rgsbadpix

August 16, 2005

Abstract

This task assembles the two node-specific tables of bad pixels for one CCD from an RGS exposure. The telemetered pixel data is analyzed in order to augment the calibration data with hot pixels that have not previously been identified. The output `BADPIXn` tables are added, by default, to the input dataset.

1 Instruments/Modes

Instrument	Mode
RGS	Spectroscopy
RGS	High Time Resolution

2 Use

pipeline processing	yes
interactive analysis	yes

3 Description

rgsbadpix combines the calibration bad pixel data from the CCF with any hot pixels found by analyzing the raw telemetry for one CCD from an RGS exposure. The result is stored in the `BADPIXn` extension tables of the input dataset, or in a separate dataset if requested (`withbadpixset`=yes). When the same bad pixel is identified by both the CCF and the telemetry analysis, the output bad pixel table mentions only the CCF version. CCF bad pixels that lie outside the telemetry window of the exposure are not included. The columns of the `BADPIXn` tables describe these properties: location, type and source. Location is specified in chip-oriented coordinates, and describes a column segment of similar bad pixels in terms of its minimum y-axis coordinate and upward extent. The type codes are described in the CAL documentation. The source codes are as follows:

1. A bad pixel obtained from the CCF, and listed there as having been uplinked to the on-board data preprocessor (DPP). Such pixels should never appear in the telemetry, and should therefore never be identified by the telemetry analysis. There is no option to exclude these from the output bad pixel tables; to do so would corrupt the exposure map.
2. A bad pixel obtained from the CCF, but not listed as having been uplinked to the DPP. These are regarded as advisory, and may be excluded from the output bad pixel tables (`withadvisory`=no). One would hope that any of these listed as hot would also be identified by the telemetry analysis.
3. A bad pixel identified by analysis of the telemetry. The current algorithm only attempts to find isolated hot pixels and columns. This process is optional and may be disabled (`withfoundhot`=no).

Four parameters control the hot pixel finding algorithm, which is an implementation of John Peterson's memo, RGS-COL-CAL-00015. First the pixels are analyzed individually for excessive activity and then the same is done for whole columns. `pixnoiselimit` and `colnoiselimit` respectively are the minimum uncalibrated energies for considering single-pixel events in the first and second phases. `pixsharpness` and `colsharpness` respectively control the degree of excessive activity required to flag a pixel or column as hot. Refer to section 8 for the precise meaning of the sharpness parameters. Frames marked with `IN_BAD_FRAME` in the `FLAG` column of the `EXPOSURE` table are considered bad and their pixels are discarded.

In High Time Resolution (HTR) mode the entire cross-dispersion dimension is collapsed into one row; hence it is not possible to consider individual bad pixels. If hot pixel finding is enabled the hot column finder works just the same way as in Spectroscopy mode but the initial search for individual hot pixels is skipped. Throughout this document a † marks items that do not apply to HTR mode data and a ‡ marks items that apply only to HTR mode data.

4 Parameters

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

Parameter	Mand	Type	Default	Constraints
ccdset	yes	dataset	events.ds	

Intermediate event list.

withbadpixset	no	boolean	no	

Enable output of the BADPIXn tables to a specified dataset, rather than to the input dataset.

badpixset	no	dataset	badpix.ds	

Name of the dataset to be created as output, if enabled.

withadvisory	no	boolean	yes	

Enable inclusion of advisory (non-uplinked) bad pixels from the CCF.

withfoundhot	no	boolean	yes	

Enable telemetry analysis and inclusion of found hot pixels.

pixnoiselimit	no	integer	0	0 - 4095

Lowest uncalibrated single-pixel energy to be considered while searching for hot pixels (Spectroscopy mode only).

colnoiselimit	no	integer	250	0 - 4095

Lowest uncalibrated single-pixel energy to be considered while searching for hot columns.

pixsharpness	no	real	5	0-

Sharpness criterion for selecting hot pixels (Spectroscopy mode only).

colsharpness	no	real	8	0-

Sharpness criterion for selecting hot columns.

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.

MissingFrames *(warning)*
The PIXELS table references frame numbers that were not found in the EXPOSURE table. This could be due to corrupted data or improper sorting of the tables.
corrective action: All pixels associated with the missing frames are discarded.

InvalidCoordinates *(warning)*
A pixel has coordinates outside of the telemetered window. This could be due to a corruption or misinterpretation of the shape code.
corrective action: The pixel is discarded.

BadPixDuplicates *(warning)*
The CCF has multiple bad pixel entries at or overlapping the same coordinates. This indicates a poorly constructed CCF.
corrective action: The duplications are discarded, but the order of precedence is not defined.

BadPixIncorrect *(warning)*
Bad pixels listed in the CCF as uplinked or dead are present in the telemetry. This is a clear indication that the CCF is incorrect.
corrective action: The dubious CCF entry is accepted as correct.

BadPixDiscrepancy *(warning)*
There may be cause for concern when a non-uplinked bad pixel listed in the CCF as hot is not also discovered by the hot pixel finding algorithm.
corrective action: The hot pixel finder is assumed to have missed these pixels.

BadPixNonsense *(warning)*
A CCF bad pixel listed for HTR mode data refers to less than a whole column.
corrective action: The fatuous CCF entry is extended to fill the whole column, though most likely it is completely bogus.

6 Input Files

- ccdset

  The following tables are required from the input dataset.

  - EXPOSURE

    Required columns:

FRAME	int32	frame identifier
FLAG	int32	advisory flags

  - PIXELS

    Required attributes:

WINDOWX0 WINDOWY0	telemetry window: origin
WINDOWDX WINDOWDY	telemetry window: dimensions
CCDID	ccd identifier
CCDOCB	on-chip binning factor
RAWY‡	nominal readout row in HTR mode

    Required columns:

FRAME	int32	frame identifier
RAWX RAWY†	int16	coordinates
ENERGY	int16	uncalibrated energy
CCDNODE	int8	node identifier
SHAPE†	int8	shape code
GRADE†	int8	number of pixels

  The following dataset attributes are also required: TELESCOP, INSTRUME, OBS_ID and EXP_ID.

7 Output Files

- ccdset

  Unless output to a separate dataset is enabled, a pair of tables named BADPIX0 and BADPIX1 are added to the input dataset, replacing any previously existing versions.

  – Attributes

CCDNODE	readout node identifier
WINDOWX0 WINDOWY0	telemetry window origin, chip coordinates
WINDOWDX WINDOWDY	telemetry window dimensions
ABUTTED	telemetry windows abut, logical

  – Columns

CHIPX CHIPY†	int16	chip coordinates of bottom pixel
YEXTENT†	int16	upward extent in pixels
TYPE	int16	bad pixel type identifier
BADFLAG	int16	bad pixel source identifier

  Attributes CCDID and CCDOCB are copied from the PIXELS table of the input dataset, and attribute INSTRUME is copied from the input dataset header. The attribute ABUTTED, described above, is set true if the last column of the telemetry window for node 0 is adjacent to the first column of the telemetry window for node 1. When, as normal, the nodes do abut, any bad pixel entries in the adjacent last and first columns will be written as duplicates in both nodes. This allows the NEXT_TO_BADPIX event attribute to be set correctly in the opposing node.

- badpixset

  When enabled this dataset is created, clobbering any previous version, and the tables named BADPIX0 and BADPIX1 are added according to the previous specification. The following attributes are also copied to the output dataset header from the input dataset header: TELESCOP, INSTRUME, OBS_ID and EXP_ID.

8 Algorithm

The following inequality must be satisfied for an individual pixel to be found hot. Here N is the number of counts (qualifying telemetered events located there) in the pixel itself and M is the number of counts in whichever of the four directly adjacent active pixels has the fewest counts. F is the number of contributing frames (that is, not counting bad frames). A pixel is not considered active if it is listed as an uplinked bad pixel in the CCF. And unless inclusion of the CCF advisory bad pixels has been disabled (withadvisory=no), advisory dead pixels are also not considered active. In the exceptional case of a pixel with no directly adjacent active pixels, it is not judged to be hot regardless of its count rate.

$$N > M + (\texttt{pixsharpness})\sqrt{N\left(1-\frac{N}{F}\right)+M\left(1-\frac{M}{F}\right)}$$

Similarly, this next inequality must be satisfied for a whole column to be found hot. However, here N is the average number of counts per pixel associated with G_N, the number of good pixels in the column. M and G_M are the corresponding average number of counts and number of good pixels for whichever of the two neighboring columns has the lowest average number of counts. To be considered good at this stage a pixel must be active in the above sense and not have been found to be a hot pixel. Columns with no good pixels are ignored, both as potential hot columns and as neighboring columns. If both of a column's neighbors have no good pixels, it is not judged to be hot regardless of its count rate.

$$N > M + (\texttt{colsharpness})\sqrt{\frac{N}{G_N}\left(1-\frac{N}{F}\right)+\frac{M}{G_M}\left(1-\frac{M}{F}\right)}$$

In the following schematic description of the program an asterix is used to mark items that do not apply to HTR mode data.

```
Initialize the bad pixel map:
   Large enough for all valid chip coordinates plus a one pixel margin on
      all sides; this margin and any median between the nodes (in case of
      windowed telemetry) is initialized with Invalid; the rest with zero

Initialize the calibration bad pixel list:
   Ignore advisory bad pixels unless withadvisory == true
   Convert bad pixel node coordinates to chip coordinates
   Clip bad segments that extend above or below the window

if withfoundhot:
   Create two histogram maps in the same dimensions as the bad pixel map,
      one for the pixel* analysis and the other for the column analysis

   Convert telemetered node coordinates to chip coordinates

   for each telemetered event in a good frame:
      Ignore and warn if any of the comprising pixels have invalid
         coordinates; otherwise test against the pixnoiselimit* and
         colnoiselimit and increment the corresponding histogram
         maps accordingly at the location of each comprising pixel

   for each calibration bad pixel:
      if uplinked || DEAD, mark in the bad pixel map

   Search the pixel analysis map* for hot pixels, creating a list of their
      coordinates and recording as statistics for each column the number
      of good pixels and associated average event count from the column
      analysis map

   if not HTR:
      Mark the found hot pixels in the bad pixel map

   Search the column statistics for hot columns, marking them in the
      bad pixel map

for each calibration bad pixel:
   if advisory && HOT && withfoundhot, warn if not already marked as
      found in the bad pixel map
   mark and reclassify as appropriate in the bad pixel map

Construct the merged list of bad pixel segments such that all locations
   marked bad in the bad pixel map are accounted for and all pixels in
   each segment have the same classification

Write the BADPIXn tables
```

9 Comments

- The hot pixel finding algorithm has some difficulty dealing with on-board split-event reconstruction (SER). Multi-pixel telemetered events (items in the `PIXELS` table with `SHAPE` greater than zero, `GRADE` greater than one) cannot be compared directly to the thresholds set by the noise limit parameters. The charge-splitting ratios obtained from the CAL are therefore used to scale the thresholds appropriately for each grade. If a telemetered event passes this aggregate cut, each of its comprising pixels must be taken to pass as well, despite that the reality might be otherwise. This over-counting appears to be unavoidable, but hopefully over the course of many frames the true location of the hot pixels will become apparent.
- In HTR mode each telemetered "pixel" is assumed to represent activity in only one detector element. This is a good assumption in practice even though the on-board binning collects 74 rows together before each readout. Consequently it is possible that some of the telemetry will be mishandled by the hot column finding algorithm. In particular the algorithm is prone to under-estimate the average number of events in the more active columns. The impact of this potential error on the efficacy of the algorithm has not been investigated.

References