

Modules

gempy/science/extract.py

- trace_slits ULF
- _find_edges
- _fit_edges
- _associate_edges
- Slit class
- SlitsTrace class

gempy/segmentation.py

- Edge class
- TraceConstraints class
- trace_edges
- "apply edge enhancing filter"

Classes – segmentation module

gempy/segmentation.py

Edge

trace : x,y(z,..) coordinates of the trace
function : name of the function being fit to the trace
order : order of the function
coefficients : array of coefficients
dimension : dimensionality of the edge
setfunction : set function, check that it is valid
setorder : based on function and order, also set length of coefficients
setdimension
fitfunction : fit function to trace
evalfunction : evaluate function and return x,y(z,...)
[**plot** : future feature]

TraceConstraints

orientation : general orientation of the edges
filter : filter applied for edge enhancement
Other attributes to support multiple slits per line TBD
setorientation
setfilter

Classes – TraceConstraints

TraceConstraints

Get data-dependent information needed for edge tracing and store in standard variables that will allow the edge tracing to be instrument agnostic.

Initialize:

The class is initialize from an Astrodata object. The filter can be passed as an argument. The orientation is set automatically based on the dispersion axis.

Attributes:

orientation: General orientation of the slit edges.
In degrees. 0 is horizontal, 90 is vertical.
filter: Filter to applied for edge enhancement. [Default: Sobel]
Allowed/implemented filters: Sobel.

Methods:

setorientation: Just set the orientation to the value of the argument.
setfilter: Set the filter to the value of the argument but first check to make sure the filter is in the supported list.

Classes – Edge

Edge

Describes an edge as traced and as fit.

Initialize:

The class is initialize from an Astrodatab object. The filter can be passed as an argument. The orientation is set automatically based on the dispersion axis.

Attributes:

trace: x,y(z) coordinates of the edge trace
function: Name of the function being fit to the trace.
order: Order of the function.
coefficients: Solution to the fit. Array of coefficients.
dimension: Dimensionality of the edge. 0 for points, 1 for line, 2 for plane.

Methods:

setfunction: Set the function to the value of the argument but first check to make sure the function is in the supported list.
setorder: Set the function's order. Also based on the function and order, set the length of the *coefficients* attribute.
setdimension: Set the dimensionality of the edge to the value of the argument but first check that the value is an integer in the 0-2 range.
fitfunction: Fit the *function* to the *trace*.
evalfunction: Evaluate the function and return x,y(z) data points. Do NOT overwrite the trace, just return the coordinates that represent the function associated with the Edge. Obviously, this can only work after a successful *fitfunction* call.
[plot: future feature]

Classes

gempy/segmentation.py

Edge
trace : x,y(z,..) coordinates of the trace
function : name of the function being fit to the trace
order : order of the function
coefficients : array of coefficients
dimension : 1 for point, 2 for line (default), 3 for plane
setfunction : set function, check that it is valid
setorder : based on function and order, also set length of coefficients
setdimension
fitfunction : fit function to trace
evalfunction : evaluate function and return x,y(z,...)
[plot : future feature]

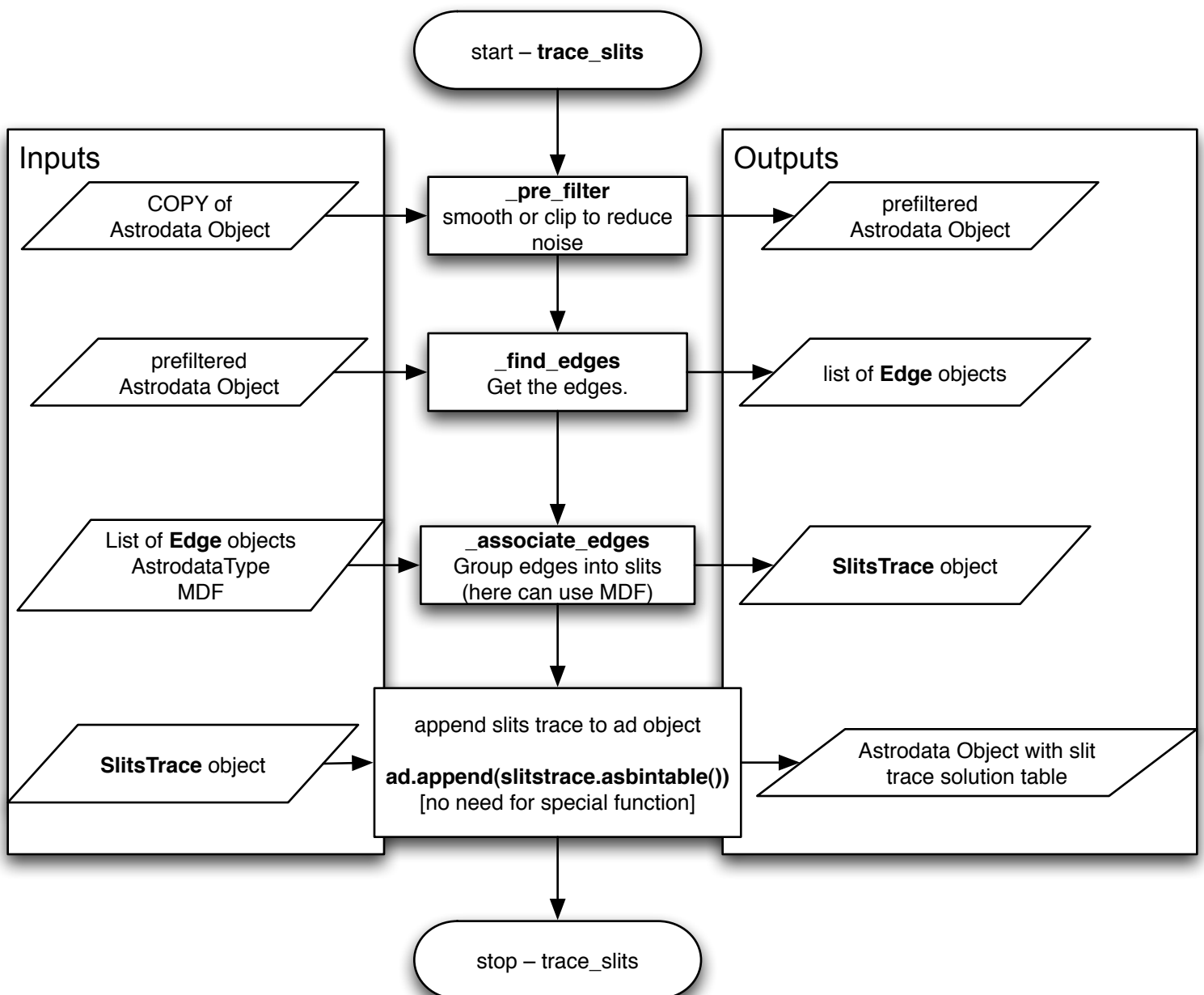
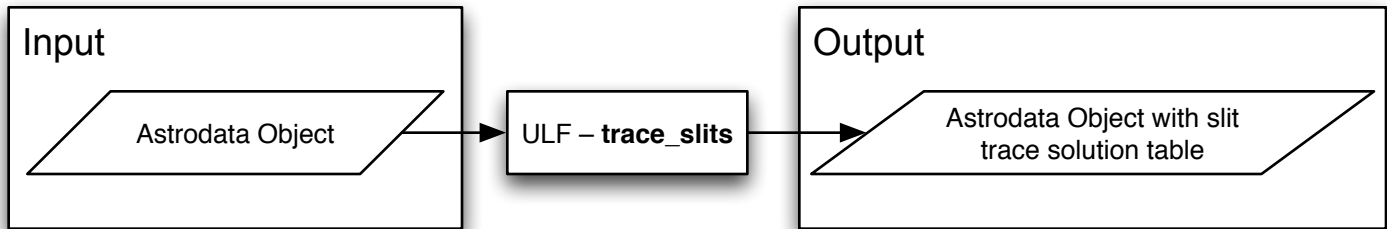
TraceConstraints
Orientation
Other attributes TBD
Methods, if any, TBD

gempy/science/extract.py

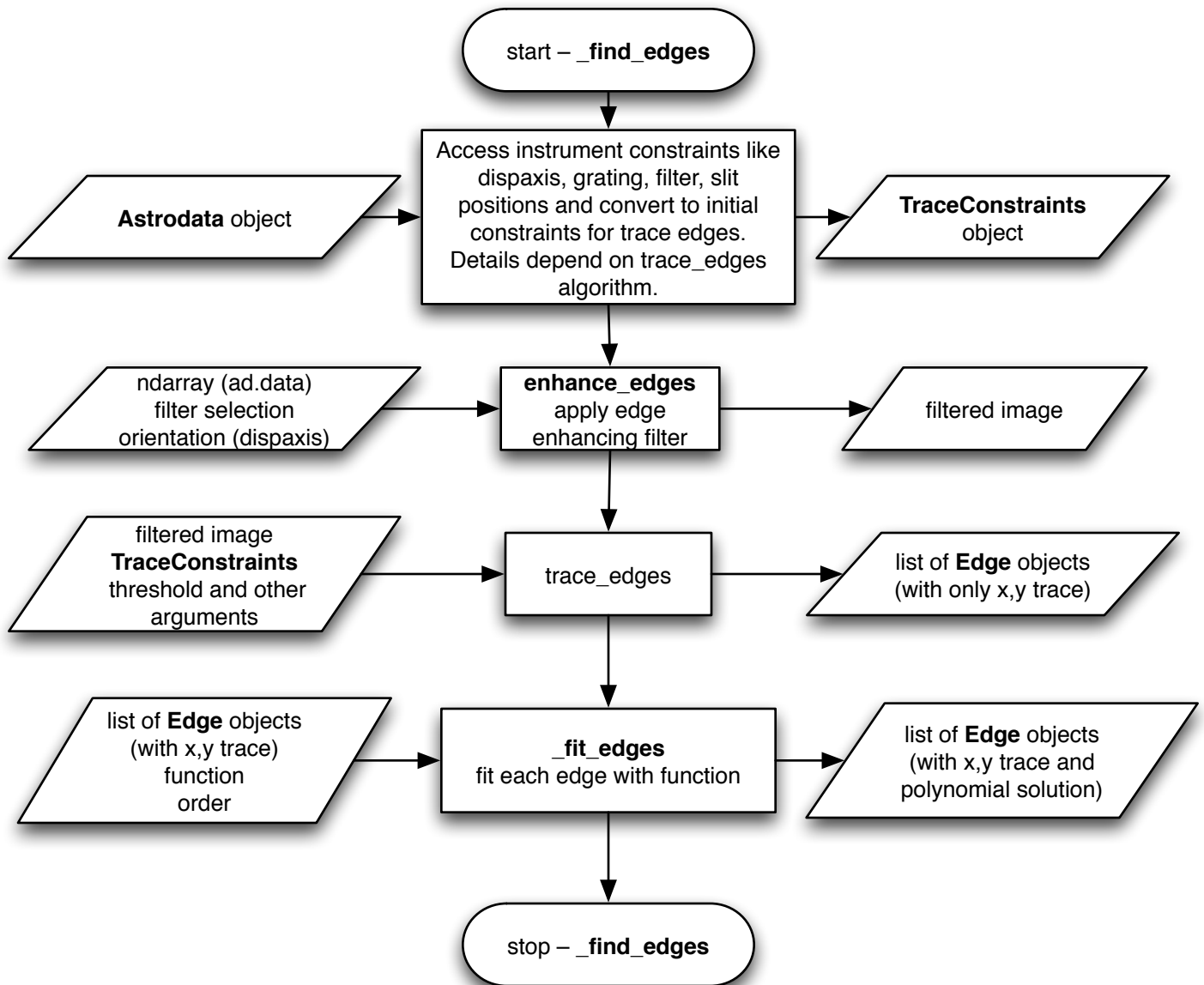
Slit
id : just some numbering
region : section of the image where the slit solution is valid. (x1,x2,y1,y2)
edges : Two Edge objects defining the long edges of the slit.
width : average width of the slit (calculated with the edges)
setregion : default can be derived from Edge objects. Arguments x1,x2,y1,y2 set to None.
[plot : future feature]
[plotregion : future feature]

SlitsTrace
slits
asbintable
[plot : future feature]

ULF – trace_slits



_find_edges



`_fit_edges`

Inputs:

- list of Edge objects (with x,y traces)
- function
- order

Output:

- list of Edge objects with fit coefficients

Design details:

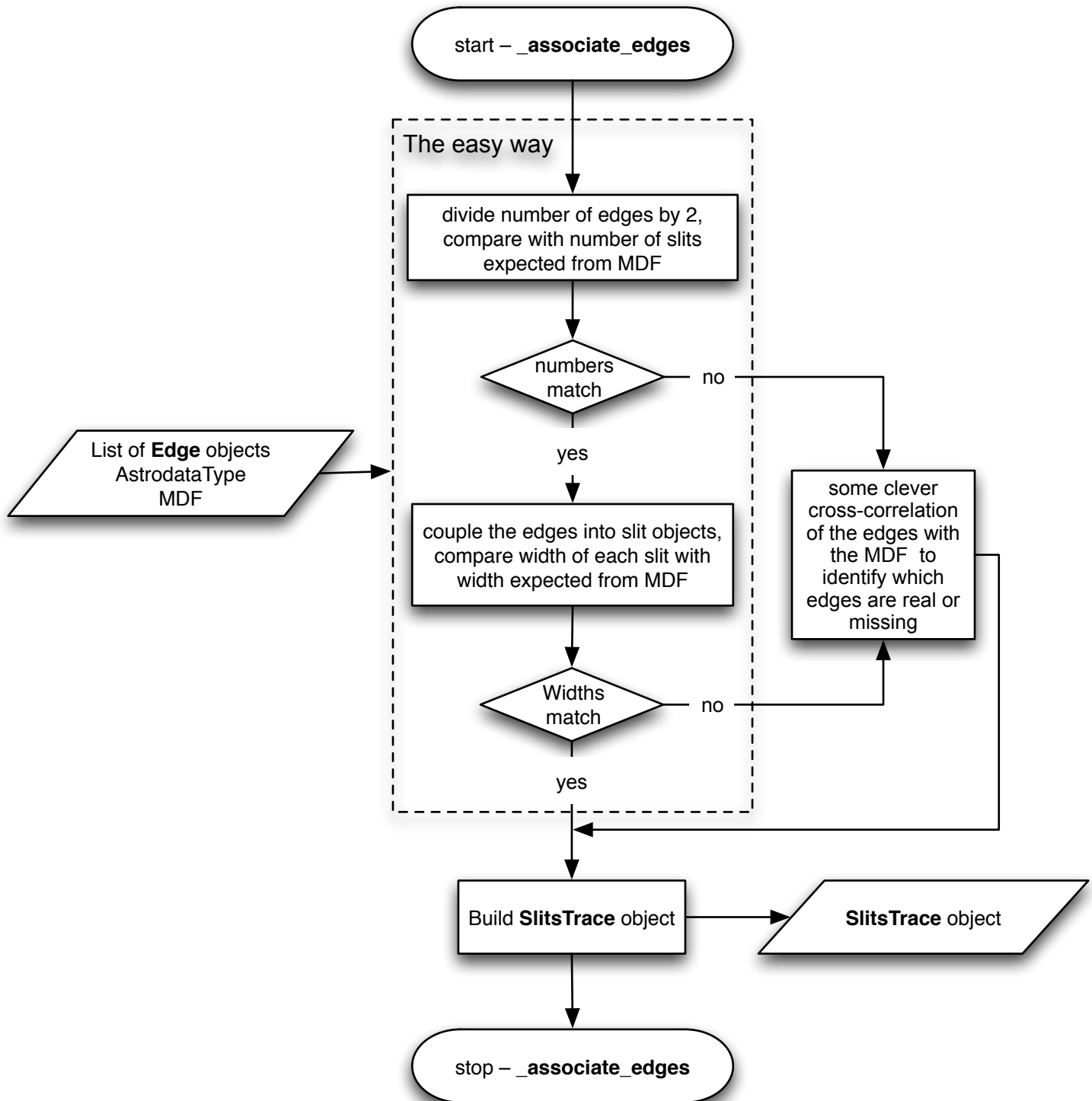
For each edge:

- set function and order
- edge.fitfunction()

NOTE: This is done in a separate routine because we might want to add quality checks or an interactive mode **in the future**.

To be completed.

`_associate_edges`



SlitsTrace.asbintable

Write the slit trace solution to a FITS binary table.

This is a method of the SlitsTrace class. Takes self as input. Returns the bintable.

Column ID	Format	Description
id	J	
region1	4E ??	
function1	15A	
order1	J	
coeff1_1	%dE	
coeff1_N	%dE	
region2	4E ??	
function2	15A	
order2	J	
coeff2_1	%dE	
coeff2_N	%dE	
cutregion1	4E ??	
cutfunction1	15A	
cutorder1	J	
cutcoeff1_1	%dE	
cutcoeff1_N	%dE	
cutregion2	4E ??	
cutfunction2	15A	
cutorder2	J	
cutcoeff2_1	%dE	
cutcoeff2_N	%dE	

trace_edges

(replaces "locate_slit_edges" from old design)

Inputs:

filtered image
TraceConstraints object
threshold in units of sigma
other arguments as necessary depending on design

Output:

List of Edge objects with the x,y trace only. (no fitting yet)

Design details:

First version assumes no slit overlapping. Minimum requirements:

- Orientation (from TraceConstraints object)
- Threshold in units of sigma

NOTE: trace edges does not care about matching edges to form slits.

NOTE: list of input arguments depends on the algorithm used.

If multiple slits per line is to be supported, the problem is a lot more complex. The easiest implementation will require initial conditions. The first version we will not attack this problem. But this is where the upgrade will go.

To be completed. Suggested tools: ndimage.label and ndimage.find_object.

enhance_edges

(replaces "generate_edges" from old design)

Inputs:

ndarray

filter to use (default: Sobel)

orientation (in degrees. default: 0)

Output:

filter image as an ndarray

Design details:

To be completed.

Edge.fitfunction

(replaces "fit_edges" from old design)

Inputs:

self

function (default: self.function)

order (default: self.order)

dimension (default: self.dimension)

Output:

Edge object with fit coefficients.

Design details:

To be completed.