Title :	VIALACTEA WP1 Deliverable Item 1.4 Filamentary Structures Extraction package		
Date:	30/09/2016	Issue:	3
Reference:	VL-IAPS-DI-2016-0001		
Custodian:	E. Schisano		

Authors :	
	E.Schisano
Contributors :	
	M.Benedettini S. Molinari D. Elia A. Di Giorgio

Document version tracking

Issue	Date	Pages	Description of changes	Comments
1	30/04/2014		Document accompanying Version 0 of the package	
2	30/09/2014		Document accompanying Version 1 of the package	Minor Changes and bug fixing
3	15/10/2015		Document accompanying Version 2 of the package	Minor Changes, bug fixing, tools for housekeeping and interpreting outputs.
4	30/09/2016		Document accompanying Version 3 of the packaged	Minor Changes, improvements on outputs, bug fixing, inclusion of libraries and simple installer.

Table of Contents

1. **Introduction**

This document is the user manual of the VIALACTEA filamentary structure extraction package.

The version 0 of the package is the deliverable 1.1 (D1.1) of the Work Package WP1 of the VIALACTEA project. This document is relative to the version 1 of the package for the deliverable 1.2 (D1.2). Two further package updates are foreseen during the project lifetime, their deliverable IDs will be, D1.3 (package version 2) and D1.4 (package version 3).

The delivery package of each one of the above listed deliverables is constituted by:

- The filamentary structure extraction package code: set of IDL functions and procedures archived into a tar file called **D1.1_FilamentPackage.tar** and delivered attached to this document

- The package user manual (this document)

2. **Applicable & Reference documents.**

Doc. Ref.	Title	Ref.	Issue	Date
AD1	The identification of filaments on far infrared and submillimiter images. Morphology, physical conditions and relation with star formation of filamentary structure"	Schisano et al. 2014, ApJ, 791, 27		2014
AD2	"Clouds, filaments, and protostars: The Herschel Hi-GAL Milky Way"	Molinari et al. 2010, A&A, 518, L100		2010
AD3	"Hi-GAL: The Herschel Infrared Galactic Plane Survey"	Molinari et al. 2010b, PASP, 122, 889, 314-325		2010
AD4	VIALACTEA WP1 Deliverable Item 1.5: Hi-GAL Catalogue of Filamentary Structures	VL-IAPS-DI-2014-002	1	2014
AD5	VIALACTEA WP1 Deliverable Item 1.6: Hi-GAL Catalogue of Filamentary Structures	VL-IAPS-DI-2014-008	2	2014

3. **Acronyms and Abbreviations**

RoI = Region of Interest

MST = Minimum Spanning Tree

4. Overview of the document

This document is intended to be used for all foreseen D1.1 and D1.2 deliveries. Therefore in addition to section 5 dedicated to the package description, the section 6, containing the package user manual, and the section 7 reporting the minor changes for the version 1, three sections dedicated to the description of delivery notes specific to the further versions have been added to the overall document contents; they are of course left blank in the present document version.

5. Filamentary Structure extraction package general description

Observations at different wavelengths and molecular tracers of the relatively best studied and nearby star-forming regions suggest complex morphologies, with the dust and gas arranged mostly along elongate, almost one-dimensional, filamentary structures. In particular the Herschel Space Observatory, thanks to its superior spatial resolution and sensitivity in the far-infrared, shown that the filamentary organization of the dense interstellar material is much more pervasive than was initially thought. Filaments appears everywhere on Herschel maps and they are actually considered a key-structures required to build the densities necessary for star formation. Moreover, since the from pre-stellar to protostellar young condensations are mostly associated within these structures, filaments are believed to define the initial physical conditions for star formation. The typical approach based on visual selection methods, where the researcher manually identifies the most obvious elongated structures and subsequentially chooses portions of these filaments, are impraticable when applied to large data sets. For example, in the framework of the Hi-GAL project, Herschel mapped the entire Galactic Plane at 70, 160, 250, 350, and 500 μm covering a total area of about 720 square degrees (Molinari et al. 2010). In every map extended filamentary structures dominate the observed emission, with hundrends of filaments that can be recognized even at a first glance (see Fig.1 and 2, from Molinari et al. 2010b).
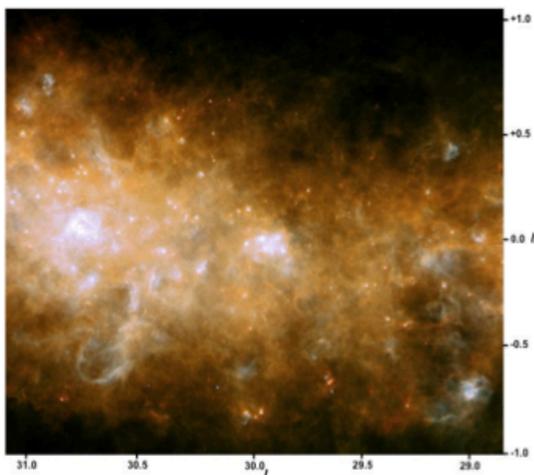

Fig. 1. Three-color image (*blue* 70 μm, *green* 160 μm, *red* 350 μm) of the 2° × 2° field around $l = 30°$.

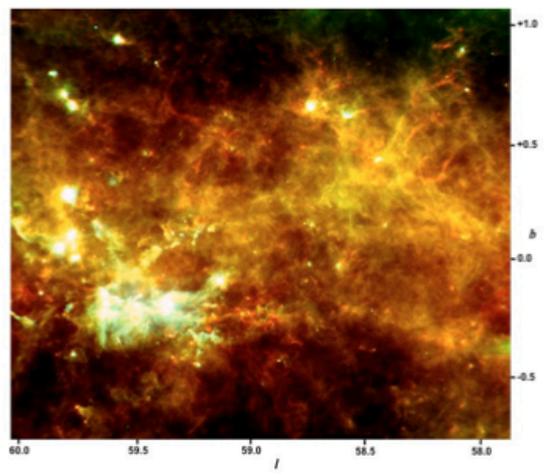**Fig. 2.** Three-color image (*blue* 70 μm, *green* 160 μm, *red* 350 μm) of the field around $l = 59°$.

5.1. Package purpose

The delivered package has the aim to fulfil the requirement of an automatic filament detection tool able to identify and to characterize, by determining the morphological and physical properties, the complex filamentary structure from 2D maps when a bright and variable background is present, like in the observations of the Galactic Plane. A full description of the approach is found in Schisano et al. 2014, here we give a brief summary divided between the detection and the extraction approaches.

The identification methods start from a precise definition for the pattern that is desired to highlight. Differently than previous studies in the field in which filaments are considered as one-dimensional structures, here we considered them as an extended 2-dimensional feature that covers a portion of the map. Thus, the final objective for the detection algorithms is to segment the map by separating the pixels that belong to the filaments from the others. Such classification has to take into account the shape of the local emission.

A filament is defined as an elongated region with a relatively higher brightness contrast with respect to its surrounding, formalizing the intuitive idea of what a filament looks like based on what the eye sees on a map. On first approximation, filaments can be considered as cylinder-like patterns that are more concave along one direction with respect to the orthogonal one. As the curvature of an intensity map along any direction is proportional to its directional second derivative, the Hessian operator is most suited to characterize the spatial properties of the emission intensity field. Furthermore, in particular, the difference in curvature between the two directions would be the highest if the chosen directions to evelutate the curvature are aligned along the cylinder axis. In such a case the main axis has a rather flat curvature while the orthogonal radially directed would have the larger convexity. Thus, it is more convenient to analyze the eigendecomposition of the Hessian matrix of the emission intensity field (in the case of the far infrared observations such emission is produced by dust thermal) to define the directions of the principal axes at each position of the observed map, the shape and the contrast of the eventual local structure.

The direction of the principal axis are defined by the two eigenvectors, A_1 and A_2, while the shape and the contrast are given by the two eigenvalues λ_1 and λ_2 that are proportional to the curvature along these direction in each pixel position. As we are focusing on the detection of emission features, we will be interested in convex morphologies, i.e. $\lambda_1 < \lambda_2 \leq 0$. In this notation we will assume that direction 1, being the one of maximum absolute curvature, will identify the cross-filament direction. In particular, for the filament features it is expected to have $\lambda_1 << \lambda_2 \leq 0$ with the filament axis defined by the direction of A_2.

Although useful for tracing the features on simulations, or relatively smooth data, this approach has some drawbacks when applied to maps of the interstellar medium. In fact, the above relation does not hold close to strong overdensities like, for example, compact clumps or cores found along the filament for which $\lambda_1 \approx \lambda_2 \leq 0$ Hence, we have to relax the formal criteria keeping in mind that the maps of the second derivatives have the advantage that they filter out the large scale emission (present on the Galactic Plane) and emphasize the more concentrated emission. All the quantities computed from the second derivative are able to pinpoint the strong variation in the gradient (i.e. change in the contrast) of the intensity distribution. Therefore, the map of λ_1 include all the filamentary regions and the main problem restrict to a convenient thresholding of this map and to a filtering out the Regions of Interest (RoIs) that do not resemble filamentary, long, elongated regions. The absolute value of the adopted threshold defines the lowest contrast that a region should exhibit to be considered as belonging to a filamentary structure. The optimal choice of the threshold depends on the condition of the map on which the user is working; in particular it depends on the strength of the diffuse background emission and of the pixel-to-pixel noise. All the RoIs are smoothered by appling a morphological

closing operator to remove discontinuous edges and are fitted by ellipses to filter out the roundish structures like large and elongated compact clumps. Small, slightly elongated RoIs, short in comparison to the instrumental point-spread function on the image are also removed from the sample. Due to the web-like shape of the filamentary structure, it may happen that multiple filaments intersect each other and, depending on the contrast threshold adopted, may be catalogued as one single roundish region. Thus, to avoid to discard such a morphology, we further determine the filling factor of the RoI, as the ratio between the area of the RoI and the area of the fitted ellipse. The RoIs with a smaller filling factor are very likely to cover filamentary-like portion of the map.

Once a set of candidate RoIs decontaminated from compact roundish clumps have been selected, the package proceed to estimate the morphological and physical properties of each structure on the map. The filament axis, called ''spine'' is identified by applying a morphological operator of thinning on each region. Under the assumption that the filament is symmetric in its profile, the operator proceed to iteratively remove pixels from the RoI boundaries until no further pixels can be removed without deleting completely the RoI. The surviving pixels represent a skeleton of the RoI, tracing all its substructures. The spine pixels are then connected through a ``Minimum Spanning Tree'' (MST), implemented with the Prim algorithm, to define the unique path that joins them together. This allows us to identify nodal points where multiple branches of the MST depart and immediately enables the classification of structures in main hubs of peripheral branches. The ''spine'' characterizes the length and the direction of the filament, the latter being estimated as the angular coefficient of the linear fit to the position of the pixels that belong to the same branch. The cross-spine size of the filament, or its width, is evaluated by fitting a Gaussian function to the brightness profile in the direction orthogonal to the spine and compute the median of all the FWHM measured along each branch. Still it is possible that for very complex features where filaments are organized in a multiple nested web-like structure, the cross-spine profile fitting often fails to converge.

The measure of the total emission associated to the filamentary structure relies on a realistic estimate of the surrounding background. Furthermore, the extend of each filamentary region is determined by identifying where the filament merges into the background. Such a position is identified either by the Gaussian fitting in the cross-spine direction and by enlarging few times the thresholded map of the eigenvalue. The merit in doing this is that whatever the threshold adopted, the thresholding is always done over the map of minimum but negative eigenvalues; in other words, the pixels selected will always belong to regions where the curvature of the brightness profile in the maximum curvature direction is within the convexity region. This implies in general a conservative identification of the filament region, because it would neglect the wings of the filaments where the emission profile changes concavity before joining the background emission. The two approach are in a very good agreement (see also Schisano et al. 2014). Once a good estimate of the region boundaries is done we select the pixels that lie in the direction perpendicular to the local spine and intersect with a 2-pixels-wide boundary region surrounding the filament RoI. The values in the two sets of pixels, one on each side of the filament, are then linearly interpolated along the cross-spine direction and produced a reliable local estimate for the background. This is repeated for all filament spine points, producing a detailed estimate of how the background varies along the filament.

It is worth noticing that the methods and algorithms followed by the package have a wide set of applications that go beyond the simple application to Hi-GAL data, however, the set of scripts described in the following were designed and tailored for Herschel observations.

5.2. **Package structure**

The filamentary detection package consists in a set of IDL scripts, optimized for IDL ver 7.1. The current delivered package refers to the stable version 0.5 that include all the tools to perform the detection of the filaments from Hi-GAL maps and to estimation their parameters. The delivered tar file includes 4 directories, with 3 directories of IDL libraries ('MST/', 'Statistics/', 'Surface/' – see also section 6.1) and the directory 'detect_filaments/' with the package itself. The package is formed by 97 files arranged in 4 directories, 'code/', 'auxiliary/', 'utilities/', 'simulation/', however the main scripts effectively necessary to fulfil the tasks of identifying and characterizing filaments (excluding the various library procedures) are restricted to 26 files located in the directory 'code/'. All those scripts are called at runtime by the procedure 'run_fildet.pro' located in the directory 'detect_filaments/'.

Several addictional scripts are delivered with the current release. The ones located in the directory 'utilities/' are scripts working as procedures written for more general purposes than the one required by the package. Due to their general nature they are not included as a stand alone library, but they are still required for a successful run of the package. In the directory 'simulation/' are located few scripts able to produce images of simulated filaments like the adopted by Schisano et al. 2014 to test the performances of the package. Finally, in the directory 'auxiliary/' there are a set of temporany scripts suited to for tasks that are not strictly related with the package purposes, but they fulfil further task like the correlation of the filaments extracted catalog with compact source ones. However, those scripts are not needed for a full run of the package and many of them are still on revision. Most of them will be eventually replaced in further versions of the package and they are included here only for sake of completeness.

5.3. **Package inputs**

The package is optimized to run on images in FITS format readable by the IDL routine 'readfits.pro' included in the 'astrolib' library. It works on single frame/layer FITS, like the maps produced by the Hi-GAL project, on which the pixels with a not defined number ('NaN') have been replaced by the same, constant, value (preferably 0).

An input ascii file defining the region where to compute the Hessian matrix is required for a complete successful run. The format file has to follow the one of the region files the application SAOImage ds9 (available at http://ds9.si.edu/site/Home.html). The ds9 application is a wide spread software adopted in the astronomical community for imaging and data visualization and it allows easily to write the necessary region file. For a precise description of the format of the region file see the section 6.2.

Either the map FITS file and the region file has to be present in the directory where the package is actually launched.

5.4. **Package outputs**

The package outputs are all collected in a directory created in the place where the it is runned. The default name of the output directory is 'output/'. In such a directory there are the intermediate products relative to the

Hessian matrix calculation (see section 6.4 for a partial description) and a further subdirectory name 'lev_NNNNN/' with NNNNN is the value of the threshold adopted at runtime with the format F9.5. After a successful run of the package it is possible to find under the path 'output/lev_NNNNN/' the main outputs:

- The full unfiltered catalog, with default name 'catalog_filaments.dat'. The catalog format follow the IPAC table format defined at http://irsa.ipac.caltech.edu/applications/DDGEN/Doc/ipac_tbl.html.

- The catalog filtered by the length of the identified structure (see Section 5.1) with default name 'catalog_filaments_selected.dat'. The catalog format follow the IPAC table format.

- A file in XDR (eXternal Data Representation) format saved by IDL SAVE procedure where two IDL variables computed at runtime are stored. Those variables are:

- `ALLBRANCHSTR` and IDL structure containing the detailed informations for all the branches identified by the algorithm.

- `ARRAYBRANCHES` an array of pointers, each of which points to a structure similar to any element of `ALLBRANCHSTR`. However, further data are added to the pointed structure, in particular, the branch position and the estimates of the filament intensity profile and its background along the branch itself. Those quantities have different lengths for each branch, thus they require a different representation through pointers.

The default name of the saved file is 'catalog_filaments_branches.xdr' and can be restored under IDL environment with the IDL procedure RESTORE.

- For each identified RoI the package creates a directory named 'FilXXXXX/', where XXXXX is the ID number associated to the filamentary region. Note that XXXXX is not a consecutive number, since the numbers are associated to all the RoIs after the thresholding including the one that are filtered out at the detection stage (see Section 5.1). In each directory there are several files storing intermediate products computed during the normal run of the package. The majority of those products can be used by advancer user to reproduce the analysis of the filamentary region without the requirement to run the package again on the whole map.

- A single directory 'RegionFiles/' containing the region files in ds9 format for data visualization. For each RoI identified by the number XXXXX there might be up to five different region files:

'Region_ROI_lineXXXXX_spine.reg' - A region file tracing the main spine of the filament defined as the longest path that can be build along the MST connections. Such main spine is also the one adopted as measure of the length of the filament. The file is written in ds9 format.

'Region_ROI_lineXXXXX_branches.reg' - An optional region file tracing all the various branches identified from the corresponding RoI after applying the thinning operator. In the case where there is a single branch, obviously corresponding to the main spine, this file is not written. The file is written in ds9

format.

'Region_ROI_newpolyXXXXX.reg' – A region representing the portion of the map encompassed within 1 FWHM (the filament width) with respect to the main spine. The file is written in ds9 format.

'Region_ROI_polyXXXX.reg' - – A region representing the portion of the map encompassed within 1 FWHM (the filament width) with respect to all the branches. The file is written in ds9 format.

'Region_ROI_listXXXX.reg' - A region representing the portion of the map encompassed within 1 FWHM (the filament width) with respect to all the branches. The file is written as an ascii file with two columns containing the coordinates for the border of the region

Warning. The coordinate system adopted for those region files is written in the header as 'wcs'. It has been noted that for some versions of ds9 there is a wrong interpretation for the coordinate system. In such cases it is better to replace the 'wcs' with the coordinate system expressed in the map header (i.e. 'galactic', 'fk5').

- Two set of region files holding all the output of the finally identified regions included in the extracted catalogs. The files are named 'All*.reg' for all the filaments included in 'catalog_filaments.dat' while 'Sel*.reg' for the filaments included into 'catalog_filaments_selected.dat'. Those files are build up from the single region files in the 'RegionFiles/' directory. Warning! It has been reported that the format of those files might not work in some version of ds9, depending on the shell and the operative system adopted (know bug that should be fixed in the next delivery). Four type of files are written:

  'AllBranches.reg' and 'SelBranches.reg' – Tracing all the various branches.
  'AllSpine.reg' and 'SelSpine.reg' – Tracing all the main spine (see above).
  'AllRoI.reg' and 'SelRoI.reg' – Tracing the defined RoIs around all the filament spines.
  'AllFilament.reg' and 'SelFilament.reg' – A merge file from the three above.

- Two FITS maps representing the residual background emission. The filamentary contribution has been subtracted at run time. Here 'namemap' stands for the name of the map file. There are two different approach in estimating the background:

  File 'namemap_background1.fits' - a planar function is fitted from the pixels along the borders in each branch subregion. The fitted plane is replacing the pixel values inside each subregion.

  File 'namemap_background2.fits' – The linear interpolation across the cross spine direction is fitted from the pixels along each branch subregion (see Section 5.1). This method has been proven to give a more reliable estimation for the background emission.

- Several binary mask files. Here 'namemap' stands for the name of the map file:

'namemap_branches.fits' – a binary mask file where all the identified branches are labeled by their ID in the ARRAYBRANCHES structure.

'namemap_finalmask.fits' – a binary mask file representing the portion of the map encompassed within the estimate width of the filament from its spine (all the branches are considered). Each subregion is labeled with ID of the corresponding filament in the 'catalog_filaments.dat'.

'namemap_RoISegmbranches.fits' – a binary mask file where the same portion of the map covered by the 'namemap_finalmask.fits' are instead labeled using the ID from the ARRAYBRANCHES structure.

'namemap_maskbranches.fits' – an auxiliary binary mask file where the RoI are draw starting from each branch and associating only the pixels within the estimate width from each branch. Note: This mask is different from the 'namemap_finalmask.fits' where a single value for the width (the median of all the branches identified in the spine) is assumed for all the branches.

6. Filamentary Structure extraction package user manual

6.1. System needs

The package requires IDL ver 7.0 or a newer version to run. Unzip the package in a directory accessible to your IDL path for a successful run. The package has shown to work smoothly in IDL ver 7.1 enviroment, but some issues have been reported in the case of IDL 8.0 and 8.1 due to ambiguity between the internal system and the package routines. Moreover, few scripts belonging to external libraries are not supported anymore by such IDL versions, implying possible interruptions in the package computation flow. Most of those incompatibilities have been corrected and the scripts have been updated, but bugs can be still present, thus it is recommended to run in the IDL 7.1. The IDL Astronomy User's Library, 'astrolib', not included in the delivered file but available at http://idlastro.gsfc.nasa.gov/ , and the 'Minimum Spanning Tree' library, included in the tar file, are needed for a successful run of the complete package.

It is highly recommended, despite not required, to have also installed in a directory accessible to the IDL path the photometry code for compact sources detection and extraction CuTEx version 0.995 or above. The astronomical imaging and data visualization application SAOImage ds9 is suggested to produce in a short time the input region file with the correct format required for a safe run of the package.

Please note that an installation procedure is not supplied with this release so you need to manually edit the script file 'fildet.log' in the directory 'detect_filaments/'. Replace in such script file the entry '/Users/Pro/' with the path location where you unzipped the package.

6.2. Configuration files

The region file has to be defined by the shape 'polygon' (see ds9 manual) and saved with the same coordinate system of the input map header in degree format. The default name for the input region file is 'Region.reg', but a different name can be defined when launching the package (see Section 6.3).

An example of such a region file is:

```
# Region file format: DS9 version 4.1
# Filename: l059_nh2_1e21.fits
global color=green dashlist=8 3 width=1 font="helvetica 10 normal roman" select=1
highlite=1 dash=0 fixed=0 edit=1 move=1 delete=1 include=1 source=1
galactic
polygon(60.032483,1.1693601,57.718318,1.1682899,57.721088,-1.1096366,60.032376,-
1.1107053)
```

It is important to note that the region file has to cover a portion of the map where there the pixel values are significative. In fact, the 'polygon' defines the portion of the map to be analysed by the algorithms to compute the eigendecomposition of the Hessian matrix of the intensity map. Such a region should be draw avoiding the border of the map and reducing the effective coverage to be analysed by roughtly ~5 pixels. This requirement is needed to avoid possible undesiderable side effects nearby the border of the intensity map that might be introduced by the discrete second derivative operator.

6.3. How to run

To run the package it is suggested to copy your map FITS and the region file (see Section 5.3 and 6.2) in a working directory. Launch your IDL session in the working directory and compile all the routines needed launching the script:

```
IDL> @fildet.log
```

After a correct compilation of the routines, the package is ready to be launched. If the map FITS file has not been yet analysed with a previous run of the package it is needed the computation of the derivatives and of the eigendecomposition of the Hessian Matrix. This is instructed to the package by the following launching string:

```
IDL> out = run_fildet(namemap, threshold, /derivate)
```

Where **namemap** is a string with the name of the map FITS file and **threshold** is the chosen threshold value. Note that the keyword **derivate** will always trigger the routines to compute the derivatives and eigendecomposition, but how the package work this is not a requirement after a first successful run. After the run the outputs obtained with the chosen threshold will be stored in the default directory 'output/lev_NNNNN/', where NNNNN is the value of the above threshold **threshold** (see Section 5.4).

Further runs of the package with a different threshold level can be launched by :

```
IDL> out = run_fildet(namemap, threshold)
```

Note that the above launching strings will run the package with the default settings. The description of the major parameters and keywords that can be manually set by the user are described below.

Parameter `clearpix`:

The derivate operator is by construction a high-frequency passband filter, with a slightly amplification of the pixel-to-pixel noise present in the input map by derivate and eigendecomposition. Thus, the simple thresholding produces several RoIs that are due to fluctuations induced by the input noise. Those spurious RoIs are generally filtered out removing all the RoIs composed by a small number of pixels. The cut value for such a filtering is defined by the input parameter clearpix, setted as default to the value of 15 pixels. To adopt a different value with respect the normal run use the launching string:

```
IDL> out = run_fildet(namemap, threshold, clearpix)
```

where **`clearpix`** is an integer defining the new adopted value. Note that **`clearpix`** is a parameter and not a keyword. As default the package store the simple thresholded mask in the FITS file 'out_RoI_order.fits' and the initially filtered (by region size) mask in file 'out_RoI_order_removed.fits' (see also Section 6.4).

Parameter `catalogname`:

To change the output file name for the two filament catalog and the branch XDR file (see Section 5.4). Note that to trigger the use of such a parameter the parameter **`clearpix`** has also to be defined in the launching string even in the case that **`clearpix`** is left to the default value:

```
IDL> out = run_fildet(namemap, threshold, clearpix, catalogname)
```

Where **`catalogname`** is a string that as default is setted to 'catalog_filaments.dat'. Such a choise will produce the output file described in Section 5.4.

Warning! Each run of the package with the same threshold (thus stored in the same directory tree) will result in deleting the previous output files. Use the **`catalogname`** parameter If multiple runs of the package with the same threshold level are needed for testing / comparison purposes.

Keyword `local`:

It is suggested to run the code with the keyword **`/local`** switched on. In such a case the threshold will not be considered as an absolute value in the units of the second derivative map (i.e. if the units of the map are MJy/sr,

the second derivative is proportional to MJy/sr/pixel^2 or in other words the variation of the gradient of the intensity field), but rather a fixed costant times a minimum threshold level evaluated in the local surrounding of each pixel. In other words the launching string:

```
IDL> out = run_fildet(namemap, threshold', /local)
```

will produce a different output with respect to the former one. In the first case all the pixels of the λ_1 map with an absolute (negative) value higher than threshold will be selected by the thresholding process. In the second case the package will compute a sigma map, save it in the output directory as 'sigmaframe.fits' and it selects all the pixels whose λ_1 (in absolute value) is `threshold'` times higher than the one stored in the same position of the sigma map. The minimum threshold level depends on the fluctuations of the λ_1 and are map dependent. Simulations of filamentary structures on Herschel maps has allowed to empirically determine a good estimate for such a fluctuations. For each pixel the adopted minimum threshold is evalutated in subregions 61 x 61 pixels wide centered at the pixel position and given by the lowest value between the 3-σ clipping standard deviation of λ_1 and the minimum value of λ_1 for which the absolute ratio $\lambda_1 / |\lambda_2| > 3$.

Keyword `rmvroi`:

As explained in the Section 5.1, at the detection stage the filamentary candidate regions are identified between all the RoIs on basis of their *ellipticity* and their *filling factor* (see Section 5.1 for a more complete description). However, in this preliminary release of the package those criteria are applied to the final catalog after all the RoIs are extracted by the routines that estimate the filament properties. This means that the more robust catalog is defined after processing all the RoI, even the ones that have a lower probability to be filamentary. Such an approach is more intense in terms of computation time.

The keyword `rmvroi` is used to define the cut off values for ellipticity and filling factor and is defined as a single float value or a 2-elements vector, [e,f]. In the case the input is a single value only the criteria for ellipticity is used to filter out the RoIs during the filament detection. On the other hand, the first element of the vector represent the maximum accepted ellipticity *e* for the RoI, while the second is the minimum coverage fraction *f* of the RoI with respect to the elliptical fit (see Section 5.1) that define the filament. The launching string is:

```
IDL> out = run_fildet(namemap, threshold, rmvroi=[e,f])
```

or

```
IDL> out = run_fildet(namemap, threshold, rmvroi=e)
```

Where `e` and `f` are the values for ellipiticty and filling factor with acceptable inputs that between 0 and 1. The default value are [0.75, 0.85]. Such a choise means that all the regions that are fitted by an ellipse with *e* > 0.75 are filtered out as long the ratio between the RoI surface area and the area of the fitting ellipse *f* > 0.85. Roundish regions, *e* > 0.75, with a *f* < 0.85 and elliptical regions, *e* < 0.75, are instead analysed by the package. To indicate to the package to not adopt only one of the two described filtering applied on the RoIs replace the input variable with -1.

Keyword `cutlength`:

As side filtering on the whole catalog of filaments it has been introduced a cut off on the structure lengths. Such a criteria has been introduced to avoid the contamination by elongate small clumps identified on the FIR/submm maps, like the Herschel ones. The filtered catalog is stored in a different file (see Section5.4) and an additional mask file is saved as intermediate product (see Section 6.4). The default value for the cut off length is 12 pixels, that for a Nyquist sampled map corresponds roughly to 4 times the PSF. Any structure whose main spine is shorter than the cut off length is removed in the filtered catalog. To change the cut off adopted for the filtered catalog use in the launching string:

```
IDL> out = run_fildet(namemap, threshold, cutlength=l)
```

Where `l` is the new cut-off in pixels.

Keyword `smooth`:

Due to the amplification of the noise produced by the derivate maps, it is often required to adopt a smoothing of the input map. However, typical smoothing filters have the drawback to dampen at the same time the noise and the signal, in this case the edges of the filaments lowering the possibility to detect the latter. The filtering with special functions like the Perona-Malik anisotropic diffusion filter is present in the code, but it is not yet accessible to the user due to further testing. For the current released version it is possible to activate smoothing of the input map before the computation of the derivates. The adopted kernel for the filtering is an adapted Gaussian function with a FWHM of 1.5 pixel, decreasing the impact of the pixel-to-pixel noise (see Schisano et al. 2014). To activate the smoothing use in the launching string:

```
IDL> out = run_fildet(namemap, threshold, /smooth)
```

Note that smoothing the input map will decrease also the values in the derivate and eigenvalues map, thus a different value of the threshold will be needed to highlight the structures with respect to the case when the detection is performed on the unsmoothed map. As default the smoothing is not active.

Keyword `ndilate`:

The detection and thresholding algorithms are able to trace well the portions of the map where there is a quick change of the variation of the intensity contast. Nevertheless the filament radial profile can extend further than the position where there is a change in the curvature. Such a correction is crucial when the background has to been estimated, see Section 5.1. The package currently works by determining the pixels along the filament borders and adopt those for the linear interpolation of the background contribution in the cross-spine direction. As default it is adopted that the pixels that lies 1-pixel further away than the intial RoI have to be included into the group of pixels belonging to the filament border, however, such a value can be setted by the user through

the keyword `ndilate`. To extend the size of the portion of the map adopted as filament border use the launching string:


```
IDL> out = run_fildet(namemap, threshold, ndilate=nd)
```


Where `nd` is the maximum distance a pixel can have from the RoI to be included into the filament border. In the case of the initial catalog presented in the document VIALACTEA D1.5, it was estimated that for a better background estimation the border has to be extended up to 3 pixels.

Keywords: `dir`, `prefix`, `regiondir`, `regionfile`:

Those keywords are only used to change the default name conventions for the inputs or outputs (see Section 5.3 and 5.4). All those keywords accept a string as input.

The keyword `dir` changes the output directory (default 'output/') where the intermediate and final output of the package are stored. In the output directory it is created the subdirectory tree described in Section 5.4.

The keyword `prefix` changes the filename of the intermediate products of the derivate and Hessian eigendecomposition. The adopted default is 'out' (see Section 6.4 where the default filenames are shown).

The keyword `regiondir` changes the subdirectory where all the region files are stored. As default is adopted 'RegionFiles/'.

Finally the user can change then name of the input region file with the keyword `regionfile`. The default name is 'Region.reg' (note the capital case). However, even when the keyword `regionfile` is adopted the package will merely copy the input region file into the file 'Region.reg' due to the adopted convention in all the scripts.

Keep in mind that for a single run it is possible to activate at the same time several of the above described keywords.

6.4. Intermediate outputs

As an intermediate output the package writes the results from the eigendecomposition of the Hessian matrix in the directory 'output/' and the binary masks obtained after the detection algorithms in the directory 'output/lev_NNNNN/', where NNNNN represents the adopted threshold.

- The results from the eigendecomposition are

  'out_evec.fits' - a cube with sizes N_1 x N_2 x 4, where in each pixel are stored the x- and y-component of the eigenvectors A_1 and A_1. Here N_1 and N_2 are the x- and y- dimension of the input map.

'out_eval.fits' - a cube with sizes N_1 x N_2 x 2, where in each pixels are store the eigenvalues λ_1 and λ_2.

'out_eval_large_mask.fits' – a file with the values of the largest eigenvalue λ_2 in the area defined by the input region.

'out_eval_large_mask.fits' – a file with the values of the smallest eigenvalue λ_1 in the area defined by the input region.

'out_eval_ratio.fits' – a file with the ratio between λ_1 and λ_2 in the area defined by the input region.

- Binary masks representing the output of the detection algorithms:

  'out_RoI_order_removed.fits' – Binary mask obtained from the thresholding of the eigenvalue map and initial filtering based on the size of the identified RoI. Each RoI is labeled by an ID number. Numbering is not progressive since it is performed before the filtering.

  'out_RoI_order_removed_cleared.fits' – Final binary mask tracing the thresholded pixels belonging to the surving RoI after all the selection criteria to identify filamentary region candidates are applied. All the region presented in the catalog 'catalog_filaments.dat' are present here

  'out_RoI_order_removed_cleared_cutlengthYYYYYY.fits' – Same as above but further filtered from region shorter than the input cut length. The YYYYYY represent the length (in pixel) of the shorter filament candidate still present in the 'catalog_filaments_selected.dat'.

Furthermore, the IDL session after the eigendecomposition of the map is stored in a file 'outSession.sav'. This file is used just to skip the intensive derivative and eigendecomposition computation and can be safely deleted.

7. **Filamentary Structure extraction package V0 delivery notes**

In this section it is reported a general description of the package outputs. In particular it is described the columns of the filament catalogs and of the of structure stored in the XDR file. Both filament catalogs follow the general IPAC table format. The columns are arranged as:

Column Name	Format	Unit	Short Description	Note
N	LONG		Progressive numbering in the catalog	

ID	LONG		Identification number of the filament corresponding to the RoI in the labeled masks	
LENGTH	DOUBLE	PIXEL	Length of the main spine defined as the longest path covered along the MST	
AREA	DOUBLE	PIXEL	Coverage Area of the identified thresholded RoI	
PERIMETER	DOUBLE	PIXEL	Perimeter of the identified thresholded RoI	
AXIS_MIN	DOUBLE	PIXEL	Length of the minor axis of the ellipse fitting the thresholded RoI	
AXIS_MAX	DOUBLE	PIXEL	Length of the major axis of the ellipse fitting the thresholded RoI	
ELLIPTICITY	DOUBLE		Ellipticity defined as ratio between AXIS_MIN and AXIS_MAX	
FILLINGFACT	DOUBLE		Filling Factor defined as ratio between the RoI and the fitting ellipse areas	
TOTAL	DOUBLE	MAP UNIT	Total emission inside the RoI	
TOTALBCK1	DOUBLE	MAP UNIT	Total emission inside the RoI where it is subtracted an estimate of the background done at the detection stage from a planar fit on the whole RoI area from the pixels on the border (1 pixel away)	Obsolete: To not be used
TOTALBCK2	DOUBLE	MAP UNIT	Total emission inside the RoI where it is subtracted an estimate of the background done at the detection stage from a planar fit on the whole RoI area from the pixels on the border (2 pixel away)	Obsolete: To not be used
TOTBACKFLAT	DOUBLE	MAP UNIT	Total emission inside the RoI where it is subtracted an estimate of the background given as the mean constant value measured along borders.	Obsolete: To not be used
NPIXELS	DOUBLE		Number of pixels in the RoI	

CHECKSPINE	INT		Check flag for a successful recovery of the spine	
WIDTH	DOUBLE	PIXEL	Mean filament width given as mean of the values along all the branches	
WIDTHERR	DOUBLE	PIXEL	Standard Deviation of the width from all the single branch measures	
NOTES	STRING		To be filled	
TOTBACKPLANE	DOUBLE	MAP UNIT	Total emission from the background in the RoI. Background estimated as combination of different planes one for each branch subregion.	
FILPLANE	DOUBLE	MAP UNIT	Total emission from the filament in the RoI, background subtracted. Background estimated as combination of different planes one for each branch subregion.	
TOTBACKINTERP	DOUBLE	MAP UNIT	Total emission from the background in the RoI. Background estimated by the linear interpolation along the cross-spine direction to the border pixel fluxes	
FILBACKINTERP	DOUBLE	MAP UNIT	Total emission from the filament in the RoI, background subtracted. Background estimated by the linear interpolation along the cross-spine direction to the border pixel fluxes	

The two IDL variables, ALLBRANCHSTR and ARRAYBRANCHES, saved in the XDR output file described in Section 5.4 have a similar format. It is possible to access to the two variables from IDL with the routine RESTORE by typing:

```
IDL> restore, filename='catalog_filaments_branches.xdr'
```

Below it is reported a brief description of the format of the ARRAYBRANCHES variable. The ALLBRANCHSTR is similar but lack of few elements in the structure.

Structure Tag	Format	Unit	Description	Note
IDFIL	LONG		Identificative number of the corresponding filament to which the branch is associated	Present also in `ALLBRANCHSTR`
NBRANCH	LONG		Progressive number to each branch valid only within the same filament	Present also in `ALLBRANCHSTR`
IDBRANCH	LONG		Identificative number of the branch corresponding to the one in the label mask.	Present also in `ALLBRANCHSTR`
X	ARRAY of FLOAT size depend on branch length		X-coordinate of the branch pixels on the map. The coordinates are expressed as the column position on the matrix representing the map. The size of the vector depends on the length of the branch.	
Y	ARRAY of FLOAT size depend on branch length		Y-coordinate of the branch pixels on the map. The coordinates are expressed as the row position on the matrix representing the map. The size of the vector depends on the length of the branch.	
LIMITS	4 ELEMENTS ARRAY [2 x 2]		X and Y Positions of the two Vertices or Nodes connected by the branch.	Present also in `ALLBRANCHSTR`
FLAGLIMIT	STRING		Flag tagging the positions connected by the branch as Vertex or Node.	Present also in `ALLBRANCHSTR`
LENGTH	FLOAT	PIXEL	Length of the branch.	Present also in `ALLBRANCHSTR`
WIDTH	FLOAT	PIXEL	Width of the filament along the considered branch. The width is estimated as a Gaussian fit in the direction orthogonal to the one of the branch.	Present also in `ALLBRANCHSTR`
ERRWIDTH	FLOAT	PIXEL	Error on the width of the filament.	Present also in `ALLBRANCHSTR`
AREASUBROI	FLOAT	PIXEL	The area of the subregion associated to the branch. The initial RoI is segmented into such subregions by associating	Present also in `ALLBRANCHSTR`

			each pixel to its closest branch.	
TOTALBCK1	FLOAT	MAP UNIT	Total emission from the background in the suRoI. Background estimated by fitting a plane to the flux of the pixels belonging to the branch border.	Present also in `ALLBRANCHSTR`
TOTALBRANCH1	FLOAT	MAP UNIT	Total emission from the filament in the suRoI. Background estimated by fitting a plane to the flux of the pixels belonging to the branch border.	Present also in `ALLBRANCHSTR`
MEANBCK1	FLOAT	MAP UNIT	Mean emission from the background in the suRoI. Background estimated by fitting a plane to the flux of the pixels belonging to the branch border.	Present also in `ALLBRANCHSTR`
MEDIANBCK1	FLOAT	MAP UNIT	Median emission from the background in the suRoI. Background estimated by fitting a plane to the flux of the pixels belonging to the branch border.	Present also in `ALLBRANCHSTR`
TOTALBCK2	FLOAT	MAP UNIT	Total emission from the background in the suRoI. Background estimated by the linear interpolation along the cross-spine direction to the border pixel fluxes.	Present also in `ALLBRANCHSTR`
TOTALBRANCH2	FLOAT	MAP UNIT	Total emission from the filament in the suRoI. Background estimated by the linear interpolation along the cross-spine direction to the border pixel fluxes.	Present also in `ALLBRANCHSTR`
MEANBCK2	FLOAT	MAP UNIT	Mean emission from the background in the suRoI. Background estimated by the linear interpolation along the cross-spine direction to the border pixel fluxes.	Present also in `ALLBRANCHSTR`
MEDIANBCK2	FLOAT	MAP UNIT	Median emission from the background in the suRoI. Background estimated by the linear interpolation along the cross-spine direction to the border pixel fluxes.	Present also in `ALLBRANCHSTR`
DIRECTION	FLOAT	DEGREE	Direction of the filament	Present also in

			estimated by the angular coefficient of the linear fit to the branch. The linear fit is performed on group of 3 consecutive pixels along the branch. The direction is defined as the mean value of all those coefficients.	ALLBRANCHSTR
ERRDIRECTION	FLOAT	DEGREE	Error on the direction of the filament. The error is defined by the standard deviation of all the coefficients.	Present also in ALLBRANCHSTR
XYDIRECTION	2 ELEMENTS ARRAY	DEGREE	The linear fit for the direction estimate is evaluated 'Y vs X' positions and 'X vs Y' positions. The two angular coefficient (modulus 180 degrees) are stored here.	Present also in ALLBRANCHSTR
RAD	ARRAY of FLOAT size depend on branch length		Curvilinean abscissa along the branch starting always from a node or the souther-eastern one.	
VALUES	ARRAY of FLOAT size depend on branch length	MAP UNIT	Map intensity values along the branch, ordered following the curvilinean abscissa direction.	
BACKVALUES	ARRAY of FLOAT size depend on branch length	MAP UNIT	Background intensity values along the branch, ordered following the curvilinean abscissa direction.	
MAXVAL	ARRAY of FLOAT size depend on branch length	MAP UNIT	Max intensity values along the branch, ordered following the curvilinean abscissa direction. In each position the max value is obtained comparing with the closest neighbours pixels along the direction orthogonal to the spine.	
MEDVAL	ARRAY of FLOAT size depend on branch length	MAP UNIT	Median intensity values along the branch, ordered following the curvilinean abscissa direction. In each position the median value is obtained from the neighbours pixels along the direction orthogonal to the spine.	
MEDBACKVAL	ARRAY of FLOAT	MAP UNIT	Median background values	

	size depend on branch length		along the branch, ordered following the curvilinear abscissa direction. In each position the median value is obtained from the neighbours pixels along the direction orthogonal to the spine.	
STDBACKVAL	ARRAY of FLOAT size depend on branch length	MAP UNIT	Standard deviation of the background values along the branch, ordered following the curvilinear abscissa direction. In each position the standard deviation is determined from the closest neighbours pixels along the direction orthogonal to the spine.	

8. Filamentary Structure extraction package V1 delivery notes

The packge version 1 does not introduce any major changes in the main routines and procedures released with the version 0. Few minor changes to allow compatibility with IDL 8.3 have been introduced. Furthermore, small fixes in the flows during the processing have been implemented. The outputs are kept the same as the version 0, but the data structure within the package has been rearranged in order to organize the information in a easier way to be organized in VO data format. However, for this stable version, named 0.5b, no differences are found in the final products of a single run of the extraction package. Finally few routines have been added to work with consecutive mosaics like the one described in D1.6.

9. Filamentary Structure extraction package V2 delivery notes

The package version 2 still does not introduce major changes in the main routines and procedures released with previous versions. Few bugs in the routines that occasionally alters the correct execution of the package has been corrected. This release, designated as version 0.6, mostly introduce new features to the main call function for a better management for the single run and for the housekeeping of outputs.
It is now possible to resume a run of the algorithms that have been interrupted for crash or unexpected returns of morphological operators. In particular, artifact regions whose shapes are not correctly treated by ``thinning'' operator and introduce branches that cannot be connected properly might stop a single running process. Since the runtime for the algorithm is strongly dependent on the number of detected candidates and can ramp up to several hours a single crash introduced by an artifacts can result in a severe loss of time.
For such reason a new keyword has been introduced called '`resume`' that if set will allow the main algorithm to recover the run and continue the analysis for the remaining regions. The new keyword accepts a long integer number as input and instructs the algorithm to jump to the region labeled with the corresponding value.

It is also included a dedicate script, called 'run_fildet_extmask.pro', derived from the main program in order to use external masks containing the filament footprints estimated by other software tools. With such a script it is able to estimate the filament parameters from RoIs determined by other algorithms.

Furthermore, a new header to the output has been added to keep track of the set of parameter adopted for the extraction. A double line header is included in the output catalog reproducing completely the launch string adopted to produce that particular catalog.

Finally, it has been included a set of procedures to extract the main results for branches and nodes from the XDR file and write them in IPAC table format. These ancillary IDL procedures are included in the package as utilities for handling the products and are name 'selbranches.pro', 'makecatalognodes.pro',

10. Filamentary Structure extraction package V3 delivery notes

The package version 3 does not substantially change the main routines and procedures with respect to previous versions. The cleaning from bugs has been continued and the package revealed to be quite stable for the use on general maps, in particular it is dedicated to the Herschel maps. Several cases of error are now handled by the code itself and in some cases new messages are printed on screen to allow an easier interpretation of the crashing reason.

This version includes a major revision of libraries needed to run the package. Obsolete libraries have been replaced with newer versions or with new dedicate libraries. All the new routines have been included in the package. A simple installation script copies and untar the package in the directory **~/IDL/detect_filament/** and prepares a file that allows to compile all the necessary routines in IDL environment. The new file replacing the old **fildet.log** is now called **fildet.list.**

The filament finder algorithm is still optimized to work in IDL environment, since it requires native IDL procedure for the morphological analysis (dilatation, erosion and thinning of Region of Interests).