omfastshift

August 16, 2005

Abstract

Corrects FAST mode event list coordinates for spacecraft drift.

1 Instruments/Modes

Instrument	Mode
OM	FAST

2 Use

pipeline processing	yes
interactive analysis	yes

3 Description

In Fast mode, because of telemetry constraints, the shift-and-add performed on the spacecraft in IMAGE mode is applied to FAST mode data on the ground. This is achieved using two SAS tasks. This task applies the 'shift' and EVSELECT applies the 'add'.

In IMAGE mode the DPU shift-and-add procedure is applied after every tracking frame integration, $t_{track} \sim$ 10-20 seconds. In FAST mode, a FAST mode time slice (FMTS) is defined with duration $t_{FMTS} < t_{track}$. Each event is tagged with a frame number corresponding to the FMTS in which the event was detected. It is therefore possible to correct each event with a fine shift calculated from a linear interpolation of the tracking frame shifts

OMFASTSHIFT calculates drift-corrected event positions based on a linear interpolation over a tracking frame integration, of drifts stored in the ODF OM tracking history auxiliary data file. These are then written as additional columns to the input ODF OM fast mode event list file.

The drift-corrected OM Fast mode event list file is passed to OMFASTFLAT which uses the OM in-orbit flatfield and the RAWX/RAWY values of the event list to flatfield the event list on an event by event basis. The output flatfielded event list is then passed to EVSELECT to produce a Fast Mode flatfielded image which can be treated by subsequent SAS tasks in the same way as an Image Mode flatfielded image.

4 Parameters

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

Parameter	Mand	Type	Default	Constraints
set	yes	string	none	

OM Fast Mode Event List Intermediate File (output of OMPREP)

thxset	yes	string	none	

OM Tracking History Data Auxiliary Intermediate File (output of OMPREP)

nphset	yes	string	none	

OM ODF Non-Periodic Housekeeping file

interpolate	no	boolean	T	

Linearly interpolate s/c drift over FAST mode time slices?

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.

Wrong datamodes for input files *(fatal)*

6 Input Files

1. OM Fast Mode Event List Intermediate File (output of OMPREP)
2. OM Tracking History Data Auxiliary Intermediate File (output of OMPREP)
3. OM ODF Non-Periodic Housekeeping File

7 Output Files

1. Drift-corrected OM Fast Mode Event list file (input to OMFASTFLAT)

8 Algorithm

```
subroutine omfastshift

Read parameters

Get a handle on the ODF OM Tracking History Data Auxiliary File.
Confirm datamode from DATAMODE keyword in primary header.

Get a handle on the tracking history data binary table extension.
Calculate drifts.
If BPE binning implemented divide drifts by two
Release handles on column and table.

Read in Intermediate Event List file.
Confirm datamode from DATAMODE keyword in primary header.

Get a handle on the event list binary table extension.
Get the number of rows in the event list binary table extension.

Read modes extension.

Get start and end time of exposure from the non-periodic housekeeping file.
Calculate tracking frame time stamps in OBT for middle of tracking frame.

Do 'shift and add' process.

   Calculate On board Time for each event taking into acount wrap around for int16 values.

   if (ibits(ftFineP(i-1), 15, 1) == 1) then

      obtEvent(i) = REAL(ftCoarseP(i-1)) + REAL( ibits(ftFineP(i-1), 0, 15) + 2**15 ) / 65536.0

   else

      obtEvent(i) = REAL(ftCoarseP(i-1)) + REAL(ftFineP(i-1)) / 65536.0

   end if

   Locate tracking frame in which event occured.
   If linear interpolation of drifts is required ...

      Find time of event since previous tracking frame.
      Interpolate drift to this time

   If not use tracking frame drift

end do


Flag events outside Science window

Convert Onboard Timestamps (OBT) to event time tags (elapsed seconds since mission
reference time 1998-01-01T00:00:00).
```

```
Write time attributes to input file.
Add new columns.
Write data to columns.
Release data.

end subroutine  omfastshift
```

9 Comments

- FAST mode data is not binned by the DPU. The drifts from the ODF OM Tracking History Auxiliary File need to be corrected for BPE only.
- All events are assumed to occur in the centre of their pixel. The output integer pixel coordinate value is the integer truncation of the real coordinate value.
- For FAST mode, in addition to a tracking frame time, a FAST mode time slice (FMTS) is defined with duration $\sim$ SAMPTIME*10ms, where SAMPTIME is integer (i.e. integer multiples of a CCD frame)
- Each event is tagged with a time stamp (FTCOARSE+FTFINE) corresponding to the specific FMTS integration within each tracking frame integration, in which each event occurred. This allows the possibility to apply a 'fine shift' to each event corresponding to the spacecraft drift, after each FMTS integration.
- omfastshift produces by default a corrected event list in which all events within a FMTS are shifted back to the reference frame using drifts calculated from a linear interpolation of the tracking frame drifts in the OM Tracking History auxiliary data file.
- A corrected event list in which all events within a tracking frame are shifted back to the reference frame using the tracking frame drifts can be called for by the setting flag interp_flag = F on the command line
- It is worth noting that there exists cases in which a FMTS crosses a tracking frame boundary. Although events may belong to two tracking frames, all events within the FMTS are given the same event timestamp and are treated equivalently.
- The drift-corrected OM FAST mode event list contains 4 additional columns: the corrected X and Y pixel coordinates and OBT time for each event and a flag denoting whether the shift-corrected event falls within the defined science window. This flag is necessary since the s/c drift causes the science window to sample a region of the sky larger than the science window. When these events are drift-corrected they may fall outside the GO defined science window area. A value of WIN_FLAG=0 is given for all events outside the science window, and WIN_FLAG=1 for all events in the science window.
- TLMIN and TLMAX are required by EVSELECT.

10 Future developments

- It might be possible to improve on the linear interpolation scheme when the spacecraft drift is better understood.
- Appropriate keywords associated with timing information will be written when XMM data conventions on this issue have been finalised.

References