

InSAR Time Series Analysis of Sentinel-1 data Using GMTSAR and iGPS

Yunfeng Tian

tianyf@gmail.com

<https://www.ngs.noaa.gov/gps-toolbox/Tian.htm>

<https://github.com/igps-ftk/iGPS>

<https://sourceforge.net/projects/igps/>

2023-02-28

Content

- Prepare Input Data
- Install and Configure Software
- Processing flowchart
- Displacements and velocity analysis
- Atmospheric Delay Correction

- SAR Data
- Orbit
- DEM

PREPARE INPUT DATA

Input Data

Create a root directory (e.g., /g4d/esa.data) to hold all input data.

Set the environmental variable esa_data to this directory, e.g. in C-shell:

```
setenv esa_unzip /g4d/esa.data
```

- **Sentinel-1 C-band SAR SLC data** (e.g., S1A_IW_SLC__1S *.zip)
 - \${esa_data}/s1/ Files can be stored anywhere, linked to here.
- **Sentinel-1 Orbits (*.EOF)**
 - \${esa_data}/aux_poeorb POE Orbits
 - 2019
 - 2020
 - \${esa_data}/aux_resorb Rapid Orbits
 - 2019
 - 2020
- **DEM**
 - \${esa_data}/topo/strm3/

Obtain Sentinel-1 Data

- ESA website
 - <https://scihub.copernicus.eu/dhus/#/home>
- ASF website (recommended)
 - <https://search.asf.alaska.edu/>

First, create a free account of those websites. The username and password will be required upon downloading request.

The downloaded zip files can be stored anywhere, but linked to \${esa_data}/s1/ directory.

Obtain Sentinel-1 Data

Get the list of data using ASF Vertex interface.

Save the download python script (*.py) to the data directory.

The screenshot shows the ASF Vertex interface. At the top, there's a search bar with "Search Type: Geographic Search" and "Dataset: Sentinel-1". Below the search bar is a map of a mountainous region in Tibet, specifically around Shigatse. A red rectangular polygon is drawn on the map, indicating the area of interest. The map interface includes tools for zooming, panning, and setting the area of interest. Below the map, it says "8 Scenes (8 of 8 Files)". On the right side, there's a "Scene Detail / 1 File" panel for a specific file: "S1A_IW_SLC_1SDV_20230211T115825_20230211T115852_047188_05A96E-DD9B". It shows details like Start Time (02/11/23, 11:58:25Z), Stop Time (02/11/23, 11:58:52Z), Beam Mode (IW), Path (41), Flight Direction (ASCENDING), Polarization (VV+VH), and Absolute Orbit (47188). It also notes "Data courtesy of ESA". At the bottom of the interface, there are buttons for "SEARCH", "Baseline", "SBAS", and "More Like This".

ASF Python download script file

Output path

© 2023 ASF | Contact | Non-Discriminatory

iGPS provides a shell script to download the files listed in the python script, e.g.

```
sh_asf_s1_get_slc_py -file download-all-2023-02-11_12-16-04.py -opath `pwd`  
-aria2_conf your_login_name_password_file
```

sh_asf_s1_get_slc_py

Uses aria2 utility to download the Sentinel-1 SLC files from ASF.

It requires three command-line options:

1. -file download-all-.py

Python downloading script obtained from ASF. Used to extract the SLC filenames.

2. -opath your_output_path

Output path on your hard disk.

3. -aria2_conf your_login_name_password_file

Your login information stored in a file.

e.g. aria2.conf:

http-user=tian

http-passwd=123

Get the Satellite Orbits from ESA

<https://scihub.copernicus.eu/gnss/#/home>

Public account:

Username: gnssguest

Password: gnssguest

Orbits files can be downloaded using iGPS script

[sh_esa_s1_get_aux_orb_gnss](#)

E.g., download the POE orbits in the past 60 days,

[sh_esa_s1_get_aux_orb_gnss -orbit poe -ndays 60](#)

sh_esa_s1_get_aux_orb_gnss -d 2018-12-03

sh_esa_s1_get_aux_orb_gnss -orbit res

sh_esa_s1_get_aux_orb_gnss -orbit poe -ndays 10

Get the Satellite Orbits from ASF

https://s1qc.asf.alaska.edu/aux_poeorb/

https://s1qc.asf.alaska.edu/aux_resorb/

All files are stored under the directory. It takes a while for listing.

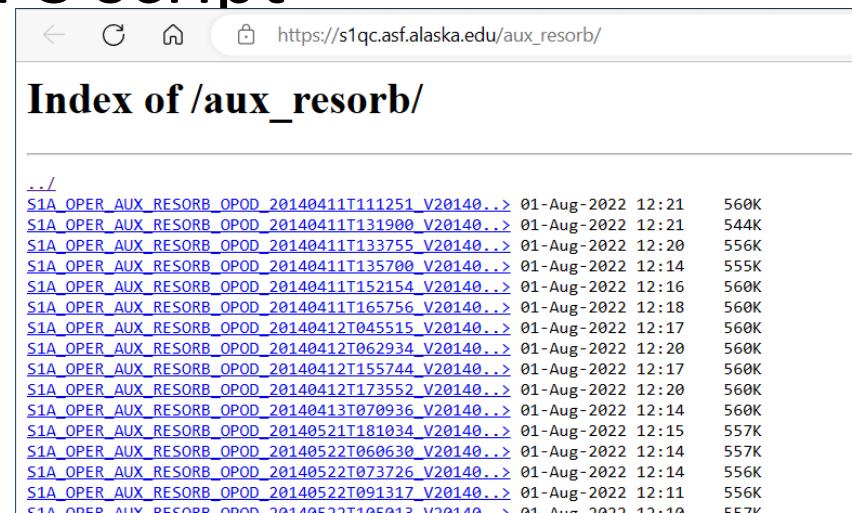
Can be downloaded using iGPS script

[sh_asf_s1_get_aux_orb](#)

This script use **wget** to download files. You may store the EARTHDATA account and password in a **wget.conf** file and pass it to the script:

http_user=xxx

http_password=yyy



The screenshot shows a web browser window with the URL https://s1qc.asf.alaska.edu/aux_resorb/. The page title is "Index of /aux_resorb/". Below the title, there is a list of files with their names, last modified dates, and sizes. The files are listed in chronological order from newest to oldest.

.. /	S1A_OPER_AUX_RESORB_OPOD_20140411T111251_V20140..>	01-Aug-2022 12:21	560K
	S1A_OPER_AUX_RESORB_OPOD_20140411T131900_V20140..>	01-Aug-2022 12:21	544K
	S1A_OPER_AUX_RESORB_OPOD_20140411T133755_V20140..>	01-Aug-2022 12:20	556K
	S1A_OPER_AUX_RESORB_OPOD_20140411T135700_V20140..>	01-Aug-2022 12:14	555K
	S1A_OPER_AUX_RESORB_OPOD_20140411T152154_V20140..>	01-Aug-2022 12:16	560K
	S1A_OPER_AUX_RESORB_OPOD_20140411T165756_V20140..>	01-Aug-2022 12:18	560K
	S1A_OPER_AUX_RESORB_OPOD_20140412T045515_V20140..>	01-Aug-2022 12:17	560K
	S1A_OPER_AUX_RESORB_OPOD_20140412T062934_V20140..>	01-Aug-2022 12:20	560K
	S1A_OPER_AUX_RESORB_OPOD_20140412T155744_V20140..>	01-Aug-2022 12:17	560K
	S1A_OPER_AUX_RESORB_OPOD_20140412T173552_V20140..>	01-Aug-2022 12:20	560K
	S1A_OPER_AUX_RESORB_OPOD_20140413T070936_V20140..>	01-Aug-2022 12:14	560K
	S1A_OPER_AUX_RESORB_OPOD_20140521T181034_V20140..>	01-Aug-2022 12:15	557K
	S1A_OPER_AUX_RESORB_OPOD_20140522T060630_V20140..>	01-Aug-2022 12:14	557K
	S1A_OPER_AUX_RESORB_OPOD_20140522T073726_V20140..>	01-Aug-2022 12:14	556K
	S1A_OPER_AUX_RESORB_OPOD_20140522T091317_V20140..>	01-Aug-2022 12:11	556K
	S1A_OPER_AUX_DECODE_OPOD_20140522T105013_V20140..>	01-Aug-2022 12:10	557K

sh_asf_s1_get_aux_orb example

- To download the POE orbits (default) in the past 30 days,
sh_asf_s1_get_aux_orb -orb poe -n 30
- To download the RES orbits in the past 3 days,
sh_asf_s1_get_aux_orb -orb res -n 3
- To use your own EARTHDATA login information,
**sh_asf_s1_get_aux_orb -wget_conf
/your/wget/username_password_file**
- Do not update page listing (if no changes in ASF server; ASF put all files in one directory, which makes it takes a while to list files):
sh_asf_s1_get_aux_orb -u n

Local Orbit Data

- The local orbit files are stored by year, because the search is usually too slow and sometime fails if put so many files in one directory.

▶	📁 aux_poeorb	16 items	Folder	18:39
▼	📁 aux_resorb	9 items	Folder	Sat
▼	📁 2020	50 items	Folder	3 Jul 2021
📄	S1A_OPER_AUX_RESORB_OPOD_20200101T045018_V20200101T004336_20200101T040106.EOF	560.0 kB	Markup	1 Jan 2020
📄	S1A_OPER_AUX_RESORB_OPOD_20200101T050252_V20200101T004336_20200101T040106.EOF	560.0 kB	Markup	1 Jan 2020
📄	S1A_OPER_AUX_RESORB_OPOD_20200101T052453_V20200101T004336_20200101T040106.EOF	560.0 kB	Markup	1 Jan 2020

Download DEM

<https://topex.ucsd.edu/gmtsar/demgen/>

- Maximum range for single download: $4^\circ \times 4^\circ$
- SRTM3: 90-m ground pixel size
- SRTM1/ASTER1: 30-m

The screenshot shows the GMTSAR website's "Generate DEM files for use with GMTSAR" page. The left sidebar includes links for Home, Downloads, Generate DEM, SAF InSAR Time Series, Contact, and Links (Get SAR data, GMT Hawaii). The main content area has a heading "Generate DEM files for use with GMTSAR". It contains instructions: "Enter boundary coordinates and hit generate to run the DEM generate script. The boundary cannot span more than 4 longitude by 4 latitude degrees. Three land data bases are available: SRTM1 - global but only extends to latitudes of +/-60 degrees, 30 m resolution; SRTM3 - global but only extends to latitudes of +/-60 degrees, 90 m resolution; ASTER1 - global but lower quality, 30 m resolution. (Note the SRTM data has been updated to V3 <http://www2.jpl.nasa.gov/srtm/>). This process is slow (10 minutes) so please be patient. When it is done hit "Download" below." Below this, there is a "Check status" section with a map showing coordinates (31.9, -115, -113, 27.5), a dropdown menu set to "SRTM3", a text input field containing "6", and a "north" button. Navigation buttons at the bottom are labeled 1, 5, and 9.

- Make sure it fully covers your area; otherwise strange fringes related to topography may appear in phase.

China Mosaic for SRTM-3

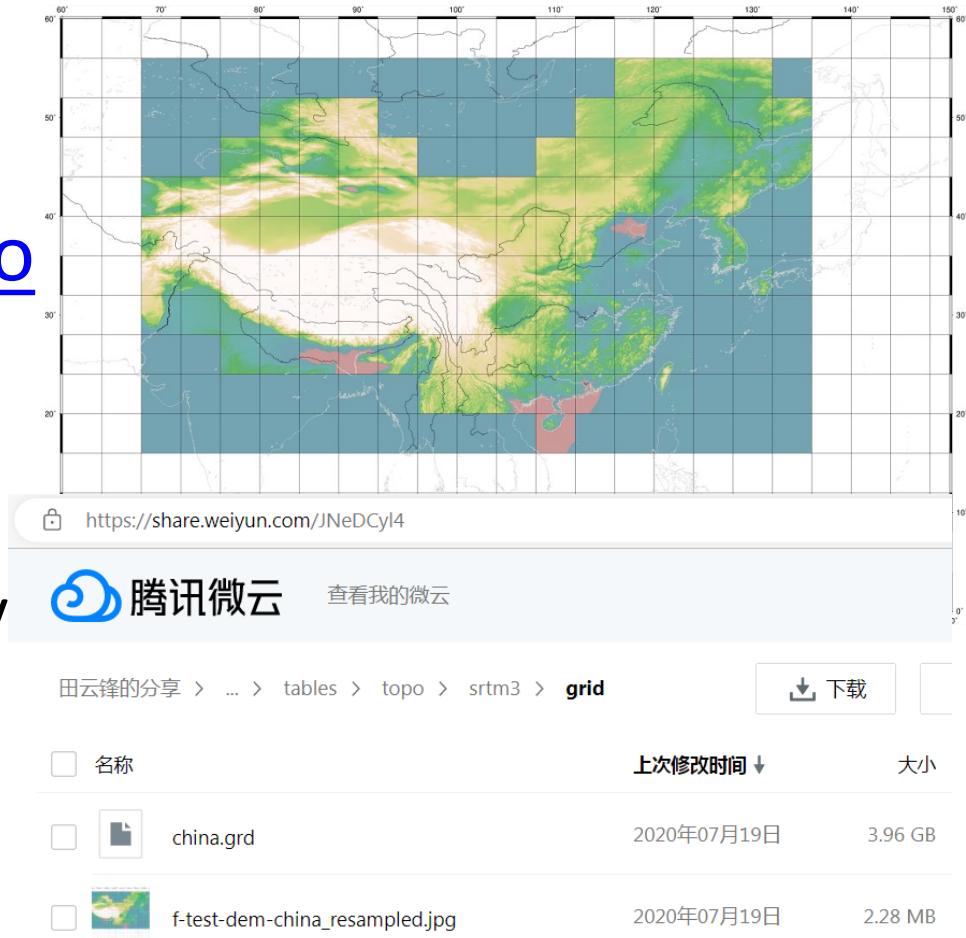
Obtain china.grd file from

- Weiyun:
<https://share.weiyun.com/JNeDCyl4>

- Save it to
\$esa_data/topo/srtm3/
grid/

Use grdcut to clip the file
for your ROI

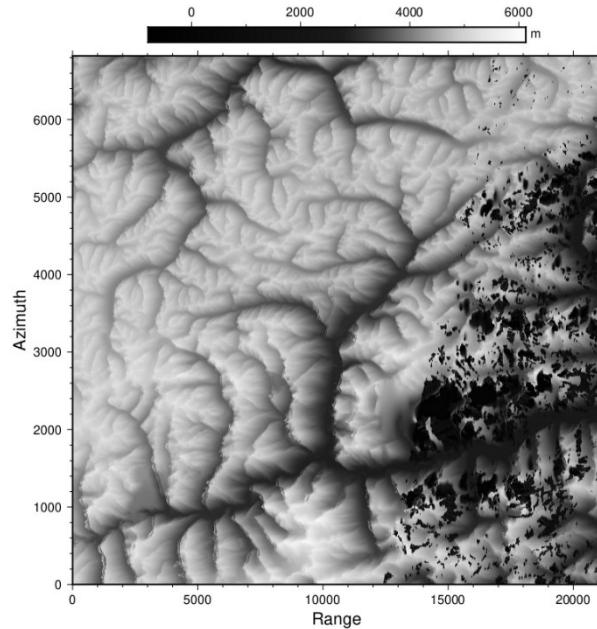
```
gmt grdcut china.grd -Gdem.grd -Rlon1/lon2/lat1/lat2
```



iGPS script **sh_s1_run_tsa** can perform the clipping automatically according to the frame KML files.

Verify DEM

- Old versions of SRTM may have holes.



- Check the topo_ra.pdf file in topo directory (after the first interferogram processed) and make sure it looks OK.

- GMT
- GMTSAR
- iGPS
- Dependencies

SOFTWARE INSTALLATION

Linux OS

First, you need a Linux OS system. Choose a stable release as you like.

- CentOS 7,8; Rocky
- Ubuntu

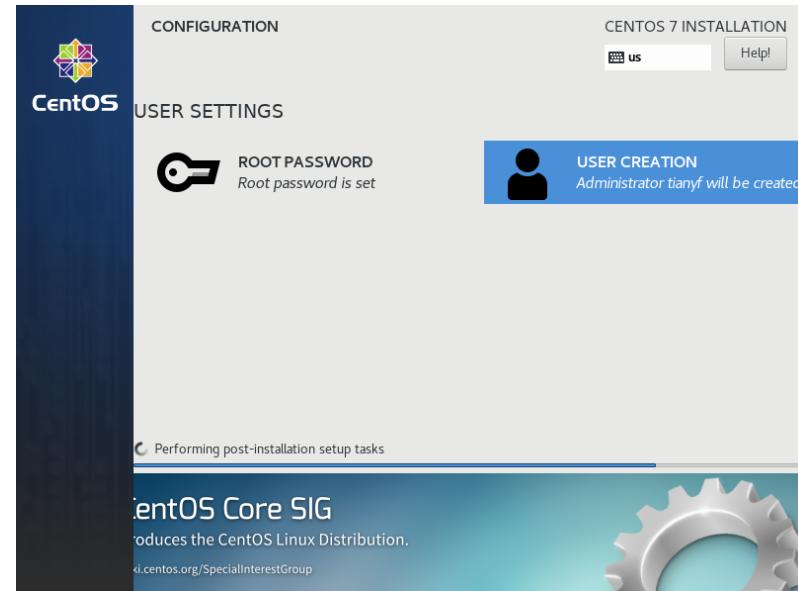


Example: Centos 7

<https://www.centos.org/download/>

To save downloading time, get the minimum distribution ISO file, e.g. CentOS-7-x86_64-Minimal-2207-02.iso

Install the system with default settings is OK.
We can add others later.



Example: CentOS 7

Install the following packages:

- `yum groupinstall "GNOME Desktop" "Server with GUI" "Development Tools" "Compatibility Libraries"`
- `yum install epel-release`
- `yum install svn autoconf gcc-c++ libtiff-devel gmt-devel gdal.x86_64 gdal-devel cmake subversion libcurl-devel netcdf-devel pcre-devel fftw3-devel lapack-devel openblas-devel aria2 wget gv dos2unix parallel`

Install GMT 6 (optional)

You can use GMT coming with OS (often a little old, e.g. GMT5 in Centos 7).

If not, install an version you like (e.g. GMT 6).

- <https://www.generic-mapping-tools.org/>
- Refer to the webpage for installation instructions.



e.g., GMT 6.0.0

You need:

- dcw-gmt-1.1.4.tar.gz
- GMT-6.0.0.tar.gz
- gshhg-gmt-2.3.7.tar.gz

Extract them into /usr/local as root user.

Install GMTSAR

- <https://github.com/gmtsar/gmtsar>
- <https://github.com/gmtsar/gmtsar/wiki/GMTSAR-Wiki-Page>

Refer to the Wiki page for installation instructions.

Test samples:

- <https://topex.ucsd.edu/gmtsar/downloads/>

- Download the gmtsar and extract to /usr/local/gmtsar-master
- Run
 - autoconf
 - ./configure
 - make
 - make install (if no error occurred in previous step)

Add /usr/local/gmtsar-master/bin to searching path in
~/.cshrc (for tcsh):

```
set path = ($path /usr/local/gmtsar-master/bin)
```

Try a command (e.g., sbas) to see whether it works.

Install iGPS

Obtain the package from

- <https://www.ngs.noaa.gov/gps-toolbox/Tian.htm>
- <https://github.com/igps-ftk/iGPS> (updated more frequently)
- <https://sourceforge.net/projects/igps/>

Extract the package to a directory. Add the following to the user variable \$PATH.

- \$iGPS/sh
- \$iGPS/ftk/bin/Linux_x86_64 (if you are using X64 Linux)

E.g., for tcsh (~/.cshrc):

```
setenv iGPS ${HOME}/iGPS
set path = ($path ${iGPS}/sh ${iGPS}/ftk/bin/Linux_x86_64)
```

Compile the ftk source code. Enter \$iGPs/ftk, run

```
./install_igpsftk
```

Or, if no need to compile the source (simply create links for existing executables).

```
./install_igpsftk_links
```

Try *hello* to check whether it works.

```
[tianyf@bogon ~]$ hello
Usage: hello ofile.name
hello: test whether the compiled FORTRAN77 code support large(>99) file unit.
```

Install GAMIT/GLOBK(optional)

- iGPS now has a Fortran ToolKit (ftk)
It calls some functions in GAMIT/GLOBK software.
 - The gplib.a files compiled using gfortran can be found in \$iGPS/ftk/gplib/.
 - If you have GAMIT/GLOBK installed, use ***arlib*** to create your own library file.
- Some iGPS Shell scripts utilize the doy executable in GAMIT software, for conversion between various formats of date
Note: the day-of-year from ***doy*** utility and GMTSAR differ by one day.
A sample doy executable can be found in \$iGPS/ftk/bin/Linux_x86_64.

```
[tianyf@bogon ~]$ doy 2023 1
Date 2023/01/01 0:00 hrs, DOY 1 JD 2459945.5000 MJD 59945.0000
GPS Week 2243 Day of week 0, GPS Seconds 0 Day of week Sun
Decimal Year 2023.000000000 GRACE Seconds 725803200.0
```

Add executable paths

If you use C-Shell (e.g., tcsh), add the following lines to your `~/.cshrc` file.
Please change the version number accordingly.

```
#for GMT (if install your own version)
set path = ($path  /usr/local/GMT-6.0.0/bin)
```

```
#for GMTSAR
set path = ($path  /usr/local/gmt5sar20200529/bin)
```

```
#for iGPS
setenv iGPS ${HOME}/iGPS
set path = ($path  ${iGPS}/sh  ${iGPS}/ftk/bin/Linux_x86_64)
```

```
#for GAMIT/GLOBK (if present)
set path = ($path  ${HOME}/gg/com  ${HOME}/gg/gamit/bin  ${HOME}/gg/kf/bin)
Please create a soft link to the GAMIT/GLOBK path in home directory first,
ln -s /usr/local/gamit-v10.71    ~/gg
```

Validate your installation

In the terminal, type the following commands to test whether the software is working.

- GMT
gmt
- GMTSAR
sbas
- iGPS
hello
- GAMIT/GLOBK (optional)
doy

If no errors occur, we are ready to go.

Summary of Data

- Sentinel-1 SLC files

`$esa_data/s1/S1?_IW_SLC__1S*.zip`

Tool:

- `sh_asf_s1_get_slc_py`

- Orbit

`$esa_data/aux_poeorb/20??/S1A_OPER_AUX_POEORB_OPOD_*.EOF`

`$esa_data/aux_resorb/20??/S1A_OPER_AUX_RESORB_OPOD_*.EOF` (for latest data)

Tool:

- `sh_asf_s1_get_aux_orb`
- `sh_esa_s1_get_aux_orb_gnss`

- DEM

`$esa_data/topo/srtm3/grid/china.grd`

~4.0GB

If use your own data (`dem.grd`), it is not necessary.

Summary of Software

- GMT
- GMTSAR
- iGPS
 - IDL (optional)
- GAMIT/GLOBK (optional)
- Tools
 - aria2, wget, curl

Set Environment Variables

- `esa_data`

Where are the data?

- `esa_unzip`

Temporary directory to unzip the SLC zip files.

- `iGPS`

Where is iGPS installed?

Set Environment Variables

- Example

For C Shell (e.g., tcsh), in `~/.cshrc`:

```
setenv esa_data /g4d/esa.data
setenv esa_unzip /sar/tmp_esa_unzip

set GMT=/usr/local/GMT-6.0.0
set path=($GMT/bin $path)

setenv GMTSAR /usr/local/gmt5sar20220909
set path = ($GMTSAR/bin $path)

setenv iGPS /home/tianyf/iGPS
set path = ($path $iGPS/sh $iGPS/ftk/bin/Linux_x86_64)

#gamit/globk
set path = (~/gg/com ~/gg/gamit/bin ~/gg/kf/bin $path)
setenv HELP_DIR ~/gg/help/
```

Now we go ahead ...

CONFIGURATION FILES FOR PROCESSING

Before Processing: Configuration Files

- Processing path (e.g., /sar/proc/test_expt)
 - batch_tops.config (GMTSAR configuration file)
 - input.lst (S1 frame list)
 - dem.grd (DEM)
 - overlapping.*.kml (if no dem.grd given; optional)
 - pins.kml (two points along track direction; optional)
 - F1 (for iw1)
 - F2 (for iw2)
 - F3 (for iw3)
 - f123 (for merging iw1,iw2,iw3)

batch_tops.config

The most important parameter is the desired pixel size of output result, controlled by multi-looking factors.

- Full resolution (25m)
 - range_dec = 4
 - azimuth_dec = 1
 - For small landslide; mine subsidence; etc.
- Normal resolution (50m)
 - range_dec = 8
 - azimuth_dec = 2
 - For large landslide; regional ground subsidence.
- Full resolution (100m)
 - range_dec = 16
 - azimuth_dec = 4
 - For crustal deformation monitoring.

Change the filter_wavelength and dec_factor accordingly.

batch_tops.config by iGPS

There are several templates in iGPS

- \$iGPS/tables/
 - batch_tops.config.interseismic
 - batch_tops.config.landslide
 - batch_tops.config.normal

Ground Pixel Size: full resolution

- ~25m Localized deformation: landslide analysis

```
# filters
```

```
# look at the filter/ folder to choose other filters
```

```
filter_wavelength = 100
```

```
# decimation of images
```

```
# decimation control the size of the amplitude and phase images. It is  
either 1 or 2.
```

```
# Set the decimation to be 1 if you want higher resolution images.
```

```
# Set the decimation to be 2 if you want images with smaller file size.
```

```
#
```

```
dec_factor = 1
```

```
# for tops processing, to force the decimation factor
```

```
# recommended range decimation to be 8, azimuth decimation to be 2
```

```
range_dec = 4
```

```
azimuth_dec = 1
```

Ground Pixel Size: normal

- ~50m Regional deformation: balance between image size and pixel size

filters

look at the filter/ folder to choose other filters

filter_wavelength = **200**

decimation of images

decimation control the size of the amplitude and phase images. It is either 1 or 2.

Set the decimation to be 1 if you want higher resolution images.

Set the decimation to be 2 if you want images with smaller file size.

#

dec_factor = **1**

for tops processing, to force the decimation factor

recommended range decimation to be 8, azimuth decimation to be 2

range_dec = **8**

azimuth_dec = **2**

Ground Pixel Size: coarse

- ~100m Large-scale deformation: interseismic slip, coseismic displacements

```
# filters
```

```
# look at the filter/ folder to choose other filters
```

```
filter_wavelength = 400
```

```
# decimation of images
```

```
# decimation control the size of the amplitude and phase images. It is  
either 1 or 2.
```

```
# Set the decimation to be 1 if you want higher resolution images.
```

```
# Set the decimation to be 2 if you want images with smaller file size.
```

```
#
```

```
dec_factor = 2
```

```
# for tops processing, to force the decimation factor
```

```
# recommended range decimation to be 8, azimuth decimation to be 2
```

```
range_dec = 16
```

```
azimuth_dec = 4
```

iGPS time series analysis (tsa) tool: input.lst example

Sample input.lst file for combining 2 consecutive frames along the track direction.



	0	10	20	30	40	50	60	70	80	90	100	
1	S1A_IW_SLC_1SSV_20141022T114046_20141022T114115_002942_003578_A850.manifest.safe									day 1	1.0000000000000000	
2	S1A_IW_SLC_1SSV_20141022T114113_20141022T114141_002942_003578_125F.manifest.safe										1.0000000000000000	
3	S1A_IW_SLC_1SSV_20141115T114045_20141115T114115_003292_003D01_EC4F.manifest.safe									day 2	1.0000000000000000	
4	S1A_IW_SLC_1SSV_20141115T114113_20141115T114141_003292_003D01_B813.manifest.safe										1.0000000000000000	
5	S1A_IW_SLC_1SSV_20141209T114114_20141209T114114_003642_004506_E890.manifest.safe										1.0000000000000000	
6	S1A_IW_SLC_1SSV_20141209T114140_003642_004506_58C1.manifest.safe										1.0000000000000000	
7	S1A_IW_SLC_1SSV_20150102T114042T114114_003992_004CE8_DB58.manifest.safe										1.0000000000000000	
8	S1A_IW_SLC_1SSV_20150102T114111_003992_004CE8_A363.manifest.safe										1.0000000000000000	
9	S1A_IW_SLC_1SSV_20150126T114043_20150126T114113_004342_0054BA_0232.manifest.safe										1.0000000000000000	
10	S1A_IW_SLC_1SSV_20150126T114111_20150126T114138_004342_0054BA_BD58.manifest.safe										1.0000000000000000	
11	S1A_IW_SLC_1SSV_20150219T114042_20150219T114112_004692_005CC5_D916.manifest.safe										1.0000000000000000	
12	S1A_IW_SLC_1SSV_20150219T114110_20150219T114138_004692_005CC5_7FC7.manifest.safe										1.0000000000000000	
13	S1A_IW_SLC_1SSV_20150303T114043_20150303T114112_004867_006107_E0EF.manifest.safe										1.0000000000000000	
14	S1A_IW_SLC_1SSV_20150303T114110_20150303T114138_004867_006107_2D6F.manifest.safe										1.0000000000000000	
15	S1A_IW_SLC_1SSV_20150327T114043_20150327T114113_005217_00696A_915B.manifest.safe										1.0000000000000000	
16	S1A_IW_SLC_1SSV_20150327T114111_20150327T114128_005217_00696A_C000.manifest.safe										0.0706051741700842	

Blank first character

S1 frame names
used for processing

Percentage of
overlapping
between frames

Get input file list: extract metainfo files

iGPS use **.manifest.safe* information to get spatial coverage of frames.

- To extract *.safe files:

sh_s1_unzip_manifest output_path input_path

e.g.,

sh_s1_unzip_manifest `pwd` `pwd`

sh_s1_unzip_manifest /g4d/esa.data/safe /g4d/esa.data/s1

- Search for frames covering the same geographical area, using metainfo in *.safe files.
 - IDL code: s1_manifest_overlapping.pro
 - ftk tool: **esa_s1_manifest_overlap**

esa_s1_manifest_overlap example

```
$ esa_s1_manifest_overlap
```

```
Usage: esa_s1_manifest_overlap path_safes target_safe opath [perc_min]
```

For a given frame

(S1A_IW_SLC__1SDV_20170327T000151_20170327T000218_015870_01A273_941C.manifest.safe), run the command

```
esa_s1_manifest_overlap  
/cygdrive/c/Downloads/esa.data/metainfo/manifest.safe/D048  
S1A_IW_SLC__1SDV_20170327T000151_20170327T000218_015870_01A273_941C.manifes  
t.safe `pwd` .8
```

The output files are:

- overlapping.S1A_IW_SLC__1SDV_20170327T000151_20170327T000218_015870_01A273_941C.manifest.safe.kml
- overlapping.S1A_IW_SLC__1SDV_20170327T000151_20170327T000218_015870_01A273_941C.manifest.safe.psxy
- overlapping.S1A_IW_SLC__1SDV_20170327T000151_20170327T000218_015870_01A273_941C.manifest.safe.txt

Kml – to view in Google Earth; psxy – to plot with gmt; txt – frame list for iGPS

esa_s1_manifest_overlap

- Sample output
- Exclude some frames by commenting out (adding non-blank character, e.g. x, in the first column) the corresponding lines



*overlapping.S1A_IW_SLC_1SDV_20170327T000151_20170327T000218_015870_01A273_941C.manifest.safe.txt - Notepad

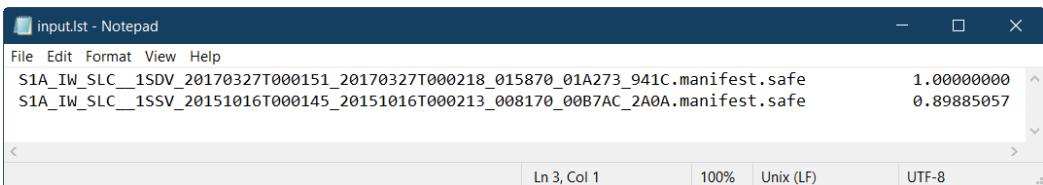
```
File Edit Format View Help
* source path: /cygdrive/c/Downloads/esa.data/metainfo/manifest.safe/D048
* target: S1A_IW_SLC_1SDV_20170327T000151_20170327T000218_015870_01A273_941C.manifest.safe
* minimum percentage of overlapping: 0.80000
* number of matched scenes: 171
S1A_IW_SLC_1SDV_20170327T000151_20170327T000218_015870_01A273_941C.manifest.safe 1.00000000
S1A_IW_SLC_1SDV_20170408T000152_20170408T000219_016045_01A7AC_5B49.manifest.safe 1.00000000
S1A_IW_SLC_1SDV_20170420T000152_20170420T000219_016220_01AD0F_E1A6.manifest.safe 1.00000000
x S1A_IW_SLC_1SDV_20170502T000153_20170502T000220_016395_01B25B_2D9D.manifest.safe 1.00000000
S1A_IW_SLC_1SDV_20170514T000154_20170514T000221_016570_01B7A9_1916.manifest.safe 1.00000000
S1A_IW_SLC_1SDV_20170526T000154_20170526T000221_016745_01BD09_E928.manifest.safe 1.00000000
S1A_IW_SLC_1SDV_20170607T000155_20170607T000222_016920_01C27F_EC2D.manifest.safe 1.00000000
S1A_IW_SLC_1SDV_20170619T000156_20170619T000223_017095_01C7E1_778C.manifest.safe 1.00000000
S1A_TW_S1C_1SDV_20170701T000156_20170701T000223_017270_01CD28_58CF.manifest.safe 1.00000000
```

Ln 8, Col 1 100% Unix (LF) UTF-8

INSAR PROCESSING

The first processing experiment

- \$HOME/gsar/test1st/
top level of directory
 - input.lst
 - frame list; two frames

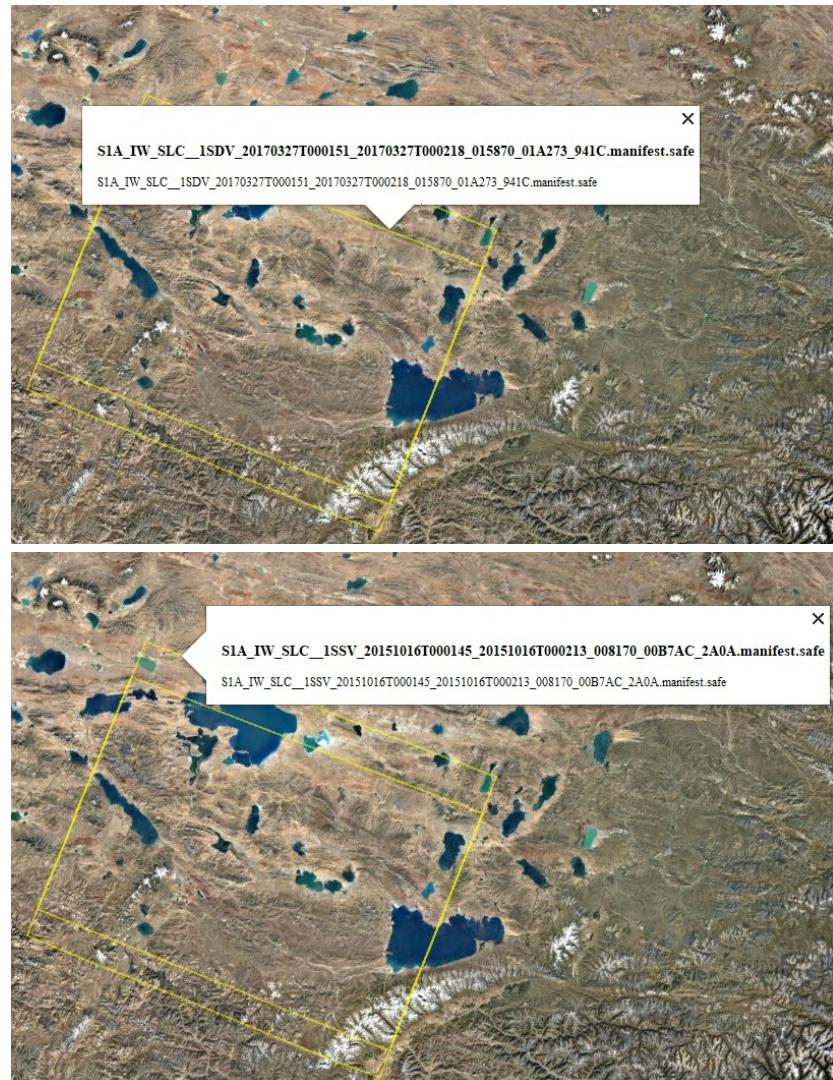


input.lst - Notepad

```
File Edit Format View Help
S1A_IW_SLC__1SDV_20170327T000151_20170327T000218_015870_01A273_941C.manifest.safe      1.00000000
S1A_IW_SLC__1SSV_20151016T000145_20151016T000213_008170_00B7AC_2A0A.manifest.safe      0.89885057

Ln 3, Col 1    100%    Unix (LF)    UTF-8
```

- overlapping.S1A_IW_SLC__1SDV_20170327T000151_20170327T000218_015870_01A273_941C.manifest.safe.kml
 - For clipping DEM from china.grd



Unpack SLC Zip Files

GMTSAR cannot handle compressed Sentinel-1 SLC files (*.zip) directly.

First, we uncompress the data to a temporary location (e.g., \$esa_unzip), using the sh_s1_unzip script:

- **sh_s1_unzip input.lst**

```
[tianyf@gpsac8 test1st]* sh_esa_s1_unzip input.lst
sh_esa_s1_unzip_node -path /g8g/esa,data/s1 -file input.lst -iw iw1,iw2,iw3 -u n
[sh_esa_s1_unzip_node]INFO:iw to extract is iw1 iw2 iw3
iw_typ is iw1
tmp is iw1 iw2 iw3
iw_typ is iw2
tmp is iw1 iw2 iw3
iw_typ is iw3
tmp is iw1 iw2 iw3
[sh_esa_s1_unzip_node]INFO:iw to exclude is
[sh_esa_s1_unzip_node]INFO:iw remain is  iw1 iw2 iw3
dos2unix: converting file input.lst to Unix Format ...
[sh_esa_s1_unzip_node]INFO: 1 / 2 S1A_IW_SLC__1SDV_20170327T000151_20170327T0002
18_015870_01A273_941C.manifest.safe
[sh_esa_s1_unzip_node]INFO: extracting S1A_IW_SLC__1SDV_20170327T000151_20170327
T000218_015870_01A273_941C.zip
unzip -n /g8g/esa,data/s1/S1A_IW_SLC__1SDV_20170327T000151_20170327T000218_01587
0_01A273_941C.zip -x **vh-* -d /g11e/esa_unzip/
Archive:  /g8g/esa,data/s1/S1A_IW_SLC__1SDV_20170327T000151_20170327T000218_01587
0_01A273_941C.zip
      creating: /g11e/esa_unzip/S1A_IW_SLC__1SDV_20170327T000151_20170327T000218_01
5870_01A273_941C.SAFE/
      creating: /g11e/esa_unzip/S1A_IW_SLC__1SDV_20170327T000151_20170327T000218_01
5870_01A273_941C.SAFE/support/
```

sh_s1_unzip

sh_s1_unzip script accepts 2 parameters

1. Data list file (e.g., input.lst)
2. Subswaths (iw1,iw2,iw3)
 - If not present, extract all subswaths;
 - Can be any combination of iw1, iw2, and iw3, e.g.,

```
sh_s1_unzip file
sh_s1_unzip file iw1
sh_s1_unzip file iw1,iw3
```

After the processing completes, use sh_s1_unzip_clean to delete the uncompressed data from \$esa_unzip directory, e.g.,

```
sh_s1_unzip_clean input.lst.ok
```

Interferometric processing

GMTSAR can only process only one subswath (e.g., iw1) each time.

If you want a whole frame, process each subswath (iw1/iw2/iw3) individually, then merge the interferograms together.

First, create a directory (e.g. F1) for processing iw1. Under F1, run:

sh_s1_run_tsa -file/input.lst -proc_type normal -iw iw1

```
[tianyf@gpsac8 F1]$ sh_esa_s1_run_tsa -file ..../input.lst -proc_type normal -iw iw1
[sh_esa_s1_run_tsa]INFO: session test1st
[sh_esa_s1_run_tsa]WARNING: DEM file ../dem.grd is mandatory and missing!
[sh_esa_s1_run_tsa]INFO: creating dem.grd file from KML files if available
[sh_esa_s1_run_tsa]INFO: geographical ranges 87.5167 91.1022 29.583 32.4332
[sh_esa_s1_run_tsa]INFO: topo range zmin zmax 0 7059.84472656
Make dem.kml and dem.png
gs -q -dNOPAUSE -dBATCH -dNOISAFER -dSCANCONVERTERTYPE=2 -dMaxBitmap=2147483647 -dUse
FastColor=true -dGraphicsAlphaBits=2 -dTextAlphaBits=4 -sDEVICE=pngalpha -g4303x342
0 -r1200 -sOutputFile='dem.png' './psconvert_5896d.eps'
/home/tianyf/gsar/test1st/F1
[sh_esa_s1_run_tsa]INFO: input file is /home/tianyf/gsar/test1st/F1/..../input.lst
[sh_esa_s1_run_tsa]INFO: subswath is iw1
[sh_esa_s1_run_tsa]INFO: ROI file is
[sh_esa_s1_run_tsa]INFO: default configuration file (/g0g/esa,data/config/batch_tops
.config.normal) copied.
[sh_esa_s1_run_tsa]INFO: 2 scenes to processing
[sh_esa_s1_run_tsa]INFO: (1) prepare the data (link & assemble/stitch ...)
[sh_esa_s1_run_tsa]INFO: output log to log.asmb
[sh_esa_s1_run_tsa]INFO: link orbit files to ./raw directory
[sh_esa_s1_run_tsa]INFO: return to /home/tianyf/gsar/test1st/F1
[sh_esa_s1_run_tsa]INFO: create spatiotemporal baseline information
```

It may take long time to align the images, create interferograms, and unwrap the phases!

Repeat the command for F2 (iw2) and F3(iw3) swaths.

"tsa stands for time series analysis "

What does sh_s1_run_tsa do?

- 1) Reading frame list to process (e.g., input.lst)
- 2) Copy configuration template (batch_tops.config) to frame root directory
- 3) Create and link dem.grd if not present
- 4) Under subswath directory (F1/F2/F3), link unpacked Sentinel-1 measurement file (*.tiff, *.xml) to raw0 directory. Link orbits (*.EOF) to raw0
- 5) Link or merge measurement files (*.tiff, *.xml) to raw directory, link orbits
- 6) Create data.in from files in raw.
- 7) Create spatial-temporal baselines (baseline*) using preproc_batch_tops.csh
- 8) In raw, align images using preproc_batch_tops_esd.csh
- 9) Create baseline file (get_baseline_table.csh) with aligned images
- 10) Create interferogram list (intf.in) from baseline_table.dat, and update batch_tops.config for master scene, using sh_s1_prep_proc_baseline
- 11) Call intf_tops.csh to processing interferograms
- 12) Create interferogram list files (intf.tab1, scene.tab1) from intf.in for SBAS, using sh_sar_prep_sbasis
- 13) Copy preview images of processed interferograms from intf_all to intf_all_png, using sh_sar_cp_intf_png
- 14) Create interferogram list files (intf.tab, scene.tab) for SBAS according to files in intf_all_png, using sh_sar_intf_tab_from_png
- 15) Run SBAS to derive displacements and velocities, using sh_sar_call_sbasis

Merge Subswaths (iw1,iw2,iw3)

1. Create interferograms for each subswath separately (without unwrapping);
2. Use GMTSAR's merge_batch.csh to mosaic phase files and then unwrap it.

At the same level as F1,F2,F3, run iGPS script

sh_s1_prep_f123

Will create a new directory (f123) and searching for interferograms to merge (intf.in.f123.1) . Enter f123 directory, and run

merge_batch.csh intf.in.f123.1 batch_tops.config

Sample intf.in.f123.1:

```
./F1/intf_all/2014298_2014346/:S1_20141026_ALL_F1.PRM:S1_20141213_ALL_F1.PRM,../F2/intf_all/2014298_2014346/:S1_20141026_ALL_F2.PRM:S1_20141213_ALL_F2.PRM,../F3/intf_all/2014298_2014346/:S1_20141026_ALL_F3.PRM:S1_20141213_ALL_F3.PRM
```

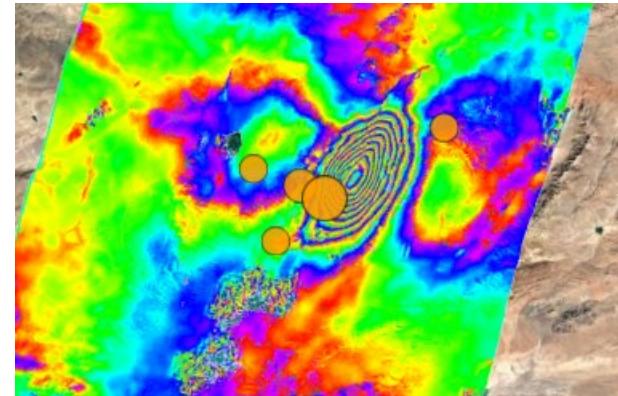
Note: merge_batch.csh will not move interferograms into intf_all directory.
Please move them manually.

**TEST RUN: 2020-07-22 RIGANPEI CO
EARTHQUAKE**

Example: Riganpei Co earthquake

- 2020-07-22 a normal earthquake on Riganpei Co fault in central Tibet

- Event page:
<https://earthquake.usgs.gov/earthquakes/eventpage/us6000b26j/executive>



Example: Riganpei Co earthquake

- Data
 - Descending 121
 - Downloading script:
iGPS\example\coseismi
c.insar.sentinel-
1.20220722.RiganpeiC
o\run_get_data

The screenshot shows the ASF Data Search interface. At the top, there are search filters for Date (Start: 7/11/2020, End: 7/31/2020), File Type (L1 Single Look Complex), Subtype (0/2 subtypes selected), and Path and Frame Filters (Path Start: 121, Path End: 121). Below the filters is a search bar with 'SEARCH' and a result count of '250 of 2 Files'. The main area displays a satellite map of the Riganpei Co region in China, with a red rectangular box indicating the search area. A yellow dot marks the center of the box, labeled 'Approximate Placement Only'. At the bottom, there is a 'Downloads' section showing '2 Files, 8.10 GB' with two entries:

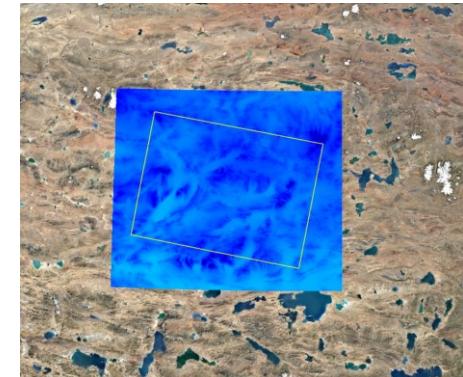
- S1A_IW_SLC_1SDV_20200726T000952_20200726T001019_0...17E0 • 4.05 GB
L1 Single Look Complex (SLC)
- S1A_IW_SLC_1SDV_20200714T000951_20200714T001018_0...72CA • 4.06 GB
L1 Single Look Complex (SLC)

Example: Riganpei Co earthquake

- Orbit
– POE
- From ESA (using public account gnssguest):
`sh_esa_s1_get_aux_orb_gnss -d 2020-07-26`
`sh_esa_s1_get_aux_orb_gnss -d 2020-07-14`

Example: Riganpei Co earthquake

- DEM
 - Get \$esa_data/topo/strm3/grid/china.grd (provided by iGPS)
 - iGPS will cut the file according to the KML range and put the dem.grd file in the root directory of processing
 - Or, obtain it from
<https://topex.ucsd.edu/gmtsar/demgen/>
Put dem.grd file in the root directory of processing



Example: Riganpei Co earthquake

Prepare Input: frames to be processed

- Extract meta data (*.safe)

```
[tianyf@bogon eq_riganpeico]$ sh_s1_unzip_manifest /home/tianyf/esa.data/s1/  
/home/tianyf/esa.data/s1  
[sh_S1_unzip_manifest]INFO: Usage: sh_s1_unzip_manifest [out_path [input_path] ]  
[sh_S1_unzip_manifest]INFO: Default:  
[sh_S1_unzip_manifest]INFO:     out_path: /home/tianyf/esa.data/safe/S1  
[sh_S1_unzip_manifest]INFO:     input_path: /home/tianyf/esa.data/s1  
[sh_S1_unzip_manifest]INFO: Currently:  
[sh_S1_unzip_manifest]INFO:     out_path: /home/tianyf/esa.data/s1  
[sh_S1_unzip_manifest]INFO:     input_path: /home/tianyf/esa.data/s1  
[sh_S1_unzip_manifest]INFO: extracting  
/home/tianyf/esa.data/s1/S1A_IW_SLC_1SDV_20200714T000951_20200714T001018_033443_03E012_72CA.  
.manifest.safe  
[sh_S1_unzip_manifest]INFO: extracting  
/home/tianyf/esa.data/s1/S1A_IW_SLC_1SDV_20200726T000952_20200726T001019_033618_03E572_17E0.  
.manifest.safe  
[sh_S1_unzip_manifest]INFO: # 2 manifest.safe files added.
```

Now we have

Name	Size
S1A_IW_SLC_1SDV_20200714T000951_20200714T001018_033443_03E012_72CA.manifest.safe	36 KB
S1A_IW_SLC_1SDV_20200714T000951_20200714T001018_033443_03E012_72CA.zip	4,152,349 KB
S1A_IW_SLC_1SDV_20200726T000952_20200726T001019_033618_03E572_17E0.manifest.safe	36 KB
S1A_IW_SLC_1SDV_20200726T000952_20200726T001019_033618_03E572_17E0.zip	2,301,178 KB

Example: Riganpei Co earthquake

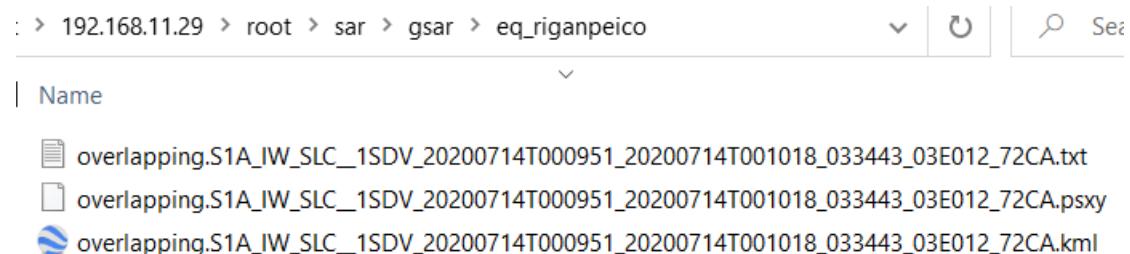
Prepare Input: frames to be processed

- Search for overlapping frames

In the root directory of processing, run

```
[tianyf@bogon eq_riganpeico]$ esa_s1_manifest_overlap /home/tianyf/esa.data/s1/  
S1A_IW_SLC_1SDV_20200714T000951_20200714T001018_033443_03E012_72CA `pwd`  
  
[]path:/home/tianyf/esa.data/s1/  
[]target:S1A_IW_SLC_1SDV_20200714T000951_20200714T001018_033443_03E012_72CA  
[]opath:/sar/gsar/eq_riganpeico  
[]perc_min: 0.80  
...  
perc for 2 is 1.0000000000000000  
[]out:/sar/gsar/eq_riganpeico/overlapping.S1A_IW_SLC_1SDV_20200714T000951_20200714T001018_033443_03E012_72CA.txt  
[]out  
KML:/sar/gsar/eq_riganpeico/overlapping.S1A_IW_SLC_1SDV_20200714T000951_20200714T001018_033443_03E012_72CA.kml  
[]out:/sar/gsar/eq_riganpeico/overlapping.S1A_IW_SLC_1SDV_20200714T000951_20200714T001018_033443_03E012_72CA.psxy
```

Now we have



Example: Riganpei Co earthquake

Unpack SLC zip files to temporary directory
(\$esa_unzip).

```
[tianyf@bogon eq_riganpeico]$ sh_s1_unzip
overlapping.S1A_IW_SLC__1SDV_20200714T000951_20200714T001018_033443_03E0
12_72CA.txt

...
[sh_s1_unzip_node]INFO: 1 / 2
S1A_IW_SLC__1SDV_20200714T000951_20200714T001018_033443_03E012_72CA.manifest.safe
[sh_s1_unzip_node]INFO: extracting
S1A_IW_SLC__1SDV_20200714T000951_20200714T001018_033443_03E012_72CA.zip
unzip -n
/home/tianyf/esa.data/s1/S1A_IW_SLC__1SDV_20200714T000951_20200714T001018_033443_03E012_72CA.zi
p -x *-vh-* -d /sar/esa_unzip
Archive:
/home/tianyf/esa.data/s1/S1A_IW_SLC__1SDV_20200714T000951_20200714T001018_033443_03E012_72CA.zi
p
  creating:
/sar/esa_unzip/S1A_IW_SLC__1SDV_20200714T000951_20200714T001018_033443_03E012_72CA.SAFE/
...
```

Example: Riganpei Co earthquake

Process the 3rd subswath (F3)

In the root directory of processing, create a F3 directory.

In F3, run

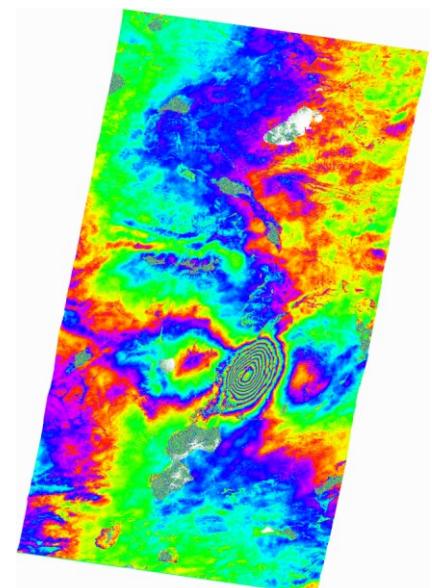
```
sh_s1_run_tsa -file  
./overlapping.S1A_IW_SLC__1SDV_20200714T000951_20200714T00101  
8_033443_03E012_72CA.txt -iw iw3 -proc_type interseismic
```

If no errors occur, we will have a bunch of files in the interferogram directory (in intf_all).

Make sure you have enough disk space and memory (>2GB?).

Name	Date modified
amp2.grd	2/19/2023 8:04 AM
corr.grd	2/19/2023 8:05 AM
corr_ll.grd	2/19/2023 8:11 AM
display_amp.grd	2/19/2023 8:04 AM
display_amp_ll.grd	2/19/2023 8:15 AM
filtcorr.grd	2/19/2023 8:08 AM
imafilt.grd	2/19/2023 8:04 AM
mask.grd	2/19/2023 8:08 AM
mask2.grd	2/19/2023 8:08 AM
phase.grd	2/19/2023 8:05 AM
phase_mask.grd	2/19/2023 8:08 AM
phase_ll.grd	2/19/2023 8:13 AM
phasefilt.grd	2/19/2023 8:08 AM
phasefilt_ll.grd	2/19/2023 8:12 AM
phasefilt_mask.grd	2/19/2023 8:08 AM
phasefilt_ll.mask	2/19/2023 8:16 AM

Check the image files for results preview.



Example: Riganpei Co earthquake

Repeat the processing for iw1 (F1) and iw2 (F2) subswaths.

In F2:

```
sh_s1_run_tsa -file  
./overlapping.S1A_IW_SLC__1SDV_20200714T000951_20200714T00101  
8_033443_03E012_72CA.txt -iw iw2 -proc_type interseismic
```

In F1:

```
sh_s1_run_tsa -file  
./overlapping.S1A_IW_SLC__1SDV_20200714T000951_20200714T00101  
8_033443_03E012_72CA.txt -iw iw1 -proc_type interseismic
```

Mosaic iw1+iw2+iw3

After processed iw1,iw2 and iw3, run

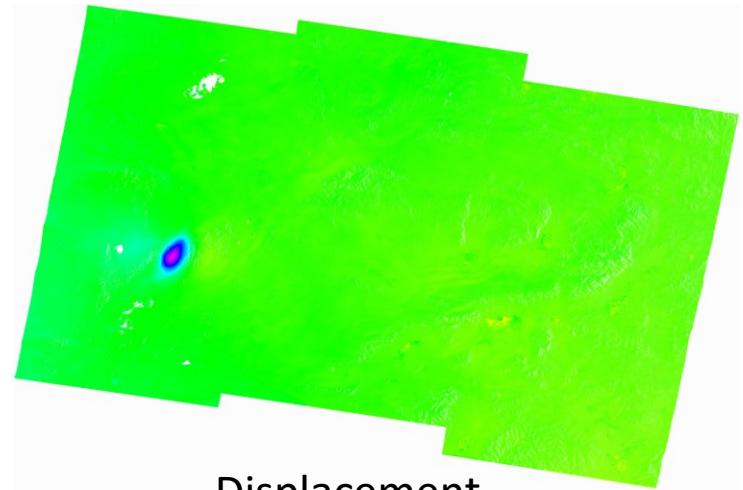
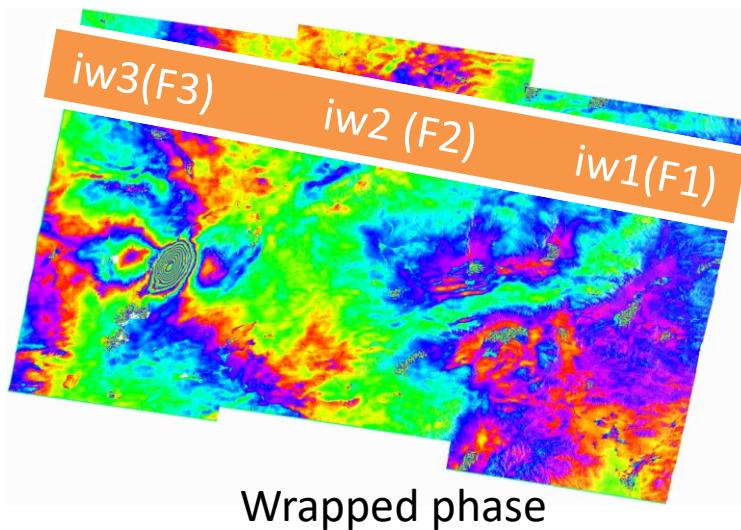
`sh_s1_prep_f123`

to prepare files for merging.

Enter f123 directory, run

`merge_batch.csh intf.in.f123.1 batch_tops.config`

to do the merging (and unwrapping) job.



LEVEL-1 PROCESSING

Interrupt merge_batch.csh, and continue ...

If some interferograms have already been created, i.e., the intf.in.f123.1 file exists, we need to get the remaining ones to process.

Searching for unfinished interferograms:

`sh_s1_prep_f123_in`

Will create log2 file and extract the unfinished interferograms list to intf.in.f123.2

Use merge_batch2.csh (revised from merge_batch.csh) to complete the task:

`merge_batch2.csh intf.in.f123.1 intf.in.f123.2
batch_tops.config`

merge_batch2.csh

Please note: merge_batch.csh requires that “The master image of first line should be the super_master.”

If you interrupted the processing and continue later, make sure that the master image of the first line remains the same. Otherwise, the size (number of rows/columns) of interferogram may differ.

iGPS revised the GMTSAR merge_batch.csh script to extract master information from the first interferogram list file (intf.in.f123.1), e.g.

merge_batch2.csh intf.in.f123.1 intf.in.f123.2 batch_tops.config

Parallel Version of Merging Scripts

- merge_batch_parallel.csh intf.in.f123.1 batch_tops.config N
- merge_batch2_parallel.csh intf.in.f123.1 intf.in.f123.2 batch_tops.config N

Where N is the number of parallel jobs.

Note: make sure the back geocoding file (trans.dat) is generated in f123 directory, which means running merge_batch.csh for at least one interferogram before running parallelizing jobs.

When the dimensions of corr and unwrap files are not consistent, one can use gmt grdcut command to clip out the common area (or the area of interest given by polygon KML file) using the iGPS tsa script sh_sar_intf_all_cut_roi

Clip and Mosaic for Single Subswath(iw1/iw2/iw3)

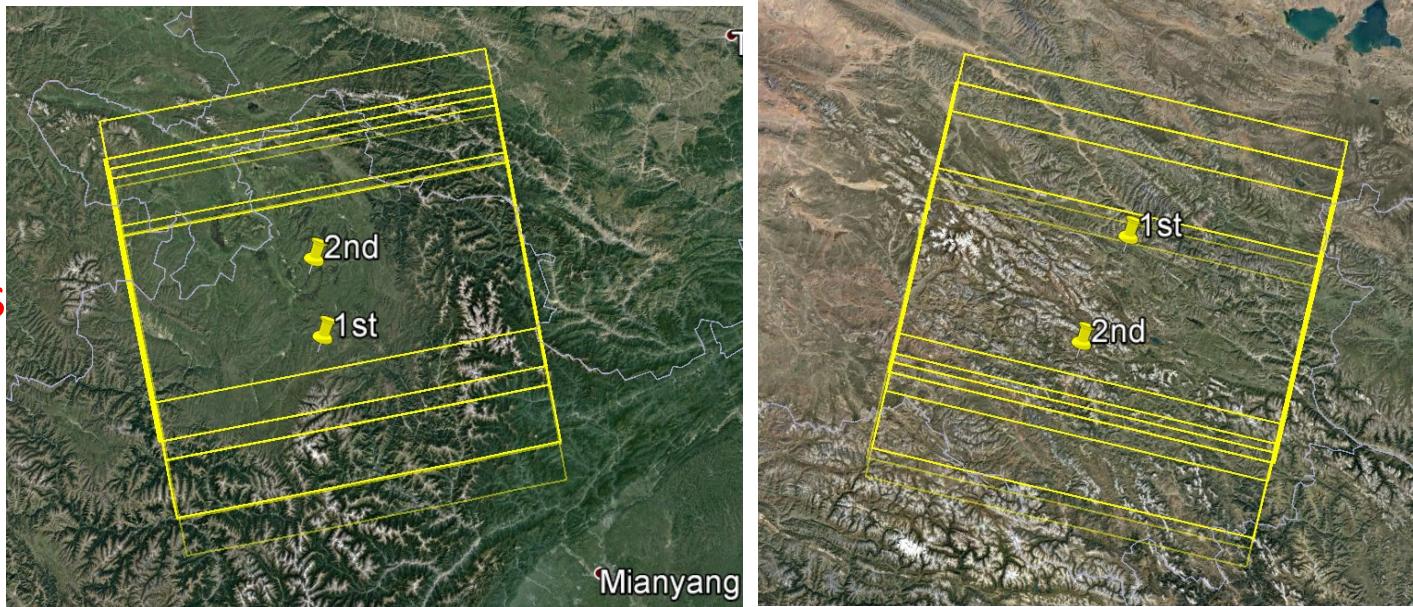
- By burst
 - For single frame
 - Merge two or more consecutive frames and then clip
- By row/column
 - Perform clipping on coregistered SLC images
 - Perform clipping on interferograms
 - Resample interferograms

Clip by Burst: from Single Frame

Prepare a KML file containing two points, with the **first** point being acquired **earlier**.

- Ascending data: 1st point in the south; 2nd point in the north
- Descending data: 1st point in the north; 2nd point in the south

The two points
should spans
at least **2-3 bursts**

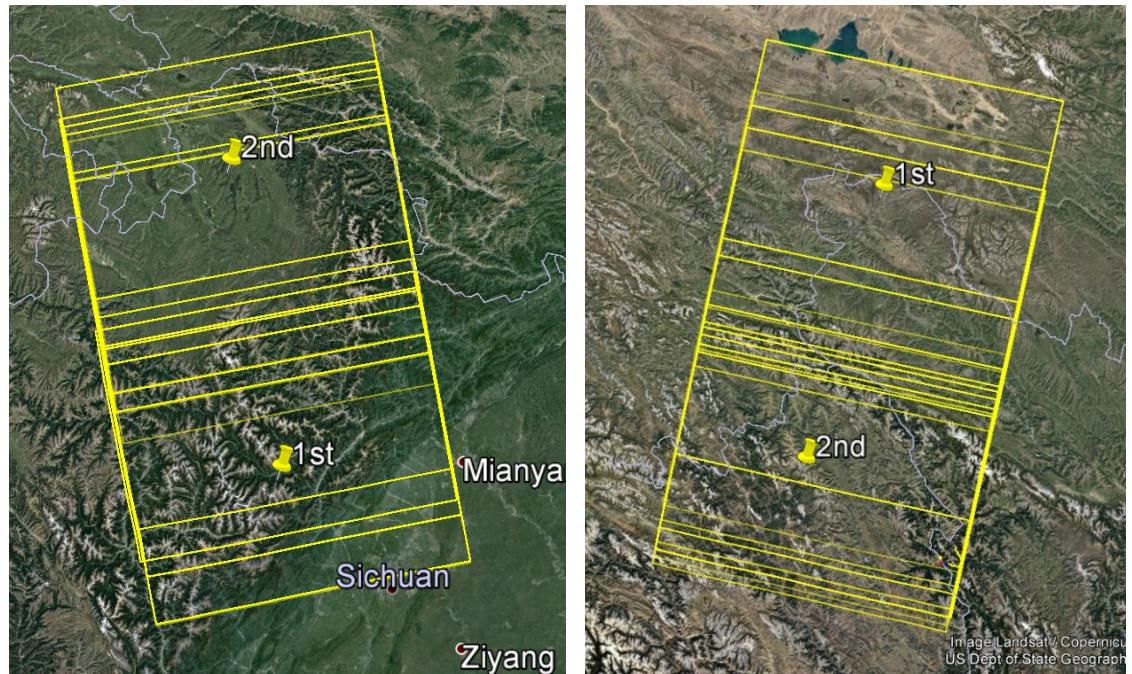


When calling sh_s1_run_tsa script, adding -roi option:
`sh_s1_run_tsa -file input.lst -roi pins.kml`

Clip by Burst: from a Long Swath (Multiple Frames)

The same as the single frame case.

The two points usually can span at most **17-20 bursts**, if first assemble all tiff files together (classic tiff file cannot exceed 4GB size).



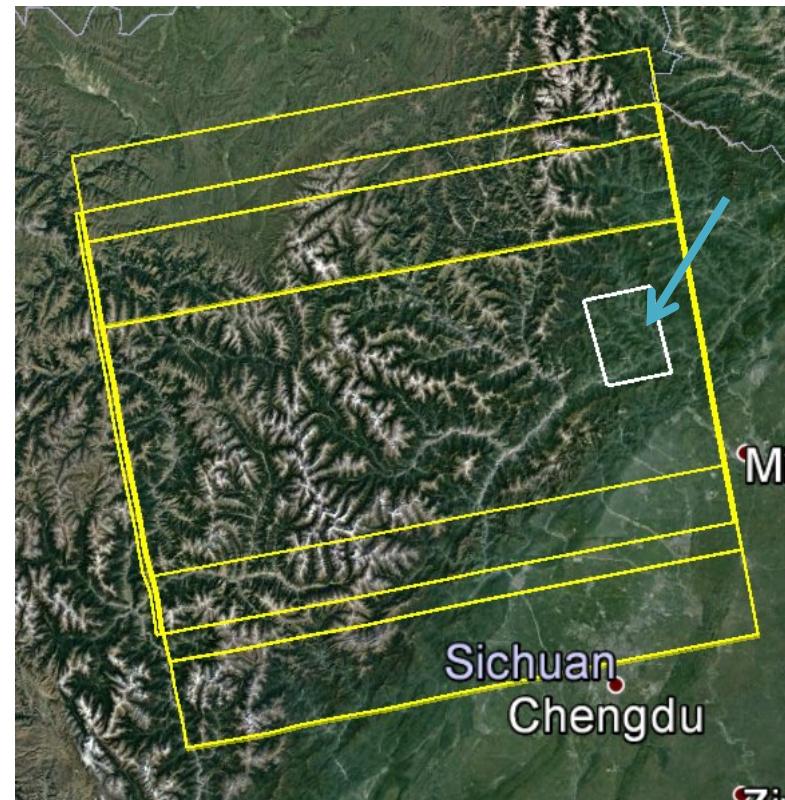
When calling `sh_s1_run_tsa` script, adding -roi option:
`sh_s1_run_tsa -file input.lst -roi pins.kml`

Clip by Row/Column: on Aligned SLC Files

sh_slc_cut script in iGPS

- Input:
 - ROI KML file
- Example:
In F1 (or F2/F3), run
sh_slc_cut -file roi.kml

iGPS's sh_s1_init_F_cut script can initiate the environment and cut the SLC file. E.g., in the root directory of processing (parent of F1/F2/F3/), run



`sh_s1_init_F_cut -file my_roi.kml -iw iw2`

Will create a F2.my_roi directory, copy necessary files to it, cut SLC in F2/raw and save new SLC files to F2/raw.my_roi/, and create a link to it in F2.my_roi

Clip by Row/Column: on Interferograms

- Purpose:
Cut corr_cut.grd and unwrap_mask.grd files in intf_all directory.
- Tool:
 - iGPS's sh_sar_intf_all_cut_roi script
- Example:
`sh_sar_intf_all_cut_roi -file roi.kml -opath intf_all_roi`
It will convert geographical range in roi.kml into radar coordinate system, and use GMT's grdcut command to clip out the data of ROI.

Resample the Interferograms: Downgrade the Resolution

- Purpose:
Resample corr_cut.grd and unwrap_mask.grd files in intf_all directory to a lower resolution.
- Tool:
 - iGPS's sh_sar_intf_all_resample script
- Example:
`sh_sar_intf_all_resample -r 2`
Here, 2 is the downgrade factor (means from 25m to 50m, or from 50m to 100m, etc.). New interferograms will be saved in intf_all_x2

Snaphu: Coherence Threshold in Unwrapping

In batch_tops.config, the threshold can be set to a much small one, e.g.,

```
threshold_snaphu = 0.01
```

After obtain the deformation results, create a mask according to the coherence images of a few interferograms, with a reasonable threshold (e.g., 0.05). In our experiences, such a value can mask out nearly all decorrelated Earth surface (i.e., water body, desert, snow/ice, and forest).

iGPS's sh_sar_sbas_corr_mask creates a mask and performs masking on the vel_ll.grd file. The mask is also used to clip displacement files.

Example:

```
sh_sar_sbas_corr_mask -corr_min .05
```

Snaphu: mst vs. mcf

- Default:
`snaphu -mst`
- To change to MCF:
`snaphu --mcf`
- They work with no big differences when the coherence is good.
- When the decorrelation is heavy,
 - it takes long time for mst. Use mcf instead.
 - Mst likely produces more unwrapping errors (?).
- Using mcf consumes much more physical memory.

Interrupt `intf_tops.csh` and Continue Processing

The step of creating interferograms (`intf_tops.csh`) is time-consuming. Sometimes one may want to rearrange the order of processing, or pause a while for other work. After interrupt the processing,

1. Under the subswath directory (F1/F2/F3), use **`sh_sar_cp_intf_png`** to copy preview image (*.png) from `intf_all` to `intf_all_png` for interferograms processed.
2. Use **`sh_s1_intf_in_unfinished`** to get list of interferograms not processed
 1. Input: `intf.in`, `intf_all_png/`, `baseline_table.dat`, `scene.tab1`
 2. Output: `intf.in.rem`
3. Run **`intf_tops.csh`** or `intf_tops_parallel.csh` to continue the processing, e.g.,
`intf_tops.csh intf.in.rem batch_tops.config`
`intf_tops_parallel.csh intf.in.rem batch_tops.config 8`
Make sure the `proc_stage` is set to 2 in the `batch_tops.config`! Otherwise (if `proc_stage = 1`), the `topo/trans.dat` file will be created each time running `intf_tops.csh`.

Get New Data? Update the Processing

Question: How to do when new S1 SLC data are added?

The answer: `sh_s1_run_tsa_update`

No need to process the old data again. It just aligns the new data frames to the reference one.

The outputs are:

- New aligned SLC files
 - raw/S1_20*
- data.in (updated)
- baseline_table.dat (updated)
- intf.in (updated)

Example:

`sh_s1_run_tsa_update -file new_data_file -iw iw1`

Note: new data list file should contain the reference one!

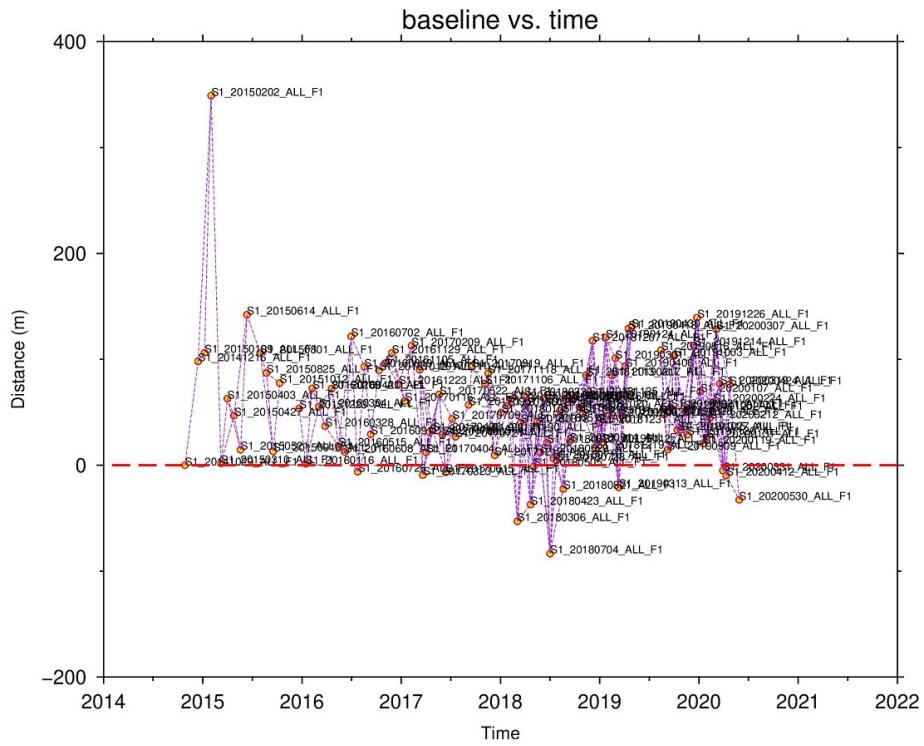
SBAS

Steps:

- 1) First, find which interferograms had been processed (saved in intf_all).
iGPS use the PNG image files to do this job. Using sh_sar_cp_intf_png, the varied preview files (phase, unwrapped phase, displacement-los) are copied from intf_all to intf_all_png directory. Run it under subswath directory (F1/F2/F3).
- 2) Create all possible interferograms list (from intf.in.) for input to SBAS.
intf.tab1 & scenes.tab1, using **sh_sar_prep_sbas**
- 3) Prepare intf.tab & scene.tab files for SBAS, according to PNG images in intf_all_png directory.
sar_sbas_tab_from_png
- 4) Use **sh_sar_call_sbas** to do SBAS analysis, creating displacements and velocities.

Plots of Baselines

- `sh_sar_plot_intf_tab`
Plot interferograms for SBAS processing
Input: intf.tab and scene.tab
 - `sh_sar_plot_intf_in`
Plot interferograms for intf_tops.csh processing
Input: intf.in



Outputs: f-baseline-time* files

sh_sar_call_sbas

sbas.A.B.CCCC.DDDD.EEEEEEEE.FFFFFFFF.GGG.HHHH.II.ZZZ

SBAS Output Path Naming Convention

A	-	resample level
	1	- full resolution (4:1; 25m)
	2	- 50m (8:2)
	3	- 100m (16:4)
	4	- 200m (32:8)
B	-	atm code
	0	- no aps correction
	1	- with -atm 1
	2	- with -atm 2
	3	- with -atm 3
	4	- with -atm 4
	5	- with -atm 5
	8	- GACOS only
	9	- with GACOS
CCCC	-	minimum time span (number of days)
DDDD	-	maximum time span
EEEEEEEEE	-	starting date
FFFFFFFFF	-	starting date
GGG	-	number of scenes
HHHH	-	number of interferograms
II	-	version number
ZZZ	-	comment

e.g.,

sbas.4.0.0001.9999.20180110.20230120.008.0008.01._____

sh_sar_call_sbas

sh_sar_call_sbas options:

- -r
Code for interferogram pixel size
- -i
Interferograms path (default intf_all)
- -a
Atmospheric correction code
- -t1
Minimum days of temporal baseline (default 0)
- -t2
Maximum days of temporal baseline (default all)
- -c
e.g.,
comments sh_sar_call_sbas -r 4 -i intf_all_x2 -p y -t1 121 -t2 500
- -OW
overwrite output? y/n
- -p
y: run sbas_parallel instead of sbas

Extract Displacement Time series

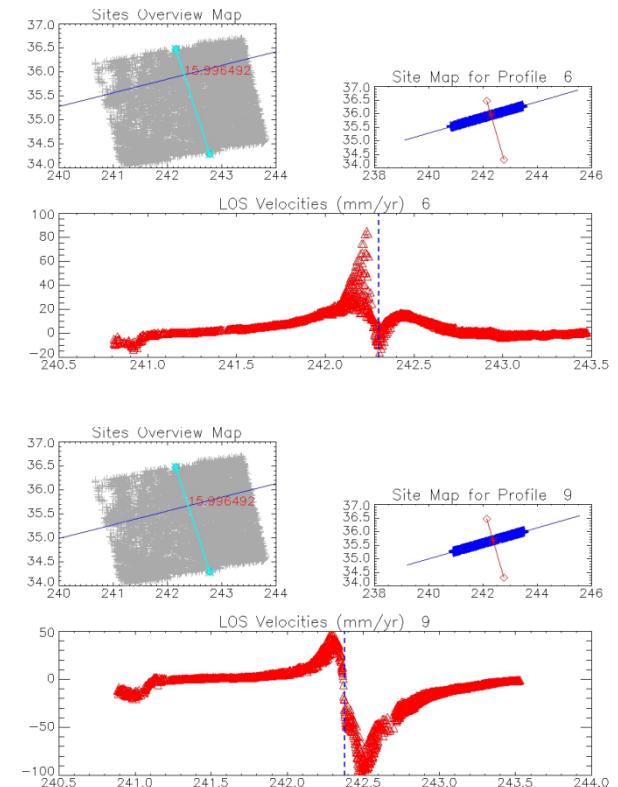
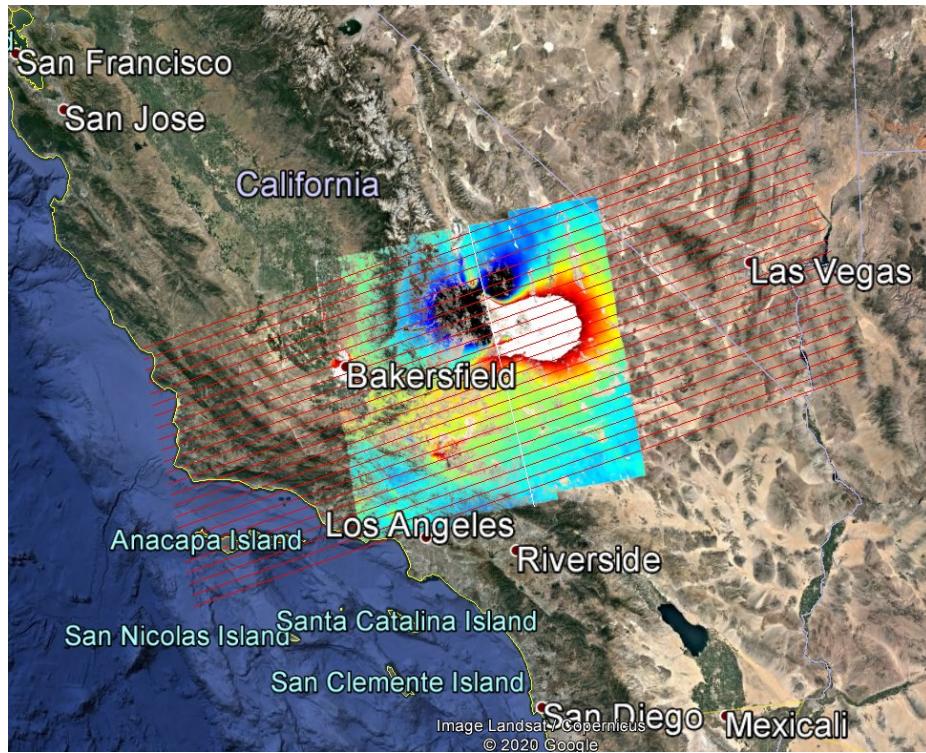
Extract displacement time series from SBAS result
for individual locations (points).

- `sh_sar_sbas_extract_time_series_lls`

Create velocity or displacement profiles

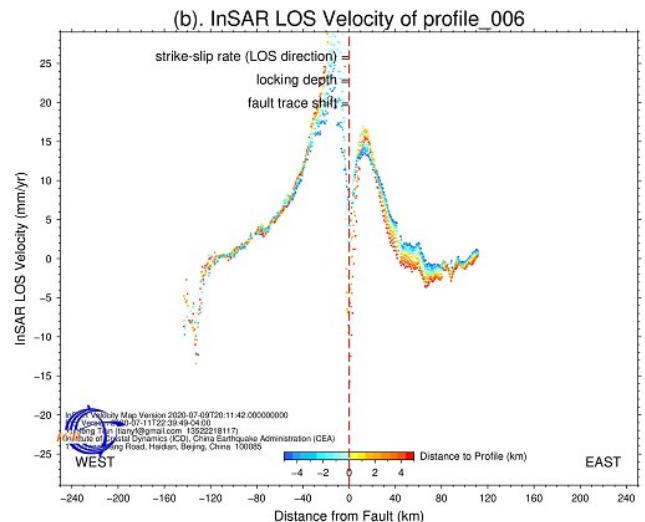
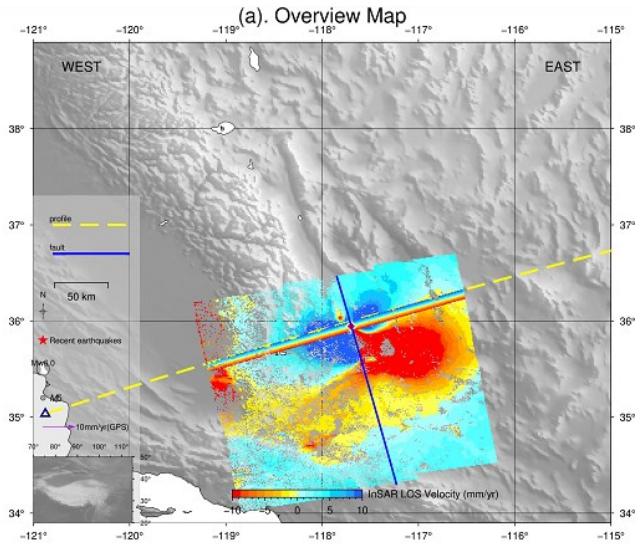
Use iGPS IDL program
SAR_LOS_PROFILES_AUTO_LLV

Create profiles perpendicular to fault line



Generate Profile Plot

- sh_sar_plot_vel_profile

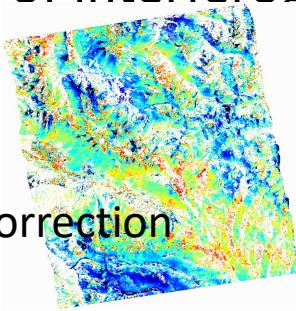


Common-Point Stacking (statistics method)
GACOS (from meteorological observations)

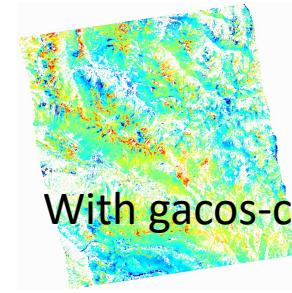
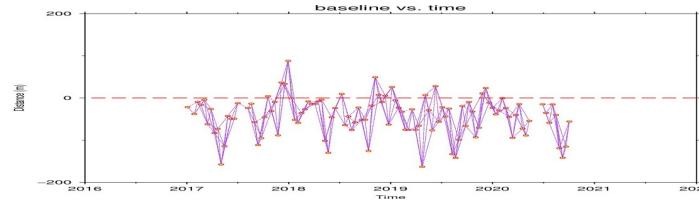
ATMOSPHERIC DELAY CORRECTION

Atmospheric Delay Correction: Is it necessary?

- When the topography variation is large, atmospheric delay correction is important for short temporal baselines or a small amount of interferograms.

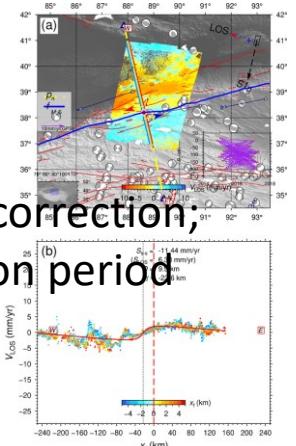


Without gacos-correction

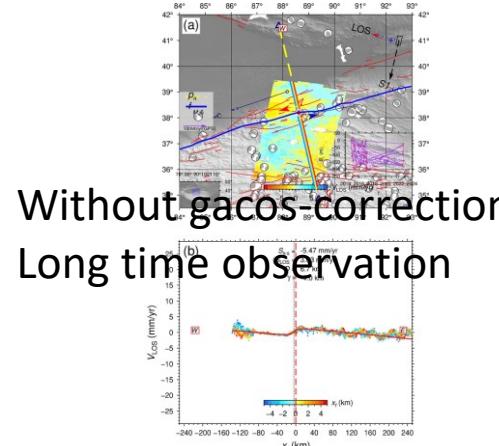


With gacos-correction

- Atmospheric delay correction, sometimes, can introduce improvement for long temporal baselines and sufficient amount of interferograms.



Without gacos-correction;
short observation period



Without gacos-correction
Long time observation

GACOS Correction

Steps:

1. Download GACOS grid data
2. Uncompress GACOS files
3. Convert binary file into NetCDF format
4. Project data from geographical coordinates to radar coordinates
5. Create GACOS corrections for interferograms
6. Apply GACOS corrections to interferograms

Step 1. Download GACOS grid data

- <http://www.gacos.net/>
- Output format: Binary grid
- Save the *.tgz files to **./tgz** directory

The screenshot shows the GACOS download interface. At the top, there are four input fields for coordinates: 33, 100, 104, and 30. Below these are dropdowns for 'Time of Interest (in UTC)*' set to 11 and 8, and a 'Date list*' dropdown containing 20200316 and 20200328. On the right, there's an 'Output format*' section with 'Binary grid' selected, and an 'Email*' field containing y.f.tian@163.com. A large blue 'Submit' button is at the bottom.

33
100
104
30

Time of Interest (in UTC)* ?
11 : 8

Date list* ?
20200316
20200328

Output format* ?
 Geotiff Binary grid

Email*
y.f.tian@163.com

Submit

The screenshot shows a file explorer window with a list of files in a 'tgz' folder. The files are all WinRAR archive files with various names and sizes, mostly around 2.28 GB. The last file listed is 'url.txt'. The file names include: 20201226T221316Vbm8l8GIR.tar.gz, 20201226T221346Vao3CUk6o.tar.gz, 20201226T231546Vk5YEQoZx.tar.gz, 20201226T231617VxllnbrNO.tar.gz, 20201226T231647V2RyYt36m.tar.gz, 20201226T231717VRgmxV0sB.tar.gz, 20201226T231747VsONV2hVK.tar.gz, and 20201227T093852VmE3SWpUd.tar.gz.

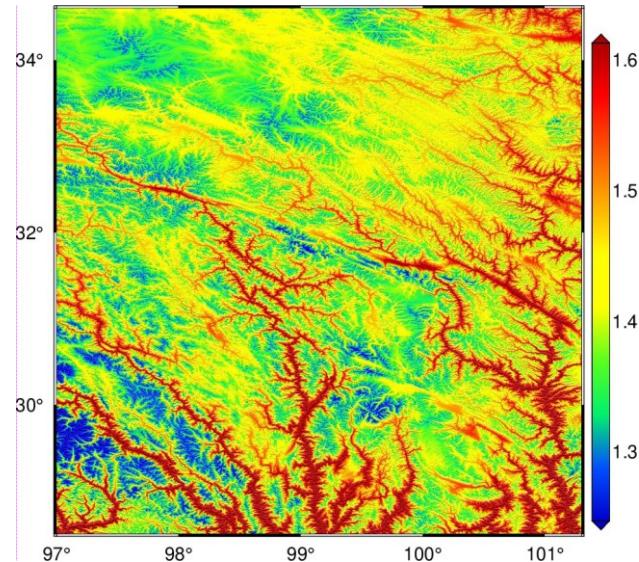
名称	大小	类型	修改时间	属性
上级文件夹				
20201226T221316Vbm8l8GIR.tar.gz	2.29 GB	WinRAR archive	12/27/2020 5:14:03 AM	-----
20201226T221346Vao3CUk6o.tar.gz	584 MB	WinRAR archive	12/26/2020 11:38:37 PM	-----
20201226T231546Vk5YEQoZx.tar.gz	2.28 GB	WinRAR archive	12/27/2020 2:17:34 PM	-----
20201226T231617VxllnbrNO.tar.gz	2.28 GB	WinRAR archive	12/28/2020 3:16:26 AM	-----
20201226T231647V2RyYt36m.tar.gz	2.28 GB	WinRAR archive	12/28/2020 7:48:04 AM	-----
20201226T231717VRgmxV0sB.tar.gz	2.28 GB	WinRAR archive	12/28/2020 8:20:39 AM	-----
20201226T231747VsONV2hVK.tar.gz	2.28 GB	WinRAR archive	12/28/2020 8:55:38 AM	-----
20201227T093852VmE3SWpUd.tar.gz	2.28 GB	WinRAR archive	12/28/2020 11:16:16 AM	-----
url.txt	560 字节	Text Document	12/27/2020 2:25:06 PM	-a-----

Step 2. Uncompress GACOS files

- sh_sar_gacos_tgz_unzip

Will create **ztd** directory and extract files into it.

名称	大小	类型	修改时间	属性
上级文件夹				
20141012.ztd	144 MB	ZTD File	12/27/2020	9:41:22 AM
20141012.ztd.rsc	380 字节	RSC File	12/27/2020	9:41:27 AM
20141012.ztd_preview.jpg	462 KB	JPEG image	12/27/2020	9:42:05 AM
20141105.ztd	144 MB	ZTD File	12/27/2020	9:43:46 AM
20141105.ztd.rsc	380 字节	RSC File	12/27/2020	9:43:51 AM
20141105.ztd_preview.jpg	462 KB	JPEG image	12/27/2020	9:44:29 AM
20141129.ztd	144 MB	ZTD File	12/27/2020	9:46:08 AM
20141129.ztd.rsc	380 字节	RSC File	12/27/2020	9:46:13 AM
20141129.ztd_preview.jpg	461 KB	JPEG image	12/27/2020	9:46:51 AM



20141012.ztd_preview.jpg

Step 3. Convert binary file into NetCDF format

- sh_sar_gacos_ztd2ll

Convert files in **ztd** directory to NetCDF (*.grd) formats and save the output in **ztd.ll** directory.

名称	大小	类型	修改时间	属性
上级文件夹				
20141012_ll.grd	84.1 MB	GRD File	1/7/2021	9:14:46 PM
20141012_ll.jpg	395 KB	JPEG image	1/7/2021	9:14:56 PM
20141105_ll.grd	83.9 MB	GRD File	1/7/2021	9:15:02 PM
20141105_ll.jpg	396 KB	JPEG image	1/7/2021	9:15:11 PM
20141129_ll.grd	83.8 MB	GRD File	1/7/2021	9:15:18 PM
20141129_ll.jpg	389 KB	JPEG image	1/7/2021	9:15:28 PM
20150116_ll.grd	83.8 MB	GRD File	1/7/2021	9:15:34 PM

Step 4. Project data from geographical coordinates to radar coordinates

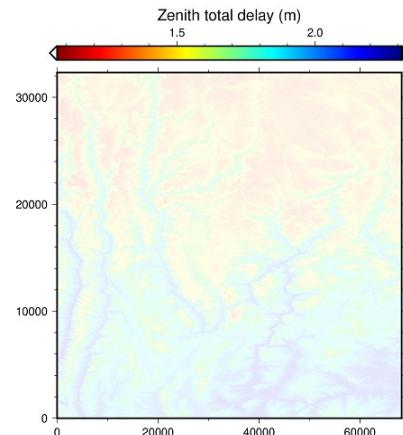
- sh_sar_gacos_ll2ra

Convert data from geographical coordinates into radar coordinates, saving results in **ztd.ra** directory.

Prerequisite:

- trans.dat (link from topo/trans.dat, e.g.,)
`ln -s /some_where/F1/topo/trans.dat`
- intf_all (link from GMTSAR processing directory)

名称	大小	类型	修改时间	属性
上级文件夹				
20141012_ra.grd	77.3 MB	GRD File	1/9/2021 10:45:22 PM	-----
20141012_ra.jpg	372 KB	JPEG image	1/9/2021 10:45:31 PM	-----
20141105_ra.grd	77 MB	GRD File	1/9/2021 11:07:24 PM	-----
20141105_ra.jpg	378 KB	JPEG image	1/9/2021 11:07:34 PM	-----
20141129_ra.grd	77 MB	GRD File	1/9/2021 11:30:22 PM	-----
20141129_ra.jpg	379 KB	JPEG image	1/9/2021 11:30:32 PM	-----
20150116_ra.grd	76.9 MB	GRD File	1/9/2021 11:54:06 PM	-----
20150116_ra.jpg	379 KB	JPEG image	1/9/2021 11:54:28 PM	-----



Step 5. Create GACOS corrections for interferograms

- sh_sar_gacos_intf_all

Results saved in **gacos_all**

iGPS also detrends the GACOS data (using grdtrend command of GMT) because the trend is also removed in interferograms.

The screenshot shows a Windows-style file explorer window. The left pane displays a tree view of directories under '099-a-m3-0081CA06_0087F6ED_0092696A-lijiang_deqin.merged'. The 'gacos_all' folder is expanded, showing sub-directories for years from 2014284 to 2016226. The '2014284_2014308' directory is selected and highlighted in blue. The right pane shows a detailed list of files within this selected directory. The columns are labeled '名称' (Name), '大小' (Size), '类型' (Type), '修改时间' (Last modified), and '属性' (Properties). The files listed are: gmt.history (145字节 HISTORY File, 1/12/2021 11:38:52 AM), phase.cpt (1.17 MB CPT File, 1/12/2021 11:38:41 AM), ztd_gacos.grd (80.3 MB GRD File, 1/12/2021 11:38:10 AM), ztd_gacos.jpg (336 KB JPEG image, 1/12/2021 11:38:19 AM), ztd_gacos_detrend.grd (90.7 MB GRD File, 1/12/2021 11:38:41 AM), and ztd_gacos_detrend.jpg (841 KB JPEG image, 1/12/2021 11:38:52 AM).

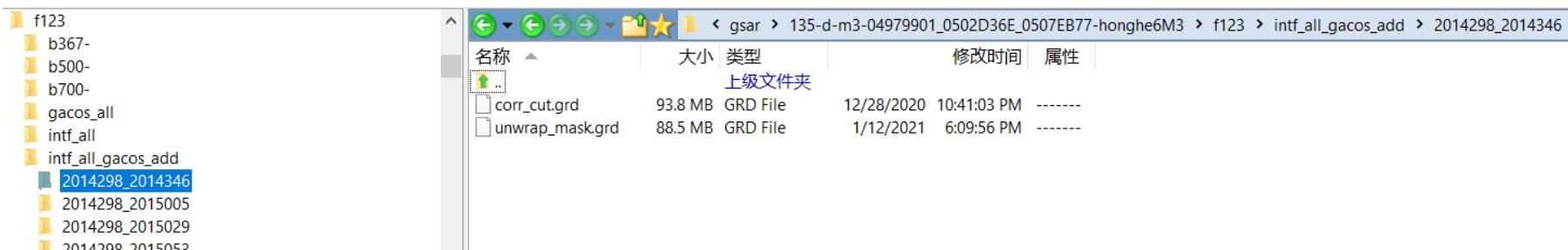
名称	大小	类型	修改时间	属性
上级文件夹				
gmt.history	145 字节	HISTORY File	1/12/2021 11:38:52 AM	-----
phase.cpt	1.17 MB	CPT File	1/12/2021 11:38:41 AM	-----
ztd_gacos.grd	80.3 MB	GRD File	1/12/2021 11:38:10 AM	-----
ztd_gacos.jpg	336 KB	JPEG image	1/12/2021 11:38:19 AM	-----
ztd_gacos_detrend.grd	90.7 MB	GRD File	1/12/2021 11:38:41 AM	-----
ztd_gacos_detrend.jpg	841 KB	JPEG image	1/12/2021 11:38:52 AM	-----

Step 6. Apply GACOS corrections to interferograms

- sh_sar_gacos_apply_intf_add

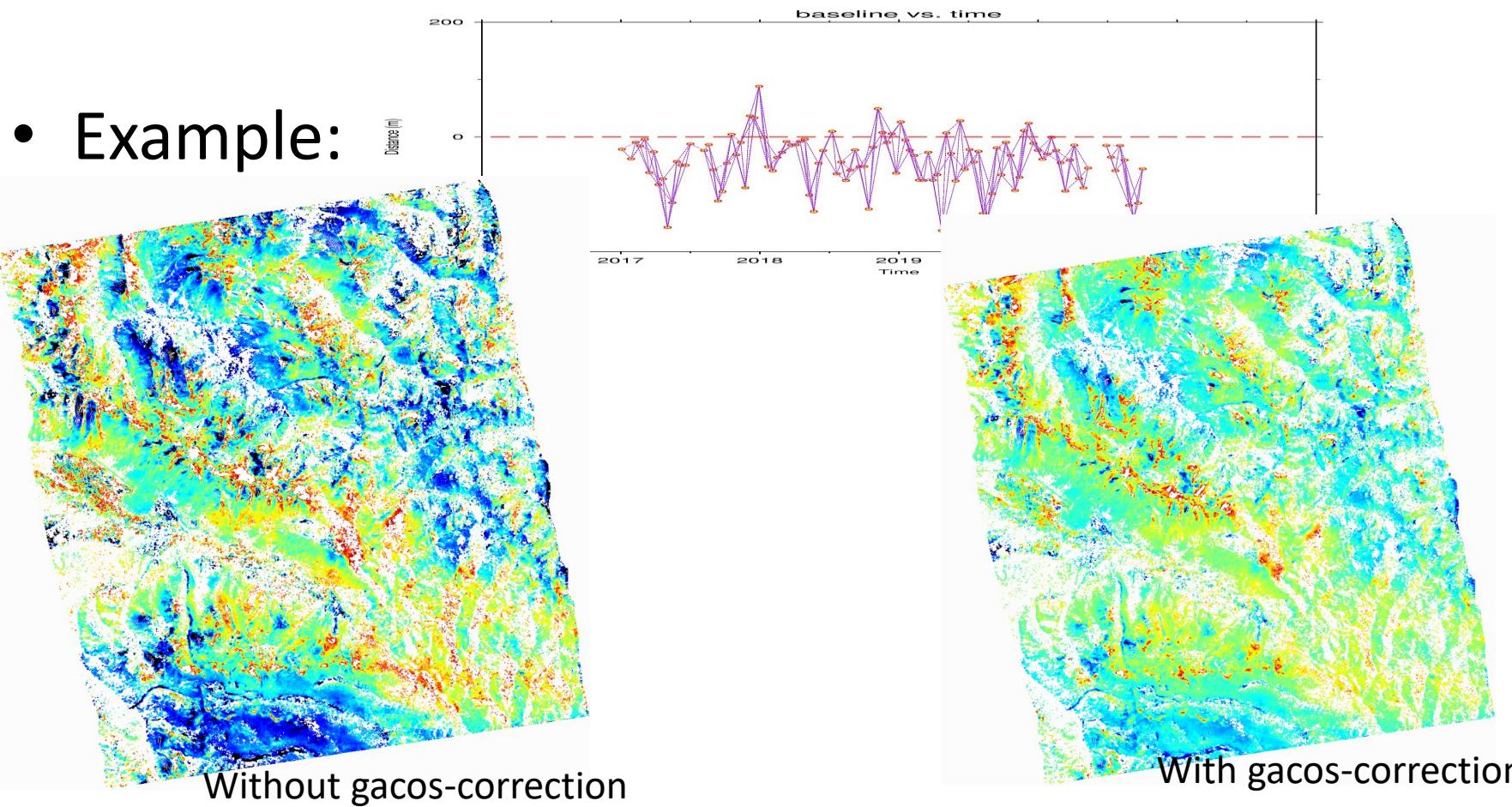
Results are saved in **intf_all_gacos_add**

It seems that the “addition” of GACOS correction to GMTSAR’s interferogram is the correct operation.



Generate velocity map and displacement time series using corrected interferograms

- `sh_sar_call_sbas -ipath intf_all_gacos_add`



Thank You

- Please report bugs to
tianyf@gmail.com

Any questions/comments/suggestions/... will be appreciated!