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.

Module description.

The SPoCA module is composed of 4 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) on a pair of images (171 and 195 A) and output the map of AR.
- tracking.x is a program written in C++. It uses several maps of AR produced by classification.x to track an AR through time.
- get_regions_HEK.x is a program written in C++. It uses one map of AR tracked by tracking.x and the corresponding (i.e. same time of observation) image in wavelength 171A, and produces different statistics about each AR.

Content of the tarball.

- rob_spoca_0.4.xml and rob_spoca.xml : xml files describing the module for the pipeline.
- hek.xsd : schema for the xml files above
- spoca.pro : The IDL module (see Module description)
- bin, cgt, classes, dsr, objects, programs, results : folders containing the C++ code (see Module description)
- classification.mk, tracking.mk, get_regions_HEK.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 4.4.1
- GNU Make 3.81
- cfitsio version 3.25

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
```

bin/get_regions_HEK.x :

Requisites :

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

Compilation :

```
make -f get_regions_HEK.mk
```

Remark :

2. The path to the executables of classification.x, tracking.x and get_regions_HEK.x must be specified as the argument cCodeLocation of the procedure spoca.pro.
3. There is a debug flag at the beginning of the module. Changing it from 0 to 1 allow to print a lot of debug information.

Module Arguments.

Sun images

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

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 module

- outputDirectory: in, required, type string, folder where spoca can store temporary files (The

modules manage the cleanup of old files)

- writeEventsFrequency: in, required, type integer, number of seconds between events written 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)
- spocaArgsPreprocessing: in, optional, type string, type of image preprocessing for spoca
- spocaArgsNumberclasses: in, optional, type string, number of classes for spoca
- spocaArgsPrecision: in, optional, type string, precision for spoca
- spocaArgsBinsize: in, optional, type string, bin size for spoca
- trackingArgsDeltat: in, optional, type string, maximal time difference between 2 images for tracking
- trackingNumberImages: in, optional, type integer, minimum number of images for the tracking
- trackingOverlap: in, optional, type integer, proportion of the number of images to overlap between tracking successive run
- getregionArgsPreprocessing:in, optional, type string, type of image preprocessing for getregions
- getregionArgsRadiusRatio:in, optional, type float, radius ratio for getregions

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 frequency in seconds to which we write events to the HEK. No events will be written before we do the first tracking.

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