European Grid of Solar Observations
Project n. IST-2001-32409

Solar Features Catalog Server
Use Cases

Title	**Solar Features Catalog Server Use Cases**
Document number	EGSO-WP5-IR4-1.0
Version	1.0
Date	16 March 2004
Editors	**J. Aboudarham (Paris Obs.) and V.Zharkova (BU)**
Contributors	R.D. Bentley (MSSL), S.Zharkov (BU), N. Fuller (Meudon)
Distribution	Internal

Document Version History

Version	Date	Released by	Details
0.1	2004/02/15	Aboudarham	First full version of the document, using J. Aboudarham and B. Bentley notes
0.2	2004/03/15	Aboudarham	Comments from B. Bentley and N. Fuller included
1.1	2004/03/16	Zharkova	Added comments by Zharkov, Zharkova and Bentley
1.2	2004/03/16	Zharkova	Edited the text and cross-references

Documents referred to:

* Features Parameters Description and Organization, version 1.8 (EGSO-WP5-IR3-1.8)

Table of Contents

I. The objectives

The main scope of this document is to define the best organisation of the SFC server that can answer the most use cases applicable to the features of interest. Obviously, the SFC server is not able to answer all the search and data comparison problems, which are up to a knowledge and creativity of each individual researcher, or user thereafter. However, we try to ensure that all the required information on every detected solar feature is presented in SFC for the users to be able to complete any searches, despite in many cases the external routines have to be developed by the users themsevles or by a third party, i.e. SSW developers.

A general description of the SFC server current structure, database population and possible extension by other features detected by any member of the solar community is given in section 2, The way SFC server can help to perform the searches associated with the use cases is described in section 3. A number of sample use cases are presented based on the current Solar Feature Catalogues (see the Feature Parameters document) populated with sunspots, active regions and filaments, which can be extended by magnetic field maps. This document is aimed to empower the scientists with some samples of the proposed searches from very simple to more complicated ones involving either only the SFC server, or a combination of the SFC, Solar Event Catalogue (SEC) and Unified Observing Catalogue (UOC) Servers.

II. A description of the SFC Server

2.1. General description

The Solar Feature Catalogue (SFC) Server is a specialised provider that delivers access to solar feature metadata. Currently, the metadata are created from the SOHO/MDI whitelight images (4 per day), the Meudon Observatory daily Ca II k3 and H-alpha images using the feature recognition software developed by the University of Bradford and the Observatory Paris-Meudon.

The features are extracted from the images with a time coverage from 1st April '02 until 31st July '02 and populated into ***a relational mysql database*** (RDB) with the searchable entries described in the ***Feature Parameters*** document. The RDB is addressed through Web Service interface on the EGSO website at Bradford University (www.cyber.brad.ac.uk/egso).

The full description of the metadata can be found in the document: " Features_ParametersVx.y.doc", where 'x.y' is the version number of the document. The current version number 1.8 is indicated on the second page of this document.

A few comments:

- Currently, the SFC search results are returned as downloadable ASCII files
- In the future the results can be returned in VOTable format; other formats TBD
- The SFC Servers will be accessible to other VO projects

- The SFC Server is a separate server with a large database of features which will be growing. In order to provide higher operability and datalink capacity, as well a straightforward update and possibility of expansion by other developers, the SFC server is preferable to be separate from any already deployed Servers (SEC or UOC).

During the search construction, in most cases a user will be offered an option to choose standard fields returned by the query, but at the moment the fields are tentative and the standards are yet to be defined (TBD). The use cases show that it is useful to have the lower and upper limit of each field (if available) defined in the SFC search interface so that a user can enter the values within the limits of those available in the database.

In order for users to customise themselves with the SFC server and searches we offer a series of the default (predefined) use cases, where the user can choose with the preset values of query parameters, which can be modifeid by the user at the later stages.

2) Feature tracking

A feature lifetime is a useful information that is very important for many scietific tasks. Many searches may assume that a feature can be followed day after day. This is a difficult problem because the feature moves and/or loses its shape, sometimes very quickly between the two observations. Hence, any attempt to follow a solar feature day after day will be based on these uncertainties that does not mean one can attempt the feature tracking.

In order to look for the birth or disappearance of a feature, one must handle very complex queries involving the whole amount of data available.

A proposed solution to this problem is to run on a regular basis, a routine that allows to follow the same feature from one observation to another one. For this routine one can assume that a gravity centre (to define…) of the feature doesn't move more than by 5° (for instance) from one day to another (for daily images) that should correspond the feature rotation with the Sun.

The results of this search can be used to create a new table in the Solar Feature Catalogue called SAME_FEATURE.

The fields of this table will be defined as follows:

Name	**Type**	**Description**
INDEX	Long	A unique index in the table. Internal use: will link the features tables to this table
MEAN_CAR_LON	Float	Result of calculation of mean position of the Carrington longitude of a feature during its lifetime
MEAN_CAR_LAT	Float	Result of calculation of mean position of the Carrington latitude of a feature during all its lifetime
VAR_LON	Float	Maximum deviation from the mean longitude
VAR_LAT	Float	Maximum deviation from the mean latitude
MIN_DATE_OBS	String	Date of the first observation of a feacture
MAX_DATE_OBS	String	Date of the last observation of a structure

FEAT_IND_MIN	Long	Value of the IND (or INDEX) field of the feature table corresponding to the first observation of the feature (ie a link to the first appearance of a feature in the corresponding table)
FEAT_IND_MAX	Lonf	Value of the IND (or INDEX) field of the feature table corresponding to the last observation of a feature (ie a link to the last appearance of the feature in the corresponding table)

After the search has been performed, a user will add a new field into each feature table, called SAME_FEAT, linking directly to the INDEX field of the SAME_FEATURE table.

There are the two options to compile this table and the SAME_FEAT field (TBD):

- Produce it after a feature has disappeared (after the first appearance of a NULL in the SAME_FEAT field) ;
- This tables can be updated on a regulardaily basis with every new image.

III. Use cases for the SFC Server

The use cases of SFC server can be separated into the following three categories:

• Searches for specific information on a specific feature ***(simple search)***
• Searches for specific information combining the characteristics of several features ***(combined search)***
• Searches for specific information linking the characteristics of one or several features to the other kind of data ***(complex search)***

In the most examples hereafter, unless these are statistical searches only, the query must be completed with a date interval defined in order not to retrieve a huge amount of data.

Some queries may require a procedural language in addidtion to the SQL language in order to allow loops or branching, for instance.

Notation 1: In the following instructions the SQL keywords are written in the UPPER CASE; tables and fields names in *italic*; special functions and variables in **bold face**.

Notation 2: The four tables of feature parameters associated with observations, preprocessing and feature detection are used according to their description in the *Features Parameters*' , v1.8 document.

Notation 3: In order to reduce the length of the queries, a short name was given to each of these tables:

FEATURE TABLE t1 (t1f if FILAMENTS, t1s if SUNSPOTS, t1a if ACTIVE REGIONS)
PROCESSED OBSERVATION TABLE t2
OBSERVATION TABLE t3

SAME_FEATURE TABLE t4

A common form of query will appear in the following cases:

Query	**Comments**
SELECT *fields* FROM t1, t2, t3, t4 WHERE	
(*t1.ind=t4.feat_ind_min*) AND	links *feature table* to the 1st appearance of the feature (in the *same_feature table*)
(*t1.proces_obs_id = t2.index*) AND	links the corresponding proces_obs_id of the *feature table* to the index of the description of the processing in the *processed observation table*
(*t2.observation_id = t3.index*)	links the corresponding observation_id of the *processed observation table* to the index of the description of the observation in the *observation table*

This allows to retrieve all the information related to the feature, or to the preprocessing, or to the observations.

Another way to do so is to create a *VIEW* function, if the mysql database software allows it, that automatically links all the tables together.

3.1. Simple search

3.1.1. List of features on a specific day

Detailed objective
Get a list of all the features with their position on a day choosen by the user.

Prerequisite for the study
None

Initial input and form of the query
Input: Date
Query:
SELECT t1.sc_arc_x, t1.sc.arc.y FROM *t1, t2, t3* WHERE
(*t1.proces_obs_id = t2.index*) AND
(*t2.observation_id = t3.index*) AND
(*t3.date-obs* = **date**)

Output
Structures coordinates.

Additional work for user
None

Notes

3.1.2. Statistics on features sizes

Detailed objective
Build statistics on the size of given features during a long period of time.

Prerequisite for the study
None

Initial input and form of the query
Input kind of feature. Get MIN and MAX of the interesting table field (*feat_area, ske_len_deg*, …). The user can then select intervals in which he wants to make the study and bound his query, leading to something like:
SELECT COUNT(*field*) FROM *t4* WHERE ((*size* LE **max**) AND (*size* GE **min**))

Output
Number of structures in the selected range.

Additional work for user
The user must keep in mind that lower values are limited, due to detection threshold. He is responsible for building the statistics from his queries.

Notes

3.1.3. Biggest features

Detailed objective
Look for structures (of one or several kinds) whose size is not less than x % below the maximum value (or find the x % biggest structures of the database - x must be very small in that case).

Prerequisite for the study
A statistics of the distribution of size of structures must be known by the user in order to make a valid query.

Initial input and form of the query
Input: x value
Query:
SELECT (*fields asked for*) FROM *Tables* WHERE
t1.area >= **x**/100*MAX(*t1.area*)

Output
Fields asked for.

Additional work for user
None

Notes

3.1.4. Preferential region of feature appearance

Detailed objective
Make a statistics on the coordinates (latitude and longitude) of appearing structures of any type (ie structures that were not present on the previous observation). Coordinates must (?) be splitted in a set or ranges.

Prerequisite for the study
Have the total number of different structures M:
M=SELECT MAX(*index*) FROM *t4*

Initial input and form of the query
For i=1,M do begin
For j=0,8 do begin
SELECT COUNT(*t3.sc_car_lon*) FROM *t1, t2, t3, t4* WHERE
(t1.same_feat=t4.index)AND
(*t1f.proces_obs_id = t2.index*) AND
(*t2.observation_id = t3.index*) AND
(*t3.date-obs=t4.min_date_obs*) AND
(*t1.sc_car_lon* >= (**J***10) AND
(*t1.sc_car_lon* < (**J**+1)*10)
endfor
endfor

Output
Count by 10 degrees range

Additional work for user

Notes
If the user needs more specific informations, he can replace
SELECT COUNT(…) FROM …
by
SELECT key1, key2, key3, … FROM …
where key1, key2, key3, … are the names of the parameters he wants to retrieve.
If the user needs only one type of feature (which might be the most frequent case), he'll have to replace t1 by the name of the table he's interested in (t1f, t1s or t1a).

3.1.5. Long duration solar features

Detailed objective
Find the longest duration features of the database, and eventually select a duration and find all the structures lasting that time.

Prerequisite for the study
Statistics on the duration of filaments may be achieved before that.

Initial input and form of the query
Input: Duration
Query:
* For the longest duration feature:
SELECT *feat_ind_min,feat_ind_max* FROM *t4* WHERE (*max_date_obs - min_date_obs*) = MAX(*max_date_obs - min_date_obs*)
* If a duration is entered:
SELECT *feat_ind_min, feat_ind_max* FROM *t4* WHERE (*max_date_obs - min_date_obs*) >= **duration**

Output
Indexes of the first and last feature(s) answering the query. From that it is possible to combine other queries in order to get date, location, …

Additional work for user
None

Notes
The user must keep in mind that lower values are limited, due to detection threshold. So only big enough structures will be taken into account.

3.1.6. Features whose area changed noticeably

Detailed objective
Find structures whose area changed dramatically from one observation to the next one.

Prerequisite for the study
None

Initial input and form of the query
Input: A minimum increasing/decreasing rate (2, 3, or more)
Query:
SELECT *t4.max_dat_obs*, *t4.min_date_obs* FROM *t1, t2, t3, t4* WHERE
    (t1.same_feat=t4.index)AND
    (*t1f.proces_obs_id = t2.index*) AND
    (*t2.observation_id = t3.index*) AND

(SELECT *t1.area* FROM t1 WHERE *t3.date-obs = t4.max_date_obs) >=*
rate * (SELECT *t1.area* FROM *t1* WHERE *t3.date-obs = t4.min_date_obs*)

Output
Fields asked for

Additional work for user
None

Notes
The rate of the area change must be strong enough to have a meaning, taking into account the uncertainties in the automatic determination of features' area.

3.1.7. Active regions emerging between two dates

Detailed objective
Find new appearing active regions.

Prerequisite for the study
This is one of the cases where the new *same_feature* table proposed in II – 2) will be very useful. The query must verify that it's not at the est limb (i.e. check skeleton centre in latitude).

Initial input and form of the query
Input: Bounding dates
Query:
SELECT (*fields asked for*) FROM *t1a, t2, t3, t4*
WHERE (*t1a.ind = t4.feat_ind_min*) AND
(*t1a.proces_obs_id = t2.index*) AND
(*t2.observation_id = t3.index*) AND
SQRT((*t1a.sc_arc_x***2 + (*t1a.sc_arc_y***2)) <= 800 AND
(*t3.date-obs* >= **date_begin**) AND
(*t3.date-obs* <= **date_end**)

Output
Fields asked for.

Additional work for user
None

Notes

3.1.8. Sunspot, active regions or filaments covering an 'important' part of the solar disk

<u>*Detailed objective*</u>
Look for solar images in which the feature coverage (for one or several structures) is bigger than x % of the solar area.

<u>*Prerequisite for the study*</u>
Have an idea of the mean coverage.
A function (**extract_day**) must be defined in order to get only the year ; month, day information from the observing date.

<u>*Initial input and form of the query*</u>
Input: Sun_area value.
Query:
SELECT SUM(*t1a.feat_area*)+SUM(*t1s.feat_area*)+SUM(*t1f.feat_area*), *t3.date-obs*
FROM *t1, t2, t3* WHERE
(*t1.proces_obs_id* = *t2.index*) AND
(*t2.observation_id* = *t3.index*) AND
(**extract_day**(*t3.date-obs*)= SELECT
(**extract_day**(*t3.date-obs*) FROM *t3* WHERE
(SUM(*t1a.feat_area*)+SUM(*t1s.feat_area*)+SUM(*t1f.feat_area*) >=
sun_area*x/100)))

<u>*Output*</u>

<u>*Additional work for user*</u>

<u>*Notes*</u>

The two special instances of this case can be explicitly described as examples of altenate kind of query:

<u>*i) Large active region*</u>
<u>*Detailed objective*</u>
Find active regions whose area is greater than a value given by the user.

<u>*Prerequisite for the study*</u>
The area can be defined by the user ('value'), or it could be a percentage of the available data, e.g. get active regions whose size is between 99 % and 100 % of the maximum area value in order to get only the longest ones. Or it could be define as a percentage of the solar disk (so, a mean value of the solar disk must be known: sol_area)

<u>*Initial input and form of the query*</u>
Input: Minimum area wanted by the user (*feat_area*) or a percent value.
Query:

SELECT (*list of fields asked for*) FROM *active_regions* WHERE (*feat_area* GE **value**)
Or
SELECT (l*ist of fields asked for)* FROM active_regions WHERE (*feat_area* GE **percent***MAX(*feat_area*))
Or
SELECT (*list of fields asked for*) FROM *active_regions* WHERE (*feat_area* GE **sol_area*percent**)

Output
Fields asked for

Additional work for user
None

Notes

ii) Large spots
Detailed objective
Find sunspots whose area is greater than a value given by the user.

Prerequisite for the study
Same as case i) above, but the user must state whether he wants to include umbra size or not in the area calculation.

Initial input and form of the query

Same as case c) above with SUNSPOT table. Inclusion or not of umbra size will modify slightly the query, combining the *feat_area* and *upixsize*cdelt1*cdelt2* (Umbra area in pixels and pixel resolution in X and Y).

Output
Fields asked for

Additional work for user
None

Notes

3.1.9. Sunspots with a few umbras

Detailed objective
Look for the same sunspots, which have a few umbras.

Prerequisite for the study
A number of umbras (**N_UMB**) has be defined prior the search by the year ; month, day information from the observing date.

Initial input and form of the query

SELECT (*list of fields asked for*) FROM (*sunspots)* WHERE *(sunspots.n_umbra GE* ***N_UMB***)

Output

The information on the last observations of SS with multiple umbras.

Additional work for user

Notes

3.1.10. Merging sunspots

Detailed objective

Look for several sunspots that become only one. This can be translated as: Several sunspots disappeared in the same observation (let say more than 3, for instance).

Prerequisite for the study

A function (**extract_day**) must be defined in order to get only the year ; month, day information from the observing date.

Initial input and form of the query

SELECT fields FROM t1s, t2, t3, t4 where
WHERE (*t1s.ind = t4.feat_ind_max*) AND
(*t1s.proces_obs_id = t2.index*) AND
(*t2.observation_id = t3.index*) AND
(SQRT((*t1s.sc_arc_x***2 + (*t1s.sc_arc_y***2)) <= 800) AND
(**extract_day**(*t4.max_date_obs*) = **extract_day**(*t3.date-obs*)) AND
COUNT(**extract_day**(*t3.date-obs*) > 3

Output

Informations on the last observations of several SS.

Additional work for user

Notes

3.1.11. Extended filament, any location

Detailed objective
Find filament with long extension.

Prerequisite for the study
Definition of what is an 'extended' filament (up to the user)

Initial input and form of the query
Input : lowest agreed size (unit to choose and conversion if necessary). Size can be skeleton length (SKE_LEN_DEG) of area size (FEAT_AREA)
Query:

```
SELECT (list of fields asked for by the user) FROM t1f WHERE
    (size GE value)
```

Output
Fields asked for.

Additional work for user
None

Notes

3.1.12. Extended filament running east-west at higher latitude (e.g. polar crown)

Detailed objective
Look for filaments whose orientation is E-W, with the skeleton centre latitude in Carrington coordinates is higher than 60° (or more, up to the user)

Prerequisite for the study
Eventually: user's definition of high latitude and amount of latitude for the E-W orientation.

Initial input and form of the query
Input: Orientation (ORIENTATION), latitude (in units up to the user : SC_CAR_LAT or SC_ARC_Y)
Query:
SELECT (*list of fields asked for by the user*) FROM *t1f* WHERE (((*orientation* LE **value_max**) AND (*orientation* GE **value_min**)) AND (*sc_car_lat* GE **value**))

Output
Fields asked for.

Additional work for user
None

Notes

3.1.13. Filaments 'disparition brusque'

Detailed objective
Search for filaments that disappear on a solar disk (butnot approaching the West limb!) and get date, time and location of the phenomenon. A filament that disappears is assumed to be the filament that was present in the previous observation (s) and then it is no more present.

Prerequisite for the study
None

Initial input and form of the query
Input: Dates of begin and end of search.
Query:
SELECT *t3.date-obs,t1.sc_car_lat,t1.sc_car_lon* FROM *t1f, t2, t3, t4*
WHERE (*t1f.ind = t4.feat_ind_max*) AND
(*t1f.proces_obs_id = t2.index*) AND
(*t2.observation_id = t3.index*) AND
(SQRT((*t1a.sc_arc_x***2 + (*t1a.sc_arc_y***2)) <= 800) AND
(*t3.date-obs* >= **date_begin**) AND
(*t3.date-obs* <= **date_end**)

Output
date-obs, sc_car_lat (or *sc_arc_lat*), *sc_car_lon* (or *sc_arc_lon*) for the last observation of the filament.

Additional work for user

Notes

3.2. Combined search

3.2.1. Filaments associated with active regions

Detailed objective
Search for filaments and active regions having the same spatial position. To simplify the query, it is assumed that the active region is a square.

Prerequisite for the study
None

Initial input and form of the query
Query:
SELECT (*fields asked for*) FROM *t1f, t1a, t2, t3* WHERE

((SELECT *t3.carrot* FROM *t1f, t2, t3*
WHERE (*t1f.proces_obs_id = t2.index*) AND
(*t2.observation_id = t3.index*)) =
(SELECT *t3.carrot* FROM *t1a, t2, t3*
WHERE (*t1a.proces_obs_id = t2.index*) AND
(*t2.observation_id = t3.index*)))
AND
t1f.sc_car_lat BETWEEN (*t1a.sc_car_lat* – SQRT(*t1a.area*) AND
t1a.sc_car_lat + SQRT(*t1a.area*)) AND
t1f.sc_car_lon BETWEEN (*t1a.sc_car_lon* – SQRT(*t1a.area*) AND
t1a.sc_car_lon + SQRT(*t1a.area*))

Output
Fields asked for.

Additional work for user
Check the exact spatial position.

Notes
This method is only available if the features are both in the same Carrington rotation.

3.2.2. Sunspots outside of active regions

Detailed objective
Look for sunspots coordinates which are not inside active regions.

Prerequisite for the study
None

Initial input and form of the query
Query:
SELECT (*fields asked for*) FROM *t1f, t1a, t2, t3* WHERE
((SELECT *t3.carrot* FROM *t1f, t2, t3*
WHERE (*t1f.proces_obs_id = t2.index*) AND
(*t2.observation_id = t3.index*)) =
(SELECT *t3.carrot* FROM *t1a, t2, t3*
WHERE (*t1a.proces_obs_id = t2.index*) AND
(*t2.observation_id = t3.index*)))
AND
t1s.sc_car_lat NOT BETWEEN (*t1a.sc_car_lat* – SQRT(*t1a.area)*
AND *t1a.sc_car_lat* – SQRT(*t1a.area*))
AND
t1s.sc_car_lon NOT BETWEEN (*t1a.sc_car_lon* – SQRT(*t1a.area*)
AND *t1a.sc_car_lon* – SQRT(*t1a.area*))

Output
Fields asked for.

Additional work for user
None

Notes
This method is only available if the features are both in the same Carrington rotation.

3.2.3. Active region important changes related to filament disappearance

Detailed objective
Look for disappearance of the filaments having the same coordinates as active regions. The task is to extract the active regions whose size, brightness or shape (the parameters, possibly, have to be defined with the respect to the reserach task) have changed dramatically.

Prerequisite for the study
None

Initial input and form of the query
This search is a mix of the searches 3.1.13 (Filament 'Disparition brusque') and 3.1.6 (Feature whose area changed noticeably).

Output
Fields asked for.

Additional work for user
None

Notes

3.3. Complex search

The most cases mentioned below involve the magnetic field magnitude and/or polarity in a solar structure. This could be done manually after downloading the ASCII file from the SFC with the feature parameters and restoring it as an IDL structure and combining it with the relevant magnetic data for the same observing epoch using the SSW packages. The a user can build the queries inside the IDL structure corresponding to what is required to evaluate evaluates concerning the magnetic field. Another solution could be to find how to describe the magnetic structure of the Sun locally and automatically fill in a database table with those informations.

Either ways a new kind of queries can be defined as follows.

3.3.1. Active regions with enhanced X-ray background

Detailed objective

Look for active regions having the same position as X ray flares (taking into account the difference in altitude, if necessary).

Prerequisite for the study
List of the coordinates of flaring events with enhanced X-ray background.

Initial input and form of the query
Input: Latitude (**lat**) and longitude (**lon**) (transformed in Carrington coordinates if necessary) and date. (**date**)
Query:
SELECT (*fields asked for)* FROM *t1a, t2, t3* WHERE
(*t3.date-obs* BETWEEN (**date** – 1 AND **date** + 1))
AND (**lon** BETWEEN (*t1a.sc_car_lon* – SQRT(*t1a.feat_area*)
AND *t1a.sc_car_lon* + SQRT(*t1a.feat_area*)))
AND (**lat** BETWEEN (*t1a.sc_car_lat* – SQRT(*t1a.feat_area*)
AND *t1a.sc_car_lat* + SQRT(*t1a.feat_area*)))
AND *t1a.proces_obs_id = t2.index*
AND *t2.observation_id = t3.index*

Output
Fields asked for.

Additional work for user
Verify the exact superimposition.

Notes
This method is only available if the structures are both in the same Carrington rotation.

3.3.2. Spots with strong magnetic fields

Detailed objective
Find sunspots, which locations coincide with strong magnetic field (to be defined, which magnitudes are considered strong).

Prerequisite for the study
List of strong magnetic fields coordinates.

Initial input and form of the query
Coordinates of the strong magnetic fields
Once magnetic field coordinates known, the query works exactly the same way as case 2)a), except that filaments are replaced by magnetic field and active regions by sunspots.

Output
Fields asked for.

Additional work for user
None

Notes
This search depends on the way magnetic field will be defined. But an interesting way could be to be able to evaluate automatically the magnetic field.

3.3.3. Active regions with leading spot with the polarity opposite to normal

Detailed objective
Scan spots in active region and check their polarity. Then compare it to the one having the highest longitude (or latitude?).

Prerequisite for the study
Get the polarity of the spots and normal polarity of leading spots (POLAR).

Initial input and form of the query
One normal polarity of leading spots known, this query consists only in looking for spots which polarity doesn't correspond to this one.

Output
Fields asked for.

Additional work for user
None

Notes
The details of this search depends on the way polarity will be defined. As in the previous case, an automatic detection of the magnetic field could be very helpful for this kind of query.

3.3.4. Active regions with one dominant polarity

Detailed objective
TBD

Prerequisite for the study
TBD

Initial input and form of the query
TBD

Output
TBD

Additional work for user
TBD

Notes
This search depends on the way polarity will be defined.

3.3.5. Active regions with closely located area of strong opposite polarity (strong field gradients)

Detailed objective
TBD

Prerequisite for the study
TBD

Initial input and form of the query
TBD

Output
TBD

Additional work for user
TBD

Notes
This search depends on the way polarity will be defined.

3.3.6. Filaments corresponding to flares locations between two dates

Detailed objective
Check for filaments corresponding to flares.

Prerequisite for the study
List of interesting flares (can come for SEC).

Initial input and form of the query
Input: Latitude (**lat**) and longitude (**lon**) (transformed in Carrington coordinates if necessary) and date (**date**) of the flare.
Query:
SELECT (*fields asked for*) FROM *t1f, t2, t3* WHERE
(*t3.date-obs* BETWEEN (**date** – 1 AND **date** + 1))
AND (**lon** BETWEEN (*t1f.sc_car_lon* – SQRT(*t1f.feat_area*)
AND *t1f.sc_car_lon* + SQRT(*t1f.feat_area*)))
AND (**lat** BETWEEN (*t1f.sc_car_lat* – SQRT(*t1f.feat_area*)
AND *t1f.sc_car_lat* + SQRT(*t1f.feat_area*)))
AND *t1f.proces_obs_id = t2.index*
AND *t2.observation_id = t3.index*

Output
Fields asked for.

Additional work for user
Check validity of results.

Notes

3.3.7. List of emerging AR corresponding to flares

Detailed objective
Check for new appearing active regions (less than a few days) corresponding to flares.

Prerequisite for the study
List of flares and list of emerging active regions [see 1)f)], coming from SEC server.

Initial input and form of the query
This search is a mix of 1)e) (Emerging active regions between two date)and 3)f) (Filaments corresponding to flares locations), changing filament table to active region table.

Output
Fields asked for.

Additional work for user
None

Notes