omcosflag

August 16, 2005

Abstract

Applies the OM tracking history to a bad pixel map

1 Instruments/Modes

Instrument	Mode
OM	FAST
OM	IMAGING

2 Use

pipeline processing	yes
interactive analysis	yes

3 Description

The shift and add process means that the counts observed in each pixel of an OSW image may have been accumulated in several different detector pixels. This task takes the OM tracking history and applies it to the map of bad pixels defined in the CCF. The result is an increase in the area of the OSW image contaminated by bad pixels. The bad pixel map produced by this task will be used by the source detection algorithms and the OMMODMAP task. The binning of the bad pixel map, and the offset of the science window, are read from the headers of the OSW image, which is an input of the task.

4 Parameters

This section documents the parameters recognized by this task (if any).

Parameter	Mand	Type	Default	Constraints
samplefactor	no	integer	1	1 - 10

Spatial oversampling factor (oversampling of bad pixel map)

timefactor	no	integer	1	1 - 10

Temporal sampling factor (subdivision of tracking frame spacing)

set	yes	string	none	

OM OSW Image filename

thxset	yes	string	none	

OM Tracking History Data Auxiliary filename

5 Errors

This section documents warnings and errors generated by this task (if any). Note that warnings and errors can also be generated in the SAS infrastructure libraries, in which case they would not be documented here. Refer to the index of all errors and warnings available in the HTML version of the SAS documentation.

badFileMode *(fatal)*
Invalid datamode <dataMode> in set <set>

allocError *(fatal)*
Can't allocate memory for internal array <name>.

deAllocError *(warning)*
Failed to release memory for <name>
corrective action:

6 Input Files

1. OM OSW Image file
2. OM Tracking History Data Auxilliary file

7 Output Files

1. Same OM OSW Image file as input image (QUALITY extension added)

8 Algorithm

```
subroutine omcosflag

   read in task parameters

   open OM OSW Image file
```

```
   determine file type from datamode in primary header
   read in science window keywords from the primary header
   extract window data from MODES binary table extension
   open OM Tracking History Data Auxiliary file
   determine file type from datamode in primary header
   extract tracking history data from OMTHX1 binary table extension
   check for poor tracking solution and warn
   extract bad pixel map from CAL
   calculate maximum drift in x and y
   extract localised bad pixel map allowing for maximum drifts
   over sample localised bad pixel map
   loop over tracking frames to construct oversampled quality array
      loop over discretised tracking steps
         logical OR oversampled quality array with shifted oversampled bad pixel map
      end loop
   end loop
   add QUALITY extension to OM OSW Image file
   resize and bin quality array with appropriate binning factors
   write quality array to QUALITY extension of OM OSW Image file
end subroutine omcosflag
```

9 Comments

- none

10 Future developments

- The QUALITY array is currently composed of 8-bit integers, but should probably be boolean.

References