

14-Apr-2007 Gordon Hurford

Modified 06-Oct-2017 Kim Tolbert
(added albedo_apply, fitmask)

Modified 27-Sep-2022 Kim Tolbert
(corrected names of conversion routines, and added note about ellipse connected parameters)

SOURCE REPRESENTATION IN HSI_VIS_FSWFIT

Source configurations are represented in two ways in hsi_vis_fwdfit and supporting routines.

The first representation is a “**source structure array**” (*srcstr*), each element of which is a structure that defines a single source component with tags that are meaningful to the end-user.

The second representation is a single “**source parameter vector**” (*srcparm*), which incorporates information corresponding to all elements of the corresponding source structure array. Within hsi_vis_fwdfit, some of the elements of the source parameter vector are optimized in each call to a general-purpose fitting routine (such as amoeba_c). The specification of which elements are used in a given call is indicated by zeroes in a corresponding “parameter_mask” vector. The interpretation of the *srcparm* vector elements is implicit in their location. The form of these parameters has been chosen to support efficient convergence of the fitting routine. There are ‘noise-free’ transformations between the source structure array and source parameter vector representations.

Note that when fitting an ellipse, the parameters (in the *srcparm* vector) that are actually optimized in the fit are the x and y components of the eccentricity measure, which are computed from the eccentricity and the position angle (in the *srcstr* array). Since they are connected, if you allow either the eccentricity or the position angle to be optimized while fitting, both will change.

Source structure array

The *srcstr* array has *nsrc* elements, where *nsrc* is the number of source components. Each structure describes one source component. The tags for each structure, defined in hsi_vis_src_structure__define.pro, are:

srcstr.srctype:	Label defining source type: ‘point’, ‘circle’, ‘ellipse’, ‘loop’ or ‘albedo’
srcstr.srcflux:	Semi-calibrated flux (ph/cm ² /s)
srcstr.srcx:	Offset (+W) relative to sun center (arcsec)
srcstr.srcy:	Offset (+N) relative to sun center (arcsec)
srcstr.srcfwhm:	Source FWHM diameter (arcsec)
srcstr.eccen:	Eccentricity of source (ecc=0 is a circle, .99 narrow ellipse)
srcstr.srcpa:	Position angle (E of N) of long axis (deg)
srcstr.loop_angle:	Angle subtended by loop as viewed from center of curvature (deg)
srcstr.albedo_ratio:	Ratio of albedo flux to the sum of the other primary source fluxes.
srcstr.srcheight:	Average height of other source components above the ‘albedo surface’ (Mm)
srcstr.albedo_apply:	Apply albedo to this source (if an albedo source is enabled)
srcstr.fitmask:	intarr(10), array of 0/1 to indicate which parameters to vary during fit

Not all tags are meaningful for all types of sources.

(In future, it may be desirable to replace srcfwhm and eccen with majorfwhm and minorfwhm.)

Source parameter vector

The *srcparm* vector is a $10 * nsrc$ element vector, reformed from a $10 \times nsrc$ array. Unfortunately, the *amoeba_c* fitting routine requires this rather awkward representation. The interpretation of the 10 elements in the 1st dimension is:

0. srctypecode = 0,1,2,3,4 for 'point', 'circle', 'ellipse', 'loop', or 'albedo'
1. flux
2. x-offset from map center
3. y-offset from map center
4. fwhm
5. Cartesian x component of eccentricity measure
6. Cartesian y component of eccentricity measure
7. Loop angle
8. Albedo_ratio
9. Source height

The corresponding masks for each type of source is as follows:

	tc	f	x	y	w	ex	ey	la	ar	sh
point:	[0, 1, 1, 1, 0,	0, 0, 0, 0, 0]								
circle:	[0, 1, 1, 1, 1,	0, 0, 0, 0, 0]								
ellipse:	0, 1, 1, 1, 1,	1, 1, 0, 0, 0]								
loop:	[0, 1, 1, 1, 1,	1, 1, 1, 0, 0]								
albedo:	[0, 0, 0, 0, 0,	0, 0, 0, 1, 1]								

where 1 indicates a parameter to be used for this type of source.

The fitmask array information in the *srcstr* structure is incorporated into the step sizes passed to *amoeba_c* (keyword *lambda*). Any parameter that is to remain fixed has a step size of 0.

The *albedo_apply* information in the *srcstr* structure is passed to the function called by *amoeba_c*, *hsi_vis_fwdfit_func*, in common *uvdata*

Conversions between source representations

hsi_vis_fwdfit_structure2array
hsi_vis_fwdfit_array2structure