SPoCA Module manual.

Written by Benjamin Mampaey (benjamin.mampaey@oma.be) on 5th March 2010.

Updated by Benjamin Mampaey on 26th March 2010.

Updated by Benjamin Mampaey on 2nd April 2010.

Updated by Benjamin Mampaey on 25th June 2010.

Updated by Benjamin Mampaey on 22nd July 2010.

Updated by Benjamin Mampaey on 30th March 2011

Updated by Benjamin Mampaey on 8th November 2011

Module description.

The SPoCA module is composed of 3 programs :

- spoca.pro which is an IDL procedure. Basically it is just an IDL wrapper to call the 2 other programs, transform pixel Cartesian coordinates into HPC coordinates, and do file cleanup.
- classification.x is a program written in C++. It detects Active Regions (AR) or Coronal Holes (CH) on images (171 and 195 A) and output the map of those regions.
- tracking.x is a program written in C++. It uses several maps produced by classification.x to track a region (AR or CH) through time.

Content of the tarball.

- rob_spoca_0.7.xml and rob_spoca.xml : xml files describing the module for the pipeline.
- hek.xsd : schema for the xml files above
- spoca_ar.pro or spoca_ch.pro: The IDL module (see Module description)
- bin, cgt, classes, dsl, programs : folders containing the C++ code (see Module description)
- classification.mk, tracking.mk : make files for the module subprograms (see Module compilation)

Module compilation.

spoca.pro :
Requisites :

- Solar Soft WCS routines

Compilation :

This procedure can be compiled into a Java-IDL bridge object using the instruction stated in SDO Event Detection System (EDS) API.

Like suggested in the aforementioned document, we have not compiled the IDL procedure so it can be done on location.

bin/classification.x :
Requisites :

- gcc version 4.4.3
- GNU Make 3.81
- cfitsio version 3.26

Compilation :

```
make -f classification.mk
```

bin/tracking.x :

Requisites :

- gcc version 4.4.1
- GNU Make 3.81
- cfitsio version 3.25

Compilation :

```
make -f tracking.mk
```

Module Arguments.

Arguments general to all modules (as specified in the document SDO EDS API)

- events: out, required, type string array, see document SDO EDS API
- write_file: in, optional, type boolean, see document SDO EDS API
- error: out, required, type string array, see document SDO EDS API
- imageRejected: out, required, type boolean, see document SDO EDS API
- status: in/out, required, type struct, see document SDO EDS API
- runMode: in, required, type string, see document SDO EDS API
- inputStatusFilename: in, optional, type string, see document SDO EDS API
- outputStatusFilename: in, required, type string, see document SDO EDS API
- numActiveEvents: out, required, type integer, see document SDO EDS API

Arguments specific to the SPoCA modules

- outputDirectory: in, required, type string, folder where spoca can store temporary files (The modules manage the cleanup of old files)
- verbose: in, optional, type integer, verbose level of the module (0 is off)
- saveFiles: in, optional, type integer, save files deleted from outputDirectory to saveDirectory below (0: none, 1: last one, 2: all)
- saveDirectory: in, optional, type string, folder to use to save images corresponding to events for debugging
- writeEventsFrequency: in, required, type integer, number of seconds between events write to the HEK
- cCodeLocation: in, optional, type string, directory of the c executables
- instrument: in, optional, type string, instrument that took the images (AIA,EIT,EUVI)
- spocaArgs: in, required, type string array, options for running the classification of spoca
- chaincodeArgs: in, required, type string array, options for specifying the numbers and precision of the chain codes
- trackingArgs: in, required, type string array, options for running the tracking of spoca
- trackingOverlap: in, optional, type integer, number of images to overlap between succesive tracking run

Remarks :

1. outputDirectory is a directory in which the module is going to store intermediate files. So it should have write access permission on that directory. No other program or person should be allowed to write in that directory as it could disturb the module. The module takes care of the cleaning of the files in that directory.
2. All sun images files passed to the module must have similar radius (1% max difference), otherwise the module will produce inaccurate results.
3. writeEventsFrequency is the cadence in seconds to which we report events to the HEK.
4. The path to the executables of classification.x, tracking.x and must be specified as the argument cCodeLocation

Arguments specific to the SPoCA AR module

- image171, image195: in, required, type string, images filename of wavelength 171 Å and 195 Å (193 Å for AIA)

Arguments specific to the SPoCA CH module

- image195: in, required, type string, image filename of wavelength 195 Å (193 Å for AIA)
- minLifeTime: in, optional, type integer, Minimal time for a CH to be alife to be exported
- minDeathTime: in, optional, type integer, Minimal time for a CH to be dead to be devinitively supressed

Remarks :

1. The module spoca_ch.pro will only report CH events if those have a minimum lifetime of 3 days.
2. Setting write_file for spoca_ch.pro will write all events to file, and not just the ones that are more than 3 days olds.
3. For CH events, the relations to previous events might include relations to events not reported to the Hek. This is because at the time of the creation of the events, it is not possible to know which previous events will be reported or not.

Version 0.2

Bug fixes

- Fixed Event_StartTime and Event_EndTime
- Fixed time in label of AR for the generation of the IVORN number
- Changed call to SPAWN in spoca.pro, for the reporting of errors
- Added last_color_assigned in status for tracking

Improvements

- Added code for Event_C1Error and Event_C2Error
- Added program get_regions_HEK.x
- Added call to get_regions_HEK.x program into spoca.pro
- Test for existence and mode of spoca_bin, tracking_bin and getregionstats_bin in spoca.pro

- Implemented the 'Clear Events' runmode, to cleanup Armaps

Version 0.3

Bug fixes

- Fixed Event_StartTime and Event_EndTime, again (because of bug in the ctime library) . Now they are outputted in ccsds format
- Fixed the start time of the very first event, so it is the time of the very first image submitted
- Fixed bug in the computation of Event_C1error and Event_C2error, and moved it to get_regions_HEK

Improvements

- Added code for areas and intensity statistics of AR
- Added a test for the quality of the pictures

Version 0.4

Bug fixes

- Fixed bugs in GetRegionStats (Cis)

Improvements

- Change the way we do tracking. Now it is done only when we write events to the Hek.
- Added support for AIA images
- AIA images pixel values are divided by the exposure time. I would like confirmation that it is not done in level 1.5.
- Added support for Rice compressed images.
- Added required keywords in event like QUALITY, FLAT, FLAT_VER, …
- Changed xml files rob_spoca.xml and rob_spoca_0.4.xml for AIA values

Version 0.5

Bug fixes

- Corrected bug in tracking.x
- Added a centers quotient factor to avoid AR center to decrease too much in case of low activity

Improvements

- Cleanup of the outputDirectory at construct
- Added the tracking info into the events. This implies that I need to store past_events in the status structure. Because it is an array that can change size, I am overwriting the status structure. Is it a problem?
- Renamed GetRegionStats_HEK to get_regions_HEK, SpoCA_HEK to classification, Tracking_HEK to tracking.
- Added parameters to spoca.pro : getregionArgsPreprocessing, getregionArgsRadiusRatio, instrument
- Added a save_folder variable at the beginning of spoca.pro to allow to save the maps of AR corresponding to the events written.

Version 0.6

Bug fixes

- Correction on the minimal size of the AR reported
- Correction in the computation of the Area_AtDiskCenterUncert
- Correction of the EVENT_NPIXELS reported
- Suppression of the optional keywords when not available (i.e. have a value of NaN)
- Corrected parsing of event relations
- Corrected the web address of the SPoCA information page

Improvements

- Suppression of the events relations duplicates
- Update of the required events keywords
- Better approximation of the center of the AR
- Added code of Ryan to check the quality of the files (as a separate procedure checkQuality)
- Added the event.optional.FRM_VersionNumber into the events
- Added the VersionNumber into the FRM_SPECIFICID for easier module reset

Version 0.7

Bug fixes

- Several bugs in the computation of statistics for the AR/CH have been corrected.
- Corrected a bug in the cgt library that prevented the tracking.x program from compiling with gcc version 4.5.

Improvements

- The program has been simplified in that it does not need to call get_regions_HEK.x anymore. The above documentation has been updated to reflect that change
- The CH detection version has been implemented, and the above documentation has been updated to reflect that.
- The spoca module does not need to parse output from the C++ programs anymore. All information and statistics about regions, statistics and chain codes are being written has tables into the map fits files.
- Idem for the tracking information and relations between events.
- We allow to run the tracking on as little as 1 map.
- Added a check on aectype in the checkQuality procedure (suggested by Ryan) (! Has been commented out though as it was incorrectly rejecting most of our test data)
- Reporting chain code for CH and AR events.
- Added 3 parameters to spoca module for debugging purposes: verbose, saveFiles and saveDirectory. See module parameters documentation above.
- Reporting if the event is clipped spatially (EVENT_CLIPPEDSPATIAL). It is set to true when the event is near the limb. This has the consequence that intensities statistics and area statistics are truncated.
- Reporting the event probability for CH (EVENT_PROBABILITY). The value of the probability is computed using the following formula:
  1 - ((event median intensities - min CH intensity) / (max CH intensity - min CH intensity))
  Where :
  - min CH intensity, has a value of 7, and is the 5th percentile of the minimum intensities of all CH events for the year 2010-2011
  - max CH intensity, has a value of 54, and is the 95th percentile of the maximum intensities of all CH events for the year 2010-2011
- The method for detecting AR has been modified to improve the stability of the detection. The new method consist in computing the memberships of the QS class by doing an HPCM2 classification on 4 classes, on images 171A & 193A. Then to extract the AR class by stipulating that AR are brighter than QS and have a QS membership lower than 0.0001.
- To improve the stability of the detection, the class centers are recorded for the last 10 classifications, and their median is used for the final classification.
- The aggregation of pixels into regions (AR or CH) has been improved to take into account the projection of the solar sphere in the image.
- AR_INTENSXXX statistics have been modified to INTENSXXX, and INTENSMEDIAN has been added.
- The spoca module parameters have been simplified and now take arrays of strings for some parameters.
- The spoca module for AR can be run every hour instead of every 15 minutes.

Version 1.0

Improvements

- The minimal chaincode points has been set to 3.