



→ FRINGE 2011 WORKSHOP

Advances in the Science and Applications of SAR Interferometry and Sentinel-1 Preparatory Workshop

InSAR-capabilities of the ESA SAR Toolbox **NEST**

Petar Marinkovic, PPO.labs

Marcus Engdahl, ESA

16/09/11

Andrea Minchella, RSAC c/o ESA

Ramon Hanssen, TU Delft

Array Systems Computing Team



Contents

- **NEST Project overview**
- **Concept of NEST-DORIS**
- **NEST-DORIS Architecture**
- **NEST-DORIS Functionalities**
- **Review of features**

NEST Project Overview

NEST (*Next ESA SAR Toolbox*) is an ESA open source software devoted to the exploitation (displaying, analysing and processing) of ESA (ERS & ENVISAT) and other spaceborne SAR data processed from Level-1 or higher.

NEST Project Overview

NEST (*Next ESA SAR Toolbox*) is an ESA open source software devoted to the exploitation (displaying, analysing and processing) of ESA (ERS & ENVISAT) and other spaceborne SAR data processed from Level-1 or higher.

NEST Team

ESA

M. Engdahl (ESA Tech. Officer)
A. Minchella (RSAC c/o ESA)

Array Systems Computing

R.Jha, L.Veci, J. Lu, S. Dai

Brockmann Consult

N.Fomferra, M. Peters

PPO.labs (InSAR)

P. Marinkovic

TU Delft (InSAR)

Prof. R. Hanssen

NEST is being developed by Array Systems Computing Inc. of Toronto Canada under ESA Contract number 20698/07/I-LG._

NEST Project Overview

NEST (*Next ESA SAR Toolbox*) is an ESA open source software devoted to the exploitation (displaying, analysing and processing) of ESA (ERS & ENVISAT) and other spaceborne SAR data processed from Level-1 or higher.

Project schedule

Kick off November 2007

Phase A

- SRR System Requirements Review (month 2)
- PDR Preliminary Design Review (month 4)

Phase B

Release 1A - 1B - **1C** 1st Public release Nov. 2008
 Release 2A - 2B - **2C** (June 2009)
 Release 3A - 3B - **3C** (March 2010)

2 year Follow-up contract

Release 4A-1.6 (November 2010) Stable

Release 4B-1.0 (September 2011)

NEST Team

ESA

M. Engdahl (ESA Tech. Officer)
 A. Minchella (RSAC c/o ESA)

Array Systems Computing

R.Jha, L.Veci, J. Lu, S. Dai

Brockmann Consult

N.Fomferra, M. Peters

PPO.labs (InSAR)

NEST-DORIS InSAR Module

(2009-2011)

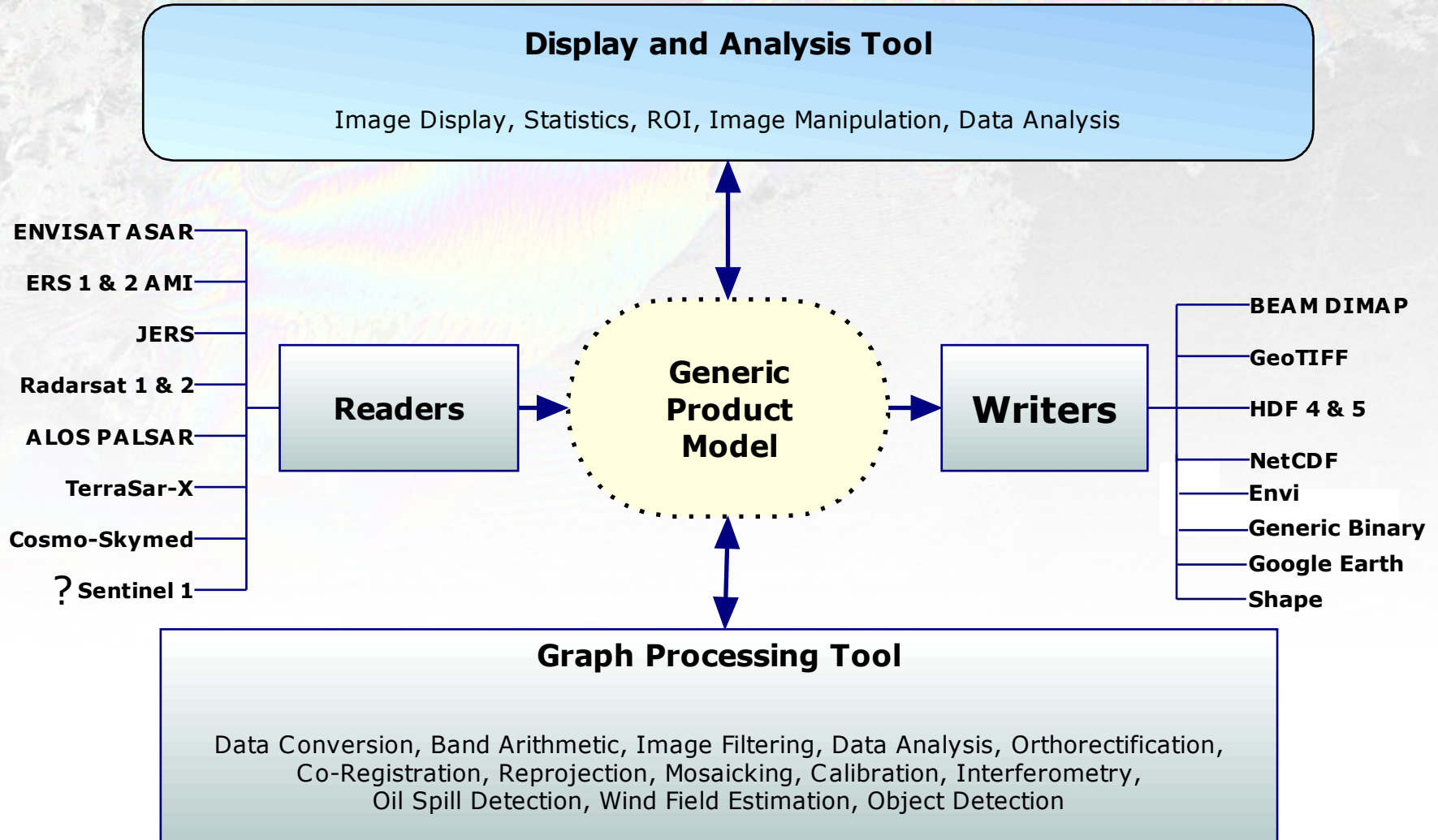
1st Release in 4A, 2nd Release in 4B

NEST is being developed by Array Systems Computing Inc. of Toronto Canada under ESA Contract number 20698/07/I-LG._

NEST Architecture highlights

- **Fully portable to multiple hardware platforms and operating systems (Windows, Linux, Mac) thanks to a 100% pure Java implementation.**
- **Modular design for easy modifications and upgrades**
- **API enables users to easily add their own modules and extend the capabilities of NEST**
- **Multi-core processor support**
- **Fully-fledged processing environment suited for operational processing on computer clusters/clouds**
- **Fully Open Source under the GNU GPL**

NEST Modules: data flow



NEST DAT and Graphs (GPF)

The screenshot shows the ENVI SAR Processor interface. The 'Graphs' menu is open, with 'Batch Processing' highlighted. The 'Graph Builder' window displays a processing graph with the following steps:

```

graph TD
    1[1-Read] --> 2[2-Apply-Orbit-File]
    2 --> 3[3-Multilook]
    3 --> 4[4-Calibration]
    4 --> 5[5-Speckle-Filter]
    5 --> 6[6-Ellipsoid-Correction-RD]
    5 --> 7[7-Terrain-Correction]
    6 --> 8[8-LinearTodB]
    8 --> 10[10-Write]
    7 --> 9[9-LinearTodB]
    9 --> 11[11-Write]
  
```

The 'Source Product' field is set to: ENVISAT-ASA_IMP_1PNUPA20070910_211132_000000162061_00301_28912_1794.N1

- Create your own processing chains
- Visual Graph Processing Framework interface
- Executed from command line or from GUI
- Allows for *batch processing* on stack of images

NEST-DORIS Outline

- **Why?**

Context of NEST-DORIS

- **What?**

InSAR processing chain

- **How?**

Design and implementation of InSAR chain

- **When?**

Where we are...

What is NEST-DORIS?

Extension to NEST for:

Read

Process

Visualize

Interpret

Store

SAR and InSAR data

What is NEST-DORIS?

Extension to NEST for:

Read

Process

Visualize

Interpret

Store

SAR and InSAR data

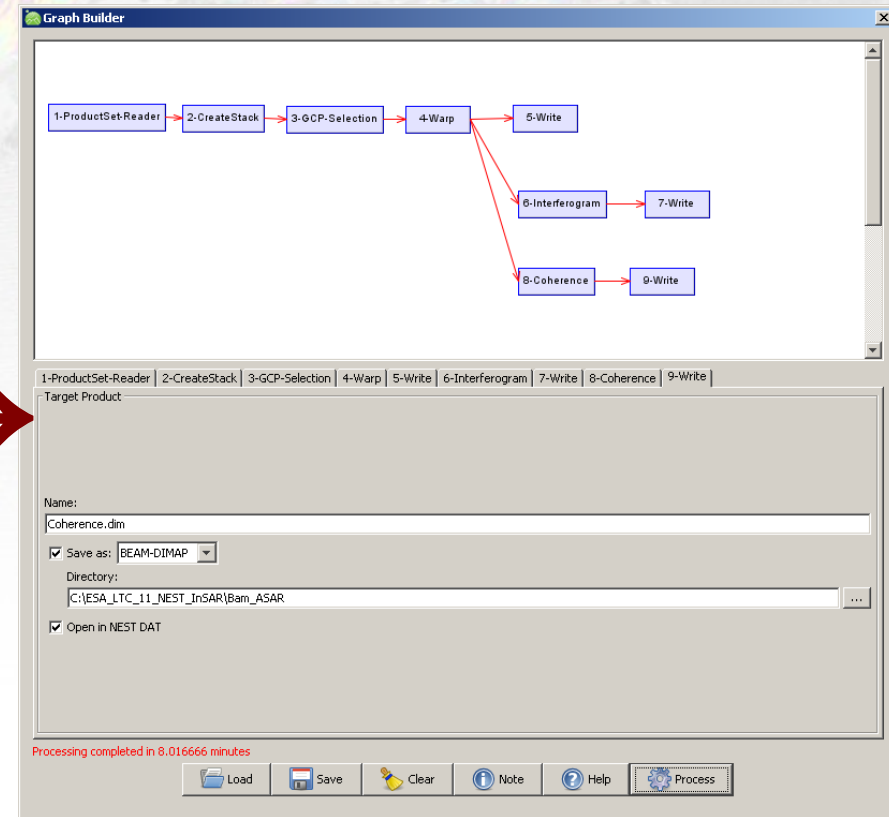
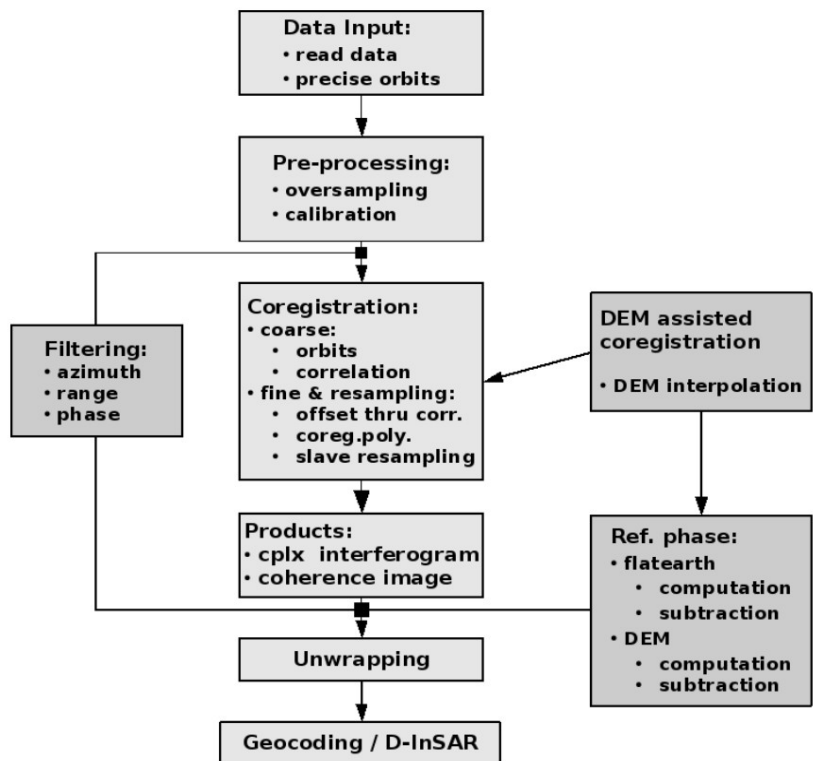
- **Algorithmic integration of DORIS processing chain**
- **Delft Object-Oriented Radar Interferometric Software**
 - TU Delft's open source InSAR processor
 - Fully operational
 - Written in C++
 - In (semi-active) development since 2000
 - Advanced algorithmic set

What is NEST-DORIS?

The screenshot displays the NEST-DORIS software interface with the following components:

- Products View:** A tree structure on the left showing two main product folders:
 - [1] Interferogram_srp_stack_ML
 - Identification
 - Metadata
 - Tie-point grids
 - Geometries
 - Bands
 - Intensity_ifg1_VV
 - Phase_ifg1_VV
 - [2] Interferogram_srp_stack_ML_TC
 - Identification
 - Metadata
 - Geometries
 - Bands
 - Intensity_ifg1_VV
 - Phase_ifg1_VV
 - Intensity_ifg1_VV_db
- Main View:** Two large panels showing the results of processing:
 - Top panel: [2] Intensity_ifg1_VV_db, displaying a grayscale interferogram of a terrain.
 - Bottom panel: [2] Phase_ifg1_VV, displaying a phase interferogram with a color-coded fringe pattern.
- Colour Manipulation - [2] Intensity_ifg1_VV_db:** A control panel at the bottom left with:
 - Editor: Sliders, Table, Discrete colors
 - Apply button
 - Statistics: Name: Intensity_ifg1_VV_db, Unit: intensity_db, Min: 39.204, Max: 124.208, Rough statistics!
 - Histogram with three vertical markers at 55.77, 81.46, and 97.15.
 - More Options button
 - Navigation, Colour, Layer M., and World Map buttons.
- Status Bar:** At the bottom, showing coordinates: [2] Intensity_ifg1_VV_db - WGS84(DD), No pos., 17:27:09, 355M of 911M.

NEST-DORIS objective



How NEST-DORIS is developed

Main design principle:

*by Principal Investigator
for Principle Investigator*

...having in mind 4 principles:

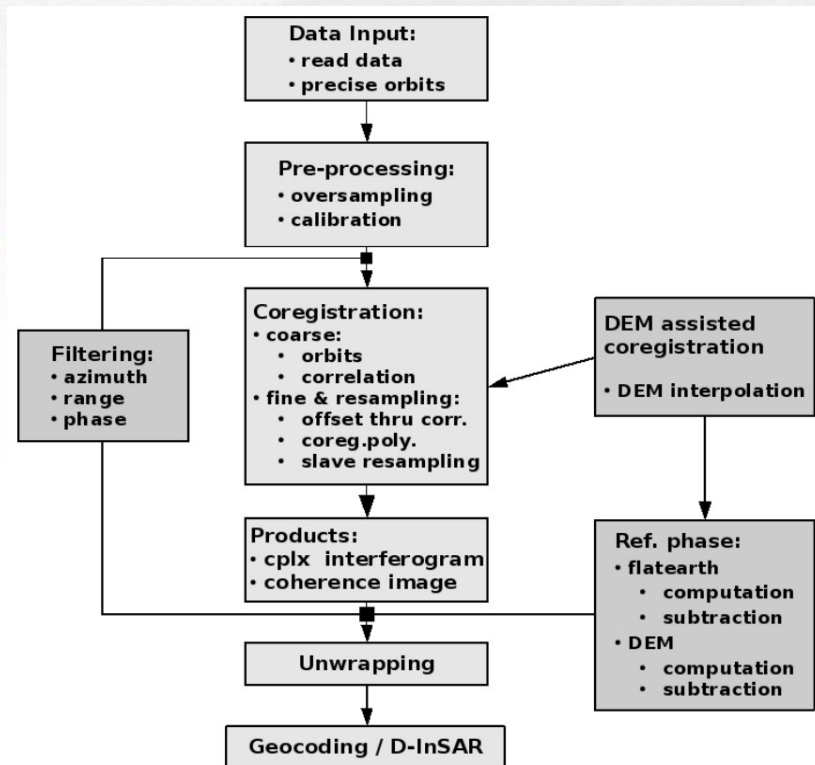
1. Design
2. Users
3. Brevity
4. Architecture

NEST-DORIS design

Four design principles

1. Designed by PI for PI

- Algorithms and implementation follow best practices
- Designed for “us” and friends



NEST-DORIS design

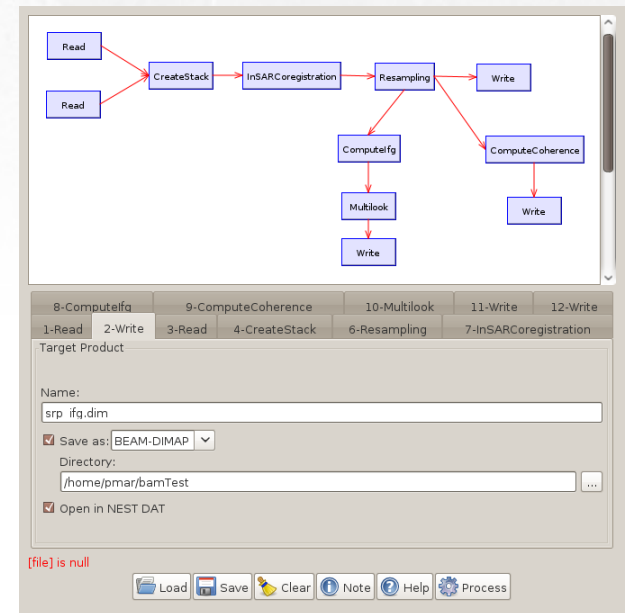
Four design principles

1. Designed by PI for PI

- Algorithms and implementation follow best practices
- Designed for “us” and friends

2. Users are given as much control as possible

- Users considered as an equal partner
- For both processing and development
- Graph Processing Framework



NEST-DORIS design

Four design principles

1. Designed by PI for PI

- Algorithms and implementation follow best practices
- Designed for “us” and friends

2. Users are given as much control as possible

- Users considered as an equal partner

3. Aimed for Brevity

- Anything that can be implicit it is
- Manual: clarifications, reservations, warnings, usual and special cases

NEST-DORIS design

Four design principles

1. Designed by PI for PI

- Algorithms and implementation follow best practices
- Designed for “us” and friends

2. Users are given as much control as possible

- Users considered as an equal partner

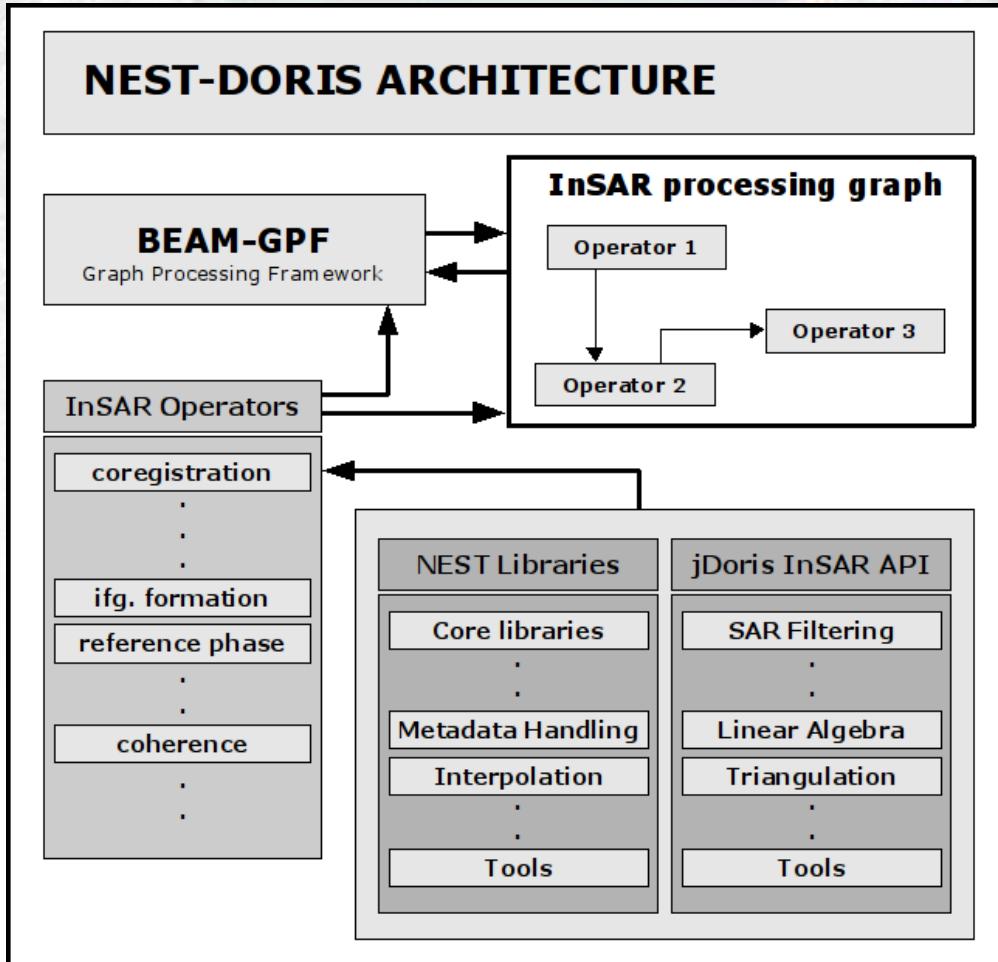
3. Aimed for Brevity

- Anything that can be implicit it is
- Manual: ...

4. Architected for research&development and education

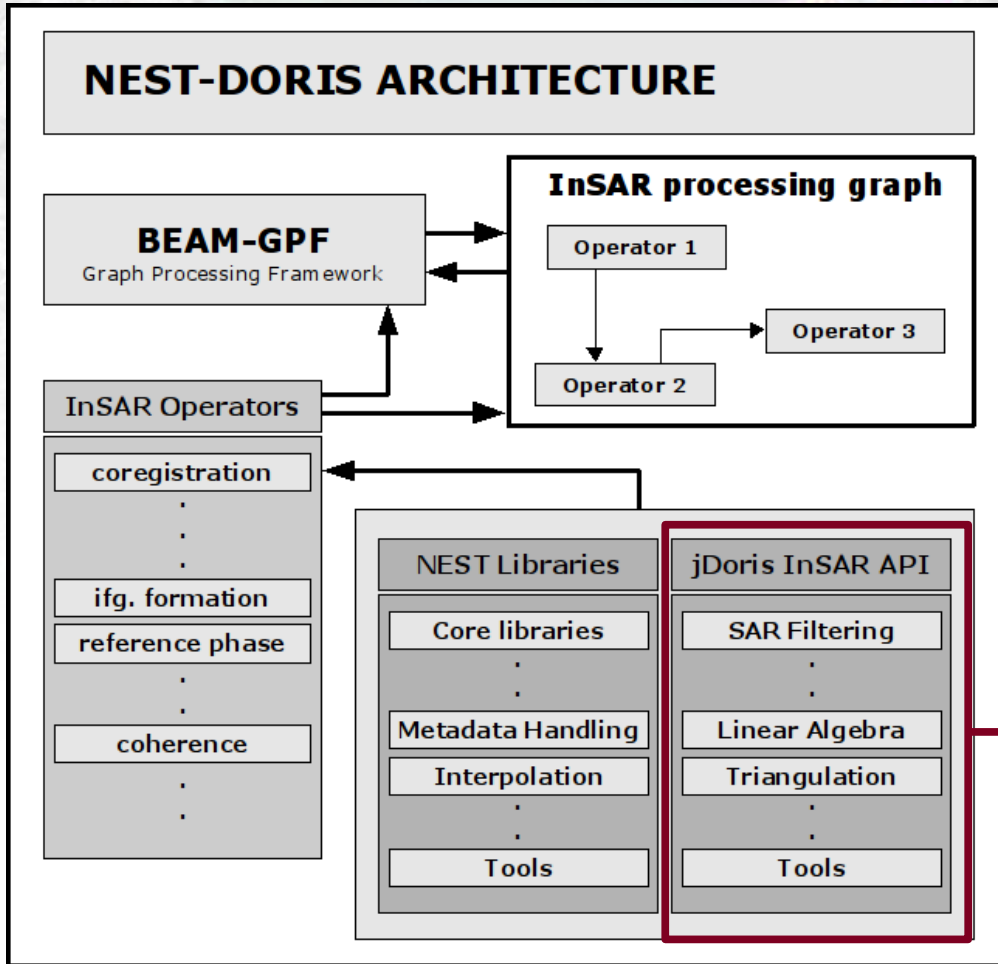
- More details on coming slides...

NEST-DORIS Architecture



- Standard architecture of NEST
- Graphic Processing Framework
- Operators built with libraries
 - BEAM & NEST
 - Other 3Rd party libs
 - jDoris

NEST-DORIS Architecture



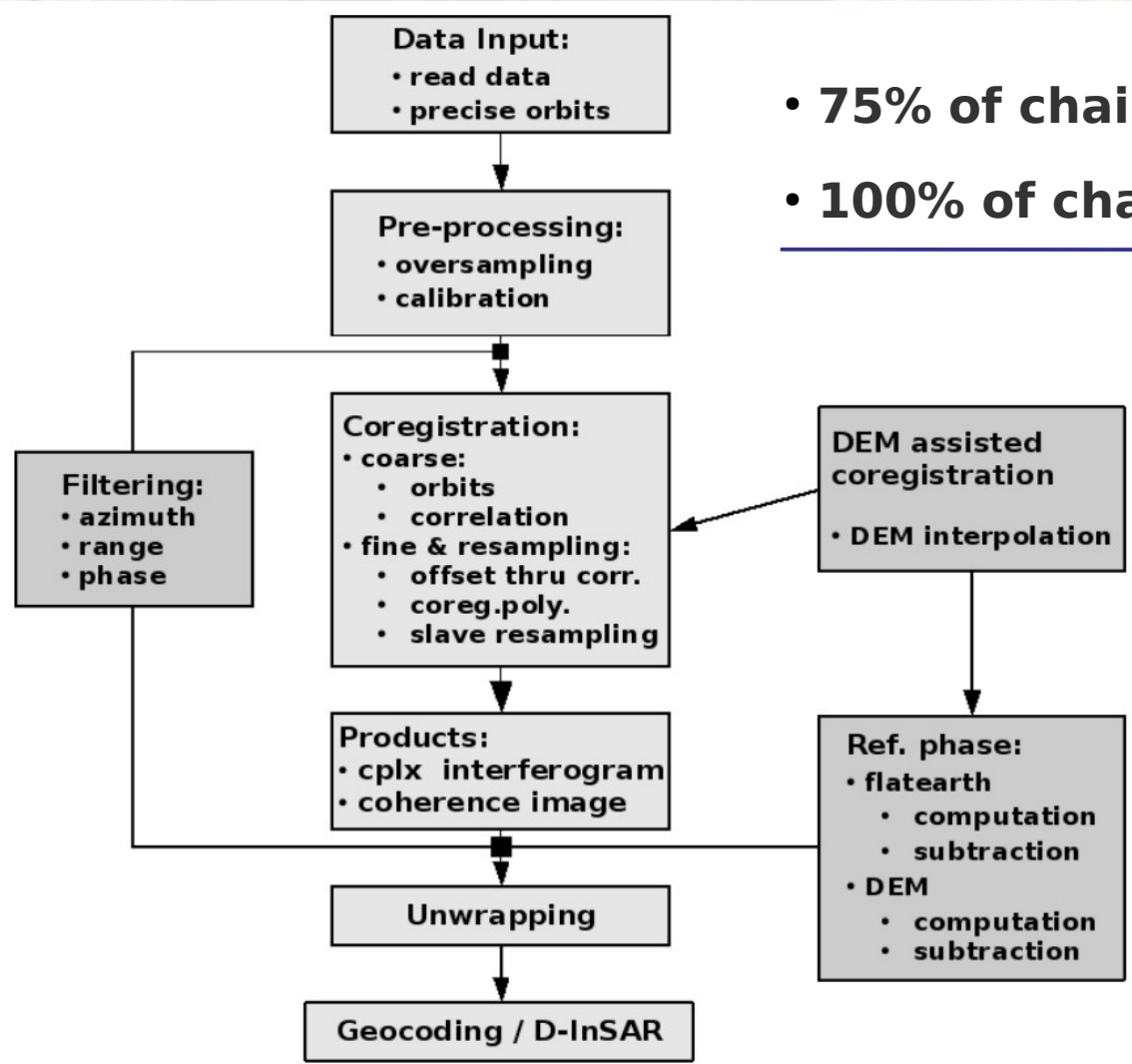
- Standard architecture of NEST
- Graphic Processing Framework
- Operators built with libraries
 - BEAM & NEST
 - Other 3Rd party libs
 - **jDoris (*new* for InSAR)**

Release of NEST-DORIS

- **Following the release schedule of NEST**
 - **NEST 4B: released September 2011**
 - Full InSAR processing chain
 - **NEST 4C: expected December 2011**
 - Advanced-full InSAR processing chain
-

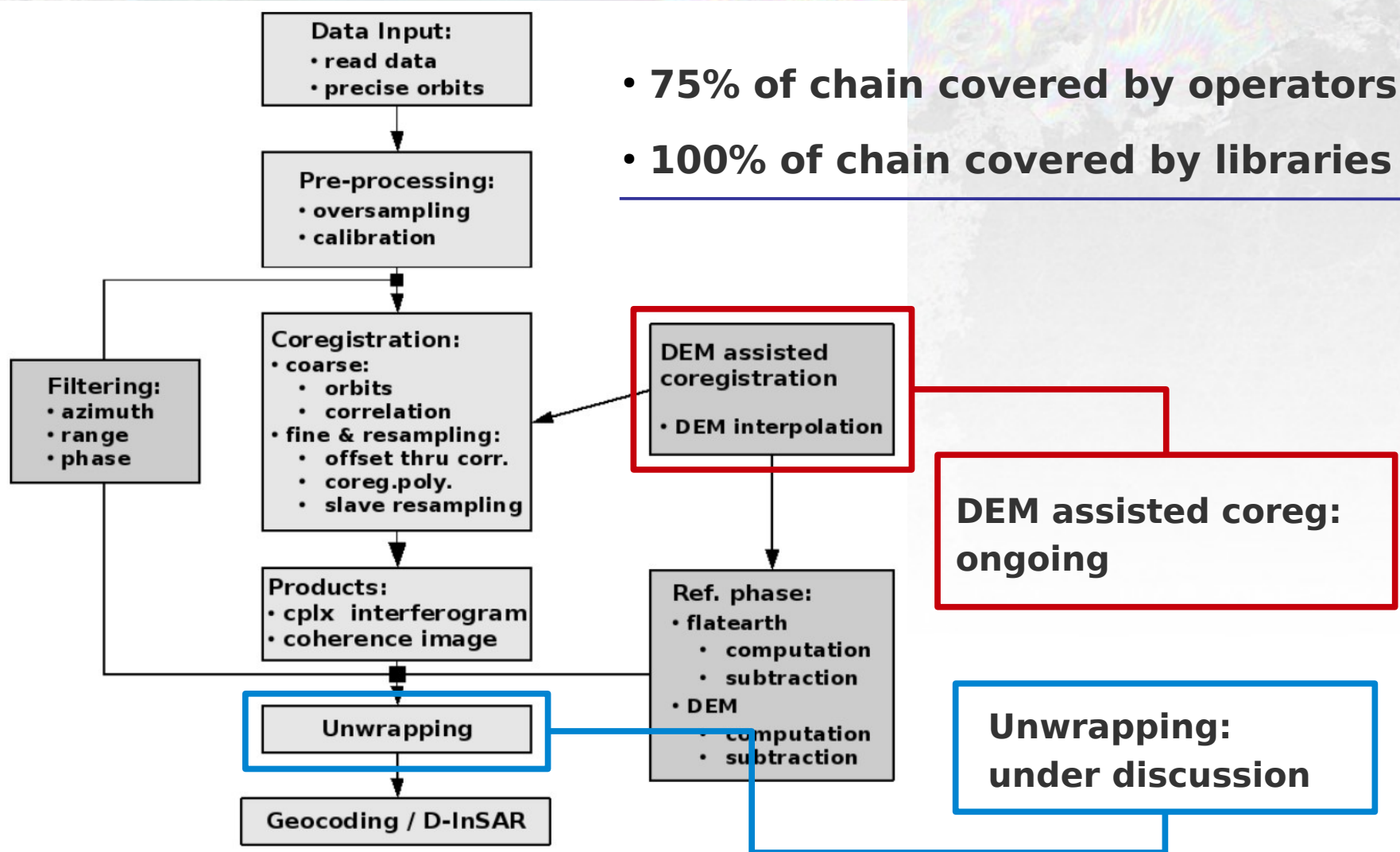
NEST-DORIS 4B release

- 75% of chain covered by operators
- 100% of chain covered by libraries

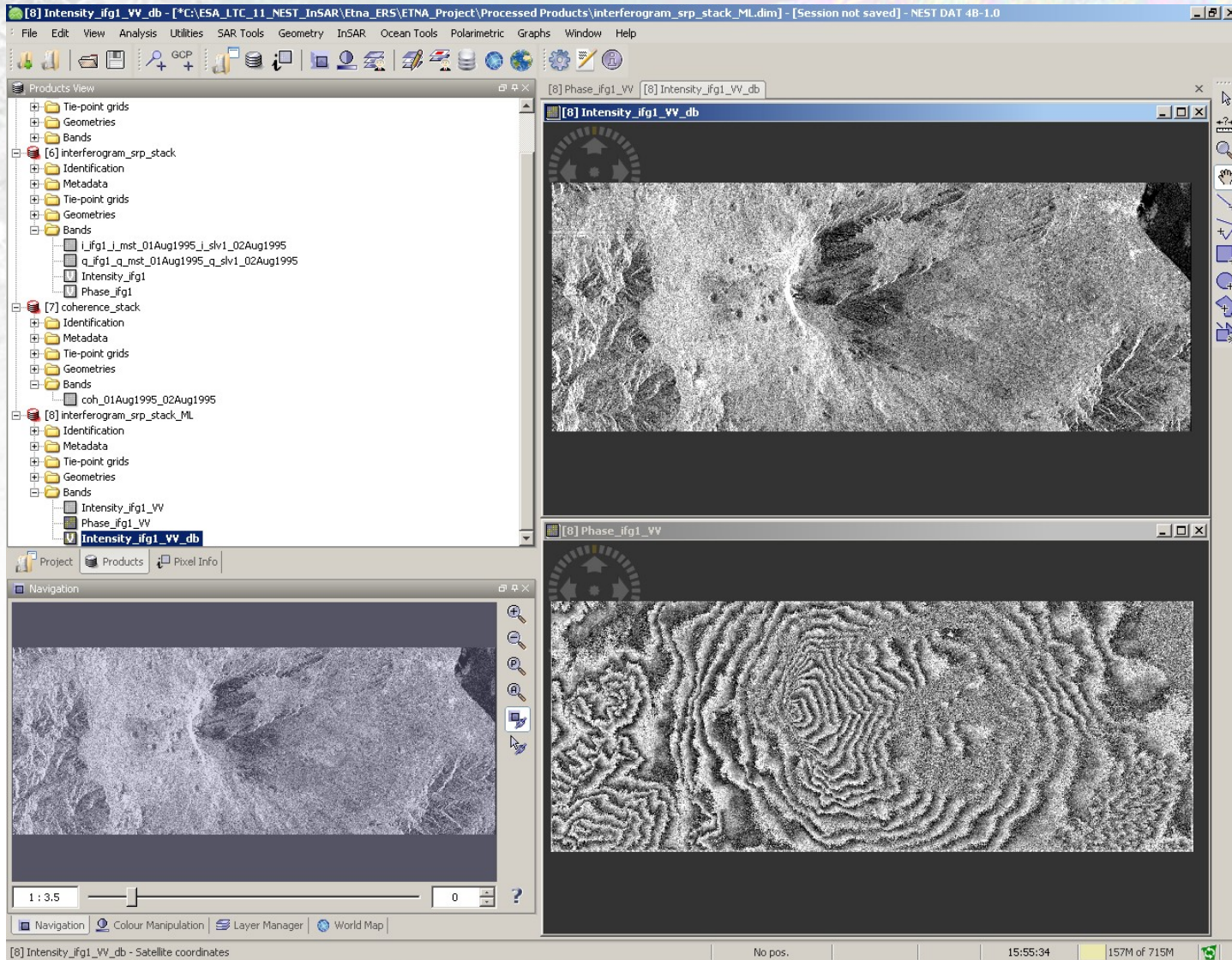


NEST-DORIS 4B release

- 75% of chain covered by operators
- 100% of chain covered by libraries



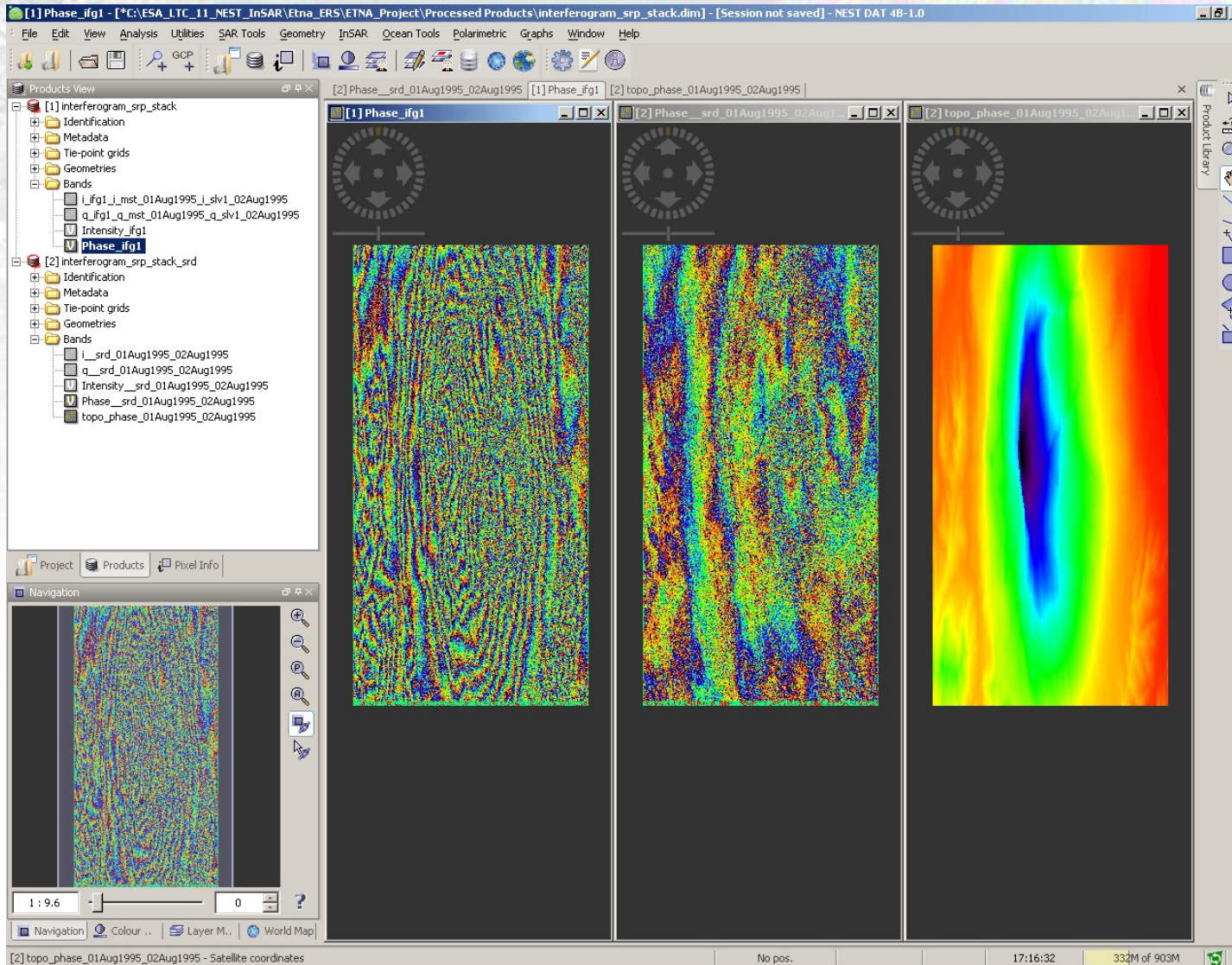
NEST-DORIS Examples



Highlights:

- Navigation
- Visualization
- Interpretation

NEST-DORIS Examples



Highlights:

- DEM handling
- Visualization
- Interpretation

NEST-DORIS Examples

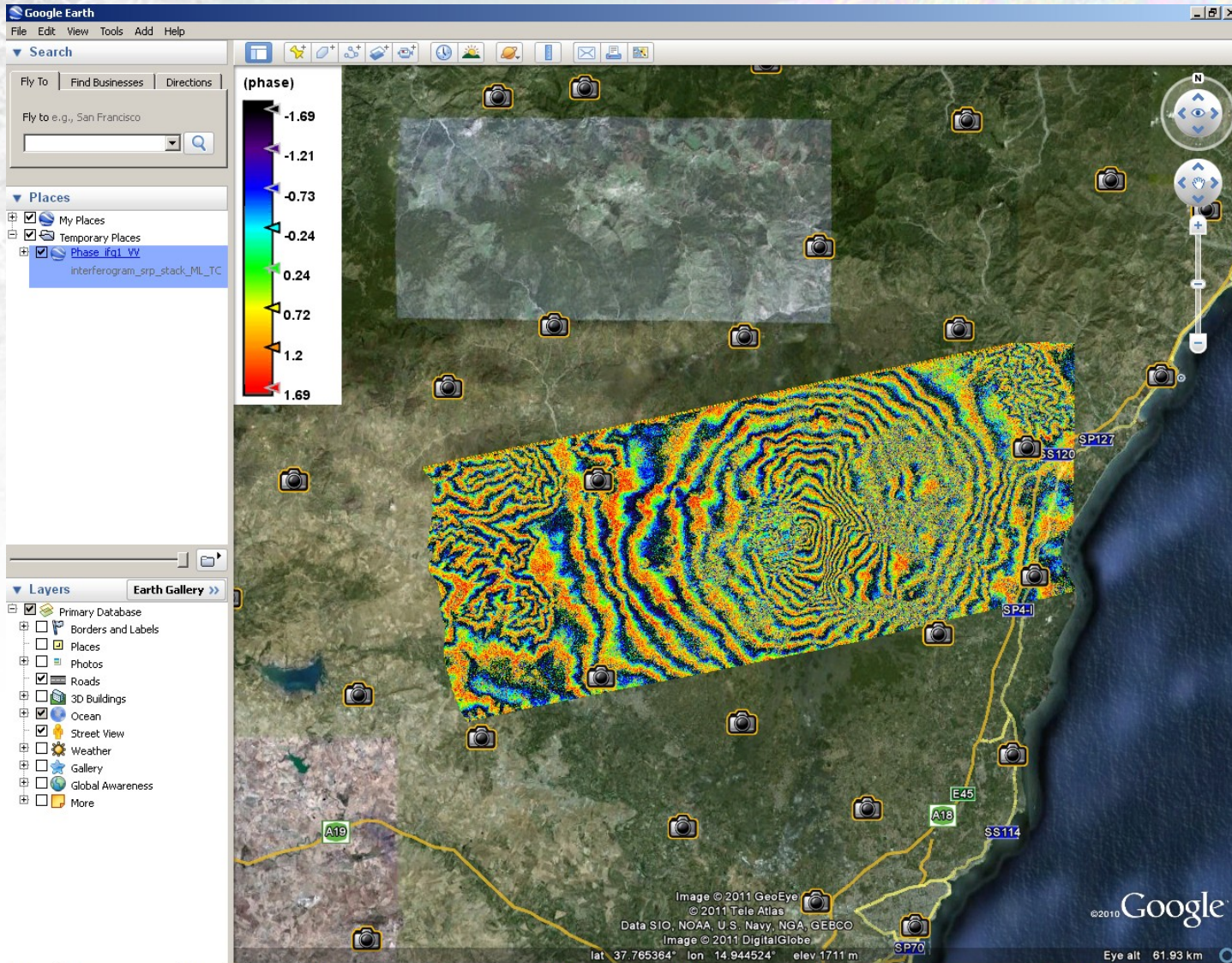
The screenshot displays the NEST-DORIS software interface with the following components:

- Products View:** A tree structure on the left showing project folders. The selected item is `[2] Intensity_ifg1_VV_db` under the `[2] interferogram_srp_stack_ML_TC` folder.
- Main View:** Two panels showing the results of processing:
 - Top panel: `[2] Intensity_ifg1_VV_db` showing a grayscale interferogram of a terrain.
 - Bottom panel: `[2] Phase_ifg1_VV` showing a phase interferogram with a colorful fringe pattern.
- Colour Manipulation - [2] Intensity_ifg1_VV_db:** A histogram window on the bottom left showing the distribution of intensity values. It includes:
 - Editor options: `Sliders` (selected), `Table`, `Discrete colors`.
 - Statistics: `Name: Intensity_ifg1_VV_db`, `Unit: intensity_db`, `Min: 39.204`, `Max: 124.208`, `Rough statistics!`
 - Value markers: `65.77`, `81.46`, `97.15`.
- Status Bar:** At the bottom, it shows `[2] Intensity_ifg1_VV_db - WGS84(DD)`, `No pos.`, `17:27:09`, and `355M of 911M`.

Highlights:

- Geo-coding
- Histograms

NEST-DORIS Examples



Highlights:

- Export
- Visualization

NEST-DORIS performance

- **Benchmarking**

- Performance comparable to C/C++ code
- Java version 7 → 20% performance improvement
- FFT libraries are bottleneck
- Performance critical blocks in C and linked to JAVA

- **Thread safe**

- Scaling is simplified

- **Reduced complexity of development and deployment**

- Faster
- Easier
- Maintainable

“Benchmarks are only as good as the programs they measure”

NEST-DORIS Summary

- **InSAR processor for R&D and education**
- **Full and 'stable' processing chain**
- **Mission support:**
 - SLC of all missions acquired in StripMap mode
- **Incubation platform for other**
- **Educational tool**
 - Already couple of courses
 - TU Delft

NEST InSAR Course Fringe 2011

www.array.ca/nest

www.ppolabs.com/nestdoris

“or simply google for NEST + InSAR + SAR”

jDoris Application Interface

- **Rich set of InSAR functionalities**
- **JAVA**
- **Designed for NEST**
 - extension not replacement
 - reusable
- **Open Source**
- **InSAR application driven**
 - Data stacks are reference → multi-image processing
 - InSAR Syntax and semantics
 - “learning API could be harder than learning the language”
 - Design Principle: *“If you are in doubt leave it out”*
- **Multi-core ready and thread-safe**