

Release Notes GAMMA Software, 20201201

Urs Wegmüller, Charles Werner, Christophe Magnard, Andreas Wiesmann,
Othmar Frey, Oliver Cartus
Gamma Remote Sensing AG
Worbstrasse 225, CH-3073 Gümligen
<http://www.gamma-rs.ch>
1-Dec-2020

Introduction

This information is provided to users of the GAMMA software. It is also available online at https://www.gamma-rs.ch/uploads/media/GAMMA_Software_upgrade_information.pdf.

This release of the Gamma software includes new programs that provide new capability, additional features to existing programs and bug fixes.

Gamma Software on Linux, OSX, and Windows

The Gamma software has been compiled and tested on Linux (different distributions), Apple MacOS Catalina (10.15.7) and BigSur (10.16.0), and Windows 10. Computationally intensive programs such as used in co-registration and resampling and geocoding have been parallelized using the OPENMP API built into the GCC compiler. Processing speed on Linux, MacOS, and Windows systems is comparable.

Linux Distribution:

The Gamma software is developed on Ubuntu 18.04 LTS 64-bit Linux and is tested extensively with this distribution. Hence it is highly recommended to run the software on this distribution. There is also a version available for Ubuntu 20.04 LTS.

Announcement: Ubuntu 18.04 LTS will no longer be supported after the mid-2021 upgrade.

Versions of the Software will also be uploaded for RHEL7 based on CentOS7 and RHEL8 based on CentOS8.

For installation instructions for the binary LINUX distributions see the HTML file `INSTALL_linux.html` (provided with the distribution E-mail or found in the main directory of the distribution).

Apple MacOS Distribution:

The software in this version has been compiled using MacOS Catalina (10.15.7) and BigSur (10.16.0). You will need to install libraries such as GDAL using MacPorts. The build uses the MINGW64 GCC 9 compiler on Catalina and the 10.2 compiler on BigSur.

Announcement: The present upgrade is the last upgrade for MacOS Catalina (10.15.7). MacOS BigSur will no longer be supported after the December 2021 upgrade.

For installation instructions for the binary MacOS distributions see the HTML file `INSTALL_macOS.html` (provided with the distribution E-mail or found in the main directory of the distribution).

Windows Distribution:

The Windows 10 version of the Gamma software is compiled with 64-bit support and multi-threaded. The build uses the MINGW64 GCC 10.1 compiler.

For installation instructions for the binary Windows distributions see the HTML file `INSTALL_win64.html` (provided with the distribution E-mail or found in the main directory of the distribution). The `.bashrc` file needs to be updated following the installation instructions.

Documentation and Program List:

The Gamma documentation browser is an HTML based system for viewing the web pages and pdf documents. The documentation browser includes for each module a Contents sidebar on the right side of the screen and a search functionality.

The program `gamma_doc` facilitates the access to the documentation related to a given module or program:

<code>gamma_doc</code>	Opens the main page of the Gamma documentation browser and shows the program list.
<code>gamma_doc DIFF</code>	Opens the DIFF&GEO documentation.
<code>gamma_doc gc_map2</code>	Opens the reference manual web page for <code>gc_map2</code> .

Further information related to the GAMMA Software is available online:

General information:

gamma-rs.ch/uploads/media/GAMMA_Software_information.pdf

Technical reports, conference and journal papers:

gamma-rs.ch/uploads/media/GAMMA_Software_references.pdf

Release notes / upgrade information:

gamma-rs.ch/uploads/media/GAMMA_Software_upgrade_information.pdf

In case the program list is incomplete, run the python script `program_list.py` after successful installation of the Gamma Software in the main folder of the Gamma Software distribution:

```
./program_list.py Gamma_documentation_base.html Gamma_documentation_contents_sidebar.html -a
```

Hardware Recommendations

Using multi-core processors (4 or more cores) will bring substantial improvement in processing speed due to parallelization of the code base. There should be at least 8 GB RAM available for each processor core with 16 GB per core recommended.

Disk storage requirements for using the Gamma Software effectively depend on the amount of input data and data products that will be produced. Based on our experience we recommend to consider at least 16 TB space, especially when working with stacks of Sentinel-1 or very high-resolution data (TerraSAR-X, Cosmo-Skymed) data. The current trend towards larger data products requires substantially increased storage capacities.

GAMMA Software Training Courses

A SAR/INSAR (MSP/ISP/DIFF&GEO/LAT) training at GAMMA (near Bern, Switzerland) and a PSI (IPTA) training are planned for spring and/or fall 2021. See also our web-site under <http://www.gamma-rs.ch/courses/training-courses.html>.

Significant Changes in the Gamma Software Modules since the mid of 2020 Release

DISP Programs

The DISP programs for screen display and raster file generation support the display of data files made. The DISP includes 4 main types of programs:

- 1) *dis...* C-programs supporting the visualization of data files on the screen.
- 2) *ras...* C-programs supporting the generation of rasterfiles (BMP, RAS, TIFF).
- 3) *vis...* Python programs supporting the visualization of data files on the screen and the generation of rasterfiles.
- 4) In addition, some further utility programs are included.

In the end-of-2020 release a substantial update to the *dis...* and *ras...* programs was carried out:

- The command line parameters changed for many *dis...* and *ras...* programs
- A few *dis...* and *ras...* programs do no longer exist or their name changed as we integrated them into fewer programs to offer overall clearer and better structured display tools.
- New functionalities were added to the *dis...* and *ras...* programs, such as the free selection of a color map.
- The default raster image type is now BMP for all operating systems.
- The left-right flipping option in most *ras...* and *dis...* programs has been removed (but is supported by *ras_ras* so that raster files can be flipped).

The changes to the *ras...* programs is relevant to you as you will need to adapt your established processing scripts! Generally, avoiding changes of program names and command line parameters has a high priority in our updating, but this time we considered such a change as necessary.

In the following we describe first the overall structure of the DISP module and then the specific changes to the *dis...* and *ras...* programs and how you can update your scripts.

Overall structure of the DISP module:

dis... and *ras...* C-programs

In many cases corresponding *dis...* and *ras...* programs exist, such as *disdt_pwr* and *rasdt_pwr* or *dis_dB* and *ras_dB*. The main difference between such a pair is that the *dis...* program displays the data on the screen and the *ras...* program generates a rasterfile. Most of their functionality and command line parameters are identical.

A first group of programs supports the visualization or rasterfile generation of a single binary data file. In the case of real numbers, the programs are structured according to the scaling used:

- *dis_linear*, *ras_linear* for linear scaling
- *dis_dB*, *ras_dB* for logarithmic (dB) scaling
- *dispwr*, *raspwr* for power-law scaling

In the case of complex numbers, we distinguish:

- *disSLC*, *rasSLC* to display SLC data
- *dismph*, *rasmph* to display the magnitude as brightness and the phase using a cyclic color scale
- *discpx*, *rascpv* to offer further options as displaying only the real or imaginary part

A second group of programs supports the visualization or rasterfile generation of a two binary data files. In the case of real numbers, the programs are structured according to the scaling used:

- *disdt_pwr, rasdt_pwr* to display a real numbered data file as color using the indicated intensity file as brightness
- *dismph_pwr, rasmp_h_pwr* to display the phase of a complex valued data file as color using the indicated intensity file as brightness

A third group of *dis...* programs (without corresponding *ras...* programs) supports the alternating display of two data sets. Using the central mouse button or the left and right arrows on the keyboard, it is possible to flip between the two images which permits comparison and identification of change or quality differences:

dis2_dB, dis2_linear, dis2dt_pwr, dis2gbyte, dis2mph, dis2mph_pwr, dis2pwr, dis2ras, dis2SLC

Finally, there are a number of further specific display and rasterfile programs included:

- *disras* to visualize a rasterfile on the screen
- *disdem_par* to visualize a DEM and indicate the map coordinates at the cursor position
- *disras_dem_par* to visualize a rasterfile (in map geometry) and indicate the map coordinates at the cursor position
- *dismph_fft* to visualize a complex valued data set and its spectrum (fft) at the cursor location
- *disshd* to visualize the terrain height as shaded relief
- *disgbyte, rasgbyte* to display binary files in the GBYTE format used to store backscatter data in 1 byte
- *disflag, rasflag* to display the phase unwrapping flag file used in the region growing method
- *distree, rastree* to display unwrapped phase and wrapped phase and unwrapping flags
- *ras24_float* to generate an RGB, HLS or HSV composite of 3 input float files
- *ras8_float* to generate an HLS or HSV composite of 2 input float files, saturation indicated on command line
- *ras2float* to convert rasterfile (generated using *ras_linear* or *ras_dB*) to float data file
- *ras2png.py* to convert rasterfile to a png rasterfile supporting transparent option
- *ras2ras* to change color map and to convert 8 to 24-bit color
- *ras_ras* to apply multi-looking, left-right flipping, and conversion of 8 to 24-bit color
- *ras3pwr* to generate a RGB rasterfile based on 3 input float data files
- *ras8_colormap* to generate your own color map
- *ras8_color_scale* to generate a color scale (as separate rasterfile)
- *svg_arrow* to draw arrows on an image and store it as SVG (scalable vector graphics)
- *svg_map* to generate an SVG map
- *svg_poly* to draw a polygon on an image and store it as SVG

vis... Python programs

The *vis...* programs are based on the Python Matplotlib package. The *vis...* programs support both the visualization on the screen as well as the generation of rasterfiles (bmp, ras, tif, jpg, png). The *vis...* display programs initially present an overview of the entire image, if a sub region of the image has not been specified on the command line. Included with the image display is a navigation tool bar that permits, panning, zooming, selection of a rectangular image subregion. It is also possible to cycle through past image views.

In the generation of rasterfiles it is possible to generate “quicklook-like” rasterfiles (e.g. with an indicated maximum dimension and written out as jpg), as well as full resolution rasterfiles similar

to the ones generated with the *ras...* programs. As an option the *vis...* programs support adding a color bar. A horizontal or vertical colorbar can also be generated using the program *vis_colormap_bar.py*.

The following *vis...* programs are included:

- *visbyte.py* to display or generate rasterfile of unsigned byte data
- *viscpx.py* to display or generate rasterfile of complex data (intensity, real, imaginary, magnitude, or phase)
- *visdt_pwr.py* to display or generate rasterfile of float data (e.g. displacement, unwrapped phase, offset, or correlation) combined with intensity taken from a separate input file
- *vismph_pwr.py* to display or generate rasterfile of the phase of FCOMPLEX data combined with intensity
- *vispwr.py* to display or generate rasterfile of float data (backscatter, unwrapped phase, correlation, deformation, ...)
- *visras.py* to display raster images (BMP, EPS, GIF, JPEG, PNG, TIFF, or PPM format)
- *vis_colormap_bar.py* to generate a horizontal or vertical colorbar with scale from colormap file or registered colormap

Further utility programs in the DISP

Format conversion programs:

- *ascii2float, float2ascii* to convert between ascii text files and binary float files
- *double2float, float2double* to convert between binary double files and binary float files
- *gbyte2float, float2gbyte* to convert between binary gbyte files and binary float files
- *short2float, float2short* to convert between binary short files and binary float files
- *uchar2float, float2uchar* to convert between uchar files and binary float files

FLOAT, FCOMPLEX, and Vector data file utility programs:

- *cpd* to copy out segments of FLOAT, FCOMPLEX and SCOMPLEX data files
- *cpx_math* to perform simple arithmetic operations on FCOMPLEX data files
- *cpx_to_real, real_to_cpx* to convert between real valued and complex valued data files
- *fill* to fill gaps in a FLOAT data file using the corresponding values of a second FLOAT data file
- *float_math* to perform simple arithmetic operations on FLOAT data files
- *mapshd* to generate a shaded relief intensity image based on a DEM
- *real_to_vec, vec_to_real* to convert between 3 real valued data files and one 3-dim. vector file
- *vec_math* to perform simple arithmetic operations on vector data files
- *replace_values* to replace values equal to, smaller or larger than a reference value with a new value
- *thres_data* to mask a data file with a thresholding on a float data set

General data file utility programs:

- *cp_data* to copy out data from a binary data set
- *create_array* to generate a binary file with a constant value
- *flip* to do right/left, top/bottom, or right/left and top/bottom flipping of a data set
- *swap_bytes* to do byte swapping of a data set

Parameter file utility programs:

- *get_value* to extract a value of a parameter file
- *set_value* to set a value of a parameter file

Ground control point and polygon selection programs:

- *gcp_2ras* to corresponding points in a pair of raster images
- *gcp_ras* select GCPs using a raster format reference image
- *polyras* to interactively determine a polygon in a displayed raster image

KML utility programs:

- *kml_map* to create KML file with link to an image (e.g. in preparation of the generation of a KMZ file)
- *kml_plan* to generate a KML file containing a polygon corresponding to the scene area and (optionally) a line corresponding to the platform path during the acquisition
- *kml_pt* to create a KML file with icons indicating values from an ASCII table

TIFF/GEOTIFF generation programs

- *data2geotiff* to convert geocoded data with DEM parameter file to GeoTIFF format
- *data2tiff* convert image data to TIFF format

Unwrapping tree editing program

- *tree_edit* to interactively edit a phase unwrapping flag file

What changed? How can I adjust my scripts?

The default raster image type is now BMP for all operating systems. To set the default raster image type to TIFF or SUN raster, add the following to your *.bashrc* (or equivalent) file:

```
export GAMMA_RASTER="TIFF"
or
export GAMMA_RASTER="SUN_RAS"
```

The programs listed in Table 1 were replaced. Executing the previous program name will only print an error message to the screen informing that the program has been removed and indicating the new program that offers the corresponding functionality. With the new program a color map can be indicated (to get the same display as before use the color map indicated in the table). With the new programs you also have a better flexibility using:

- a cyclic or not cyclic color display
- 8 or 24-bit color scales
- different color maps (e.g. *terrain.cm* to display terrain heights, or *hls.cm* to display displacement values)
- a raster file instead of a FLOAT file as background image (now both are possible)

Table 1: Replaced programs

Replaced program(s)	New program(s)	Color map of previous program	Cyclic flag of previous program
<i>discc / rascc</i>	<i>disdt_pwr / rasdt_pwr</i>	<i>cc.cm</i>	0
<i>rasdt_cmap</i>	<i>disdt_pwr / rasdt_pwr</i>		
<i>disdt_pwr24 / rasdt_pwr24</i>	<i>disdt_pwr / rasdt_pwr</i>	<i>rmg.cm</i>	1
<i>dishgt / rashgt</i>	<i>disdt_pwr / rasdt_pwr</i>	<i>rmg.cm</i>	1
<i>rashgt_shd</i>	<i>rasdt_pwr</i>	<i>rmg.cm</i>	1
<i>disrmg / rasrmg</i>	<i>disdt_pwr / rasdt_pwr</i>	<i>rmg.cm</i>	1
<i>dismph_pwr24/rasmph_pwr24</i>	<i>dismph_pwr/rasmph_pwr</i>	<i>rmg.cm</i>	
<i>dismph_pk</i>	<i>dismph</i>	<i>rmg.cm</i>	
<i>dismph_ub</i>	<i>dismph</i>	<i>rmg.cm</i>	
<i>disbyte / rasbyte</i>	<i>dis_linear</i>	<i>gray.cm</i>	0
<i>dis2byte</i>	<i>dis2_linear</i>	<i>gray.cm</i>	
<i>dis2cc</i>	<i>dis2dt_pwr</i>	<i>cc.cm</i>	0
<i>dis2hgt</i>	<i>dis2dt_pwr</i>	<i>rmg.cm</i>	1
<i>dis2rmg</i>	<i>dis2dt_pwr</i>	<i>rmg.cm</i>	1

To identify which command lines of scripts need to be updated, you can search:

- (1) for program names of replaced programs
- (2) for “ras” to identify all *ras** program calls

You can then update the corresponding lines with the new program name and the new command line parameters.

The left-right flipping option in most *ras...* and *dis...* programs has been removed (but is supported by *ras_ras* so that raster files can be flipped).

To show the order of the parameters on the command line, examples of the usage are indicated below for the *disdt_pwr* and *rasdt_pwr* commands (to display the unwrapped phase):

disdt_pwr 20201012_20201018.unw 20201012.mli 2400 1 1800 -9.42 9.42 1 rmg.cm

usage: disdt_pwr <data> <pwr> <width> [start] [nlines] [min] [max] [cflg] [cmap] [scale] [exp] [bits]

input parameters:

data (input) data in *FLOAT* format (deformation, height, unwrapped phase, correlation...)
pwr (input) intensity image coregistered with data (enter - for none, *FLOAT* or raster image)
width number of samples/row of data and *pwr*
start starting line of data and *pwr* file (enter - for default: 1)
nlines number of lines to display (enter - or 0 for default: to end of file)
min minimum data value (enter - for default: 0.0000e+00)
max maximum data value (enter - for default: 1.0000e+00)
cflg cyclic data display flag:
 0: display $\min \leq \text{data} < \max$ (default)
 1: display $(\text{data} - \min) \text{ modulo } (\max - \min)$
cmap colormap file (enter - for default: *hls.cm*)
 NOTE: colormaps are text files in *\$DISP_HOME/cmaps*, examples: *cc.cm*, *rmg.cm*, *hls.cm*, *gray.cm*,
turbo.cm, *BuYIRd.cm*
 colormap swatches: *\$DISP_HOME/cmaps/*.png*, *\$DISP_HOME/cmaps/cmocean/*.jpg*,
\$DISP_HOME/colorcet/.png*
scale intensity display scale factor (enter - for default: 1.0)
exp intensity display exponent (enter - for default: 0.35)
bits bits/pixel (enter - for default)
 8: 8-bit indexed color map (default)
 24: RGB 8-bits/color

rasdt_pwr 20201012_20201018.unw 20201012.mli 2400 1 1800 1 1 -9.42 9.42 1 rmg.cm
20201012_20201018.unw.bmp 1 .35 24

*usage: rasdt_pwr <data> <pwr> <width> [start] [nlines] [pixavx] [pixavy] [min] [max] [cflg] [cmap] [rasf]
 [scale] [exp] [bits]*

input parameters:

data (input) data in *FLOAT* format (deformation, height, unwrapped phase, correlation...)
pwr (input) intensity image coregistered with data (enter - for none, *FLOAT* or raster image)
width number of samples/row of data and *pwr*
start starting line of data and *pwr* (enter - for default: 1)
nlines number of lines to display (enter - or 0 for default: to end of file)
pixavx number of pixels to average across (enter - for default: 1)
pixavy number of pixels to average down (enter - for default: 1)
min minimum data value (enter - for default: 0.0000e+00)
max maximum data value (enter - for default: 1.0000e+00)
cflg cyclic data display flag:
 0: display $\min \leq \text{data} < \max$ (default)
 1: display $(\text{data} - \min) \text{ modulo } (\max - \min)$
cmap colormap file (enter - for default: *hls.cm*)
 NOTE: colormaps are text files in *\$DISP_HOME/cmaps*, examples: *cc.cm*, *rmg.cm*, *hls.cm*, *gray.cm*,
turbo.cm, *BuYIRd.cm*
 colormap swatches: *\$DISP_HOME/cmaps/*.png*, *\$DISP_HOME/cmaps/cmocean/*.jpg*,
\$DISP_HOME/colorcet/.png*
rasf (output) image filename, extension determines the format, enter - for default: **.ras*
 **.bmp* BMP format
 **.ras* Sun raster format
 **.tif* TIFF format
scale intensity display scale factor (enter - for default: 1.0)
exp intensity display exponent (enter - for default: 0.35)
bits bits/pixel:
 8: 8-bit indexed color map (default)
 24: RGB 8-bits/color

IPTA display tools (pdisdt_pwr, prasdt_pwr, pdismph_pwr, prasmph_pwr, dis_data_pt)

New programs were implemented in the IPTA module to replace the previously available visualization scripts. Table 2 lists the replaced scripts and the corresponding new programs. The new programs are faster and avoid conflicts caused by the generation of a temporary file. They all work with 24-bit color tables. They include the following new features:

- support for point lists in both slant-range / azimuth and map coordinates
- support for both float and raster image as background image
- when clicking on the image, the index and value(s) of the nearest point are displayed
- choice of cyclic or non-cyclic display for FLOAT data
- free choice of colormap
- additional options for point and background image aspect
- the programs for alternating display of two point-datasets now support use of a background image

Table 2: Replaced IPTA display programs

Replaced programs	New program
<i>pdisdt_pwr24 / pdisdt_pwr24_map</i>	<i>pdisdt_pwr</i>
<i>pdismph_pwr24 / pdismph_pwr24_map</i>	<i>pdismph_pwr</i>
<i>pdis2dt / pdis2dt_map</i>	<i>pdis2dt_pwr</i>
<i>pdis2mph / pdis2mph_map</i>	<i>pdis2mph_pwr</i>
<i>prasdt_pwr24 / prasdt_pwr24_map</i>	<i>prasdt_pwr</i>
<i>prasmph_pwr24 / prasmph_pwr24_map</i>	<i>prasmph_pwr</i>

In addition, the program *ras_data_pt* was modified to include the features listed below, of which some were previously available in *rasdt_cmap_pt*. As a consequence, *rasdt_cmap_pt* was removed from the software.

New features in *ras_data_pt*:

- support for point lists in both slant-range / azimuth and map coordinates
- choice of cyclic or non-cyclic display
- free choice of colormap
- additional options for point and background image aspect

Radio Frequency Interference (RFI) filtering in SLC data (SLC_RFI_filt)

RFI is sometimes present in SAR data. Often the RFI is caused by a narrow band source. Considering the power-spectrum, e.g. using *dismph_fft*, the RFI results in many cases in a strong spike. The new ISP program *SLC_RFI_filt* analysis the power-spectrum of SLC data, identifies spikes, masks the spike in the frequency domain, and mitigates the RFI in the SLC by applying the inverse fft to the masked power spectrum.

SLC_RFI_filt was successfully applied to PALSAR and Sentinel-1 data (see also the related new demo example and Figure 1).

Matlab wrapper

Matlab is a widely-used interactive development environment and programming language. Many researchers and engineers, including among Gamma Software users, use Matlab for processing data, analyzing results and generating figures.

We now offer the possibility to integrate the Gamma Software into Matlab through a wrapper composed of the *mat_gamma* and *par_file* classes. It permits a smooth usage of the Gamma Software within an interactive use of Matlab as well as within Matlab scripts.

Using the wrapper:

- Binary images, point lists and data, parameter files, tab files, can be easily read, inspected, and written.
- Gamma Software program calls become Matlab function calls where variables can be used as function arguments, and system outputs can be stored in variables or written to log files.
- Usage within an interactive Matlab environment permits function name search and automatic completion.
- The documentation for each function can be easily accessed.

The Matlab wrapper can also be used in GNU Octave, a free software alternative mostly compatible with Matlab.

Installation

The Gamma Software installation has to be completed first.

The Gamma Software Matlab wrapper consists of two files: *mat_gamma_base.m* and *par_file.m* located in the main directory of the Gamma Software distribution. Hence, to be able to use the wrapper, the main directory of the Gamma Software distribution has to be added to the Matlab path. Note that this path has to be updated after each release.

Once the path has been set, run the following command to build up the *mat_gamma.m* class file:

```
mg = mat_gamma_base();
```

This only has to be run once after each software release.

Then, to access the methods included in *mat_gamma* class, run the following command:

```
mg = mat_gamma();
```

This command has to be run at the beginning of each Matlab session or in case of Matlab scripts, at the beginning of the script.

Matlab wrapper contents

The *mat_gamma* class includes the following elements:

- Gamma Software programs as functions: each Gamma Software program or script becomes a Matlab function with the same name. The mandatory and optional arguments described in the Gamma Software program usage have to be entered in the same order in the Matlab function, with the arguments being separated by commas or as a character array (or a mix). In addition, keyword arguments are available in the Matlab function for piping inputs and writing outputs to files, variables, and to the Matlab command window.

- A *gamma_doc* function for easily accessing the HTML-based Gamma Software documentation and displaying it in a Web browser.
- A *run_cmd* function for running commands in the shell that are not in the Gamma Software.
- A *which* function for locating files in the Gamma Software and all directories defined by the environment variable *PATH*.
- Functions for reading, updating, and writing binary images.
- Functions for reading and writing IPTA point lists and point data, as well as tab files.
- A set of variables used throughout the *mat_gamma* class that may be useful when writing scripts.

The *par_file* class includes the following elements:

- Parameters stored in “containers.Map” objects (keyword – value).
- Functions for reading, modifying, and writing Gamma Software parameter files.

Additional documentation

A user's guide is available as a PDF document: User's Guide to Gamma Software integration in Matlab / Octave

A demo example has been prepared and is available for download (see section Gamma Software Demo examples).

Gamma Software Demo examples

In this period again some Gamma Software Demo examples were added/modified. Their access is limited to Gamma Software users with a valid license. The access information is provided with the software delivery.

Note that as a consequence of the changes in the display program, all the demo examples will be checked and updated accordingly. This process will extend into the beginning of 2021.

New / modified demo example:	Contents
Gamma_demo_RFI_filtering.tar.gz	Demo showing how to apply Radio Frequency Interference (RFI) filtering to C-band (Sentinel-1) and L-band (PALSAR) SLC data using band-pass filtering (<i>bpf</i> or <i>SLC_RFI_filt</i>).
mat_gamma_demo.tar.gz	Demonstration how the Gamma Software can be used in a Matlab environment.
Gamma_demo_PALSAR1_SBAS_Yibal.tar.gz	SBAS demo example showing a time-series analysis process using multi-look PALSAR-1 DInSAR phases of a multi-reference stack to monitor the subsidence over an oilfield in Oman (Fig. 2).
IPTA_demo_S1_Yibal_PSI.tar.gz	PSI demo example showing a time-series analysis process using single-look Sentinel-1 DInSAR phases of a single-reference stack to monitor the subsidence over an oilfield in Oman (Fig. 3).

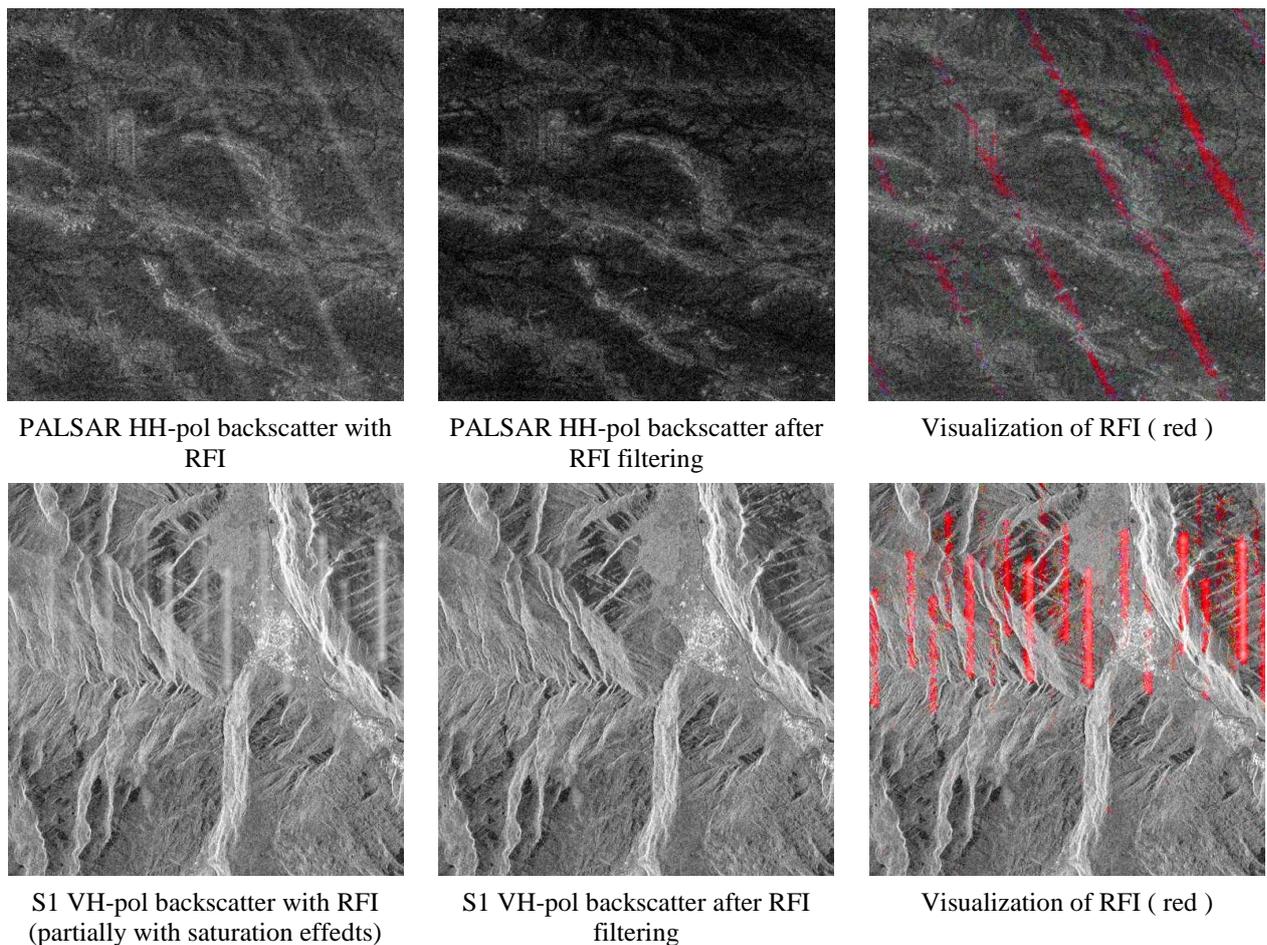
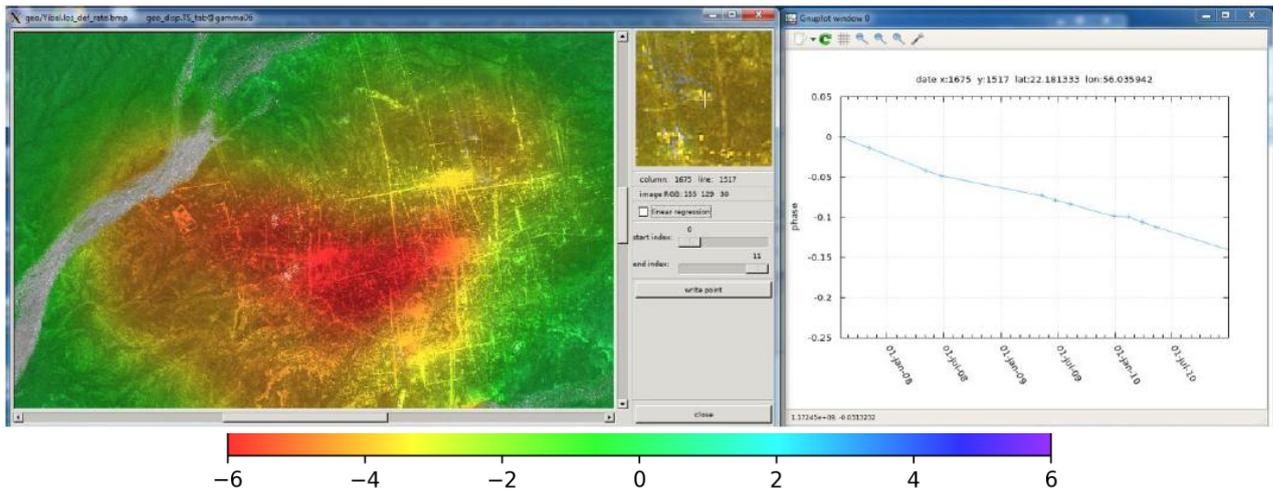
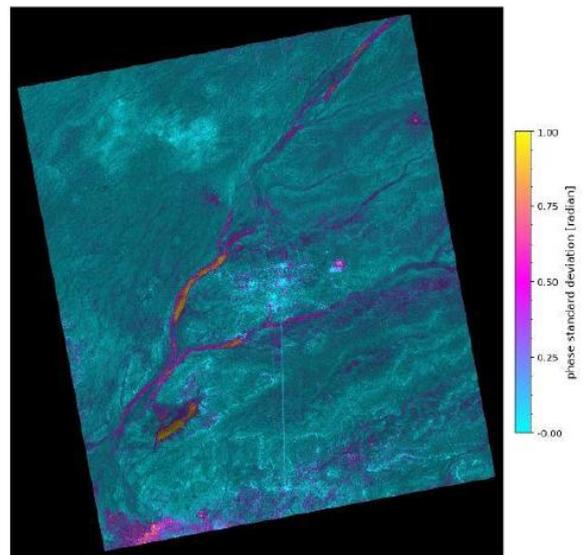


Figure 1 RFI filtering demo results at L-band (PALSAR over Oman) and C-band (Sentinel-1 IWS data over Switzerland).



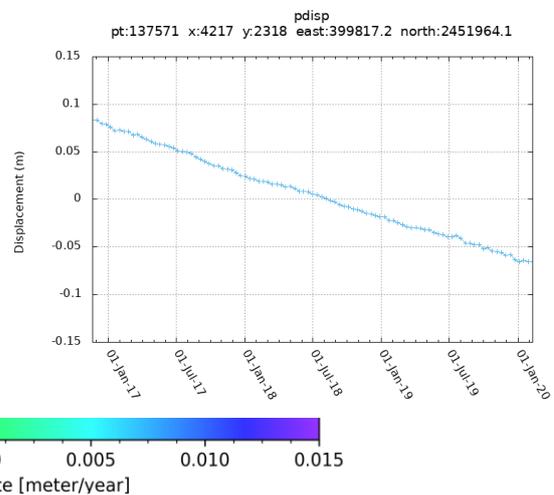
Displaying average deformation rate and LOS deformation time series using *vu_disp2d*.



Visualization of deformation rate as kmz in Google Earth.

Visualization of phase standard deviation. High values indicate high phase noise which means low quality.

Figure 2 PALSAR SBAS-style displacement time series demo results over an oil field in Oman.



LOS deformation rate [meter/year]

Figure 3 Sentinel-1 PSI-style displacement time series demo results over an oil field in Oman.

MSP

-

ISP

S1_path_number: New script to determine S1 path (track) number based on the zipfile name.

par_RSAT2_SG: The program was modified. It now produces MLI binary data and parameter files in slant range geometry and supports SCF data in addition to SGF and SGX data.

typedef_ISP.h, *ISP_io*: Added the following parameters to the Gamma Portable Radar Interferometer (GPRI) data structure (*GPRI_par*): *GPRI_RAW_par*, *TX_RX_SEQ*, *RX_channel*, *RX_port*, *TX_port*, to better support measurements of a polarimetric GPRI.

ERS_ASF_SLC *ERS_ESA_PRI* *ERS_ESA_SLC*: These scripts were used for reading Exabyte tapes. Removed, since the same data are now more easily downloaded from ESA.

DIFF&GEO

sub_phase: Updated. Input data files can now either be *FCOMPLEX*, *SCOMPLEX*, or *FLOAT* format. New is the ability to subtract the phase of a complex image from another complex image.

pixel_area: Added calculation of the *sig2gam_ratio* that permits direct scaling of ellipsoid corrected sigma0 data (such as usually available from Sentinel-1 after importing the data) into terrain-corrected gamma0 data. Increase default value of *nstep* to 16 to reduce sampling error. Added test for when the incidence angle is close to 90 degrees and so that $\cos(\text{inc})$ is never less than 0.01. This prevents negative values for gamma0. The input layover-shadow map is now mandatory.

DISP

ras_cpt.c, *ras_cpt_scale*: These programs have been deleted and replaced by *rasdt_pwr* and *vis_colormap_bar.py*. Scripts have been added to inform users of the change.

short2float.c *float2short*: Now also support unsigned short data type.

rasdt_pwr, *disdt_pwr*: Added functionality to programs to permit selection of colormap, specify cyclic or fixed range color, and 8 or 24 bits/sample. The input background intensity can be provided either from a float data file or a raster image Supports displaying data either between min and max or wrapping the data into the range between in and max before display (corresponds to using a cyclic or non-cyclic color scale). Left-right flipping option was removed.

rasmph_pwr, *dismph_pwr*: Added functionality to programs to permit selection of colormap, specify cyclic or fixed range color, and 8 or 24 bits/sample. The input background intensity can be provided either from a float data file or a raster image Supports displaying the phase of the complex values using a cyclic color map with a 2π cycle. Left-right flipping option was removed.

dis_dB, *dis_linear*, *dispwr*, *disSLC*: Updated programs that display 1 data file only. Programs support selection of the data colormap (except for *disSLC*). *dis_linear* supports displaying data either between min and max or wrapping the data into the range between in and max before display (corresponds to using a cyclic or non-cyclic color scale).

ras_dB, *ras_linear*, *raspwr*, *rasSLC*: Updated programs that display 1 data file only. Programs support selection of the data colormap (except for *disSLC*). *ras_linear* supports displaying data

either between min and max or wrapping the data into the range between in and max before display (corresponds to using a cyclic or non-cyclic color scale).

dis2hgt, *dis2rmg*, *dis2cc*: These programs have been deleted and replaced by *dis2dt_pwr*. Scripts have been added to inform users of the change.

dis2SLC, *dis2_dB*, *dis2_linear*, *dis2mph*, *dis2pwr*: Updated programs that support alternating display of 2 data files. Programs support selection of the data colormap (except *dis2SLC*). *dis2_linear* supports displaying data either between min and max or wrapping the data into the range between min and max before display. *dis2mph* displays both the phase and the power-law scaled intensity of both complex data sets. Programs support specifying *xoff* and *yoff* for the second data set relative to the first one. *dis2pwr* and *dis2mph* support specifying a level for image scaling instead of using the image average intensity. New possibility to select one or the other image was added: using the keyboard left and right arrows.

dis2dt_pwr, *dis2mph_pwr*: Added programs to display two real or complex valued data sets on top of an intensity background. Programs support selection of the data colormap. The input background intensity can be provided either from a float data file or a raster image. *dis2dt_pwr* supports displaying data either between min and max or wrapping the data into the range between in and max before display (corresponds to using a cyclic or non-cyclic color scale).

rascpv, *discpv*: Added programs to display real, imaginary, phase or intensity of FCOMPLEX or SCOMPLEX numbers. The intensity can be displayed using either logarithmic or power-law scaling.

cpv_math: Added absolute value $\text{abs}(d1)$ operation.

dis_dB, *ras_dB*, *dispwr*, *raspwr*, *dis_linear*, *ras_linear*: Added functionality to permit selection of colormaps. In *dis_dB*, *ras_dB* absolute scaling or relative scaling (relative to average value of data set) can now be selected.

LAT

ras_ras: Update program to remove support for logarithmic scaled raster files. Moved the program to the DISP.

sigma2gamma: Added test for incidence angles > 90 that result in negative gamma values.

IPTA

IPTA_users_guide.pdf: The IPTA users guide has been substantially updated by adding a full SBAS style (multilook phases, multi-reference stack) procedure description.

ccs_pt, *sigma_pt*: Modified. If *np_max* not specified, use all the points in the search region, otherwise take $\text{MIN}(np_max, nw)$.

atm_mod_2d_pt, *atm_sim_2d_pt*: Added new programs to calculate a 2D grid of atmospheric model parameters for a set of interferograms. The model parameters determined in *atm_mod_2d_pt* are used by *atm_sim_2d_pt* to calculate the model phase for all points and records. This update supports the spatially adaptive atmospheric path delay estimation for the IPTA vector data format in a similar way as *atm_mod_2d* and *atm_sim_2d*.

pt2data, *pt2data_inpaint*: Now also support point data in map coordinates.

ccs_pt, *cct_sp_pt*, *sigma_pt*: Corrected to correctly sort list of points within the specified radius when *np_max* is specified. Corrected error in handling of FLOAT data that are 0.0 (no_data). The

FCOMPLEX representation is now (0.0,0.0) rather than (1.0, 0.0) so that these are correctly recognized as no_data.

mb: Added smoothing constraint2 for the first and last deformation velocity based on a penalty for rapid change: $\text{cost} = -2*v[0] + v[1] + v[2]$ and $\text{cost} = -2*v[-1] + v[-2] + v[-3]$. This tends to smooth the first and last velocities to be in better agreement with their temporal neighbors (to reduce high deviations at the beginning and end of the time series where the solution is less constraint).

pdisdt_pwr, *prasdtd_pwr*: New programs to display point data (FLOAT) with a linear scaling using a specified colormap combined with intensity background. Support displaying and raster image generation in both slant range and map geometry. Replace the programs *pdisdt_pwr24*, *pdisdt_pwr24_map* *prasdtd_pwr24*, and *prasdtd_pwr24_map*.

pdismp_pwr, *prasmph_pwr*: New programs to display phase from FCOMPLEX point data using a specified colormap combined with intensity background. Support displaying and raster image generation in both slant range and map geometry. Replace the programs *pdismp_pwr24*, *pdismp_pwr24_map*, *prasmph_pwr24* and *prasmph_pwr24_map*.

pdis2dt_pwr: New program to display 2 point-data sets (FLOAT) with a linear scaling using a specified colormap combined with intensity background. Replaces the programs *pdis2dt* and *pdis2dt_map*.

pdis2mp_pwr: New program to display phase from 2 FCOMPLEX point data sets using a specified colormap combined with intensity background. Replaces the programs *pdis2mp* and *pdis2mp_map*.

IPTA_users_guide.pdf: Updated version of the IPTA users guide. Includes now a full description of an SBAS type process (using multi-look phase, using a multi-reference stack, deriving a non-uniform deformation time series).

mk_2d_im, *mk_2d_im_geo*: Scripts were removed, use *ras_data_pt* instead.

mk_diff_all *mk_int_all*: Deprecated scripts were removed.

Python wrapper

-

Matlab wrapper

mat_gamma_base.m, *par_file.m*: The Gamma Software is integrated into Matlab through a wrapper: the ***mat_gamma*** and ***par_file*** classes. It permits a smooth usage of the Gamma Software within an interactive use of Matlab as well as within Matlab scripts.

- Binary images, point lists and data, parameter files, tab files, can be easily read, inspected, and written.
- Gamma Software program calls become Matlab function calls where variables can be used as function arguments, and system outputs can be stored in variables or written to log files.
- Usage within the interactive Matlab environment permits function name search and automatic completion.
- The documentation for each function can be easily accessed.

The Matlab wrapper can also be used in GNU Octave, a free software alternative mostly compatible with Matlab.