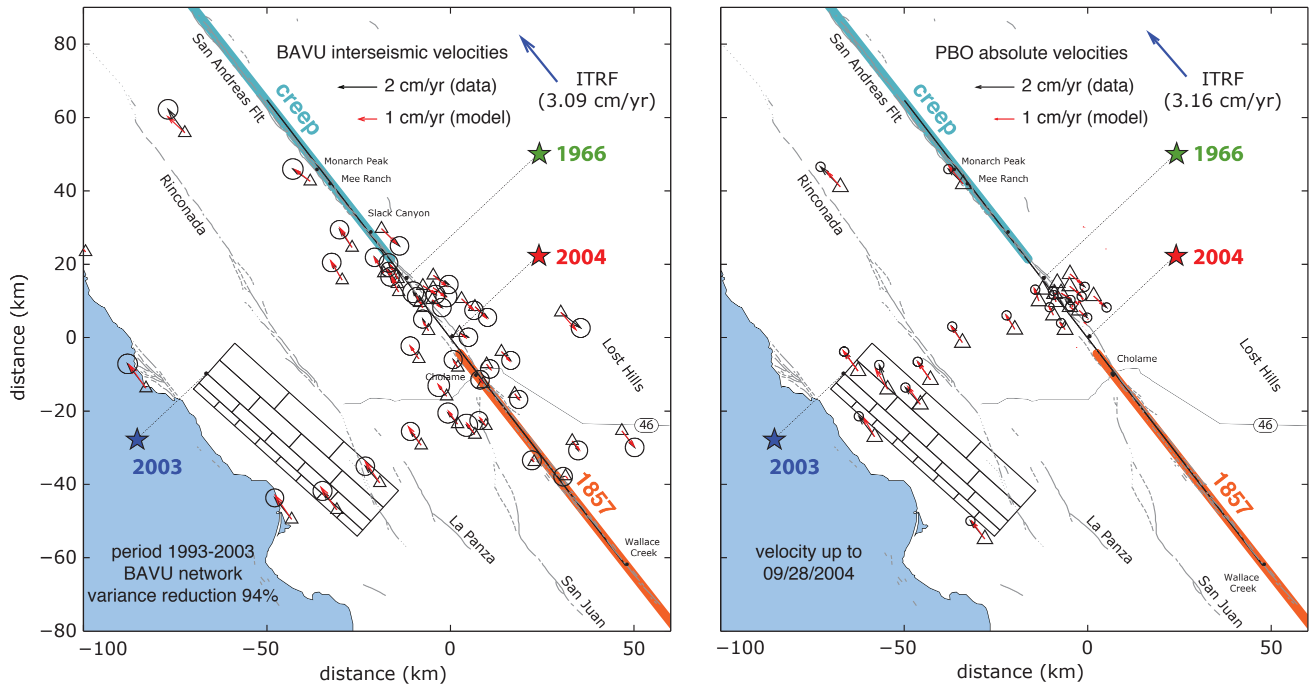


Figure 4. A) Velocity field and forward model at the Bay Area Velocity Unification (BAVU) network [*d'Alessio et al.*, 2005]. The velocity is relative to the SAF and the residual ITRF velocity is shown for reference. Baseline velocity, forward model and residuals are shown in Fig. A7. B) Velocity field and forward model at the EarthScope Plate Boundary Observatory (PBO) network. Baseline velocity, forward model and residuals are shown in Fig. A6.

sult of slip on faults. We also include a dilatation source below Paso Robles to account for reservoir extraction in this area. We consider five fault segments, which are discretized and smoothed independently of each other. They are the San Simeon, Cholame, Parkfield and creeping segments, complemented by the root of the SAF, which consists of seven large patches and one infinitely long fault with a locking depth of $D = 13$ km. Because slip is allowed at shallower depth, this implies that the locking depth is 13 km or less.

With the Cholame, Parkfield and creeping segments, we represent a stretch of the SAF going from Wallace Creek to the south, to 20 km north of Monarch Peak, in the middle of the creeping segment. In a preliminary study, we tested the necessity of allowing creep on the La Panza, Rinconada and Lost Hills faults. Among these, we found that only shallow creep were occurring on the Lost Hill Fault, and that ignoring this effect had little impact on the inversion results elsewhere.

We discretize the fault segments into rectangular patches of varying size, with length and width increasing with depth. The patch sizes on the San Simeon, Cholame and creeping segments are chosen manually so as to obtain a resolution above a critical value of 0.5 (Fig. A9). At Parkfield, we prefer to sample the fault more finely, so as to obtain a better spatial resolution. The geometry of the San Simeon segment is inspired from the work of *Johanson and Bürgmann* [2010] and guided by seismicity and constraints from geodesy. With the chosen discretization, the resolution is above 0.8 and 0.7 in the Cholame and creeping segments, respectively. Resolution is close to one in the top 4 km at Parkfield, but rapidly decrease at greater depth. The San Simeon segment is a dipping fault and with the chosen sampling this gives rise to a much increased resolution, compared to neighboring vertically-dipping segments (Fig. A9).

We invert for fault slip, dilatation opening and other non-tectonic parameters using the method described in Section 3.2, using the relative velocity between pairs of GPS stations and the gradient of InSAR data. The results vary depending mostly on our choice of the weight w put on InSAR and the choice of the underlying rigidity structure. Within a reasonable range, the weight of smoothing influences little the results because patches are overall large and well resolved. We therefore chose a nominally small value to control the intensity

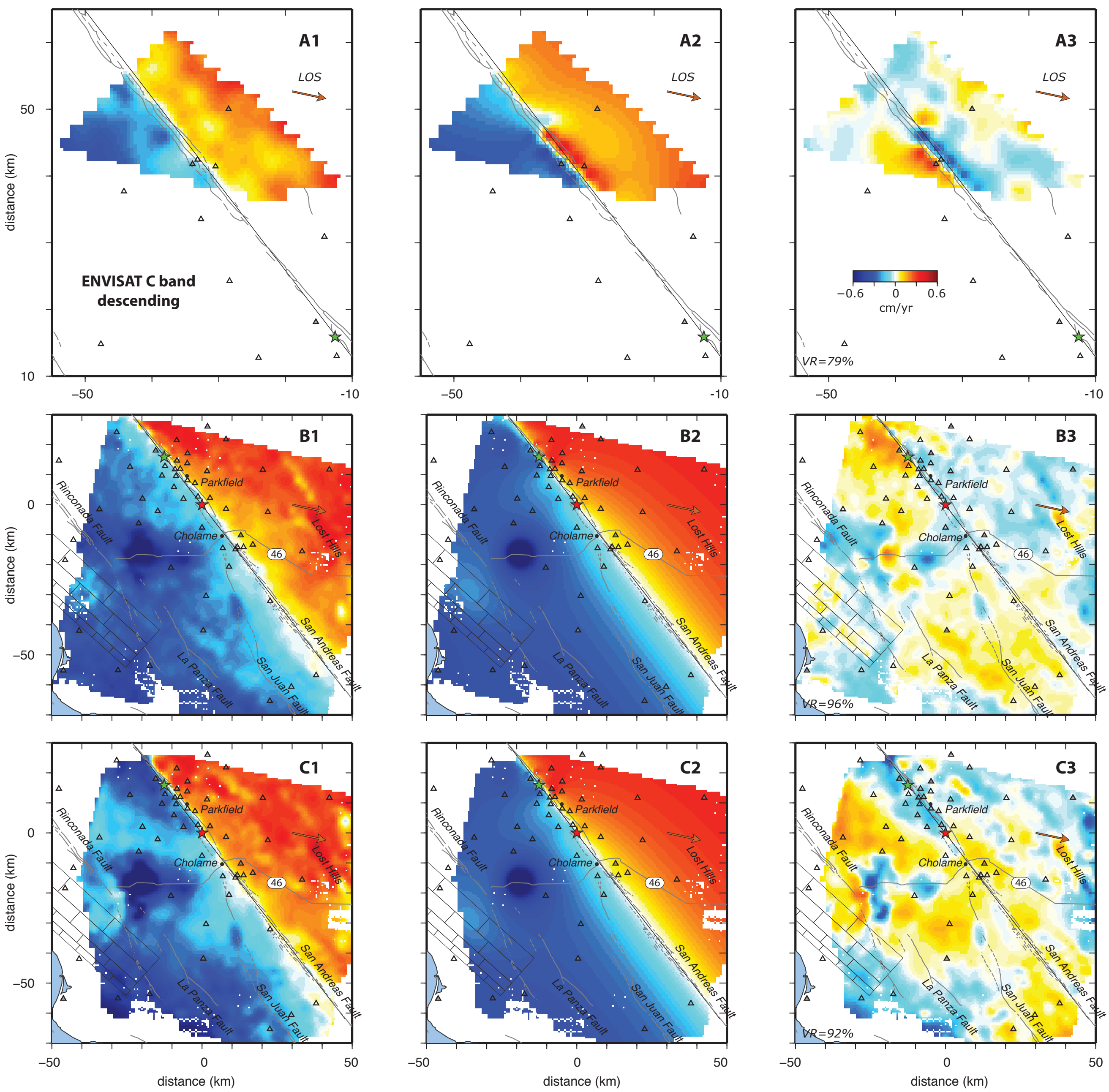


Figure 5. InSAR average velocity (1), forward model (2) and residuals (3) for the period 1992-2004, for A) ALOS data, B) ENVISAT data processed by *De Michele et al.* [2011] and C) ENVISAT data processed with MInTS.

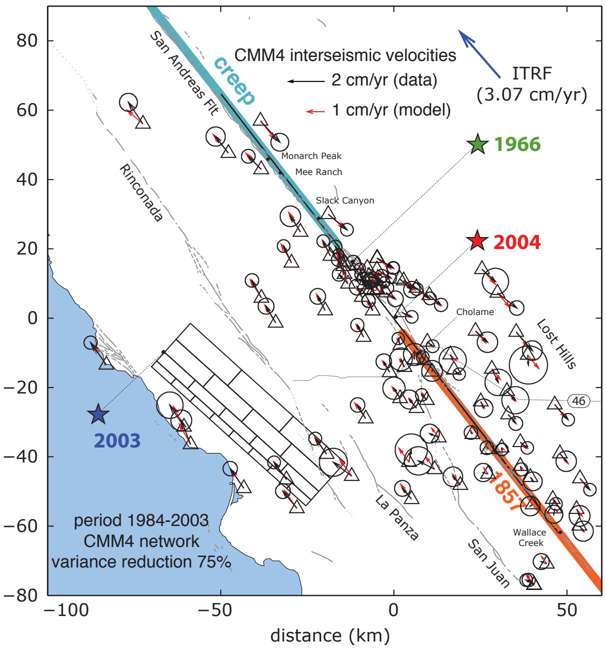


Figure 6. GPS velocity field and forward model at the SCEC Crustal Motion Map 4 (CMM4) compilation network [*Shen et al.*, 2011]. Baseline velocity, forward model and residuals are shown in Fig. A8.

of smoothing.

Our preferred model uses a weight on InSAR data of $w = 0.3$ and a depth dependence of elastic moduli prescribed by the Preliminary Reference Earth Model (PREM) [*Dziewoński and Anderson*, 1981]. GPS data is reduced by 94% and the variance reduction for InSAR varies between 79 and 96%. The fit to the GPS velocity of the BAVU network is shown in Fig. 4A. The forward model explains the data well and there is no systematic patterns in the residuals. The fit to the PBO velocity is shown in Fig. 4B. The same model explains these data well. The horizontal velocity of the ITRF reference frame is a by-product of our inversion and we find that the PBO and BAVU solutions share the same reference frame within 0.7 mm/yr. The observed, modeled and residual baseline velocities are shown in Figs. A7 and A6, for the BAVU and PBO networks, respectively. The residuals in the near field are larger and result probably from local effects (crustal structure, topography, damage zone, and/or parallel fault strands) or to variations of fault slip at finer scales than allowed in the inversion.

The fit to the InSAR data is shown in Fig. 5. For simplicity, we show the absolute velocity and the LOS displacements (as opposed to gradients). As the orbital error is estimated in the inversion, we remove it from the data for illustration purposes. The large wavelength of the measurements is in general well explained, but every interferogram stack shows a pattern of irregular residuals. All of them show residual LOS velocity near the SAF. When comparing the same data processed by *De Michele et al.* [2011] and with MInTS, the residuals show an opposite polarity. This gives us some confidence that some of these residuals are due to atmospheric noise, which is not completely eliminated from the time series processing. The ALOS data also show near-field residuals. These can be due to tropospheric noise or local effects and, as only a few GPS stations are present at this location, it is challenging to assess the origin of this misfit. As GPS and InSAR absolute velocities are not directly included in the inversion, it is remarkable that they can be explained well by a modeled tuned to other types of product.

The 2003 Mw 6.5 San Simeon earthquake ruptured in the period of observation and was captured by 4 out of the 89 interferograms used in the stack. The San Simeon coseismic deformation is largely averaged out in the stacking procedure, but its effect is likely to create an apparent creep on the San Simeon segment (Fig. 7). Based on this result alone, we cannot conclude that the San Simeon is ongoing steady creep rates in the 1992-2003 period.

The inferred creep rate along the SAF can be decomposed based on overall behavior into three segments (Fig. 7). The Cholame segment, to the south, appears locked from its southernmost extension in the model to north of the 2004 Parkfield epicenter. This result is consistent with previous findings [*Harris and Segall*, 1987; *Segall and Harris*, 1987; *Segall and Du*, 1993; *Bakun et al.*, 2005], where no southern termination of the Parkfield locked zone was clearly identified. In the Cholame segment, the depth of the locked zone is poorly resolved and may be deeper. But at least the first 13 km of the fault seem unambiguously locked. The Parkfield segment shows a complex pattern of creep and locked regions. A patchy distribution of shallow creep shows some correlation with seismicity. And in general little slip occurs in the middle of the seismic streaks identified by *Waldhauser et al.* [2004]. Some creep occurs below the bottom seismic streak suggesting that microseismicity at Parkfield marks a transition in fault behavior [*Barbot et al.*, 2012]. Slip rates increase to the north of the 1966 epicenter to transition to the creeping segment. Surface creep at Parkfield is patchy, with isolated regions of fault motion. To the north, the creeping segment exhibit more widespread creep, but the velocity is not uniform, with some isolated areas appearing locked. The sliding rate between the North American and the Pacific plates are inferred to be 33.8±2.2 mm/yr across the SAF, with a slight increase below the creeping segment.

It is informative to consider the uncertainty on fault slip rates based on the bootstrap method described in Section 3.3 (Fig. 7). These uncertainties reflect potential biases caused by data noise or insufficient data coverage on the inversion and include the effect of the regularization and positivity constraints that may not be accounted for by other estimates. The uncertainty is larger at the edge of the domain covered by data, as expected. But at other locations, the Cholame segment exhibits an overall low uncertainty. In the Parkfield segment, uncertainties are large at depth, to the north and at shallow depth. This indicates that data noise can affect the estimation of shallow creep and that it may be more widespread than inferred in the best fitting model. The region surrounded by seismic streaks does not suffer from large uncertainties, so the best fitting model seems robust at this location. The creeping segment suffers from large uncertainties, due mostly to insufficient redundancy in the surface coverage, which leads to the possibility of outliers biasing the results. However, the apparent locked region between Mee Ranch and Monarch Peak does not seem to be much influenced by data noise and the estimated slip rates there seem robust. The uncertainties on the San Simeon segment are the greatest where the highest slip rates take place, but smaller than half the estimated rates, so the results seem to hold at this location.

Inference of deep interseismic velocity from GPS and InSAR data can be subjected to bias from modeling assumptions. For example, comparing with inversions assuming a homogeneous structure, we notice that ignoring the depth dependence of rigidity leads to a systematic reduction of the deep velocity by about 2 mm/yr. Relative weight of InSAR versus GPS data in the inversion influences this result even more. Inversions with InSAR weight $w = 1$ to 10 give rise to estimates of deep velocity from 28.8 to 18.1 mm/yr and systematically worsen the fit to GPS. This indicates that InSAR overall favors slower deep interseismic velocity than GPS. Our preferred value of $w = 0.3$ gives rise to a deep velocity of 33.8 mm/yr, in better agreement with the geological slip rates of *Toké et al.* [2011] and *Sieh and Jahns* [1984].

To explore the variability of fault slip further back in time, we repeat our inversion using the large compilation of surface velocity from campaign GPS surveys compiled in SCEC Crustal Motion Map 4 (CMM4) compiled by *Shen et al.* [2011] and that started as early as 1986, which is 20 years after the 1966 Mw 6 Parkfield earthquake. We find that all the CMM4 data can be explained by a very similar model of fault slip (Fig. A8). This result indicates that the strong coupling from the Cholame to far north into the Parkfield segment was persistent for at least 20 years before 2004.

5. Interseismic creep after the 2004 Mw 6 Parkfield earthquake

The postseismic transient following the 2004 Mw 6 Parkfield earthquake is characterized by rapid afterslip, with a relaxation time scale of about three months, and a slower lower-crustal relaxation with a time scale of one year [*Johanson et al.*, 2006; *Barbot et al.*, 2009; *Bruhat et al.*, 2011]. To avoid the contamination of a strong postseismic transient, we look at geodetic data two years after the 2004 event. Although there is a continuum between post- and interseismic deformation, we focus our attention on this period where the deceleration of surface displacements is less obvious and can be considered the early interseismic stage. In the 2006-2012 period, the PBO network has been greatly augmented compared to its pre-2004 version, consisting of about 80 permanent stations in the domain considered (Fig. 8). The PBO network offers near- and far-field stations, which can place better constraints on the plate convergence rate. The available InSAR catalog offers a complete coverage with multiple look angles of the SAF trace in our domain of interest (Fig. 1). However, in the time period considered, less independent interferograms are available (Table 1) and we can expect more contamination by tropospheric noise. Furthermore, the look direction of ascending orbits is almost perpendicular to the SAF so the ENVISAT descending stack is the most sensitive to SAF displacements. We discard ALOS LOS data north-east of Coalinga, Lost Hills and the 5 freeway due to the agricultural activity in Central Valley that reduces coherence and increases non-tectonics signals. Some data in all stacks are also ignored based on a threshold on an estimate of the signal-to-noise ratio obtained from MInTS. The increased coverage allows us to sample the creeping segment more finely while keeping a high inversion resolution (Fig. A11).

We simultaneously invert the PBO average velocity field in the 2006-2012 period (Fig. 8) and data from four InSAR tracks (Fig. 9). We use the same inversion parameters as for the 1992-2004 period, including smoothing strength, relative InSAR weight and the rigidity structure. The GPS baseline velocity is reduced

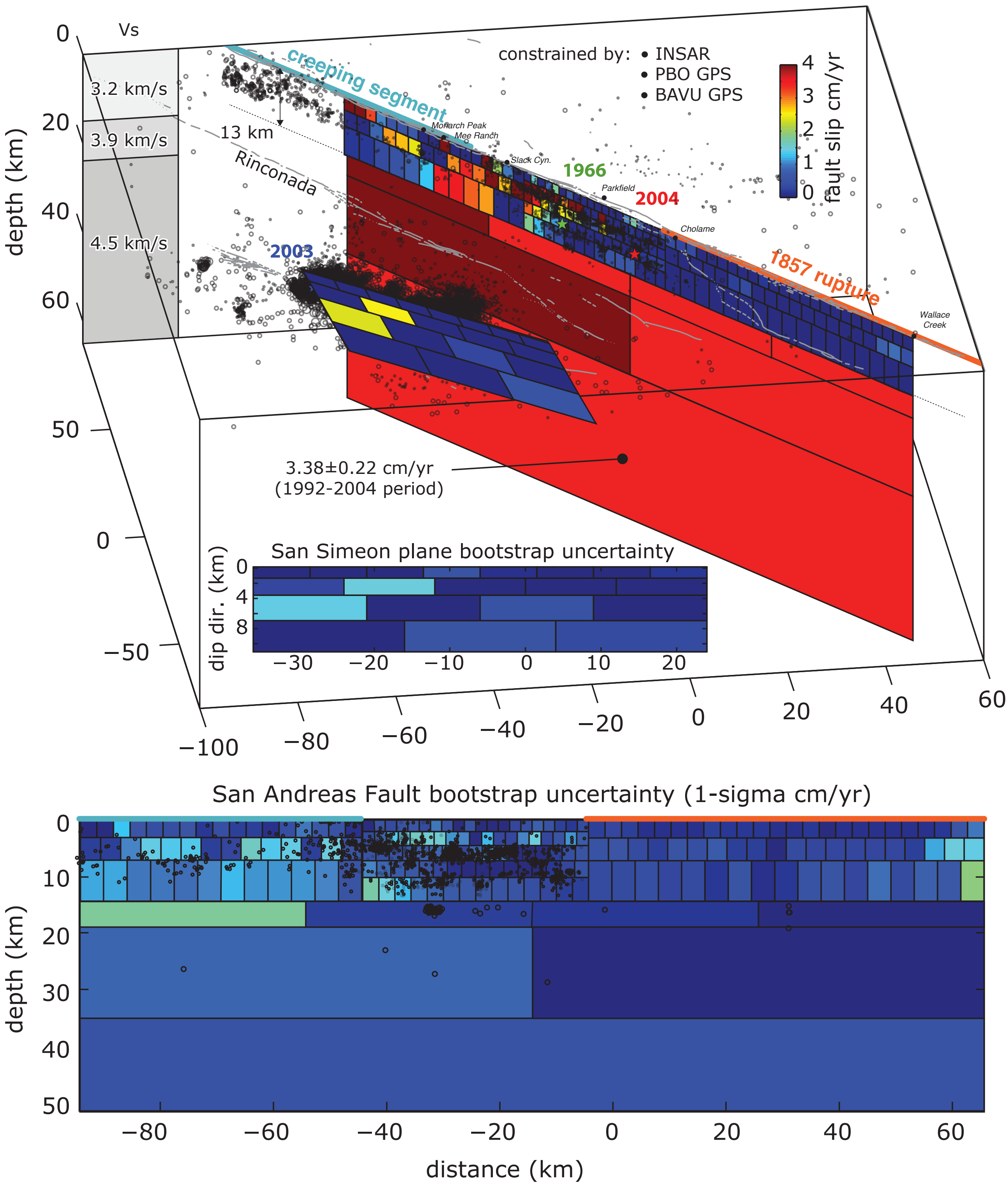


Figure 7. Spatial distribution of interseismic fault creep in the 1999-2004 period from the joint inversion of GPS and InSAR data. The 1-σ uncertainty from bootstrap analysis indicates large uncertainties in the creeping segment.

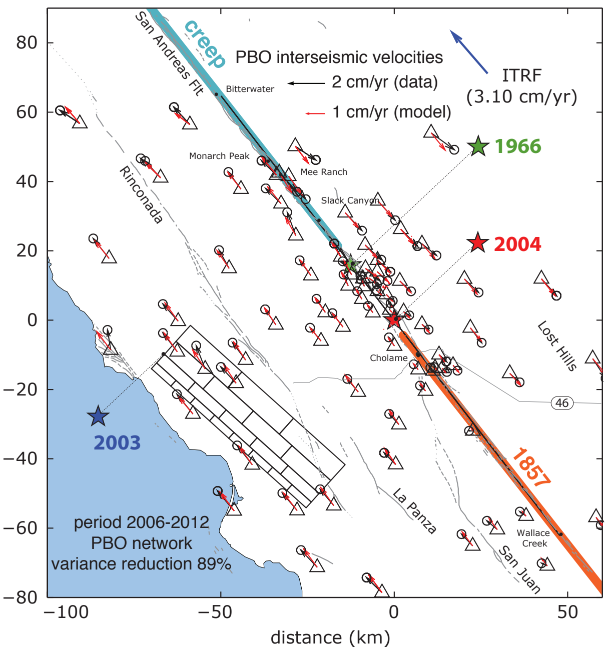


Figure 8. Velocity fields and forward model at the PBO network in the 2006-2012 period. The velocity is relative to the SAF and the residual ITRF velocity is shown for reference. Baseline velocity, forward model and residuals are shown in Fig. A10.

by 89%. The largest residuals for the absolute velocity occur in the far field of the SAF, close to the San Simeon segment or near the boundary of our domain of observation. The most apparent misfit is in the SAF fault-perpendicular direction, probably due to edge effects, i.e., the fact that we ignore the deformation in the northern creeping segment, which affects some stations in our domain of interest. The fit to the GPS near-field station is good, with no obvious patterns in the residuals.

The InSAR gradient data is reduced by 45% to 79% in the ascending direction and by 82% in the descending direction (Fig. 9), consistent with sensitivity of ascending and descending orbits to strike-slip on the SAF. While some tectonic signal may be better explained using a smaller fault discretization or a more realistic material property structure, we attribute most of the unexplained signal to residual tropospheric noise, particularly for the ascending tracks. The large wavelength signal is well reproduced by the model, but some scattered residuals appear for all tracks. The ascending ENVISAT and ALOS data can be most easily compared as their horizontal look direction is similar (yet, ENVISAT is more sensitive to vertical deformation). The large ALOS residuals that correlate with the SAF cannot be found in the ascending ENVISAT residuals. Some residuals localize near the SAF in the ENVISAT descending track, but as similar patterns can be sometimes found in the ALOS data but not in other datasets, it is likely that some residuals may be due to troposphere correlated with topography. There are large residuals in the Cholame segment in the ALOS data. These patterns do not change polarity across the fault and may not be of tectonic origin.

The pattern of interseismic slip rate in the 2006-2012 period is shown in Fig. 10. The Cholame segment seems locked from its southern extension to 8 km south of Cholame. A deep locked segment appear at the northern extension, next to the Parkfield segment, but the uncertainties from the bootstrap analysis indicate that some creep may in fact occur there. Bootstrap analysis also indicates the possibility of creep below Wallace Creek, but these uncertainties are due to side effects as data coverage is sparser at the boundary of the domain of interest.

In the Parkfield segment, deep creep takes place below the bottom seismic streak, but not in the region surrounded by microseismicity. The bootstrap analysis indicates that little slip occurs there, regardless of reasonable noise added to the data before inverting, so this result seems robust and little affected by data noise. Some shallow creep also takes place in the top 4 km layer, particularly to the north, at the beginning of the creeping section, where it correlates with microseismicity.

In the creeping segment, the velocity seems larger than plate rate, which is indicative of transient behavior. Some areas, for example below Monarch Peak or Slack Canyon appear locked, but the bootstrap analysis indicates that some slip may occur there. Considering the best fitting inversion and the bootstrap analysis together, the transition between the Parkfield and creeping segments seem marked by an area of reduced slip rates below 6 km depth. As this region did not produce large earthquakes in recorded history, it likely experiences cycles of accelerated and reduced aseismic slip. Overall the slip rates on the creeping segment are not uniform, compatible with the results of *Rolandone et al.* [2008] obtained with GPS data alone, and those of [*Ryder and Bürgmann*, 2008] obtained from InSAR analysis.

Some creep appears to occur on the San Simeon segment, concentrating in the same area that moves in the pre-2004 period. These results are subject to much variability, given the bootstrap uncertainty, and are sup-

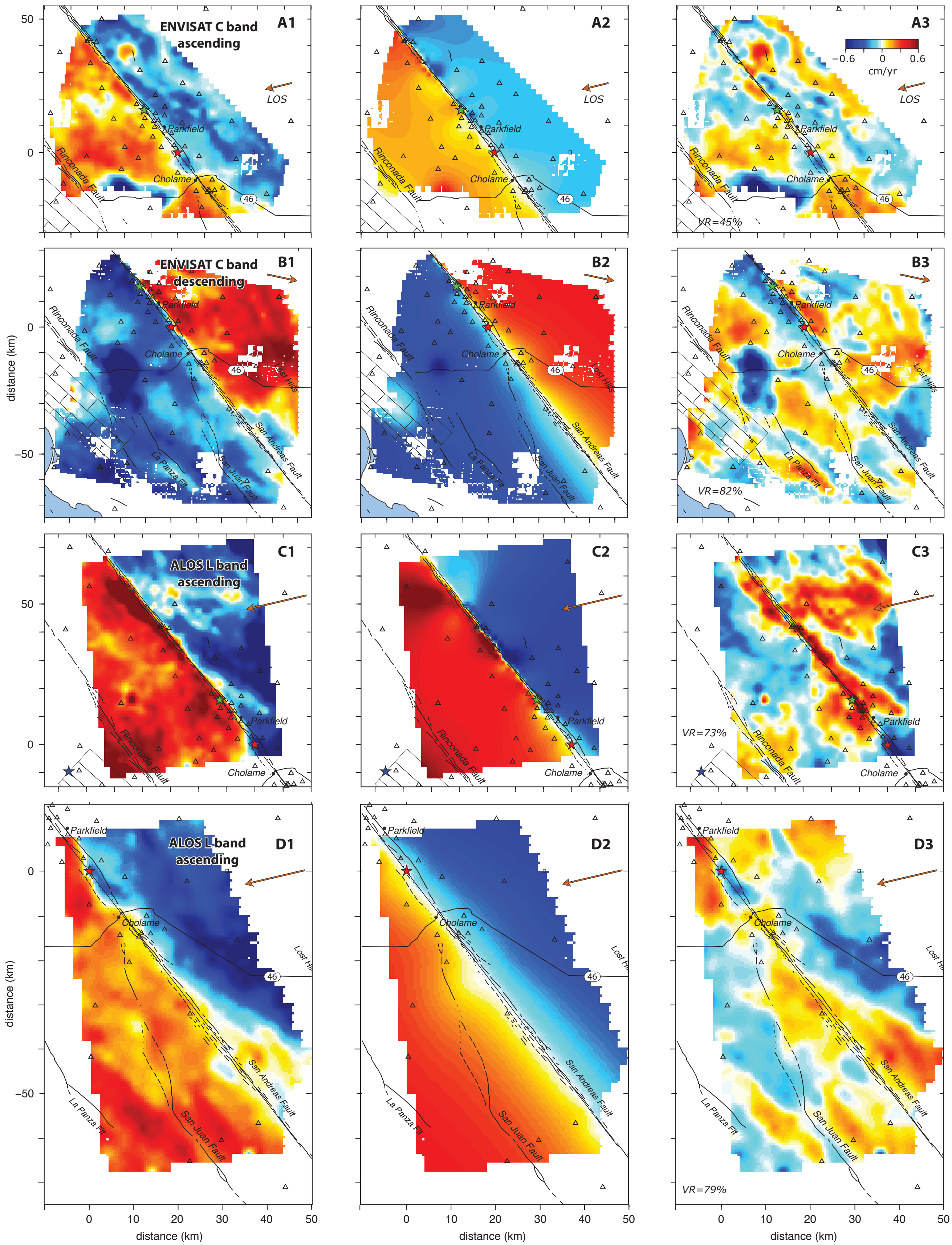


Figure 9. InSAR average velocity (1), forward model (2) and residuals (3) for the period 2006-2012, for A-B) ENVISAT data, C-D) ALOS data. All processed with MInTS.

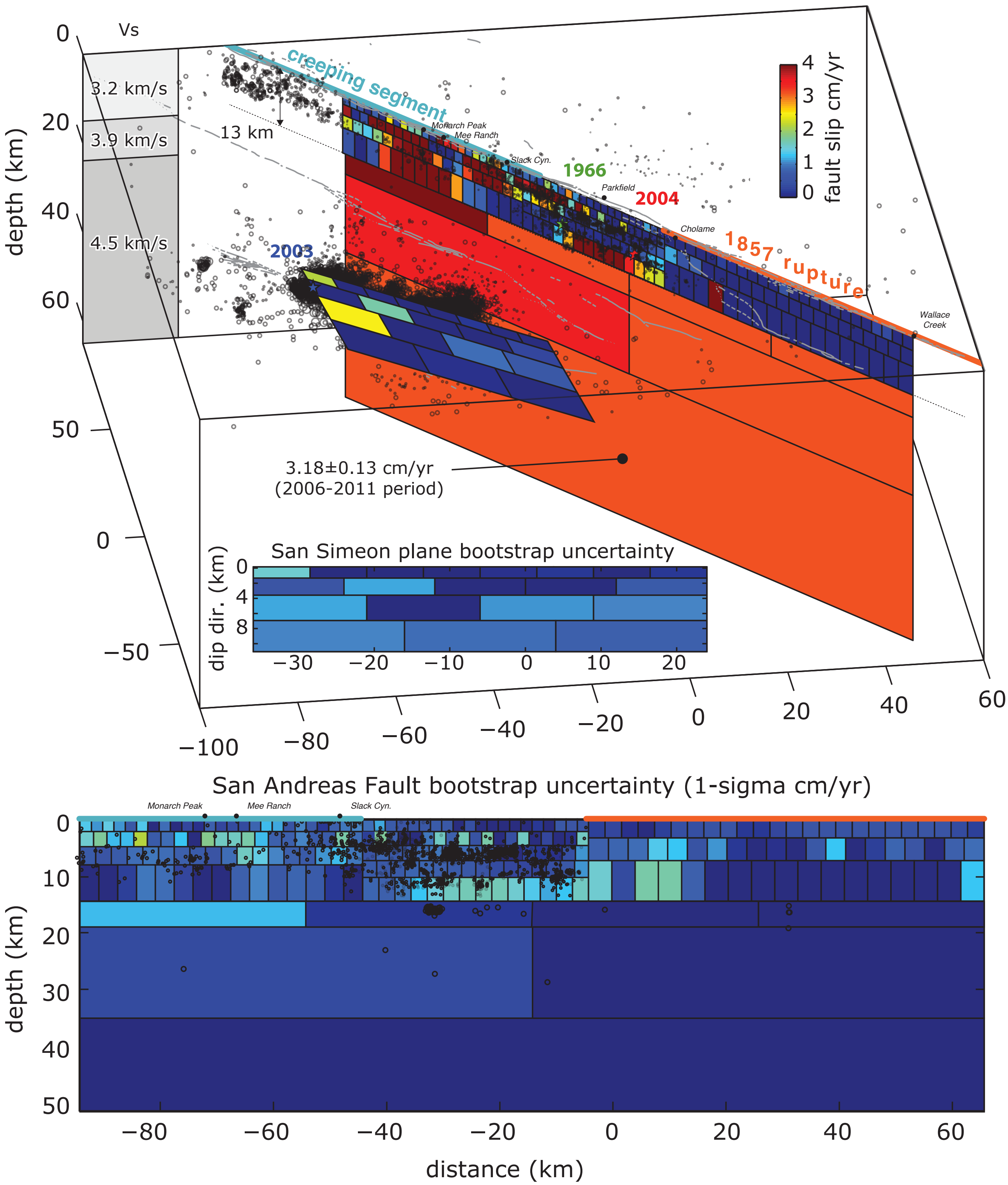


Figure 10. Spatial distribution of interseismic fault creep in the 2006-2012 period from the joint inversion of GPS and InSAR data. The 1-σ uncertainty from bootstrap analysis indicates larger uncertainties at the bottom of the Parkfield segment.

ported only by some GPS stations and the ENVISAT descending data, so they should be interpreted with caution.

The inferred convergence rate between the North American and Pacific plates in the 2006-2011 period is 31.8 ± 1.3 mm/yr, which agrees with the results for the pre-2004 period within the error bounds.

6. Discussion

The distribution of velocity across the SAF shows notable differences at key different periods of the earthquake cycle. These differences may shed light on important questions regarding the mechanics of earthquake generation in the region. In particular the kinematics may inform how earthquakes in the Parkfield region interact with neighboring segments and how the segmentation of the SAF into different segments operates over time. We discuss some implications below.

6.1. The evolution of apparent coupling at the transition between the Cholame and Parkfield segments: A southern termination of the Parkfield seismogenic zone?

The seismic behavior of the Parkfield segment was under much scrutiny to understand the kinematics of the recurrence of Mw 6 events. Using geodetic [*Harris and Segall*, 1987; *Segall and Harris*, 1987; *Segall and Du*, 1993; *Murray et al.*, 2001; *Bakun et al.*, 2005; *Murray and Segall*, 2005; *Murray and Langbein*, 2006; *Barbot et al.*, 2009; *Bennington et al.*, 2011] or seismological [*Niu et al.*, 2003; *Custódio et al.*, 2005; *Uchide et al.*, 2009; *Ziv*, 2012] data, many aspects of the inter-, co-, and postseismic deformation have been documented. In particular, there is a possibility that the streaks of microseismicity described by *Waldhauser et al.* [2004] and *Thurber et al.* [2006] represent the boundary of the seismogenic zone at Parkfield [*Barbot et al.*, 2012].

Recently, *Barbot et al.* [2012] presented a physical model of the earthquake cycle based on rate-and-state friction that explains the many seismological and geodetic observations, including some variability of recurrence times of Mw 6 earthquakes, and the change of hypocenter location between 1966 to 2004. To explain the nucleation of earthquakes near the 2004 hypocenter and the termination of previous ones to the south, the model assumes a termination of the seismogenic zone a few kilometers south of the 2004 hypocenter. Previous models of fault coupling based on geodetic data do not indicate a clear boundary between the Parkfield and Cholame segments, and high coupling is thought to be continuous from Wallace Creek and beyond to the location of the 1966 hypocenter. Our analysis of the geodetic data from 1986 to 2004 and 1992 to 2004 confirms this view. However, the pattern of surface displacements in the 2006-2012 period indicates the presence of a 10 km long region of low coupling immediately south of the 2004 hypocenter. As our inversion results show similar deep slip rates in the two periods of observation, we are confident that the difference between the two observed behaviors is real and not an artifact of data sampling or noise.

We interpret these results as an indication for measurably decelerating creep during the interseismic cycle. This interpretation is broadly compatible with the expected behavior of rate-and-state friction faults, which shows a continuous transition between accelerated postseismic transient and interseismic creep [*Rice*, 1993; *Lapusta et al.*, 2000; *Barbot et al.*, 2012; **?**]. The great 1857 earthquake is thought to have ruptured the SAF from Cajon Pass to Bitterwater, in the creeping segment, breaking through the Parkfield segment. The implication of this scenario is that the area of low coupling imaged in the 2006-2012 period represents a soft boundary between the Cholame and Parkfield segments. This area would have a combination of frictional properties and confining pressure that is capable of arresting a series of Mw 6 earthquakes in the Parkfield segment, but that would allow larger ruptures to propagate through both segments on occasions. If the 1857 earthquake was in fact preceded by foreshocks in the Parkfield segment a few hours before the rupture [*Sieh*, 1978a, b; *Toppozada et al.*, 2002], it would mean that the area of low coupling can arrest most events rupturing in the Parkfield segment, and that ruptures north and south of this boundary always represent separate seismic events.

6.2. Microseismicity and the seismogenic zone at Parkfield

The remarkably organized spatio-temporal distribution of microseismicity at Parkfield suggests that clusters of small earthquakes delineate the boundary between areas of different friction properties, and in particular surround the seismogenic zone. This interpretation is supported by analysis of seismic data from the 2004 Parkfield earthquake by *Ma et al.* [2008] and *Uchide et al.* [2009] that show that most of the seismic slip can be confined in the area circumscribed by microseismicity. If slip on the Parkfield segment is controlled by rate- and state-dependent friction [*Dieterich*, 1979; *Ruina*, 1983; *Rice and Ruina*, 1983; *Marone*, 1998; *Rice et al.*, 2001], and if the seismogenic zone is a large velocity-weakening asperity with a small enough nucleation size surrounded by a velocity-strengthening domain, the areas of the fault that creep during the interseismic and postseismic periods and those that slip seismically should be mostly mutually exclusive, except around the boundaries between velocity-weakening and velocity-strengthening areas [*Tse and Rice*, 1986; *Marone et al.*, 1991; *Lapusta et al.*, 2000; *Lapusta and Liu*, 2009; *Kaneko et al.*, 2010]. The velocity distribution in the 2006-2012 time interval shows a large region of high coupling encircled by microseismicity and completely surrounded by fault estate of presumably velocity-strengthening friction slipping with a higher velocity (Fig. 10). This result gives additional credit to the hypothesis that the background seismicity marks the boundary of the seismogenic zone at Parkfield and that the seismogenic zone may be a homogeneous region of velocity-weakening friction.

6.3. Reasons for apparent high coupling at the Cholame/Parkfield transition zone

It may be surprising that the low coupling area south of Parkfield in the 2006-2012 period is not detectable before 2004. It is possible that the surrounding locked areas in the Parkfield and Cholame segments affect its behavior by reducing its velocity. If the region was pinned by surrounding locked domains, it could not slip at the same velocity as it would otherwise free of boundaries. Our results indicate that, in this context, analyses of surface deformation late in the earthquake cycle may over-estimate the area of plate coupling. Another explanation is that there would be significant weakening in afterslip periods followed by marked hardening during the interseismic period (the velocity averaging to plate rate over the period separating seismic events) allowing the creep to be much lower than plate rate late in the interseismic period. This behavior is not expected in simple models of slip evolution based on rate-and-state friction, but a number of potentially important factors - additional mechanisms of fault weakening such as thermal pressurization and flash heating, heterogeneities in friction, dilatancy effects - may allow significant excursions of slip velocity above and below plate rate [*Bilek and Lay*, 2002; *Toro et al.*, 2004; *Nakatani and Scholz*, 2004; *Hillers et al.*, 2006; *Noda*, 2008; *Beeler et al.*, 2008; *Brantut et al.*, 2008; *Rubin*, 2008; *Sone and Shimamoto*, 2009; *Fukuyama and Mizoguchi*, 2010; *Segall et al.*, 2010; *Shibazaki et al.*, 2011; *Faulkner et al.*, 2011].

6.4. Variation of slip rates in the creeping segment

Despite a greater sensitivity to noise than other regions and often insufficient data coverage, we observe notable variations of slip velocity in the creeping segment, as previously highlighted by *Murray and Segall* [2005] and *Ryder and Bürgmann* [2008]. To accommodate long-term relative motion between the North American and the Pacific plates, the area of slip deficit will have to accelerate eventually. Similarly, areas slipping at a larger velocity than plate rate must eventually slow down to satisfy the long-term slip rate. It is possible that these areas of accelerated and slower slip represent “ghost transients” [**?**], i.e., periods of anomalous velocity that is longer than the period of observation. Long periods of transient deformation can be due to viscoelastic relaxation following large earthquakes. But at shallow depth, oscillatory slip velocity can be the result of spontaneous behavior of rate-and-state friction faults within a certain range of friction parameters [*Ruina*, 1983]. Velocity-weakening friction with relatively large critical nucleation size (the so-called h^* parameter), which occurs under low effective confining pressure or large critical weakening distance can produce large excursions of velocity above and below plate rate without external perturbations [*Ruina*, 1983; *Scholz*, 1998; **?**]. A better understanding of these processes will require a finer temporal coverage of the evolution of deformation for an extended period.

7. Conclusions

The accumulation of geodetic measurements in the last 30 years in a broad region around the Parkfield segment of the San Andreas Fault allows us to compare the kinematic behavior of the fault at key periods of the earthquake cycle, and to address fundamental questions about fault segmentation and the generation of earthquakes in the area.

A combination of synthetic aperture radar and GPS measurements indicates that the Cholame and Parkfield segments formed a continuous domain of high coupling for at least 20 years before the 2004 Mw 6 earthquake, consistent with previous analyses. However, the Parkfield Mw 6 earthquakes propagating southward terminated 20 km to the south into the locked zone without propagating into the Cholame segment. Additionally, the latest Mw 6 earthquake of 2004 initiated close to where previous events arrested, indicating a dramatic change in fault properties in this area.

Analysis of geodetic data in the early stage of the interseismic period - after 2006, two years after the latest Parkfield earthquake - shows that the area of most coseismic slip during the 2004 rupture is now locked, and singles out a large domain of low coupling south of the Parkfield segment, consistent with the presence of an obstacle between the Cholame and Parkfield segments, and demonstrates a pronounced segmentation of the San Andreas Fault into two separate domains at this latitude. The strong variations of velocity during the interseismic period is surprising and may be the result of a form of enhanced hardening during the interseismic period; or perhaps more simply caused by the locked Cholame segment to the south.

These observations indicate that the Parkfield seismogenic zone may be formed by a single area of velocity-weakening friction - with small enough critical nucleation size - surrounded north and south by fault estate with velocity-strengthening friction. This also implies that the stable distribution of micro-seismicity at Parkfield forms a marker for the transition of friction properties from weakening to strengthening. If this scenario holds in other regions, analysis of microseismicity in combination to geodesy may offer a great tool to map lateral variations of fault properties in active tectonic areas.

Acknowledgements

This work was supported by the Tectonics Observatory and the Keck Institute for Space Studies at the California Institute of Technology. We would also like to thank Professors Jean-Philippe Avouac and Nadia Lapusta for useful discussions and comments.

References

Bakun, W., and T. McEvilly, Recurrence models and Parkfield, California, earthquakes, *J. Geophys. Res.*, *89*(B5), 3051–3058, 1984.

Bakun, W. H., and A. G. Lindh, The Parkfield, California, Earthquake Prediction Experiment, *Science*, *229*(4714), 619–624, 1985.

Bakun, W. H., et al., Implications for prediction and hazard assessment from the 2004 Parkfield earthquake, *Nature*, *437*(7061), 969–974, 2005.

Barbot, S., Y. Fialko, and Y. Bock, Postseismic Deformation due to the Mw 6.0 2004 Parkfield Earthquake: Stress-Driven Creep on a Fault with Spatially Variable Rate-and-State Friction Parameters, *J. Geophys. Res.*, *114*(B07405), doi:10.1029/2008JB005748, 2009.

Barbot, S., N. Lapusta, and J. P. Avouac, Under the hood of the earthquake machine: Towards predictive modeling of the seismic cycle, *Science*, *336*(6082), 707–710, 2012.

Beeler, N. M., T. E. Tullis, and D. L. Goldsby, Constitutive relationship and physical basis of fault strength due to flash heating, *J. Geophys. Res.*, *113*(B01401), doi: 10.1029/2007JB004988, 2008.

Bell, M., J. R. Elliott, and B. Parsons, Interseismic strain accumulation across the Manyi fault (Tibet) prior to the 1997 Mw 7.6 earthquake, *Geophys. Res. Lett.*, *38*(L24303), 6, doi:10.1029/2011GL049762, 2011.

Ben-Zion, Y., J. R. Rice, and R. Dmowska, Interaction of the San Andreas Fault creeping segment with adjacent great rupture zones and earthquake recurrence at Parkfield, *J. Geophys. Res.*, *98*(B2), 2135–2144, doi: 10.1029/92JB02154, 1993.

Bennington, N., C. Thurber, K. Feigl, and J. Murray-Moraleda, Aftershock distribution as a constraint on the geodetic model of coseismic slip dor the 2004 Parkfield earthquake, *Pure Appl. Geophys*, *168*, 1553–1565, 2011.

Bilek, S. L., and T. Lay, Tsunami earthquakes possibly widespread manifestations of frictional conditional stability, *Geophys. Res. Lett.*, *29*(14, 1673), 4, doi: 10.1029/2002GL015215, 2002.

Bock, Y., et al., Southern California Permanent GPS Geodetic Array: Continuous measurements of regional crustal deformation between the 1992 Landers and 1994 Northridge earthquakes, *J. Geophys. Res.*, *102*(B8), 18,01318,033, 1997.

Brantut, N., A. Schubnel, J.-N. Rouzaud, F. Brunet, and T. Shimamoto, High-velocity frictional properties of a clay-bearing fault gouge and implications for earthquake mechanics, *J. Geophys. Res.*, *113*(B10401), 18, 2008.

Bruhat, L., S. Barbot, and J. P. Avouac, Evidence for postseismic deformation of the lower crust following the 2004 Mw6.0 Parkfield earthquake, *J. Geophys. Res.*, *116*, 10, 2011.

Burford, R. D., and P. W. Harsh, Slip on the San Andreas Fault in central California from alignment array surveys, *Bull. Seism. Soc. Am.*, *70*, 1233–1261, 1980.

Cavalié, O., C. Lasserre, M.-P. Doin, G. Peltzer, J. Sun, X. Xu, and Z.-K. Shen, Measurement of interseismic strain across the Haiyuan fault (Gansu, China), by InSAR, *Earth Plan. Sc. Lett.*, *275*(3-4), 246–257, 2008.

Chen, C., and H. Zebker, Phase unwrapping for large sar interferograms: statistical segmentation and generalized network models, *IEEE Trans. Geosci. Rem. Sens.*, *40*(8), 1709–1719, 2002.

Custódio, S., P. Liu, and R. J. Archuleta, The 2004 Mw6.0 Parkfield, California, earthquake: Inversion of near-source ground motion using multiple data sets, *Geophys. Res. Let.*, *32*(L23312), 4 PP., 2005.

d'Alessio, M., I. A. Johanson, R. Bürgmann, D. A. Schmidt, and M. H. Murray, Slicing up the San Francisco Bay Area: Block kinematics and fault slip rates from GPS-derived surface velocities, *J. Geophys. Res.*, *110*(B06403), 19, doi: 10.1029/2004JB003496, 2005.

De Michele, M., D. Raucoules, F. Rolandone, P. Briole, J. Salichon, A. Lemoine, and H. Aochi, Spatiotemporal evolution of surface creep in the Parkfield region of the

San Andreas Fault (1993-2004) from synthetic aperture radar, *Earth Planet. Sci. Lett.*, *308*(1-2), 141150, 2011.

Dieterich, J. H., Modeling of rock friction 1. experimental results and constitutive equations, *J. Geophys. Res.*, *84*(B5), 2161–2168, 1979.

Dong, D., P. Fang, Y. Bock, F. Webb, L. Prawirodirdjo, S. Kedar, and P. Jamason, Spatiotemporal filtering using principal component analysis and Karhunen-Loeve expansion approaches for regional GPS network analysis, *J. Geophys. Res.*, *111*(B03405), doi:10.1029/2005JB003806, 2006.

Dziewoński, A. M., and D. L. Anderson, Preliminary reference earth model, *Phys. Earth Plan. Int.*, *25*(4), 297–356, 1981.

Elliott, J. R., J. Biggs, B. Parsons, and T. J. Wright, InSAR Slip Rate Determination on the Altyn Tagh Fault, N. Tibet, in the Presence of Topographically Correlated Atmospheric Delays, *Geophys. Res. Lett.*, *35*(L12309), 5, doi:10.1029/2008GL033659, 2011.

Faulkner, D. R., T. M. Mitchell, J. Behnsen, T. Hirose, and T. Shimamoto, Stuck in the mud? earthquake nucleation and propagation through accretionary forearcs, *Geophys. Res. Lett.*, *38*(L18303,), 5, 2011.

Fay, N. P., and E. D. Humphreys, Fault slip rates, effects of elastic heterogeneity on geodetic data, and the strength of the lower crust in the Salton Trough region, southern California, *J. Geophys. Res.*, *110*(B09401), 2005.

Fukuyama, E., and K. Mizoguchi, Constitutive parameters for earthquake rupture dynamics based on high-velocity friction tests with variable sliprate, *Int. J. Frac.*, *163*(1-2), 15–26, 2010.

Goldstein, R. M., and C. L. Werner, Radar interferogram filtering for geophysical applications, *Geophys. Res. Lett.*, *25*(21), 4035, 1998.

Harris, R. A., and P. Segall, Detection of a locked zone at depth on the Parkfield, California, segment of the San Andreas Fault, *J. Geophys. Res.*, *92*(B8), 7945–7962, 1987.

Hetland, E. A., P. Musé, M. Simons, Y. N. Lin, P. S. Agram, and C. J. DiCaprio, Multiscale InSAR Time Series (MInTS) analysis of surface deformation, *J. Geophys. Res.*, *117*(B02404), 17, doi:10.1029/2011JB008731, 2012.

Hillers, G., Y. Ben-Zion, and P. M. Mai, Seismicity on a fault controlled by rate- and state-dependent friction with spatial variations of the critical slip distance, *J. Geophys. Res.*, *111*, doi:10.1029/2005JB003859, 2006.

Huiskamp, G., Difference formulas for the surface Laplacian on a triangulated surface, *J. Comp. Physics*, *95*(2), 477–496, 1991.

Ito, Y., K. Obara, K. Shiomi, S. Sekine, and H. Hirose, Slow earthquakes coincident with episodic tremors and slow slip events, *Science*, *315*, 503–506, 2007.

Johanson, I. A., and R. Bürgmann, Creep and quakes on the northern transition zone of the San Andreas Fault from GPS and InSAR data, *Geophys. Res. Lett.*, *32*(L14306), 5, 2005.

Johanson, I. A., and R. Bürgmann, Coseismic and postseismic slip from the 2003 San Simeon earthquake and their effects on backthrust slip and the 2004 Parkfield earthquake, *J. Geophys. Res.*, *115*(B07411), 18, 2010.

Johanson, I. A., E. J. Fielding, F. Rolandone, and R. Bürgmann, Coseismic and postseismic slip of the 2004 parkfield earthquake from space-geodetic data, *Bulletin of the Seismological Society of America*, *96*(4B), S269–S282, 2006.

Johnson, K. M., and P. Segall, Viscoelastic earthquake cycle models with deep stress-driven creep along the San Andreas Fault system, *J. Geophys. Res.*, *109*(B10403), doi:10.1029/2004JB003096, 2004.

Jolivet, R., R. Cattin, N. Chamot-Rooke, C. Lasserre, and G. Peltzer, Thin-plate modeling of interseismic deformation and asymmetry across the Altyn Tagh fault zone, *Geophys. Res. Lett.*, *35*(L02309), 5, doi:10.1029/2007GL031511, 2008.

Jolivet, R., C. Lasserre, M.-P. Doin, S. Guillaso, G. Peltzer, R. Dailu, J. Sun, Z.-K. Shen, and X. Xu, Shallow creep on the Haiyuan Fault (Gansu, China) revealed by SAR Interferometry, *J. Geophys. Res.*, *117*(B06401), 18, doi:10.1029/2011JB008732, 2012.

Kaneko, Y., J.-P. Avouac, and N. Lapusta, Towards inferring earthquake patterns from geodetic observations of interseismic coupling, *Nature Geoscience*, *3*, 363–369, 2010.

Kositsky, A. P., and J. P. Avouac, Inverting geodetic time series with a principal component analysis-based inversion method, *J. Geophys. Res.*, *115*, 2010.

Langbein, J., Noise in GPS displacement measurements from Southern California and Southern Nevada, *J. Geophys. Res.*, *113*(B05405), doi:10.1029/2007JB005247, 2008.

Lapusta, N., and Y. Liu, Three-dimensional boundary integral modeling of spontaneous earthquake sequences and aseismic slip, *J. Geophys. Res.*, *114*(B09303), 25 PP., 2009.

Lapusta, N., J. R. Rice, Y. BenZion, and G. Zheng, Elastodynamics analysis for slow tectonic loading with spontaneous rupture episodes on faults with rate- and state-dependent friction, *J. Geophys. Res.*, *105*(B10), 23,765–23,789, 2000.

Lindsey, E., and Y. Fialko, Geodetic slip rates in the southern San Andreas fault system: Effects of elastic heterogeneity and fault geometry, *J. Geophys. Res.*, 2012.

Liu, Y., and J. R. Rice, Aseismic slip transients emerge spontaneously in three-dimensional rate and state modeling of subduction earthquake sequences, *J. Geophys. Res.*, *110*(B08307), doi:10.1029/2004JB003424, 2005.

Lundgren, P. E., A. Hetland, Z. Liu, and E. J. Fielding, Southern San Andreas-San Jacinto fault system slip rates estimated from earthquake cycle models constrained by GPS and interferometric synthetic aperture radar observations, *J. Geophys. Res.*, *114*(B02403), 18, 2009.

Ma, S., S. Custódio, R. J. Archuleta, and P. Lui, Dynamic modeling of the 2004 Mw 6.0 Parkfield, California, earthquake, *J. Geophys. Res.*, *113*(B02301), 16, 2008.

Marone, C., C. H. Scholz, and R. Bilham, On the mechanics of earthquake afterslip, *J. Geophys. Res.*, *96*, 8441–8452, 1991.

Marone, C. J., Laboratory-derived friction laws and their application to seismic faulting, *An. Rev. Earth Planet. Sc.*, *26*, 643696, 1998.

Miyazaki, S., P. Segall, J. McGuire, T. Kato, and Y. Hatanaka, Spatial and temporal evolution of stress and slip rate during the 2000 Tokai slow earthquake, *J. Geophys. Res.*, *111*(B03409), 17, doi:10.1029/2004JB003426, 2006.

Murray, J., and J. Langbein, Slip on the San Andreas Fault at Parkfield, California, over Two Earthquake Cycles, and the Implications for Seismic Hazard, *Bull. Seism. Soc. Am.*, *96*(4B), S283–S303, 2006.

Murray, J., and P. Segall, Spatiotemporal evolution of a transient slip event on the San Andreas fault near Parkfield, California, *Bull. Seism. Soc. Am.*, *110*(B09407), 12, 2005.

Murray, J. R., P. Segall, P. C. an W. Prescott, and J. Svarc, Inversion of GPS data for spatially variable slip-rate on the San Andreas Fault near Parkfield, CA, *Geophys. Res. Let.*, *28*(2), 359–362, 2001.

Nakatani, M., and C. H. Scholz, Frictional healing of quartz gouge under hydrothermal conditions: 2. quantitative interpretation with a physical model, *J. Geophys. Res.*, *109*(B07202), 10, 2004.

Niu, F., P. G. Silver, R. M. Nadeau, and T. V. McEvilly, Migration of seismic scatterers associated with the 1993 parkfield aseismic transient event, *Nature*, *426*, 544–548, 2003.

Noda, H., Frictional constitutive law at intermediate slip rates accounting for flash heating and thermally activated slip process, *J. Geophys. Res.*, *113*(B09302), 12, 2008.

Okada, Y., Surface deformation due to shear and tensile faults in a half-space, *Bull. Seism. Soc. Am.*, *75*(4), 1135–1154, 1985.

Pritchard, M. E., M. Simons, P. A. Rosen, S. Hensley, and F. H. Webb, Co-seismic slip from the 1995 July 30 Mw= 8.1 Antofagasta, Chile, earthquake as constrained by InSAR and GPS observations, *Geophys. J. Int.*, *150*(2), 362376, 2002.

Rice, J. R., Spatio-temporal complexity of slip on a fault, *J. Geophys. Res.*, *98*(B6), 9885–9907, 1993.

Rice, J. R., and A. L. Ruina, Stability of steady frictional slipping, *J. Appl. Mech.*, *50*, 343–349, 1983.

Rice, J. R., N. Lapusta, and K. Ranjith, Rate and state dependent friction and the stability of sliding between elastically deformable solids, *J. Mech. Phys. Solids*, *49*, 1865–1898, 2001.

Roeloffs, E., and J. Langbein, The earthquake prediction experiment at Parkfield, California, *Rev. Geophys.*, *32*(3), 315–336, 1994.

Rogers, G., and H. Dragert, Episodic tremor and slip on the Cascadia subduction zone: The chatter of silent slip, *Science*, *300*(5627), doi:10.1126/science.1084783, 2003.

Rolandone, F., R. Bürgmann, D. C. Agnew, I. A. Johanson, D. C. Templeton, M. A. d'Alessio, S. J. Titus, C. DeMets, and B. Tikoff, Aseismic slip and fault-normal strain along the central creeping section of the San Andreas fault, *Geophys. Res. Lett.*, *35*(L14305), 5, 2008.

Rosen, P. A., S. Henley, G. Peltzer, and M. Simons, Updated Repeat Orbit Interferometry Package Released, *Eos Trans. AGU*, *85*(5), 47, 2004.

Rubin, A. M., Episodic slow slip events and rate-and-state friction, *J. Geophys. Res.*, *113*(B11414), doi: 10.1029/2008JB005642, 2008.

Ruina, A., Slip instability and state variable friction laws, *J. Geophys. Res.*, *88*, 10,359–10,370, 1983.

Ryder, I., and R. Bürgmann, Spatial variations in slip deficit on the central San Andreas Fault from InSAR, *Geophys. J. Int.*, *175*(3), 837852, 2008.

Scholz, C. H., Earthquakes and friction laws, *Nature*, *391*, 37–42, 1998.

Segall, P., *Earthquake and volcano deformation*, Princeton University Press, Princeton, NJ, 2010.

Segall, P., and Y. Du, How similar were the 1934 and 1966 Parkeld earthquakes?, *J. Geophys. Res.*, *98*, 45274538, 1993.

Segall, P., and R. A. Harris, Earthquake deformation cycle on the San Andreas Fault near Parkfield, California, *J. Geophys. Res.*, *92*(B10), 10,511–10,525, 1987.

Segall, P., A. M. Rubin, A. M. Bradley, and J. R. Rice, Dilatant strengthening as a mechanism for slow slip events, *J. Geophys. Res.*, *115*(B12305), 2010.

Shen, Z.-K., R. King, D. Agnew, M. Wang, T. Herring, D. Dong, and P. Fang, A unified analysis of crustal motion in southern california, 1970–2004: The scec crustal motion map, *Journal of Geophysical Research*, *116*(B11), B11,402, 2011.

Shibazaki, B., T. Matsuzawa, and A. Tsutsumi, 3D modeling of the cycle of a great Tohoku-oki earthquake, considering frictional behavior at low to high slip velocities, *Geophys. Res. Lett.*, *38*(L21305), doi: 10.1029/2011GL049308, 2011.

Sieh, K., Slip along the San Andreas Fault associated with the great 1857 earthquake, *Bull. Seism. Soc. Am.*, *68*(5), 1421–1448, 1978a.

Sieh, K., Central California foreshocks of the great 1857 earthquake, *Bull. Seism. Soc. Am.*, *68*(8), 1731–1749, 1978b.

Sieh, K. E., and R. H. Jahns, Holocene activity of the San Andreas Fault at Wallace Creek, California, *Geology*, *95*, 883–896, 1984.

Sone, H., and T. Shimamoto, Frictional resistance of faults during accelerating and decelerating earthquake slip, *Nature Geosci.*, *2*, 705–708, 2009.

Thatcher, W., Systematic inversion of geodetic data in Central California, *J. Geophys. Res.*, *84*(B5), 2283–2295, 1979.

Thurber, C., H. Zhang, F. Waldhauser, J. Hardebeck, A. Michael, and D. Eberhat-Phillips, Three-dimensional compressional wavespeed model, Earthquake relocation, and focal mechanisms for the Parkfield, California, region, *Bull. Seism. Soc. Am.*, *96*(4B), S38–S49, 2006.

Toké, N. A., J. R. Arrowsmith, M. J. Rymer, A. Landgraf, D. E. Haddad, M. Busch, J. Coyan, and A. Hannah, Late holocene slip rate of the san andreas fault and its accommodation by creep and moderate-magnitude earthquakes at parkfield, california, *Geology*, *39*(3), 243–246, 2011.

Toppozada, T. R., D. M. Branum, M. S. Reichle, and C. L. Hallstrom, San Andreas Fault zone, California: M $\geq$ 5.5 earthquake history, *Bull. Seism. Soc. Am.*, *92*(7), 2555–2601, 2002.

Toro, G. D., D. L. Goldsby, and T. E. Tullis, Friction falls towards zero in quartz rock as slip velocity approaches seismic rates, *Nature*, *427*, 436–439, 2004.

Tse, S. T., and J. R. Rice, Crustal earthquake instability in relation to the depth variation of frictional slip properties, *J. Geophys. Res.*, *91*(B9), 9452–9472, 1986.

Uchide, T., S. Ide, and G. C. Beroza, Dynamic high-speed rupture from the onset of the 2004 Parkfield, California, earthquake, *Geophys. Res. Let.*, *36*(L04307), 6 PP., 2009.

Waldhauser, F., W. L. Ellsworth, D. P. Schaff, , and A. Cole, Streaks, multiplets, and holes: High-resolution spatio-temporal behavior of Parkfield seismicity, *Geophys. Res. Lett.*, *31*(L18608), doi:10.1029/2004GL020649, 2004.

Wang, H., T. J. Wright, and J. Biggs, Interseismic slip rate of the northwestern Xianshuihe fault from InSAR data, *Geophys. Res. Lett.*, *36*(L03302), 5, 2009.

Wang, R., F. Martin, and F. Roth, Computation of deformation induced by earthquakes in a multi-layered elastic crust - FORTRAN programs EDGRN/EDCMP, *Comp. Geosci.*, *29*, 195–207, 2003.

Wdowinkski, S., Y. Bock, J. Zhang, P. Fang, and J. Genrich, Southern California permanent GPS geodetic array: Spatial filtering of daily positions for estimating coseismic and postseismic displacements induced by the 1992 Landers earthquake, *J. Geophys. Res.*, *102*(B8), 18,057–18,070, 1997.

Wesnousky, S. G., Earthquakes, quaternary faults, and seismic hazard in California, *J. Geophys. Res.*, *91*(B12), 12,587–12,631, 1986.

Williams, S. D. P., Y. Bock, P. Fang, P. Jamason, R. M. Nikolaidis, L. Prawirodirdjo, M. Miller, and D. J. Johnson, Error Analysis of Continuous GPS Position Time Series, *J. Geophys. Res.*, *109*(B03412), doi: 10.1029/2003JB0022741, 2004.

Wood, H. O., The 1857 earthquake in California, *Bull. Seism. Soc. Am.*, *45*, 47–67, 1955.

Wright, T. J., B. Parsons, P. C. England, and E. J. Fielding, InSAR Observations of Low Slip Rates on the Major Faults of Western Tibet, *Science*, *305*(5681), 236–239, 2004.

Young, J. J., J. R. Arrowsmith, L. Colini, L. B. Grant, and B. Gootee, Three-dimensional excavation and recent rupture history along the cholame segment of the san andreas fault, *Bull. Seism. Soc. Am.*, *92*(7), 2670–2688, 2002.

Zebker, H., S. Hensley, P. Shanker, and C. Wortham, Geodetically accurate insar data processor, *IEEE Trans. Geosci. Rem. Sens.*, *48*(12), 4309–4321, 2010.

Ziv, A., Inference of coseismic slip via joint inversion of GPS and aftershock data: The 2004 Parkfield example, *J. Geophys. Res.*, *117*(B03307), 2012.

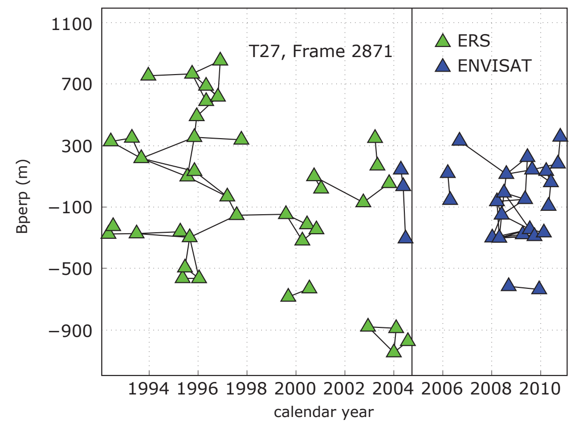


Figure A1. Time vs. perpendicular baseline plot for ERS/ ENVISAT data, track 27, frame 2871. See Fig. 1 for the surface footprint.

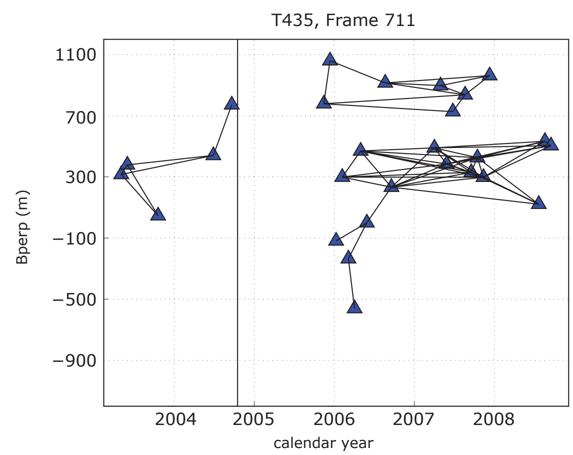


Figure A2. Time vs. perpendicular baseline plot for ENVISAT data, track 435, frame 711. See Fig. 1 for the surface footprint.

S. Barbot, P. Agram, M. Simons and M. De Michele, California Institute of Technology, Pasadena, CA 91125. (e-mail: sbarbot@caltech.edu)

This preprint was prepared with AGU's LaTeX macros v5.01, with the extension package 'AGU^{++}' by P. W. Daly, version 1.6b from 1999/08/19.

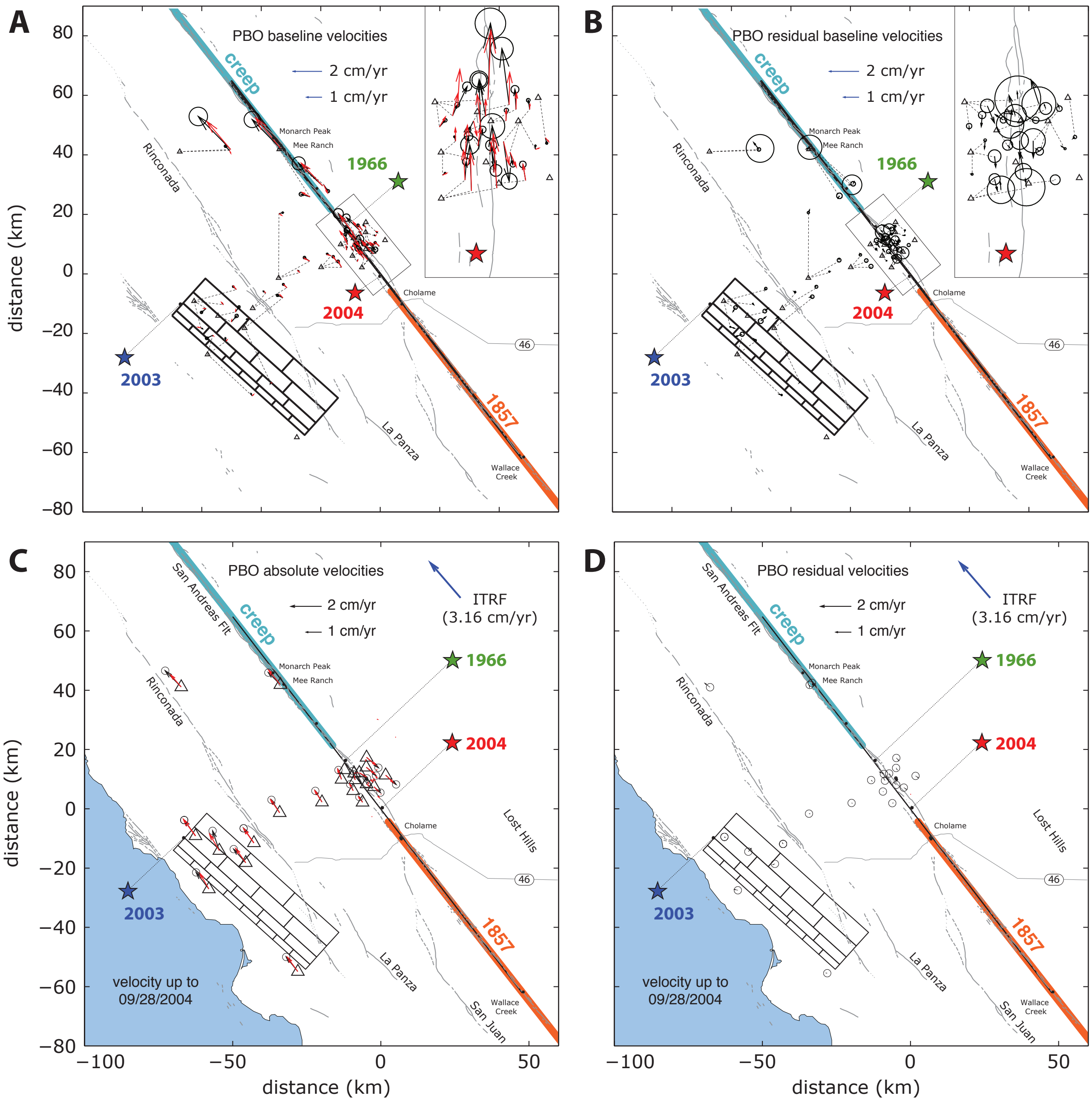


Figure A6. Absolute and baseline velocity field at the EarthScope Plate Boundary Observatory (PBO) network. A) Baseline velocity and forward model; B) residual baseline velocity. The relative velocity between two stations X and Y is represented by a vector starting in the middle of the baseline. A line connects station X and the velocity vector. C) Velocity and forward model; D) residual velocity. The velocity is relative to the SAF and the residual ITRF velocity is shown for reference.

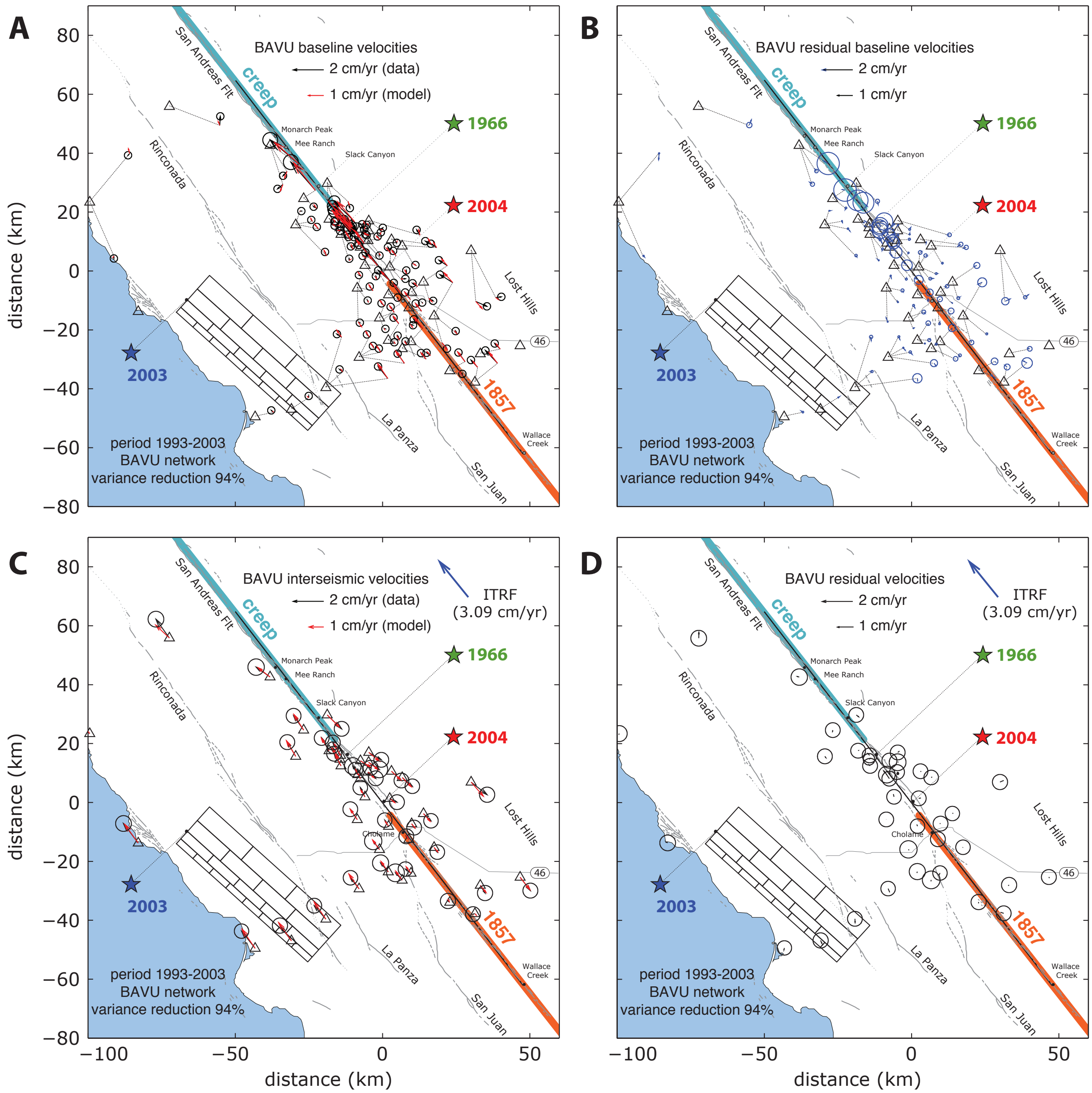


Figure A7. Absolute and baseline velocity field at the Bay Area Velocity Unification (BAVU) network [*d'Alessio et al.*, 2005]. A) Baseline velocity and forward model; B) residual baseline velocity. The relative velocity between two stations X and Y is represented by a vector starting in the middle of the baseline. A line connects station X and the velocity vector. C) Velocity and forward model; D) residual velocity. The velocity is relative to the SAF and the residual ITRF velocity is shown for reference.

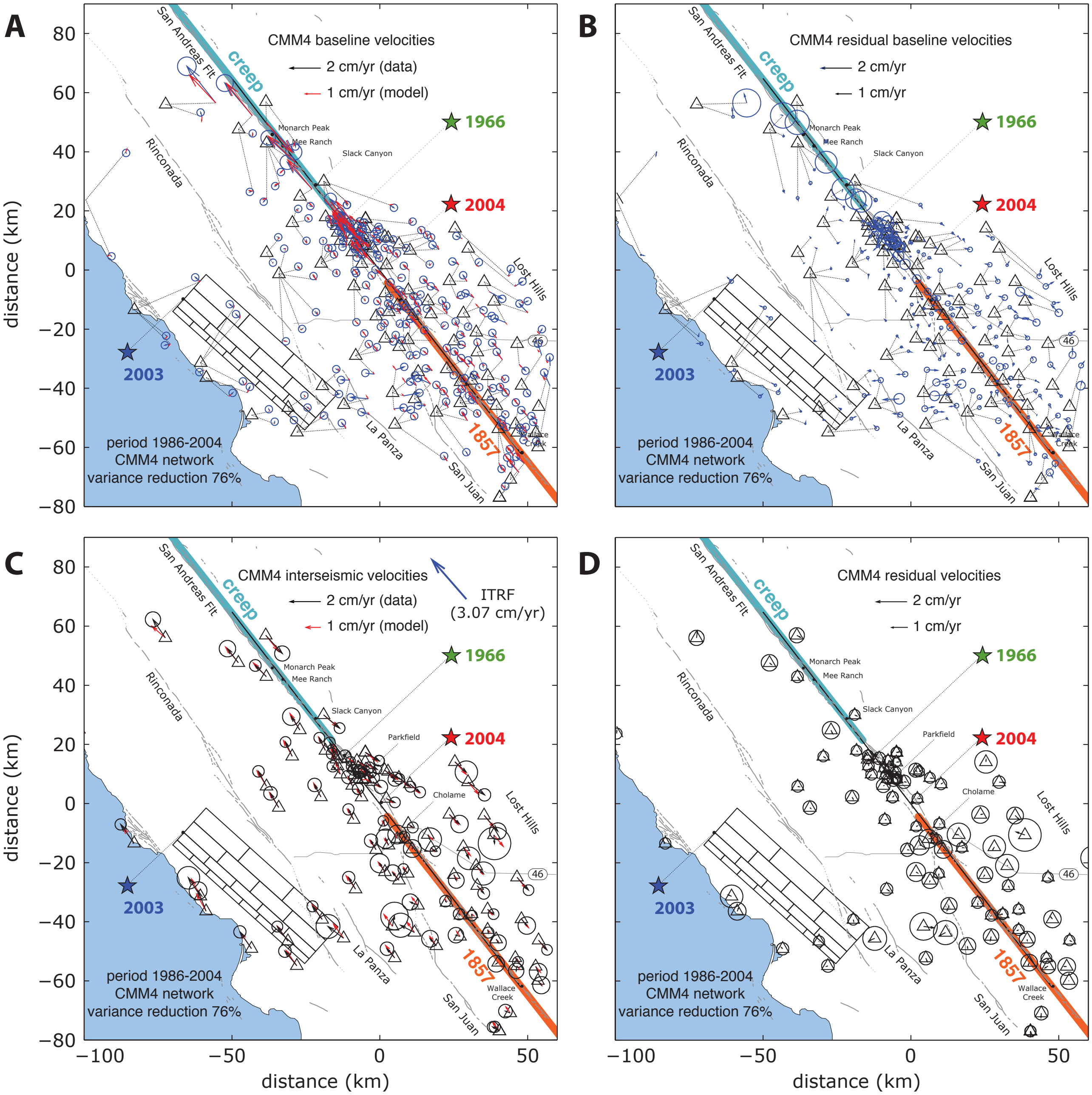


Figure A8. Absolute and baseline GPS velocity field at the SCEC Crustal Motion Map 4 (CMM4) compilation network [*Shen et al.*, 2011]. A) Baseline velocity and forward model; B) residual baseline velocity. The relative velocity between two stations X and Y is represented by a vector starting in the middle of the baseline. A line connects station X and the velocity vector. C) Velocity and forward model; D) residual velocity. The velocity is relative to the SAF and the best-fit velocity of the stable North American Plate in the ITRF reference frame is shown for reference.

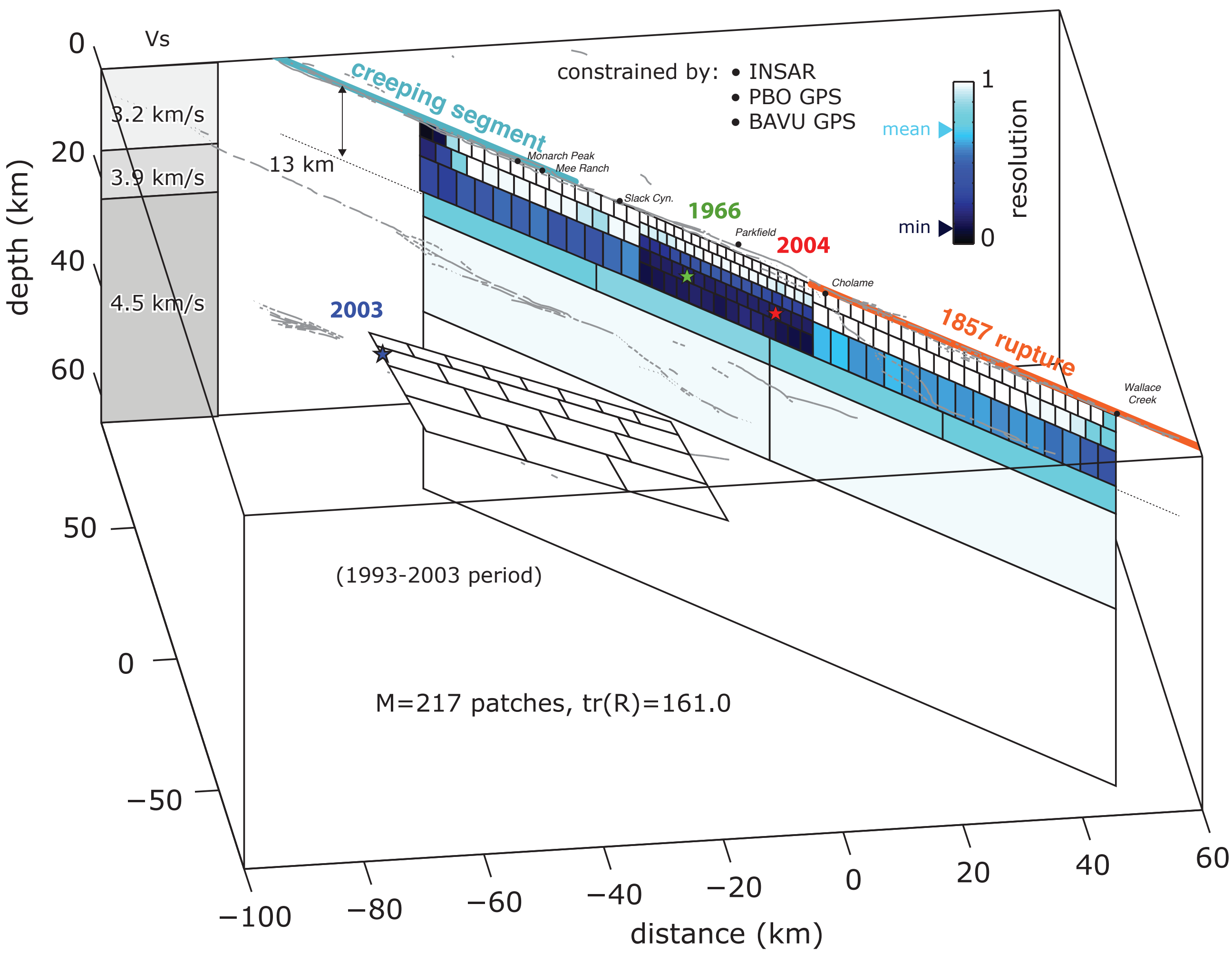


Figure A9. Resolution of the joint inversion of InSAR and GPS data for the 1992-2004 period using GPS data from the BAVU and PBO networks and InSAR data from the ALOS, ERS and ENVISAT satellites.

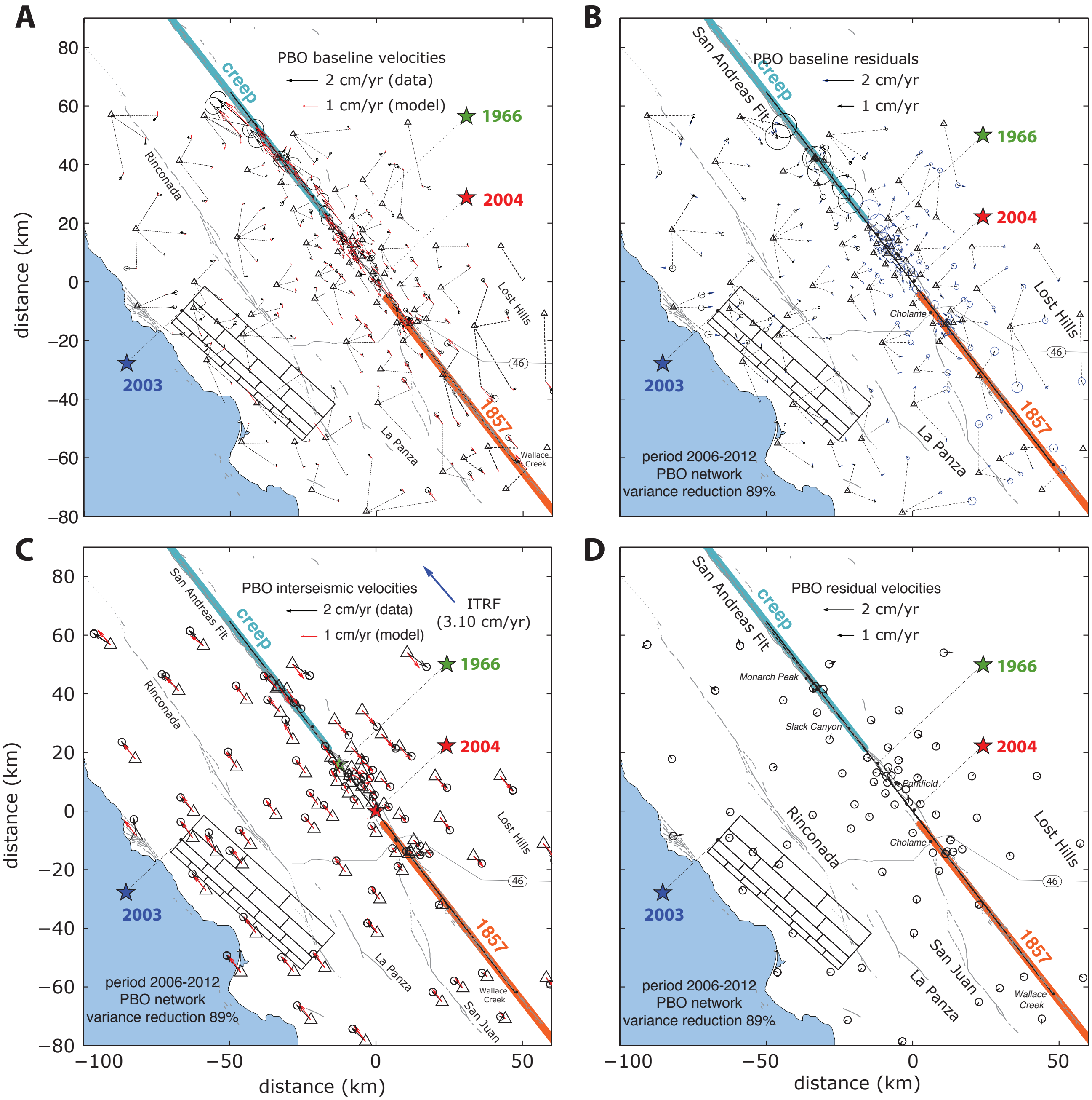


Figure A10. Baseline and absolute velocity fields at the PBO network in the 2006-2012 period. A) Baseline velocity and forward model; B) residual baseline velocity. The relative velocity between two stations X and Y is represented by a vector starting in the middle of the baseline. X line connects station X and the velocity vector. C) Velocity and forward model; D) residual velocity. The velocity is relative to the SAF and the residual ITRF velocity is shown for reference.

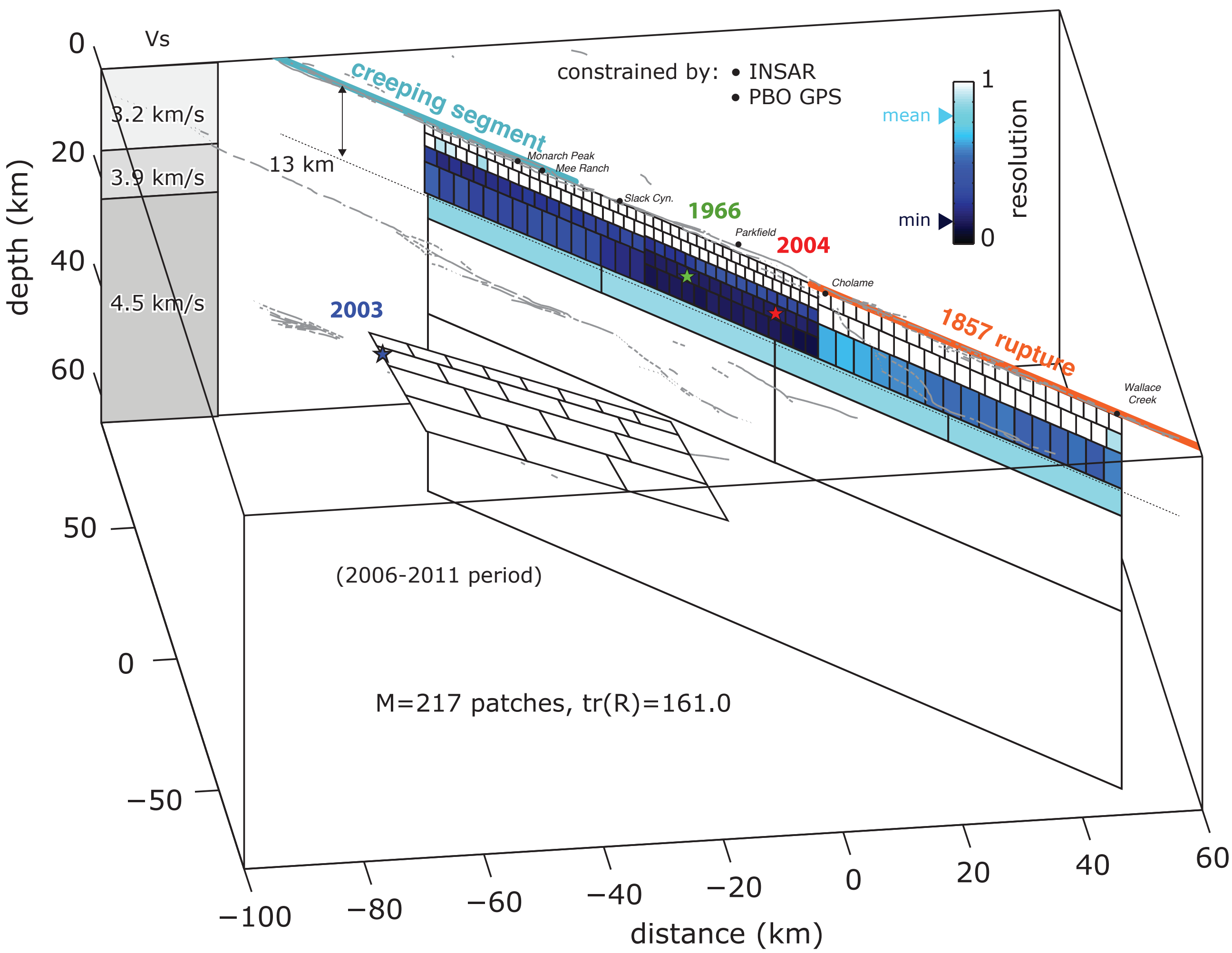


Figure A11. Resolution of the joint inversion of InSAR (ENVISAT and ALOS satellites) and GPS (PBO network) data for the 2006-2012 period. The improved coverage above the creeping segment allows us to refine fault sampling at this location, compared to the other period considered.

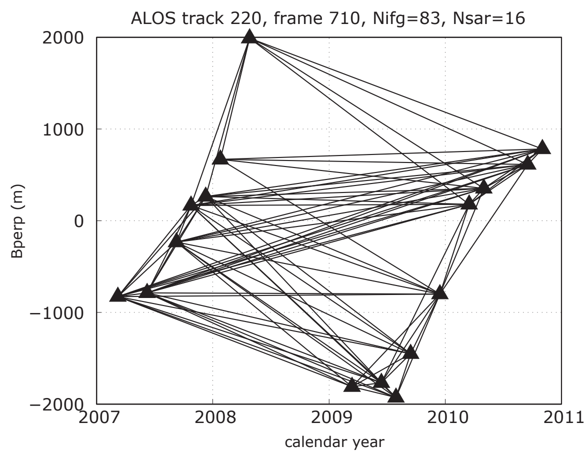


Figure A3. Time vs. perpendicular baseline plot for ALOS ascending data, track 220, frame 710. See Fig. 1 for the surface footprint.

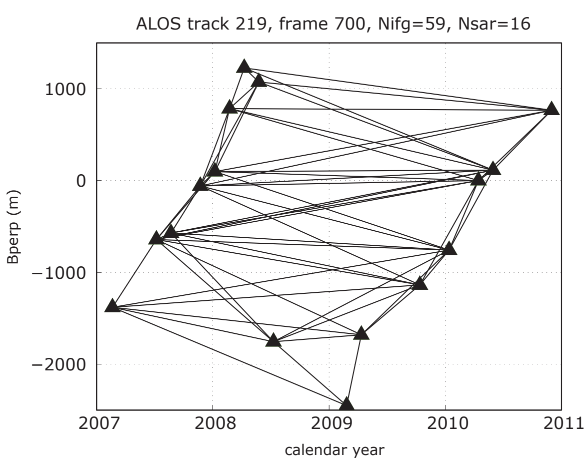


Figure A4. Time vs. perpendicular baseline plot for ALOS ascending data, track 219, frame 700. See Fig. 1 for the surface footprint.

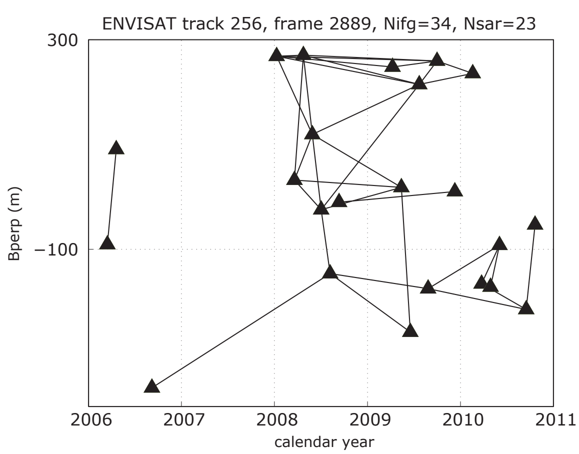


Figure A5. Time vs. perpendicular baseline plot for ENVISAT track 256, frame 2889. See Fig. 1 for the surface footprint.

Attachment 4:

Modelling the San Andreas Fault with
space geodetic data

Preparing Space Geodetic Monitoring of the Mentawai Seismic Gap of the Sunda Megathrust

Sylvain Barbot[1], Piyush Agram[2] and Emma Hill[1]

[1]Earth Observatory of Singapore, Nanyang Technological University, Singapore
[2]California Institute of Technology, California, USA

Abstract.

The Mentawai seismic gap of the Sunda Megathrust has accumulated about 10 m of fault slip deficit since the great 1797 Mw 8.9 earthquake and the recent ruptures along the megathrust have increased the chance for a great-to-giant earthquake there in the near future. To prepare for this threatening eventuality and better understand the degree of segmentation of this section of the subduction, we investigate the potential of GPS to image slip on the Mentawai segment from the trench to below the brittle-ductile transition. Building on the continuously-operating Sumatra GPS Array, we test the potential of augmented networks for monitoring fault slip during the currently quiescent period. Using a global optimization procedure that identifies the most favorable distribution of inland stations, we find that a combination of island and mainland stations can much better resolve fault slip with a 30×30 km spatial resolution from about 30 km offshore the farthest islands to the coast of Sumatra. With exact data, slip near the trench could be perfectly resolved at a 30 km resolution, but insufficient signal-to-noise ratios prevent GPS data from providing any strong constraints on slip so far away from the network. The presence of the Batu, Siberut, Sipora and Pagai islands hundred kilometers off the mainland is a favorable topology that will allow us to monitor fault activity with much improved accuracy and mitigate seismic and tsunami hazards in Indonesia. Similar theoretical exercises can guide optimal deployment of geodetic networks to address specific questions related to seismic and volcanic activity.

1. Introduction

The oblique convergence between the Australian and Eurasian plates is accommodated near Sumatra by dip slip along the Sunda megathrust and dextral trench-parallel slip along the Great Sumatran Fault at a cumulative rate of 5.7 to 6.2 cm/yr (Fig. 1). In the last decade, a sequence a large earthquakes struck the Sunda thrust starting to the north with the 2004 moment-magnitude (Mw) 9.15 Sumatra-Andaman earthquake [*Lay et al.*, 2005; *Ishii et al.*, 2005; *Subarya et al.*, 2006; *Chlieh et al.*, 2007], and continuing southward with the 2005 Mw 8.6 Nias earthquake [*Hsu et al.*, 2006; *Konca et al.*, 2008] and the 2007 sequence of Mw 7.9 and Mw 8.4 ruptures [*Konca et al.*, 2008]. More recently, the 2010 Mw 7.8 Mentawai earthquake ruptured a near-trench portion of the Mentawai segment [*Newman et al.*, 2011; *Lay et al.*, 2011; *Bilek et al.*, 2011; *Hill et al.*, 2012], generating a large tsunami that devastated the city of Padang, Indonesia.

Earthquakes in the Sumatran section of the Sunda thrust are thought to cluster in time [*Natawidjaja et al.*, 2007; *Sieh et al.*, 2008; *Prawirodirdjo et al.*, 2010], as illustrated by the recent events since 2004. Up to now, the largest unbroken section is the Mentawai segment of the Sunda thrust, which last ruptured during the 1797 Mw 8.8 and the 1833 Mw 9.0 earthquakes (Fig. 1). Despite the partial rupture of the Mentawai segment in 2007 and 2009, only a fraction of the slip that occurred in 1833 and of the slip deficit that accumulated since 1833 has been relieved [*Konca et al.*, 2008] and currently the greatest and most imminent seismic hazards in the region come from the Mentawai segment [*Nalbant et al.*, 2005].

Fault slip evolution on the megathrust is partitioned along depth into large intermediate-depth earthquakes, shallow ruptures, and aseismic slip. Events such as the 1833 Mw 9.0 rupture cause tsunamis on the neighboring islands and mainland coasts and seem to occur every 230 years, on average. However, the variability observed in paleoseismic records precludes a precise empirical forecast of the next rupture [*Sieh et al.*, 2008]. Tsunami earthquakes also occur on the shallow portion of the megathrust, generating tsunamis with larger runoff and peak wave height than anticipated for their magnitude. Examples of such events include the 1907 Mw 7.6 [*Kanamori et al.*, 2010] and the 2010 Mw 7.8 Mentawai earthquakes [*Newman et al.*, 2011; *Lay et al.*, 2011; *Bilek et al.*, 2011; *Hill et al.*, 2012]. At greater

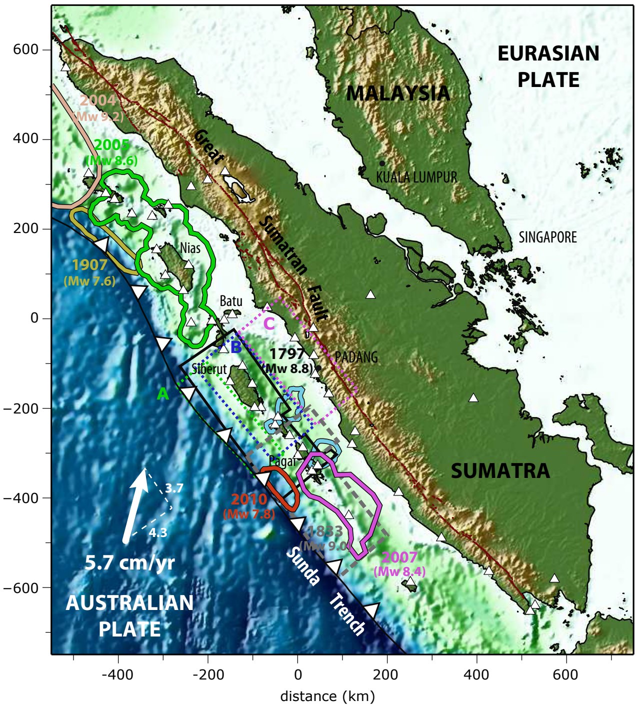


Figure 1. In the last 300 years, the Sunda Megathrust has generated several great to giant earthquakes including the 1797 Mw 8.8 (black profile), the 1833 Mw 9.0 (dashed grey profile), the great 2004 Mw 9.2 Sumatra-Andaman, the 2005 Mw 8.6 Nias (green profile) and the 2007 Mw 8.4 ruptures. The three zones of interest includes the tsunamigenic near-trench area A (dashed green box), the seismogenic zone B (dashed blue box), and the deeper, mostly aseismic, fault area C (dashed purple box).

depths, transient slow slip events can occur, such as the 1962 event with an equivalent moment magnitude of Mw 8.4 [*Natawidjaja et al.*, 2007].

Since 2004, earthquakes and tsunamis emanating from the Sunda megathrust have claimed more than 200,000 lives and it is certain that new ruptures on the Mentawai segment carry significant risk for the local population. It is of paramount scientific and humanitarian importance to build a better physical understanding of the earthquake cycle [e.g., *Barbot et al.*, 2012b, a] and shed light on the full seismic potential of the Mentawai segment. Despite long and on-going efforts to better image fault slip evolution on the Sunda thrust [*Natawidjaja et al.*, 2007; *Prawirodirdjo et al.*, 2010], it is still notoriously challenging to effectively resolve slip on many portions of the fault (Fig. A1), in particular near the trench [*Hill et al.*, 2012] or at great depths, even when including paleogeodetic markers and other data [*Chlieh et al.*, 2008].

The goal of the study is to identify the potential and limits of GPS to study seismo-tectonics in a subduction context and to establish the best strategy to design and conduct future geodetic surveys in the Mentawai region. In the next Section, we describe a methodology to optimize the layout of future GPS networks and propose arrangements of new stations that augment the Sumatra GPS Array (SuGAr) to best resolve slip at various depths along the Mentawai seismic gap.

2. Optimal GPS networks for monitoring of the Mentawai segment

We seek to design a GPS network in Sumatra that can resolve slip on the Mentawai segment of the Sunda megathrust. The intention of the survey is to monitor interseismic deformation in the Mentawai seismic gap while preventing potential bias originating from slip at other locations, such as the nearby Great Sumatran Fault or other segments of the Sunda thrust.

2.1. Methodology of GPS network design

We discretize the Sumatran section of the megathrust into 30 km-wide rectangular patches (Fig. A1) using the USGS slab geometry at depth [*Hayes et al.*, 2012]. We also include the Great Sumatran Fault, which we cut into coarser patches 90-to-400 km long and 15 km deep. We consider the thought experiment where the GPS time series of interseismic displacements in the horizontal direction are used to model slip on individual fault patches on the megathrust and the Great Sumatra Fault, and to invert simultaneously for steady drift of the entire local network relative to an arbitrary reference frame. To simplify the inversion, we allow only dip slip on the Sunda thrust and dextral slip on the Great Sumatran Fault. This results in 537 unknown parameters. We ignore the effect of splay faults that are known to also generate earthquakes [*McCloskey et al.*, 2010] and more distributed deformation.

The GPS displacements are obtained from the model parameters using the equations of *Okada* [1985] to form the Green's function matrix $\mathbf{G}$. Following *Backus* [1970] and *Tarantola* [2004], we describe the resolution of the under-determined inversion in the presence of noise by the operator

$$\mathbf{R} = \mathbf{G}^t \left(\mathbf{G}\mathbf{G}^t\right)^+ \mathbf{G} \ , \tag{1}$$

where $(\mathbf{G}\mathbf{G}^t)^+$, the Moore-Penrose pseudo inverse of $\mathbf{G}\mathbf{G}^t$, ignores eigenvectors with an amplitude lower than a given threshold ψ. In practice, we use $\psi = \sigma^2/u^2$, where $\sigma = 1\,\mathrm{mm/yr}$ is a representative estimate of the uncertainties of long-term GPS velocity, and $u = 1\,\mathrm{m/yr}$ is the unit value used to define the Green's function. We define the number of well-resolved parameters in a particular region by $r_A^N = \mathrm{tr}(\mathbf{W}_A\,\mathbf{R})$, where N is the number of GPS stations used in the inversion, and $\mathbf{W}_A$ is a diagonal matrix with values of one if the model parameter corresponds to a patch in domain A, and of zeros otherwise.

SuGAr is a network of 50 continuous GPS stations that cover a $1200 \times 400\,\mathrm{km}$ seismically active zone (Fig. 1). We first investigate the effect of augmenting SuGAr with a single idealized amphibious GPS station measuring horizontal motion on the resolution in the Mentawai seismic gap (Fig. 2). Underwater locations represent a potential seafloor geodesy datum. Because of the shallow angle subduction, surface or seafloor measurements can be done close to the fault and a new measure can constrain slip on two fault patches simultaneously. The difficulty is that to achieve this efficiency, most new stations must be on the seafloor. Can we monitor better fault kinematics with more inland stations?

To determine the optimal number and location of new GPS stations required to monitor specific regions of the fault reliably, we adopt a sequential algorithmic approach [*Reeves and Zhe*, 1999]. We choose the GPS stations that will augment the existing SuGAr network one by one from a predetermined grid. The subset selection algorithm of *Reeves and Zhe* [1999] is originally designed to reduce the uncertainty in parameter estimates from a linear system. We modify this approach to choose the location of the GPS stations that maximizes the minimum resolution of fault patches in a predefined region. We further optimize our GPS station locations using the hybrid improvement algorithm described by *Broughton et al.* [2010], which allows for substitution of a previously selected station by a better candidate in the network. With the density of candidate stations considered, the algorithm requires evaluating the resolution matrix (1) about five million times.

2.2. Potential and limits of optimal GPS networks for the Mentawai section

Using our robust nonlinear optimization procedure, we maximize the resolution of those patches located in the recently unruptured areas of the Mentawai segment by choosing a set of new GPS stations from a list of candidate locations. Candidate stations are distributed regularly on a 15 km-spaced grid in mainland Sumatra, and on a 5 km-spaced grid on the islands of Batu, Siberut, Sipora and Pagai (Fig. A2). Decreased spacing offshore Sumatra allows more GPS stations closer to the source.

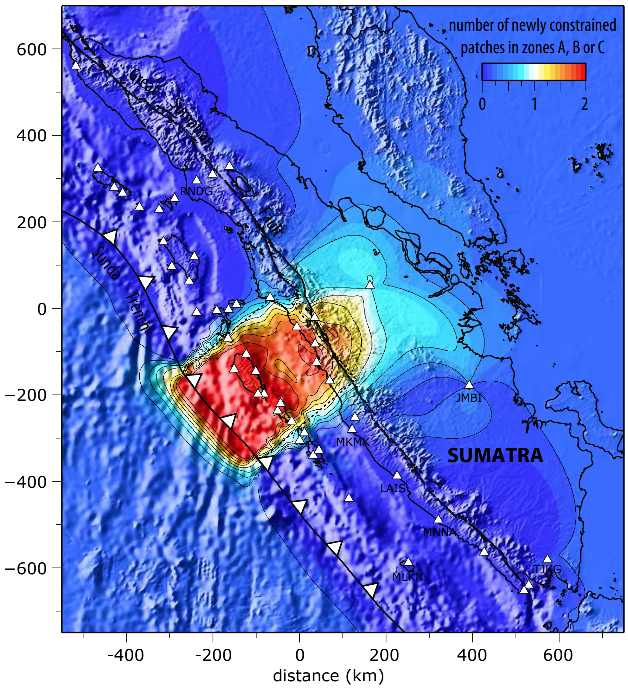


Figure 2. Map view of the increment in well-resolved model parameters (the difference between the weighted traces of the reference and augmented resolution matrices for the Mentawai seismic gap) for a potential new GPS station or seafloor datum complementing the existing SuGAr network (50 white upright triangles).

The Sumatran section of the Sunda thrust exhibits a complex spatio-temporal pattern of seismic and aseismic activity, including tsunami earthquakes near the trench (zone A), large ruptures at intermediate depths (zone B), and transient aseismic events below the brittle-ductile transition (zone C) [e.g., *Yamanaka and Kikuchi*, 2004; *Uchida and Matsuzawa*, 2011; *Lay et al.*, 2012]. We divide the Mentawai segment into three depth intervals susceptible to manifest these distinct behaviors (Fig. 1) and find augmented networks that maximize the resolution of fault slip in each of these three zones individually (Fig. 3).

We find that it is impossible to resolve the farthest patches in zone A using GPS data with typical noise levels. This is in contrast with inverting exact data, in which case near-trench slip can be resolved exactly with numerous new stations (Fig. A3). Monitoring of zone B can be improved the most efficiently with a minimum resolution approaching $r = 0.8$ after adding 10 to 20 stations. With 20 new, optimally located, stations, most the resolution of most patches in zone B approaches 1 (Fig. 3B). The resolution of slip in zone C can be modestly improved up to a minimum resolution of $r = 0.3$. This is achieved by adding more stations on the western coast of Sumatra (Fig. A4), but it will be impossible to fully constrain those patches located 60 to 100 km below the surface (Fig. A2). The positions of the new stations required to optimize zones B and C are listed in Tables A1, A2, and A4. These results imply that a modest upgrade of the SuGAr network can be directed toward a better monitoring of two of the three zones of interest.

3. Conclusion

Seismic activity on the Sunda thrust includes large seismic events, such as the 2004 Sumatra-Andaman rupture, tsunami earthquakes, and transient aseismic events. The Mentawai segment of the Sunda megathrust is the last unruptured portion of the Sumatran section of the Sunda subduction zone and represents the greatest and most imminent hazards originating from the Sunda megathrust for the coasts of Sumatra and the Indian Ocean. Despite the large number of continuous GPS stations operating in the SuGAr network and the presence of numerous paleogeodetic markers, it is still challenging to monitor precisely fault activity along the Mentawai segment.

To better prepare for anticipated ruptures and other aseismic fault activity on the Mentawai segment, we explore the capacity of GPS networks to monitor slip during the dormant phase of the seismic cycle. This task is challenging because deformation can mostly be measured by inland stations, many tens of kilometers away from the source. Using a robust nonlinear optimization scheme, we identify the optimal distribution of GPS networks augmenting the current configuration of the SuGAr network. Intermediate depths earthquakes can be well imaged with only 10 to 20 more stations. But strong constraints on fault slip near the trench will require an extension of the GPS data offshore, using seafloor geodesy. However, deploying one of the suggested networks would allow monitoring of fault activity on the Sunda thrust with much improved accuracy. Efforts are under way at the Earth Observatory to seize this unique opportunity of observation before a sizable earthquake strikes the region as it would allow a far better understanding of the fault kinematics than is currently available at any other subduction in the world.

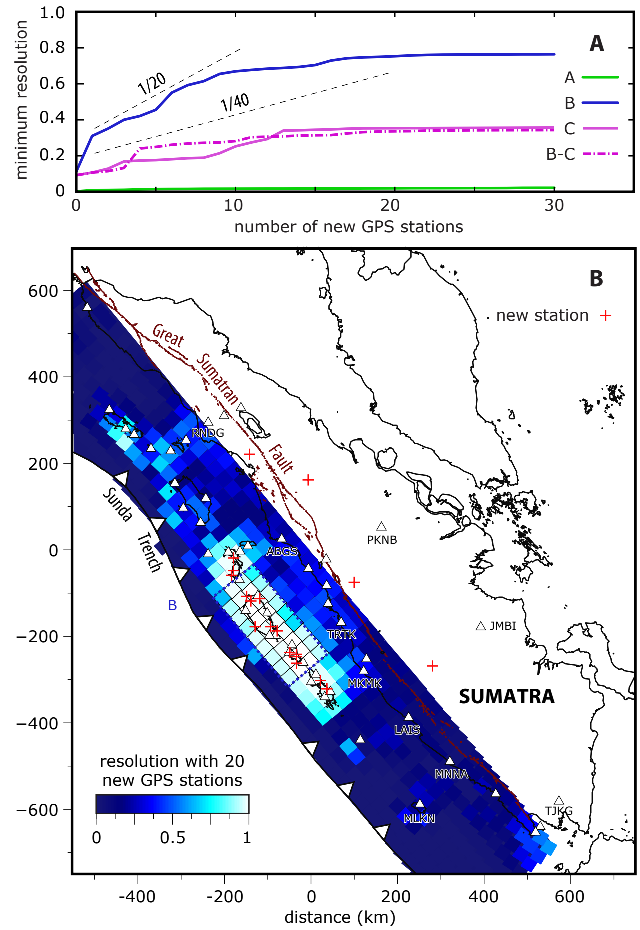


Figure 3. Experiment design to monitor the slip on the slab from the trench to the coast. A) Minimum resolution in a region of interest as a function of the number of additional new stations in an optimally distributed GPS network complementing the 2012 SuGAr network. Profiles A, B and C correspond to optimizing slip resolution in 30×30 km patches near the trench, seismogenic zone and deeper, respectively. B) Spatial distribution of the optimal network with 20 additional stations (red stars) and corresponding resolution on the fault. Station coordinates listed in Table A2.

Acknowledgements

This work was supported by the Tectonics Observatory, the Seismological Laboratory and the Keck Institute for Space Studies at the California Institute of Technology.

References

Backus, G., Inference from inadequate and inaccurate data, I, *Proc. Nat. Ac. Sci.*, *65*(1), 1970.

Barbot, S., P. Agram, and M. Simons, Optimization of GPS networks for tectonics application: Case of the Parkfield segment of the San Andreas Fault, California, *Geophys. Res. Lett.*, p. submitted, 2012a.

Barbot, S., N. Lapusta, and J. P. Avouac, Under the hood of the earthquake machine: Towards predictive modeling of the seismic cycle, *Science*, *336*(6082), 707–710, 2012b.

Bilek, S. L., E. R. Engdahl, H. R. DeShon, and M. El Hariri, The 25 October 2010 Sumatra tsunami earthquake: Slip in a slow patch, *Geophys. Res. Lett.*, *38*(L14306), 5, 2011.

Broughton, R., I. Coope, P. Renaud, and R. Tappenden, Determinant and exchange algorithms for observation subset selection, *IEEE Trans. on Image Processing*, *19*(9), doi:10.1109/TIP.2010.2048150, 2010.

Chlieh, M., J.-P. Avouac, K. Sieh, D. H. Natawidjaja, and J. Galetzka, Heterogeneous coupling of the Sumatran megathrust constrained by geodetic and paleogeodetic measurements, *J. Geophys. Res.*, *113*(B05305), 2008.

Chlieh, M., et al., Coseismic Slip and Afterslip of the Great Mw 9.15 Sumatra-Andaman Earthquake of 2004, *Bull. Seism. Soc. Am.*, *97*(1A), S152–S173, doi: 10.1785/0120050631, 2007.

Hayes, G. P., D. J. Wald, and R. L. Johnson, Slab1.0: A three-dimensional model of global subduction zone geometries, *J. Geophys. Res.*, *117*(B01302), doi: 10.1029/2011JB008524, 2012.

Hill, E. M., et al., The 2010 Mw 7.8 Mentawai earthquake: Very shallow source of a rare tsunami earthquake determined from tsunami field survey and near-field GPS data, *J. Geophys. Res.*, *117*(B06402), 21, 2012.

Hsu, Y.-J., M. Simons, J.-P. Avouac, J. Galetzka, K. Sieh, M. Chlieh, D. Natawidjaja, L. Prawirodirdjo, and Y. Bock, Friction Afterslip Following the 2005 Nias-Simeulue Earthquake, Sumatra, *Science*, *312*, 1921–1926, 2006.

Ishii, M., P. M. Shearer, H. Houston, and J. E. Vidale, Extent, duration and speed of the 2004 Sumatra-Andaman earthquake imaged by the Hi-Net array, *Nature*, *435*, 933–936, doi:10.1038/nature03675, 2005.

Kanamori, H., L. Rivera, and W. H. K. Lee, Historical seismograms for unraveling a mysterious earthquake: The 1907 Sumatra earthquake, *Geophys. J. Int.*, *183*, 358–374, 2010.

Konca, A. O., et al., Partial rupture of a locked patch of the Sumatra megathrust during the 2007 earthquake sequence, *Nature*, *456*, 631–635, 2008.

Lay, T., C. J. Ammon, H. Kanamori, Y. Yamazaki, K. F. Cheung, and A. R. Hutko, The 25 October 2010 Mentawai tsunami earthquake (Mw 7.8) and the tsunami hazard presented by shallow megathrust ruptures, *Geophys. Res. Lett.*, *38*(L06302), 5, 2011.

Lay, T., H. Kanamori, C. J. Ammon, K. D. Koper, A. R. Hutko, L. Ye, H. Yue, and T. M. Rushing, Depth-varying rupture properties of subduction zone megathrust faults, *Journal of Geophysical Research*, *117*(B4), B04,311, 2012.

Lay, T., et al., The Great Sumatra-Andaman Earthquake of 26 December 2004, *Sience*, *308*(5725), 1127–1133, doi: 10.1126/science.1112250, 2005.

McCloskey, J., D. Lange, F. Tilmann, S. S. Nalbant, A. F. Bell, D. H. Natawidjaja, and A. Rietbrock, The September 2009 Padang earthquake, *Nature Geosci.*, *3*(2), 70–71, 2010.

Nalbant, S. S., S. Steacy, K. Sieh, D. Natawidjaja, and J. McCloskey, Earthquake risk on the Sunda trench, *Nature*, pp. 756–757, doi:10.1038/nature435756a, 2005.

Natawidjaja, D. H., K. Sieh, J. Galetzka, B. W. Suwargadi, H. Cheng, R. L. Edwards, and M. Chlieh, Interseismic deformation above the Sunda Megathrust recorded in coral microatolls of the Mentawai islands, West Sumatra, *J. Geophys. Res.*, *112*(B02404), 27, 2007.

Newman, A. V., G. Hayes, Y. Wei, and J. Convers, The 25 October 2010 Mentawai tsunami earthquake, from real-time discriminants, finite-fault rupture, and tsunami excitation, *Geophys. Res. Lett.*, *38*(L05302), 7, 2011.

Okada, Y., Surface deformation due to shear and tensile faults in a half-space, *Bull. Seism. Soc. Am.*, *75*(4), 1135–1154, 1985.

Prawirodirdjo, L., R. McCaffrey, C. D. Chadwell, Y. Bock, and C. Subarya, Geodetic observations of an earthquake cycle at the Sumatra subduction zone: Role of interseismic strain segmentation, *J. Geophys. Res.*, *115*(B03414), 15, 2010.

Reeves, S. J., and Z. Zhe, Sequential algorithms for observation selection, *IEEE Trans. on Signal Processing*, *47*(1), doi:10.1109/78.738245, 1999.

Sieh, K., et al., Earthquake supercycles inferred from sea-level changes recorded in the corals of West Sumatra, *Science*, *322*, 1674–1678, 2008.

Subarya, C., M. Chlieh, L. Prawirodirdjo, J.-P. Avouac, Y. Bock, K. Sieh, A. J. Meltzner, D. H. Natawidjaja, and R. McCaffrey, Plate-boundary deformation associated with the great SumatraAndaman earthquake, *Nature*, *440*, 46–51, doi:10.1038/nature04522, 2006.

Tarantola, A., *Inverse Problem Theory and Methods for Model Parameter Estimation*, Soc. Ind. App. Math, Philadelphia, PA, USA, 2004.

Uchida, N., and T. Matsuzawa, Coupling coefficient, hierarchical structure, and earthquake cycle for the source area of the 2011 off the pacific coast of tohoku earthquake inferred from small repeating earthquake data, *Earth Planets and Space*, *63*(7), 675, 2011.

Yamanaka, Y., and M. Kikuchi, Asperity map along the subduction zone in northeastern japan inferred from regional seismic data, *Journal of geophysical research*, *109*(B7), B07,307, 2004.

S. Barbot and E. Hill, Earth Observatory of Singapore, Singapore (e-mail: sbarbot@ntu.edu.sg)

P. Agram, California Institute of Technology, Pasadena, CA 91125

This preprint was prepared with AGU's LaTeX macros v5.01, with the extension package 'AGU^{++}' by P. W. Daly, version 1.6b from 1999/08/19.

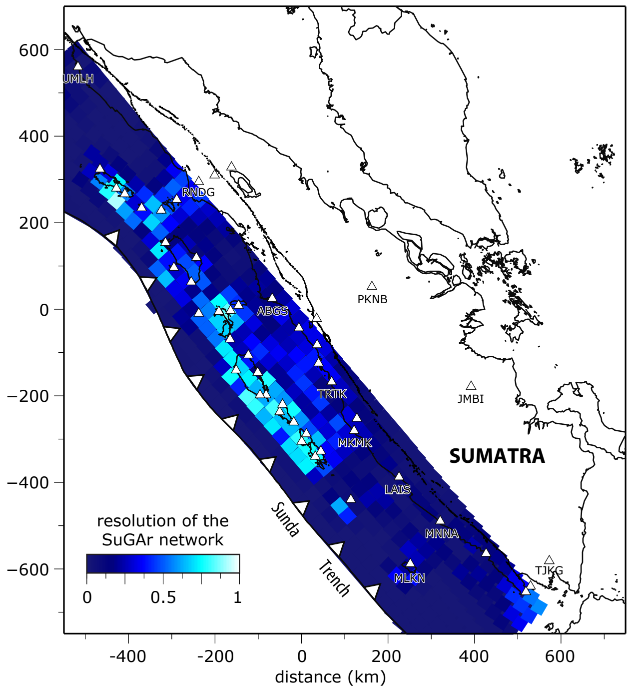


Figure A1. Current resolution of the SuGAr geodetic network with 50 continuously operating stations. With many stations located on remote islands off the mainland of Sumatra, the network can accurately detect slip concentrated in the nominally seismogenic zone, but the resolution is too small near the trench to constrain fault slip at that location.

Table A1. Coordinates of an optimal network of 10 stations for the best resolution near the trench (zone B).

longitude	latitude
98.40830	-0.42765
98.67820	-0.96784
98.76817	-0.96784
98.90311	-1.05788
98.90312	-1.19293
99.17300	-1.59806
99.30795	-1.68809
99.71280	-2.18328
99.71280	-2.36334
100.20761	-2.72347

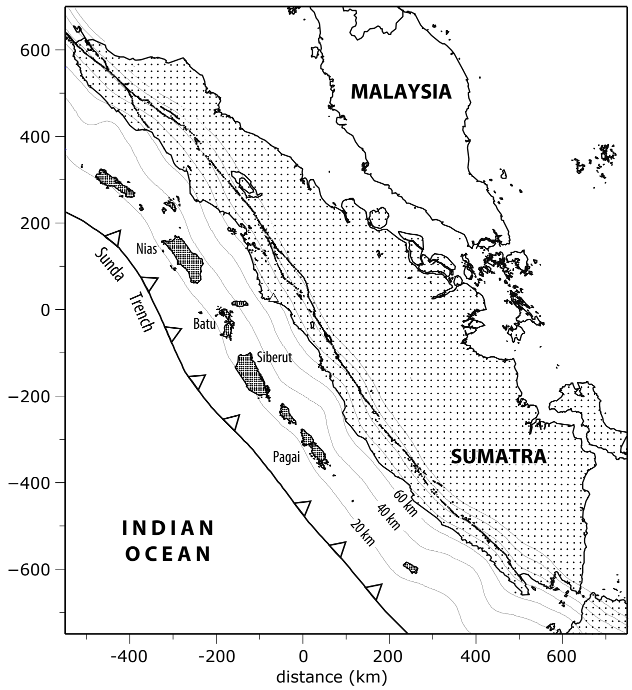


Figure A2. Position of the candidate GPS stations (black dots) considered for the network optimization. Station spacing is 15 km in mainland Sumatra and 5 km in the islands of Batu, Siberut, Sipora and Pagai. The density of candidate stations is increased in the Islands because of their proximity to the trench. The trench-parallel profiles are the 100 km depth contours of the Sunda slab depth.

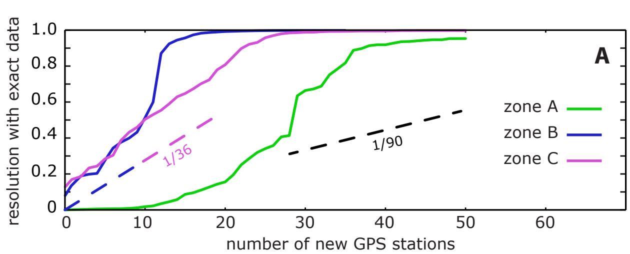


Figure A3. Potential resolution of zones A, B and C when using noise-free data. In this case, we evaluate the pseudo inverse in eq. (1) with a cut-off value ψ set to numerical precision. In this ideal case, slip near the trench can be very well resolved at a spatial resolution of 30 km.

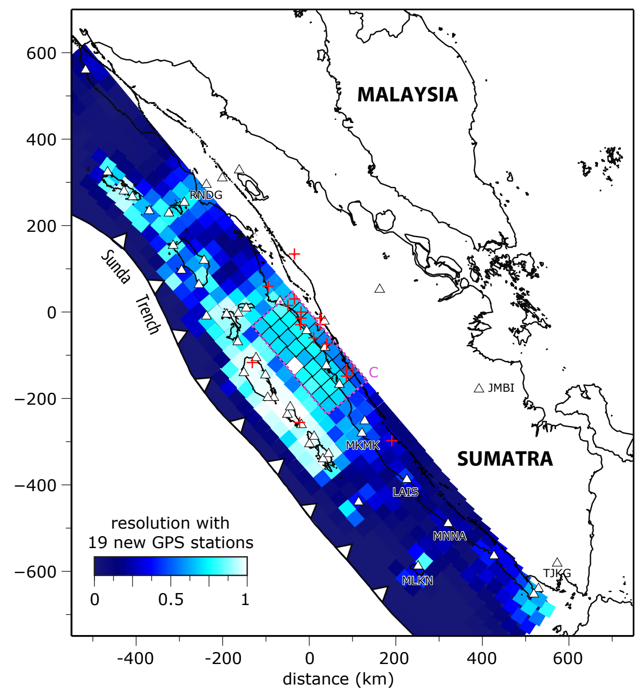


Figure A4. Experiment design to monitor the slip toward the bottom of the Sunda trench. Spatial distribution of the network optimizing resolution with 19 additional stations (red stars) and corresponding resolution on the fault. Most new stations are located in mainland Sumatra. Location of new stations in Table A4.

Table A2. Coordinates of an optimal network of 20 stations for the best resolution in the seismogenic zone (zone B).

longitude	latitude
98.40831	-0.15755
98.40830	-0.42765
98.36332	-0.51768
98.67820	-0.96784
98.94810	-1.01286
98.76816	-1.05787
98.85813	-1.59807
99.17300	-1.59806
99.30795	-1.68809
99.57785	-2.13826
99.71280	-2.18328
99.71280	-2.22829
99.71280	-2.22829
99.71280	-2.36334
100.20761	-2.72347
100.34256	-2.90353
98.73958	2.01923
99.95833	1.48077
100.90625	-0.67307
102.53125	-2.42307

Table A3. Coordinates of an optimal network of 10 stations for the best resolution at great depth (zone C).

longitude	latitude
98.993078	-1.282961
99.281251	0.269230
99.687504	0.269230
99.822917	-0.134615
99.958331	-0.134615
100.635414	-0.673077
100.499998	-0.942305
100.906251	-1.750001
101.447919	-1.884617
100.906249	-2.288462

Table A4. Coordinates of an optimal network of 19 stations for the best resolution at great depth (zone C).

longitude	latitude
98.813152	-1.057877
99.802765	-2.318332
99.687496	1.211536
99.145834	0.538462
99.687504	0.269230
99.552085	0.134615
99.822915	0.000000
99.822917	-0.134615
100.229170	-0.134615
99.822915	-0.269231
100.229171	-0.269231
100.229171	-0.269231
100.093749	-0.538461
100.364583	-0.673077
100.364583	-0.673077
100.770831	-1.211538
100.906258	-1.211536
100.770834	-1.346151
101.718751	-2.692306