Super-Boresight Pointing History File (SBPHF) Generation and Usage for S13.0

F. Masci, 05/03/2005

1. Purpose

For each raw Boresight Pointing History File (BPHF) associated solely with _IRAC_ AORs, we plan to create a refined version (SBPHF) using the results of astrometric pointing refinement from the post-BCD thread. We have focussed on IRAC due to the availability of good astrometric match statistics with 2MASS in one or more of it's bands. With a refined BPHF, one can then attempt to refine all IRAC bands self-consistently since all wavelength-dependent arrays observe the sky simultaneously.

2. Plan

For S13.0, the plan is to generate the SBPHFs offline (and manually by the pipeline operator) on a pipeline drone, archive and register them in the SODB:telemetryFiles table (i.e., ingest them) following DQA, and optionally use them in subsequent reprocessing of IRAC campaigns with the pointing transfer thread using a configurable namelist.

3. SBPHF Script Functions

The SBPHF generation script ($SIRTF_SCRIPTS/GenSBPHFsForCampaign.pl) performs the following functions.

* Reads in the following required inputs from the command-line: a campaign string, instrument-FOV (Euler angle offsets) table file and an output directory for the SBPHF products and QA files. Executing the script with no command-line arguements gives a tutorial.

* Query for all raw (12-hour based) BPHFs in the specified campaign using the "getBestTelemetryFilesForCampaign" stored function.

* Query for all AOR-based refined pointing table files in the specified campaign using the "getRefinedPointingTblFilesForCampaign" stored function.

* Sorts all AOR-based refined pointing table files for each input raw (12-hour based) BPHF into unique filelists.

* Executes the "gnsbph" module on each input raw BPHF and corresponding refined pointing table list for the entire campaign. Results (QA files and SBPHFs) are written to the user-specified output directory.

* The output SBPHFs are named in the following manner: if the input raw BPHF is named "BPHF.0STARTSCLK.PV.pntg" for example, where PV = Parent Version, then the output SBPHF will be named: "SBPHF.0758376000.PV1.pntg"

if no prior SBPHFs from the same parent (i.e., SBPHF.0758376000.PV*) exist in the SSC archive. If a version already exists in the archive (say, SBPHF.0758376000.PV1.pntg), then the output will be named "SBPHF.0758376000.PV2.pntg". In other words, the latest refined version ("2" here), is concatenated with the parent version.

* Exits with a report on the actual number of SBPHFs made compared to the total number of raw-BPHF in the campaign.

* Conditionally archives and registers SBPHFs in the database if the "-a" flag was set on the command-line. If this flag was set, no SBPHF generation is performed, instead, all pre-existing SBPHFs in the present working (or execution) directory will be archived. Files are archived by calling the "fastIngestSTS" module. See below for an example test run.

4. Input Namelist/Calibration files

Three configuration/calibration files are currently required by the script:

* The namelist: "gnsbph.nl".
  This has been checked into the operations TFS under the IRAC-1_INT POV. If not already there, please see AI-1488.

* The _calibration_ file: "instrument_FOV.tbl".
  This is the latest version that is delivered together with all other IRAC calibrations.

* For the ingesting (archival and SODB registration step), ensure that the appropriate namelist for "fastIngestSTS" (i.e., ingest.nl) is present under "/scr/$SOS_REL/cdf/ingest/" on the drone where you plan to perform your ingesting (presumably the same drone where you initially created the SBPHs). This namelist has been checked into CVS under: "$SOS_VERSION/downlink/cdf/ingest/".

5. Example Test (or Operational Procedure) - 4 Basic Steps: A to D

Step A: Generation of SBPHFs

Here's a suggested step-by-step guide on how to generate a set of SBPHFs for a given campaign. The example below uses campaign IRAC003700.

*** Please perform this when you have (PAO) processed a substantial number of AORs in a given campaign through to, and including the post-BCD pointing refinement step (IRAC PlscriptId 27). By substantial, I mean as many AORs as you have available and processed for a given campaign.

1. source /sos/config/$SOS_REL/downlink/operator.csh
   where $SOS_REL has been pre-defined in the I&T or OPS environment.

2. Create a directory, e.g., "superBS/" under a /scr/.. drone partition or area which has >~300MB. This is the maximum you will ever need for a campaign execution run.

3. cd superBS/ and execute the following:

```
GenSBPHFsForCampaign.pl -n gnsbph.nl -c IRAC003700 -f instrument_FOV.tbl \
-o OutSBPHFs_IRAC003700 | tee LOG_IRAC003700.txt
```

A log of the output will be captured in the file "LOG_IRAC003700.txt".

4. This run will create a subdirectory "OutSBPHFs_IRAC003700/" in the execution directory containing all outputs. e.g.,
```
SBPHF.0758376000.10.RefndTbl.list
SBPHF.0758376000.103.AOR0008701440.ps
SBPHF.0758376000.103.pntg
SBPHF.0759067200.03.RefndTbl.list
SBPHF.0759067200.033.AOR0008697600.ps
SBPHF.0759067200.033.pntg
.
.
```
etc.. for as many BPHFs and AORs as are present in the campaign.

5. There will be a QA ".ps" file for every AOR found in the campaign with existing refined pointing table (from post-BCD processing), and a ".pntg" file for each raw BPHF. If actual QA ascii tables are requested for offline analysis (e.g., DQA), then the PO should set "rmqain = .true.," in the gnsbph.nl namelist.

Step B: QA and SBPHF Validation

Before proceeding with the actual archival, the SBPHFs generated from the above run will need to go through some sort of QA or validation, either by the DQA team, or a developer with prior knowledge. The primary QA files of interest are the ".ps" files. There will be one of these files per AOR. Each file will have three panels showing a time-series of the difference between residuals in [refined - raw(actual)] pointing values in the boresight and instrument science frame for RA, Dec and Position Angle (PA). The smaller these differences, the more accurate the refined boresight solution. One should pay careful attention to the RA and Dec plots (top two panels) and raise a red flag if either of these exceed ~0.6 arcsec. The PA residual differences (bottom panel) can vary in a wild manner, especially if one is near a pole, so please ignore the PA plot for now. Also, for your information the symbol shapes in each of the three panels represent the following:

ch1 = "+" sign
ch2 = "x" sign
ch3 = "o" sign
ch4 = "triangle" sign

It is envisaged that once these files have been quality assured, the responsible DQA liason will make recommendations on whether to go ahead and use the corresponding SBPHFs or just a subset of them. A list of good SBPHFs will be given to the pipeline operator to ingest (see Step C).

Step C: Archival and SODB Registration of SBPHFs

Following QA of a specific set of SBPHFs for a campaign, they will be ready for archiving (in the SSC raw-archive) and registration in the SODB:telemetryFiles table. This can be performed using the following steps:

1. cd to the directory containing the SBPHFs you desire to archive.

2. Ensure your ingest directory structure is set-up on the relevant drone, i.e., ensure the following directories exist with the appropriate namelist: ingest.nl (where a copy is under: $SOS_VERSION/downlink/cdf/ingest/).

   "$SOS_LOCAL/ingest";
   "$SOS_LOCAL/ingest/sirtfx";
   "$SOS_LOCAL/ingest/sirtfx/rawin";
   "$SOS_LOCAL/ingest/sirtfx/rawwork";
   "$SOS_LOCAL/cdf/ingest/ingest.nl";

3. Execute the following:
   GenSBPHFsForCampaign.pl -a

   This performs the following:

   (i). archives (copies) each SBPHF into the relevant "/sos/archive/raw/timeperiod/YYYY.MM/refinedPointing/" directory (e.g., on the SSCON), where "YYYY.MM" is the year.month string derived from the start-sclk in the SBPHF name. Note the new FileType: "refinedPointing".

   (ii). registers each SBPHF into the SODB:telemetryFiles table using the full "vbesting" machinery, i.e., the latest version ingested will automatically get "vbest=1". See the "SBPHF Script Functions" section above for the SBPHF versioning format.

   (iii).writes ingest-logging information to log files under "$SOS_LOGS/ingest/timeperiod/YYYY.MM/rptg/"

Step D: Usage in SSC Pointing Transfer (Reconstruction) Software

Once the desired SBPHFs have been archived and registered, they can be used in subsequent campaign reprocessing by setting a flag in the pointing transfer namelist: "pointing_transfer.nl". This namelist has been checked into the operations TFS under each IRAC-[ch]_INT POV, where ch = 1, 2, 3 and 4. If not already there, please see AI-1573. For book-keeping purposes, this namelist has also been checked into CVS under: "$SOS_VERSION/downlink/cdf/ptgXfer/".

The reason for making SBPHF usage namelist configurable is so the pipeline operator has flexibility in using different "pointing_transfer.nl" versions for PAO and campaign reprocessing by including it under different respective PLID subdirectories under $SOS_LOCAL/cdf on each drone.

The relevant parameter in this namelist is "UseRefinedBPHF".

If usage of SBPHFs in pointing transfer is desired, set the following: "UseRefinedBPHF = 1,". If it is not desired, set: "UseRefinedBPHF = 0,". For the case where "UseRefinedBPHF = 1" was set but no SBPHF version could be found, the software will write a warning to the output log and default to using the (regular) raw BPHF version.

Step E: Final Verification of Super-Boresight Refined Products (Optional)

This step may be performed by a DQA or IST member as a sanity check
to ensure that the refined BPHFs (i.e., SBPHFs) did indeed lead to
an improvement in the final reconstructed pointing after reprocessing.

This can be verified by checking that the refined pointing residuals
(differences between actual and refined pointing) for each BCD in the
"refinedPointing.tbl" table (archived as an ensemble product for each AOR)
from the post-BCD pointing-refinement thread are appropriately small,
i.e., better than ~0.05 arcsec. The relevant fields in this table are:
"RARESID" and "DECRESID".

Note that these fields are also represented as keywords in each BCD
FITS header and, for S13, will also be loaded into a new database table:
"QA_ptg_refine" and replicated to the archive database: "archive@sodb1".
So for example, given a reqkey (say AOR=8697600), in dbaccess:

```
set role table_read;
select b.dceId,b.raresid,b.decresid,b.paresid
       from dces a, qa_ptg_refine b
       where a.reqkey=8697600
       and a.dceId=b.dceId;

    dceid        raresid       decresid        paresid

  7995200 0.017448007191 -0.09696227429  -0.0201035662
  7995205 -0.08729420762 -0.06513663859 0.039250865035
  7995215 -0.11062923799 -0.03604434379  -0.0344475417
  7995220 -0.19667841142 -0.02324099286 -0.04627916283
     .           .              .             .
     .           .              .             .
     .           .              .             .
```

6. Cognizants/Contacts

Please contact either of the following people if a problem
arises, or, if products are ready for validation and analysis.
We will be more specifically interested in looking at the
".ps" files (step 5. above) and the contents of the output
logs, (e.g., "LOG_IRAC003700.txt" in step 3. above).

* Frank Masci (fmasci@ipac.caltech.edu), x3154

* Howard McCallon (hlm@ipac.caltech.edu), x1862

* Russ Laher (laher@ipac.caltech.edu), x2596

* John Fowler (jwf@ipac.caltech.edu), x8574