VisibilityGuide.doc
ghurford@ssl.berkeley.edu
23-Dec-2005

Users' Guide to RHESSI Visibilities

This write-up summarizes the properties of RHESSI visibilities and the current version of the visibility software.

Basic Properties of RHESSI Visibilities

A RHESSI visibility is a complex observable number that can be derived from RHESSI data and which represents a calibrated measurement of a single Fourier component of the source distribution measured at a specific spatial frequency and energy- and time-range.

As a complex number, it can be represented either by its amplitude and phase, or by the corresponding Cartesian representations, variously called the X and Y, cosine & sine, or real and imaginary components.

As a calibrated measurement, it incorporates all known *spatial* calibration information so that no further reference to aspect solution, grid response, etc is needed to interpret it. Note that the energy range of a freshly-measured visibility refers to the *detected* energy, not *photon* energy. As a 'semicalibrated' value, it therefore incorporates the diagonal elements of the detector response, but not the off-diagonal components. With that caveat, a measured visibility has units of photons/cm2/second.

Except for contributing to statistical errors, background is inherently removed in the process of calculating visibilities. Background (in this context) refers to any photons that have not passed through both grids.

The spatial frequency at which the visibility is measured is characterized either by its Cartesian coordinates (traditionally labeled u, v) or by the polar equivalent as the magnitude of the spatial frequency and its orientation. For RHESSI the magnitude of the spatial frequency is determined the subcollimator number (and harmonic) and is given by 1/(angular pitch) of the subcollimator. For the first harmonic, this is equivalent to 0.5/(FWHM resolution) (about 0.22 arcsec^-1 for grid 1 of 0.0027 arcsec^-1 for grid 9). Note that the courser grids have the smaller spatial frequency. By convention, the orientation of a RHESSI visibility is defined by its position angle in the u,v plane, measured east from solar north. As RHESSI rotates, the magnitude of the spatial frequency measured by a given subcollimator remains fixed, while the position angle of the spatial frequency *decreases* as the measured visibility moves in a circle in the u,v plane.

Except for the units, RHESSI visibilities are equivalent to those measured by radio interferometers, a feature which opens the way to using full-featured image reconstruction software developed for radio applications.

Since the x-ray source is real (both practically and mathematically), visibilities measured at opposite points in the u,v plane are complex conjugates. This implies that the amplitudes of visibilities measured at opposite half rotations must be equal, and the phases of visibilities measured at opposite half rotations must sum to zero. To within statistical expectations, this condition should hold independent of source morphology. If this is not the case, then either the source time variability is relevant, or there is a calibration or calculation error. It also follows that such 'conjugate visibilities' can (and should) be combined without losing any imaging information.

As observables, visibilities are inherently linear both in terms of their dependence on the data and in terms of their dependence on the source. For example, the (complex) sum of two visibilities (at the same spatial frequency) measured over two separate time or energy intervals should give the same result as a measurement over the combined time or energy interval. Similarly, the visibilities of a multicomponent source should equal the sum of the corresponding visibilities of the individual components.

A visibility poster from the 2005 SPD, can be found at http://sprg.ssl.berkeley.edu/~ghurford/SPD05.pdf

Some possible application of RHESSI visibilities

- Fe line imaging by suppressing the continuum
- Measurement of source sizes
- Spatial variations on sub-second timescales
- Albedo isolation and characterization
- Averaging data over long time intervals
- Imaging in ‘photon energy’ rather than ‘detected energy’
- Improved pileup corrections
- Separation of nuclear and electron contributions to gamma continuum
- Subtracting ‘pre-event’ imaging contributions

to be fleshed out

Interpreting visibilities

to be written

Software Overview

In terms of function, the visibility software can be divided into three areas:

Calculation: Modules for calculating visibilities from the level-0 data.
Editing: Modules for combining and/or editing visibilities measured at different times and energies.
Interpretation: Modules for displaying and/or interpreting visibilities in terms of the properties of the source.

A typical sequence would be to calculate visibilities from level-0 data using hsi_vis_usershell; save, display, edit and/or combine visibilities depending on the goals of the analysis; and then interpret the resulting visibilities in terms of the properties of the source. To evaluate spectral or temporal differences, this process is repeated (most easily using a script).

For now, the intermediary among these routines is an array of **'visibility structures'** that contain all the information necessary to display, combine and interpret the observed visibilities. The format of the visibility array is the same for the calculation output, editing input and output and interpretation input. Visibility arrays can be SAVEd and RESTOREd as compact .sav files.

Each visibility structure (September '05 version) corresponds to a single subcollimator, harmonic, u, v, energy and time combination. They are combined into an overall array of such structures in no particular order, so that it is more akin to a 'bag of visibilities' than a visibility cube.

The tags in the visibility structure are;

• isc	INTEGER	Subcollimator index	(=0,,,,8)
• harm	INTEGER	Harmonic number	(=1,2,3)
• erange	FLTARR(2)	Energy range:	(keV)
• timerange	DBLARR(2)	Time range:	(anytim)
• u	FLOAT	u=East-west spatial frequency component	(arcsec^-1)
• v	FLOAT	v=North-south spatial frequency component	(arcsec^-1)
• obsvis	COMPLEX	Observed (semicalibrated) visibility	(ph/cm2/s)
• totflux	FLOAT	"Total flux" or semicalibrated 'DC' term	(ph/cm2/s)
• sigamp	FLOAT	Average statistical error in obsvis components	(ph/cm2/s)
• chi2	FLOAT	Reduced CHI^2 in fitting visibility to stacked event list	
• xyoffset	FLTARR(2)	West, north heliocentric offset of phase center	(arcsec)

The current visibility 'package' includes the following .pro modules:

hsi_vis_usershell
: User function, intended for editing by the user, to calculate and returns a visibility structure.

hsi_vis_display
: User procedure to display contents of a visibility structure

hsi_vis_avg_size
: User procedure to calculate the rotationally averaged source diameter

hsi_vis_size_vs_pa
: User procedure to calculate the source size as a function of position angle

hsi_vis_fluxcalc
: Temporary shell that calculates (but does not save) visibilities and calculates source parameters.

hsi_vis_fwdfit
: User procedure to fit a predetermined source shape (circular or elliptical gaussian, loop or two circular gaussians) to visibilities

hsi_vis_combine
: User function that combines visibilities corresponding to similar (and/or conjugate) u,v points.

hsi_vis_edit
: User function that returns an edited version of the input visibility structure. This can be used to eliminate bad visibilities, combines conjugate visibilities and/or selects visibilities on the basis of their energy.

hsi_vis_scale
: User function that applies a complex gain to all the input visibilities.

hsi_vis_bpmap
: User diagnostic procedure to display an unnormalized backprojection map made from visibilities.

hsi_vis_consistency_check
: User procedure to verify the consistency of conjugate visibilities – primarily useful for grid calibration. Not yet adapted to the Sept 2005 visibility format.

The following supporting modules are, in general, transparent to the user.

hsi_vis_fwdfit_array2structure
hsi_vis_fwdfit_bifurcate
hsi_vis_fwdfit_fixedconfig
hsi_vis_fwdfit_func
hsi_vis_fwdfit_parmrange
hsi_vis_fwdfit_print
hsi_vis_fwdfit_structure2array
hsi_vis_fwdfit_fixedconfig
hsi_vis_fwdfit_makealoop
hsi_vis_fwdfit_plotfit
hsi_vis_fwdfit_print
hsi_vis_fwdfit_parmrange
hsi_vis_fwdfit_sigmacalc
: Supporting modules for hsi_vis_fwdfit

hsi_vis_plotfit

hsi_gaussfit
: Supporting module for use in determining source sizes.

hsi_vis_select
: Supporting function that returns the optionally-ordered indices of visibilities satisfying specified criteria. Optionally returns position angles of u,v points.

hsi_vis_sort
: Supporting function that identifies visibilities with equivalent u,v values.

hsi_visibility_fit
: Support module for the generation of visibilities

Several modules, written by Richard Schwartz to support the generation of visibilities, have been integrated into ssw.

There are also a set of routines developed by Ed Schmahl et al that use MEM_NJIT to convert a set of visibilities to a map. These are documented elsewhere and are being integrated into ssw.

Calculation of visibilities:

The calculation of visibilities is done by defining an image object, setting its parameters, calculating the calibrated eventlist, converting the calibrated eventlist to a stacked eventlist and then converting the stacked eventlist to a set of visibilities. These steps are transparent if the user makes the obvious edits to the code or inputs to hsi_vis_usershell.

The principle decision that the user needs to make is the number of position angles at which visibilities should be calculated. The minimum number is set by the ratio of the maximum diameter of the source to the subcollimator resolution. It can be calculated automatically using this criteria by setting phz_n_roll_bins_control=0. Smaller values provide better s/n for each visibility, but may compromise sensitivity to sources near the edge of the field of view. Larger values may compromise the accuracy of visibilities, especially if the source is highly time-variable or if it is near the axis of rotation. (Typical values are in the range 8 to 32.) In any case, it is recommended that the number of roll bins be even.

As an interim 'feature', a file, 'visfit.ps' can be created to display the visibility fits for each roll bin.

Incorporate material from the roll-bin memo

Use of other user-modules

Details on the arguments of user functions and procedures can be found in each user module.

to be improved

Cookbook examples of visibility generation and use:

Example 1. Measure the size and flux of a simple source.

1. IDL> *hsi_vis_usershell, time_range=time_range, energy_range=energy_range, $*
   xyoffset=xyoffset, /PLOTFIT to specify desired time, energy range and xyoffset. This creates visibility structure, out.

2. Optionally examine visfit.ps plots to see quality of fits.

3. IDL> *oute = hsi_vis_edit(out)* converts 'out' to an edited version, oute, with outliers eliminated

4. IDL> *hsi_vis_display, oute, /ps* displays edited visibilities in idl.ps

5. Optionally examine visibility display in idl.ps

6. IDL> *outec = hsi_vis_combine(oute, /conj)* combines conjugate visibilities in 'oute'

7. IDL> *SET_PLOT='win'* or 'x' for unix users

8. IDL> *hsi_vis_size_vs_pa, outec* Optionally estimate diameter (and flux) vs posn angle

9. IDL> *hsi_vis_fwdfit, outec*, */SHOWMAP* fit an elliptical gaussian to visibilities and display it. Other options are */CIRCLE, /LOOP* (curved gaussian) and */MULTI* (two circles)
   An instructive residual plot written to the current plotting device.

10. IDL> *SAVE, outec, 'myfilename'* optionally save the edited visibility structure

Example 2. Make a spectral line map.

1. Edit hsi_vis_usershell to correspond to the line energy range.
2. IDL> *vis1 = hsi_vis_usershell()* ; vis1 corresponds to line+continuum visibilities.
3. Edit hsi_vis_usershell to correspond to a range of energies in the nearby continuum.
4. IDL> *vis2 = hsi_vis_usershell()* ; vis2 corresponds to continuum visibilities
5. IDL> *vis1e = hsi_vis_edit(vis1)* ; Eliminate bad points in measured visibilities
6. IDL> *vis2e = hsi_vis_edit(vis2e)*
7. IDL> *vis1ec = hsi_vis_combine(vis1e, /conj)* ; Average conjugate visibilities
8. IDL> *vis2ec = hsi_vis_combine(vis2e, /conj)*
9. By 'other means' determine k, such that k * flux in nearby continuum equals the expected continuum flux in the line energy range.
10. IDL> *vis2ecn = hsi_vis_scale(vis2ec, -k)* ; Apply the –ve normalization factor
11. IDL> *visline = hsi_vis_combine([vis1ec, vis2ecn], /ADD, VISCOUNT=2)* ;visline = line visibilities
12. Go to step 9 in previous example using visline as input to map.
13. Note that visibilities corresponding to intermediate results can be saved, displayed, evaluated, etc as desired.

Short-term plans

- Integration into ssw (RS)
- User's guide to vis_fwdfit
- Flesh out documentation
- More testing.
- Improvements to the calculation of visibilities
    - handlling of time-varying source flux
- Improvements to hsi_vis_fwdfit.
    - incorporate edit and combine functions into usershell
    - support for ALBEDO, POINT options
    - support for user-specified search limits
- Module to shift phase center
- Module(s) to support high time resolution applications
- Scripts or shells to automate typical combinations of tasks
    - spectral line mapping
    - time sequences
    - source parameter variations