Heliophysics Integrated Observatory

Project No.: 238969
Call: FP7-INFRA-2008-2

Heliospheric Feature Catalogue

Version 2.1

Title:	**Heliospheric Feature Catalogue**
Document No.:	HELIO_OBSPM_R2_nn_RP_HFC_2.1
Date:	20 August 2011
Editor:	**J. Aboudarham**, OBSPARIS
Contributors:	Xavier Bonnin, Christian Renié, Nicolas Fuller (OBSPARIS), David Perez-Suarez (TCD)
Distribution:	Project

CAPACITIES

e-infrastructure

Revision History

Version	Date	Released by	Detail
1.0	9/5/10	J. Aboudarham	First version
1.1	14/7/10	J. Aboudarham	Add time tracking table description
2.0	1/10/10	J. Aboudarham	Update tables according to the need for data without pre-processing, and define minimum information needed by HFC. PP_setup and PP_output are merged.
2.1	20/08/11	X.Bonnin	Update Observatory and FRC_Info. PP_Information table is renamed to PP_Output.

Note: Any notes here.

Contents

List of acronyms

- **DB** : Database
- **DoW**: Description of Work (available at: http://www.helio-vo.eu/internal/documents/HELIO_DoW_Final_090904.pdf)
- **EGSO**: European Grid of Solar Activity: program developed under European FP5. See http://www.egso.org
- **FR** : Feature Recognition
- **GDL**: Gnu Data Language. Open source version of IDL (http://gnudatalanguage.sourceforge.net/)
- **HEK**: Heliophysic Events Knowledgebase (http://lmsal.com/hek/)
- **HFC**: Heliophysics Feature Catalogue
- **IDL**: Interactive Data Language. Programming language specialised in image processing widely used in the astronomical community (http://www.ittvis.com/ProductServices/IDL.aspx).
- **IVOA** : International Virtual Observatory Alliance (http://www.ivoa.net/)
- **JRA**: Joint Reasearch Activity
- **SDO:** Solar Dynamics Observatory (http://sdo.gsfc.nasa.gov/)
- **SQL**: Standard Query Language
- **TBC**: To Be Confirmed
- **TBD**: To Be Defined
- **WP**: WorkPackage

I. Introduction

The scope of this document is to describe the Heliospheric Feature Catalogue (HFC).
HFC will be a place to store information concerning features, but also a place that can be accessed through HELIO, or directly by users. So it has to be in accordance with HELIO standards.

The constraints HFC will have to manage concern Input/Output. It will receive information from various detection codes. This information should be stored in the HFC. Then these stored data can be extracted through standard DB queries, via a dedicated user interface, directly by the HELIO core software, possibly by (or for) the SDO HEK.

Several kind of information may be useful to store in the HFC in order to take full advantage of it. It can be separated in the following sections:

* *Pre-processing information*: In some cases pre-processing of the data is necessary in order to achieve automatic detection of features. For instance it could be to centre a full Sun image (otherwise feature position may be wrong), or remove interferences from a frequency/time diagram (a dynamic spectrum). If a standard pre-processing code is applied to the data, at least the name of the code should appear in this section. In many cases, pre-processing will be limited to tranform data in a format that is suitable for the Input of the features extraction code. We shall generalize this section as a **Preprocessing table**.
* *Feature extraction code information*: Each feature is obtained via an extraction code. We must know the name of the code, and probably some parameters of the code are tuned in order to optimize detection for a specific dataset. Those parameters should also appear in this code section. We shall generalize this section as a **Processing table.**
* *Original data information*: This section will contain a set of information describing the original data from which features were extracted (instrument name, observing date, ...). We shall generalize this section as a **Original table.**
* *Feature description*: This section will contain a full description, different for each feature, that characterize it in order for the scientist which extract this information to be able to derive science from it, and to make advances queries. A chain code or a thumbnail may join morphological information. We shall generalize this section as a ***Feature*** **table**. (each feature will have its own table with its own name)
* *Feature follow up in time*: This section will join features that are the same one, observed at different time, on different images. This could join features extracted from the same instrument as well as those extracted from different instruments providing the same kind of observation. We shall generalize this section as a **Tracking table.** Each feature table will be associated, when necessary, with one tracking table.
* *Feature local connection*: This section aims at connecting various extracted features that are spatially closed. Features are extracted from different instruments, but our general knowledge, or the study of the HFC leads to the assumption that they are spatially connected. It could concern photospheric and chromospheric features, for instance; or magnetospheric and ionospheric events as well. We shall generalize this section as a **Local table**

* *Feature distant connection*: This section is probably to be left blank! As distant connection is very model dependant, it is more likely that this interaction will be calculated in real time, based on the model, and background information available, that the user considers. But on the long term, one can imagine that scientific studies that succeed in showing long distance connection between features can provide information that could be entered in this section. Which will enrich, as science will progress. We shall generalize this section as a **Propagation table**

It should be noticed that tracking may have no meaning for some features. For instance, many heliospheric features already own (as their characteristic) information on spectral and time drift.

Connections between the various parts of the database are as hereafter:

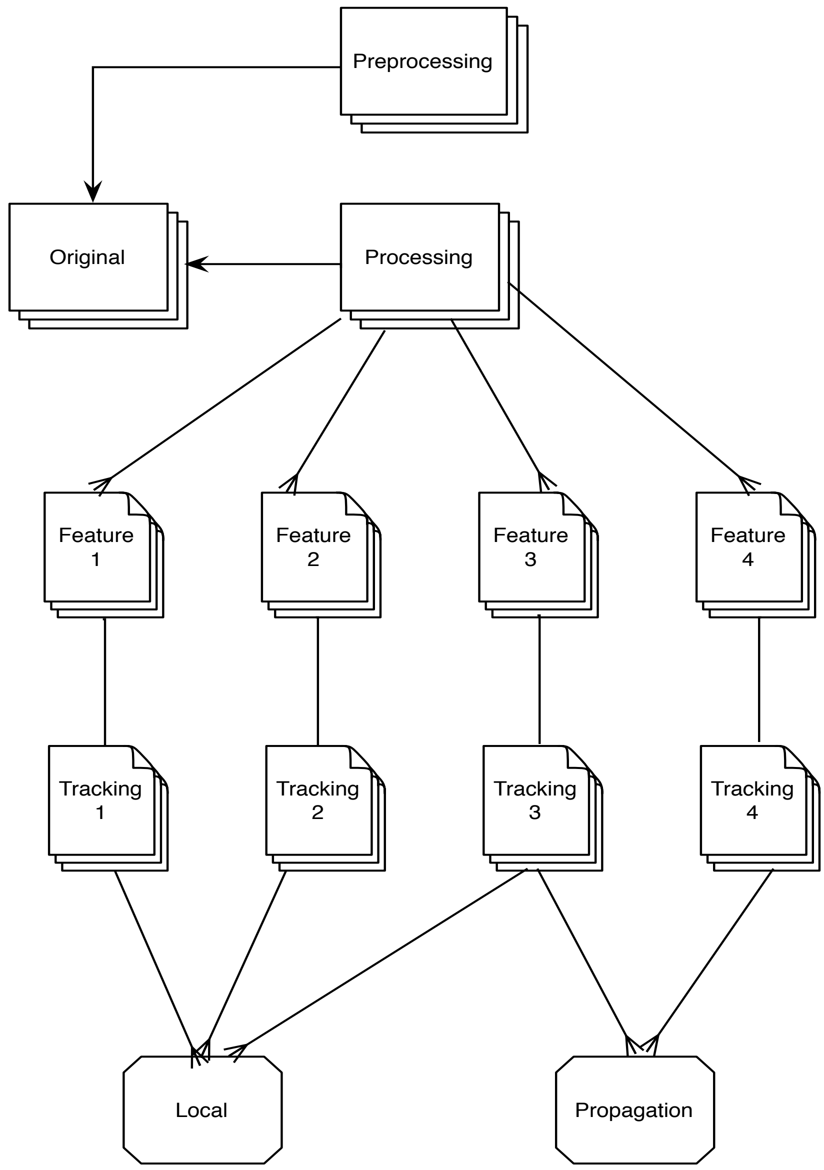


Figure 1: The various elements to take into account in the database.

Relevant Documents

- HELIO Description of Work (DoW): http://www.helio-vo.eu/internal/files/HELIO_DoW_Final_090904.pdf
- EGSO feature description: http://www.egso.org/documents/Features_ParametersV2.3.pdf
- Image Processing Document: https://grid.ie/helio/wiki/HelioFGFeatureDetection?action=AttachFile&do=get&target=FeatureDetectionReport.pdf
- HELIO Release Strategy version 1: http://www.helio-vo.eu/internal/documents/HELIO_Release_Strategy_20091228.pdf
- IVOA VOEvent recommendation: http://www.ivoa.net/Documents/PR/VOE/VOEvent-20060810.html
- IVOA VOTable format definition: http://www.ivoa.net/Documents/VOTable/

II. EGSO legacy

EGSO built a database, the Solar Features Catalogue (SFC). The organization of this database is described in the Features Parameters document mentioned above.
Very roughly, let's say that SFC was mainly made of 3 tables corresponding to processing, original image description (mainly date, wavelength and observatory) and extracted feature description (this last table existing for each feature we worked on).

III. HFC I/O

1) Input from detection codes

Detection codes should provide all the necessary information allowing to fill in the tables mentioned in the introduction, i.e. original data used, pre-processing, detection code, and, of course, feature description. The feature description table will be unique for each feature. It means that at list a small document explaining the parameters extracted from the feature should exist. Otherwise, no standardization of the information could ever be done!
The content needed by the first three tables mentioned above is described in chapter V.
Not all the field may need to appear, but a template for features description will be provided.
The other tables connecting detected features may be filled in later on, as the information comes from different codes that will be written after the detection codes.

2) Standard output

This output can be used directly buy any user. It corresponds to a standard SQL query. It may be used directly by a code, which recovers the data and format them, or make, for instance statistics on it, as well as by a web interface that helps to query the HFC (typical php/MySQL connection).

3) VOTable output

This output is to be used by the HELIO core software.
The exact format needed by the HELIO developers team will be updated in following versions of this document. But it is likely that standard SQL query will allow to build an xml file following the HELIO core system standards.

4) VOEvent output

This output is to be used by the HEK. It is not a first priority, but we have to keep it in mind.
VOEvent is an xml-based standard format introduced by IVOA. It allows to describe transient events. Before providing this format, we shall first have to check whether it can be applied to all the features we'll have to manage.

IV. Features' useful information

In this document, the following *definitions* are used to describe the various elements of the whole process:

- *Initial observation*: This is the raw image obtained from an observatory's database/archive, on which the whole process will be applied.
- *Cleaning code*: This is the code which corrects the image faults (in term of the processing used here) of the initial observation, leading to pre-processed images.
- *Pre-processing*: This is the step corresponding to the cleaning code usage.
- *Pre-processed observation*: This is the image after application of the cleaning code. In this image, the Sun is circular, centered in the image and scaled to the standard dimensions (image dimensions 1024x1024 pixel and solar radius 420 pixel), limb darkening is corrected in the case of spectroheliograms, and observational faults are corrected (such as intensity fluctuations due to clouds, strips due to dust on the slit, etc). As long as the working database is in test mode, the pre-processed images will be kept in order to quickly re-process them, if required.
- *Raster Scan (RS)*: Represents contents of a bounding rectangle as runs of pixel values 0, 1 and 2 where 0 corresponding to quiet sun, 1 to penumbra, 2 to umbra. The pixel at the start of the first run corresponds to bottom left corner of the bounding rectangle.
- *Chain code (CC)*: This is a description of a feature boundary or a skeleton. A CC starting point is given (in pixels or arcseconds) with a set of directions indicated by numbers 0 to 7 as defined in Fig. 4. With these CC values the shape of the feature structure can be superimposed on any full solar disk image.
- *Skeleton*: In the case of filaments, a skeleton is a single line without spurs representing the filament. It is expressed as a chain code above.
- *Rectangular image (Bounding rectangle*): is the smallest rectangle in the pre-processed image that contains a full detected feature with its pixel values.
- *Gravity Center*: The centre of gravity is the average location of the weight of features.
- *Skeleton center*: The middle point of a filament skeleton

Pixel origin: The starting position (0,0) for counting pixels up from the bottom left corner by rows and columns as shown in Fig. 4.

1) Original data

Some essential information come from the original data, such as instrument, date/time, possibly wavelength…
The **frequency of observation** used for feature extraction must be known, i.e. features extracted once a day or any other value. In some cases, the user may need to process a higher number of original images, if available, in order to catch some specific phenomenon. But it is not our goal to process all the available images. Some instruments provide high cadence images and process all of them will slow down noticeably both detection and tracking, without any obvious scientific interest.
In case of **data missing** (bad weather on ground, of telemetry problem in space), this must appear explicitly in order that the user won't mix lack of feature with lack of observation!
Moreover, original data can be flagged concerning their '**quality**': bad seeing or interferences can lead to low quality detection of features. The user must be warned of this state.

2) Pre-processing

Pre-processing may include several corrections applied to the original image. Normally it should only enhance the efficiency of detection. But one can imagine that in some case original data may be modified by pre-processing (for instance if a model is to be applied to raw data before detection). In that case, information concerning this pre-processing should be available.
If original images don't have their 'quality' flagged, it should be mentioned at this stage.
Note that there should be a 'No pre-processing' entry where data that don't need pre-processing should refer to. Pre-processing is mostly used for ground-based observations of the sun.

3) Processing

Processing information provides for instance the name of the code used, and the values of the possible parameters. As the features recognition codes aim at being as generic as possible, there could be some parameters that are used for tuning. This information should be available to users.

4) Characterization of features

Each feature has its own set of characteristic figures. Not only the date of appearance and disappearance of the feature is of interest. If one wants to take full advantage of a HFC, all scientifically useful information should appear there. This is why a document should summarize for each feature a list of important figures with their description. It could be interesting to have the opportunity to be able to post this information during a query, to help the user.

5) *Time tracking*

Time tracking consists in being able to follow, observation after observation, a feature, whether it bends, split, merge or support any change of shape or position.
This information is very important for all people studying features as it is there that stands all the information concerning their evolution. As mentioned earlier, time tracking has no meaning for some kind of features.
The time tracking aspect is common to all features that need it: it just consists in linking features detected. So each feature detected will be associated with a unique ID which will be used to follow feature evolution in time.

a) Time interval

The time interval of the observation used for detection should appear (for instance, in the shape of some Date_obs and Date_end fields…).
This information should allow to accelerate queries, knowing immediately whether detection exists at the requested time or not.

b) Gaps

The format of this information is still TBD. But it is needed. For instance, the period of time 2008-2010 lacks most of the time filaments on the solar disk. It is an important information to know that there is no filament. But we must not confuse this with missing observation due to bad weather!
One could probably imagine a table that is automatically filled in giving a complete list of observing time OR a complementary table of NON-observing time. The difficulty is to automate it. It probably means that when detecting features automatically, a set of information should be provided EVEN if no feature is detected.

6) *Local tracking*

Local tracking consists in linking observations that don't come from the same 'kind' of observations. It could be close regions in the heliosphere, or in the solar atmosphere. For instance, active regions exist in the solar photosphere, chromosphere, transition region, low corona and corona. It would be an important added-value to be able to connect those regions between various altitudes.
But this tracking should be very carefully defined as local features don't always extend spatially. Moreover, in some case, a delay could occur between features apparition at various places. Some king of propagation model should be defined to estimate this delay.
We cannot ensure that this work could be done in the frame of HELIO. But we must anticipate that and foresee this information. It could probably be defined the same way as long-scale propagation information.

7) *Propagation*

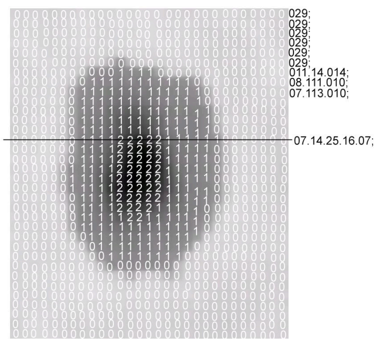
Time tracking, local tracking and propagation information will appear the same way in the database: A unique ID which links various features.
In the case of propagation through heliosphere, this link will be, most of the time, the result of the calculation through a propagation model. Several are already available in the world.

The difficulty results from the fact that no model is fully adopted by the whole community, and no model provide a fully reliable information. So it's likely that users may possibly want to apply by themselves the propagation model they trust for that particular event and features at various dates in various places will be queried for.
This does not mean that information on propagation should not be recorded in the HFC. First, in the future, propagation model will become much more reliable and provide useful information to store. Then, people who work 'by hand' on data from various places of the solar system will validate not a model, but a connection between features scattered in the heliosphere. This information requires to be stored there.

8) Feature representation and position on Sun

The are mainly two way that allow an easy representation of features that can be stored in a DB: Raster scan and Chain code. Those methods are explained hereafter:

- **RASTER SCAN** is used to represents the contents of a rectangular region bounding an extracted sunspot with pixel values equal to 0 corresponding to quiet sun, 1 to penumbra, 2 to umbra (as shown in Fig. 2). The one dimensional array of pixel data formed by concatenating the rows of the rectangular region starting from the bottom left corner is encoded using Run-Length Encoding (RLE).. In the encoded data each run, comprises the value of the pixels forming the run followed by the length of the run, is separated by a decimal point. Examples of RLE data for four sunspots are shown in Fig. 2. The encoding method is called a Raster Scan and the routine used is called Raster Scan 2.



(a)

```
Raster Scan 2;    08.12.01.11.08.19.06.110.05.113.02.114.02.17.22.112.26.110.26.14.01.14.27.14.01.15.25.111.25.15.01.15.22.17.03.13.
 Raster Scan 2;    01.15.02.15.01.116.21.16.21.15.23.13.01.23.13.01.11.21.13.02.14.02.;
  Raster Scan 2;    08.11.017.16.011.19.08.111.05.114.02.116.02.19.21.15.03.14.22.13.23.14.01.14.23.13.23.110.22.13.23.110.22.13.22.
   Raster Scan 2;    07.12.08.14.05.17.03.18.02.11.23.15.01.11.24.17.23.117.01.110.03.13.21.14.04.11.21.15.04.16.06.15.07.11.04.;
```

(b)

Fig. 2. (a) a RASTER SCAN example, (b) a cropped section of the ASCII file shows the RASTER SCAN RLE representation

- **Chain Code:** The boundary of a region is represented by a chain code (CHAIN_CODE) which starts from any boundary pixel and lists the directions required to move from one pixel to the next anti-clockwise round the boundary until the starting pixel is reached (Fig. 3). The direction to the next boundary pixel is represented by an ASCII codes for the 0 to 7. The coordinates of the pixel at the start of a chain code (CC_PIX_X, CC_PIX_Y, CC_ARC_X, CC_ARC_Y) are stored in units of pixels and arcsecs.

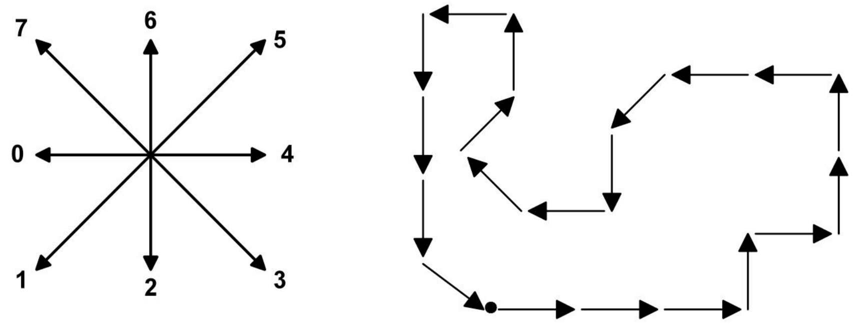
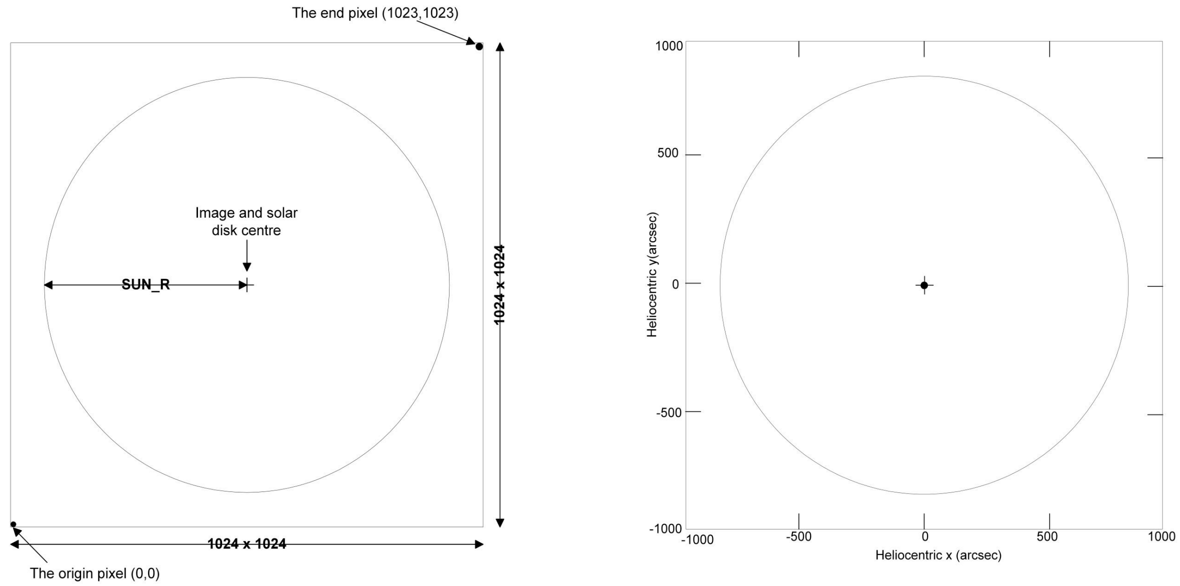


Fig. 3 : Chain Code in 8-connectivity, and its derivative code: 4 4 4 6 4 6 6 0 0 1 2 0 75 6 0 2 2 2 3

Positions of pixels or arcsec on the Sun follow the rules indicated in Fig. 4:



Pixel origin pixel

Arcsec origin pixel

Fig. 4: Image dimensions and origin locations

V. HFC description

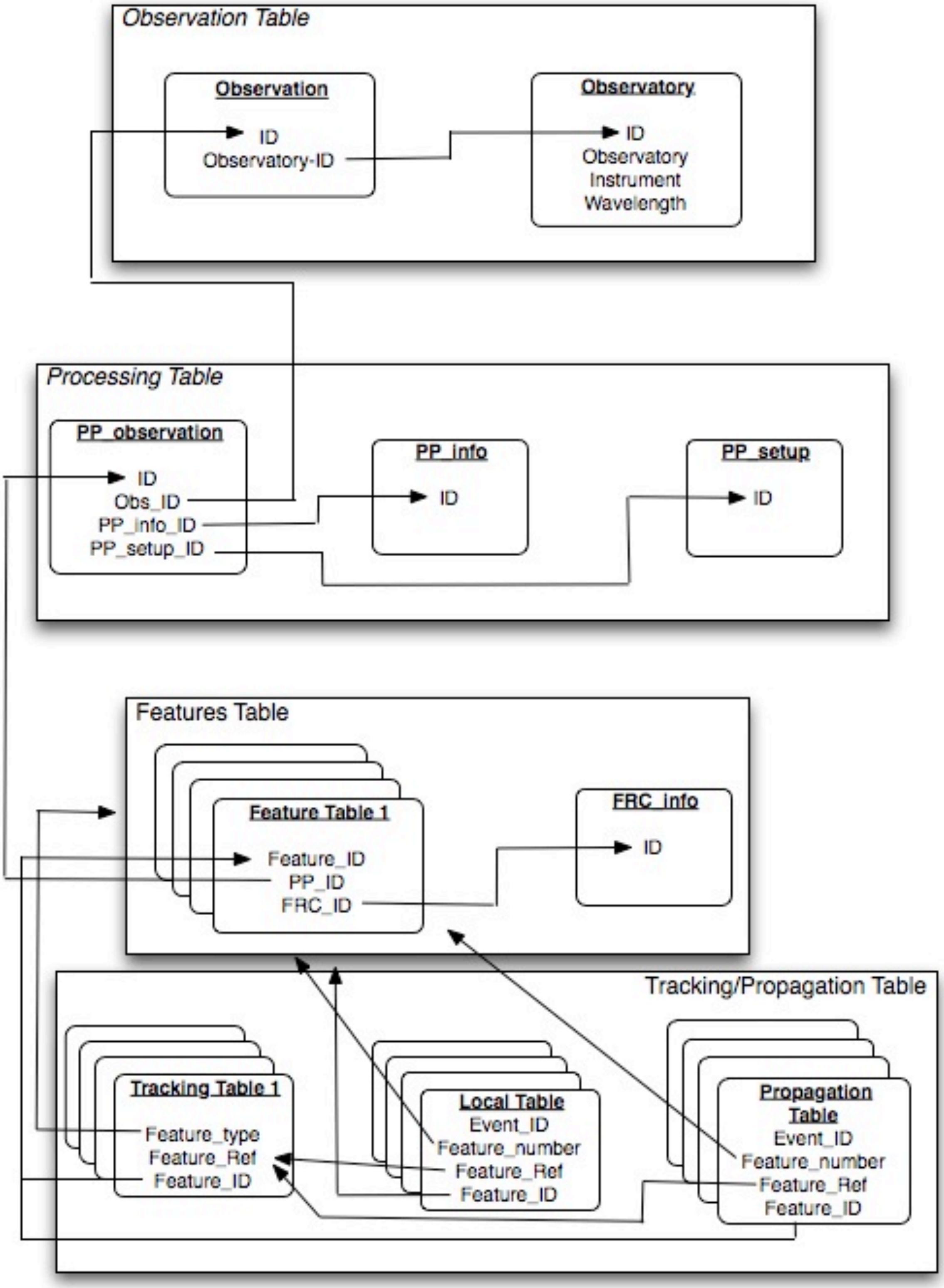


Figure 5: Global organization of Heliospheric Features Catalogue (FRC = Feature Recognition Code)

Figure 5 shows roughly the organization of HFC, in order to have an overview of it. The detail of tables will appear in the next section, where the full description of fields will appear.

1) *Description of tables*

a) Observatory table

Name: OBSERVATORY
Description: This table contains all useful information concerning the observatory/telescope/instrument
Keywords: OBSERVAT, INSTRUME, TELESCOP, UNITS, WAVENAME, WAVEUNIT, WAVEMIN, WAVEMAX, SPECTRAL_NAME, OBS_TYPE

Name	Format MySQL	Description	Notes
ID_OBSERVATORY	Int(11)	Index to identify file and link files together	Primary Key
OBSERVAT	Varchar(255)	Name of the observatory that made the observation	e.g. Meudon
INSTRUME	Varchar(150)	Name of the instrument that made the observation	e.g. Spectroheliograph
TELESCOP	Varchar(150)	Name of the sub-part of the instrument that made the observation	e.g. C2 in case of LASCO
UNITS	Varchar(100)	Units of the original observation intensity	e.g. counts in most cases
WAVEMIN	Float	Minimum value of wavelength	
WAVEMAX	Float	Maximum value of wavelength	
WAVENAME	Varchar(50)	Name of the wavelength of the original observation (if any)	e.g. H alpha, Ca II, Fe IX

WAVEUNIT	Varchar(10)	A, nm, or others	6768 A (Ni I line in MDI)
SPECTRAL_NAME	Varchar(100)	Spectral domain in which observations were made	e.g. optical, SXR, EUV
OBS_TYPE	Varchar(100)	Remote-sensing, in situ	
COMMENT	text		

b) Observation table

Name: OBSERVATIONS
Description: This table contains all the useful information concerning the original image.
Keywords: OBSERVATORY_ID, DATE-OBS, DATE-END, JDINT, JDFRAC, EXP_TIME, C_ROTATION, BSCALE, BZERO, BITPIX, NAXIS1, NAXIS2, R_SUN, CENTER_X, CENTER_Y, CDELT1, CDELT2, QUALITY, FILENAME, COMMENT

Name	Format	Description	Notes
ID_OBSERVATIONS	Int(11)	Primary Index	
OBSERVATORY_ID	Int(11)	Pointing to observatory, wavelength etc	ID in Observatory table
DATE-OBS	Datetime	Date and time of the start of the observation in UTC	e.g. 2003-10-01T17:15:32.123
DATE-END	Datetime	Date and time of the end of the observation in UTC	Same format as DATE-OBS
JDINT	Int(11)	Julian day of the observation, integer part	Internal use
JDFRAC	Double	Julian day of the observation, fraction part	Internal use
EXP_TIME	Float	Exposure time (if available), in seconds and fractions of s	e.g. 27.53
C_ROTATION	Int(7)	Carrington rotation	Must be calculated from DATE_OBS
BSCALE	Double	As extracted from the header	
BZERO	Double	As extracted from the header	
BITPIX	Int(3)	Coding of the original image	
NAXIS1	Int(6)	First dimension of the original image (X)	

NAXIS2	Int(6)	Second dimension of the original image (Y)	
R_SUN	Double	Radius of the Sun, in pixels	(as extracted from the Header, or deduced from pre-processing)
CENTER_X	Double	X coordinate of Sun centre in pixels	(as extracted from the Header, or deduced from pre-processing)
CENTER_Y	Double	Y coordinate of Sun centre in pixels	(as extracted from the Header, or deduced from pre-processing)
CDELT1	Double	Spatial scale of the original observation (X axis) (in arcsec)	
CDELT2	Double	Spatial scale of the original observation (Y axis) (in arcsec)	
QUALITY	Varchar(20)	Quality of the original image (in terms of processing)	Has to be defined for each kind of observation and feature
FILENAME	Varchar(100)	Name of the original file	Path is useful for some MDI files
DATE_OBS_STRING	Varchar(150)		
DATE_END_STRING	Varchar(150)		
COMMENT	Text	As extracted from the Header	
LOC_FILENAME	Varchar(200)		Internal use
ID2	Int(11)		Internal use

Note that this table is a preliminary version and should be enriched by information describing heliospheric data.

c) Pre Processing Information Table

Name: PP_INFO
Description: This table contains the information related to the cleaning code and who run it.
Keywords: INSTITUT, CODE, VERSION, CONTACT

Name	**Format**	**Description**	**Notes**
ID	Int(11)	Primary Index	Internal use, may be used as an unique ID
INSTITUT	Varchar(150)	Institute responsible for running the cleaning code	e.g. Meudon, CDPP, TCD
CODE	Varchar(150)	Name of the cleaning code	
VERSION	Varchar(50)	Version of the cleaning code	
CONTACT	Varchar(150)	Person responsible for running cleaning code	

Remember that one entry should mention ‘no pre-processing’.

d) Processed observation table

Name: PP_OUTPUT

Description: This table contains the values amended or extracted during the pre-processing stage and the information related to the cleaning code setup – describing the algorithm and the adjustable parameters listed in the normalization processing. These parameters, along with version and name from pp_info table, are sufficient to rerun the code to obtain the pre-processed image if needed.

Keywords: PP_INFO_ID, PP_SETUP_ID, OBSERVATION_ID, EFIT, STANDARD, LIMBDARK, BACKGROUND, LINECLEAN, QSUNINT, PERCENT, NAXIS1, NAXIS2, CENTER_X, CENTER_Y, R_SUN, DIVISION, INORM, RUN_DATE, EL_CEN_X, EL_CEN_Y, EL_AXIS1, EL_AXIS2, EL_ANGLE, STDEV, STDEVGEO, ALGERR, CDELT1, CDELT2, BITPIX, QSUN_INT, LOC_FILE, RUN_DATE

Name	Format	Description	Notes
ID_PP_OUTPUT	Int(11)	Primary Index	Internal use, may be used as an unique ID
PP_INFO_ID	Int(6)	Pointing to information about Pre-Processing code, version, institute	
PP_SETUP_ID	Int(6)	Pointing to information about Preprocessing setup	
OBSERVATION_ID	Int(11)	Pointing to Observation	
EFIT	tinyint(1)	Has ellipse fitting been used	Yes(1) or No(0)
STANDARD	tinyint(1)	Has standardisation been used	Yes(1) or No(0)
LIMBDARK	tinyint(1)	Has limb darkening removal	Yes(1) or No(0)
BACKGROUND	tinyint(1)	Has background cleaning been	Yes(1) or No(0)
LINECLEAN	tinyint(1)	Has line cleaning been used	Yes(1) or No(0)
QSUNINT	tinyint(1)	Was Quiet Sun Intensity	Yes(1) or No(0)
PERCENT	Float	Used in ellipse fitting	Default = 0.5
NAXIS1	Int(8)	First dimension of the pre- age (X)	
NAXIS2	Int(8)	Second dimension of the pre- age (Y)	
CENTER_X	Float	X coordinate of Sun centre in	Origin 0,0
CENTER_Y	Float	Y coordinate of Sun centre in	Origin 0,0
R_SUN	Float	Radius of the Sun in pixels	
DIVISION	tinyint(1)	Method used to normalise	Division (1), ssubstraction (0)
INORM	Float	Normalizing parameter for od	Optional
RUN_DATE	datetime	Date where the PP code was	

		run	
LOC_FILE	Varchar(100)	Name of the pre-processed file, including the path from the local organization	
EL_CEN_X	Double	X coordinate of Ellipse centre in pixels	0 if no ellipse fitting used
EL_CEN_Y	Double	Y coordinate of Ellipse centre in pixels	
EL_AXIS1	Double	Ellipse long axis (in pixels)	
EL_AXIS2	Double	Ellipse short axis (in pixels)	
EL_ANGLE	Double	Ellipse angle (°)	
STDEV	Double	Standard deviation(in pixels)	
STDEVGEO	Double	Standard deviation geometric(in pixels)	
ALGERR	Double	Algebraic error (in pixels)	
CDELT1	double	Spatial scale of the pre-processed observation (X axis) (in arcsec)	Calculated based on standardization params
CDELT2	double	Spatial scale of the pre-processed observation (Y axis) (in arcsec)	
BITPIX	Int(3)	Coding of the pre-processed image	Changes if line cleaning is used
QSUN_INT	Float	Quiet Sun value estimated after pre-processing	
RUN_DATE	datetime		
PR_LOCFNAME	Varchar(150)		
ORG_FNAME	Varchar(150)		
ID_ASCII	Int(11)		

This table should enrich with possible information concerning heliospheric data.

e) Feature Recognition Code Information table

Name: FRC_INFO
Description: Similar to PP_INFO, this table contains information about different versions of the feature recognition codes and the people responsible for running it.
Keywords: INSTITUT, CODE, VERSION, FEATURE_NAME, CONTACT

ID	Int(11)	Primary Index	Internal use, may be used as an unique ID
INSTITUT	Varchar(150)	Institute responsible for running the FR code	e.g. Meudon, CDPP, TCD
CODE	Varchar(100)	Name of the FR code	
VERSION	Varchar(50)	Version of the FR code	
FEATURE_NAME	Varchar(100)	Features Detected	e.g. Sunspot
CONTACT	Varchar(150)	Person responsible for running FR code	
REFERENCE	Varchar(150)	Any document or article that	

		describes the FR code	

2) Queries efficiency

A log of the queries done should be stored by the database in order to be able to create possible foreign keys that could help to increase noticeably the speed of queries' answers.

3) Features description tables

Each feature will have its own description. Those characteristics are totally feature dependant. So separate document will describe features characteristic and information stored in the HFC. One feature, one document.
This document should explain in detail to what morphological information each HFC field corresponds.
And this document will be the source of information for users running individually FR codes.
Anyway, the description table should contain at least a link towards the original observation and necessary information to be able to place the feature on a solar map (ie a description, whether it is a chain code or other kind of representation, together with the resolution of the description).

4) Tracking information tables

Time tracking of features on the Sun consists in linking a feature observed at one time to a feature detected in the next observation, and so being able to say: "It is the same feature that moved with the solar rotation.". As some features bend, do not necessarily follow the solar differential rotation, may merge or split, and even disappear during one or two days before to reappear, this is not obvious.
But the representation of this in the HFC is much more simple. A single small table is enough to link each feature detected to a unique feature's ID which corresponds to a unique (see hereafter) features moving all along the solar disk during approx. 14 days. As this information may be unreliable in some case, we propose to add a column giving a level of trust of the tracking. And a last column will contain a reference to a behaviour of the feature, corresponding to a specific behaviours table:
<blank>: Normal behaviour (ie follow the rotation, even if it bends or move at a special rate)
1 = Appearance (of the feature, nothing to do with arriving on the visible side of the Sun!)
2 = Disappearance (of the feature, nothing to do with leaving the visible side of the Sun)
3 = Splitting (1 features becomes 2 or more features)
4 = Merging (2 or more features become 1 or more feature)
5 = Disappearance followed by appearance of the same feature (e.g. "disparition brusque")
6 = Appearance following a disappearance, as above
7 = Abnormal behaviour

The feature tracking table will be as follow:

ID	Int	Primary index	Internal use (may be used as a unique index)
FEAT_TYPE	Int	Number referring to a list of features	The list of features will enrich as soon as new features will appear in the HFC
FEAT_ID	Int	Index of the feature during a rotation	If the level of confidence of tracking is strong enough, this index can be used during more than one rotation.
FEAT_DETEC_ID	Int	Index of the detected feature in the corresponding features table	This refers to the Features description table
LVL_TRUST	Int	Percentage of confidence of the tracking	1 to 100. Can be left blank if no estimation is made.
PHENOM	Int	Number referring to the behaviour of the feature.	Can be blank, or 1 to 7, depends on what happens

Example:
Observation at t0 gives a sunspot number n0a
At t1, the spot splits and at nearly the same location, 3 spots are present, n1a, n1b and n1c.
The corresponding part of the tracking table will look something like:

ID	FEAT_TYPE	FEAT_ID	FEAT_DETEC_ID	LVL_TRUST	PHENOM
id	5 (assuming this corresponds to sunspots!)	f0 (a number that automatically increments)	n0a	100	
id+1	5	f0	n1a	70	3
id+2	5	f0	n1b	70	3
id+3	5	f0	n1c	70	3

And if n1b and n1c merge at t2, leading to n2a spot, the table will continue as:

id+4	5	f0	n2a	100	4

An additional table will take place also at the same level as tracking table: the Group Table.
For instance, sunspots appear mostly in a group. Wolf number, which is a measurement of the level of activity of the Sun, takes into account both the number of sunspots and the number of groups of sunspots. The fields of the group table are:

ID	Int	Primary index	Internal use (may be used as a unique index)
GROUP_ID	Int	Index of the group during a rotation	
FEAT_ID	Int	Index of the feature during a rotation	This refers to the FEAT_ID in the tracking table.

Due to the lack of experience concerning the connection of features in space, it's too early to define by now the organization and content of those tables; but one can imagine that it won't be too far from the Group Table.

VI. HFC prototype

A prototype of HFC is currently running in OBSPARIS. It is running on Darwin 10.3.0 operating system under MySQL 5.0.77.
The goal of this simple prototype is to provide an access for other service activities.

VII. Minimum information needed by HFC

When a data provider wants to include his detections in the HFC, there must be a minimum set of information that must be provided in order HFC to work properly and be able to answer efficiently to queries.
Hereafter, we propose two levels of information that is needed in HFC. The first one concerns mandatory information. If it's not provided, the scientific use of features detection won't be possible. So if this is not available, those data won't be added to the HFC.
The second set of information is a recommended set of elements that would help to allow efficient queries.
Last, but not least, the provider should give a complete description of all the information he deduces from his detection, in order the user of the HFC to be able to take full benefit of the detection.