link error

Page Web
Deformation MSBAS

Maxime Jaspard , Gilles Celli , Nicolas d'Oreye

Argentina : Domuyo and Laguna Del Maule region
Congo : V i runga Volcanic Province
Luxembourg : Grande region
La Réunion : Piton de la fo urnaise

Table of Contents

Generalities

WARNING
The present document is written for the specific configuration at ECGS for 4 specific targets . Password and user name s are replaced by green italic fake names.

We also compute and store the geocoded products and MSBAS inversion s and host the web pages on different servers. Their names are also generalized here in green italic .
S cripts for MasTer processing are stored on COMPUTE_SERVER .
Geocoded products and MSBAS inversions results are stored on STORAGE_SERVER.
Web pages are hosted on WEB_SERVER .

Adapt th ese variables and names to your needs.
Note that at ECGS, we are dealing with two WEB_SERVER with similar names ending with number 2 and 3. Specificities associated to those duplicate servers are to be found in the present document and may not apply to your needs.

Some specific scripts related to the web page management are not stored in the MasTer GitHub repository because they contain ECGS logins, addresses etc. However, a cleaned version can be requested to maxime@ecgs.lu .

Document purpose.
The purpose of this document is to have a clear overview of the different scripts used in the Web pages “ defo_ region ”, and how the Web folder is structured to troubleshoot easily in case of issue.
A section at the end of the document will explain how to create new region.

Web page purpose.
The purpose of this web page is to display the main products produced by MasTer with an automatic update (every 3hours). These products can be the last amplitude images, the speed deformation maps, the time series deformation for pre-defined pairs of points …
Some tool has been developed to display time series, interferogram, coherence map or delta deformation between 2 dates with different scrolling menu to allow the web user to choose dates, orbit, acquisition mode etc …

Web Server location
The web pages are located on WEB _SERVER 2 server in the following folders:
Congo: /Library/Server/Web/Data/Sites/defo- vvp
Argentin a : /Library/Server/Web/Data/Sites/defo- domuyo
Lux embourg : /Library/Server/Web/Data/Sites/defo-lux

Piton de la fournaise : /Library/Server/Web/Data/Sites/defo- PF

All these pages are redirected to WEB _SERVER 3 to the following folder: / opt/local/www/apache2/html /*

Update Web pages

The web pages are update d every 3 hours, the scripts are running on WEB _SERVER 3 .
COMPUTE_SERVER and STORAGE_SERVER must be available on the network. If one of these servers are not available, the process will try to connect every minute . After 3 attempts, an email will be sent to maxime@ecgs.lu.
Web pages folder
For each region, we do have one specific folder named as ‘ defo _ region ’ .
In this main folder ‘ defo_xxx ’, we will always find the following subfolder:

Bootstrap Standard data used for the web pages style in interaction with html tag
css , src , includes , js Source code , library use for the html
Documents Contain all (static) documents used by scripts but that won’t be created or modified by these scripts (Do not delete th ese document s) . All documents are common to any region except “ satview.tif ” and “ terrain.tif ” which are mandatory to display some location on different maps style such as google earth or google relief. The creation of these image s is explained at the end of the document in “ Side procedure ”.
images Contain all the data periodically downloaded, modified and generated by scripts. All the files in this folder will be re-generated by the main script process. (All the files can be deleted if needed). As this folder is quite important, the next paragraph will explain deeply his structure.
TS_Custom This folder is the working directory and contains all the documents related to the “Custom Times series” tool on the webpage .
DB_Interfero This folder is the working directory and contains all the documents related to the “ Interfero Creation ” tool on the webpage .
Move_map This folder is the working directory and contains all the documents related to the “ Delta deformation ” tool on the webpage .
RGB_map This folder is the working directory and contains all the documents related to the “ RGB images ” tool on the webpage .
Files All the files located in ‘ defo_xxx ’, are php script, logo, gif, log files…
Parameters.txt This file contains all the parameters, path, data specific to one region. This file will be used in a lot of scripts to create variables. We will not explain each parameter here, but you will find in the file itself a small explanation written for each parameter. All the parameters that must be adapted to a specific regions are explained in the scripts explanation.

In the main Web pages ‘ defo_ region ’, we will also find all the php scripts used to build the page on the web. The first mandatory script read after typing “http:// WEB _SERVER 2.ecgs.lu/ def o _ region ’ is “ index.php ”.

Images folder
This folder contains all the images and information present on the web pages. Scripts will work almost exclusively with this folder to store the data. Image folder contains subfolders :

Amp_Coh_Defo Contain all the amplitude, coherence and deformation images (downloaded and modified).
GD_Linear_Rate Contain all MSBAS binary files (downloaded and modified).
Time_Series TS_all is used to store temporarily data during main process
TS_sorted contain all the time series images present in web pages
z.. acq _mode contains image in different acquisition mode if it exist.
RGB Contain all RGB overlay images that has been automatically ran from script “main.sh” and displayed on web page.
xTimeData Files containing time information present in the web pages.
Baseline Plot Storage of baseline plot document (updated automatically) TS _Custom folder
This folder is owning all the documents and files related to the specific “Custom Times series” tool on the webpage . This webpage is used to generate a customizable time series jpg file requested by a web user.

data Amplitudes and deformation with xml linked file created by gdal and updated when needed. These images are converted in specific format to generate an interactive map online.
Request contains one folder per request that includes time series jpg file.
RequestArchive contains a text file per request including information of the request.
RequestSent contains a text file per request to confirm request has been sent by email successfully.

DB_Interfero folder
This folder is owning all the documents and files related to the specific “ Interfero Creation” tool on the webpage . This webpage is used to generate an interferogram and coherence map between 2 images selected by the visitor.

D at 1_date2 One folder is created by request. The folder contains working files during the process but keep only the final product (interfero + coherence map)

Move_Map folder
This folder is owning all the documents and files related to the specific “ Delta Deformation” tool on the webpage . This webpage is used to generate a differential map between 2 deformations images selected by the visitor.

Product One folder contains all the last (max 10) png files available on the web pages.

Script execution

Generalities

All the shell or python scripts used to generate online data are stored on WEB _SERVER 3 server at following location: a dmin _user /scripts/InSAR-deformation-web-scripts/ script OK .
All these scripts are common for any region.
The script “Main.sh” is executed periodically as a cron job and will used several secondary scripts in the process. The difference between a region to another is only the argument used for ‘Main.sh’ which is the file ‘parameters.txt’.

```
0 1,5,9,13,17,21 * * * /Users/ admin_user /scripts/InSAR-deformation-web-scripts/script4/Main.sh /opt/local/www/apache2/html/ defo-vvp /parameters.txt > /opt/local/www/apache2/html/ defo-vvp /crontab_log.txt 2>&1
0 2,6,10,14,18,22 * * * /Users/ admin_user /scripts/InSAR-deformation-web-scripts/script4/Main.sh /opt/local/www/apache2/html/ defo -PF/parameters.txt > /opt/local/www/apache2/html/ defo -PF/crontab_log.txt 2>&1
0 3,7,11,15,19,23 * * * /Users/ admin_user / scripts/InSAR-deformation-web-scripts/script4/Main.sh /opt/local/www/apache2/html/ defo -lux/parameters.txt > /opt/local/www/apache2/html/ defo -lux/crontab_log.txt 2>&1
0 0,4,8,12,16,20 * * * /Users / admin_user / scripts/InSAR-deformation-web-scripts/script4/Main.sh /opt/local/www/apache2/html/ defo-domuyo /parameters.txt > /opt/local/www/apache2/html/ defo-domuyo /crontab_log.txt 2>&1
```

Main Script

The main script will manage sequentially the whole process . The command information is recorded in a log file located in :
“ / Library/Server/Web/Data/Sites/defo -region / crontab_log .txt”.
In the command line, we can see all the step of the script with a timespan for each line. We will use this structure to give all the information about each step.
As described above, th e script take s one argument, the file “parameters.txt” located in the main folder “defo- region ”
Here is an example of the command window after the run of the main script:

```
2021-05-18 08:00:00___
2021-05-18 08:00:00___---------- START ----------
2021-05-18 08:00:00___
2021-05-18 08:00:00___.....
2021-05-18 08:00:00___==> VARIABLES CREATION
2021-05-18 08:00:01___-----> Variable declaration done
2021-05-18 08:00:01___.....
2021-05-18 08:00:01___==> CHECK CONNECTIVITY TO INSAR SERVER
2021-05-18 08:00:01___-----> Check server connectivity
2021-05-18 08:00:01___-----> Server connectivity ok
2021-05-18 08:00:01___.....
2021-05-18 08:00:01___==> MAIN PROCESS
2021-05-18 08:00:01___----- > Start of the loop: we do have 1 modes:
2021-05-18 08:00:01___
2021-05-18 08:00:01___=====> VARIABLE DECLARATION for mode 1:
2021-05-18 08:00:01___=====> DOWNLOAD DATA FROM STORAGE_SERVER SERVER
2021-05-18 08:00:09___=====> ORBIT ASCENDING
```

2021-05-18 08:00:09___----------> Check if new deformation binary has been created for EW, UD or LOS Asc
2021-05-18 08:00:09___----------> These deformation binary files has already been processed
2021-05-18 08:00:09___
2021-05-18 08:00:09___=====> ORBIT DESCENDING
2021-05-18 08:00:09___----------> Check if new deformation binary has been created for LOS Desc
2021-05-18 08:00:09___----------> This deformation binary file has already been processed
2021-05-18 08:00:09___----- > End of the loop -
2021-05-18 08:00:09___.....
2021-05-18 08:00:09___==> TIME SERIES MANAGEMENT
2021-05-18 08:00:10___Reassign Variable to point to the mode 1 as per default
2021-05-18 08:00:10___
2021-05-18 08:00:10___-----> Download all time series from InSAR server in jpeg format
2021-05-18 08:00:11___-----> Rename timeLines _ into timeLine for LOS and create thumbnail in mini folder
2021-05-18 08:00:12___-----> Rename all Combi.jpg to Combi_EW_UD.jpg
2021-05-18 08:00:12___-----> Create Thumbnail for all Combi files

```
2021-05-18 08:00:17___-----> Create Sentinel 1 amplitude image per areas and draw pairs on it
2021-05-18 08:00:21___-----> Create Google maps image per areas and draw pairs on it
2021-05-18 08:00:49___-----> Create Google sat image per areas and draw pairs on it
2021-05-18 08:01:19___-----> Remove all zoom.jpg that do not have any combi file anymore and verify if all file a present
2021-05-18 08:01:19___Create symbolic link in each type of view subfolder \( S1, maps and earth\)
2021-05-18 08:01:27___.....
2021-05-18 08:01:27___==> TIME SERIES MANAGEMENT (ADDITIONAL MODE)
2021-05-18 08:01:27___Download file from STORAGE_SERVER server and copy them in a folder of the related mode
2021-05-18 08:01:27___.....
2021-05-18 08:01:27___==> CHECK LAST DATE FOR EACH ORBIT
2021-05-18 08:01:27___.....
2021-05-18 08:01:27___== > TOOL 'CALCULATE TIME SERIE' PREPARATION
2021-05-18 08:01:27___-----> Coordinates.txt already exist
2021-05-18 08:01:27___.....
2021-05-18 08:01:27___== > TOOL 'INTERFERO CREATION' PREPARATION
2021-05-18 08:01:29___.....
2021-05-18 08:01:29___== > TOOL 'DELTA DEFORMATION' PREPARATION
2021-05-18 08:01:30___-----> List availabale deformation maps for Mode
2021-05-18 08:01:30___.....
2021-05-18 08:01:30___== > DOWNLOAD BASELINE PLOT
2021-05-18 08:01:37___
2021-05-18 08:01:37___---------- END ----------
2021-05-18 08:01:37___
```

Depending on the availability of new data, the process can take more or less time. In the case above, we can see that the deformation files are already been processed for ascending and descending orbit, which generate a relative short process (+- 3 minutes). This can take 15-20 minutes if the process need s to update all the data.

Secondary Script

Several secondary scripts will be call in the main scripts, and also several others scripts will be call in these secondary scripts . Here under the hierarchy of the scripts:

Main.sh InSAR_Web_Connect.sh
Mask_Builder.py
Amplitude_Average.py
Fiji_Amp_Defo_Coh.sh CreateColorFrame.py
ImageCreator.sh Amplitude_log10.py
KMzCreator.sh GetMinMax .py
Envi2ColorKmz.sh
Fiji_Amp_Defo_Coh _basic .sh CreateColorFrame.py
ZoneMaker _ 2 .0 .sh
ZoneMaker _ 1 .0 .sh
TS_ReadLastPoint.sh

GetXYfromENVI.py
CheckNewInterfero.sh
Update_RGB.sh Run_RGB.py

Main Process explanation (Main.sh)

The main script is automatically processed using crontab every 3 hours.
We can split the script in the next subdivision:
- VARIABLES CREATION
- CHECK CONNECTIVITY TO INSAR SERVER
- MAIN PROCESS (ASCENDING AND DESCENDING)
- VARIABLES DECLARATION FOR MODE x
- DOWNLOAD DATA FROM STORAGE_SERVER SERVER
- CREATION OF SPEED DEFORMATION IMAGE
- CREATE KMZ files
- CREATION OF SPEED DEFORMATION IMAGE (EPSG:4396 FORMAT)
- TIME SERIES MANAGEMENT
- TIME SERIES MANAGEMENT (ADDITIONAL MODE)
- CHECK LAST DATE FOR EACH ORBIT
- TOOL 'CALCULATE TIME SERIE' PREPARATION
- TOOL 'INTERFERO CREATION' PREPARATION
- TOOL 'DELTA DEFORMATION' PREPARATION
- UPDATE RGB MAPS
- DOWNLOAD BASELINE PLOT

variables creation :
Syntax of variables: W_** = Path to a folder in the Web Server
GD = Ground Deformatio n
TS = Time series
A_** = Ascending
D_** = Descending

- All the variables are read and derived from “parameters.txt” file.

check connectivity to insar server
- Check if all mounted path s are available.
- If not, execute InSAR_Web_Connect.sh that will create and give appropriate right to mounting point. Then, mount the path to the mounting point. After 3 attempts, a mail is sent to maxime@ecgs.lu and the script stops.

main process
- The main process consists of the creation of speed deformation and amplitudes maps

- VARIABLES DECLARATION FOR MODE x
- Affectation some variables to the target mode.
- DOWNLOAD DATA FROM WEB SERVER
- Download all MSBAS_LINEAR _RATE binary files in (images/ GD_Linear_Rate) from STORAGE_SERVER Server.
- Write t he modified date of these file s in $ WebSite / xTimeData /GD_LR_**_time.txt.

- **ORBIT ASCENDING * :**
- Compare these dates with the previous one:
- if new data, continue
- if data are same than last run, jump to “ORBIT DESCENDING”
- Delete all previous file .
- Copy all new file from STORAGE_SERVER Server (binary + header)
- Record this new creation date for each deformation binary file.
- Download 10 last Amplitude file and create an average amplitude bin file using Amplitude_Average.py
- CREATION OF SPEED DEFORMATION IMAGE

- Creates all mask using Mask_Builder.py
- Creates speed deformation map using Fiji_Amp_Defo_Coh.sh
- Creates last and average amplitude image using ImageCreator.sh
- Record the last date of amplitude image.
- CREATE KMZ FILES
- Using KMzCreator.sh
- CREATION OF SPEED DEFORMATION IMAGE (EPSG:4396 FORMAT)
- Delete previous file
- Copy defo files and converted them in EPSG:3857 using gdalwarp
- Creates speed deformation map using Fiji_Amp_Defo_Coh _basic .sh
- **ORBIT DESCENDING:**
- - idem as Orbit ORBIT ASCENDING* with descending binary file (ampli + defo)
-
-

time series management
- Reassign variable to point to the mode 1 in case of multi-mode
- Download all time-series images from InSAR server in ‘image/ Time_Serie ’ folder .
- Rename some files and create thumbnail
- Create cropped S1 images and draw pairs of point on it (idem for google maps and earth image) using the script ZoneMaker_2.0.sh .
- Performed verification and alert by email if number of files not coherent
- Sort by areas all files by creating symbolic link in different folder using the script ZoneMaker_1.0.sh .

time series management (additional mode)
- If time series of same pairs are available for other acquisition mode, we will download these files and manage to get them on the web page with a simple click
- Download all time-series images for extra acquisition mode from InSAR server in ‘image/ Time_Serie / zModeName ’ folder .
- Change right on files, and create thumbnails .
- Create symbolic link of all cropped image (S1, maps and earth) in the target folder for the web interface display.

check last date for each orbit
- For each mode and each deformation orientation (LOS Asc , LOS Desc and Up/Down), we will loop in the MSBAS/

Region_TS …/_ Time_series directory, and extract from the text files the date of the last available data in time series using TS_ReadLastPoint.sh .
- Example of path:
(
/D3602/MSBAS/_VVP_S1_Auto_20m_400days/zz_LOS_TS_Asc_Auto_2_0.04_VVP/_Time_series/timeLine769_875.txt
)

tool ‘calculate time serie ’ preparation
- This tool doesn’t require any other periodic process, as the last speed deformation image is already updated or created in the main process for this tool (see Main Process/Creation of speed deformation image in format EPSG:4396).
The only thing we want to do here, is to verify is the file “coordinates.txt” is well present (used to geo-localized the speed deformation map on a web interactive map). If the file is missing for any reason, we will generate it again using GetXYfromENVI.py

tool ‘ interfero creation’ preparation
- The maintenance and updating process of all data involved in this tool is performed by a specific script named CheckNewInterfero.sh that take the file ‘paramters.txt’ as argument.

tool ‘ delta deformation’ preparation
- For this tool, we want to update the list of MSBAS binary deformation file for each acquisition mode and each deformation direction. The procedure is just redirecting the output of an ‘ls’ command in text file. This will allow the web interface to propose in a scrolling menu all the available deformation file in order to compare two of them.

Update RGB maps
- This part will check if the RGB overlay image (Amplitude Primary, secondary and relative coherence file) has been created with the last amplitude images. If not, the RGB image will be created and displayed on the web page.

download baseline plot
- The baseline plot is available on the web page, and need only a conversion between eps form to jpg format.
- We just loop throw each acquisition mode, copy locally the baseline plot for each deformation direction and convert the file from eps to jpeg.

Scripts (Main Process)

InSAR_Web_SMB_Connect.sh
Dependencies:
- Script need to be allow to run as root in sudoers file:
(admin _user ALL=(ALL) NOPASSWD: /Users / admin_user / scripts/InSAR-deformation-web-scripts/script4/InSAR_Web_SMB_Connect.sh)
Arguments:
- Argument 1 = parameters file
Action:
- Create mounting point if doesn’t exist
- Change owner of these mounting points
Amplitude_Average.py
Dependencies:
- python 3.8 including Numpy function
Arguments:
- Argument 1 = Folder where binary amplitude files are located
- Argument 2 = Binary amplitude average file (output path)
Action:
- Loop in the folder and convert in array each file that contain ‘ deg ’ in the end
- Calculate the logarithm (base 10) of the array
- Addition all the arrays and divide by the number of files
- Write output in Argument 2

Mask_Builder.py
Dependencies:
- python 3.8 including Numpy function
Arguments:
- Argument 1 = Binary deformation file (input)
- Argument 2 = Mask file (output)
Action:
- Convert input as an array
- Divide each element by himself and multiplicate by 0.9
- Write output in Argument

Fiji_Amp_Defo_Coh.sh

Comments:

- This script will create an image that illustrate a deformation file, masked by a coherence file and stacked on the amplitude file for better spatial representation. The color of each pixels represents the deformation value. These values are represented on a legend.

Dependencies:

- Fiji (ImageJ)
- gnu sed for more compatibility
- Python + Numpy + script: CreateColorFrame.py
- ${PATHFIJI} need to be declared in . bashrc

Arguments:

- Argument 1 = Amplitude file (average file)
- Argument 2 = Coherence file (mask file)
- Argument 3 = Deformation file (binary file)
- Argument 4 = AMPLI_COH_MSBAS_LINEAR_RATE_** (output file)

Action:

- Variables declaration
- Extraction of width and length data from amplitude header file
- Add color frame in binary file using CreateColorFrame.py
- Create the ImageJscript in a temporary file FijiMacroDefo2Amp.txt and execute it with ImageJ command.

This script will create the 2 following tif file:

Amplitude average

Speed Deformation filtered by coherence and stacked on the amplitude.

- Extract the data from TempFile (written by CreateColorFrame.py) + variables declaration.
- Draw on the colorframe a vertical bar and the value at lowest, zero , highest value + direction.
- Crop the speed deformation image following parameters and draw a black rectangle on the legend.
- Extract the legend in a separate jpeg file

CreateColorFrame.py

Dependencies:

- Python 3.8 + Numpy
- gnu sed for more compatibility
- Python + Numpy + script: CreateColorFrame.py

Arguments:

- Argument 1 = Deformation file (binary file)
- Argument 2 = Coherence file (mask binary file)
- Argument 3 = Amplitude file (average binary file)
- Argument 4 = Width (data from amplitude header file)
- Argument 5 = Temp file (used to report value in the mother script)
- Argument 6 = Parameters file (used to re ad parameters)

Action:
- Variables declaration
- Convert in array the 3-binary file
- Read parameters file for margin and legend width.
- Record the min/max value of deformation file
- Calculate delta = (max-min)/ LegendWidth
- Copy the 3 arrays ‘ Raw ’ to a new array ‘ Mod ’
- Record the max value of Amplitude array (= lightest value of image)

- Create a rectangle in the top left corner in the amplitude image where the legend will be located. The size of this rectangle is defined with variables from parameters file.
- In amplitude file, write the recorded max value to create a light square
- In coherence file, write 0.0 to avoid any deformation info at this place

- Create another rectangle in the deformation image where the graduated color frame will be located.
- In deformation file, create a horizontal frame in this rectangle with an incremental loop . (using parameters file for size)
- In coherence file, write 0. 9 to make deformation info visible at this place .
- We write in the 1 st pixel of deformation file a value = 120% of highest value.

Fiji will colorize the maps with a **complete (RGBR)** resolution of RGB from lowest value to highest value. It means that lowest value=Red and highest value = Red. To avoid this, we force one pixel to be equal to 120% of highest value. Like this our deformation map will be colorize from Red= Lowest value to Pink= Highest Value

Here is a small table to understand why we force a value of a pixel 120% higher than the highest pixel. Consider for example the lowest value = 0 and highest = 10 0 :

Color code	**Deformation [%] (without forced pixel)**	**Deformation [%] (with forced pixel)**
Red	0 [No deformation]	0 [No deformation]
Yellow	17	2 0
Green	33	4 0
Magenta	50	6 0
Blue	67	8 0
Pink	84	10 0 [highest deformation value]
Red	100 [highest deformation value]	12 0 [forced pixel]

We see that with forced pixel , we won’t have any more value between pink and red in the deformation map . (excepted one pixel)
- Write the 3-output binary file (= Input file + _2.0)
- Create an array “ array legend ” where we write the min/max value of deformation, and the 3 position s where these value s and the 0 must be written in the legend. Write this array “ Array_legend ” in a temp file which will be used by the mother script to draw the legend.

ImageCreator.sh

Comments:
This script is used to create an image file from a binary file. The script will search in a folder for the latest binary file modified with a file name ending by “deg”, and then using image J convert the file in a Jpeg format.

Dependencies:
- python for script ‘ Amplitude_log10.py ’
- ImageJ
- Gnu gsed
- Header file available (envi format)
Arguments:
- Argument 1 = Folder containing the amplitude file
Action:
- Read crop values + min/max brightness value from parameters file
- Search for the more recent amplitude file + his header file

- Call python script to perform a logarithm fonction on the amplitude file, and write the output in same folder as LastImage using Amplitude_log10.p y .
- Extract data from hdr file
- Create the jpeg using ImageJ.
- Crop the image and save in the same folder (images/ Amp_Coh_defo) folder

Amplitude_log10.py

Dependencies:
- python with Numpy
Arguments:
- Argument 1 = File input (raw)
- Argument 1 = File input (log10 of each value)
Action:
- Perform the log10 function of file in argument 1 that contain.
- Write the new file in argument2.
KMzCreator.sh

Dependencies:
- python for script ‘ GetMinMax. py ’
- script ‘ Envi2ColorKmz.sh ’
Arguments:

- Argument 1 = Coherence binary file
- Argument 2 = Deformation binary file

Action:
- Create needed path for argument. (scripts and documents).
- Retrieve min/max value of the defo file using ‘ GetMinMax.py ’ .
- Copy the header with the new name (*_ WithMask).
- Extract the min/max value written by ‘ GetMinMax.py ’ to use as argument.
- Execute ‘ Envi2ColorKmz.sh ’ script to generate the KMz .

GetMinMax .py

Dependencies:
- python with Numpy

Arguments:
- Argument 1 = Deformation binary file

Action:
- Create a variable containing pathname argument 1 ending by “_info.txt”
- Convert input binary file into array
- Retrieve min and max value from the array as variables.
- Write the value in a file created with variable above “*_info.txt”

Envi2ColorKmz.sh

Dependencies:
- color table created (argument 1)
- gdal function in Bash

Arguments:
- Argument 1 = Deformation binary file (*_ withMask)
- Argument 2 = Color table
- Argument 3 = Minimum value of deformation file
- Argument 4 = Maximum value of deformation file

Action:
- Rescale the value of Arg1 file from [min max] to a new range: [0 - 1 0]
- Create RGB file using gdal function and colortable .
- Convert into KMz
- Change header filename and remove temporary file

Fiji_Amp_Defo_Coh _basic .sh

Comments:
- This script will create an image that illustrate a deformation file, masked by a coherence file and stacked on the amplitude file for better spatial representation. Value of deformation are NOT represented on a legend.

Dependencies:

- Fiji (ImageJ)
- gnu sed for more compatibility

- ${PATHFIJI} need to be declared in . bashrc
Arguments :
- Argument 1 = Amplitude file (average file)
- Argument 2 = Coherence file (mask file)
- Argument 3 = Deformation file (binary file)
- Argument 4 = AMPLI_COH_MSBAS_LINEAR_RATE_** (output file)
Action :
- Variables declaration
- Extraction of width and length data from amplitude header file
- Create the ImageJscript in a temporary file FijiMacroDefo2Amp.txt and execute it with ImageJ command. This script will create the 2 following jpeg file :
Amplitude average
Speed Deformation filtered by coherence and stacked on the amplitude.

ZoneMaker _2.0 .sh

Comments:
- This script will create all the maps where we can visualize the pair of point for which time series are calculated. Some maps will show all the pairs of point in a certain zone, other map will show all the different zone in the region, some maps will show only one pair of point in a specific zone etc …

Dependencies:
- ImageJ
Arguments:
- Argument 1 = Input file (image) from which all crop will be done . (This image must be in the document folder as satview.tif or terrain.tif)
- Argument 2 = Path to Time Series folder
- Argument 3 = Name of TS_sorted directory depending on input type file (sentinel map, google earth ...)
- Argument 4 = Rate of resolution between input file and pixels coordinate extract from name of the file
- Argument 5 = Yes/No (Creation of areas in case of change)

Action:
- Variables declaration
- Define different areas with a maximum of 5 . Area0 is the main area and can be the original image . Many coefficients must be calculated to adapt the resolution of the crop to the area size in order to draw points the same manner in all areas.
- If 5th argument is “Yes”, do the next step:
- Draw in Area0 all the others areas with a colored rectangle.
- For each area (1 to 5), create from Argument 1 the crop of the area 3 times in different resolution. One will

contain all the pairs of points in the area, the second will be duplicated for each pairs of point to draw points with a high resolution, and the third will be duplicated for each pairs of point to draw cross with a low resolution (thumbnail).

- Loop the folder (images/ Time_Series / TS_all) and perform the following action for each time series in UD/EW direction downloaded from STORAGE_SERVER server :
- Check if this pair has already been drawn on a map:
If not or if argument 5 = “Yes”, do:
- Write coordinates in an array
- Sort in different areas (by coordinate or by name)
- Calculate coefficient based on distance between 2 points.
- Draw the pair of points in the common image for all pairs.
- Calculate new coefficient based on distance between 2 points.
- Create another jpeg image and draw the pair of points on this new image (high resolution with circle to represent points).
- Create another jpeg image and draw the pair of points on this new image (ow resolution with cross to represent points).
(All these images are still in the “ TS_all ” folder for the moment)

ZoneMaker_1.0.sh
Comments:
- This script will sort all the files in specific folders used by web application to switch easily between maps (S1, google maps or earth) and between areas.

Dependencies:
- nothing
Arguments:
- Argument 1 = Path to Time Series folder
- Argument 2 = Type of zoom (ampli , earth or maps)
- Argument 3 = Target directory (ampli , earth or maps)
Action:
- Variables declaration (including coordinate of each zone)
- Loop In the “ TS_all ” folder for all time-series files and all maps with a type corresponding to the argument 2. For each of these files, do the following:
- Write coordinates in an array
- Sort in different areas (by coordinate or by name)
- Create a symbolic link for each file in the loop in the corresponding area folder within the target folder (Argument 3).

TS_ReadLastPoint.sh

Comments:
- This script will extract from txt file in MSBAS directory of STORAGE_SERVER server the first and last date of deformation files used to build the speed map deformation shown in the web page. This is done for each acquisition mode in all deformation direction.

Dependencies:
- nothing
Arguments:
- Argument 1 = Folder containing the timeline info
- Argument 2 = information tag
- Argument 3 = output file
- Argument 3 = mode number

Action:
- Retrieve in folder one file containing " timeLine " in his name
- Read in the last line of the file the date and time (must be last data in timeline)
- Format each data of date and time to publish in the web page.
- Write this formatted time data in the output file

GetXYfromENVI.py

Comments:
- This script will extract from an ENVI some data and calculate the coordinate (absolute) of the maps , write these values in a file that will be used by javascript to get an interactive map .

Dependencies:
- gdal
Arguments:
- Argument 1 = Amplitude file (binary file format ENVI EPSG:4326)
- Argument 2 = output file

Action:
- Open argument using gdal
- Extract width, height and get the GeoTransform output of the file.
- Calculate absolute coordinate and writhe them in an array

- Write this array in the output file.

CheckNewInterfero.sh

Comments:
- This script will update a database of all interferogram available in InSAR server. The database contains the following info for each interfero : Satellite, Orbit, Supermaster date, master date, slave date.
- This script will update a second database that contains all the pairs that haven't been taken into account because not in the spatial baseline.

Dependencies:
- 2 mysql database with appropriate column format (see Initiate_WebPages .sh)
Arguments:
- Argument 1 = parameters file

Action:
- Variables declaration
- Update different text file in DB_Interfero folder where we write all the available satellite, Region, acquisition mode, orbit, SuperMaster …
- Record the latest slave available in all different MassProcessing
- If the latest slave is newer than last time, do:
- Fill the first DB using text file to loop in each Satellite, Orbit, Supermaster etc... and fill the DB with all available Interferogram.
- Fill the second DB using text file "approximateBaselinesTable.txt" file in the STORAGE_SERVER 1650/SAR_SM/MSBAS/… directory. Extract from this file all the pairs inside the temporal baseline but outside the spatial baseline (This DB is used to show on the web page in grey the potential pairs that are not calculated because out of the spatial baseline)
- Send the results to maxime@ecgs.lu

Update_RGB.sh

Comments:
- This part will check if the RGB overlay image (Amplitude Primary, secondary and relative coherence file) has been created with the last amplitude images. If not, the RGB image will be created and displayed on the

web page.

Dependencies:
- 1 mysql database with appropriate column format (see Initiate_WebPages.sh)
- Run_RGB.py

Arguments:

- Argument 1 = parameters file

Action:
- Read all parameters value from parameters and RGB_parameters file
- Write default RGB min/max value in RGB_parameters.txt (also used for RGB manual tool)
- Check sequentially for the last 6 amplitude images (normally 3 asc and 3 desc), if the RGB image has been created.For each of the last 6 amplitudes file, several pair (primary/secondary) have been calculated. We want to create the RGB overlay image with only one pair that used the more recent primary image .
- If an RGB image is not done yet, create it using Run_RGB.py.
- Delete all oldest folder containing RGB images that have not been checked in previous process.

Run_RGB.py

Comments:
- This script is called by Update_RGB .sh and will create the RGB overlay image from 2 amplitude file and the relative coherence file

Dependencies:
- Specific python environment for this script including spectral package and numpy 1.22.2
- RGB_parameters.txt

Arguments:
- Both amplitude image (Primary + secondary) and coherence image file
- RGB_parameters.txt
- "-crop" is optional, the script will crop the image regarding parameters in RGB_parameters file

Action:
- Initialisation of dictionary 'Color' which link image and color (Primary amplitude = red, secondary amplitude = green ...)
- Read parameters from RGB_parameters file
- Prepare all variable (filename) and hdr file
- Modify hdr file for final RGB image
- Copy original binary file locally after replacing nan by 0
- Opening new binary file as an image and create jpeg for each input file .
- Concatenate the 3 arrays in a single numpy array of dimension 3x1 and write results in a binary file
- Opening the new array 3x1, and create rgb jpeg file

Tool “Calculate Time Serie”

Introduction

A tool has been developed to allow request from web user to calculate a time series for a specific pair of point between two dates. The points and two dates are customizable.

By clicking “Calculate time series” button on the top of the web pages, an interactive map of the region is display. The user can choose the map background style between google reliefs map or satellite view. On the front of this map, the legend allows to display the amplitude image (ascending or descending) or one of the 4 deformation images. Two pointers are available on the map , they are draggable and their coordinates are interactively display on the maps. With the display of all deformation file, we can easily see if the points are coherent or not on ascending or descending orbit. To get the time series calculated by email, the user must fill an email address , a request name and the timespan of the time series (start date + end date). By default, these dates are the extreme available (Oldest and newest deformation file available).

Behind this request, the web pages will start a shell script on WEB _SERVER 3 server. This script will interact with the server(s) hosting the geocoded products and MSBAS results to generate the time series file with the coordinates requested by web user. Then, jpeg files will be sent by email.

If the two last character of the “request name” are “+t”, the process will send in addition to the graphics the text files of the time series.

Scripts hierarchy

Index.php TS_ Creation.php Link.sh. CheckCoh.py
Custom_TS.sh TimeSeriesInfo_bis
PlotTS.sh (COMPUTE_SERVER)
PlotTS_all_comp.sh (COMPUTE_SERVER)

TS_Creation .php

- Variables + function declaration
- Check connectivity to COMPUTE_SERVER and COMPUTE_SERVER
- Collect all request information in a form.
- When submitted, do the following:
- Write Coordinate, mode and dates in “TS_Data.txt”.
- Write email, request info and dates in “ Request_info.txt ”.
- Execute “Link.sh parameters.txt”

- Display content of “ TS_Custom /message.txt” on the page (produced by link.sh to tell the web user the result of the request)
- The rest of the script are static text to display in html, and in javascript mainly the management of the layers and the draggable points in the map display.
Several j avascript library are mandatory as proj4js , leaflet …

Link.sh

Comments:
This script is called by TS_Creation .php after having submit ted the request. It will analyze the request and give a feedback to the web user about what is going to be calculated depending on the coherence of the requested pair of points.
This script is run under the user "_www" which is the default user when a script is called by a php script. This user does have limited right, we will use another script "Custom_TS.sh" to process the request.

Dependencies:
- script python: CheckCoh.py
- script shell: Custom_TS.sh
Arguments:
Argument 1: parameters file

Action:
- Check if the coordinates are different from the last request.

- Check if a request is already running. If yes, message info is display to the web user (Request already running, please wait)
- Extract coordinate data from file "TS_Data.txt" .
- Extract mode name + mode number from file "TS_Data.txt" .
- Extract others information from file " Request_info.txt " .
- Resume the request info and write it in "message.txt" to display on the page.
- Check coherence in both orbit of two points using " checkCoh.py " .
- Check coherence value in " CheckCoh.txt " produced by checkCoh.py
- Complete the "message.txt" to inform user if coherence is ok for pair of point.
- Write in a file "temp.txt" in Request folder the request Id (current date, coordinate + description) and the orbit where the time serie can be calculated. (LOS_Asc , LOS_Desc or both if coherence is > 0 in both orbit) This file will be used by Custom.sh to know what to do.
- Execute " Custom_TS.sh " under admin_ user user using sudo prefix (This execution must be written in / etc / sudoers file) The character '&' is written in the end of the line to continue the execution of the script (Link.sh) without waiting for the end of Custom_TS.sh, which can take more than 1 minute.
- Copy the file " TS_Data.txt " to " TS_Data_Done.txt " to avoid a duplicate consecutive calculation.
- Then, the web page " TS_Creation.php " will be refreshed and the web user will read the info about the calculation in progress.

CheckCoh.py

Comments: This script is called by Link.sh to check if pair of points choose by web user are coherent in ascending and descending orbit.
Arguments:
- Argument 1 4 = Coordinates of 2 points
- Argument 5 = Number of pixels in a row
- Argument 6 = Binary file mask ascending
- Argument 7 = Binary file mask descending
- Argument 8 = Output file with results (CheckCoh.txt")
Action:
- Convert coordinates in integer and write both binary file in arrays.
- Calculate the pixel number in the array using X and Y coordinate.
- Extract pixel value of both points in ascending and descending mask.
- Write the value in output file "CheckCoh.txt" .

Custom_TS.sh

Comments:
This script is called by Link.sh. It will manage the times series creation with the coordinates entered by web user. An email is sent with all the requested files to the web user.

Dependencies:
- Must be connected by local network to COMPUTE_SERVER and STORAGE_SERVER
- mutt must be installed and configured to send email
- /Request/temp.txt must be present TS_Custom folder
- Request_info.txt must be present in TS_Custom folder
- message.txt must be present in TS_Custom folder
Arguments:
- Argument 1 = Parameters file
- Argument 2 = Acquisition mode number
Action:

- Variables declaration
- Extract data from file, create working directory and store request
- Check connectivity to COMPUTE_SERVER and STORAGE_SERVER
- SSH command to COMPUTE_SERVER and do:
- Create if doesn't exist a folder " Region" in WEB_PAGES folder
- Check if STORAGE_SERVER is available from COMPUTE_SERVER
- Sent an email and create error.info file if STORAGE_SERVER server not available
- If "error.info" exists, sent an email to the web user to inform about the issue

- Check which calculation we are going to do (all, LOS Asc or LOS Desc) and copy the file “TS_Data.txt” in COMPUTE_SERVER inside the region folder.

- Create on COMPUTE_SERVER the time series for each mode by doing :
- Write on COMPUTE_SERVER in “ Region_Rundir.txt “ the region and MSBAS path of the current request .
- Execute SSH command to COMPUTE_SERVER , extract ing data from “ Region_Rundir.txt “ and “TS_Data.txt”, and run the script “PlotTS.sh” to create the time series.
- Remove eps file, and move the tie series file locally in web server. (Remove everything from COMPUTE_SERVER).
- Link each file to a variable and perform a binary count of the files .
- Preparation of data to send by email .
- S end the appropriate message and files to the web by decoding the binary count
- If number of files is not coherent, send an alert message to the support team, and inform the web user about the issue.

Tool “ Interfero Creation ”

Introduction

This tool has been developed to allow request from web user to illustrate on a map any interferogram available from InSAR server.

As the previous tool, the user can choose the map background style between google reliefs map or satellite view. On the front of this map, the legend allows to display t he last request (maximum 10). For each request, the coherence and the interferogram are available as a map .

Each interferogram calculated are composed of two deformation files (master and slave) responding to the temporal and spatial baseline. The oldest date is the master file and the newest is the slave file.

Two button s are available if we want to search pairs based on the slave or the master file. After this selection, the scrolling menu allow the user to choose the Satellite, Orbit, slave (or master) and corresponding master (or slave).

In the last scrolling menu, the date in grey are not available because no interferogram exists for these dates. These dates are potential pairs inside the temporal baseline but outside the spatial baseline (the spatial baseline is written at the right side of the underscore in [meter])

After submitted, the process will download from STORAGE_SERVER server the interfero and coherence binary file, and convert them to show on an interactive map.

Scripts hierarchy

Index.php InterferoMaps .php InterferoSlave.php RunFrom_php.sh RunFrom_php2.sh InterferoMaster.php

InterferoMaps .php

- Display last 10 requested pairs available on the maps
- Display 2 buttons to perform a new request.
- The rest of the script are static text to display in html, and in javascript mainly the management of the layers and the draggable points in the map display.

Several js library are mandatory as proj4js , leaflet …

Interfero_master / slaves .php

Comments:
This script is called by InterferoMaps.php . It allow s the user to choose which interferogram and coherence file to display on the map using scrolling menu

Dependencies:
- parameters file must be present at the same hierarchy of this script.
- “ db_connection.php ” must also be present here.
- Directory ' DB_Interfero ' must be owned by _www user.

- A mysql database must be created with 2 tables.
- Database name = InSAR_MASSPROCESS
- login = InSAR/ your_pwd
- Table1 = Region (ex: VVP)
- Table2 = Region _all (ex: VVP_all)

Action:
- Create function
- Create common variable
- Choose satellite write in ' DB_Interfero /aaPathToPair.txt' the satellite.
- Choose orbit write in ' DB_Interfero /aaPathToPair.txt' the orbit.
- Choose master write in ' DB_Interfero /aaPathToPair.txt' the master.
- Choose slave write in ' DB_Interfero /aaPathToPair.txt' the slave.
- A resume of the request data is display with the button “Submit”
- If “Submit” is clicked do the following:
- Collect the name of the Supermaster for this interfero (from DB)
- Collect the path for the interfero (from DB)
- Run the script Interfero_map .sh to process the request.
- The process will reload the page “ InterferoMaps.php ” when the maps are created.

db_connection.php

Comments:
This script is included in function of previous script . It contains the open and close function of Database with all the information such as host, database name, login, password etc …

Interfero_map.sh

Comments:
This script is called by Interfero_master / slave.php . This is an intermediate shell script (run as _www user with limited right) between php script and a full right shell script execution (run as admin_user user).

Dependencies:
- Last command “ exec sudo -u admin_user ${ W_Script_Path }/ Interfero _map2.sh” must be defined in / etc / sudoers .

Arguments:

- Path to interfero binary file

Action:

- Variables declaration
- Execute Interfero_map2.sh under admin_user user.

Interfero_map2.sh

Comments:
This script is called by Interfero_map.sh and will create the interferogram and coherence map in a format to display on an interactive map on the web page

Dependencies:
- Parameter files
- Fiji_RawToJpeg.sh
- Mutt command configured to send email

Arguments:

- Argument 1: ‘ cmdfile.txt ‘ which contain the path of the binary interferogram file of the request.
Action:
- Variables declaration + clean files
- Create working directory + copy inside Interfer o /binary file
- Find corresponding amplitude file and copy it in the working directory
- Create jpeg from binary file (Interfero + coherence) using Fiji_RawToJpeg.sh
- Keep only last 10 working directory + delete working directory if it contains less than 3 files

Fiji_ RawToJpeg .sh
Comments:

- This script will create an image that illustrate a deformation file, masked by a coherence file and stacked on the amplitude file for better spatial representation. The color of each pixels represents the deformation value. These values are represented on a legend. The image is formatted to be displayed on an interactive map on the web page.
Dependencies:

- Fiji (ImageJ)
- gnu sed for more compatibility
- Python + Numpy + script: CreateColorFrame.py
- ${PATHFIJI} need to be declared in . bashrc

Arguments:

- Argument 1 = Amplitude file (average file)
- Argument 2 = Coherence file (mask file)
- Argument 3 = Deformation file (binary file)
- Argument 4 = Output file
- Argument 5 = Parameters file

Action:
- Variables declaration
- Prepare unity, path, temp file, clean working directories
- Format each binary file from to EPSG:3857
- Extract coordinate values and map-size from new hdr file (after conversion)
- Add color frame in binary file
- Build the ImageJscript and create Interfero png file
- Create Coherence png file
- Extract the data from TempFile
- Draw on the colorframe 3 vertical bar and the min/0/max value

CreateColorFrame _Interfero .py

Comments:

- This script is almost the same than CreateColorfarme.py, but is used to create the deformation legend of interferometer file.
So here under are the differences between both scripts:

- Library gdal is used to extract from deformation file (envi file) the geographical paramters o f the file. Then, coordinate s are calculated and written i n a text file (Argument 8) which will be used by php script to display the map on the web page.

- The following variables are calculated differently to display the legend at the top center of the map:
- StartLeft , sq_L and k
- Other variables are different because we use a raw binary amplitude (in CreateColorfarme.py , the amplitude file is already an average and logarithm fct is already performed)
- m ax_AAM , Array_AmpliMod , scale of Min/Max value.

Tool “Delta Deformation”

Introduction
This tool has been developed to allow request from web user to illustrate on a map the difference between 2 deformations files created in MSBAS.
As the other tool, the user can choose the map background style between google reliefs map or satellite view.
On the front of this map, the legend allows to display t he last request (maximum 10). For each request, the acquisition mode, deformation direction and both date s are written in the legend .
The user selects first the deformation direction and satellite acquisition mode if many are available, then the scrolling menu are automatically updated with available dates for this mode.
Finally, the user click the button “calculate differential…” and the process calculates the difference between deformation value (date 1 – date 2) for each pixel, creating a new map in few seconds that will be displayed on the web page.

Scripts hierarchy

Index.php Delta_map .php Delta_map.sh Delta_map2.sh

Delta_map .php (unimode or multimode)

- PHP Variables declaration
- HTML header
- Check if server InSAR is available
- Acquisition mode and orbit selection
- Update Scrolling menu Date1 and Date2
- Date1 and Date2 selection
- Execute shell script to calculate the deformation differential map
- Display maps

Delta_map.sh

Comments:
This script is called by Delta_ map (unimode or multimode). php . This is an intermediate shell script (run as _www user with limited right) between php script and a full right shell script execution (run as admin_user user).

Dependencies:
- Last command “ exec sudo -u admin_user ${ W_Script_Path }/ Delta _map2.sh” must be defined in / etc / sudoers .

Arguments:

- Argument 1 = Date1

- Argument 2 = Date2
- Argument 3 = Acquisition mode
- Argument 4 = Deformation direction

Action:
- Variables declaration
- Execute Delta _map2.sh under admin_user user.

Delta_map2 .sh

Comments:

- This script will pick up the both deformation files selected by the web user. A python script will create a new binary resulting of the difference between these files. Finally, we will convert this binary file into a map to display on the web page.

Dependencies:
- Fiji (ImageJ)
- gnu sed for more compatibility
- shell script: Fiji_RawToJpeg.sh
- Python + Numpy + script: CreateColorFrame.py + Mask_Builder.py + Binary_differential.py
- ${PATHFIJI} need to be declared in . bashrc

Arguments:
- Argument 1 = Amplitude file (average file)
- Argument 2 = Coherence file (mask file)
- Argument 3 = Deformation file (binary file)
- Argument 4 = Output file
- Argument 5 = Parameters file

Action:
- Variables declaration
- Find binary files in InSAR server
- Create working directory and copy into binary + header files
- Create masks for both deformation fil e
- Create jpeg from binary file (Delta-deformation) using Fiji_RawToJpeg.sh
- Keep only last 10 working directory + delete working directory if it contains less than 3 files

Binary_differential .py

Dependencies:
- python 3.8 including Numpy function

Arguments:
- Argument 1 = Binary deformation file 1 (input)

- Argument 2 = Binary deformation file 1 (input)
- Argument 3 = Output file

Action:
- Convert input file into array with numpy
- Subtract array 1 by array 2
- Write result in Argument 3

Tool “ RGB images ”

Introduction

This tool has been developed to overlay 3 images . Each of them colored respectively in Red, Blue and Green. The 3 images are composed of two amplitude images (primary and secondary for a specific pair) and the relative coherence image calculated by MasTer.

As the previous tool, the user can choose the map background style between google reliefs map or satellite view. On the front of this map, the legend allows to display t he last request (maximum 6). For each request, the 3 source images are available and the product RGB .

Two buttons are available if we want to search pairs based on the secondary or the primary amplitude . After this selection, the scrolling menu allow the user to choose the Satellite, Orbit, secondary (or primary) and corresponding primary (or secondary).

Then, a histogram is displayed for each file, and the user must write min/max value used to stretch the colorized image on a specific pixel value (at least to reduce noise).

After submitted the stretch value, the product RGB and 3 source images are available in the legend of dynamic map.

Scripts hierarchy

Scripts Flow

Index.php	RGB_Maps.php	RGB_Slave.php	RGB_map.sh	histogram.php	RGB_map3.sh
		RGB_Master.php	RGB_map2.sh		RGB_map4.sh
			Histogram.py		Run_RGB.py
					Make_coord_from_hdr.py

RGB_ Maps .php

- Display last 6 requested RGB product with their source images.
- Display 2 buttons to perform a new request.
- The rest of the script are static text to display in html, and in javascript mainly the management of the layers and the draggable points in the map display.

Several js library are mandatory as proj4js , leaflet …

RGB _master / slaves .php

Comments:
This script is called by RGB_ Maps.php . It allows the user to choose the dates of primary and secondary amplitude images.
These scripts are based on “ Interfero_master / slave.php ” and they use the same directory “ DB_Interfero ” to manage the scrolling menu. (No need to duplicate this function)

Dependencies:
- parameters file must be present at the same hierarchy of this script.
- “ db_connection.php ” must also be present here.
- Directory ' DB_Interfero ' must be owned by _www user.
- A mysql database must be created with 2 tables.
- Database name = InSAR_MASSPROCESS
- login = InSAR/ your_pwd
- Table1 = Region (ex: VVP)
- Table2 = Region _all (ex: VVP_all)

Action:
- Create function
- Create common variable
- Choose satellite write in ' DB_Interfero /aaPathToPair.txt' the satellite.
- Choose orbit write in ' DB_Interfero /aaPathToPair.txt' the orbit.
- Choose master write in ' DB_Interfero /aaPathToPair.txt' the master.
- Choose slave write in ' DB_Interfero /aaPathToPair.txt' the slave.
- A resume of the request data is display with the button “Submit”
- If “Submit” is clicked do the following:
- Collect the name of the Supermaster for this RGB product
- Collect the path for the interfero (from DB)
- Run the script RGB _map .sh to process the request.
- The process will reload the page “ Histogram .php ” for stretch value (min/ max) selection .

RGB _map.sh

Comments:
This script is called by RGB _master / slave.php . This is an intermediate shell script (run as _www user with limited right) between php script and a full right shell script execution (run as admin_user user).

Dependencies:
- Last command “ exec sudo -u admin_user ${ W_Script_Path }/ RGB _map2.sh” must be defined in / etc / sudoers .

Arguments:

- Path to interfero binary file

Action:
- Variables declaration

- Execute RGB _map2.sh under admin_user user.

RGB _map2.sh

Comments:
This script is called by RGB_map.sh and will create both amplitude and coherence map in a format to display on an interactive map on the web page
The histogram for each file will aslo be created to allow user to enter a manual stretch for ech file.
Dependencies:
- Parameter files
- Histogram.p y

Arguments:
- Argument 1: ‘ cmdfile.txt‘ which contain the path of the binary interferogram file of the request.
Action:
- Create RGB_parameters file if empty or inexistant
- Extract primary and secondary date, and add indice to allow multiple product with same pair of images.
- Create directory and copy 3 source file in it (2 amplitudes + coherence)
- Convert image file from envi to EPSG:3857
- Create the histogram for the 3 converted images
- Write in RGB_parameters the path for each file and indice

Histogram.php

Comments:
This web page is automatically displayed after submitted both dates for RGB overlay image creation. The user must enter stretch min/max value in order to give better nuance on each image. This is used to reduce noise or to give the focus on a specific range of amplitude or coherence value (vegetation, water, flat ground …)

Dependencies:
- RGB_parameters.txt
- RGB_histogram.jpg (created by RGB_map2.sh)

Action:

- Display previous min/max value under histograms.
- Record new value on submit and run RGB_map3.sh

Histogram.py

Dependencies:
- Python + Numpy
- gnu sed for more compatibility

Arguments:
- Argument 1 = amplitude 1 array
- Argument 2 = amplitude 2 array
- Argument 3 = Coherence array
- Argument 4 = Output file

Action:
- Create the histograms of 3 arrays using matplotlib and save the picture in argument 4.

RGB_map3.sh

Comments:
This script is called by Histogram .php . This is an intermediate shell script (run as _www user with limited right) between php script and a full right shell script execution (run as admin_user user).

Dependencies:
- Last command " exec sudo -u admin_user ${ W_Script_Path }/ RGB _map2.sh" must be defined in / etc / sudoers .

Action:
- Variables declaration
- Execute RGB _map 4 .sh under admin_user user.

RGB_map4.sh

Comments:
- This script is called by RGB_map3.sh and will the png image of RGB product to display on an interactive map on the web page
- RGB product is composed of an overlay of 3 images (2 amplitudes file and the relative coherence)

Dependencies:
- Parameter files
- Run_RGB.py

- make_coord_from_hdr.py
Action:

- Extract from RGB_parameters the path of the 3 necessary image files to use for RGB overlay product
- Create RGB file using Run_RGB.py
- Convert from ENVI to EPSG:4326 usin g "_LL" extension
- Write new coordinate in temp1.txt using "make_coord_from_hdr.py" and "_LL" hdr file created before.
- Keep only last 6 working directory + delete working directory if it contains less than 3 files

Run_RGB.py

Comments:
- This script is called by RGB_map4.sh and will create the RGB overlay image from 2 amplitude file and the relative coherence file

Dependencies:
- Specific python environment for this script including spectral package and numpy 1.22.2
- RGB_parameters.txt

Arguments:
- Both amplitude image (Primary + secondary) and coherence image file
- RGB_parameters.txt
- "-crop" is optional, the script will crop the image regarding parameters in RGB_parameters file

Action:
- Initialization of dictionary 'Color' which link image and color (Primary amplitude = red, secondary amplitude = green ...)
- Read parameters from RGB_parameters file
- Prepare all variable (filename) and hdr file
- Modify hdr file for final RGB image
- Copy original binary file locally after replacing nan by 0
- Opening new binary file as an image and create jpeg for each input file .
- Concatenate the 3 arrays in a single numpy array of dimension 3x1 and write results in a binary file
- Opening the new array 3x1, and create rgb jpeg file

make_coord_from_hdr.py

Dependencies:
- Python + Numpy

- gnu sed for more compatibility
Arguments:
- Argument 1 = envi file (binary file)
- Argument 2 = coordinate file (output)

Action:
- Open the amplitude envi file and calculate the 4 corners to use later in php with leaflet library
Create new region

Parameters file

Comments:
The first step to build a web page for a new region is to fill the parameters file. The easiest way is to copy an existing one and replace all the parameters value.
This file is used in several scripts to pick up numerical value , path, IPadress , area name etc … for a specific region.
The syntax is the following:
Value[tab][tab]#[space]Paramterer name (description)
Here under an example:

192.1 11 . 222 . 333 # COMPUTE_SERVER _IP (Ip Address COMPUTE_SERVER)

(!! No space between the value and the hashtag !!)
The value of the parameter is the IP address " 192.1 11 . 222 . 333 ".
The parameter name is " COMPUTE_SERVER _IP " .
In () is any comment to explain the purpose of the parameter .
In several scripts, we will call a function name " GetParam (COMPUTE_SERVER _IP)" that will return the ip address " 192.111.222.333 "

The parameters do have a small explanation to remember their purpose (or the name itself). If a group of parameters are almost exclusive to a few numbers of scripts, a small explanation with script's name is written above the group of parameters. Here under an example for the scripts python which manage the legend of deformation maps:

CreateColorFrame script + Fiji_Amp_Defo_Coh script
40 # Margin
500 # LegendWidth
0.4 # ColorBackgrdLegnd (0 = white --> 1 = grey)
20 # LegendTxtSize (Size of the text in the legend)
80 # LegendHeight (Height of the legend in pixels margin include)
30 # FrameTop (distance between top of color frame and top of the legend)
46 # FrameBott (distance between bottom of color frame and top of the legend)

Initiate_WebPages . sh

Comments:
A script called “ Initiate_WebPages ” has been developed to help creating a new region.
This script requires the “parameters file” as argument and will create all necessary folders and files, give the appropriate right user and owner for each of them. It will create both mysql database and optionally fill the tables.

Prerequisite :

Mysql Ver 8. 0
Parameters file completed

Argument:
- Parameters file
Action:
- Variables and function declaration
- Create all required directory for webpage (in apache web folder “html”)
- Create mounted point on the web server
- Create and manage right and owner of files that will be handle by apache user (_www)
- Create database on mysql server
- Optionally fill the database as first run
- Manage right and owner

Run Mains.sh

When the parameters file is ready and is moved in the region folder “ defo -region” on WEB _SERVER 3 in the html folder of apache2, we can run “Main.sh” with parameters file as argument. Then checked the crontab_log.txt file to verify the process.

Troubleshooting

log files

In order to troubleshoot efficiently in case of issue, the following logfiles are available:

Crontab_log.txt

This logfiles is the output of the main process (Main.sh). It will show exactly what you should see if you ran the script in a command tab. Information here are only the timeframe of each step of the main process. It can be helpful to troubleshoot the last execution and the duration of each step.

Image/Logfile.txt
This file contains the output of almost each command of the main process (Main.sh). It contains full of information and must be used to troubleshoot any error code returns from a function or any commands called in the main script.

Loginterfero1.txt

This file contain s the log of the script " Interfero_map.sh" and can be used to check the command created in php script and send to "Interfero_map2.sh" for execution.

Loginterfero2.txt

This file contains the log of the script "Interfero_map2.sh". It can be used to check the execution of interferometer and coherence map creation.

Delta_Map_log.txt

This file contains the log of the script "Delta_Map2.sh". It can be used to check the execution of Delta map creation.

Log_RGB_ map[1-4] .txt

This file contains the log of the respective scripts " RGB_ map [1-4]. sh ". It can be used to check the tool execution of RGB overlay image .

Common operations
- If script is not run since long time check crontab on WEB _SERVER 3 .
- If the webpages are always in maintenance status, check if STORAGE_SERVER or COMPUTE_SERVER (for custom time series creation) are well mounted on WEB _SERVER 3 .
- If, for any reason, you want to delete all images, time series … by the main process, in order to rebuild everything from scratch, you can do the following action:
- find /Library/Server/Web/Data/Sites/defo- region /images -type f -delete
Be careful, only the files in folder Images can be deleted.

- If time series are shifted on the web page, it means that the number of time series for a same pair of point is not consistent. The previous action of deleting everything in the 'images' folder can help if the issue is on the webserver side. I f mismatches persist, the mismatch is also present in STORAGE_SERVER .

Side Procedure

Satview.tiff and terrain.tif creation

These 2 files are mandatory to display the time series point position on a google earth background or relief background. By default, the map used is the amplitude image but the web page gives the option to switch between 3 maps when clicking on the following button:

The background image must be in the same resolution rate , the same location and the same projection as the sentinel 1 amplitude image. To create this specific map, here is the procedure using QGIS software.

- Open in QGIS a layer 1 with the desired background:

- earth = http://mt0.google.com/vt/lyrs=p&hl=en&x={x}&y={y}&z={z}
- terrain = http://mt0.google.com/vt/lyrs=s&hl=en&x={x}&y={y}&z={z}

- Opening a ‘envi’ file (amplitude, deformation…) of the region as layer2 . We will use this layer 2 as a template to create the desired maps of the region.

- Right-click on layer1 and select ‘Export – >save as’:
- Output mode = Raw data
- Format = Geotiff
- Disable ‘Create VRT’ (untick the box to the right of “Format” box)
- Change the CRS manually as the one of layer2 (You can find it in hdr file of layer2)
- In extend, click on “ L ayer” an d select layer2 (envi file):

This will give the same geographical limit to your map as the one of layer2.

- By default, the resolution of layer1 will be much higher than layer2. To have the same resolution, you can change the resolution factor itself or the layer size regarding the information found in hdr file of layer2.

For example: if we are working with this hdr file:

Samples = 5361
Lines = 4801
Map info = {UTM, 1.0, 1.0, 245000.000000, 6080050.000000, 50.000000 * , 50.000000, 19, South, WGS84, units=Meters}

If we change “layer resolution” in QGIS by value 50 * for vertical and horizontal , the layer size will automatically change to 5361x4801 pixels . In this case, our map will have the same resolution as layer2 (= envi file from amplitude, deformation, …) . To do so, change the value in “Horizontal” and “Vertical” boxes with 50. See the changes in boxes “Columns” and “Rows”.

If we want a higher resolution (especially useful for earth view), we can decrease the Vertical and Horizontal Layer resolution factor in QGIS by an integer (= RateResolutionFactor) . If I enter ‘10’ instead of ‘50’ in the “Horizontal” and “Vertical” boxes , the number of Columns and Rows (see boxes below “Horizontal” and “Vertical”) will be multiplied by 5, giv ing now a layer size of 26805 x 24005 pixels.

This ‘ RateResolutionFactor ’ will be written later in parameters file to be taken into account in different scripts .
Remem b er: you should enter the factor in the “ Horizontal ” and “ Vertical ” boxes, not “ Columns ” and “ Rows ” , otherwise your tiff image will be very small resolution .

- Save your file and move it in the “Document” folder of your webpage with the name satview.tiff and terrain.tiff.
- Convert it as jpg as well using convert: **convert satview.tif satview.jpg**

  (Tiff version is used to display on the web page, while the jpg version is used to create insets in time series LOS plots if needed).

- Write in to the parameters file TS_parameters.txt, the “ RateResolutionFactor ” used to create both maps :

10 # RateResoTerrain (Rate of pixels number in the terrain view compare with envi files)
10 # RateResoSatellite (Rate of pixels number in the satellite view compare with envi files)

To properly assess the value of the RateResolutionFactor to write in the file, divide the size (either the number of lines or columns) of the jpg image by the size of the MSBAS products.

- As explained before, these maps are used to display pair of point used for time series deformation on different maps as below. (For better localization)

link error

If changes are made in the localization of pair of points, the system will detect it automatically and will recreate the appropriate picture with new localization, and will delete the obsolete one.

The maps created above is also used to display the pair of points in specific sub-area as below.

link error

If we change the position of these sub-areas, we would like also to recreate these pictures but this will not be done automatically, and will not be done systematically (waste of time). To recreate these pictures , we can set to ‘ yes’ the two following parameters in parameters file:

No # BuildMapsView (Run script ZoneMaker_2.0 with GoogleMaps View: only if change in areas) [Yes / No]
No # BuildSatView (Run script ZoneMaker_2.0 with GoogleSat View: only if change in areas) [Yes / No]