High Energy Solar Physics Data in Europe
(HESPE)

Grant Agreement no.: 263086

Beneficiaries

- **Università di Genova (UNIGE)***
- **Fachhochschule Nortwestschweiz (FHNW)**
- **University of Glasgow (UNIGLA)**
- **Universitaet Graz (UNIGRA)**
- **Centre National de la Recherche Scientifique (CNRS)**
- **University of California at Berkeley (UCB)**

*** NASA Goddard Space Flight Center (GSFC) and CNR - SPIN, Genova (CNR) are 'Third Party' in UNIGE**

Coordinator: prof. Michele Piana (UNIGE)

Deliverable D6.2: Release of computational tools: Release of computational tools for instrument-independent visibilities (Month 36)

This deliverable provides IDL codes implementing image reconstruction algorithms for hard X-ray visibilities. The main feature of these codes is in their complete independency of the hardware characteristics of the observing instrument. In fact previous releases implementing these same algorithms have been prepared for taking as input RHESSI visibilities and therefore depend on RHESSI hardware properties at two levels:

1. The visibility bags that are the input data for these codes are typically structures containing tags, some of which explicitly refers to properties pertaining the instrument that recorded the visibilities. Past releases utilized such tags during the data processing.
2. The routines often exploit parameters characterizing the geometrical design of RHESSI collimators' grids. Further these routines often call sub-routines that are strongly RHESSI-dependent.

HESPE releases of these codes overtake both these aspects. Specifically, as far as the first level is concerned, HESPE codes still take in input visibility bags containing instrument-dependent tags (for example, bags generated by RHESSI or STIX) but utilize in the processing just the instrument-independent ones. As far as the second level is concerned, all calls to RHESSI-specific routines have been deleted and the RHESSI-specific functions and sub-routines have been re-written in an instrument-independent manner.

As an example of this procedure, consider the coverage of the spatial frequency (u,v) plane provided by RHESSI. The radius in this plane, corresponding to a specific RHESSI collimator, can be computed as the inverse of the grid pitch (which is an instrument-dependent way to obtain it). However, this same radius can be obtained by computing the norm of the vector whose Cartesian components are the spatial frequency u and v (and this does not depend on the hardware characteristics).

In the following we provide a schematic description of the HESPE routines for visibility-based imaging and specify whether these software packages are already at disposal of the users in either SSW or in the RHESSI and STIX GUIs. All releases can be downloaded from the HESPE Web Site at http://www.hespe.eu/software/hespe-imaging-routines/

Name of the algorithm: visibility back projection

Acronym: VIS_BPMAP

Name of the main routine: vis_bpmap.pro

Description: discretized inverse Fourier transform of the visibility set ('dirty map')

Input: any visibility bag

Keywords:

- bp_fov: FOV (arcsec)
- pixel: pixel size (arcsec)
- quiet: suppresses output to the screen
- peakxy: location of the map maximum
- noplot: suppresses plot output
- edgeflag: warning on critical peak position (flag)
- label: plot title
- time: event time range
- uniform_weighting: choice between natural and uniform weighting (flag)
- spatial_frequency_weight: set by uniform_weighting (if used)
- data_only: choice on the output format (flag)
- loopstyle: for debugging

Output:

- 'dirty map'

Dependencies (hardware): none

Dependencies (software): IDL/SSW

Availability:

- SSW: yes
- RHESSI GUI: no
- STIX GUI: yes

Name of the algorithm: visibility-based CLEAN

Acronym: VIS_CLEAN

Name of the main routine: vis_clean.pro

Description: construction of point sources (CLEAN components) obtained from the 'dirty map' (output of back-projection) by iteratively searching pixels of maximum intensity. In the final step the map of CLEAN components is convolved with an ideal PSF

Input: any visibility bag

Keywords:

- niter: maximum number of iterations allowed
- image_dim: image dimension
- pixel: pixel size (arcsec)
- gain: CLEAN loop gain factor
- clean_box: FOV where to clean
- negative_max: stopping criterion (flag)
- beam_width: PSF beam width (FWHM; arcsec)
- spatial_frequency_weight: weights on the frequency components

Outputs:

- 'dirty map'
- map of CLEAN components
- CLEAN map without residuals
- CLEAN map with residuals added
- map of residuals
- iteration number
- structure containing all outputs

Dependencies (hardware): none

Dependencies (software): IDL/SSW

Availability:

- SSW: yes
- RHESSI GUI: no
- STIX GUI: yes

Name of the algorithm: visibility forward fit

Acronym: VIS_FWDFIT

Name of the main routine: vis_fwdfit.pro

Description: construction of a parametric map that best fits the observed visibilities. The parametric forms allowed are: single Gaussian source, two Gaussian sources, loop, albedo

Input: any visibility bag

Keywords:

- srcin: structure defining the parametric source type
- multi, loop, circle, albedo: fit with two Gaussians, loop, single Gaussian, albedo
- fit_mask: array of binary values indicating which models are used in the fit
- nophase: forces all input phases to zero
- maxiter: sets maximum number of iterations per stage
- absolute: enables alternative way to compute the fit
- syserr: estimate of the systematic errors
- noedit: suppresses editing and combining of visibilities
- noerr: forces fit to ignore input statistical errors
- showmap: display of final map
- noplotfit: suppresses plot fit display
- srcout: structure containing fitted source parameters
- fitstddev: standard deviation of the fitted parameters
- qflag: flag on quality of fit
- redchisq: contains the reduced chi2 of the fit
- niter: number of iterations needed for fitting

Outputs: print of the fitted parameters

Dependencies (hardware): none

Dependencies (software): IDL/SSW

Availability:

- SSW: yes
- RHESSI GUI: no
- STIX GUI: yes

Name of the algorithm: interpolation/extrapolation of visibilities

Acronym: UV_SMOOTH

Name of the main routine: uv_smooth.pro

Description: construction of a visibility surface by means of interpolation; inverse Fourier transform from limited data by means of projected Landweber algorithm

Input: any visibility bag

Keywords:

- reconstructed_map_visibilities: interpolated and extrapolated visibility surfaces
- noplot: disables plot of the uv-plane sampling
- uv_window: allows refresh of current window (flag)

Output: map

Dependencies (hardware): none

Dependencies (software): IDL/SSW

Availability:

- SSW: yes
- RHESSI GUI: no
- STIX GUI: yes