



Publication Year	2016
Acceptance in OA	2020-12-18T11:33:00Z
Title	AVU/BAM: software refurbishment (design and implementation) for the CU3 Gaia verification pipeline
Authors	BUZZI, RAFFAELLA, RIVA, Alberto, Pecoraro, Marco, LICATA, ENRICO, Messineo, Rosario, GAI, Mario, DRIMMEL, Ronald, LATTANZI, Mario Gilberto
Publisher's version (DOI)	10.1117/12.2233417
Handle	http://hdl.handle.net/20.500.12386/29001
Serie	PROCEEDINGS OF SPIE
Volume	9913

AVU/BAM: Software refurbishment (design and implementation) for the CU3 Gaia Verification Pipeline

Buzzi Raffaella^a, Riva Alberto^a, Pecoraro Marco^b, Licata Enrico^b, Messineo Rosario^c, Gai Mario^a, Drimmel Ronald^a and Lattanzi Mario Gilberto^a

^aOsservatorio Astrofisico Torino, Via Osservatorio 20, Pino Torinese (TO), Italy;

^bEuriX, Via Carcano 26, Torino (TO), Italy;

^cALTEC SpA, C.so Marche 79, Torino (TO), Italy;

ABSTRACT

AVU/BAM is the Gaia software for the Astrometric Verification Unit (AVU) devoted to the monitoring of the Basic Angle Monitoring (BAM), one of the metrology instruments onboard of the Gaia Payload.

AVU/BAM is integrated and operative at the Data Processing Center of Turin (DPCT), since the beginning of the Gaia Mission. The DPCT infrastructure performs the ingestion of pre-elaborated data coming from the satellite and it's responsible of running the code of different Verification Packages.

The new structure of the pipeline consists of three phases: the first is a pre-analysis in which a preliminary study data is performed, with the calculation of quantities needed to the analysis; the second one processes the interferograms coming from the instrument; the third phase analyzes the data obtained from the previous processing.

Also it has been changed part of the long-term analysis and was added a phase of calibration of the data obtained from the processing.

Keywords: Basic Angle, Astrometric Analysis software, Interferometer

1. INTRODUCTION

Gaia is an European Space Agency (ESA) astrometric mission, with the goal of achieving the largest, most precise three-dimensional map of our Galaxy, measuring the position and proper motions of more than a billion stars [1][2][3]. These measurements are made through the instruments on board the Gaia satellite, that was launched on 19 December 2013 and now is orbiting in L2 (the second Lagrangian point of the Sun-Earth system). The payload rotates around its axis with a spin of approximately 6 hours and the axis has an inclination of 45° with respect to the Sun. The main instruments that make the measurements on the payload are two telescopes (Astro1 and Astro2) that scan the sky: they have the same focal plane perpendicular to the payload axis. The angular separation of the two lines of sight is the Basic Angle (BA), that is around 106,5° by design, with a sinusoidal variation due to the influence of the sun.

The Basic Angle Monitoring (BAM) is one of the metrology instruments onboard of the Gaia Payload [4]. The main aims of this instrument are two. The first one is to monitor the expected sinusoidal variation, as a time function, of the Basic Angle between the two telescopes and any possible fault. The second one is to provide a calibrated measure of this variation, starting from the raw data and by removing the instrumental contribution from the calculated values. The AVU/BAM is the software for the Astrometric Verification Unit (AVU) that elaborates the measures coming from the instrument [5][6][7][8].

The European consortium responsible to ESA for the Scientific Ground Segment of the Gaia mission is the DPAC (Data Processing and Analysis Consortium), that is organized in nine Coordination Units (CU). These Units operate in six Data Processing Centers (DPC), located in different European countries: Data Processing Center Esac (DPCE) Madrid, Data Processing Center of Barcelona (DPCB), Data Processing Center of Turin (DPCT), Data Processing Center of IoA (DPCI) Cambridge, Data Processing Center of CNES (DPCC) Toulouse, Data Processing Center of Geneva (DPCG). The AVU/BAM software is part of the CU3 and is integrated and operative at the DPCT, since the beginning of the Gaia Mission. The DPCT team has designed and built a structure in which the AVU/BAM software is executed, producing the expected output of the analysis of the raw data; the main results are then stored in the Main DataBase (MDB) while other outputs are stored in the Local DataBase

(LDB)[9].

The AVU/BAM software's main objective is the calculation of the Basic Angle Variation (BAV) as a function of time. The calculation of BAV is based on the analysis of interferometric fringes coming from the two lines of sight. Indeed BAM is an interferometer: starting from a single coherent light source (laser), four collimated beams are created and each pair of beams is sent to each telescope. These pairs of beams create two interferometric images in the focal plane of the telescopes, in the CCD dedicated to BAM. The CCD pixels have dimensions of 10 micron in the AlongScan (AL) direction and 30 micron in the Across Scan (AC) direction. For each pattern, is acquired a window of 1000 (AL) x 320 (AC) pixels, but each sample is defined with an AC binning of 4 pixels, so at the end for each LOS is generated a BAM telemetry packet with a collected window of 1000 (AL) x 80 (AC) samples. These images are used mainly to monitor, through the BAV calculation, the movement of the primary mirror of the respective telescope.

This calculation is performed using two different algorithms, since the variation in each instant of the interferometric images is proportional to the variation of the Basic Angle. The first algorithm calculates the deviations of each single Line of Sight (LOS) with the correlation of each image with a template, defined using a given number of images (i.e.100) from the same LOS. The two deviations are then confronted to have the BAV value. The second one instead calculates the BAV correlating directly two images coming from the two telescopes and that can be considered taken in the same instant (this is performed for 1D images and for the 2D images). A change of the correlation value is proportional to a variation of the BA between the two LOS. We noticed that these different algorithms appear to be sensitive to different factors, so in some way they can be considered complementary. In addition other related quantities are produced, as for example the Fringe Flux, the Fringe Contrast, the Fringe Tilt, the Fringe Period.

The scope of this paper is to present the structural refactoring of the AVU/BAM software, performed in the last year. This refurbishment was basically made to have a more streamlined infrastructure and to decrease the processing time, so that new analyzes could be performed.



Figure 1. Interferometric pattern acquired by the actual Gaia

2. ENVIRONMENT FOR SOFTWARE DEVELOPMENT AND DEPLOYMENT

AVU/BAM is part of CU3, the CU that takes care of the core astrometric processing. This unit processes the raw data coming from the satellite: these data are packed and then ingested by the work packages of this unit. In particular AVU/BAM is part of the Astrometric Verification Unit, which verifies the astrometric results obtained by IDT (Initial Data Treatment @DPCE). In fact the same raw data processed by the IDT are sent to DPCT, that analysing such data and estimating, through the execution of its software, the same quantities calculated by IDT, implements the verification of the baseline process results. The Verification Unit includes three work packages, one for the Astrometric Instruments Model (AIM)[6], one for the Global Sphere Reconstruction (GSR)[5] and one for the BAM[7][8]. All AVU softwares are deployed and executed in the DPCT infrastructure. The code is written in the Java language and is based on particular classes defined in the GaiaTools (GT) and called DataTakers. These classes allow to be attached to the DataTrain, a component in the processing flow, that reads through the data and passes each single data entry for processing. To use the DataTakers, the ingested data from the DPCE must be packaged in special structures called wrappers, that must be defined in the DataModel (DM). They are specific packets that are able to contain a serie of objects, built according to the tables defined in the DM. They are designed to be able to contain both input and output data. The writing of the input data in the wrapper is made through the Connector Library, a DPCT structure that allows to manage the workflow processes. The Connector Library creates the data packets (wrappers) and passes them to the DataTakers relative to each project.

For AVU/BAM the input wrappers contain basically the BamObservations: the pixel signals of the interferometric patterns coming from the CCD dedicated to BAM on the Gaia focal plane. They are from both Fields of View (FOV), and represent a window with size of 1000 x 320 pixels; due to an onboard AC binning however, the images received have the dimension of 1000x80 samples and are encoded in form of matrices of integers. In addition the input wrappers contain the outputs received from the DPCE: these data are the results of the elaborations executed, on the same input data, by IDT.

Throughout the DPCT pipeline, the outputs of the process are written by the code in other wrapper's objects and, at the end of the process, the Connector Library reads these output data and writes them as objects into the MDB and, for the secondary results, in the LDB. As for the wrappers, also all these objects are structured according to tables defined in the DM.

Since the Gaia launch, the AVU/BAM software processes the actual data from the satellite. Before the launch the software has been validated during the various phases, such as OR3 (Operation Rehearsal) and OR4, in which the results and functionality inside the pipeline were evaluated using simulated data. In addition, every time the code changes, a validation is performed using a deployment environment similar to the DPCT's one; through defined Test Cases instead, is executed a scientific validation.

3. CODE REFURBISHMENT

This refurbishment has been decided to achieve a leaner infrastructure and to significantly reduce processing times. Here we consider the features of the AVU/BAM code before and after the refactoring and we make a comparison between the two scenarios. The code involved in this refactoring is related to the daily pipeline.

3.1 Pre-Refurbishment Structure

Originally the code had a linear and incremental workflow. Each wrapper (BamWrapper) contained all data and information relating to a BamObservation received from the DPCE. Then in the course of the process the output calculated serially one after the other, were stored within the same wrapper, so that at each step all the informations were carried. The pipeline infrastructure was based on DataTakers: through the Connector Library the DataTakers related to the AVU/BAM code were called through the setup method and then fed, through the Take method, with data enclosed in the wrapper. Each DataTaker, in turn, fed with the same data the various modules and classes related to it: each module performed its own calculations or its analysis and saved the results in a series of objects, built according to the tables in the Data Model. These objects were added to the wrapper itself. One after other all DataTakers with their related classes were called. All the steps were sequentially performed for all the wrappers. Moreover, all the wrappers were processed twice: the first time to obtain a series of pre-processing parameters and the second time for the process itself. The two phases were organized through a particular DataTaker, called CoordinatorDT, which had the unique purpose to organize the workflow. During the first phase, the CoordinatorDT class allowed access to the DataTakers of the pre-processing, that had the aim to perform a series of elaborations to obtain parameters needed for the real process. These calculations were basically related to the parameters used by the various algorithms for BAV calculation, for example was defined the "template" used for the calculation of the LOS deviation and the indentation number used for the direct correlation between the two LOS, but there was also performed the analysis of temporal discontinuities in the acquisition of images. After sending, sequentially, all the BamObservations to the pre-processing phase, the CoordinatorDT class started the real process by analyzing all the BamObservations again. The call structure of the DataTakers is represented in Fig. 2.

The RDP (Raw Data Processing) module was the core of the workflow. It consisted in the basic processing of the BamObservations coming from DPCE and the calculation of the BAV values through different algorithms, previously selected. In addition were also calculated some other quantities relative to the fringes, like flux, period and contrast. This was achieved through a class that had different versions that allowed to perform calculations in different way, depending on the order and on the quantities required: these quantities were calculated in parallel. The output of this kind of class consisted of a large bidimensional array containing all the calculated values and parameters. All calculated quantities were then setup in the wrapper. After this phase some analysis on the calculated data were performed, like for example the dft analysis, the comparison between different kinds of BAV, the binning of calculated data and, at the end the dataset creation for plots. Also this analyses were done with the same logic: each wrapper was passing through the various DataTakers and related classes and the

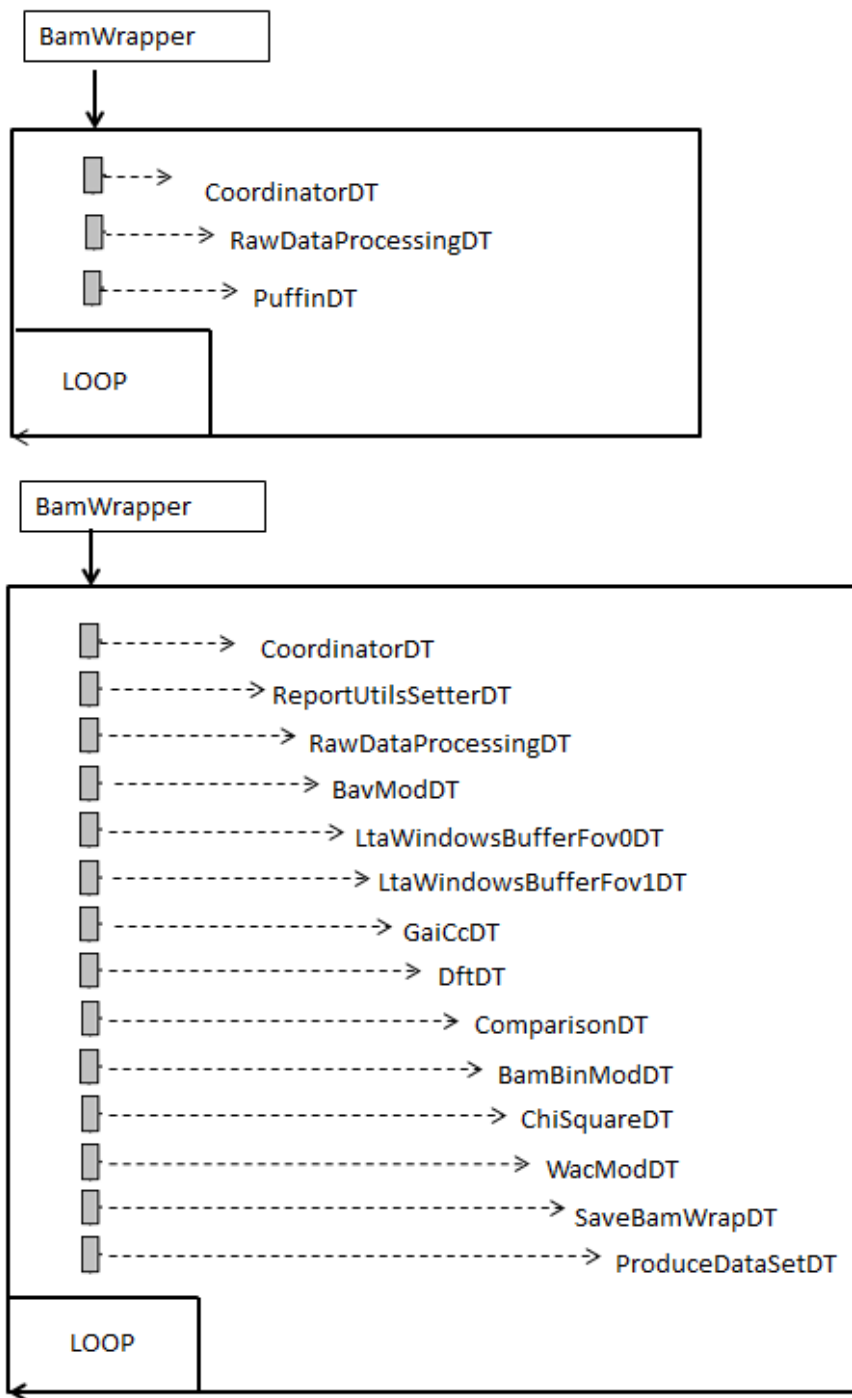


Figure 2. Pre-Refurbishment DataTakers Structure

outputs were stored inside the wrapper itself. Also this step was performed sequentially on all wrappers. Another feature of this pre-refurbishment version is that was used a library called AVU/COMMON, common to all AVU projects. It contained some utility classes made available to all projects of the same environment. In addition, in the daily pipeline there was an analysis over long period data: some statistical analysis and data trend was performed in the Long Term Analysis (LTA) module.

3.2 Post-Refurbishment Structure

Last year's refurbishment was performed to have a more streamlined infrastructure and to decrease the processing time, but also to make the code more readable.

3.2.1 Main Objectives

The restructuring is based on two main concepts:

1. Make the code more readable and the structure less heavy. To obtain this, the classes of the core of process, which calculated the BAV and all the parameters related to the fringes, have been replaced with some more specific modules, structured in a simpler way: one module for each kind of quantity and one class for the calculation of each quantity.
2. Shorten the process time. For this aim the workflow process has been structured so that each couple of images is processed independently from the others. In this way we are able to parallelize the process on multiple nodes. Furthermore, the process has been divided into three stages: at the end of each stage data are saved on the local DataBase and only needed data are passed to the next stage: this for avoiding to carry around all data for the whole process.

3.2.2 Current Implemented Structure

According to the concepts defined above, the workflow has been deeply changed. Now the structure of the daily processing is divided into three phases:

1. FIRPO (FIRstProcessingOperation), the Pre-Processing Phase: in this phase each wrapper (BamFirpoWrapper) contains only one BamObservation and the related data evaluated by IDT; here basically are calculated all the parameters previously calculated in the first phase (old structure). In addition are also performed other calculations that have as unique input the BamObservation (eg. CHISQUARE and the Image Intensity) and is carried out the coupling of temporally correlated images coming from the two lines of sight. Moreover a check for any critical images is performed and data continuity intervals are defined.
2. BANANA (BAMNewANALYSIS), the Processing phase: replaces the core of the previous structure. The BamObservations are processed in pairs (coupling occurs in the previous step). Here is calculated the BAV, using different algorithms and also a number of properties related to the two interferograms: Flux, Contrast, Tilt, Shift in AC direction, etc. We noted that the different algorithms used for BAV calculations appear to be sensitive to different situations: this allows a more accurate analysis and calibration. In this phase each wrapper (BamBananaWrapper) contains a couple of BamObservations, coming from the two FOVs and with the same acquisition time, the related data coming from IDT and the parameters calculated in the previous step and useful to the process. Moreover the heavy super-structure used previously is now replaced by multiple modules: each one takes care of the calculation of one quantity or of some related quantities through a serie of classes devoted to calculate only one of them. So we have:
 - BavCalcDT: is a DataTaker that takes care of calculation of BAV and of the Shift in the AC direction. The first kind of BAV is obtained using a "template" defined in the first phase. This "template" is calculated for each FOV as the mean of the first 100 images coming from the same FOV. Then each image is correlated with the related "template" obtaining the displacement of the image. Then the

BAV is obtained as the difference between the two displacement values calculated for the two FOVs. The second kind of BAV, instead, is obtained as the direct cross correlation of the two images coming from the two FOVs and with the same acquisition time. This is performed for 1D images and for 2D images. Such algorithm is used also for the calculation of the Shift in the AC direction.

- FringeFluxDT: is the DataTaker that calculates the Flux as the mathematical sum of the individual pixel values. This is calculated for both images coming from the two FOVs.
- FringeContrastDT: is the DataTaker that calculates the Contrast as:

$$P = \frac{p_{Max} - p_{min}}{p_{Max} + p_{min}}$$

where p_{Max} is the highest pixel value and p_{min} is the lowest one.

- FringePeriodDT: is the DataTaker that calculates the fringe Period of the interferograms.
- FringeTiltDT: is the DataTaker that calculates the Tilt. This a new feature, that estimates how the images are tilted with respect of the axis of AC direction. Through a correlation of each line of the 2D image with a specific one (the central one) the Shift of each line is evaluated with respect to the others.

In each BamBananaWrapper the output is the results of the process of a single couple of BamObservations.

3. JOE (JointOutputEvaluator), the Analysis phase: a single wrapper (BamJoeWrapper) is created with a series of arrays containing the BAV values, the AC Shift values and the Tilt values of all the couples, the BamElementaries, objects containing all the fringe quantities (Flux, Period, Contrast, etc.) and in addition some parameter coming from the first Pre-Processing phase are copied in the wrapper. Various types of analysis are made on these data:

- A Fourier analysis is performed to extrapolate the main sinusoidal components of some datasets.
- Data are averaged in time intervals of 15 minutes, as design specification, to decrease the influence of noise
- The discontinuities of Chi Square values are analyzed
- The empirical errors are calculated as:

$$\sigma = \sqrt{\frac{\sum_{i=1}^N x^2}{N} - \left(\frac{\sum_{i=1}^N x}{N}\right)^2}$$

- A comparison of the BAV values obtained with the different algorithms is performed
- Data are organized in DataSet used for the plots that will appear in a daily report automatically produced by the pipeline

In this phase the BamObservations are no longer involved, and in the wrapper there is only the output of the processing phase. The actual call structure of the DataTakers for the three phases is represented in Fig. 3.

In addition other two important changes were performed in the software. They don't concern the daily pipeline, but they are related to the time series:

- The Long-Term Analysis (LTA) has been changed: now there is a series of analysis performed over some quantities, using averaged data in 6-hour time intervals (WEEKLY module).
- A calibration in time was added, so that the BAV values are read in function of a chosen starting point.

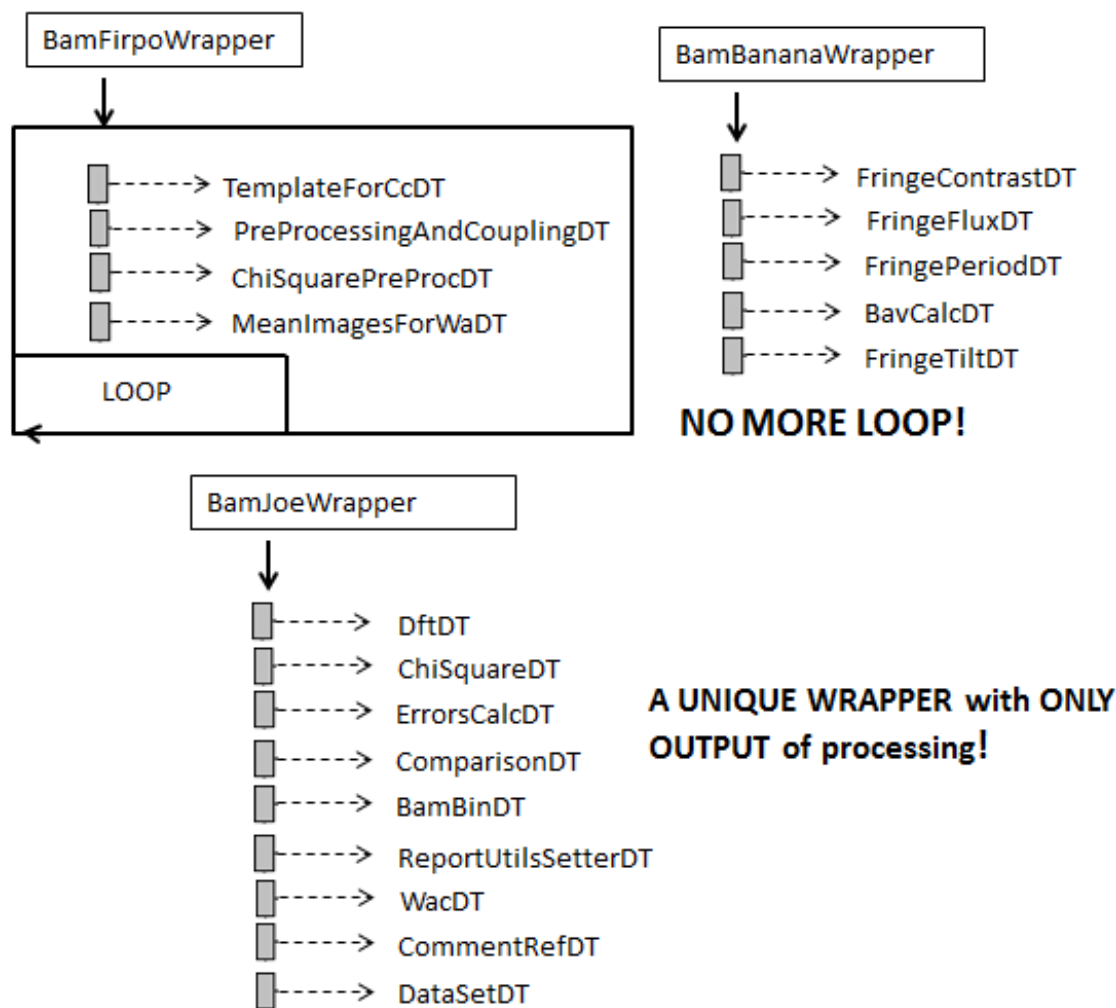


Figure 3. Post-Refurbishment DataTakers Structure

3.3 New Structure Vs Old Structure

The main differences between the actual structure and the initial one are related to:

- **Readability:** now the structure is much simpler and easier to manage with respect to the old one and the code is more readable: especially the structure of the core of the process it was very articulate and with a code difficult to follow. Now, however, the division into individual packages and modules that perform specific calculations allows the use of a code easier to read.
- **Speed:** the pipeline is much faster, despite new processing and analysis were added to the pipeline. In fact, before the refactoring, the whole process lasted two hours, while now a quarter hour is enough, therefore there was a decrease by a factor of eight.
- **Dependency from DPCT:** the old code was more portable, while now is strictly integrated with the DPCT infrastructure, precisely because the present structure involves the creation of three different wrappers, also filled with output of the previous stages and this is done by the DPCT infrastructure.

- AVU/COMMON dependency: the AVU/COMMON library is no longer used. Following the restructuring performed, all the functions necessary for the project have been optimized and embedded in a subset of utility classes specific for AVU/BAM

4. CONCLUSIONS

The Basic Angle Monitoring (BAM) device is an interferometer on-board on the Gaia satellite, devoted to monitor the variation of the Basic Angle between the two telescopes, as function of time. This is performed through the AVU/BAM (Astrometric Verification Unit) java software that is deployed and executed at DPCT (Data Processing Center Torino) in Turin. This software analysis the BamObservations, i.e. the interferometric fringes coming from the satellite. Last year this code has been refactored with respect to the initial structure. This was made for having a simpler and more readable code and especially to decrease the processing time of daily pipeline, so that new analysis could be performed. The result is a more streamlined infrastructure and a significant reduction of processing time, but also a better integration in the DPCT infrastructure.

ACKNOWLEDGMENTS

This work has been funded by ASI under contracts Gaia Mission - The Italian Participation to DPAC, I/058/10/0-1 and 2014-025-R.1.2015

REFERENCES

- [1] de Bruijne, J. H. J., Rygl, K. L. J., and Antoja, T., “Gaia astrometric science performance - post-launch predictions,” *EAS Publications Series* **67-68**, 23–29 (2014).
- [2] Timo, P., “Gaia mission status,” *EAS Publications Series* **67-68**, 15–21 (2014).
- [3] de Bruijne, J., Kohley, R., and Prusti, T., “Gaia: 1,000 million stars with 100 ccd detectors,” *Space Telescopes and Instrumentation 2010: Optical, Infrared, and Millimeter Wave*. **7731**, 77311C (2010).
- [4] Mora, A., Biermann, M., Brown, A. G. A., Busonero, D., Carminati, L., Carrasco, J. M., Chassat, F., Erdmann, M., Giesen, W. L. M., Jordi, C., Katz, D., Kohley, R., Lindegren, L., Loeffler, W., Marchal, O., Panuzzo, P., Seabroke, G., Sahlmann, J., Serpell, E., Serraller, I., van Leeuwen, F., van Reeve, W., van den Dool, T. C., and Vosteen, L. L. A., “Gaia on-board metrology: basic angle and best focus,” *Proceedings of the SPIE* **9143**, 91430X (2014).
- [5] Vecchiato, A., Abbas, U., Bandieramonte, M., Becciani, U., Bianchi, L., Bucciarelli, B., Busonero, D., Lattanzi, M. G., and Messineo, R., “The global sphere reconstruction for the gaia mission in the astrometric verification unit,” *Software and Cyberinfrastructure for Astronomy II. Proceedings of the SPIE* **8451**, 84513C (2012).
- [6] Busonero, D., “Astrometric instrument modeling in the context of gaia astrometric verification: tasks and activities,” *Modeling, Systems Engineering, and Project Management for Astronomy V. Proceedings of the SPIE* **8449**, 84490F (2012).
- [7] Riva, A., Lattanzi, M. G., Drimmel, R., Gai, M., Busonero, D., Buzzi, R., Pecoraro, M., Russo, F., and Messineo, R., “The basic angle monitoring (bam) software tool in the context of gaia’s astrometric verification,” *Proceedings of the SPIE* **9150**, 91501Z (2014).
- [8] Riva, A., Gai, M., Lattanzi, M. G., Russo, F., and Buzzi, R., “Bam: A metrology device for a high precision astrometric mission,” *III Workshop on Robotic Autonomous Observatories, Revista Mexicana de Astronomia y Astrofisica (Serie de Conferencias)* **45**, 35–38 (2014).
- [9] Martino, M., Ciampolini, A., Messineo, R., Mulone, A., Pigozzi, E., Icardi, V., Solitro, F., Castronuovo, M. M., Lattanzi, M. G., Morbidelli, R., Drimmel, R., Sarasso, M., Busonero, D., Crosta, M., Gardiol, D., and Vecchiato, A., “The italian contribution to the gaia data processing and archiving,” *The Astrophysics of Planetary Systems: Formation, Structure, and Dynamical Evolution, Proceedings of the International Astronomical Union* **276**, 539–541 (2011).