SPoCA Module manual.

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

Updated by Benjamin Mampaey on 26th March 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.
- SPoCA_HEK.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_HEK.x is a program written in C++. It uses several maps of AR produced by SPoCA_HEK.x to track an AR through time.
- GetRegionStats_HEK is a program written in C++. It uses one map of AR tracked by Tracking_HEK.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_0.2.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)
- SPoCA_HEK.mk, Tracking_HEK.mk, GetRegionStats_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/SPoCA_HEK .x :

Requisites :

- gcc version 4.4.1
- GNU Make 3.81

Compilation :

```
make -f SPoCA_HEK.mk
```

bin/Tracking_HEK .x :

Requisites :

- gcc version 4.4.1
- GNU Make 3.81

Compilation :

```
make -f Tracking_HEK.mk
```

bin/GetRegionStats_HEK .x :

Requisites :

- gcc version 4.4.1
- GNU Make 3.81

Compilation :

```
make -f GetRegionStats_HEK.mk
```

Remark :

2. The path to the executables of SPoCA_HEK .x, Tracking_HEK .x and GetRegionStats_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 and 195

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
- 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, number of images to track at the same time
- trackingOverlap: in, optional, type integer, proportion of the number of images to overlap between tracking successive 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 the same radius, otherwise the module will produce inaccurate results.
3. writeEventsFrequency is the frequency in seconds to which we write events to the HEK. Because we only write after doing a tracking, we will write less often to the Hek than writeEventsFrequency, if the delta time between the first image and the last image passed to the tracking program is superior to writeEventsFrequency.

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 GetRegionStats_HEK.x
- Added call to GetRegionStats_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 GetRegionStats_HEK

Improvements

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