Release Notes GAMMA Software, 20210701

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-Jul-2021

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 (10.16.1), 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 20.04 LTS 64-bit Linux and is tested extensively with this distribution. Hence it is highly recommended to run the software on this distribution.

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 (11.2). You will need to install libraries such as GDAL using MacPorts. The build uses the GCC 9 compiler on Catalina and the GCC 10 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 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). Notice that installing the latest GAMMA_LOCAL_w64 version is mandatory because a new GCC compiler and new libraries were used to build the software. Furthermore, 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:

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

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
```

Python and Matlab wrappers

The Gamma Software is integrated into Python and Matlab through wrappers.

The *py_gamma* Python module permits a smooth usage of the Gamma Software within Python scripts as well as within a Python Interactive Development Environment (IDE) such as Spyder or PyCharm or using Jupyter Notebooks.

In the same way, the Matlab (and Octave) wrapper, composed of *mat_gamma* and *par_file* classes, permits a smooth usage of the Gamma Software within an interactive use of Matlab as well as within Matlab scripts.

Hardware Recommendations

Using multi-core processors (6 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 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 End of 2020 Release

ICEYE Support – Assessment of interferometric time-series analysis

Since the last software release, we received two multi-temporal stacks acquired by ICEYE X6 with one day repeat intervals. We used a stack over Mohave USA, to test SBAS-like multi-temporal interferometric procedures using a multi-reference stack of multi-look differential interferometric phases and a stack over the Japanese City Ichinomiya to assess the possibility of using Persistent Scatterer Interferometry (PSI).

ICEYE DInSAR and a SBAS results over Mohave, USA

The multi-temporal backscatter and coherence show a good potential for land cover applications (landcover classification, parameter retrieval) – without having seriously investigated and assessed these. Using the backscatter and coherence an RGB composite of the average coherence, the average backscatter and the backscatter temporal variability was calculated for the Mohave data Figure 2). A very large number of differential interferograms was calculated. The main phase components are orbital phase, topographic phase, atmospheric, deformation phase, and phase noise. The orbital and topographic phase can well be modeled. Terrain height corrections retrieved (relative to the Copernicus 1 arc second DEM) were generally small and mainly related to the higher spatial resolution of the data (as compared to the reference DEM) and some temporal changes of the terrain height (e.g. in a landfill area). Figure 1 shows the comparison of the Copernicus 1 arc second DEM and corrected DEM after the updating using the ICEYE data. Having a large stack available permits of course to average deformation rate or height estimates over many pairs and thereby reducing the errors caused by the atmospheric phase. For one section deformation values up to about 2cm over the interval of less than two months covered by the 51 scenes could be retrieved (Figure 3). For this a multi-reference stack using multi-look DInSAR phases was used (using the program *mb*).

SLC spotlight mode scenes acquired by X6 on 37 dates between 20210310 and 20210419 were used for this assessment (range_pixel_spacing: 0.42m, azimuth_pixel_spacing: 0.19m). Examples of differential interferograms are shown in Figure 4. One strength of PSI is that it is also applicable for stacks including long spatial baselines. For this single-pixel phase values need to be used. For point like scatterers the phase remains coherent also for very long spatial baselines. Different PSI tests were conducted. Below some results of the test using the last 17 scenes with a multi-reference stack (related time-baseline plot is shown in Figure 5) are shown. In this stack baselines up to 500 m were included. Including also longer time intervals (e.g., connecting scene 1 with scene 17) resulted in a nice 2-dimensional net without strong correlation between time intervals and baselines.

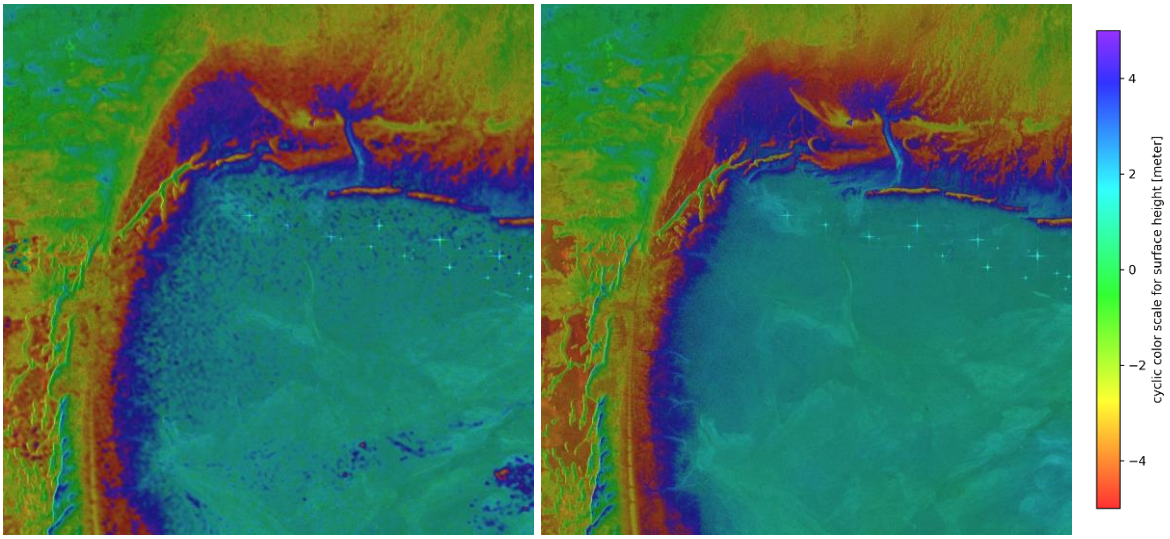


Figure 1 Oversampled 1-arc-second Copernicus DEM height (left) and ICEYE based updated DEM for a 5.6 km x 5.6 km area (section 1) including parts of a salt lake area with corner reflectors. A fine cyclic color scale was used to make difference between the two height maps better visible.

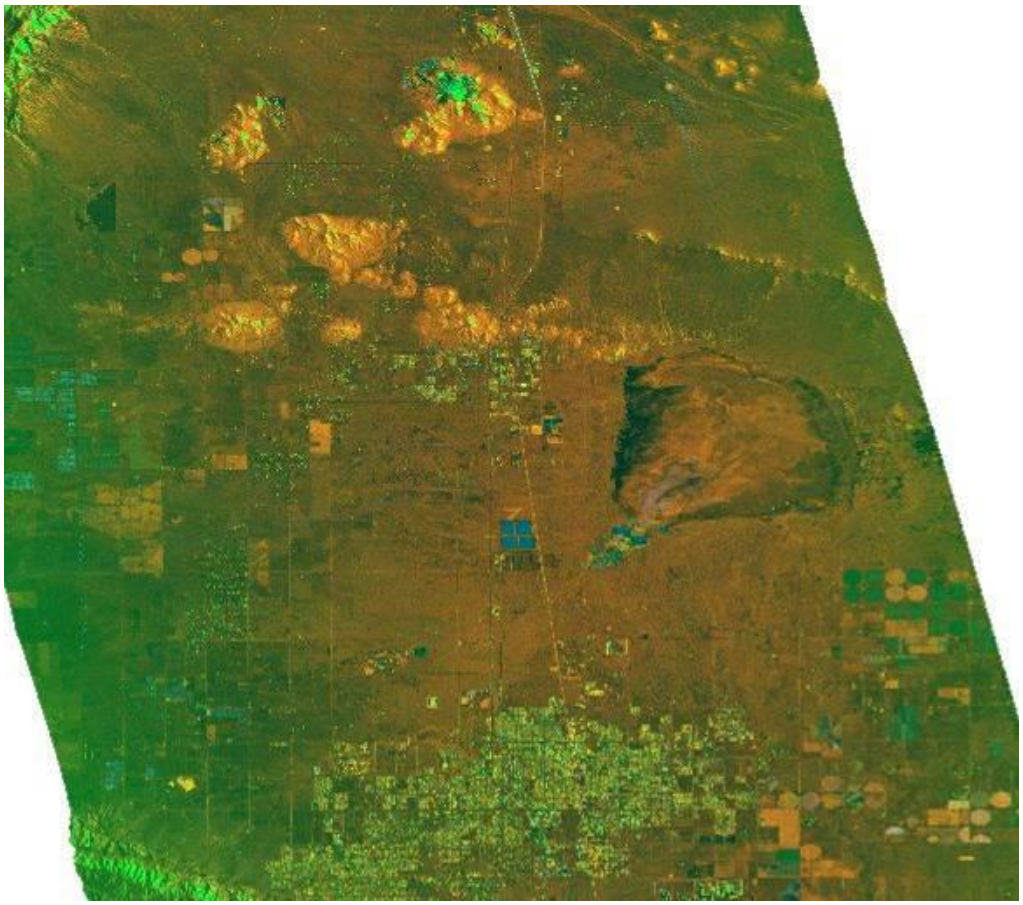


Figure 2 RGB composite of average coherence (red channel, linear scaling), the average backscatter (green, log scaling), and the backscatter temporal variability (blue, log scaling).

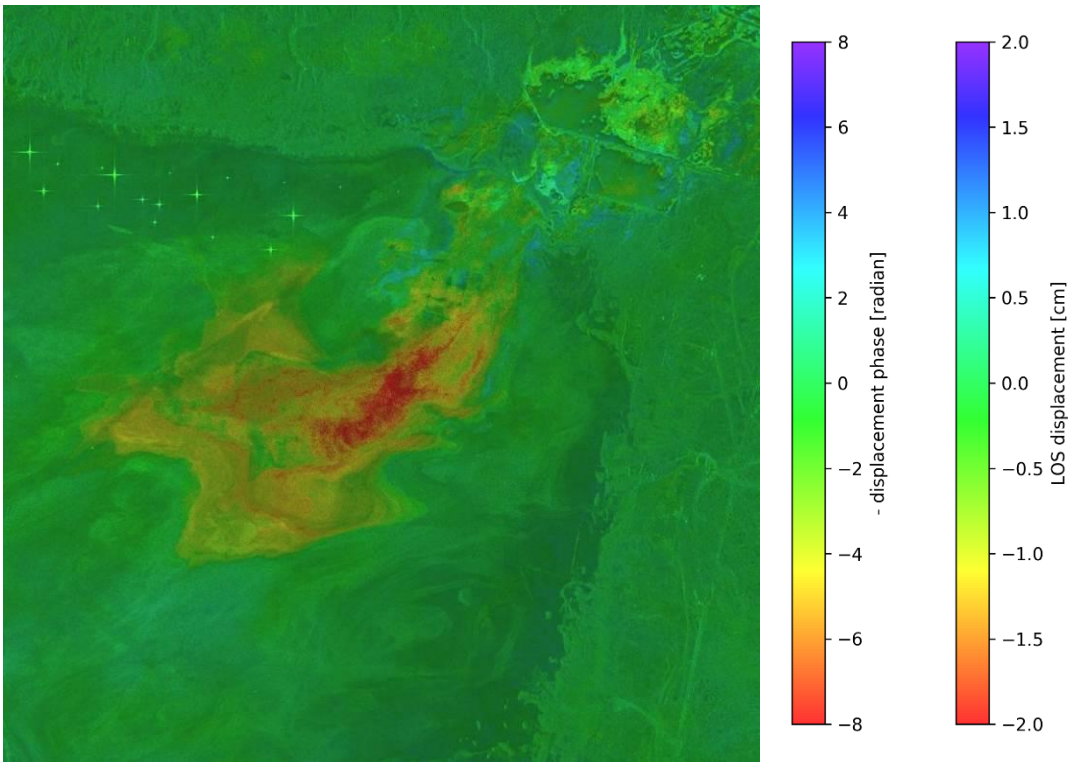
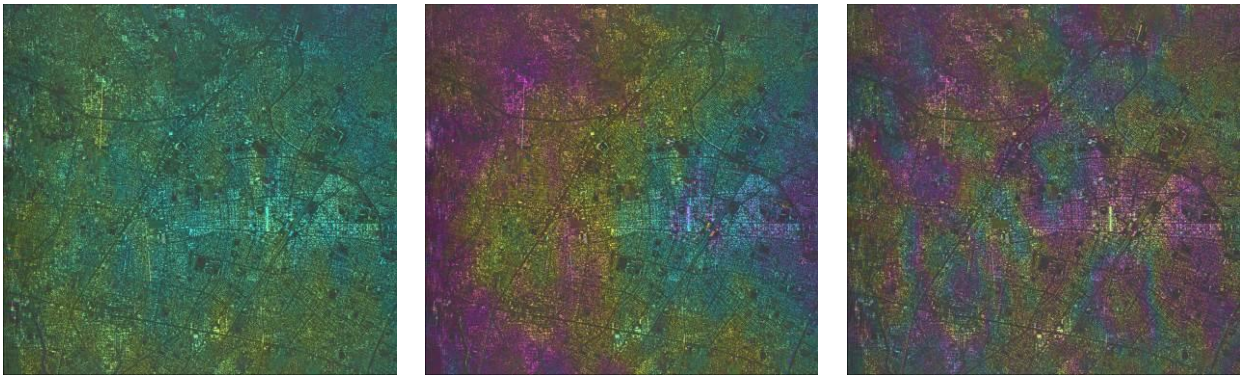


Figure 3 In the salt lake area of the Mojave site deformation values up to about 2cm over the interval of less than two months covered by the 51 scenes could be retrieved. Negative LOS displacements correspond to a displacement away from the satellite.



20210410_20210411, dt=1d, B⊥=-24m | 20210410_20210412, dt=1d, B⊥=-97m | 20210410_20210413, dt=1d, B⊥=-215m

Figure 4 Examples of ICEYE differential interferograms.

ICEYE PSI results over Ichinomiya, Japan

All the technical steps (e.g. geocoding, co-registration, interferogram generation) worked well. The PSI result provides for the selected point like scatterers high quality topographic heights (shown in Figure 6) and deformation rates. The calculated height estimation error was between 0.1m and 0.2m. Thanks to the accurate point heights the geocoding makes the point locations match very well with features as buildings, traffic lights, etc. The heights could not be validated in detail but considering the estimated height in Google Earth shows that the location and spatial variation correspond well to the image context. The estimated deformation rate is not very useful in this case. Because of the short overall time span covered by the data and the high stability of the area the estimation error seems to be larger than the expected deformation rates.

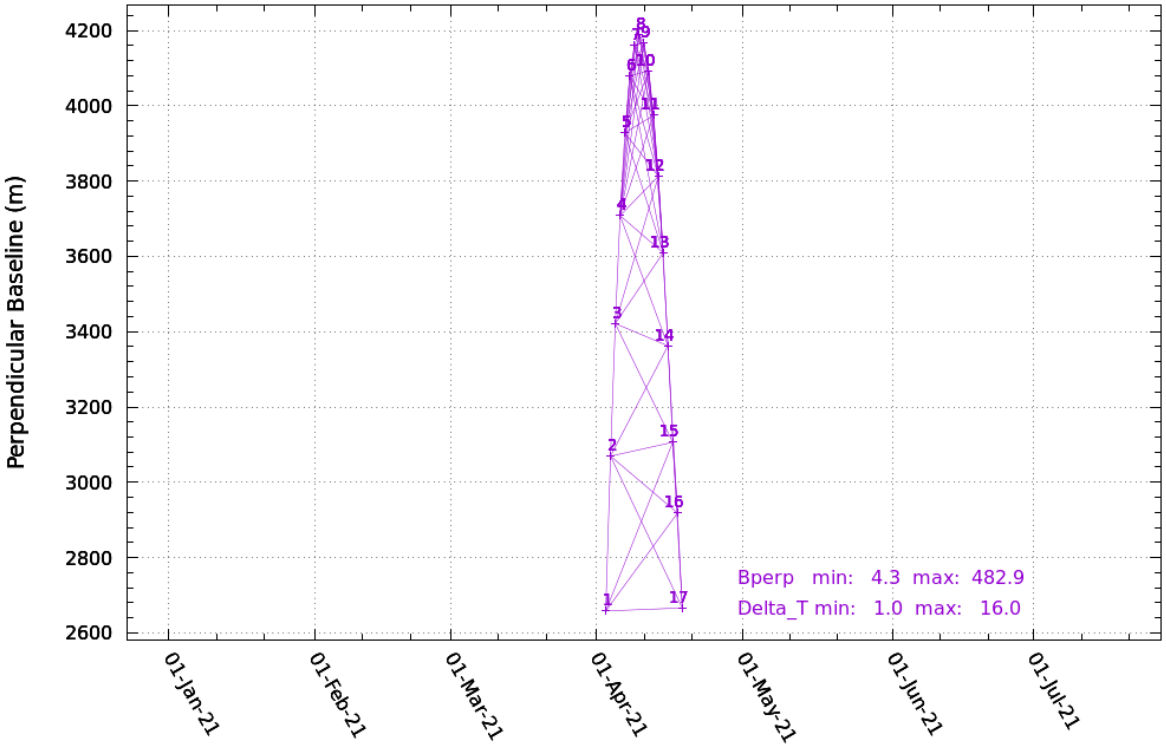


Figure 5 Time-baseline plot for multi-reference stack for the scenes between 20210403 to 20210419.

Figure 6 ICEYE PSI based terrain heights visualized in Google Earth.

Capella SLC and geocoded data

Capella Space Corp. builds and operates a constellation of X-band SAR satellites providing high-resolution SAR images. Currently, three operational satellites have been launched and additional satellites should follow soon.

Capella currently provides data acquired using three different modes: stripmap, sliding spotlight, and spotlight. Spotlight images have a very high spatial resolution of 0.5 x 0.5 m, while sliding spotlight and stripmap data cover larger areas at slightly coarser resolutions. A spotlight image from Capella-Open-Data Collection of Venezia, Italy, is shown in Figure 7.

Capella-Open-Data Collection is a set of freely accessible Capella data under a CC BY 4.0 license. It includes approx. 60 datasets stored as SLCs, terrain-geocoded and ellipsoid-geocoded images. They are spread all over the Earth and show a large variety of natural and man-made landscapes, acquired under the various modes provided by Capella SAR satellites.

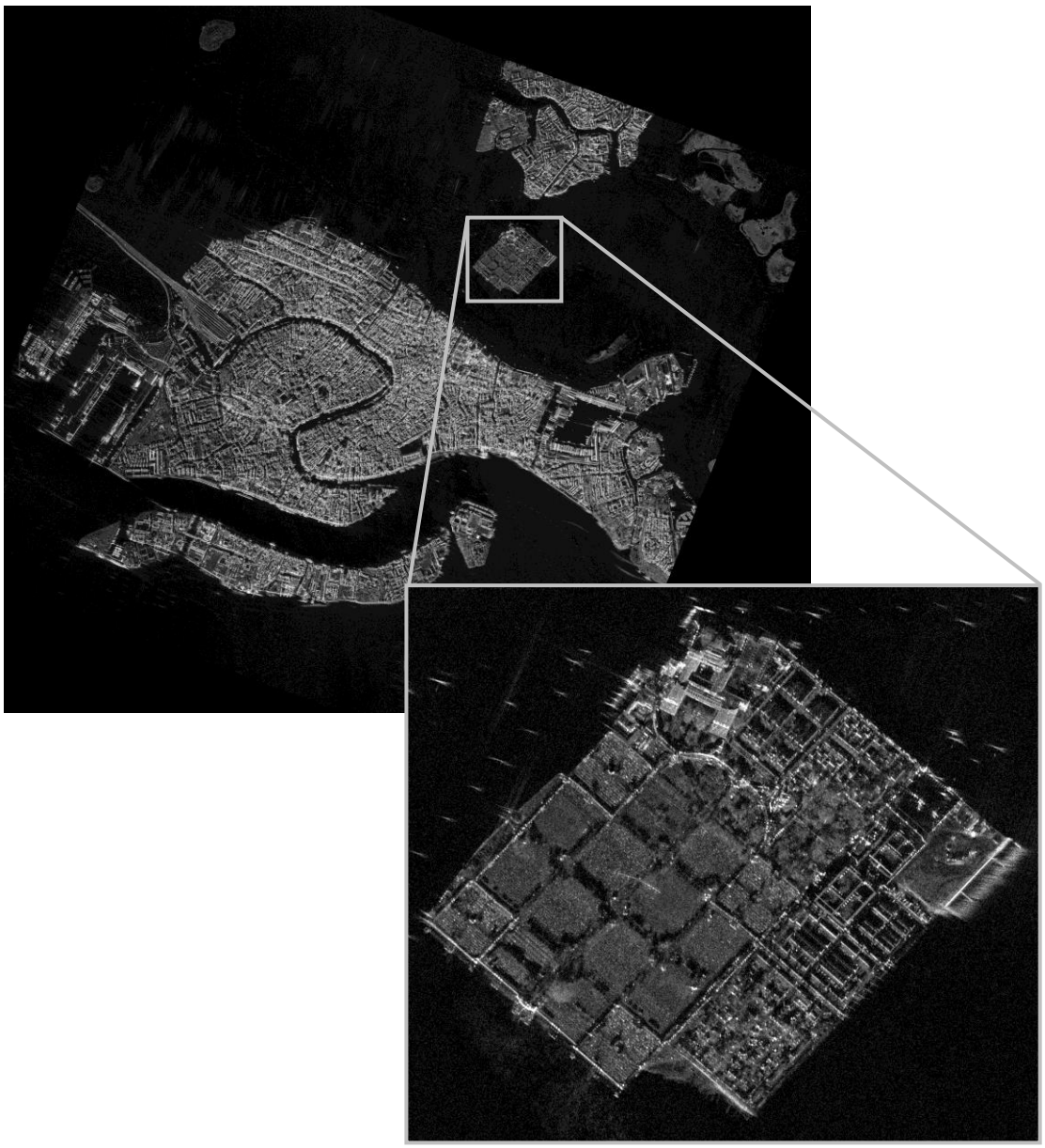


Figure 7 Capella spotlight image of Venezia, Italy. The island of San Michele is enlarged. Image © Capella Space Corp, All Rights Reserved.

People interested in having access to the whole Capella-Open-Data Collection or to Capella commercial products are welcome to register on the following webpage:
https://console.capellaspace.com/user/login/

The GAMMA programs *par_Capella_SLC* and *par_Capella_geo* were added to support the reading Capella X-band SAR SLC and geocoded (both terrain- and ellipsoid-geocoded) data products.

A demo example was added to demonstrate how to read Capella data. The demo also includes a comparison between data acquired using different acquisition modes. Data used in the demo are from the Capella-Open-Data collection.

NovaSAR-1 SRD data

In addition to NovaSAR-1 SLC (single look complex) and GRD (ground range detected) data, we now also support NovaSAR-1 SRD data (slant range detected). Hence, the following GAMMA programs are used to read NovaSAR-1 data:

- SLC data: *par_NovaSAR_SLC*
- GRD data: *par_NovaSAR_GRD*
- SRD data: *par_NovaSAR_SRD*

Burst selection for Sentinel-1 and ScanSAR data

Sub-swath and burst selection for Sentinel-1 has been updated and is now supported by the script *S1_BURST_tab_from_zipfile.py*. The previous script *S1_BURST_tab_from_zipfile* now is simply a wrapper that calls *S1_BURST_tab_from_zipfile.py*. It is strongly recommended to directly call the new script, though calling *S1_BURST_tab_from_zipfile* will also work and provide the same results as before.

S1_BURST_tab_from_zipfile.py creates text files (*burst_number_table* and *BURST_tab* files) that describe which sub-swaths and which bursts will be read from one ZIP file or from a list of ZIP files. A list of ZIP files may only include consecutive images in the same orbit (same date).

Four modes are supported:

1. Generate *burst_number_table* file for reference
2. Generate *burst_number_table*, *burst_number_table_aoi*, and *BURST_tab* files for reference using a KML file for burst selection
3. Generate *burst_number_table* file for reference, and *burst_number_table* and *BURST_tab* files for list of ZIP files
4. Generate *burst_number_table* and *BURST_tab* files for list of ZIP files using reference *burst_number_table*

Hence a convenient way to read Sentinel-1 data is as follows:

1. Generate a KML file of a polygon defining the area of interest. This KML file can be generated either using a program such as Google Earth or using the new tool *poly2kml*.
2. Generate lists of ZIP files for each date of your time-series data. It can be done as follows, e.g., for data acquired on 23-Jun-2021: *ls *20210623*.zip > 20210623.zip_list*
3. Use option 2 in *S1_BURST_tab_from_zipfile.py* to generate the *burst_number_table_aoi* file corresponding to the area of interest for the reference data.

4. Convert the Sentinel-1 data to the format used in the Gamma Software using *S1_import_SLC_from_zipfiles* (for reference and secondary data) using the reference *burst_number_table_aoi*

Several ways to read Sentinel-1 data using *S1_BURST_tab_from_zipfile.py* are detailed in the demo example ***Gamma_demo_S1_burst_number***.

In addition, for any ScanSAR and TOPS data, a *BURST_tab* file listing the bursts that intersect a polygon defined in a KML file can be generated using the script *ScanSAR_BURST_tab_poly.py*. Running *SLC_copy_ScanSAR* with this *BURST_tab* file will generate burst SLC data using the bursts intersecting the area defined in the polygon(s).

The new tool *poly2kml* is complemented by the tool *kml2poly* doing the opposite transformation.

Resampling data using resamp_image_par

A new high quality and very flexible resampling tool named *resamp_image_par* was added to the Gamma Software. It permits resampling 2D data from the geometry defined in a first parameter file to the geometry defined in a second image parameter file. The parameter files should relate to the same scene, resampling data from one acquisition geometry to another acquisition geometry is not supported (i.e., different date / tracks / ...). The program supports SLC/MLI, DEM, DIFF and offset parameter files (i.e., *.par, *.dem_par, *.diff_par, and *.off_par files). The first and second parameter files must be of the same type.

There are numerous possible applications, among which the following can be listed. The program can be used to resample:

- a multi-looked image to the single-look geometry
- a multi-looked image to a different number of looks
- an offset field to its related image size (e.g., offset field to MLI geometry) or in the reverse direction, i.e., from the related image to the offset geometry
- an offset field to a differently spaced offset field
- a simulated interferogram in multi-look geometry to the single look geometry
- a strongly multi-looked interferogram to a “less” multi-looked geometry (useful for unwrapping the phase)

The program includes an optional low-pass filter (Gaussian filter) when down-sampling, to avoid aliasing.

Additional information is available in the HTML-based documentation.

Generation of KML files for point data

A convenient way to visualize point data is by generating a KML file and visualizing it in a program such as Google Earth. However, such KML files quickly become too heavy when a large number of points has to be displayed. To support large number of points, they have to be separated in tiles with different level-of-detail. This way, when looking from far away, only a subset of the points is visible, while when getting very close, all the points for the narrow field of view will be displayed.

KML generation was already supported by the program *kml_pt* (in the DISP package), which uses a text file containing the point data as input to generate the KML file. It makes it very flexible, as

any kind of value can be used for the color scale and for the values shown in the "balloon" when clicking on a point. *kml_pt* was modified to generate tiled KMLs as follows:

- The KML file name specified by the user is the main KML file.
- A subdirectory contains all KML tiles: the subdirectory name can be specified by the user; its default name is "kml"

A new program called *kml_ts_pt* was added to the IPTA package:

- *kml_ts_pt* supports generation of KML and KMZ files for visualizing point deformation time-series on Google Earth.
- The inputs are the files typically used in IPTA processing (contrary to a text file in *kml_pt*).
- It also generates tiled files with different level-of-detail depending on the viewpoint distance, in order to support large numbers of points.
- The output can be either a KML file and its associated subdirectory containing the KML tiles, or a KMZ file (KML files zipped in a single file). The output name extension defines the file type.
- A color bar is automatically generated; the field used for the color scale can be specified by the user.
- Time-series plots of the displacement data can also be added to the "balloon" shown when clicking on a point.

A new script called *kml_plist.py* was added to the IPTA package:

- The script supports generation of KML and KMZ files for visualizing point lists on Google Earth.
- It calls *kml_ts_pt* using a simplified set of parameters and options (simple usage).

A result of *kml_ts_pt* for the Aletsch IPTA demo is shown in Figure 8.

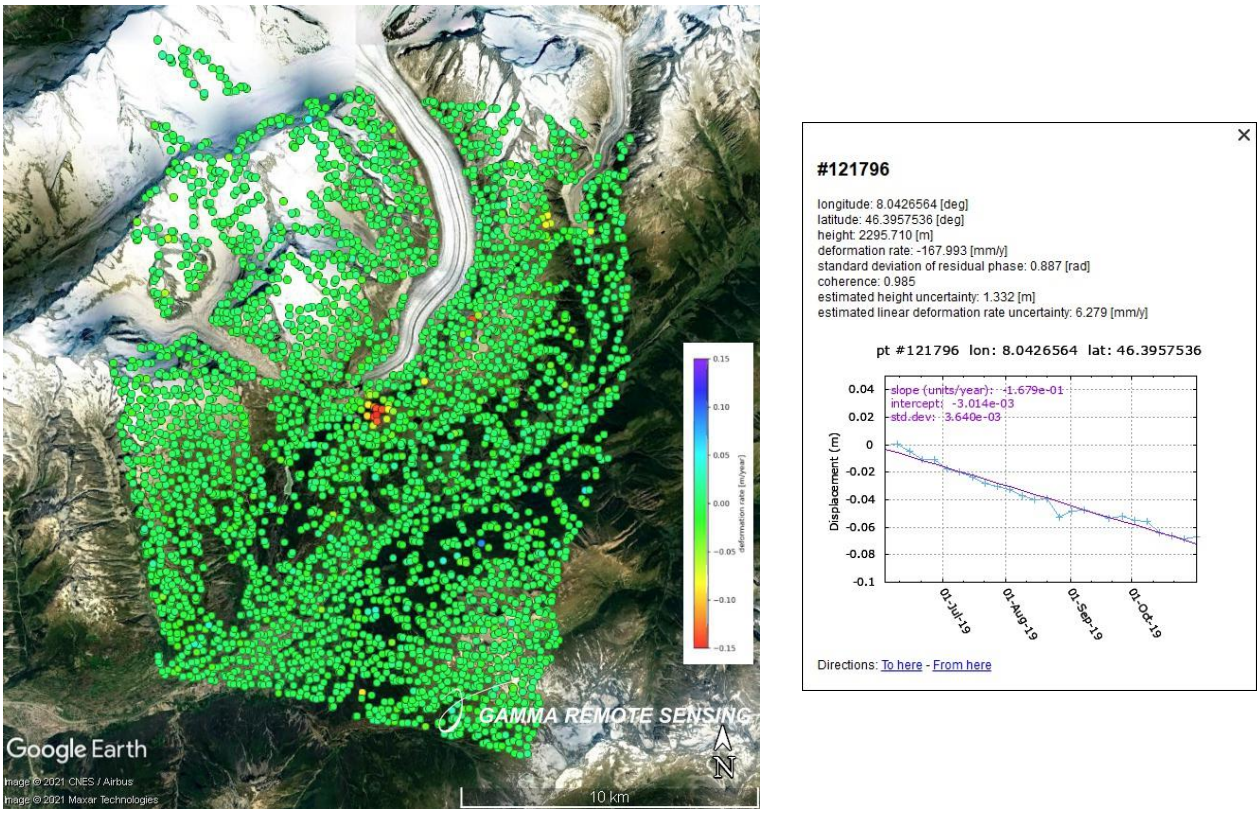


Figure 8 Left: Visualization of the deformation rate for Aletsch demo example. Only the top-most level-of-detail is visible when looking at the whole are. Right: Example of a "balloon" for one of the points that includes the deformation time-series plot.

IPTA programs

The maximum number of points supported by IPTA programs used to be limited to (a fraction of) the maximum size of a 32-bit integer, i.e., on the order of 10^9 points. While this was typically more than enough for most IPTA processing, it was inching toward being a limitation for processing large areas. Hence the IPTA programs were all reviewed, and the maximum number of points now depends on the size of a 64-bit integer, i.e., on the order of 10^{18} points.

The adaptation of IPTA programs to support map geometry in addition to SAR geometry went ahead, with now nearly half of the programs supporting both geometries. This process will continue in the next releases.

The following programs were parallelized to speed-up processing: *fspf_pt, spf_pt, ts_rate_pt*

Deramping ScanSAR and TOPS data

Deramping of ScanSAR and TOPS data (Sentinel-1) is now performed using the scripts *ScanSAR_deramp_reference.py* and *ScanSAR_deramp_2nd.py*. They replace the scripts *S1_deramp_TOPS_reference* and *S1_deramp_TOPS_slave*, respectively.

Visualization Programs

An autoscaling option is now available through the *cflg* command line parameter in following programs:

> *disdt_pwr, dis2dt_pwr, rasdt_pwr, dis_linear, dis2_linear, ras_linear, pdisdt_pwr, pdis2dt_pwr, prasdt_pwr*

This option scales the image between the minimum and maximum values found in the data.

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 in the previous release, all the demo examples were checked and updated accordingly.

New / modified demo example:	Contents
Gamma_demo_Capella.tar.gz	Demo example on the commands for reading and using Capella SAR data products acquired using various modes and provided as SLCs and geocoded products (see Figures 7, 9, and 10).
Gamma_demo_radcal.tar.gz	This demo example shows how the topography-related backscattering component can be removed from SAR images to provide radiometrically "flattened" images. The process relies on precise geocoding and on the calculation of the illuminated area contributing to each individual pixel in slant-range geometry using a digital elevation model (DEM) (see article [1] and Figures 11 and 12). [1] O. Frey, M. Santoro, C. L. Werner and U. Wegmüller, "DEM-Based SAR Pixel-Area Estimation for Enhanced Geocoding Refinement and Radiometric Normalization," *in IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 1, pp. 48-52, Jan. 2013, doi: 10.1109/LGRS.2012.2192093.
Gamma_demo_S1_burst_number.tar.gz	Demonstrates the identification of Sentinel-1 burst numbers in S1 IWS or EWS SLC data ZIP-files and the preparation of BURST_tab files (for indicated zipfiles) for selection of bursts: - corresponding to an indicated reference (zipfile) - corresponding to an indicated burst_number_table - covering an area of interest specified in a KML file This demo is very useful for learning how to efficiently read and import S1 data covering a specific area-of-interest for time-series processing.

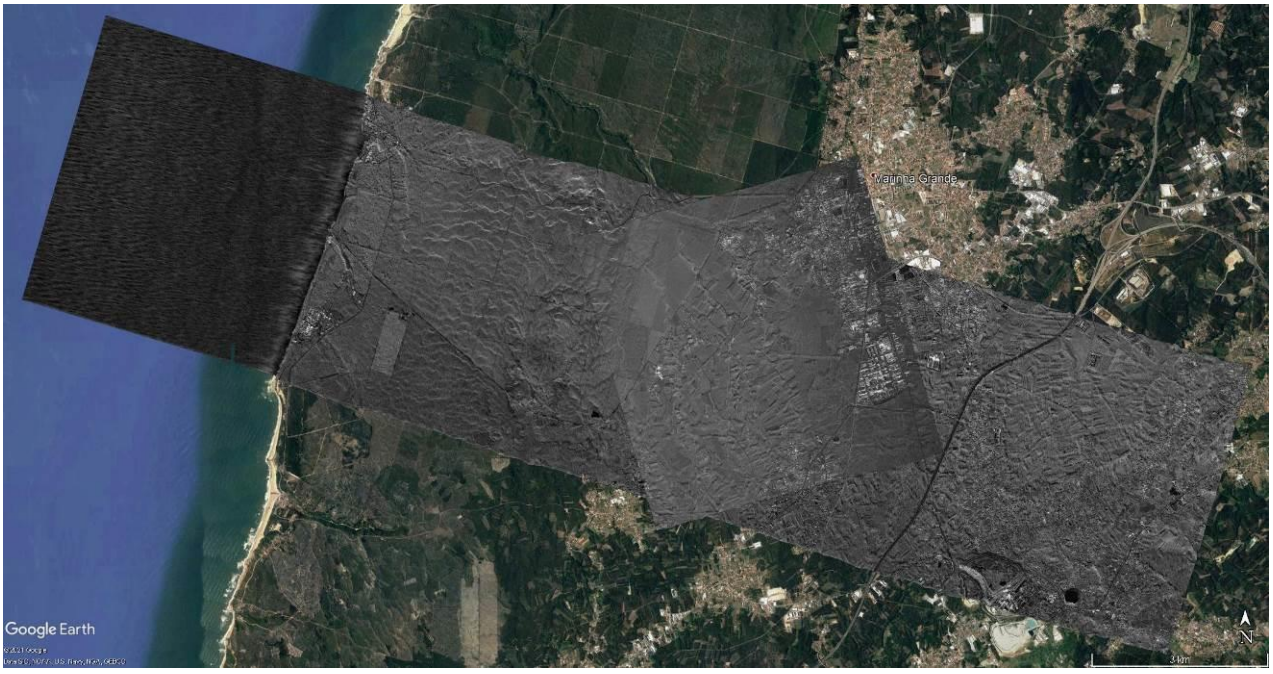


Figure 9 Stripmap and spotlight images acquired over Marinha Grande, Portugal. We notice the larger coverage of the stripmap image. The images were acquired using Capella-2 satellite, which is on a 45° orbit (i.e., non-polar). This results in an uncommon image orientation for spaceborne SAR images. SAR images © Capella Space Corp, All Rights Reserved.

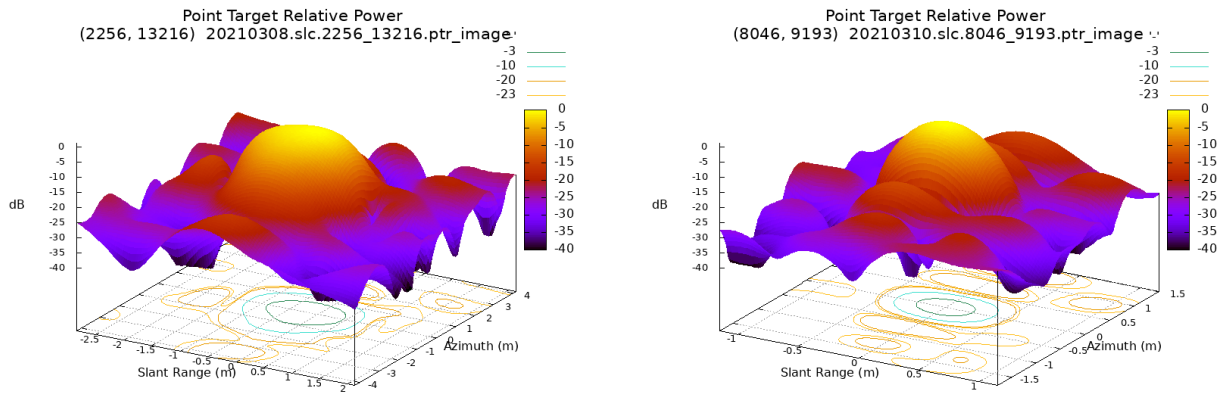
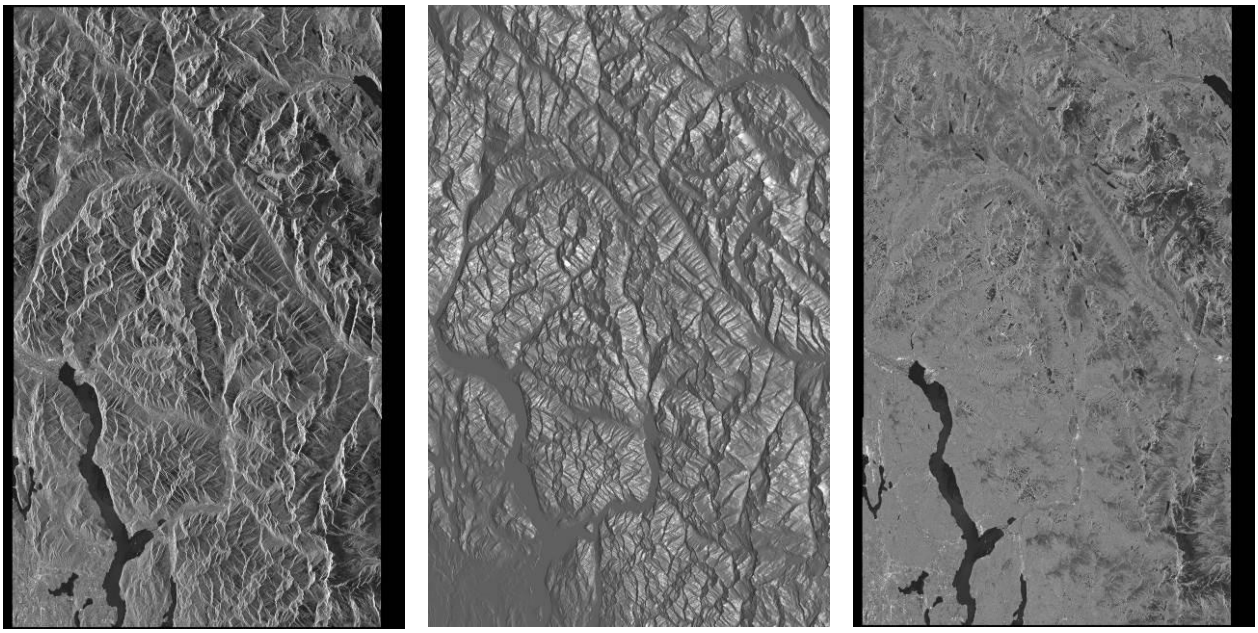


Figure 10 Point target analysis for strong scatterers on the stripmap (left) and spotlight (right) images shown in Figure 9. The measured 3 dB resolutions (slant range, azimuth) are (0.922 m, 1.457 m) for the point target on the stripmap image and (0.474 m, 0.43 m) for the point target on the spotlight image.



S1 image with ellipsoid-based sigma0 radiometric calibration

Image of the radiometric calibration factor applied to convert the image on the left to the image on the right

S1 image with terrain-based gamma0 radiometric calibration

Figure 11 Removal of topography-related backscattering component on a Sentinel-1 image acquired over Swiss and Italian alps.

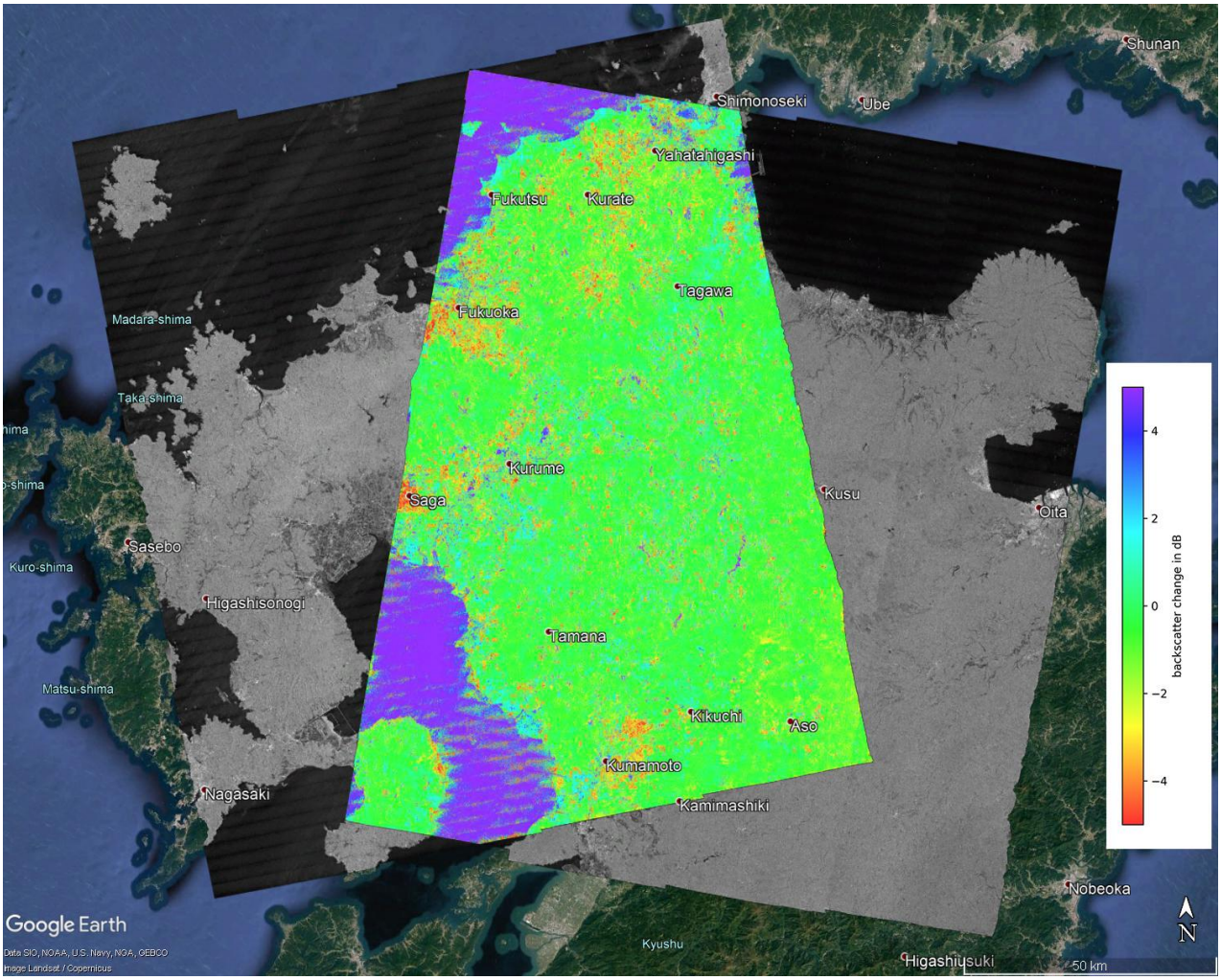


Figure 12 Ascending and descending RCM cross-pol (HV) ScanSAR images acquired during and after floods in Kyushu, Japan. The topography-related backscattering component was removed from both images, enabling their comparison for change detection in spite of their different acquisition geometry. The intersecting area shows the ratio between both images. While many changes are due to different scattering mechanisms in built-up areas or issues due to layover/shadow areas, we can observe some actual change in several locations, e.g., where flooded areas in the 1st acquisition that are not flooded anymore in the 2nd one. The RCM data were made available in the framework of The International Charter Space and Major Disasters for emergency management.

MSP

-

ISP

par_Capella_SLC: New program for generating SLC parameter and image files for Capella SLC data.

par_NovaSAR_SRD: New program for generating MLI parameter and image files for NovaSAR SRD data.

ScanSAR_BURST_tab_poly.py: New script to generate output *BURST_tab* defining sub-swaths and bursts intersecting polygon(s) defined in a KML file.

S1_BURST_tab_from_zipfile.py: New script to generate S1 burst_number_table and BURST_tab files to support burst selection. Replaces *S1_BURST_tab_from_zipfile* and adds the possibility to use polygon(s) defined in a KML file to select bursts in an area of interest.

ScanSAR_deramp_reference.py: New script to deramp ScanSAR and TOPS burst SLC data. Replaces *S1_deramp_TOPS_reference*, which was previously located in the DIFF module.

ScanSAR_deramp_2nd.py: New script to deramp a co-registered 2nd burst SLC using the azimuth phase ramp determined for the reference burst SLC. Replaces *S1_deramp_TOPS_slave*, which was previously located in the DIFF module.

SR_to_GRD, GRD_to_SR: Corrected calculation of start time (sr_t0) and azimuth line time (sr_tazi) when an image in MLI geometry is specified using the MLI parameter file. The azimuth line time already takes into account the decimation in the azimuth direction.

typedef_ISP.h, ISP_io: Added the float parameter *TX_atten_dB* to the *GPRI_PAR* structure (used when processing data acquired with a Gamma Portable Radar Interferometer (GPRI). Added reading and writing this parameter in *ISP_io*. The parameter sets the level of the CHUPA attenuator for the GPRI-II transmitter.

par_S1_SLC, par_S1_GRD: Look-up tables which do not cover the full range distance of the bursts are now completed using linear extrapolation based on the 2 last valid samples.

image_stat: Now also computes the minimum, maximum and median values.

S1_import_SLC_from_zipfiles: The phase correction applied to IW1 swath burst SLCs acquired before an ESA S1 processor change in March 2015 now depends on the IPF and not on the date.

multi_look_ScanSAR, ScanSAR_burst_MLI: Added a new scale factor option for the output MLI. The default scale factor is calculated from the calibration gain in the SLC parameter file.

SLC_copy: Corrected calculation of the heading angle with the GPRI-II. The azimuth heading angle is now defined as being the heading at the homerun position (azimuth angle = 0).

par_ASF_PRI: Update radiometric calibration, data now cropped to remove blank values at near and far range, title now automatically copied from CEOS leader file.

ISP_io: Corrected screen display of burst_win and ext_burst_win parameters so that they lineup.

par_S1_SLC: Added test for burst offset delta == 0. When the offset is 0 there is no need to copy any of the burst parameters. Furthermore, *strcpy* does not permit overlapping copy.

DIFF&GEO

gc_GPRI_map: The C program was replaced by a script. The new script keeps the same usage as before but internally uses *gc_map2* and *lk_vec_lt*.

S1_deramp_TOPS_reference: Program was replaced by *ScanSAR_deramp_reference.py* and moved to the *ISP* module.

S1_deramp_TOPS_slave: Program was replaced by *ScanSAR_deramp_2nd.py* and moved to the *ISP* module.

gp_geo_fit.py: Program to optimize range distance and azimuth start angle of GPRI data based on a DEM.

dem_import: Added new option to select reading horizontally or vertically in GeoTIFF / GDAL supported raster format. Reading vertically can be useful in some cases for VRT DEMs that include only one image in vertical direction but many images in horizontal direction.

par_data_geo: Added new [priority] option. An existing *DEM_par* file can be used as input by setting the priority flag to 0. Setting the priority flag to 0 while no *DEM_par* file exists will trigger an error message. If the priority flag is set to 1, a *DEM_par* file will be generated using information from the GeoTIFF / GDAL supported geocoded raster data, any existing *DEM_par* file with the same name as specified in the command will be overwritten.

resamp_image_par: Added new program to resample 2D data with geometry of the input image parameter file to the geometry of a second image parameter file of the same scene.

DISP

DISP_lib: The default raster type on Linux was changed from SUN raster to BMP. BMP is now the default raster type for all OS.

disdt_pwr, dis2dt_pwr, rasdt_pwr, dis_linear, dis2_linear, ras_linear: An autoscaling option is now available through the *cflg* command line parameter.

poly2kml: New program to generate a KML file from a polygon drawn on a georeferenced image.

data2geogiff: Added new option *COGflg* to generate Cloud Optimized GeoTIFF tiling and changed the compression algorithm used from PACKBITS to LZW.

kml2poly: New program to generate a polygon file from a KML file and a DEM/MAP parameter file.

visbyte.py, viscpx.py, visdt_pwr.py, vismph.py, vispwr.py, visras.py: Updated programs for more consistent visualization and raster image production. The image size defined using "-z" option is now working as expected, image size now maximized within the figure, improvement of the colorbar position and size, improvement of text size and position displaying values of clicked pixel.

visdt_pwr.py: Modified, so that it accepts now 0 as a valid value for short integer input data

data2geotiff: Can now also write UNSIGNED SHORT integer data (uint16).

flip: Can now also flip INTEGER, DOUBLE, and SUN/BMP/TIFF format raster images.

kml_pt: Now generates tiled KML files, in order to display large number of points on Google Earth. Added new options for defining the icon size and the subdirectory containing the KML tiles. This permits visualizing larger point data sets more conveniently in google Earth.

LAT

-

IPTA

ts_rate_pt: Parallelized using OpenMP and introduced block processing to limit memory use.

pdisdt_pwr, pdis2dt_pwr, prasdt_pwr: Added an autoscaling option through the cflg command line parameter.

vu_disp, gras8_disp, gras24_disp: Modified to check if time-series data are valid and if there is a point in the search region before writing a data point file. Corrected check if trend-line can be calculated

disp_prt_2d: New program to generate point displacement histories in text format based on 2D data (e.g., when working with *mb*).

sigma_pt: An SLC or DEM parameter file is now mandatory. The usage has been adapted accordingly and now includes the <par> input parameter in third position. Also modified to support point data in map coordinates.

fspf_unw_pt: Now also supports point data in map coordinates. Filtering and unwrapping in map coordinates can be very helpful for data acquired over mountainous areas / areas with steep topography.

83 IPTA programs: Update to support larger number of points: point indices and point counters now use 64-bit integers.

msk_pt: Added new mode option that adds an inverted behavior of the raster mask file: when setting mode option to 1 (contrary to 0 for default behavior), areas set to 0 in the mask are considered valid while non-zero areas in the raster mask are masked in the output products.

cct_pt, ccs_pt, cct_sp_pt: Now also support point data in map coordinates.

data2pt, pt_density, pt_density_reduction: Now also support point data in map coordinates.

disp_prt, disp_prt_2d: Added a new [precision] option to select the precision of the numbers (number of decimals) written to the output displacement time series text file.

kml_ts_pt: New program to create a KML or KMZ file for visualizing point deformation time-series.

kml_plist.py: New script to create a KML or KMZ file for visualizing a point list.

spf_pt, fspf_pt: Parallelized using OpenMP.

Python wrapper

-

Matlab wrapper

-