rgsevents

August 16, 2005

Abstract

This RGS task constructs an `EVENTS` table from the raw telemetered pixel data for a single CCD exposure. This table is added to the input dataset along with two `REJPIXn` tables (one for each node), which augment the existing `BADPIXn` tables. The input dataset is assumed to have been processed by **rgsbadpix**.

1 Instruments/Modes

Instrument	Mode
RGS	Spectroscopy
RGS	High Time Resolution

2 Use

pipeline processing	yes
interactive analysis	yes

3 Description

rgsevents adds the EVENTS and REJPIXn tables to the intermediate event list for a single RGS CCD exposure, which has been processed by **rgsbadpix**. These tables are sorted in increasing order of frame number. The EVENTS table is also cross-referenced with the existing PIXELS table so that the correspondence between the reconstructed events and the raw telemetered pixels is not lost. Throughout this document a † marks items that do not apply to High Time Resolution (HTR) mode data and a ‡ marks items that apply only to HTR mode data.

The event reconstruction algorithm defines an event as a maximal set of adjacent pixels. That is, any two pixels which share an edge are contained within the same event. The user has the option to disable event reconstruction, which only means that each qualifying telemetered pixel is treated as a separate event regardless of its proximity to other pixels. To cross-reference the reconstructed events with their comprising telemetered pixels a column named EVENT is added to the PIXELS table, in which is recorded for each pixel the EVENTS table row number of the event in which it is included, or null if it is not associated with an event. From the raw pixel data the following properties are computed for each event and stored in the EVENTS table. Also stored for each event are three properties copied from the EXPOSURE table: frame number, flags and timestamp.

- energy

  Total energy, both calibrated and raw.

- grade

  Total number of pixels, or area of the event.

- shape

  Chip-oriented DPP shape code for any event confined within a two-by-two pixel region; otherwise null. The shapes are shown below, with the numeral indicating both the code number for each pattern and the relative position of the reference pixel. Note that in the PIXELS table the telemetered coordinates mark the location of the reference pixel, but in the EVENTS table the centroid coordinates may lie anywhere within the bounding rectangle of the event.

  ```
          **    *          *     *    **     **
    0     1*    2    3*    4*    5*   6       7
  ```

- coordinates

  Coordinates of the event centroid in various units:

  - Chip-oriented pixel coordinates, in integer format.
  - RGS angular coordinates, without aspect correction.
  - RGS detector coordinates in a system chosen by the user. Optional.

- flags

  These flags indicate conditions which cast serious doubt upon the quality of the event as an x-ray detection:

  - ON_BADPIX

    At least one pixel included in the event is found in the bad pixel tables.

  - NEXT_TO_BADPIX

    The pixel under the event centroid shares an edge with a pixel from the bad pixel tables.

– ON_WINDOW_BORDER
  The pixel under the event centroid lies in the first or last row or column of the telemetered window.
– ON_NODE_INTERFACE
  The pixel under the event centroid shares an edge with the opposing node.
– BELOW_ACCEPTANCE
  The calibrated energy of every pixel included in the event lies below the acceptance threshold.
– BAD_SHAPE
  The event does not lie entirely within a two-by-two pixel region.

These flags provide information about the composition of the event:

– ON_NODE0
  At least one pixel included in the event lies in node zero.
– ON_NODE1
  At least one pixel included in the event lies in node one.
– ONBOARD_SER†
  At least one pixel included in the event represents an on-board reconstruction.

Included also are any flags set by **rgsframes** in the EXPOSURE table. For each flag present in the output table a Sparse message declares the total number of events so flagged.

For each node the CCF provides PI thresholds for acceptance and rejection—not to be confused with similarly named thresholds used on-board by the Data Preprocessor (DPP). Since these thresholds are directly applicable only to single-pixel energies, the average highest single-pixel energy is computed for pixels with on-board split-event reconstruction (SER) using an empirical charge-splitting ratio provided for each grade (no SER correction is applied in HTR mode). Any pixel with PI less than the rejection threshold is ignored, just as if it had not been telemetered. Further, any event that does not include at least one pixel with PI at or above the acceptance threshold is flagged with BELOW_ACCEPTANCE. The floating-point pixel PI values are summed over the pixels comprising each event and "pixelized" by rounding to the nearest integer. The pixelization of the event PI values, as well as the minimum and maximum (pixelized) values present in the data, are described by standard FITS attributes on the PI column of the EVENTS table. It may be unwise to assume that the minimum value is nonnegative.

A null shape code implies nothing about the shape of the event. In particular, it does not imply that the event is not confined within a two-by-two pixel region. A previous version of the DPP does not provide enough information about pixels of grade two and three for the ground software to determine the shape. Use the BAD_SHAPE event flag to test whether an event is two-by-two confined. The event centroid is computed as a PI weighted sum over randomized sub-pixel coordinates (to avoid aliasing effects). For pixels with on-board SER the combined energy is treated as if spread uniformly over the telemetered shape. If the shape is not available for a pixel, its energy is placed at a random offset from its upper right corner. HTR mode y-coordinates are randomly distributed over the full 384 row height of the chip (disabling randomization produces the nominal readout row used in the energy corrections). Randomization may be disabled via the CAL state. This program does not attempt to identify cosmic-rays by pulse-height analysis. If an event falls within a two-by-two pixel region it is considered to be an x-ray; otherwise it is something with a bad shape—possibly a cosmic-ray. Note that if any of the pixels in a two-by-two region have no shape information, BAD_SHAPE is not set as long as all pixels are under grade three. This should not be an issue with the current DPP, and in any case should pertain only to events straddling the node boundary.

While determining the flags for each event, special note is taken of pixels with certain flags, when these flags have not been set identically in the bad pixel map. These are additional rejectable pixels and may be needed when constructing the exposure map. The flags in question are: `BAD_SHAPE`, `ON_BADPIX`, `NEXT_TO_BADPIX`, `ON_WINDOW_BORDER` and `ON_NODE_INTERFACE`. The chip coordinates of each such pixel are added to the `REJPIXn` table for its node, along with its frame number and any of these flags that were set on its associated event. The `REJPIXn` tables augment the `BADPIXn` tables for the purpose of mapping exposure.

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 processed by **rgsbadpix**.

withdetcoord	no	boolean	no	

Enable output of detector coordinates as columns in the `EVENTS` table.

detcoord	no	string	cam	chip—cam—sac

Coordinate system type for the detector coordinates, if enabled. The options refer to the standard XMM coordinate systems, CHIPCOORD (with sub-pixel accuracy), CAMCOORD1 and SACCOORD.

reconstruct	no	boolean	yes	

Enable event reconstruction.

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.

BadParameter *(error)*
An unrecognized value was specified for `detcoord`.

BoundsMismatch *(error)*
The telemetry window bounds do not match between the `PIXELS` and `BADPIXn` tables.

InvalidCoordinates *(warning)*
A pixel with coordinates outside of the telemetered window.
corrective action: The invalid pixel is discarded.

EventOverlap *(warning)*

A frame with more than one pixel at the same location.
corrective action: The extra pixels are discarded.

MissingFrames *(warning)*

Pixels attributed to a non-existent frame.
corrective action: Such pixels are discarded.

6 Input Files

- `ccdset`

  The following tables are required from the input spectroscopy dataset, which is expected to have been processed by **rgsbadpix**.

  – `EXPOSURE`

  Required columns:

`FRAME`	int32	frame identifier
`FLAG`	int32	advisory flags
`TIME`	real64	timestamp

  – `PIXELS`

  Required attributes:

`DATATYPE`	instrument mode
`CCDID`	ccd identifier
`CCDOCB`	on-chip binning factor
`TELAPSE`	total integration time
`WINDOWX0 WINDOWY0`	telemetry window: origin (node coordinates)
`WINDOWDX WINDOWDY`	telemetry window: dimensions
`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
`PI`	real32	calibrated energy

  – `BADPIX0` and `BADPIX1`

  Required attributes:

`CCDNODE`	readout node identifier
`WINDOWX0 WINDOWY0`	telemetry window: origin (chip coordinates)
`WINDOWDX WINDOWDY`	telemetry window: dimensions
`ABUTTED`	abuts on opposing node

  Required columns:

`CHIPX CHIPY`†	int16	coordinates
`YEXTENT`†	int16	pixel count
`BADFLAG`	int16	provenance

  The dataset is also required to possess the `INSTRUME` attribute.

7 Output Files

- ccdset

  These tables from the input spectroscopy dataset are modified as follows.

  – EVENTS

    The table of events is created or replaced. Any flag-selection attributes of the EXPOSURE table are reproduced here.

    * General attributes

XMMEA_3	ON_NODE0 selection expression
XMMEA_4	ON_NODE1 selection expression
XMMEA_12†	ONBOARD_SER selection expression
XMMEA_19	BAD_SHAPE selection expression
XMMEA_20	ON_NODE_INTERFACE selection expression
XMMEA_21	ON_BADPIX selection expression
XMMEA_24	NEXT_TO_BADPIX selection expression
XMMEA_25	ON_WINDOW_BORDER selection expression
XMMEA_27	BELOW_ACCEPTANCE selection expression
DATATYPE CCDID CCDOCB TELAPSE	copied from PIXELS table

    * PI column attributes

TCTYP	channel axis type (no applicable standard)
TCUNI	channel axis units (PI)
TCRPX	reference channel number (0)
TCRVL	center of reference channel (0.0)
TCDLT	uniform channel width (1.0)
TDMIN TDMAX	minimum and maximum data values present

    * Columns

FRAME	int32	frame identifier
CHIPX CHIPY	int16	chip-oriented pixel coordinates
DETX DETY DETZ	real32	optional RGS detector coordinates
BETA XDSP	real32	RGS angular coordinates
PI	int32	total calibrated energy
SHAPE	int8	chip-oriented shape code
GRADE	int8	total number of pixels
FLAG	int32	advisory flags
PHA	int32	total telemetered energy
TIME	real64	frame timestamp

  – REJPIXn

    The tables of rejectable pixels are created or replaced.

    * Attributes

CCDNODE	readout node identifier
INSTRUME	copied from the dataset header
CCDID CCDOCB	copied from PIXELS table

    * Columns

FRAME	int32	frame identifier
FLAG	int32	rejectability flags
CHIPX CHIPY†	int16	chip-oriented pixel coordinates

  – PIXELS

    The event cross-referencing column is created or replaced.

EVENT	int32	row number of comprising event

8 Algorithm

In this section an asterix marks items that do not apply to HTR mode data.

```
Initialize the bad pixel map:

   mark ON_WINDOW_BORDER along the window edges:
      top* and bottom*, outer sides, and inner sides if not abutting
   mark ON_NODE_INTERFACE along window inner sides if abutting
   for each bad pixel region:
      mark ON_BADPIX along the region itself
      mark NEXT_TO_BADPIX on all directly adjacent pixels:
         above* and below*, and along both sides

Perform event reconstruction:

   for each frame:
      initialize map of pixels not below the rejection threshold
      for each pixel still on the map:
         initialize new event
         move pixel from map to stack
         until stack is empty, taking the top pixel:
            add pixel to event
            move all adjacent pixels from map to stack

Or, if event reconstruction is disabled:

   for each frame:
      for each pixel not below the rejection threshold:
         construct a new event

Update the dataset:

   set the event cross-reference to null for all pixels
   for each event:
      cross-reference the event with each of its pixels
      compute and store event properties
      add additional rejectable pixels to the REJPIX tables
```

9 Comments

10 Future developments

References