**Interferometric synthetic-aperture radar time series program**

**(INSARTS, version 20201228)**

**Chen Yu (**[**chen.yu@newcastle.ac.uk**](mailto:chen.yu@newcastle.ac.uk)**)**

**0 Before running**

The program is written in C++ enabling multiprocessing through OPENMP. The time series inversion is calculated by a general Gauss-Markov linear model (e.g., weighted least square) and then filtered in the frequency domain by FFTW. The estimation is run on a pixel-by-pixel basis and each pixel can be estimated in parallel, therefore it is expected to run fast when OPENMP is enabled.

Make sure you have the following libraries installed:

* FFTW (<http://www.fftw.org/>)
* OPENMP (Optional) (<https://www.openmp.org/>)

**1 Tips to run**

1.1 To run the program, simply type:

./insarts cfg

**The unit of the output is the same as the input.**

1.2 The interferograms should be stored in a ‘**little-endian**’ **float** (4 byte) binary file with the row major order. The filename of interferograms should contain at least two dates (first and second date) and in the format of ‘\*YYYYMMDD\*YYYYMMDD\*’ (separate the two dates with any character(s)). **All the input interferograms should be in the same size.**

1.3 There are several **swithers** in the configuration file (highlighted below in green) with which you can turn on/off the correspondent section.

1.4 In order to find satisfied filter strength (which is sometimes tricky), you can turn off if\_remove\_orbit, if\_remove\_eds, if\_check\_loop\_closure and if\_inverse\_time\_series, and then run the program several times with different filtering parameters (the program will load the original time series output of the time series inversion and overwrite any existing filtered results).

**2 The outputs**

2.1 Cumulative displacement maps in the same size as the input interferograms. It is named as ‘culmap-date1-date2.disp’ which is the cumulative displacement from date2 to date1.

2.2 ‘linear\_mean\_velocity’ which is the mean **annual** displacement (cumulative displacement divided by the time interval).

2.3 ‘least\_square\_sigma’ which is the mean square root of observation residuals.

2.4 ‘linear\_stacking\_mean\_velocity’, ‘linear\_stacking\_mean\_velocity\_residual’ and ‘linear\_stacking\_mean\_velocity\_sigma’ when ‘if\_stackinbg’ is turned on. These files correspond to the equations in Section 4.3.

2.5 Filtered cumulative displacement maps named as ‘culmap-date1-date2.disp.filtered’.

2.6 A phase loop closure mask when ‘if\_check\_loop\_closure’ is turned on. It is in the same size as the input interferograms. (0.0 -> no mask, 1.0-> masked). A phase\_closure\_std map and a phase\_closure\_failed\_ratio map.

2.7 filelist\_used, filelist\_deleted and filelist\_statistics, if some ifgs were deleted during the processing.

2.8 baseline\_dependent\_dem\_error map when ‘if\_est\_hgt’ is turned on.

**3 Parameter definitions and configuration**

|  |  |  |
| --- | --- | --- |
| Parameter name | Data type | Explanation |
| **Part 1: filenames and foldernames** | | |
| resultfolder | String | * **MUST BE CREATED BEFORE RUNNING.** * All results and intermediate files will be stored in this folder. |
| headerfile | String | * A rsc style header file, must contain "WIDTH,FILE\_LENGTH,X\_FIRST,Y\_FIRST,X\_STEP,Y\_STEP", an example is given in Appendix 1. |
| filelist | String | * A filelist containing all the filenames of ifgs that are to be processed. * **NO PATH**. only filenames are allowed. |
| datafolder | String | * foldername where ifgs are stored |
| baselines | String | * A baseline text file defines the baseline of each ifg. * Two column text file, first column: filename, second column baseline in meters. * An example is given in Appendix 2. |
| **Part 2: System related parameters** | | |
| maxcap | Int | * Maximum capacity of memory defined as pixel size, over which the ifg will be divided. * Default=5000000 (roughly 20Mb per ifg). |
| max\_parallel\_threads | Int | * Maximum allowed paralleled threads. * Adding more threads will accelerate the program but will cost significantly more computational resources of your system. * Default=4. |
| **Part 3: Define a reference point/area** | | |
| ref\_method | Int | * The method of choosing a reference point: * 0: To automatically choose an arbitrary point which is valid on all ifgs. * 1: To use the mean phase of each ifg as its reference. * 2: To search for a stable area (defined by ref\_window) which has the minimum variance of all times. The variance is calculated for each ifg within the defined ref\_window and the mean of all ifgs are used. Default=2. |
| ref\_lat | Float | * Latitude of the reference point in degree (-90 to 90). Put a random large value (e.g., 9999) to disable it. |
| ref\_lon | Float | * Longitude of the reference point in degree (-180 to 180). Put a random large value (e.g., 9999) to disable it. |
| ref\_row | Int | * Row number of the reference point (zeros to the length of the ifg). Put a random large value (e.g., 9999) to disable it. |
| ref\_col | Int | * Column number of the reference point (zeros to the width of the ifg). Put a random large value (e.g., 9999) to disable it. * Only if (ref\_lat, ref\_lon) is not defined, the program will use (ref\_row, ref\_col). * If all (ref\_lat, ref\_lon, ref\_row, ref\_col) are not defined, the program will search for a reference point using the method defined by ref\_method. |
| ref\_window | Int | * Window size to extract reference point value. * The variance in ref\_method is calculated within this window. * Default=5. |
| **Part 4: Define interferogram quality control parameters** | | |
| if\_remove\_orbit | Int | * 0: Switch off. * 1: Remove orbital ramp and write out new files. |
| if\_remove\_eds | Int | * 0: Switch off. * 1: Remove elevation dependent signals (eds) by (phase=a\*H+b). |
| if\_use\_existing\_corrected\_ifg | Int | * 0: Do not use existing files. * 1: Use existing files. * If set to 1, the program will search the resultfolder and use any existing files instead of reprocessing. * Files that are not found in resultfolder will be processed. * Only file names will be checked. So, make sure parameters such as file size remain the same. * As these corrections are run after the phase closure check and will use the phase closure mask, it is recommended to set this to 0 once the parameters of the phase closure check have changed. * Default = 1. |
| orbit\_quad\_fit\_sampling | Int | * Resampling factor when estimating orbit quadratic polynomial. * Default=4. |
| eds\_block\_size | Float | * Block size to estimate a and b in elevation dependent signal removal. * Unit in km. * Default=100. |
| demfile | String | * A DEM file used only in EDS removal. |
| demfile\_header | String | * The header file of the provided DEM file. Use the same format as defined in Appendix 1. |
| if\_keep\_corrected\_ifg | Int | * if keep the corrected ifgs after orbital removal and eds removal. Ifgs after orbital removal will be named as \*. remove\_orbit and ifgs after eds removal will be named as \*.remove\_hgt. * 0: delete. * 1: keep. * Default=0. |
| minimum\_temporal\_ifg\_ratio | Float | * temporal\_ifg\_ratio=(valid\_ifg\_number)/(total\_ifg\_number). Pixels whose temporal\_ifg\_ratio is less than this minimum\_temporal\_ifg\_ratio will be masked out. * Default=0.3. |
| **Part 5: Define phase closure checking parameters (unwrapping errors)** | | |
| if\_check\_loop\_closure | Int | * 0: Switch off. * 1: Perform an unwrapping error check. * Default=1. |
| if\_use\_existing\_phasemask | Int | * 0: Do not use existing files. * 1: Use existing files. * If set to 1, the program will search the resultfolder and use the existing phase\_closure\_mask file. * Default = 1. |
| if\_delete\_bad\_ifgs | Int | * 0: Do not delete ifgs that have a mean phase closure std (calculated from all of its loops) larger than whole\_ifg\_closure\_std\_threshold. * 1: Delete those ifgs. * Default=1. |
| whole\_ifg\_closure\_std\_threshold | Float | * A threshold deciding whether to delete ifgs that have too many unwrapping errors. * Typically, an ifg will have 2-5 loops (depending on your network) and each loop will produce a phase closure map. The phase closure std is calculated for each ifg from all of its possible loops. * Typical values are 3.14 to 6.28. * Set it to -1 to decide automatically. If is the phase closure std for the i-th ifg, the threshold is given by (mean+mstd\*confidence\_level).   where:  Default=-1. |
| loop\_misclosure\_threshold | Float | * A threshold to define a failed loop. * This is for a single pixel. * Default=3.14. |
| loop\_misclosure\_ratio\_threshold | Float | * A threshold to delete pixels that have many failed loops. * This is performed pixel-by-pixel and for each pixel, we calculate the number of failed loops (phase loop closure larger than loop\_misclosure\_threshold) and the total loop number. * Pixels with (failed loop number)/(total loop number)>loop\_misclosure\_ratio\_threshold will be deleted from all ifgs. * Default=0.5. |
| loop\_closure\_mean\_std\_threshold | Float | * A threshold to delete pixels that have too large closure phases. * This is performed pixel-by-pixel and for each pixel, and for each pixel we calculated the std of its closure phases from all of its possible loops. Pixels with a std that is larger than this loop\_closure\_mean\_std\_threshold are deleted from all ifgs. * Typical values are 3.14 to 6.28. * Set it to -1 to decide automatically. If is the phase closure std for the i-th pixel, the threshold is given by   where:  Default=-1. |
| confidence\_level | Float | * Used by whole\_ifg\_closure\_std\_threshold and loop\_closure\_mean\_std\_threshold. This is to define a confidence level of deleting ifgs. The larger the value, the more confident in deleting ifgs (i.e., the lesser the deleted ifgs). * Default=2.0 (two-sigma). |
| if\_allow\_unlooped\_ifgs | Int | * 0:Exlcude those ifgs. * 1: Allow those ifgs. * Unlooped ifgs are those whose total loop number < minimun\_loop\_num. * Default=1. |
| if\_allow\_unlooped\_pixel | Int | * 0:Exlcude those pixels. * 1: Allow those pixels. * Unlooped pixels are those whose total loop number < minimun\_loop\_num. * Default=1. |
| minimun\_loop\_num | Int | * Minimum number of loops. * Default=0. |
| **Part 6: Definbe constant parameters** | | |
| wavelength | Float | * Wavelength of the radar signal in meters. * Default=0.055165 (C-band). |
| incidence | Float | * Incidence angle in degree. |
| orbit\_altitude | Float | * Orbit altitude in meters. |
| **Part 7: Define time series inversion parameters** | | |
| if\_inverse\_time\_series | Int | * 0: Switch off. * 1: Perform time series inversion. |
| if\_est\_hgt | Int | * 0: Do not estimate the DEM error. * 1: Estimate the DEM error. A baseline file must be provided. * Defaut=1. |
| temporal\_constraint | Int | * 0: No temporal constraint on displacement time series. * 1: Apply a linear constraint on displacements (D=vt). * 2: Apply a logarithmic constraint on displacements (D=vln(t). * Default=1. |
| temporal\_strength | Float | * The strength of the temporal constraint. Defined as the standard deviation of the weight. * Default=0.1. |
| simple\_log\_origin | Int | * If temporal\_constraint=2, then an origin date must be defined which is earlier than the start date of the time series. * Define this in the format of YYYYMMDD. |
| **Part 8: Define simple stacking parameters** | | |
| If\_stacking |  | * 0: Switch off. * 1: To calculate a linear mean velocity map based on stacking as shown in Section 4.3.n * The simple linear stacking will only calculate a mean velocity map, not the displacement time series. * Default=0. |
| **Part 9: Define time series filtering parameters** | | |
| if\_apply\_filter | Int | * 0: Switch off. * 1: Apply a spatial filter only. * 2: Apply a temporal filter only. * 3: Apply a temporal filter and then a spatial filter. * 4: Apply a spatial filter and then a temporal filter. * 5: Apply a 3D Gaussian filter in the frequency domain. * Default=5. |
| spatial\_filter\_method | Int | * Not used in this version. * Currently, only a Gaussian low pass filter in frequency domain is supported. |
| spatial\_filter\_sigma | Float | * Define a Gaussian sigma. * Signals with a wavelength (=sigma/pixel\_size) shorter than this will be downplayed (low pass). * The smaller the value, the smoother the result. Typical values are about 30% of the ifg length (number of pixels). * Set it to -1 to use 0.3\*(width+length)/2.0, where width and length are total row and column numbers of the ifg. * Default=-1. |
| temporal\_filter\_method | Int | * Not used in this version. * Currently, only a Gaussian low pass filter in frequency domain is supported. |
| temporal\_filter\_sigma | Float | * Define a wavelength cutoff in days. * Any signals with a wavelength shorter than this will be downplayed (low pass). * The smaller the value, the smoother the result. Typical values are about 1~3 times of the average date interval of the time series in days. * Set it to -1 to use (firstdate-lastdate)/(number of dates). * Default=-1. |

**4 Some maths**

**4.1 What is a phase closure loop.**

Now we have three interferograms namely:

A=20201201-20201213.phs

B=20201201-20201225.phs

C=20201213-20201225.phs

We can form one phase closure loop using these three interferograms and if there are no unwrapping errors on all of them, then the following equality holds for each of the pixel:

A-B+C=0

We define the residual of this loop as misclosure residual. Unwrapping errors will introduce multiple cycles of misclosure residual (e.g., 3.14, 6.28 radians). Other data processing procedures such as multi-looking will also introduce misclosure residuals but should be small and random.

This program uses misclosure residual to detect and delete interferograms or pixels that have unwrapping errors as discussed in (Morishita et al., 2020). Two quantities are especially important, i) the overall failed loop number (e.g., a loop is failed when its misclosure residual is larger than a predefined value); ii) the standard deviation of the misclosure residual of a pixel on all loops. We do this on a pixel-by-pixel basis and also calculate the mean statistics for each interferogram in order to pick out some bad candidates.

**4.2 Time series inversion**

We use a weighted least square to estimate the displacement time series. For each interferogram (between date i and date j), we have the following observation equation:

(1)

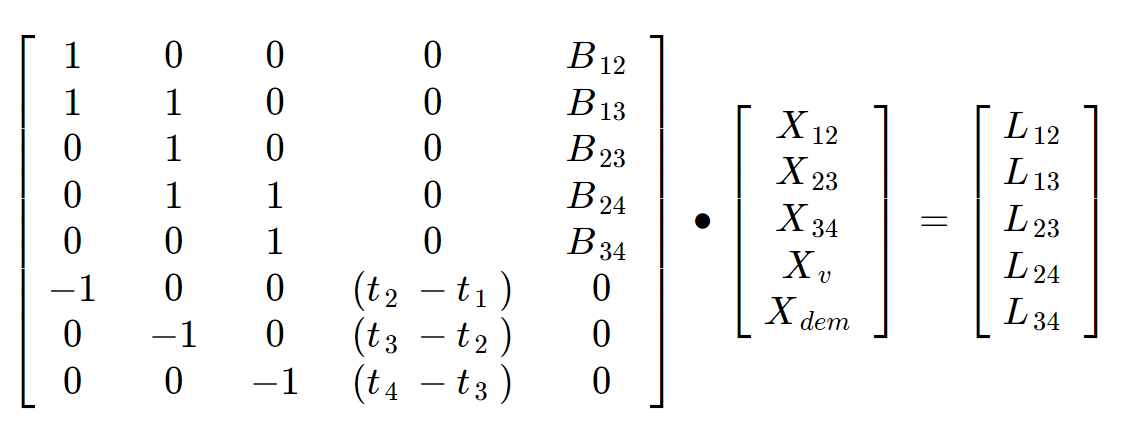
If there is a linear constraint, then:

(2)

If there is a logarithmic constraint, then:

(3)

For example, if we have five interferograms, the displacement can be estimated with a linear constraint applied as:

 (4)

Where B is the coefficient defined by interferogram baselines as in (1). The above equation can be written as:

(5)

The solution is given by:

(6)

Where P is the weight matrix of the observation. The linear strength is defined as the weight of the linear equations.

**4.3 Linear simple stacking**

To estimate only a linear velocity without displacement time series, we use the following observation equation:

(7)

The solution can be given as:

(8)

The mean square root of observation residual is:

(9)

The standard deviation of the estimated linear velocity is:

(10)

**4.4 Frequency domain filtering**

We use a Gaussian filter kernel in the frequency domain to perform a low pass filter both spatially and temporally.

For 1D temporal filtering, we use:

(11)

For 2D spatial filtering, we use:

(12)

For 3D spatial-temporal filtering, we use:

(13)

Where sigma0, sigma1, sigma2 defines the strength of the filter for temporal and spatial, respectively. We use the same strength for x and y so sigma1 is always equals to sigma2.

Essentially it means signals with wavelengths that are shorter than () are downplayed and signals with wavelengths that are shorter than () are downplayed.

**LICENSE**

INSARTS is a freely available software package. We only ask that proper credit be given to the authors. A license file can be downloaded at: <ftp://www.gacos.net/pub/share/software/licence>

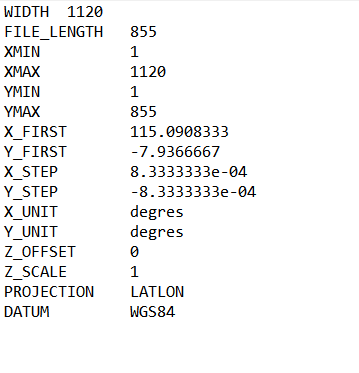
**References**

Li, Z., Fielding, E. J., & Cross, P. (2009). Integration of InSAR time-series analysis and water-vapor correction for mapping postseismic motion after the 2003 Bam (Iran) earthquake. IEEE Transactions on Geoscience and Remote Sensing, 47(9), 3220-3230.

Yu, C., Li, Z., & Penna, N. T. (2020). Triggered afterslip on the southern Hikurangi subduction interface following the 2016 Kaikōura earthquake from InSAR time series with atmospheric corrections. Remote Sensing of Environment, 251, 112097.

Morishita, Y., Lazecky, M., Wright, T. J., Weiss, J. R., Elliott, J. R., & Hooper, A. (2020). LiCSBAS: An Open-Source InSAR Time Series Analysis Package Integrated with the LiCSAR Automated Sentinel-1 InSAR Processor. Remote Sensing, 12(3), 424.

**Appendix 1: an example of headerfile**



**Appendix 2: an example of baseline file**

